Hyperiondev

TASK

Exploratory Data Analysis on the Automobile Data Set

Visit our website

Introduction

Exploratory Data Analysis (EDA) is performed to extract meaningful insights from a data set. I have used the Python programming language and no. python libraries pandas, NumPy to handle arrays and matrices and Matplotlib and Seaborn to create plots. The code file is created in Jupyter notebook. It is important to understand the variables within a data set in order to extract meaning from them and generate predictive models.

The automobile data set contains 205 rows and 26 columns.

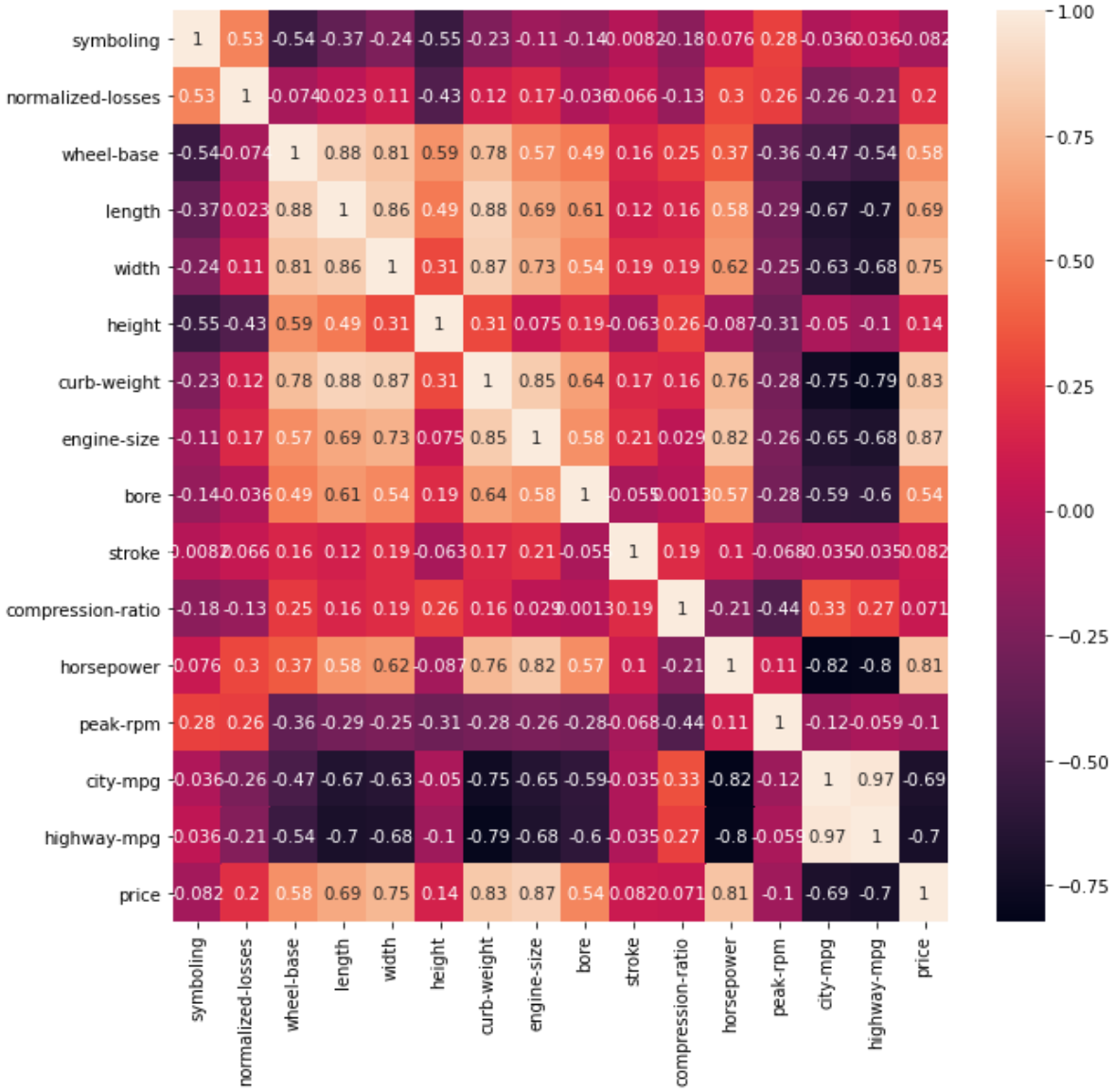
This data set consists of the following:

a) the characteristics of each vehicle, that is the height, highway-mpg, curb-weight, body-style, make of each vehicle, etc.

b) symbolling of each vehicle. Each vehicle is assigned a value between -3 and +3. This value indicates the risk factor of the vehicle, after the initially assigned risk factor associated with its price, whereby -3 is safe and + 3 is risky.

c) the normalized losses represent the average loss per car per year.

The heatmap below illustrates the correlation between all the features within the dataset.



The dataset shows that the features with a positive correlation to price are the vehicles' horsepower, engine-size, curb-weight, width and length. A negative correlation exists between price and peak-rpm and both highway- and city-mpg.

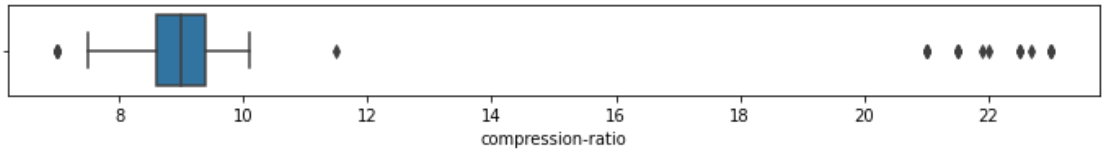
Using the skew() method to calculate the skew for each feature. The results of which are shown below:

```
symboling           0.197370
normalized-losses   0.765976
wheel-base          1.031261
length              0.154446
width               0.875029
height              0.029173
curb-weight         0.705804
engine-size         1.979144
bore               -0.032622
stroke             -0.693778
compression-ratio   2.584462
horsepower          1.141584
peak-rpm            0.107729
city-mpg            0.680433
highway-mpg         0.549507
price               1.809675
```

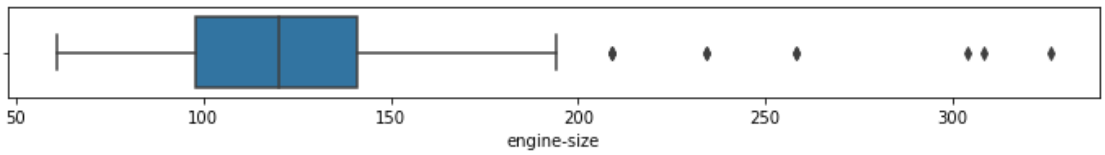
The target feature, price, is positively skewed. Other features that are positively skewed include wheel-base, engine-size and the compression-ratio. Majority of the features in the automobile dataset indicate not skewness in the distribution. Bore and stroke are negatively skewed.

The distribution of the features and the presence of outliers within the automobile dataset can be visualised using the boxplot method in seaborn. I have excluded the object variables. The results of which is shown below:

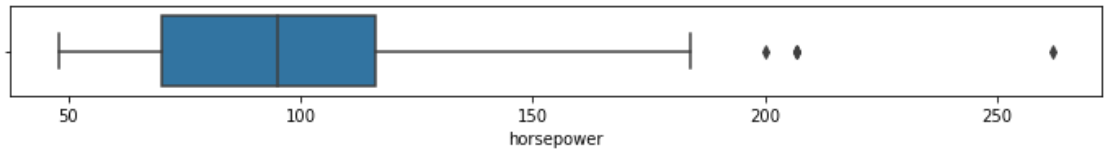
The target feature has multiple high extreme values, where one vehicle costs more than 45,000. 75% of the vehicles a priced below 16,500. There are no vehicles with extremely low prices.



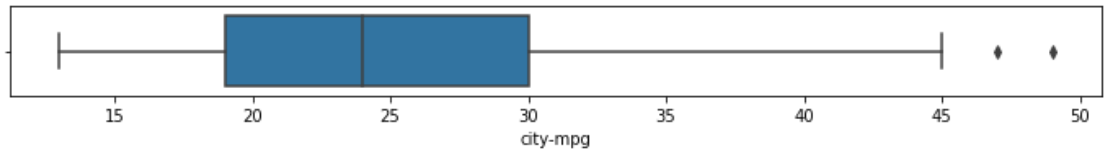
Compression-ratio consists of global outliers, where the data values are far from other values. 75% of the vehicles have a compression ratio lower than 9.4.



There are vehicles with extremely high engine-sizes but 75% of vehicles have an engine-size of 141.



The average horsepower within the dataset is 103 and the horsepower of 75% of the vehicles is below 116.



The minimum city-mpg is 13 and the maximum is 49. 75% of the vehicles have a city-mpg of less than 30.

DATA CLEANING

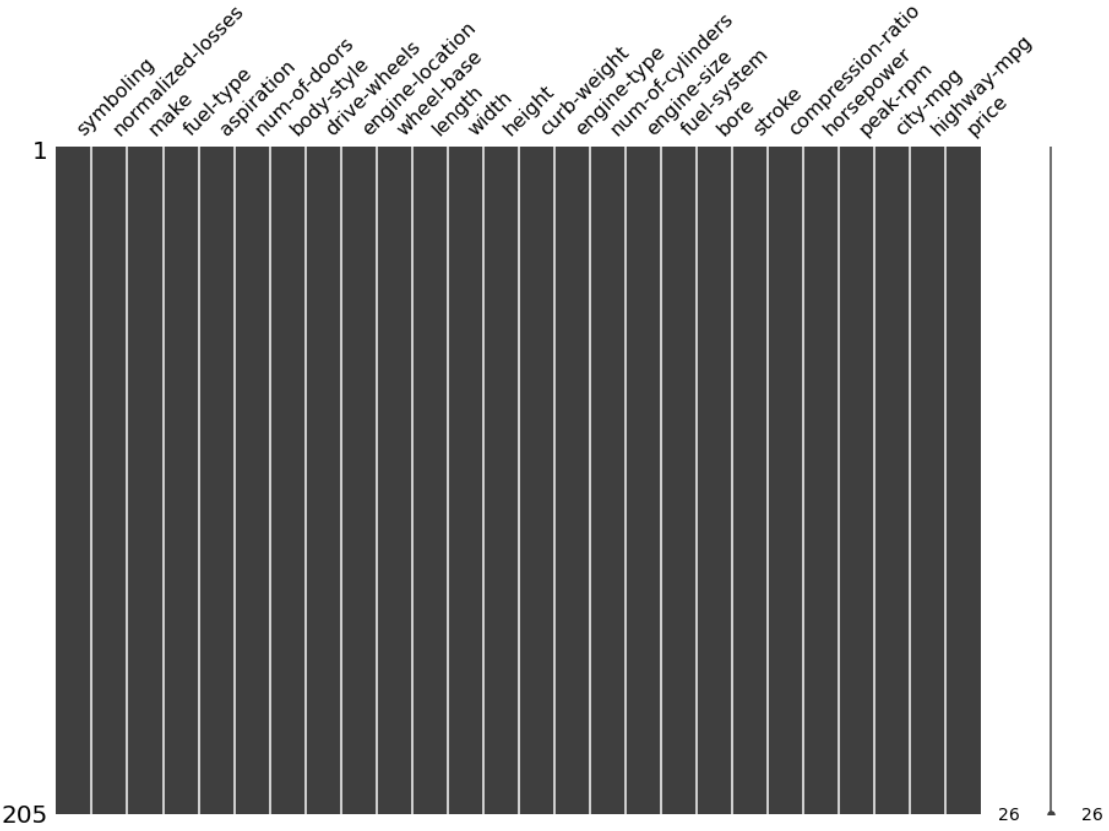
To determine the data types contained in each column of the DataFrame I use the info() method, and obtain the following:

```
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  205 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   aspiration         205 non-null    object
 5   num-of-doors       205 non-null    object
 6   body-style         205 non-null    object
 7   drive-wheels       205 non-null    object
 8   engine-location    205 non-null    object
 9   wheel-base         205 non-null    float64
 10  length             205 non-null    float64
 11  width              205 non-null    float64
 12  height             205 non-null    float64
 13  curb-weight        205 non-null    int64
 14  engine-type        205 non-null    object
 15  num-of-cylinders   205 non-null    object
 16  engine-size        205 non-null    int64
 17  fuel-system        205 non-null    object
 18  bore               205 non-null    object
 19  stroke             205 non-null    object
 20  compression-ratio  205 non-null    float64
 21  horsepower         205 non-null    object
 22  peak-rpm           205 non-null    object
 23  city-mpg           205 non-null    int64
 24  highway-mpg        205 non-null    int64
 25  price              205 non-null    object
dtypes: float64(5), int64(5), object(16)
```

Some variables have incorrect data types, for example they are stored as objects as opposed to numeric data types. Take note that some features contain '?' data values, these '?' values need to be converted to 'np-nan' so that Python can recognise them as missing values and I can convert the features to floats.

MISSING DATA

Missingno is a python library that offers various visualisations to show the distribution of missing values within a DataFrame. The matrix plot below indicates that each column of the automobile DataFrame contains a value and that no cell is empty.



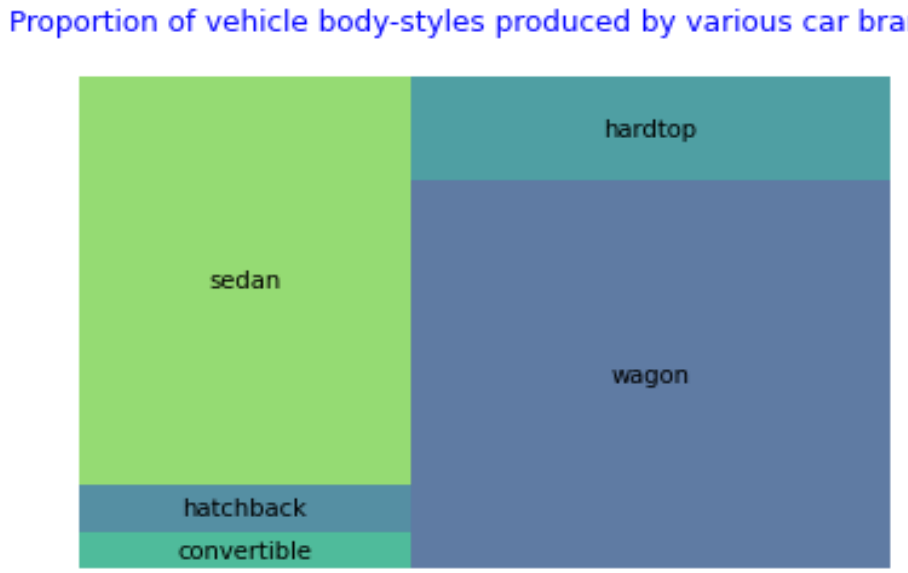
However, on close inspection of the first 5 rows of vehicles_df one can see that some columns contain a special character, specifically '?'. Pandas cannot accurately detect the '?' symbol as a missing value. Therefore, the columns that contain the special characters need to be dealt with separately.

There are 4 out of 205 records that have the '?' data value in the 'price' column.

The normalized-losses and horsepower columns also contain '?' data value. In order to replace the '?' with the mean of each column, '?' needs to be converted into np.nan in order for pandas to recognise it as a missing value and not a number.

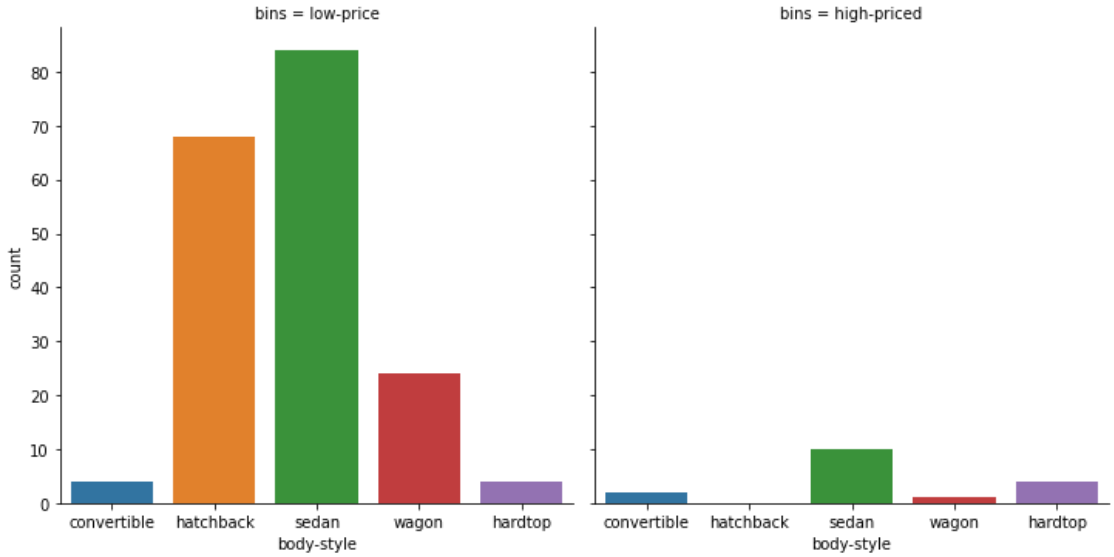
DATA STORIES AND VISUALIZATIONS

The Treemap below indicates that the most vehicles are wagons followed by sedans. The dataset contains fewer convertibles and hatchbacks. Therefore, we have more data for sedans and wagons and can conduct a thorough analysis than for convertibles and hatchback.



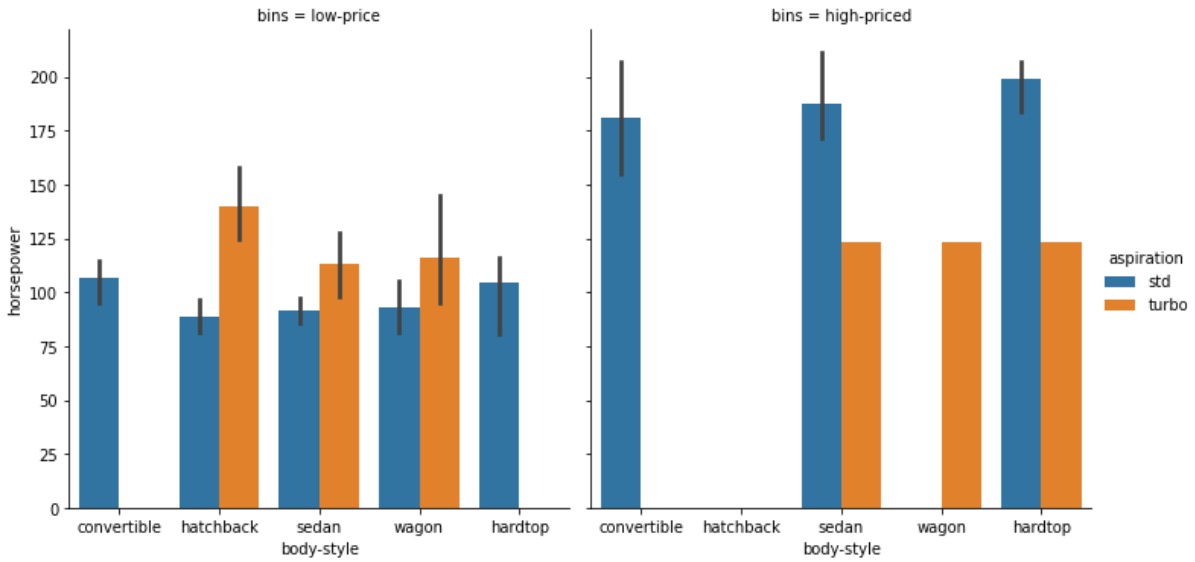
Define a sequence of prices that is evenly spaced, use the cut() function to segment and sort the price values into bins. Vehicles within the price range, [5118, 25259] are labelled low-priced and vehicles within the price range, [25259, 45400] are labelled high-priced.

The number of vehicles per body-style within each price range is illustrated below.

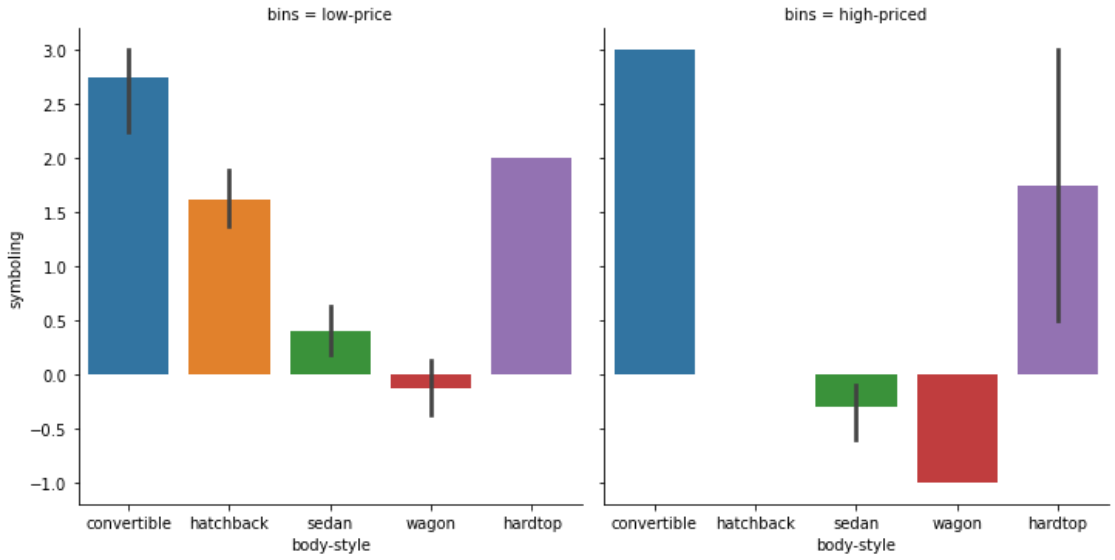


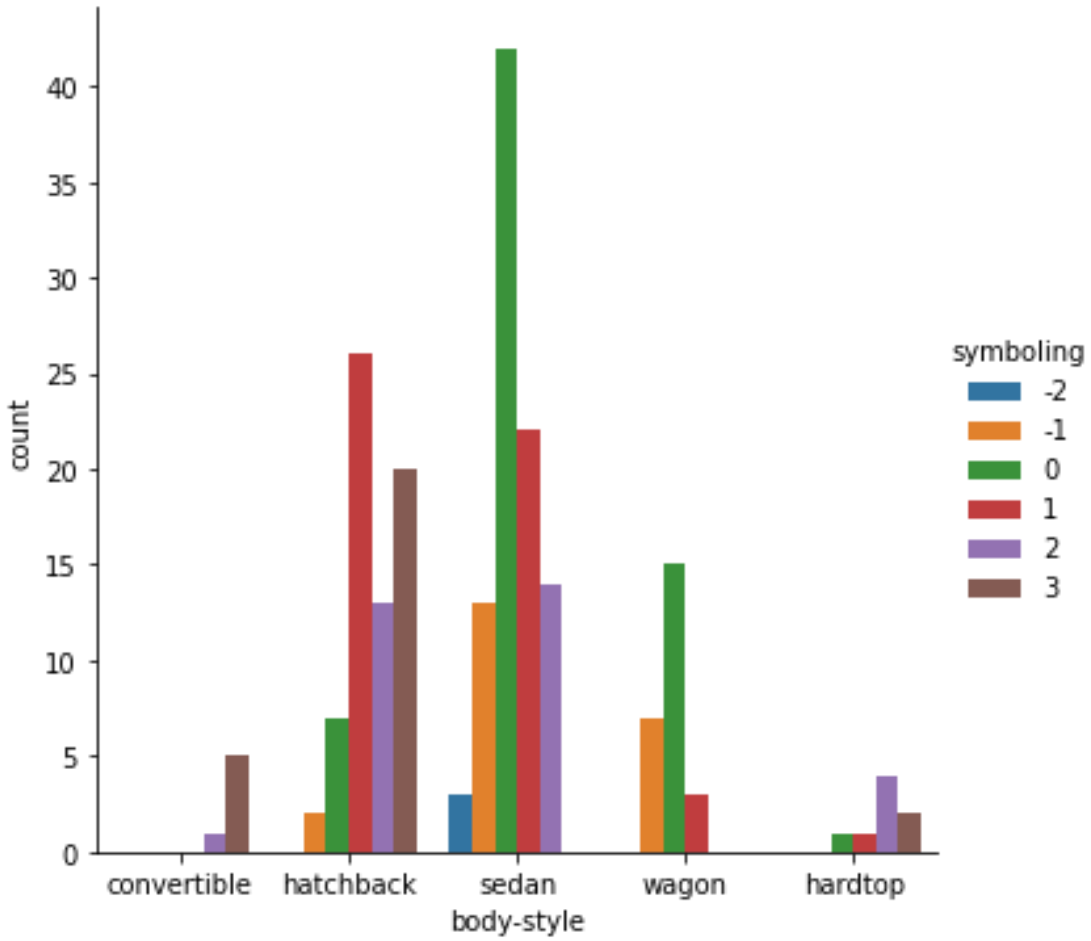
The data set contains more low-priced vehicles than high-priced vehicles. More data on high-priced vehicles is required in order to determine any relationship or pattern. Sedans are the most vehicles that are low-priced. The dataset contains only low-priced hatchbacks.

Visualise the horsepower of high-priced vehicles and low-priced vehicles. Further indicate the distribution of the vehicles between naturally aspirated vehicles and turbocharged vehicles. Low-priced sedans have a higher horsepower if they are turbo charged. However, high-priced sedans have a higher horsepower especially if the vehicle is naturally aspirated. Turbocharged hatchbacks in the low-price range have the highest horsepower compared to other vehicle types in both price ranges. Naturally aspirated hardtops have the highest horsepower within the dataset.

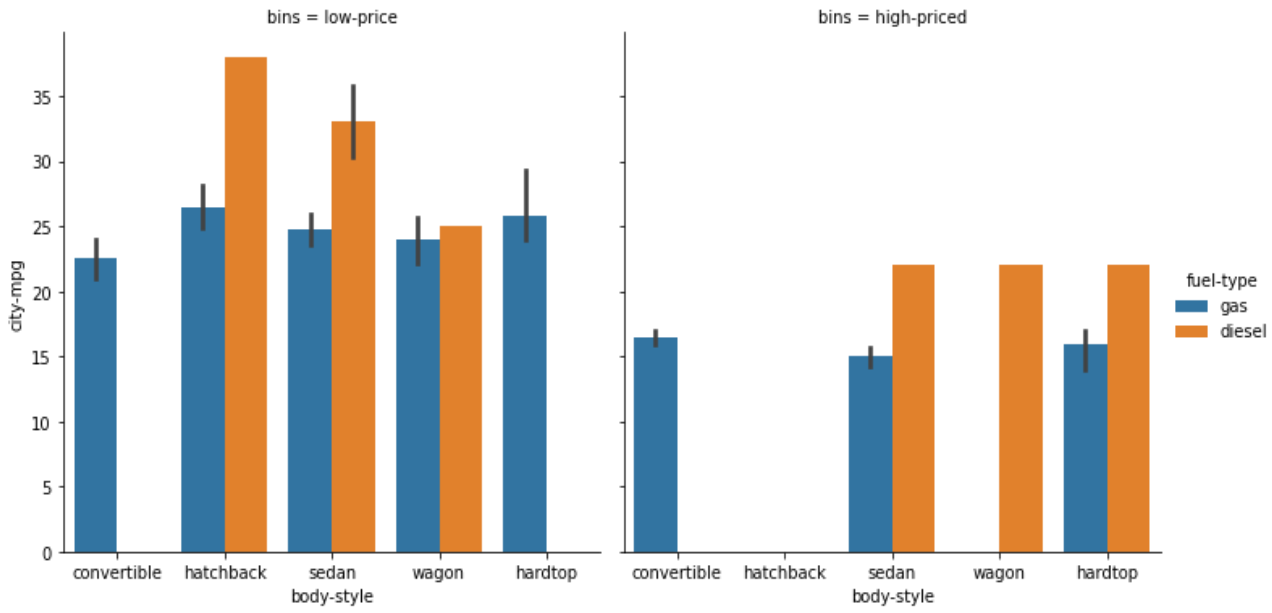


Each vehicle within the dataset is assigned a risk value between -3 and 3, where -3 is safe and +3 is risky. The dataset contains no vehicles that are assigned (-3) safe, the safest vehicles are high-priced wagons. The riskiest vehicles are convertibles regardless of the price. Sedans are less risky as the price of the vehicle increases.





The bar graph above indicates the number of vehicles per risk value. Convertibles are riskier and are assigned 2 or 3. A large number of sedans are neither risky nor safe, more data needs to be obtained to find the reason why they are assigned 0.



City-mpg indicates the miles per gallon of a vehicle when driving within a city. It refers to driving that includes the regular stopping and braking one does when driving within a city. A high mpg translates to more miles a vehicle can cover per gallon.

The bar graph above indicates that high priced sedans, wagons and hardtops that use diesel all have the same city-mpg. A cheaper hatchback that runs on diesel will give you more miles per gallon than any other vehicle type. Low priced vehicles that run on gas/petrol seem to have a higher city-mpg than the more expensive vehicles.

THIS REPORT WAS WRITTEN BY: Natacha Nsengiyumva