Assessment 1: Data Cleaning and Summarising

Data Preparation

Step 1: Data retrieving and approach to task

Ensuring I import pandas as pd. I imported the csv using pd.read_csv() function. In this function I ensured that the data was separated by #, that there were no headers to import and to insead use the variable names I listed in the function.

My approach to data cleaning is investigate each variable and complete a repeatable process that checks for correct data type, typos, extra whitespaces, inconsistent casing, and sanity checks to check for impossible values, in order to clean and prepare the data for analysis.

Step 2: Check data types

To check data types I used the automobile.dtypes function.

Where I needed to convert the data type, for example in Symboling I used automobile['Symboling']=pd.Categorical(automobile.Symboling, ordered=True, categories=[-3,-2,-1,0,1,2,3]) to ensure this was an ordered (and categorical).

Or in the case where order isn't important but I still want to conver the data type into categorical I would use: automobile['Make']=pd.Categorical(automobile.Make, ordered=False)

Step 3: Typos

Using automobile['*variable*'].value_counts() I can identify the count of each value in the variable next. This allows me to pick up typos such as VOL00112OV to VOLVO, FOURR to FOUR, TURRRRBO to TURBO etc.

Step 4: Extra-white spaces

Using the value_counts() function to view the variable's values, I am able to determine if theire are white spaces. I used the str.strip() function on all variables regardless of if an apparent duplicate was shown as part of my repeatable and systemetic process. For example automobile['Make'] = automobile['Make'].str.strip() strips the whitespaces in the Make variable which was identified in VOLVO.

Step 5: Upper/lower case

To convert text data to all upper case I used the str.upper() function. I used this for all variables with text including Make For example automobile['Make'] = automobile['Make'].str.upper() converts all the Makes of cars into uppercase.

Step 6: Sanity Checks

Again using value_counts() function, I can view for impossible values. These are values that do not make sense, for example Symboling has a range of -3 to 3, therefore a value of 4 is impossible. I converted all instances of 4 into the closest possible value 3 using the replace() function. automobile['Symboling'].replace(4,3, inplace=True).

For Price, I found that some cars were priced at $0. To deal with this I first needed to import numpy as np. Then I injected Nan values where the Price was 0 using replace() funciton like so automobile['Price'].replace(0, np.NaN, inplace=True)

To get a more accurate mean by make of the car, I used the groupby() functino to fill nan values with the mean of that car make like so automobile['Price'] = automobile.groupby('Make')['Price'].apply(lambda x:x.fillna(round(x.mean())))

Step 7: Missing values

I filled the Nan values in the numeric variables of the data set using this code. I purposely chose the simplest method to fill the Nan value with the column-wise mean. The code used for this is: automobile=automobile.fillna(automobile.mean(axis=0))

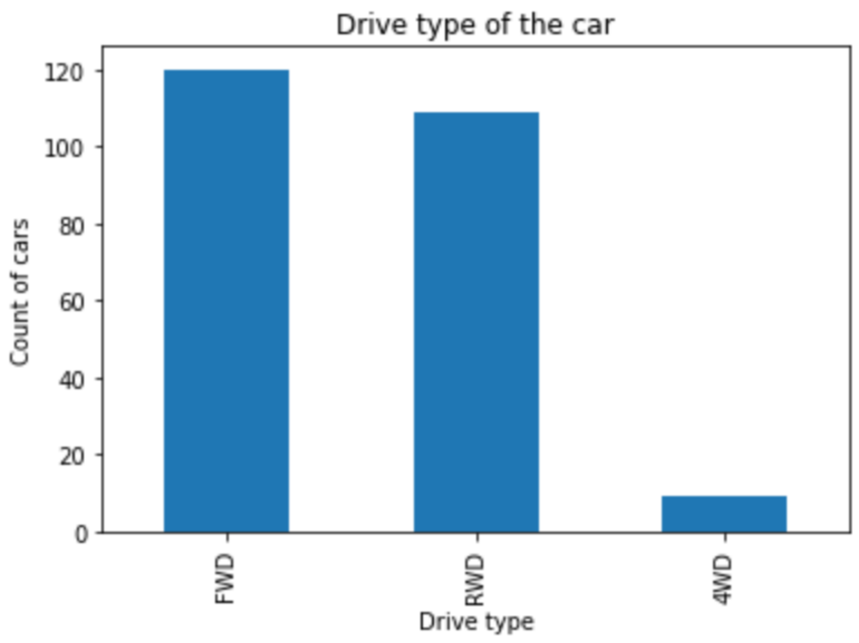
To ensure that my Nan values were filled I still used the isnull() function at the beginning of my investigation on each variable to double check for Nan values.

To treat missing values in categorical data, it doesn't make sense to use an average. Instead we used the mode to assign the most common value in the variable. This was required in Num of doors, where I used the mode to replace the missing values with FOUR with automobile['Num of doors'] = automobile['Num of doors'].fillna(automobile['Num of doors'].mode()[0])

Data Exploration

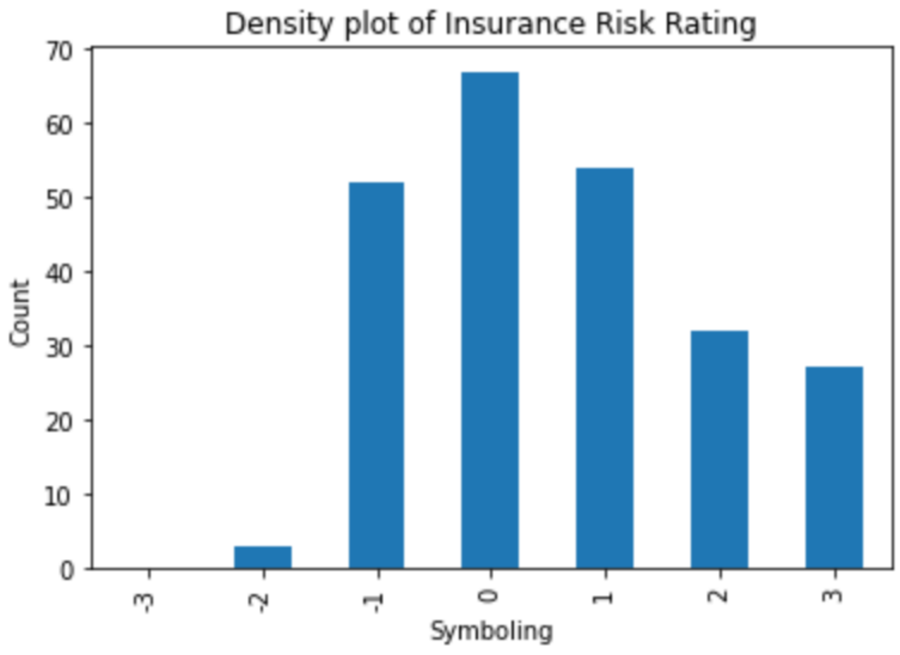
Subsection 1:

Graph of a Nominal variable (Drive wheels or Drive type of the car)



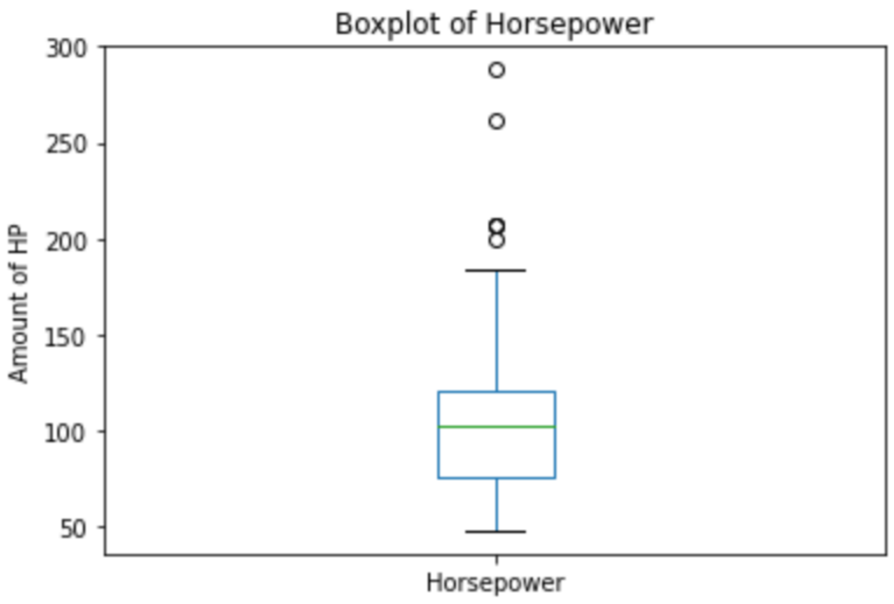
Since there are 3 categories (not ordered) the simplest visual representation of the variable "Drive wheels" (drive type of the car) is in a bar chart. This allows us to show side by side comparison of the number of cars in each drive type. The bar chart defaults to the highest count on the left to lowest count on the right of the x axis.

Graph of ordinal data (Symboling)



Similar to the graphing of nominal data, a bar graph is the best visual representation for ordinal data. In this example however, we need to ensure that the insurance risk rating is ordered from left to right on the x axis correctly.

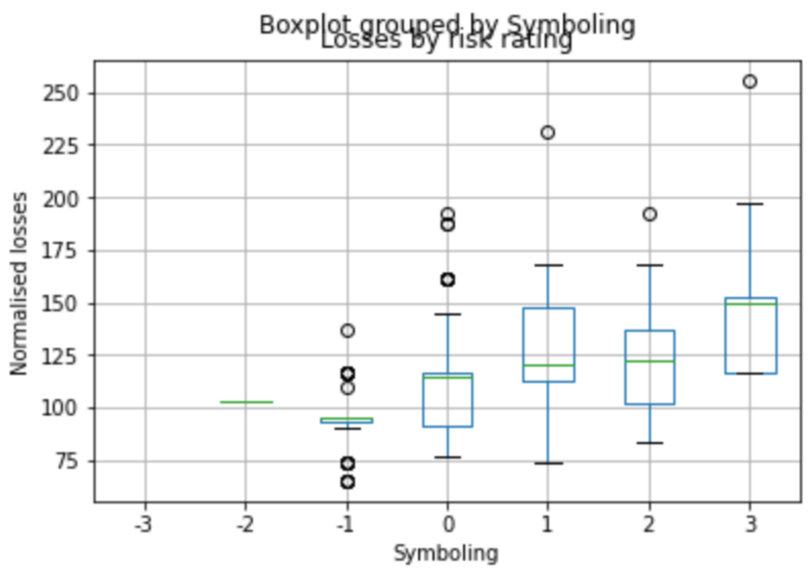
Graph of numerical data (Horsepower)



A box plot is ideal for displaying numerical data as it gives us some simple summary statistics including the median, the interquartile range, the minimum and maximum and clearly shows the outliers.

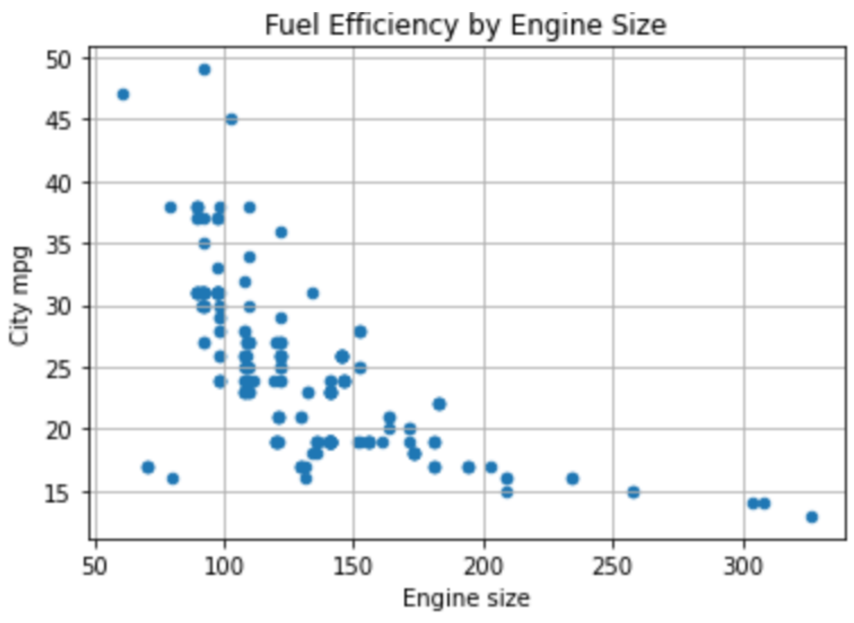
Subsection 2:

Hypothesis 1
Is there are relationship between insurance risk rating and normalised losses? I hypothesis that vehicles which carry a higher risk rating have higher normalised losses.



We can see from the graph above, that the box plots show an apparent postive relationship between normalised losses and symboling as we can see the box slightly shift upwards as symboling is increases (3 indicates higher risk).

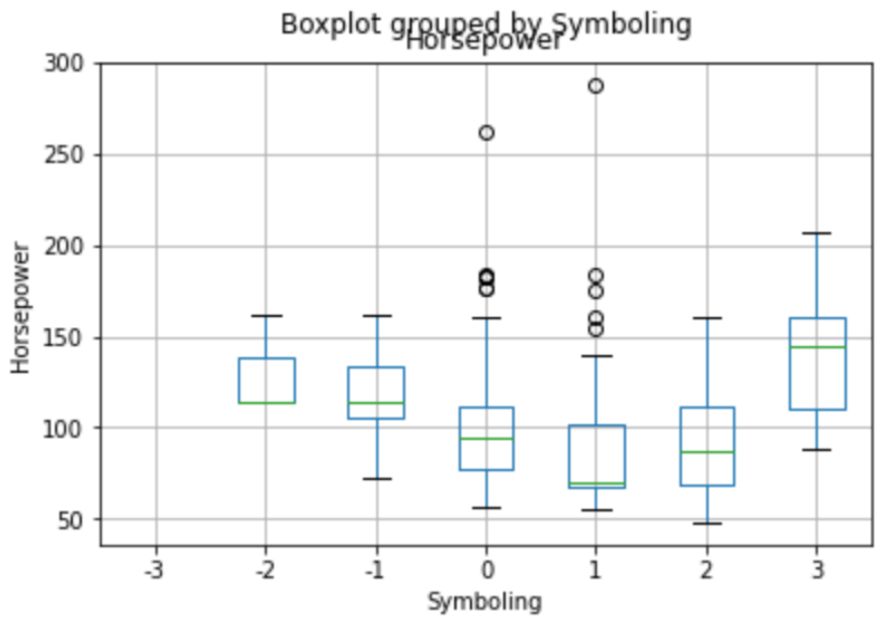
Hypothesis 2
Is there a relationship between engine size and fuel efficiency (city mpg)? I hypothesise that larger engines are less fuel efficient particularly with city driving where there are lots of stops and starts.



We can see from the graph above that engine size has an inverse relationship with city mpg, suggesting that the hypothesis might be accurate.

Hypothesis 3

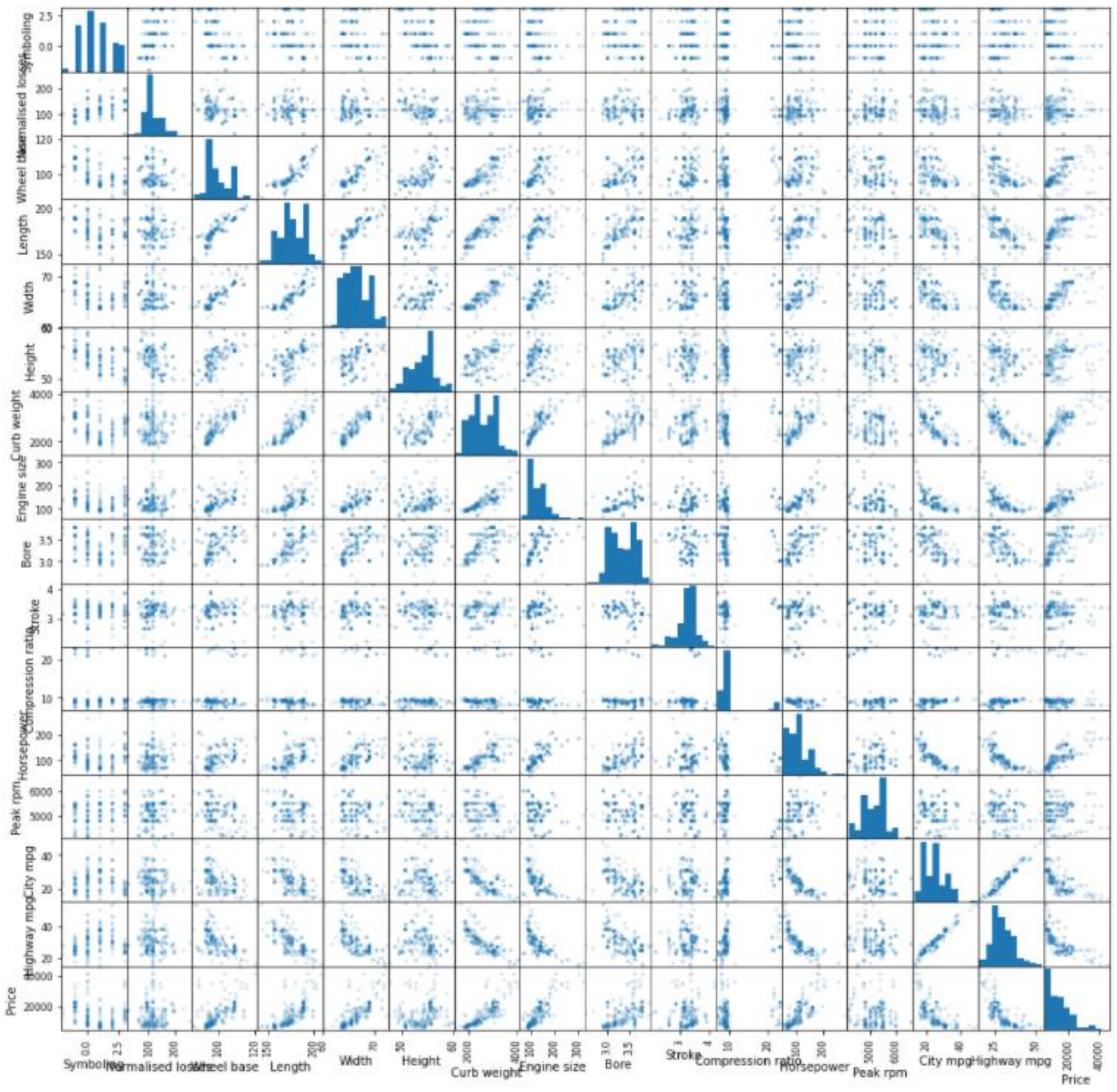
Is there a relationship between horsepower and the insurance risk rating? I hypothesis that Cars with more horsepower have a higher risk rating (symboling).



I hypothesised that the more horsepower that a car has the higher the risk rating. We see the median horsepower actually decrease as symboling increases from -2 to 1 before increasing again from 1 to 3. Also of note is seeing the max range generally shift upwards as risk gets higher. So perhaps the effect is seen on the higher end of risk rating scale but overall the relationship appears inconclusive.

Subsection 3:

```
from pandas.plotting import scatter_matrix
auto_num = automobile.select_dtypes(include=np.number)
scatter_matrix(auto_num, alpha=0.2,figsize=(16,16),diagonal='hist')
plt.show()
```



What can I observe from this graph?

The scatter matrix above shows the bivariate relationship between each combination of numeric variables in the data set, this allows many relationships to be explored in one visualisation. Since we didn't find a strong correlation between Horsepower and Symboling we'll explore further where horsepower appears to have a relationship with other variables:

- There appears to be a positive relationship between horsepower and price.
- There appears to be a positive relationship between horsepower and curb weight.
- There appears to be a positive relationship between horsepower and engine size.
- There appears to be an inverse relationship between horsepower and city mpg / highway mpg (fuel efficiency).