Steve Horvath

Alex Talento

Elizabeth Klink

Data Pipeline Between Candidate Donations and Home Values

**EXTRACT:**

The first data source our group extracted came from the candidate campaign donations from the FEC database. We utilized the Splinter and Beautiful Soup Python libraries to navigate and the parse the FEC data. Splinter allowed us to navigate to the website and export the data to the local machine. Once the CSV file is saved on the local machine, we created variables to save the file locations of each dataset. We then utilized Pandas to read in the dataset and allow us to make the necessary transformations.

The second data source our group extracted came from the Zillow database. We managed to locate the CSV file via Inspect in Google Chrome to allow us to automatically read in the CSV via Pandas. We did not need to dynamically extract this data for future use because the filters applied on the data set are statically updated and therefore the CSV will not change for future iterations. Once loaded into Pandas, it allowed us to transform the data.

**TRANSFORM:**

Transforming the election data, we selected only the necessary columns for analysis which include Candidate Name/Committee, the Zip Code, and Contribution Amount. The zip code we had to change to a string in order to split out the unnecessary values and only hold the first five digits. We dropped any values that contained no data and dropped any zip codes that were labelled “0.0”. Since we had 3 different data frames to account for the large data sets between all the candidates, we combined all 3 via concat method in Pandas to merge into one data frame. We now have one cleaned dataset containing all contributions for each Presidential candidate by zip code.

Transforming the housing values data set called for removing all the monthly home value data points to only 2019 figures. This means data between January 2019 and June 2019 only. In order to get average home value for 2019, we added up the available amounts from January through June and divided by 6. We created a new column “Avg 2019 Home Value” to hold this data. We then formatted the numbers to currency via formatting. The home values data frame is now ready to be loaded into a database

**LOAD**:

Our group decided to load the data sets into a Postgres SQL database. The relational database aspect made sense since we can group our data by zip code between the tables. We started by creating a connection string that connects to Postgres and creates the database we named “campaign\_db”. We then created engine to facilitate the creation and loading through Python. We load the database into Postgres using the Pandas to\_sql method and append it to the newly created database. To test it has been successful we run the Pandas method pd.read\_sql\_query and confirm via SQL query that the data is loaded and ready for analysis