Data Preparation

After reading the data using pandas read_csv function , I did not get the data in readable format because the separator was not ',' it was '#'. Hence had to change the 'sep' attribute in read_csv to '#'. Since there was no header in the given csv data, I had to specifically give the names to each column to provide some meaning to this data. I did this by giving it in the attribute 'names' in the read_csv command.

After getting the data in readable format ,I first trimmed and converted the whole data to lower case for making the data consistent. I did not use the command 'data.dropna()' because I did not want to lose important data.

I used the below code to get the original data set for comparing values with the given data set

```
#code used for viewing online original data set.

import urllib2

url  = "https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data"
request_url = urllib2.Request(url) # requesting connection from the url
original_data = urllib2.urlopen(request_url) # opening the url
# to read data from the URL
orgdata = pd.read_csv(original_data, sep=',', decimal='.', header=None,
                      names=['Symboling','normalized_losses','make',
                          'fuel_type','aspiration','num_of_doors','body_style','drive_wheels',
                          'engine_location','wheel_base','length','width','height','curb_weight',
                          'engine_type','num_of_cylinders','engine_size','fuel_system','bore',
                          'stroke','compression_ratio','horsepower','peak_rpm','city_mpg',
                          'highway_mpg','price'],chunksize=300)
piece = orgdata.get_chunk(1000) #to get the chunk of data from the data set
piece #to print the data
```

This code was modified as we want to view the original data set. Eg, filtering the data based on car manufacturer, filtering data based on a particular value etc. Another way to view and find values in the dataset is to use excel for filtering and finding values.

Firstly, when we take Symboling column into consideration the value should be in between -3 to +3. In the given data set there were three values which were '4' so had to either remove from the data set or change its value to the corresponding value from the original data set. Here I found a similar value in the original data set hence I changed it to -1 to prevent losing valuable data.

For the normalized losses column, we have 47 null values in the given data set and also in the original data set we have 41 NaN values. So the safest way to fill up the values for this column was to find the mean of the column, instead of removing them. Initially I filled all the NaN values with 0.0 and then replaced the 0's with mean. I made sure that I don't calculate the mean with the 0's replaced by NaN values.

Later, I removed all the NaN values present for the number of doors column as I had no values in the original data set to match the same in the given data set. I chose to remove because it is not correct to incorrectly assume the number of doors and also there were only 1 row with NaN hence removed it.

There were 4 NaN values and 6 NaN values in price column. Apparently, 6 NaN valued rows were similar to two rows in the original data set. After careful observation, I found out that all the NaN valued manufacturers were volvo and one of the rows with fuel type diesel matched to one of the rows in original data set hence changed the price to the corresponding value. For the other 3 rows there was a row that matched in the original dataset hence I replaced these three rows with the corresponding price, there by not losing any data in the progress. The other 4 NaN valued rows were removed because we couldn't add average price or assume values hence, removing them was the right choice.

To fix the typos, I found all the misspelled words in the data and replaced them with the correct ones using replace function which can be applied to string type data types. Firstly, in the 'make' column I replaced all the 'vol00112ov' and changed it to volvo. Secondly, I changed all the 'turrrrbo' in aspiration column to 'turbo' to maintain proper consistency. Lastly, I replaced all the 'fourr' to 'four'.

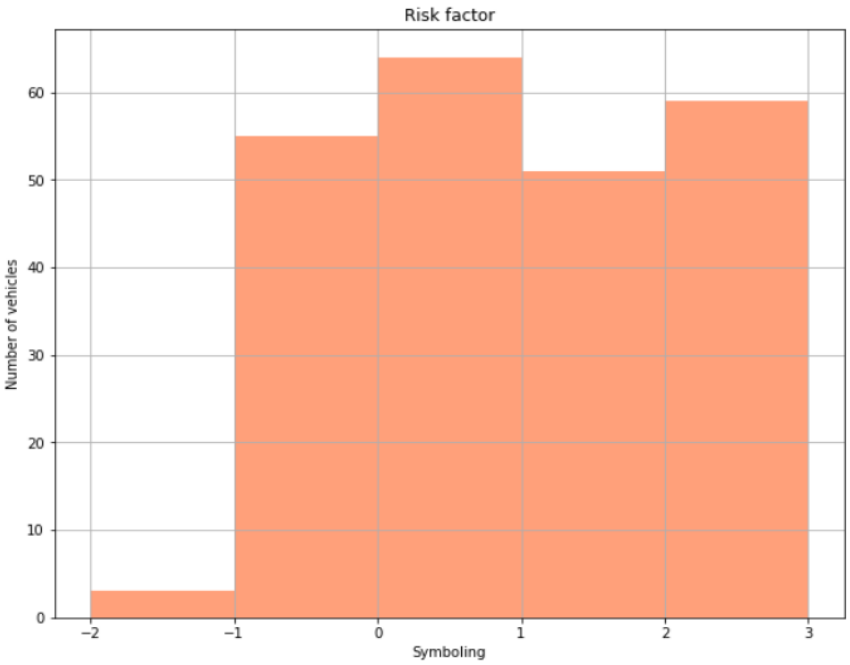
Then I checked other variables for outliers on excel by checking range of the variables in the excel sheet. All the continuous variables in this data set do not start from 0 so we have to make sure that all the data is in between the specified values as mentioned in the UCI website. If there were any outliers we have to remove

them to achieve data consistency. Since there were no impossible values the data we obtained after all the above processes is not clean and ready for data exploration.

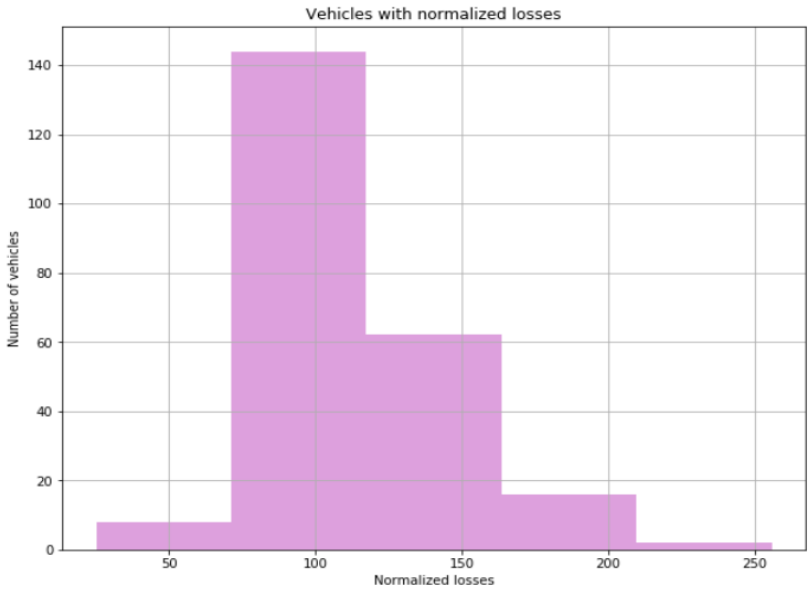
Data Exploration

Subsection 1:

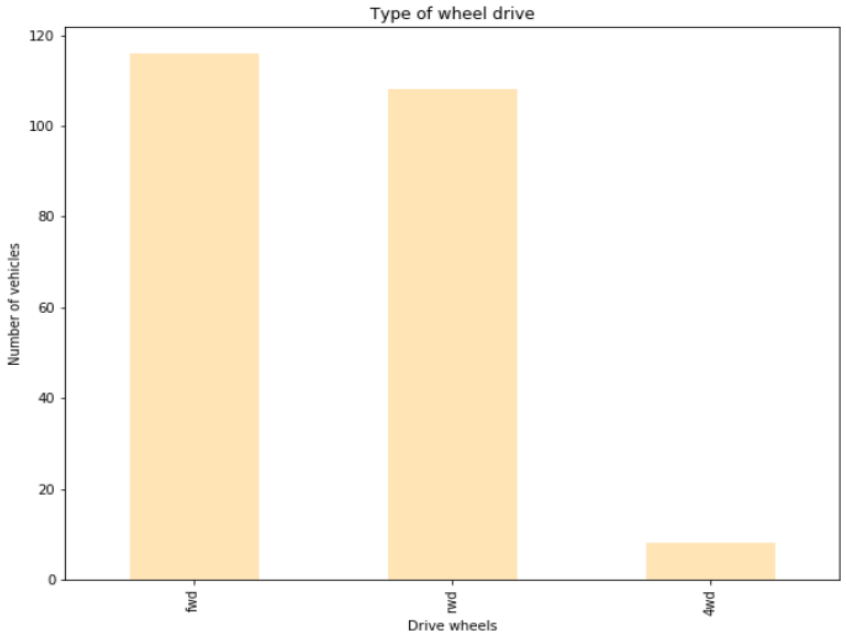
1. Using Symboling column as the **Ordinal** variable as each value in this column is categorical and it also has order(rank). This plot describes risk factor with number of vehicles which fall under this category. It basically shows the how many vehicles falls under various risk rating.



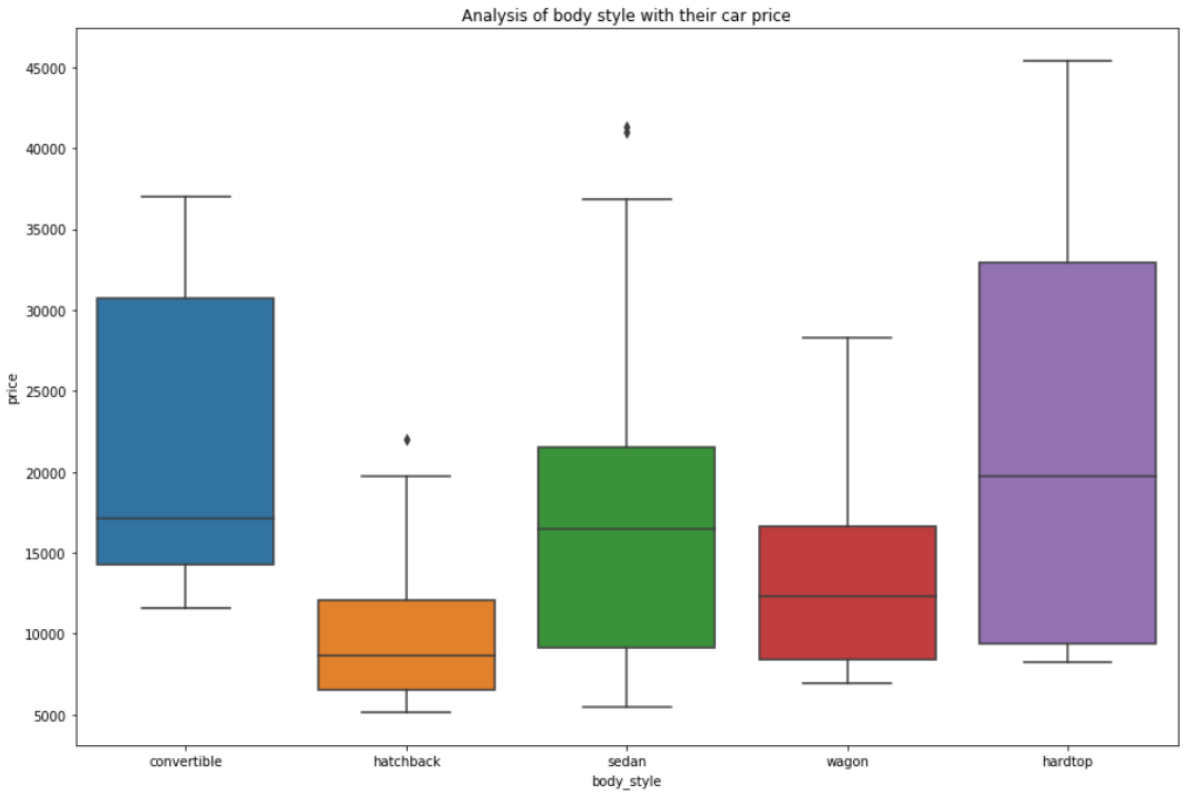
2. This plot depicts normalized losses in use as compared to other cars with the number of vehicles which fall under this factor. I chose this variable as the **numerical** variable as it is continuous and discrete.



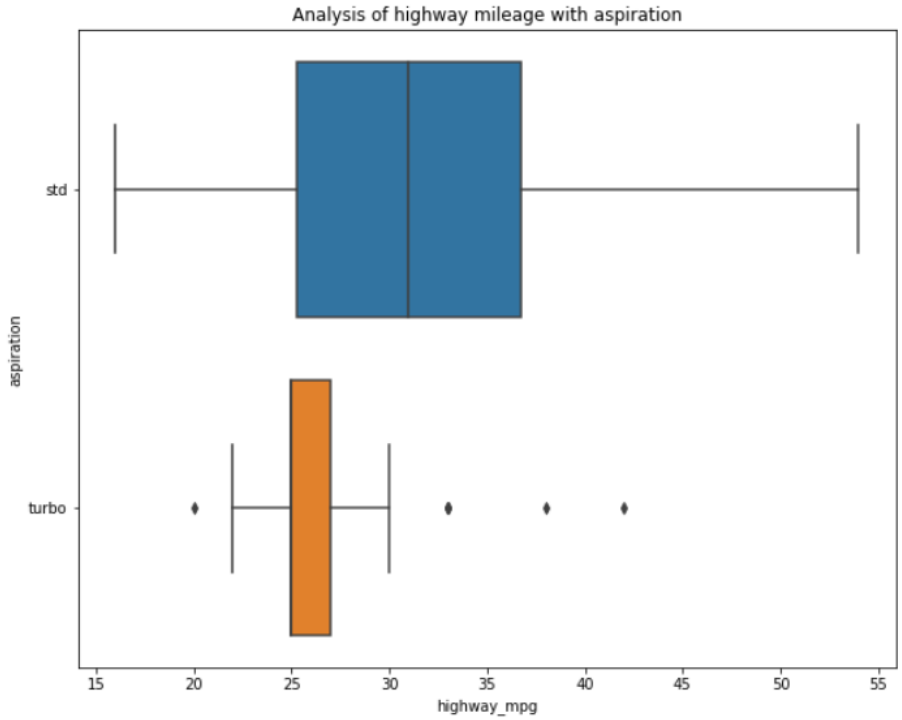
3. I chose Drive wheel as **nominal** variable as it is categorical with no order or rank. This plot describes the number of vehicles manufactured based on the type of wheel drive i.e whether they are front wheel drive or rear wheel drive or four-wheel drive.



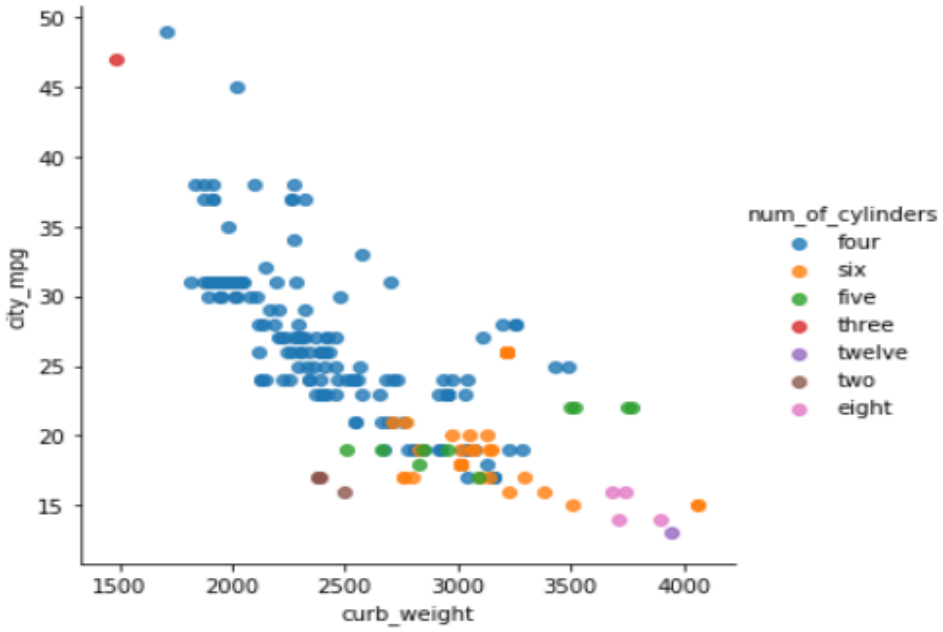
Subsection 2:



1. This graph describes the body style versus the price. From this graph we can suggest a buyer that the cheapest style of car is hatchback. Hardtop are the body which have the most widest price range. From the graph we can also make out that convertibles price starts only at 14700, which is actually higher than the costliest hatch back. The costliest car has the body type hardtop.

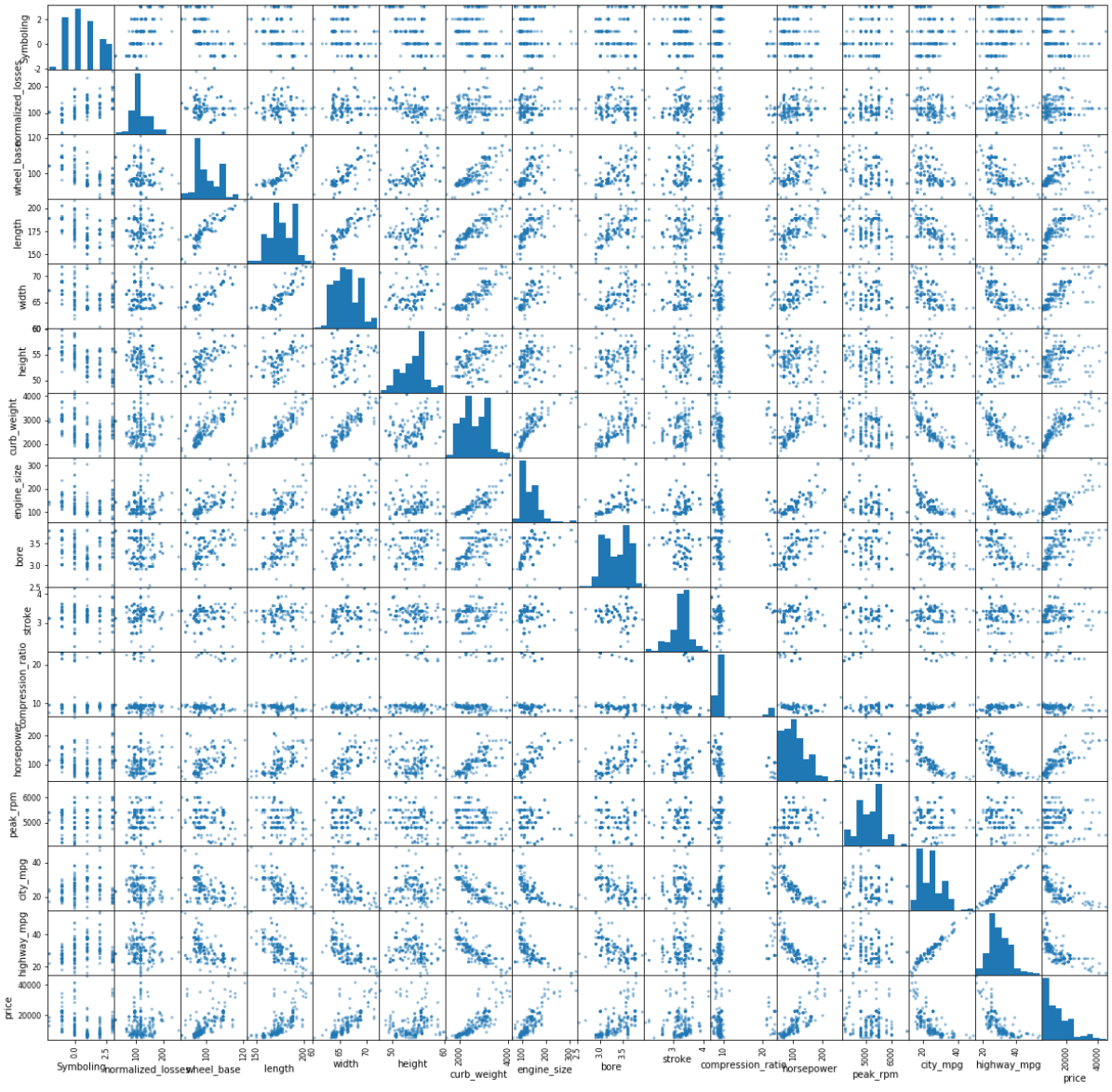


2. The above graph depicts the analysis of highway mileage with that of aspiration. We can get to know that the standard aspiration cars on an average have better highway mileage. Though both type of cars start off from 25, standard aspiration cars have a much better average mileage than turbo cars. But there is one car in standard aspiration which has even lesser mileage than the least mileage car with turbo engine. On the other hand we also have a standard car with mileage of almost 54,which is almost double the highest mileage of a turbo car.



3. The above graph depicts how the curb weight of the car affects mileage in city limits. We can see that we are actually comparing the cars with different cylinder types while formulating a relation between curb weight and cuty mileage. We can incur from this graph that four cylinder car is by far the best car to choose when we want a mileage car for city commutes. We can also conclude that more the weight of the car less is the mileage of the car, hence this suggests weight od the car is directly proportional to mileage.

Subsection 3:



This type of scatter plots are very useful when formulating multiple regression analysis. This graph contains all the pair wise scatter plots of all the continuous or numerical variables from the dataset.
Scatter plots are good for determining rough liner correlations of metadata of continuous variables and are not so good for depicting discrete variables. We can figure out that wheel base harmonizes with length and width of the car and also price harmonizes with engine size and weight of the car.

References:

1. https://seaborn.pydata.org/index.html : For seaborn related plots.
2. https://plot.ly/python/ : Using plotly in python
3. https://www.r-bloggers.com/scatterplot-matrices/ : Scatter Plot Matrix
4. https://matplotlib.org/examples/color/named_colors.html : Colors for graphs
5. http://archive.ics.uci.edu/ml/datasets/automobile : For data related queries