SkillHives Data Analysis Challenge

-Lokesh Sangewar
IIM Indore.

Aim and Objective:

An automobile dataset was provided. The aim of the task was to perform exploratory data analysis (EDA) on the dataset, either on R or Python.

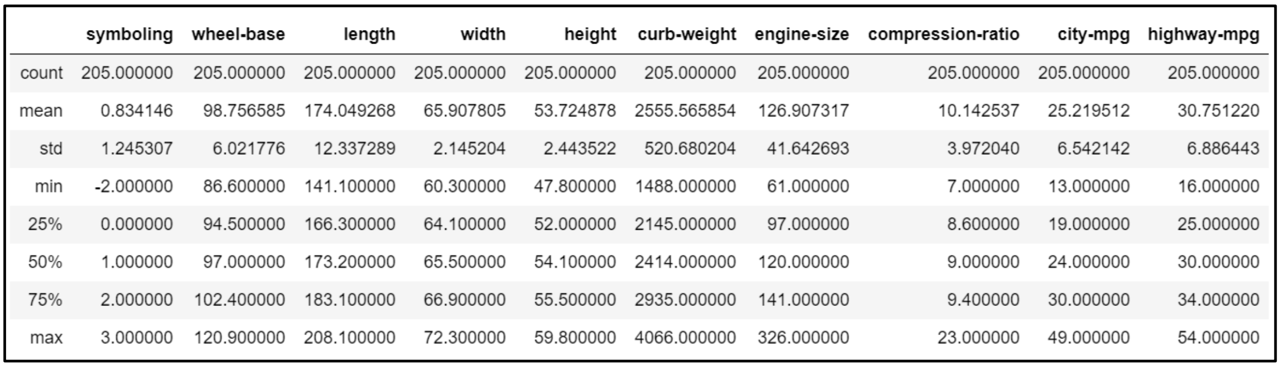
Language used: Python 3.6.2.

Hardware and Software Requirements: I used Jupyter Notebook to write the code. The submission file is an .ipynb file. Hence, necessary tools, at least the Jupyter Notebook, will be required to run the same.

Explanation:

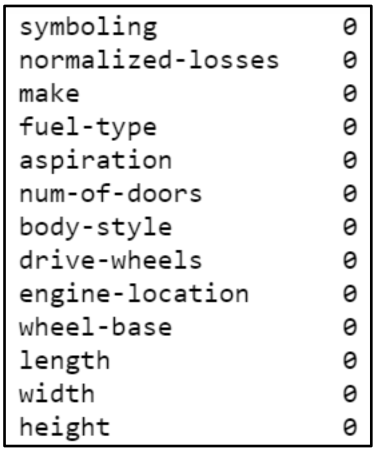
The dataset provided was an automobile dataset. The aim of the task was to perform exploratory data analysis on the same.

The dataset was first loaded, and then the head() command was used to have a glimpse at the same. Descriptive statistical analysis was performed. The results are as follows:

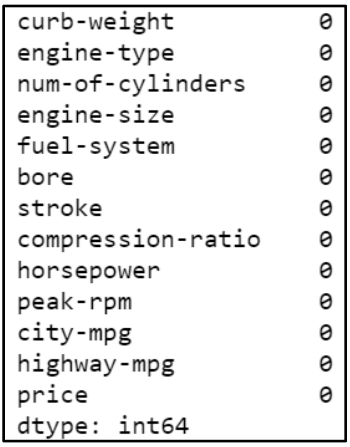


	symboling	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	city-mpg	highway-mpg
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	10.142537	25.219512	30.751220
std	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	3.972040	6.542142	6.886443
min	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	7.000000	13.000000	16.000000
25%	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	8.600000	19.000000	25.000000
50%	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	9.000000	24.000000	30.000000
75%	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	9.400000	30.000000	34.000000
max	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	23.000000	49.000000	54.000000

We observe, the average mileage offered by the vehicles on the highway was 30.75 while in the city was around 25. I then checked if there were any null values.



```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
aspiration           0
num-of-doors         0
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
```

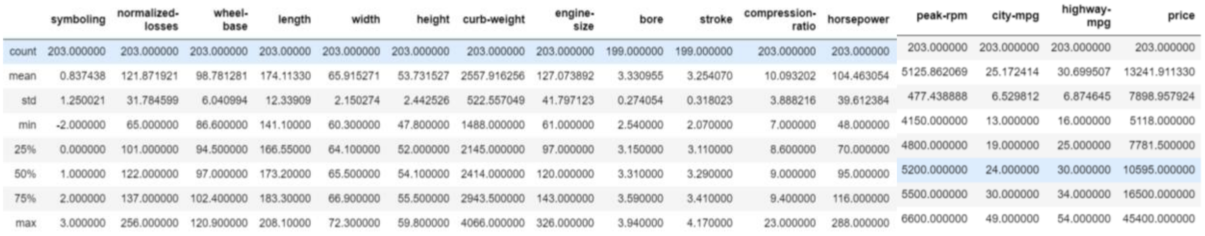


```
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 0
stroke               0
compression-ratio    0
horsepower           0
peak-rpm             0
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

We observe, at first glance that there are no missing values. However, I did observe certain ‘?’ marks in the dataset. We can treat those as missing values. And hence, to deal with them, we will have to clean the data.

The missing value treatment given to most of the columns was to impute them with the mean value of the columns. The variables where missing values imputation with mean was required were *normalized-losses*, *horsepower*, *peak-rpm*, and *price*. Additionally, there were ‘?’ marks in the *num-of-doors* column too. However, mean imputation in this case would imply replacing the ‘?’ marks with 3. But in general, there exist, as of now, no vehicles with 3 doors. Hence, it would not make sense to perform mean imputation. Hence, for simplicity, these values were deleted. After all the columns were dealt with, and the rest of the data cleaned, missing values were checked again, and I found out there were 4 missing values in *bore* and *stroke* columns respectively.

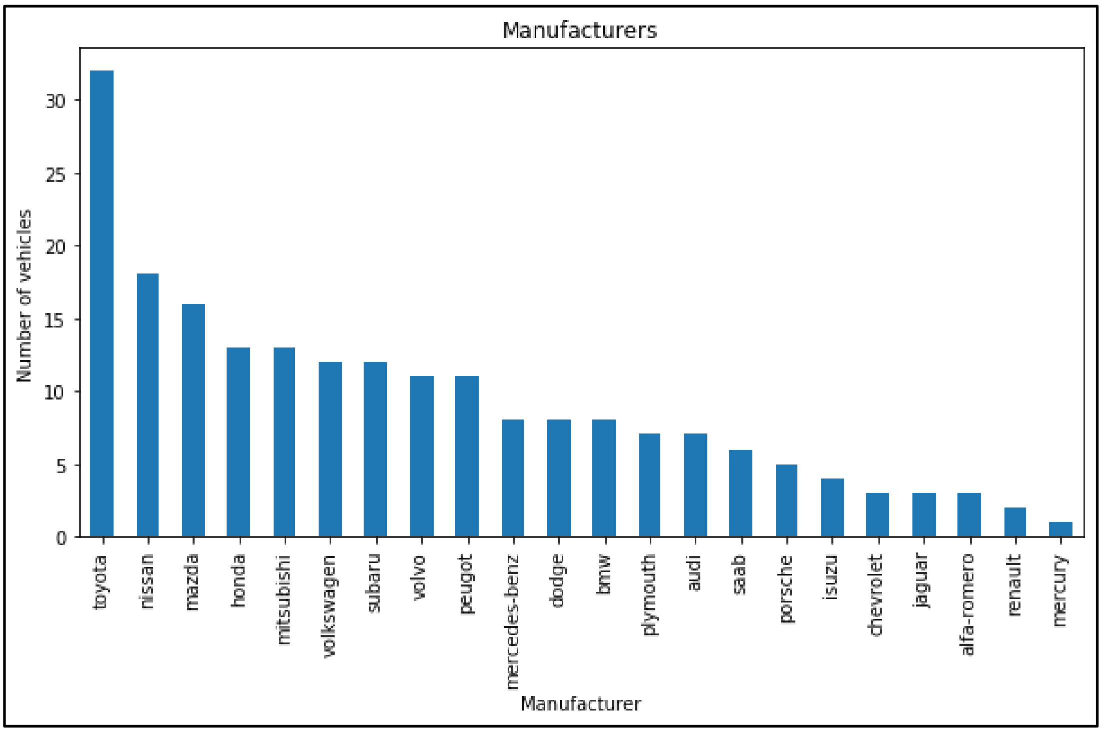
Descriptive Statistical analysis was again performed which gave the following result:



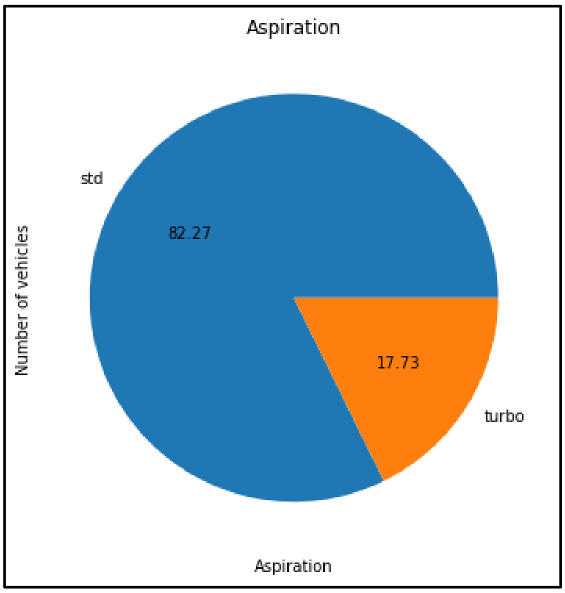
	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
count	203.000000	203.000000	203.000000	203.00000	203.000000	203.000000	203.000000	203.000000	199.000000	199.000000	203.000000	203.000000	203.000000	203.000000	203.000000	203.000000
mean	0.837438	121.871921	98.781281	174.11330	65.915271	53.731527	2557.916256	127.073892	3.330955	3.254070	10.093202	104.463054	5125.862069	25.172414	30.699507	13241.911330
std	1.250021	31.784599	6.040994	12.33909	2.150274	2.442526	522.557049	41.797123	0.274054	0.318023	3.888216	39.612384	477.438888	6.529812	6.874645	7898.957924
min	-2.000000	65.000000	86.600000	141.10000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000	4150.000000	13.000000	16.000000	5118.000000
25%	0.000000	101.000000	94.500000	166.55000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.600000	70.000000	4800.000000	19.000000	25.000000	7781.500000
50%	1.000000	122.000000	97.000000	173.20000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000	5200.000000	24.000000	30.000000	10595.000000
75%	2.000000	137.000000	102.400000	183.30000	66.900000	55.500000	2943.500000	143.000000	3.590000	3.410000	9.400000	116.000000	5500.000000	30.000000	34.000000	16500.000000
max	3.000000	256.000000	120.900000	208.10000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	288.000000	6600.000000	49.000000	54.000000	45400.000000

After this, I proceeded to perform Univariate Analysis.

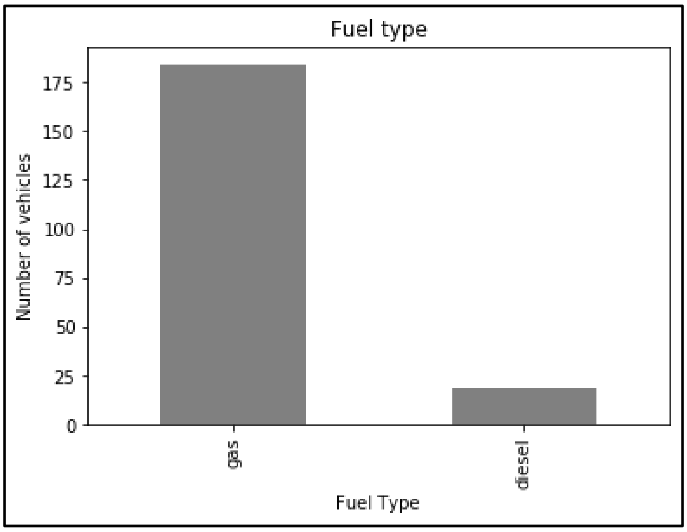
```
df['make'].value_counts().plot(kind='bar')
plt.title('Manufacturers')
plt.ylabel('Number of vehicles')
plt.xlabel('Manufacturer');
```



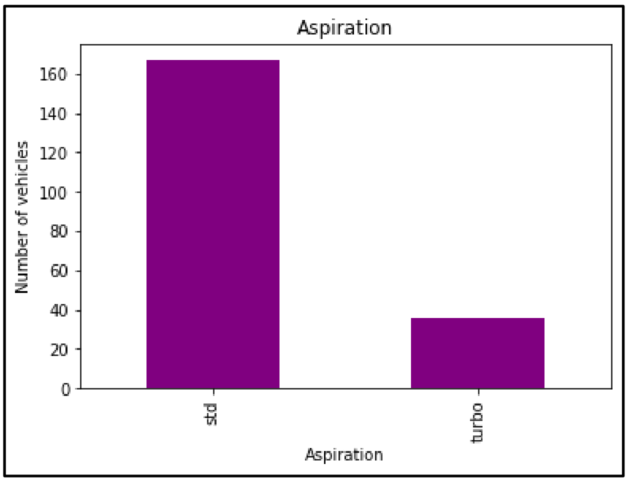
```
df['aspiration'].value_counts().plot.pie(figsize=(6,6),
autopct='%.2f')
plt.title('Aspiration')
plt.ylabel('Number of vehicles')
plt.xlabel('Aspiration');
```



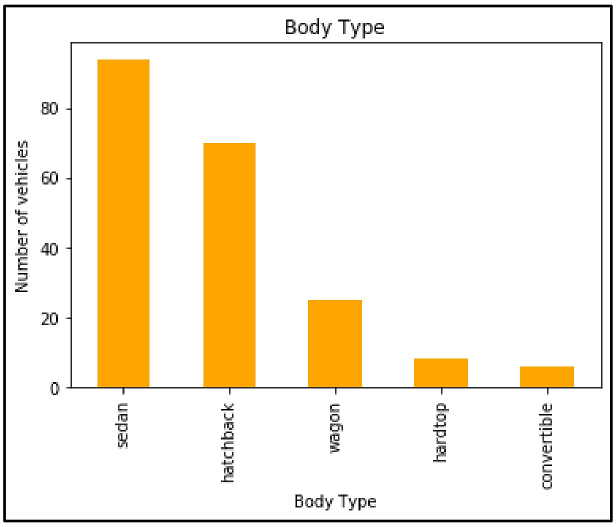
```
df['fuel-type'].value_counts().nlargest(10).plot(kind='bar',
color='grey')
plt.title('Fuel type')
plt.ylabel('Number of vehicles')
plt.xlabel('Fuel Type');
```



```
df['aspiration'].value_counts().nlargest(10).plot(kind='bar',
color='purple')
plt.title('Aspiration')
plt.ylabel('Number of vehicles')
plt.xlabel('Aspiration');
```



```
df['body-style'].value_counts().nlargest(10).plot(kind='bar',
color='orange')
plt.title('Body Type')
plt.ylabel('Number of vehicles')
plt.xlabel('Body Type');
```



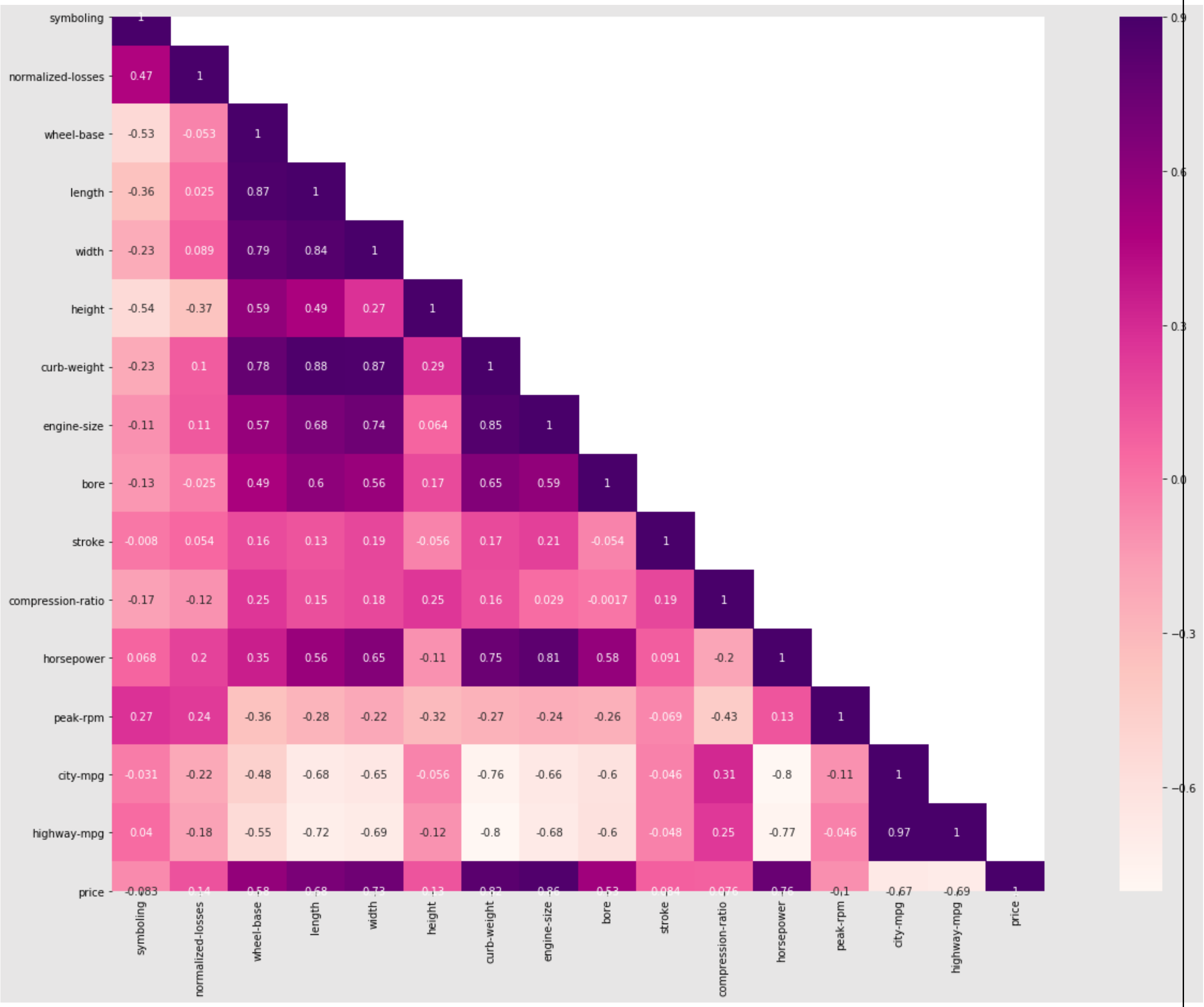
Above are a few graphs, along with their corresponding codes. The conclusions that can be drawn from the univariate analysis are:

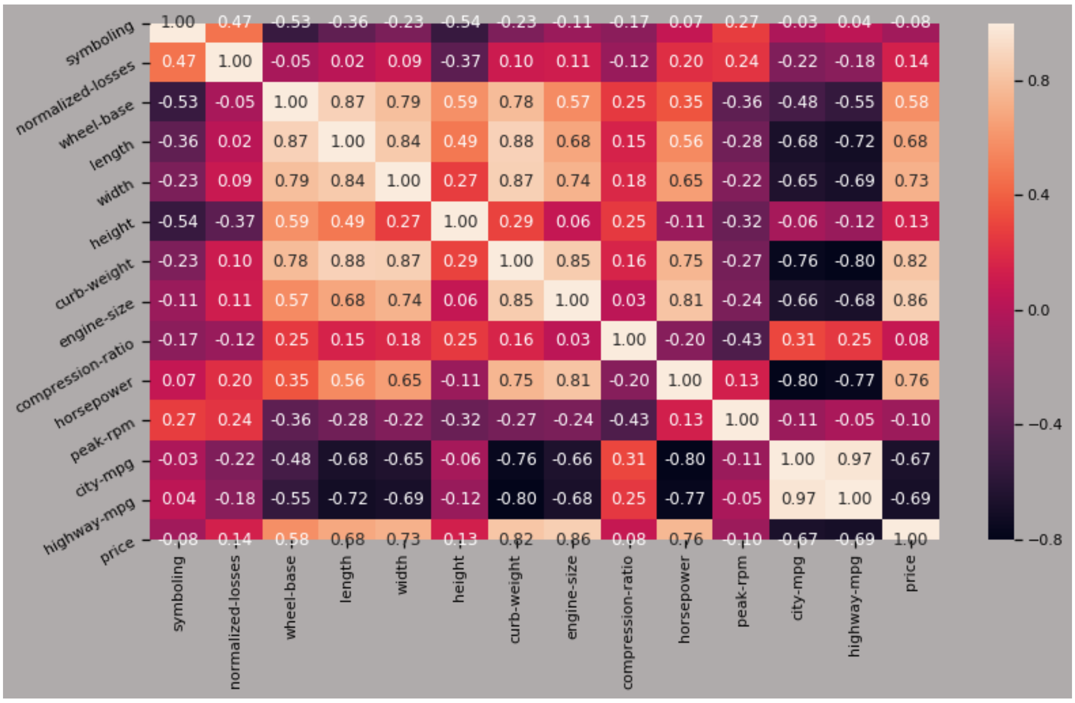
- Toyota is the leading manufacturer, followed by Nissan.
- Mercury and Renault manufactured a very small fraction of the vehicles.
- More than 90% of the vehicles are gas, while the most preferred system of aspiration is std.
- We also notice that Sedans are the most popular, while convertibles are the least.

After this, a correlational heat map was developed to perform correlation analysis.

```
corr = df.corr()
mask = np.array(corr)
mask[np.tril_indices_from(mask)] = False
fig,ax= plt.subplots()
fig.set_size_inches(25,15)
```

```
sn.heatmap(corr, mask=mask,vmax=.9, square=True,annot=True,
cmap="RdPu")
```





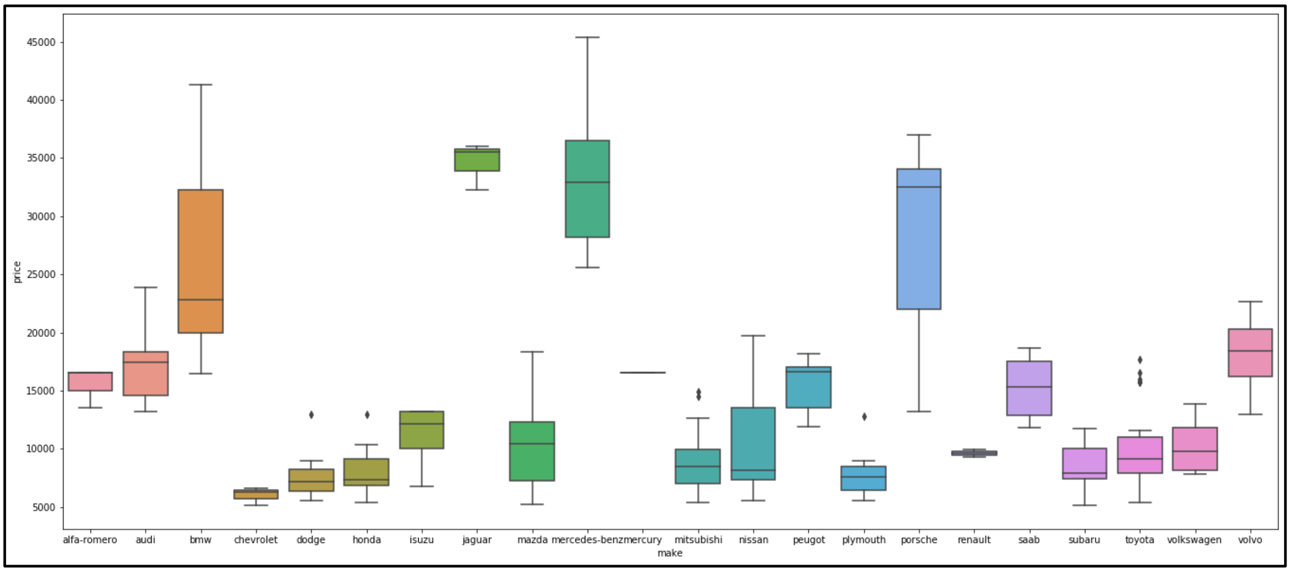
	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
symboling	1.00	0.47	-0.53	-0.36	-0.23	-0.54	-0.23	-0.11	-0.17	0.07	0.27	-0.03	0.04	-0.08
normalized-losses	0.47	1.00	-0.05	0.02	0.09	-0.37	0.10	0.11	-0.12	0.20	0.24	-0.22	-0.18	0.14
wheel-base	-0.53	-0.05	1.00	0.87	0.79	0.59	0.78	0.57	0.25	0.35	-0.36	-0.48	-0.55	0.58
length	-0.36	0.02	0.87	1.00	0.84	0.49	0.88	0.68	0.15	0.56	-0.28	-0.68	-0.72	0.68
width	-0.23	0.09	0.79	0.84	1.00	0.27	0.87	0.74	0.18	0.65	-0.22	-0.65	-0.69	0.73
height	-0.54	-0.37	0.59	0.49	0.27	1.00	0.29	0.06	0.25	-0.11	-0.32	-0.06	-0.12	0.13
curb-weight	-0.23	0.10	0.78	0.88	0.87	0.29	1.00	0.85	0.16	0.75	-0.27	-0.76	-0.80	0.82
engine-size	-0.11	0.11	0.57	0.68	0.74	0.06	0.85	1.00	0.03	0.81	-0.24	-0.66	-0.68	0.86
compression-ratio	-0.17	-0.12	0.25	0.15	0.18	0.25	0.16	0.03	1.00	-0.20	-0.43	0.31	0.25	0.08
horsepower	0.07	0.20	0.35	0.56	0.65	-0.11	0.75	0.81	-0.20	1.00	0.13	-0.80	-0.77	0.76
peak-rpm	0.27	0.24	-0.36	-0.28	-0.22	-0.32	-0.27	-0.24	-0.43	0.13	1.00	-0.11	-0.05	-0.10
city-mpg	-0.03	-0.22	-0.48	-0.68	-0.65	-0.06	-0.76	-0.66	0.31	-0.80	-0.11	1.00	0.97	-0.67
highway-mpg	0.04	-0.18	-0.55	-0.72	-0.69	-0.12	-0.80	-0.68	0.25	-0.77	-0.05	0.97	1.00	-0.69
price	-0.08	0.14	0.58	0.68	0.73	0.13	0.82	0.86	0.08	0.76	-0.10	-0.67	-0.69	1.00

The results of the correlational analysis are:

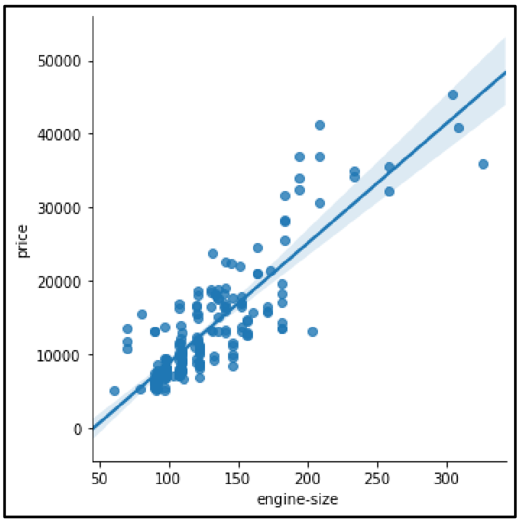
- We observe price is highly positively correlated with size and curb weight.
- Curb weight is also correlated with engine size, length, width, height and wheel base.
- Additionally, we observe sort of high negative correlation between the mileage offered and length, width, height, curb wight and engine size.
- Symbolling and normalized also display a slight correlation.

Next, bivariate analysis was performed.

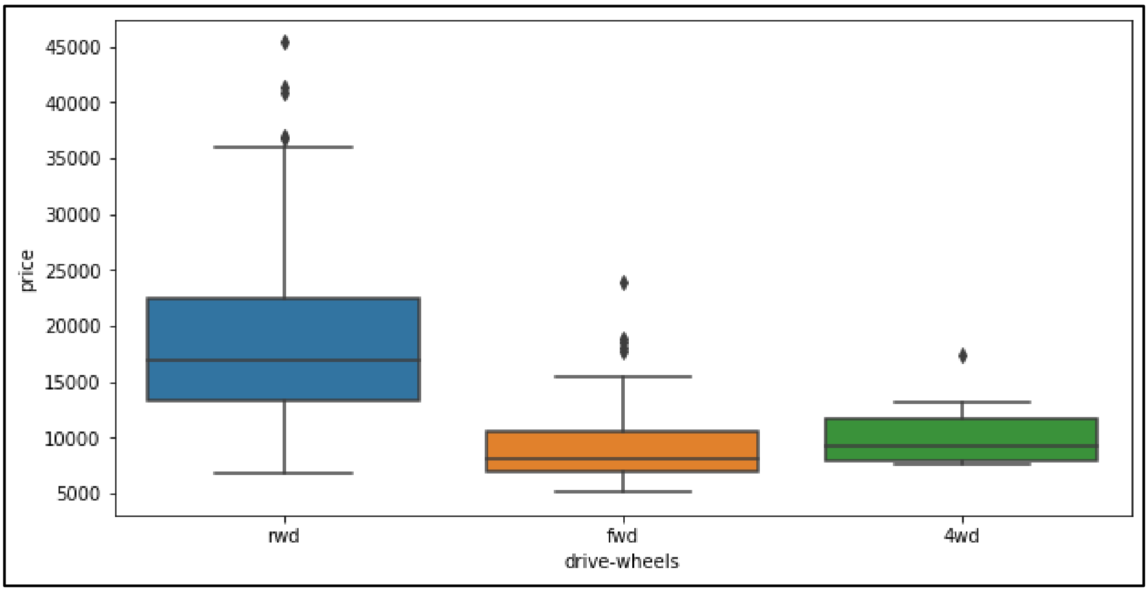
```
plt.rcParams['figure.figsize']=(23,10)
ax = sn.boxplot(x="make", y="price", data=df)
```



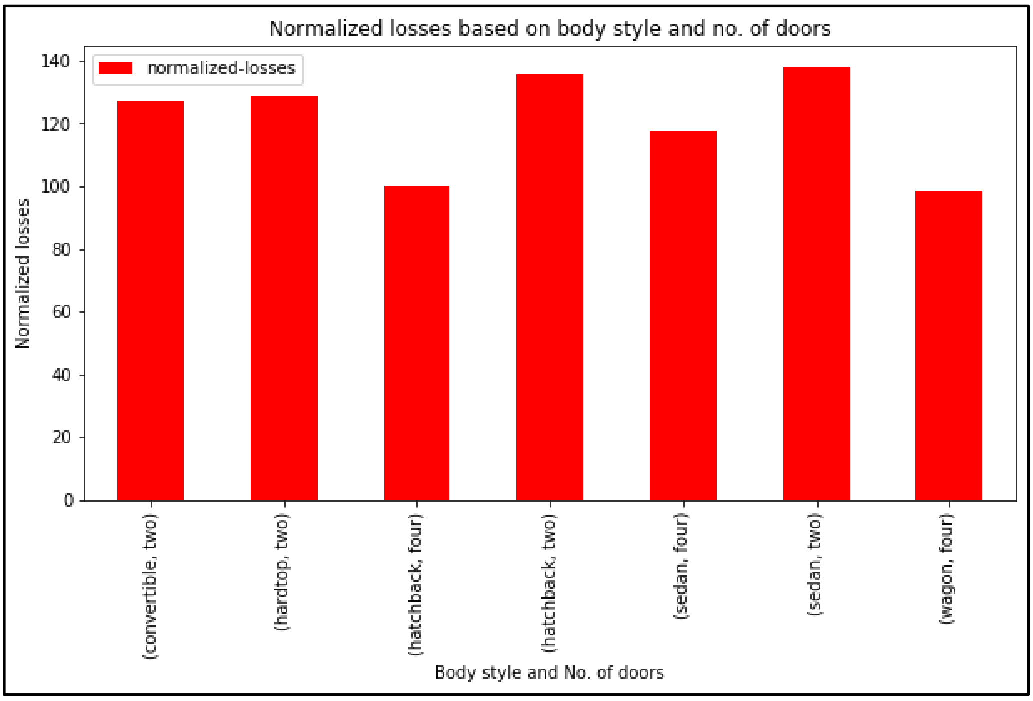
```
g = sn.lmplot('engine-size','price', df);
```



```
plt.rcParams['figure.figsize']=(10,5)
ax = sn.boxplot(x="drive-wheels", y="price", data=df)
```



```
pd.pivot_table(df,index=['body-style','num-of-doors'],
values='normalized-losses').plot(kind='bar',color='red')
plt.title("Normalized losses based on body style and no. of
doors")
plt.ylabel('Normalized losses')
plt.xlabel('Body style and No. of doors');
```



The results of the bivariate analysis are as follows:

- Mercedes makes the most expensive cars, while Chevrolet makes the least expensive.
- Many cars are in the price range 10000 and 20000.
- A positive linear trend was observed between price and engine size, in addition to price and horsepower.
- It was also observed that heavy cars usually gave less mileage, be it the city or highway.
- Rear wheel drive cars are more expensive, while front wheel drive cars are pretty cheap.
- Normalized losses differ with the body style and number of doors. It is maximum for a sedan with 2 doors. Additionally, it is also observed that the losses are higher for 2 door cars than 4 doors.

This ends the exploratory data analysis of the dataset.

Next, I tried my hand at model building. I omitted some features, and tried building a linear regression model, to predict price, using the features *symbolling*, *normalized-losses*, *wheel-base*, *length*, *width*, *height*, *curb-weight*, and a few more. The linear model thus generated had an R^2 score of 0.83, which is pretty good.

Future Scope of the Work: In addition to exploratory data analysis, we can use all the features provided, treat the remaining missing values, and use various machine learning and deep learning models to predict the price of a car given certain features such as *make, curb-weight, fuel-type, aspiration,* and many more.