Car Sales and Emissions

Our project was to extract two sources of data into a Python kernel, transform the sets of data to fit our liking, and load them into an online database. For our sets of data, we chose a CSV file from kaggle.com that showed the sales of many different car manufacturers, and another CSV file from data.world that looked at fuel emissions from each car manufacturer and model.

After saving each CSV file locally, we then extracted them into a jupyter notebook. Inside of our jupyter notebook, we imported Pandas and SqlAlchemy so that we could start restructuring our CSV files into a more desirable format and load them to Postgres.

To transform the data sets, we selected only the necessary columns from the larger dataframes and renamed any columns to make the ‘joining’ process easier. We then grouped each dataframe by the manufacturer, using the average value of all columns with numeric values. One of the challenges we came across was that the two dataframes did not have an equal amount of rows. To amend this, we first created two empty lists, one for each dataframe. We then looped through both dataframes and checked to see if the manufacturer from the first dataframe was also in the second dataframe. If so, we appended the values in that row from each dataframe into their respective lists. After the loop was completed, we had two lists of values from each dataframe that had an equal number of manufacturers. Finally, we created two new dataframes from these lists.

We chose to load our dataframes into PostgreSQL so that we could join them together and create a view out of the joint table. First we had to import create\_engine from SqlAlchemy so that we could connect to Postgres. Next, we went into Postgres to make our database and create our tables, making sure that the two tables had the exact same column names from our Pandas dataframes. Once our tables were created in Postgres, we used the Pandas function .to\_sql to send our dataframes to our SQL database. Within SQL, we joined the two tables along the ‘manufacturer’ column to create one table with all of the columns from the two previous tables.