Andrew Plum

Professor Nguyen

MATH 183

4/24/2023

Semester Project Time Log

Sessions:

1. I had done some general exploration before this next log.
2. Started 4/15/2023 at 4:17 pm and finished 4/15/2023 at 7:53 pm.
   1. Read the papers on data analysis recommended by you. Did further internet research.
   2. Looked over the dataset.
3. Started 4/24/2023 at 8:30 pm and finished 4/24/2023 at 11:57 pm.
   1. Made jobs by location heatmap of USA
      1. Finished 4/24/2023 at 10:49 pm
      2. It took me way longer than I wanted it to. I ran into an issue px.choropleth with plotly.express as px. I was initially very confused because I thought the parameters I input should have worked, they were state names, but instead they needed to be the abbreviations of state names. It took me a bit to realize this because no error was appearing so I was in the dark trying to solve it. I looked for a solution online; eventually I found it by reading the documentation of the function. It then took me a bit more time to code it up.
   2. More data exploration
4. Started 4/25/2023 at 5:36 pm and finished 4/25/2023 at 8:11 pm.
   1. Graphing with the jobs by location dataset
      1. Bar graph
      2. Histogram
      3. Pie graph
      4. Scatter plot
   2. US Census data at 7:44 pm
      1. Read article about getting US census data
         1. Had to get a key
      2. Tried to import data to notebook – was unsuccessful
5. Started 4/29/2023 at 11:07 am and finished 4/29/2023 at 3:05 pm.
   1. Got the US census data working
      1. I had to read some documentation and watch a video
      2. Got population and merged that into the main dataframe I was working with and tried to format it for stuff from there.
6. Started 5/1/2023 at 2:15 pm and finished 5/1/2023 at 10:41 pm.
   1. Met with Xavier to further discuss what we have and what to do next
   2. US Census Data
      1. Made a few more graphs with population and jobs per 1000 people at 4:52 pm
         1. I had to calculate jobs per 1000 people and add as a column
      2. Tried to get population density by state from US Census API at 6:43 pm
         1. I fixed a few things here and there that I noticed
         2. I was stumped by a weird issue where the data in JSON format did not want to be stored in a dataframe
            1. **JSONDecodeError**: Expecting value: line 1 column 1 (char 0)
            2. I didn’t know if I was messing up getting the data from the API or if my conversion function wasn’t working properly. Was weird because I was able to get the population by state data fine at this point.

Sometimes it worked for some reason, but it was inconsistent

Eventually, I realized that there was maintenance occurring with the website and that was possibly why it wasn’t working so I decided to stop at 10:41 pm.

It turns out that the maintenance was causing the problem.

1. Started 5/2/2023 at 6:22 pm and finished 5/2/2023 at 8:32 pm.
   1. Made graphs
      1. Used population density, population, job count, jobs per 1000 people and their respective rankings.
   2. Spent time trying to fix a problem where the ranking for the population was wrong.
      1. I had to swap a few lines of code around to fix it.
2. Started 5/2/2023 at 10:37 pm and finished 5/3/2023 at 1:09 am.
   1. I worked on the Job Posting Samples dataset
      1. I extracted the job title frequency into a dataframe
         1. Put job title strings in list
         2. Unique it
         3. Counted frequency
      2. I graphed it
         1. Bar graph
         2. Scatter plot
3. Started 5/6/2023 at 5:13 pm and finished 5/6/2023 at 10:26 pm.
   1. I continued work on the Job Posting Samples dataset
      1. In the “job\_titles\_df”, I broke down the job postings by the state they were posted in.
         1. Had to use regular expressions to extract the location of the job posting.
         2. I was able to draw conclusions about the industries in each state with this dataframe
   2. Using both “job\_titles\_df” and “modified\_census\_with\_DC\_df” I drew conclusions about the industries of the top 5 states ranked by Jobs Per 1000 People and the top 5 states ranked by Job Count
      1. I plotted some bar graphs to help illustrate the findings
4. Started 5/7/2023 at 6:05 pm and finished 5/6/2023 at 10:03 pm.
   1. Worked on presentation.
   2. Combined the notebooks.
   3. Finished documentation.
5. T
6. Future Ideas
   1. Census
      1. Graphs
      2. Breakdown industry by geography
   2. Fed data
      1. Jobs in each state by gdp
      2. Use regular expressions to gather relevant qualitative data on skills needed for data science.
         1. Graph it
         2. Are there trends to what’s most useful
   3. Machine learning
      1. Get to if you have time, use it for the industry growth projection

Questions/Reminders:

1. Assignment #9
2. Extra Credit
3. Project
   1. Scatter plot jobs per capita by population density?
   2. Should I only leave graphs in the notebook with key findings
   3. What else should I do? Does what I have so far look good?
4. Lightcast Palantir
   1. EMAIL HIM MENTIONING THAT AT THE END OF THE YEAR
5. Email Xavier notebook

I have a pandas dataframe named "modified\_job\_postings\_df"; one of the columns is "Job Title" which stores string objects and the other is "Location" which also stores string objects. In the "Location" column the string starts with the city name, a comma followed by a space, and then the capitalized abbreviation of the state; here are three examples: "Calabasas, CA", "Jacksonville, FL", "Springfield, MO". I also have a dataframe named "job\_titles\_df" which has a column named "Job Title" which stores the unique string objects, stored in the "Job Title" column of the "modified\_job\_postings\_df" dataframe. Add a column for each of the 50 states. Name the column based on the states names capitalized abbreviation; for example California is CA. There should be 50 columns added each of them should store integers and store elements starting at zero when created. Next, look at the "Job Title" column of the "modified\_job\_postings\_df" dataframe, and based on the string object stored in the column, on the same row of the dataframe, look at the location column and use regular expressions to extract the capitalized state abbreviation. If the string object is "Calabasas, CA" the extracted string object extracted using regular expressions would be "CA". Based on the extracted capitalized state abbreviation, go to the "job\_titles\_df" dataframe and add one to the element in the column that is named the same as the capitalized state abbreviation and whose row has the same string object stored in the "Job Title" column of the "modified\_job\_postings\_df" dateframe as the same string object stored in the "Job Title" column of the "job\_titles\_df" dataframe. For example, if I have in the "modified\_job\_postings\_df" on the first row an string object in the "Job Title" column that is "Data Scientists" and on the same row the "Location" column has a string object that is "Calabasas, CA", then in the "job\_titles\_df" dataframe, in the element on the row that reads "Data Scientists" and the column that reads "CA", you are going to add +1 to the integer already stored there.