Ashwin Gnanam

File/IO Report:

Overview of Data:

This dataset was created by Doaa Alsenani on Kaggle. It is composed of data scraped from https://www.auctionexport.com/ and it includes information about 28 brands of new and used cars being sold in the US. The file itself is in the csv format with fields of *number, price, brand, model, year, title_status, mileage, color, vin, lot, state, country, condition*. This data is used by auction export to organize and filter through the various cars they have listed on their site using certain features such as price, condition, brand, etc.

Data Structure Description:

The general data structure used in this project is as follows:

```
{price range: [{index number of car: [precise price, brand, model, year,
colour, condition, mileage, vin number, (state, country)]}]}
```

Example from dataset:

```
{'$0-5k': [{'0': ['6300', 'toyota', 'cruiser', '2008', 'black', 'clean
vehicle', '274117.0', '  jtezu11f88k007763', ('new jersey', ' usa')]}]}
```

This data structure helps with my idea for a future application. Using this format it would be very easy to show the user a graphical representation of the data or filter by different keys to receive a list of all vehicles within a certain price range and then being able to select different entries based on their "number" value. I can imagine doing this by using input() functions and getting the user to choose the price range and then use arrow keys to look at different cars that fall under it.

* *Note: due to the size of the data, the dictionary that holds all the data is too large for my vscode terminal to render. I wrote the dictionary to a file just in case you have a similar issue.*

Possible Application:

As mentioned above, this data structure can easily be parsed into a graphical representation for the user. This could be a table, a graph, or even a full application gui. In terms of ui, there are three main ways I could build an interface that users can use to access the data structure. I could use the built in input() function to make a basic terminal-based application where the user would input numbers or letters to denote their selection when browsing. I could also use the pygame module to create a full gui so that users will be able to access values more intuitively. Another option is using python django to make a web application that users can use to access the data online. A consequence of generating the data from a csv file is that it can be easily updated. This will suit the application because it can be kept up to date without much work.