**Databootcamp - Project 2 – Final Report**

**Extract:**

We were able to locate data on car accident data for the following:

* United Kingdom, 2005 to 2015 <https://www.kaggle.com/tbsteal/canadian-car-accidents-19942014#NCDB_1999_to_2014.csv>
* Canada, 1999 to 2014, <https://www.kaggle.com/silicon99/dft-accident-data>

All files in the repositories were csv.

Due to the limitations we found with kaggle, we went the route of downloading the data directly from kaggle, and hosting locally on our machines, as well as on GitHub.

Having to store the data on GitHub, meant that we needed to also download GitLFS, which allowed us to upload the large files.

Once we had a firm grasp on the data, we loaded the necessary csv files into Panda Dataframes to be used for transformation purposes.

**Transform:**

In general, we saw that both the Canada and UK had sufficient data to be sufficient in terms of a population, but also provided enough content which would allow us to do any future analysis.

Regardless if it was Canada data, or UK data – csv’s were loaded the same way, but to ensure information was consistent, and streamlined, for each Canada and UK, we did some necessary merges so all data was centralized.

From here, we then roughly determined which pieces of data were consistent between both the UK and Canada datasets. Doing this allowed us to save a lot of time in cleanup, as it allowed us to simply shave any columns which were unnecessary/inconsistent. To accomplish this, we dropped many columns.

Next, we needed to ensure that date formats were aligned. In our case, we wanted to ensure that we could track by year, and by month. For the UK data, we used DatetimeIndex to first convert the date format, at which point we extracted the Year and Month and placed them in their own columns. This way, it would match what was on the Canada data already.

Doing this also allowed us to open up the option to determine the year which the vehicle was made (as this was present in the Canada data, but not UK data). This was done by simply subtracting the age of the vehicle, from the year of the accident, and placing the value within its own column.

The next step from here was to ensure that we had consistent domain values for all our like columns. From there we needed to spend some time together to determine how to best categorize each value, and give consistent wording between both data sets.

This meant we needed to write dictionaries, and then map these values into our main datasets for each country.

For example, under column “Gender” it would show values “1”,”2”, or ”3.” When reviewing the additional csv file called “Gender,” you would then see that the values above represented “male”, “female”, and “undisclosed” accordingly.

From there, we decided to export our new clean, concise, dataframes into csv so that we could load the data appropriately.

**Load:**

We loaded the data into a Postgres relational database. We chose to do this because our data was already structured in a way in which we did not need to utilize the unstructured design of noSQL databases.

To do above, we first created a create\_db.py file which housed the function to create the database in Postgres for the user if it does not already exist. A connection is then made to the database.

The final data is loaded into dataframes from the staging folder for both the UK and Canada data. It is then pushed to the collisions database created earlier via the to\_SQL function in Pandas.

Finally, it is loaded back into a dataframe to test whether it was loaded successfully.