RMIT University Vietnam - SGS Campus

COSC 2789 - Practical Data Science

Assignment I Report

by

Anh Nguyen - s3616128

August 29th, 2019

TABLE OF CONTENTS

I. Data Preparation

1. Data Types

To view data types of a dataframe, simply use:

```
data.dtypes
```

In this case, the dataframe's data types are correct. No additional work needed.

2. Typos

To detect typos in the dataframe, I created a function to list all unique values within a column.

```
def list_unique_values_in_a_column(df, column):
  return df[column].unique().tolist()
```

Once a typo is found, a correct word will be passed through a function to replace the typo.

```
def fix_typo(df, column, typo, correct_word):
  mask = df[column] == typo
  df.loc[mask, column] = correct_word
```

3. Redundant Whitespaces and Capitalized Characters

In case of redundant whitespaces, I applied a function to strip all excessive whitespaces from columns that contain string data type.

```
data = data.apply(lambda x: x.str.strip() if(x.dtype == 'object') else x)
```

Similarly, another function was created to lower all characters in the dataset to ensure maximum compatibility.

```
data = data.apply(lambda x: x.str.lower() if(x.dtype == 'object') else x)
```

4. Missing Values

In order to find missing values, I created two functions:

- *List_unique_values_in_a_column* : this function will list all the unique values contained in a column. Results have to inspect manually. This function is primarily used to find typos in a given dataframe. However, it can also tell if a column contains NaN values.
- *Check_if_column_has_nan_value* : the primary function used for detecting NaN value in a column. Primarily used on numerical columns.

Once the missing values are identified, replacement values are determined on a case-by-case basis.

For example, the *peak-rpm* column has several missing values. Peak rpm of an engine is determined by the fuel system. Thus, I replaced the missing values with the mean peak-rpm of a corresponding fuel system in order to maintain data integrity.

For more details, please refer to the Jupyter notebook file.

5. Impossible Values

The *check_boundaries* function is created to detect any anomaly in numerical columns. It will compare the minimum and maximum values of a column with the lower bound and upper bound values respectively. Additionally, it can also detect 0 values in a column.

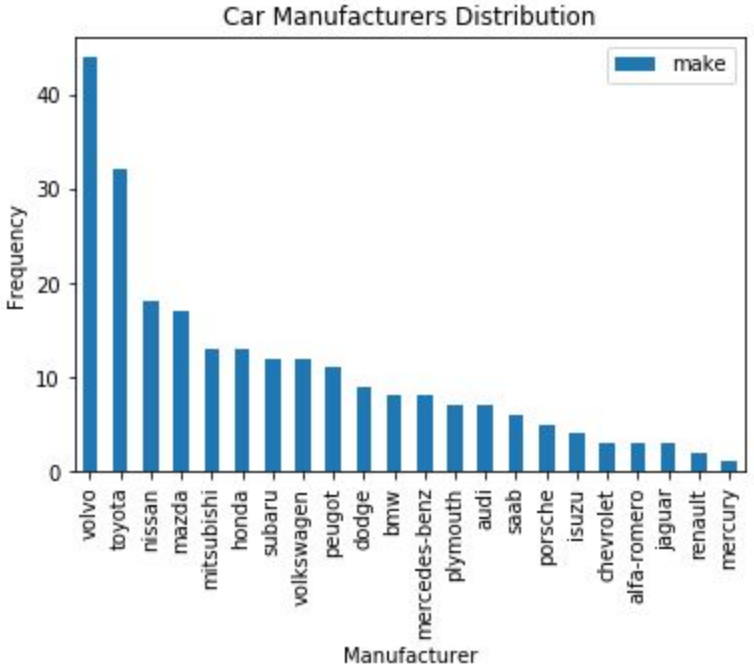
To replace impossible values, the replacement values are also determined on a case-by-case basis. For example, in the *symboling* values, the lower and upper boundaries are -3 and 3 respectively. Any value that is smaller than the lower bound will be replaced will the lower bound value. It works the same way for values that exceed the upper bound.

There are cases where it is hard to determine an upper bound value. I recommend using *None* in the upper bound parameter of the boundary check function. It means that there is no upper bound.

II. Data Exploration

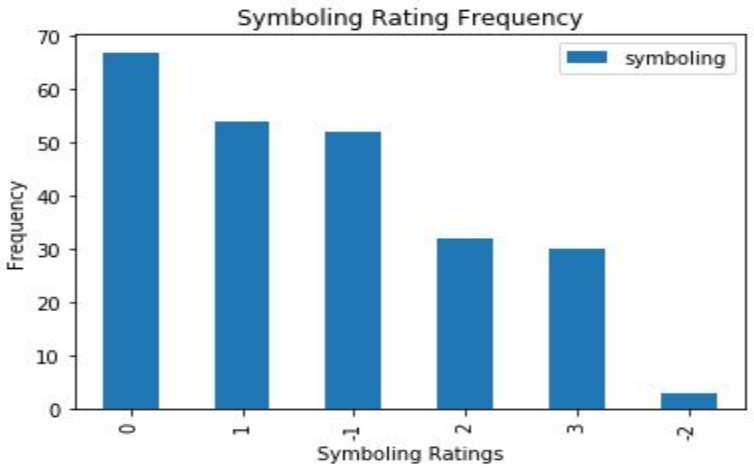
1. Single Value Graphs

- Nominal Value: Make



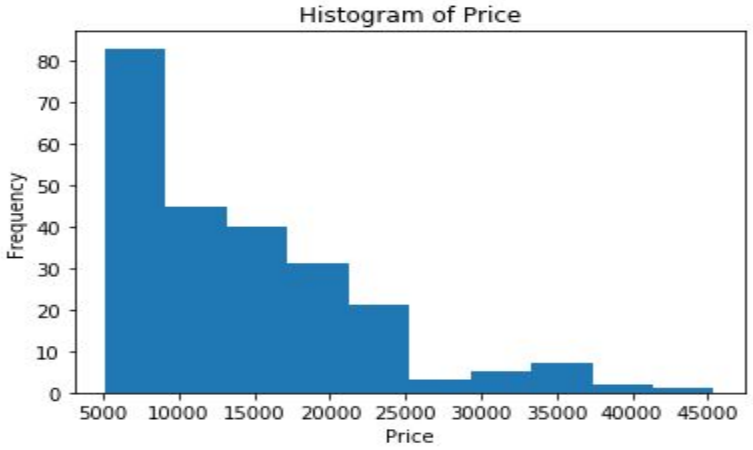
Since *make* is a nominal value, a column or bar chart is most appropriate to represent the distribution of values within the *make* column in the dataframe.

- Ordinal Value: Symboling



Similar to the above, a column or a bar chart is suitable to summarize the frequencies of ordinal values in the *symboling* column.

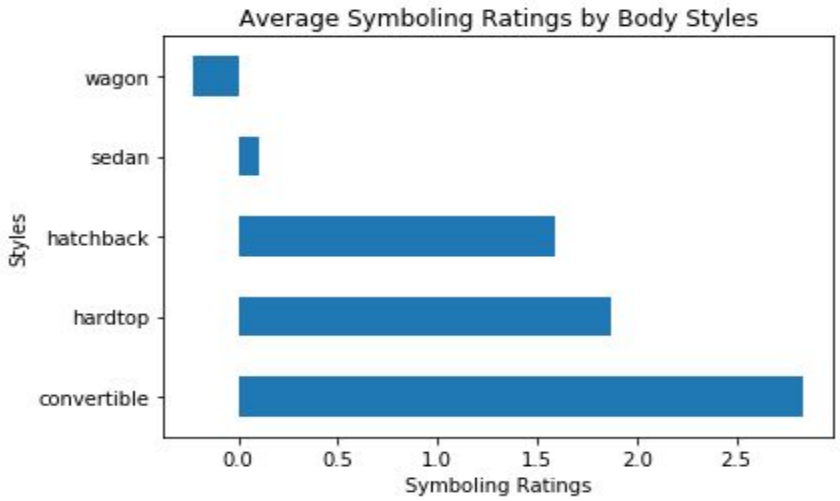
- Numerical Value: Price



Since the *price* column has many distinct values, it is appropriate to use a histogram because the data is grouped (such as 5000-10000, 10000 - 15000 etc.).

2. Pair Value Graphs
   - Body Styles and Symboling

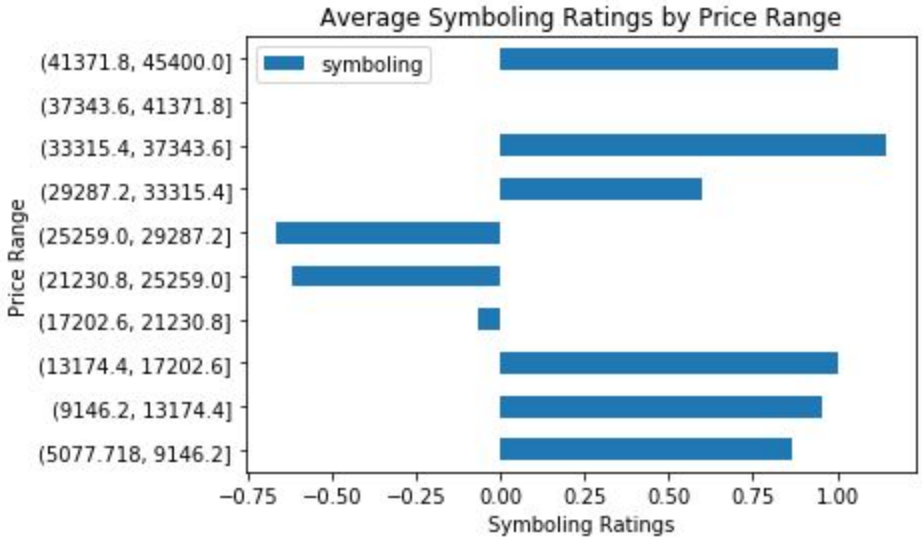
     Hypothesis: Family-oriented vehicles (sedans and wagons) tend to be less risky to insure compared to individual-oriented vehicles (hatchback, hardtop, convertible).



This graph confirms the above hypothesis. Since family-oriented vehicle owners tend to drive with their family members inside which make them drive more carefully. Whereas individuals who own sports cars tend to drive more recklessly which makes them riskier to insure.

- Price and Symboling

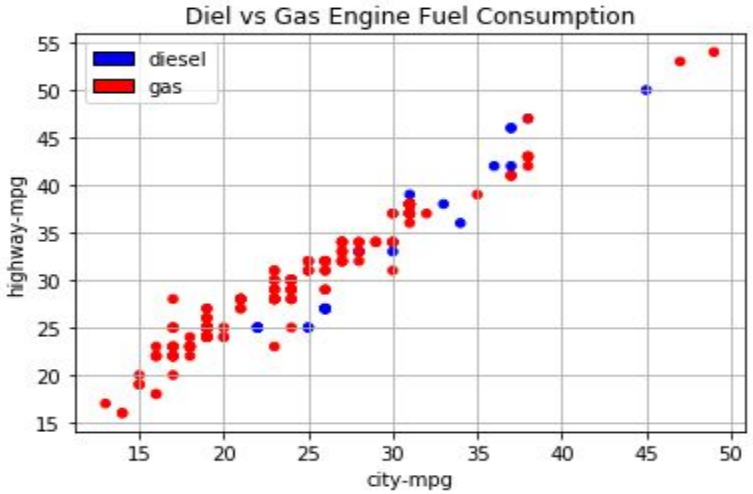
  Hypothesis: How does the price of a car affect its symboling rating?



  There seems to be a sweet spot from 17,000 to 30,000 USD where cars are considered least risky to insure. More expensive cars are less popular and their parts are expensive to obtain which increases the risk. Interestingly, less expensive cars have comparable riskiness to the more expensive ones. Perhaps, they are made of lower quality materials and prone to break down?

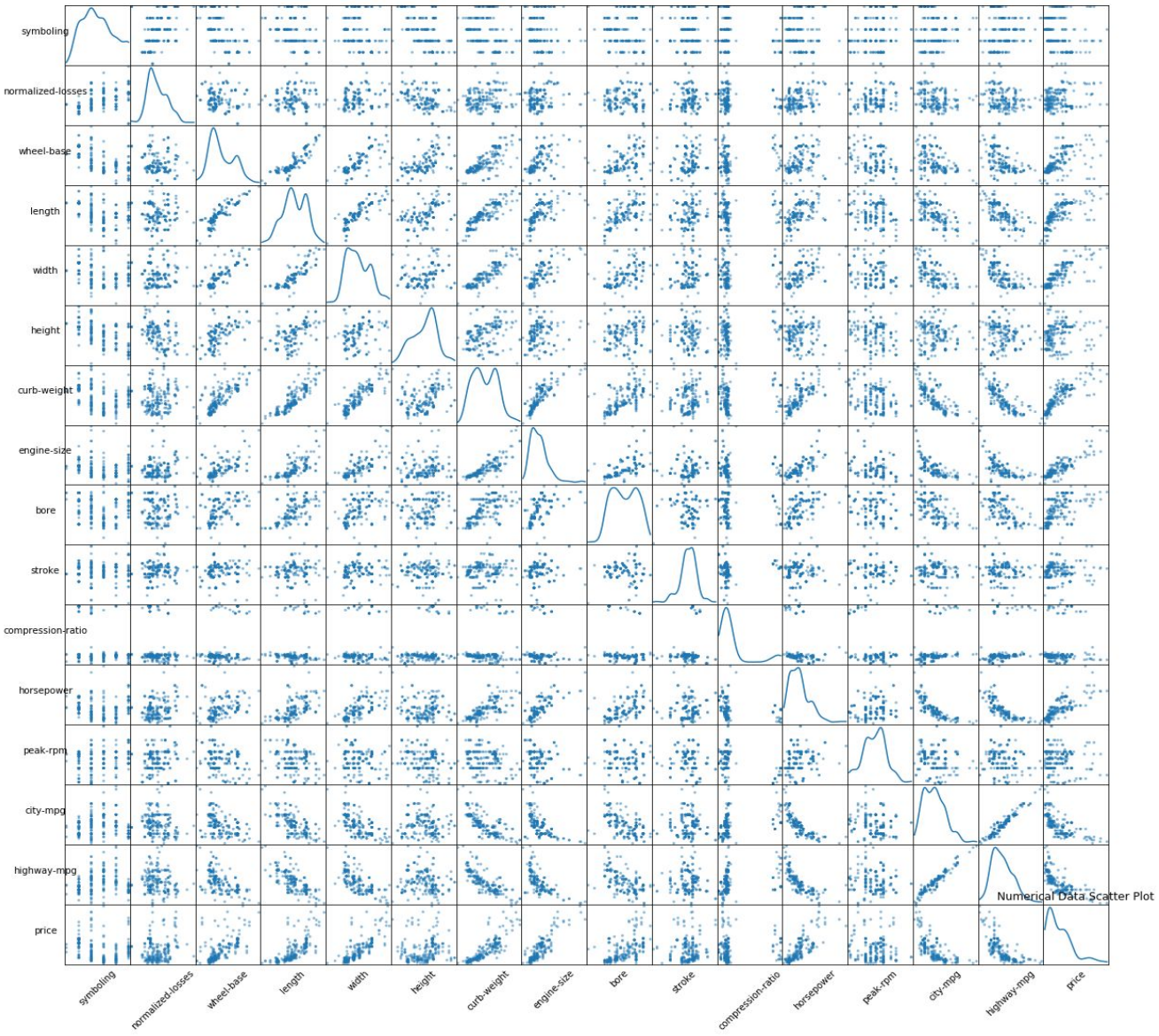
- Fuel Types and City MPG

  Hypothesis: Diesel-powered cars are more fuel-efficient than gas-powered cars.



From the above plot, gasoline-based cars tend to have lower city-mpg than diesel cars.

- Scatter Matrix



Viewing plots in the bottom row (*price* row specifically) shows a clear relationship between price and other numerical features. Specifically, there is a positive correlation between price and length, width, curb-weight, engine-size, and horsepower - as one increases, and so does the others. On the other hand, there is a negative (inverse) relationship between price and city-mpg which is reasonable since more expensive cars have bigger engines which, in turn, consume more fuel. Additionally, these relationships are not quite linear, they seem logarithmic.