

**TASK**

**Exploratory Data Analysis on the Automobile Data Set**

[](http://www.hyperiondev.com/portal/)

**Introduction**

I was presented with the document file automobile.docx containing data regarding vehicles.

After importing the necessary modules and libraries I printed out the below data frame.

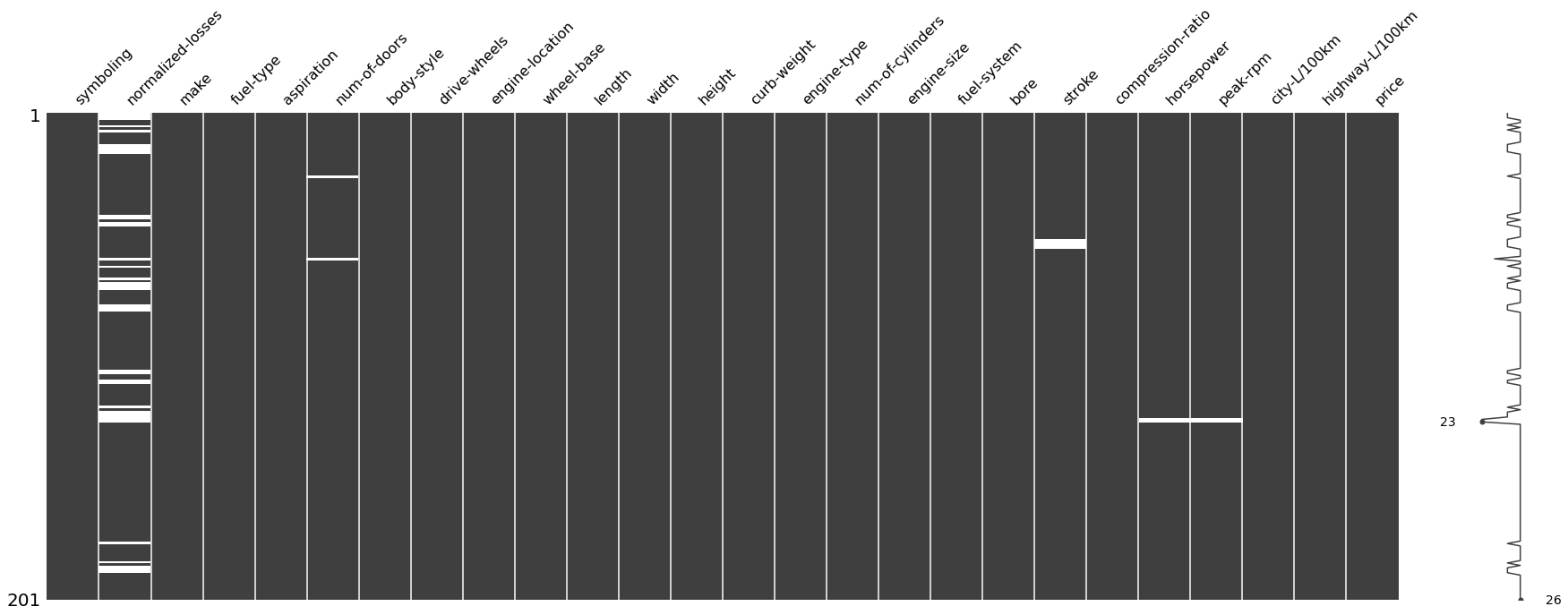
|  | **symboling** | **normalized-losses** | **make** | **fuel-type** | **aspiration** | **num-of-doors** | **body-style** | **drive-wheels** | **engine-location** | **wheel-base** | **...** | **engine-size** | **fuel-system** | **bore** | **stroke** | **compression-ratio** | **horsepower** | **peak-rpm** | **city-mpg** | **highway-**  **mpg** | **price** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 3 | ? | alfa-romero | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 | 9.0 | 111 | 5000 | 21 | 27 | 13495 |
| **1** | 3 | ? | alfa-romero | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 | 9.0 | 111 | 5000 | 21 | 27 | 16500 |
| **2** | 1 | ? | alfa-romero | gas | std | two | hatchback | rwd | front | 94.5 | ... | 152 | mpfi | 2.68 | 3.47 | 9.0 | 154 | 5000 | 19 | 26 | 16500 |
| **3** | 2 | 164 | audi | gas | std | four | sedan | fwd | front | 99.8 | ... | 109 | mpfi | 3.19 | 3.40 | 10.0 | 102 | 5500 | 24 | 30 | 13950 |
| **4** | 2 | 164 | audi | gas | std | four | sedan | 4wd | front | 99.4 | ... | 136 | mpfi | 3.19 | 3.40 | 8.0 | 115 | 5500 | 18 | 22 | 17450 |

5 rows × 26 columns

**DATA CLEANING**

As observed from the above table we have some values that are indicated as a question mark ‘?’.

I replaced all the ‘?’ with a NaN value by using .replace(). I also checked each of the columns to determine how many missing values each had by using df.isnull().sum(). Further, I pulled the percentage to evaluate the percentage of missing data, and obtained a 0.899% missing value. I concluded the data is sufficient to work with. I also pulled a matrix to visualize the missing data as shown below:



MISSING DATA

From my evaluation I determined that there were 41 data points missing in the normalized-losses column, 2 missing data points from the num-of-doors column, 4 from the bore column, 4 from the stroke column and 4 from the price column.

The normalized-losses can still be estimated from the rest of the data as it is a normalized amount given throughout the column.

Stroke and bore can also be estimated from the data frame.

The missing data points from the price column could not be deduced, as every automobile has a different price.

DATA STORIES AND VISUALIZATIONS

The first visualization I pulled shows the comparison of the price to the wheel-drive column. I used a box plot to show how the drive-wheel influences the price of the automobile. The main reason I chose a box plot was because it shows the median of the pulled data to obtain a better understanding of the data. I also used the .value\_counts() to see how many different wheel drives were included in the data. It returned the following:

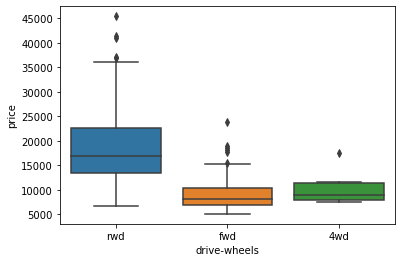
fwd 120

rwd 76

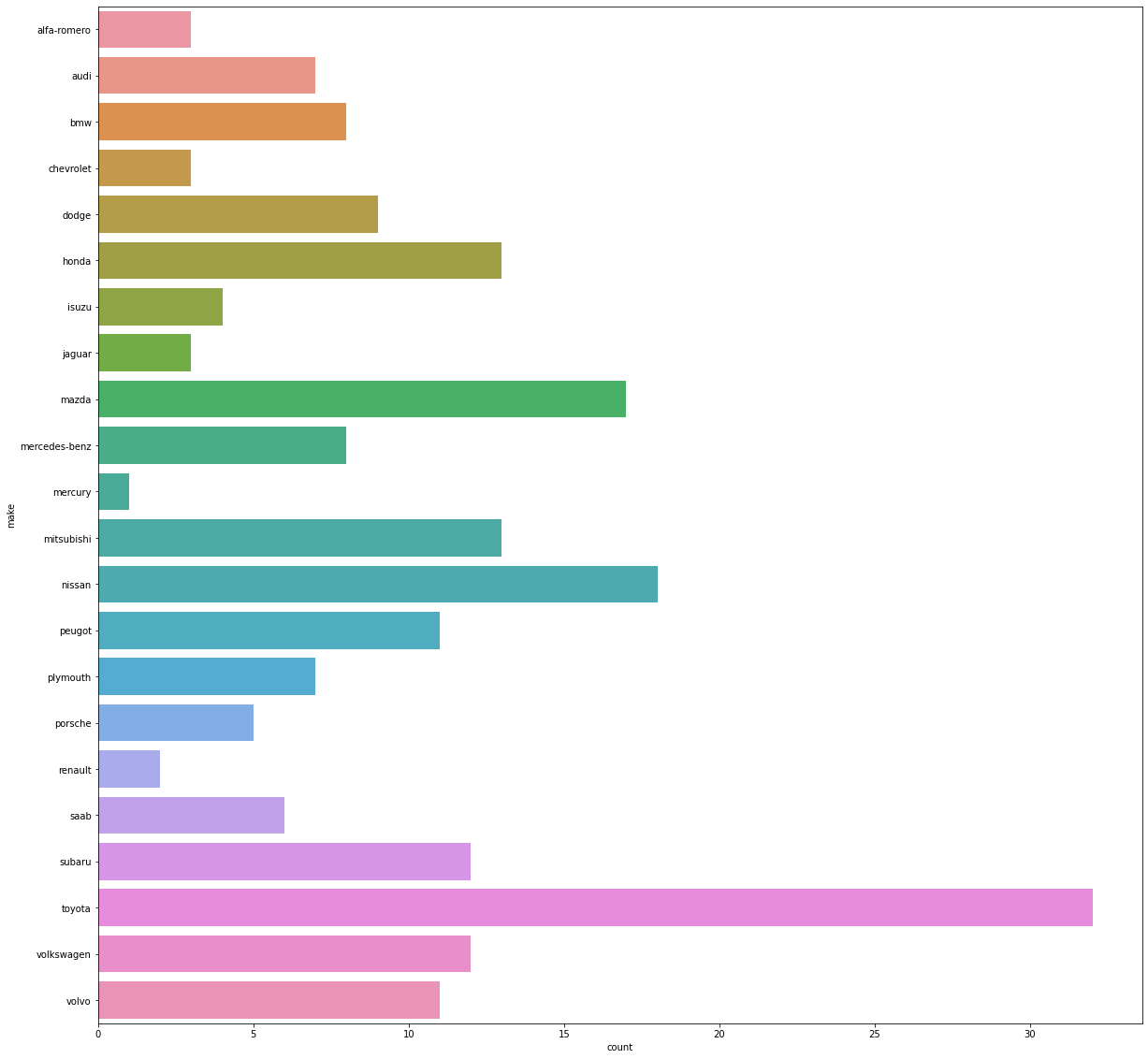
4wd 9

Name: drive-wheels, dtype: int64

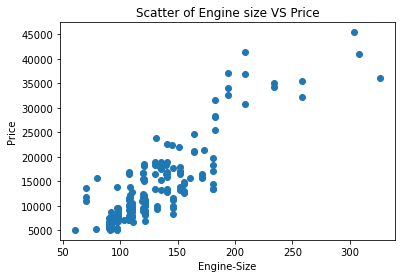
BOX PLOT

****

Further, I did a count plot to show the amount of automobiles per make and obtained the below result:



The next visualization I did was a scatter plot to see if the engine-size can predict the price of the vehicle. I obtained the following result:



From the scatter plot we observe that the price of the vehicle increases as the size of the engine increases.

As seen in the box plot, there were three different types of wheel-drives. Next I considered which drive-wheel is more expensive on average. I grouped the data by ‘drive-wheels, ‘body-style’ and ‘price’, and obtained the averages as seen below:

| **drive-wheels** | **body-style** | **price** |
| --- | --- | --- |
| **0** | 4wd | hatchback | 7603.000000 |
| **1** | 4wd | sedan | 12647.333333 |
| **2** | 4wd | wagon | 9095.750000 |
| **3** | fwd | convertible | 11595.000000 |
| **4** | fwd | hardtop | 8249.000000 |
| **5** | fwd | hatchback | 8396.387755 |
| **6** | fwd | sedan | 9811.800000 |
| **7** | fwd | wagon | 9997.333333 |
| **8** | rwd | convertible | 23949.600000 |
| **9** | rwd | hardtop | 24202.714286 |
| **10** | rwd | hatchback | 14337.777778 |
| **11** | rwd | sedan | 21711.833333 |
| **12** | rwd | wagon | 16994.222222 |

As observed, the data is not that easy to read. I used a pivot table to present the data in a more reader-friendly manner:

**price**

**body-style convertible hardtop hatchback sedan wagon**

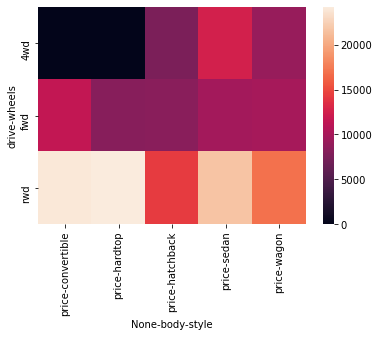
**drive-wheels**

**4wd 0.0 0.000000 7603.000000 12647.333333 9095.750000**

**fwd 11595.0 8249.000000 8396.387755 9811.800000 9997.333333**

**rwd 23949.6 24202.714286 14337.777778 21711.833333 16994.222222**

I also did a heat map, to present the data in another way:

****

**THIS REPORT WAS WRITTEN BY : Ryno Viljoen**

