Exploring Insights in the Automobile Dataset

Overview

This report presents a comprehensive analysis of the Automobile Dataset, aiming to uncover valuable insights and patterns within the data. The dataset contains information about various aspects of automobiles, including technical specifications, manufacturer details, and pricing.

Objective

The primary objectives of this analysis are:

Explore Data Distribution: Investigate the distribution of numeric and categorical features to understand the overall structure of the dataset.

Identify Correlations: Analyze relationships between different variables, particularly focusing on factors that may influence the pricing of automobiles.

Visualize Patterns: Use visualization techniques to highlight patterns, trends, and potential outliers in the data.

Inform Decision-Making: Derive meaningful conclusions and insights that can inform decision-making processes related to automobile characteristics and pricing.

Dataset Description

The dataset comprises 26 columns, encompassing a mix of numerical and categorical variables. Some of the key features include the make of the car, technical specifications like engine size and horsepower, and pricing information.

```
In [60]: df.columns.values

Out[60]: array(['symboling', 'normalized-losses', 'make', 'fuel-type',
                'aspiration', 'num-of-doors', 'body-style', 'drive-wheels',
                'engine-location', 'wheel-base', 'length', 'width', 'height',
                'curb-weight', 'engine-type', 'num-of-cylinders', 'engine-size',
                'fuel-system', 'bore', 'stroke', 'compression-ratio', 'horsepower',
                'peak-rpm', 'city-mpg', 'highway-mpg', 'price'], dtype=object)
```

```
df.shape
```

```
(205, 26)
```

```
df.head(2)
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47

stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
2.68	9.0	111	5000	21	27	13495
2.68	9.0	111	5000	21	27	16500

Finding missing values

It is found that the missing values in the dataset represented by the “?”, so we replace these values with “nan”.

```
In [8]: df.replace('?', np.nan, inplace = True)
        df
```

Out[8]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19

```
df.isnull().sum()
```

```
symboling             0
normalized-losses    41
make                  0
fuel-type             0
aspiration            0
num-of-doors          2
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                  4
stroke                4
compression-ratio     0
horsepower            2
peak-rpm              2
city-mpg              0
highway-mpg           0
price                 4
dtype: int64
```

Based on the summary above, following columns has missing values:

1) "normalized-losses": 41 missing data

2) "num-of-doors": 2 missing data

3) "bore": 4 missing data

4) "stroke" : 4 missing data

5) "horsepower": 2 missing data

6) "peak-rpm": 2 missing data

7) "price": 4 missing data

Convert Numerical to float

I converted the specified columns to float (e.g., 'normalized-losses', 'bore', 'stroke', 'horsepower', 'peak-rpm', 'price) to facilitate mathematical operations, handle missing values, and ensure compatibility, as these tasks often require numeric input.

In [15]:
```
df['normalized-losses']=df['normalized-losses'].astype(float)
df['bore']=df['bore'].astype(float)
df['stroke']=df['stroke'].astype(float)
df['horsepower']=df['horsepower'].astype(float)
df['peak-rpm']=df['peak-rpm'].astype(float)
df['price']=df['price'].astype(float)
```

Dealing with missing values

We replace the missing values in column 'num-of-doors' with the mean values, column 'bore' and 'stroke' with their respective mode values, then we replace the missing values in column 'num-of-doors' with the value 'four' as it is found to be most frequent value in the column.

```
In [16]: df["horsepower"].replace(np.nan, 112, inplace=True)
```

```
In [17]: df["num-of-doors"].replace(np.nan, "four", inplace=True)
```

```
In [25]: mean_peakRPM = df['peak-rpm'].mean()
```

```
In [26]: df["peak-rpm"].replace(np.nan, mean_peakRPM, inplace=True) #5100
```

```
In [32]: mode_bore = df['bore'].mode()[0]
```

```
In [33]: df["bore"].replace(np.nan, mode_bore, inplace=True)
```

```
In [39]: mode_stroke = df['stroke'].mode()[0]
```

```
In [40]: df["stroke"].replace(np.nan, mode_stroke, inplace=True)
```

```
In [42]: median_norm_losses = df['normalized-losses'].median()
```

```
In [43]: df["normalized-losses"].replace(np.nan, median_norm_losses, inplace=True)
```

```
In [45]: df.dropna(subset=["price"], axis=0, inplace=True)

         # reset index, because we droped rows
         df.reset_index(drop=True, inplace=True)
```

Visualizations

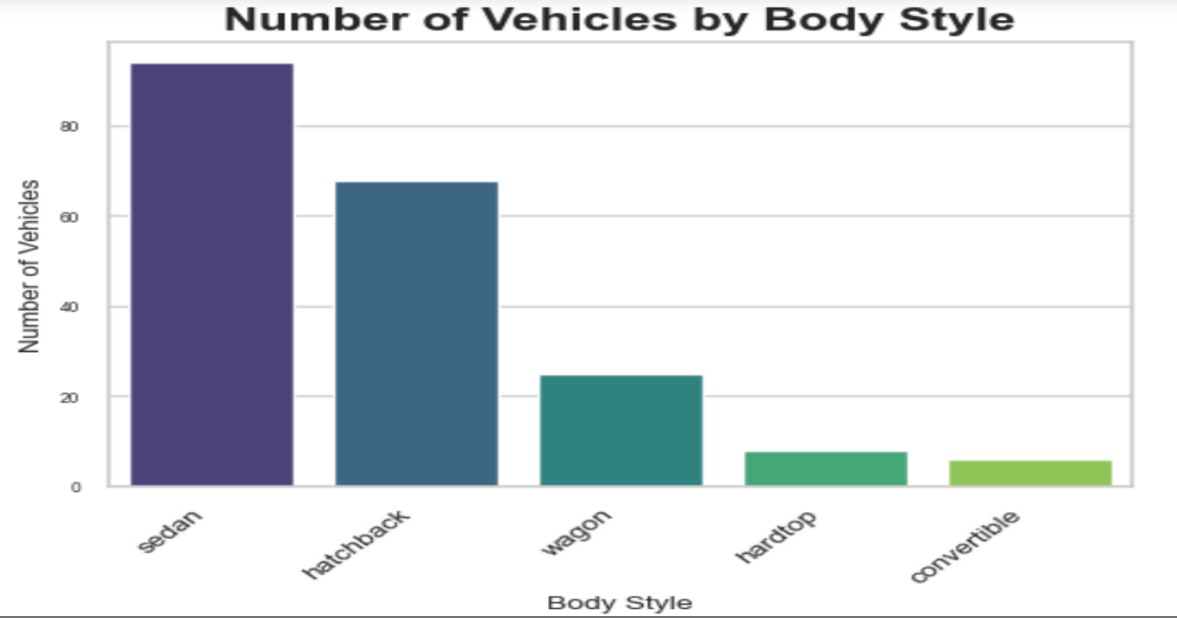


Figure 1

Its comparison between the body style and the number of vehicles. It was found that the car with body style 'sedan' has the most number of count whereas 'convertible' has the least. Hatchback was found to be the second favorite body style model.

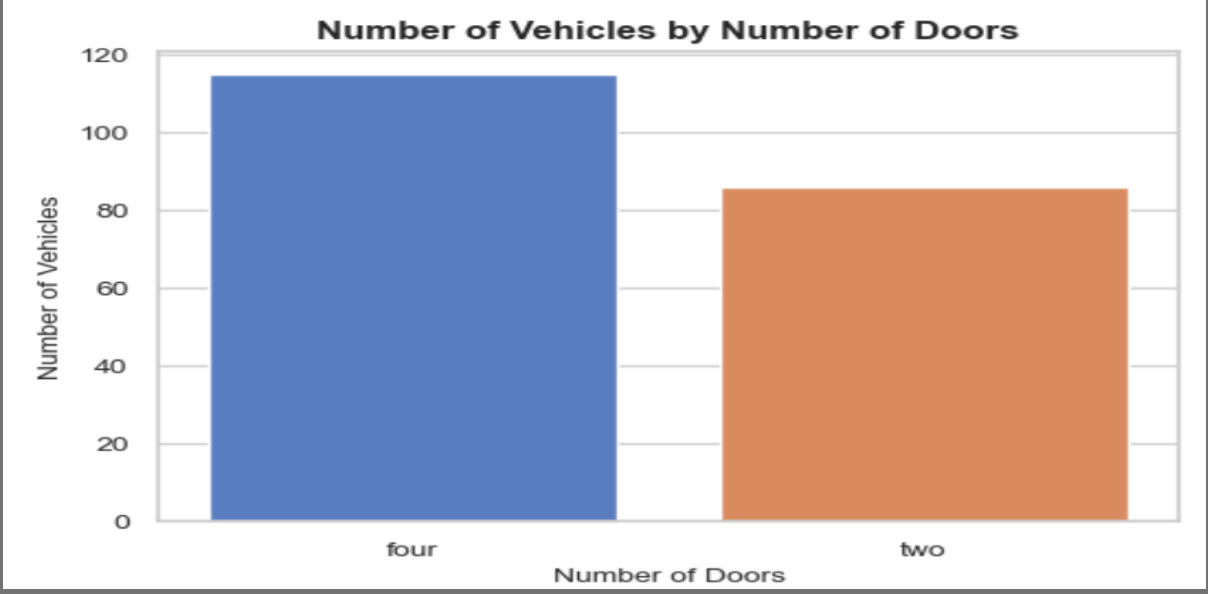


Figure 2

Its comparison between the door style and the number of vehicles. It was found that the most car has the door style 'four'. Two door car models are comparatively low.

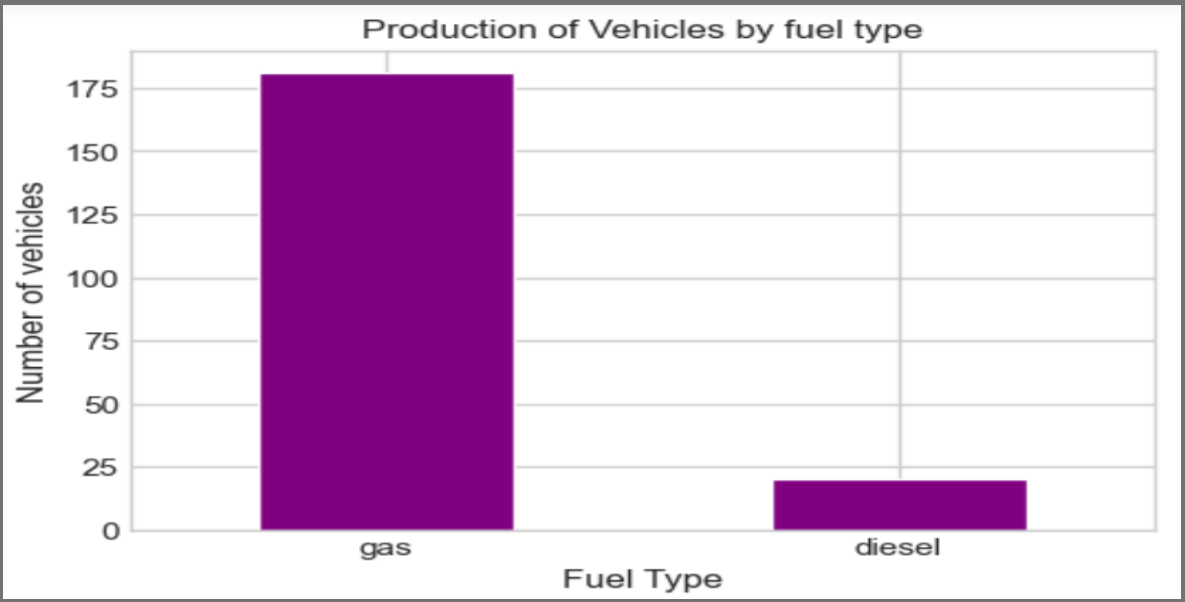


Figure 3

Its comparison between the fuel type and the number of vehicles. It was found that the most cars have gas/petrol as its fuel. Surprisingly, count of cars with diesel as fuel is comparatively very negligible.

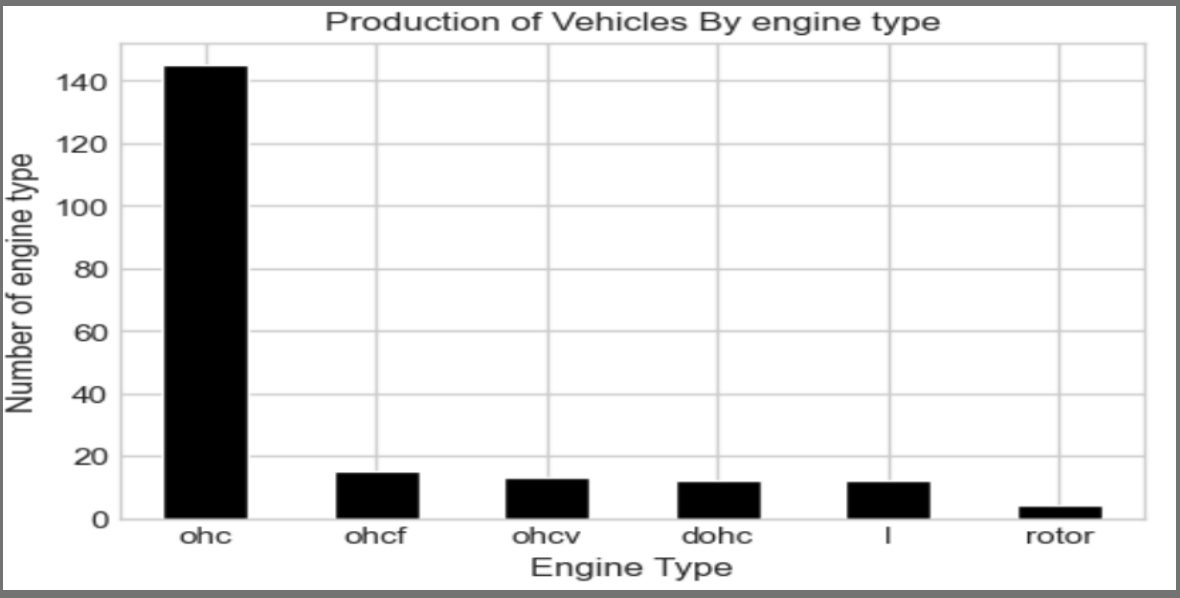


Figure 4

Its comparison between the engine and the number of vehicles. It was found that the most cars have engine style ‘ohc’ .Dohcv has the least number of count. Huge difference were seen with 1st and 2nd engines. Clear dominance of ohc engine was clearly visible.

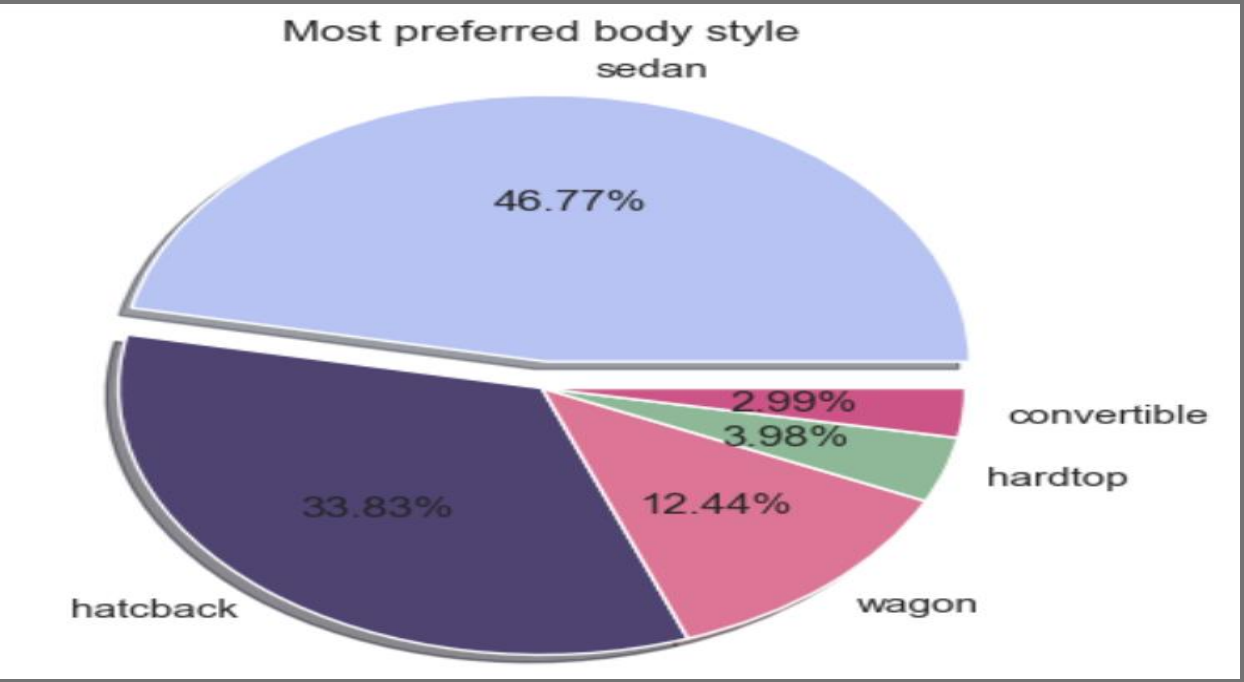


Figure 5

It is found that the most preferable body style among all is the sedan, and less prefared one is convertible.

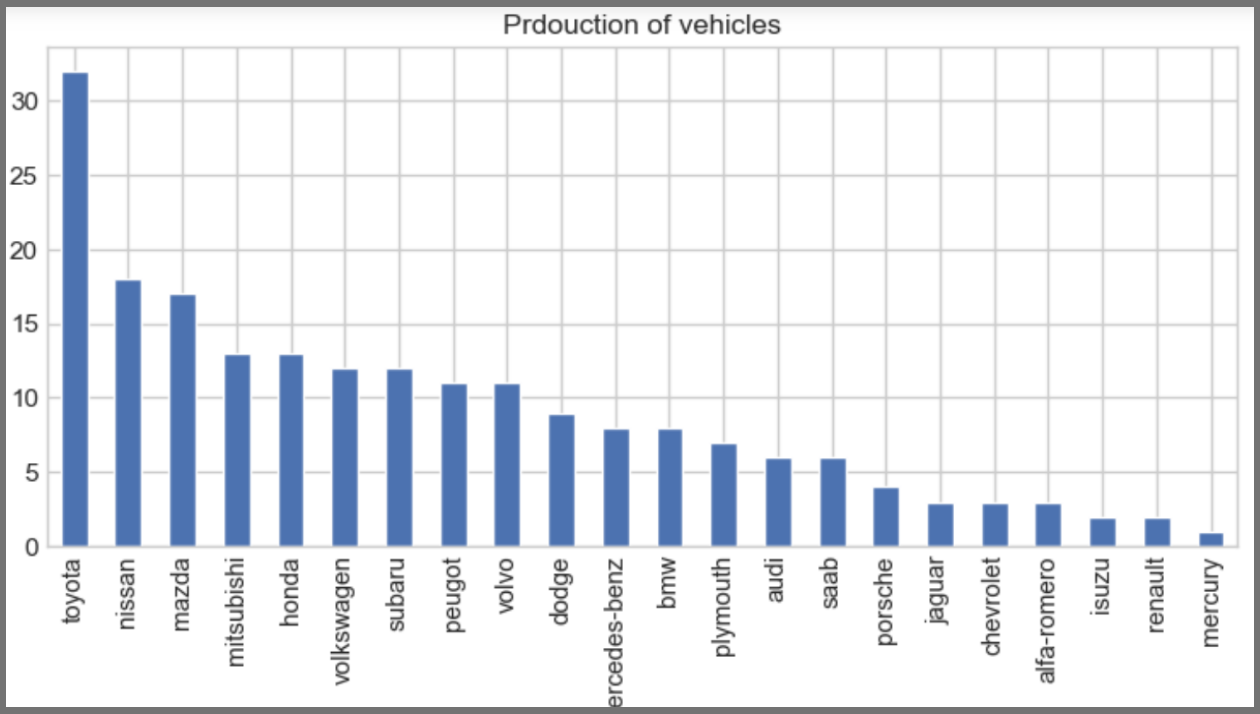


Figure 6

The figure above shows that the Toyota is in the top in the production of vehicles following Nissan placed second position, and the company produced the less vehicle during the period is Mercury.

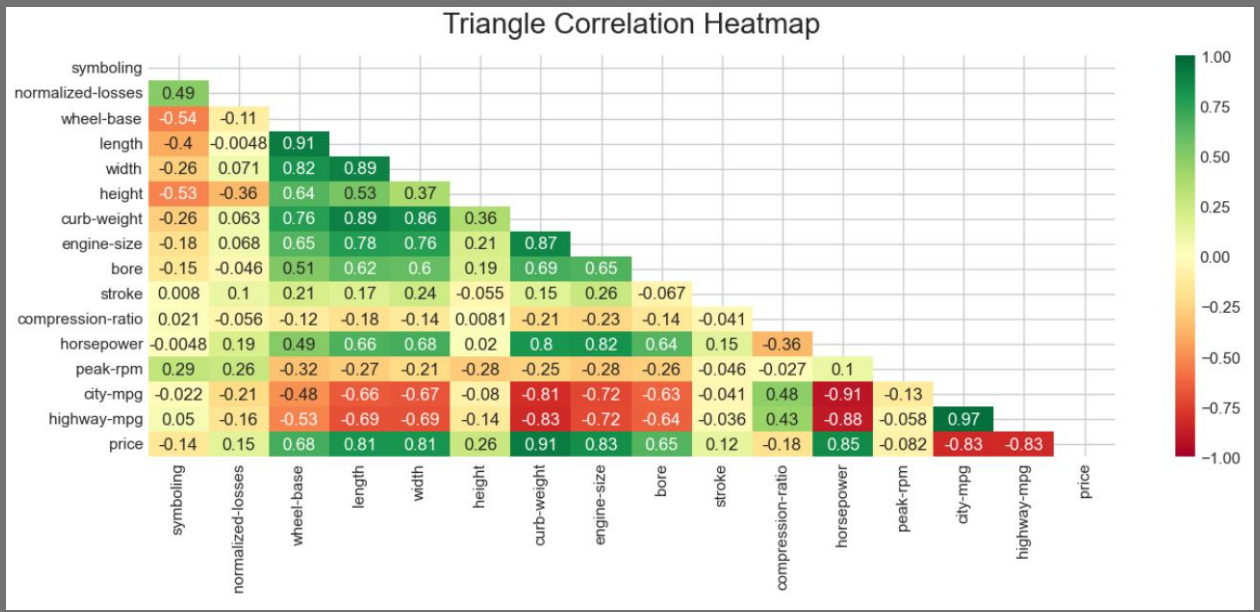


Figure 7

In the Numerical data, according to the correlation heatmap, we came to the conclusion that the features of length, width, weight, curb-weight, engine size, bore and horsepower have a direct relationship with the price, and the city-mpg and highway-mpg features have the opposite relationship with price.

In the Category data, the car price is higher in the cars with the engine in the rear of the car. Also, the price of the car increases with the increase in the number of cylinders.

BMW, Benz, Porsche and Jaguar manufacturers have cars with higher prices.

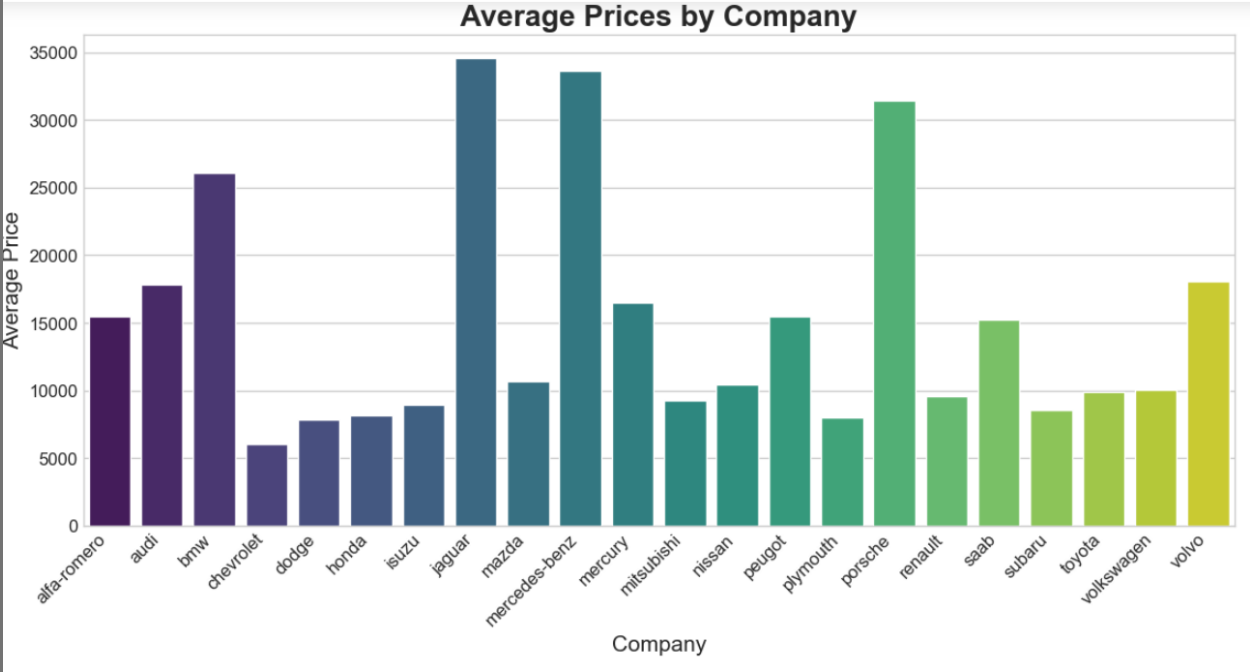
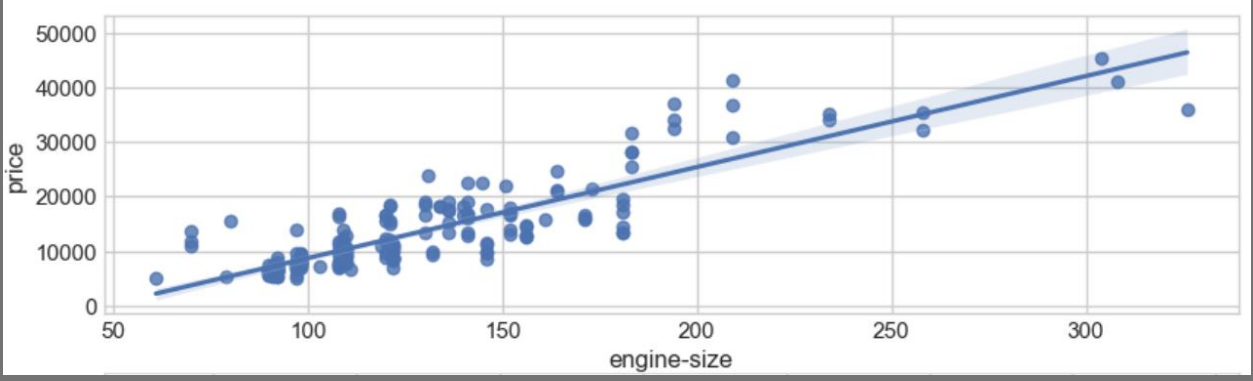
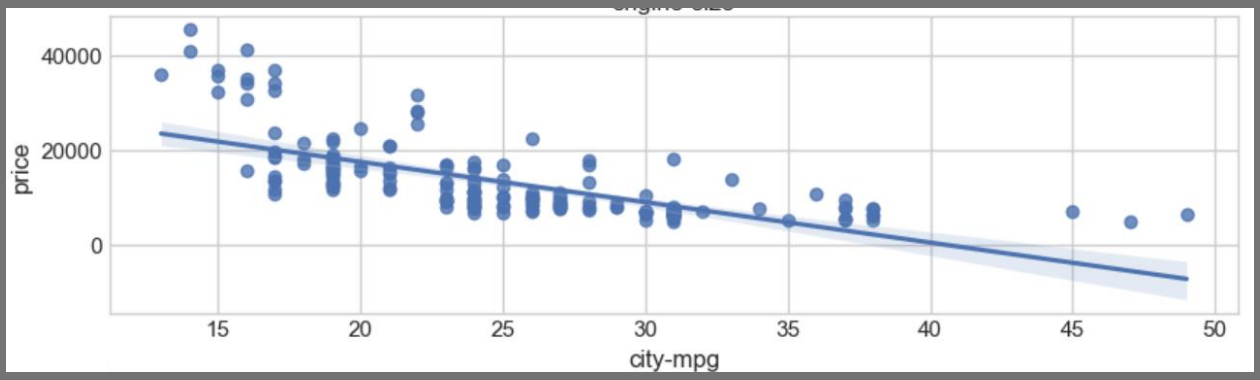
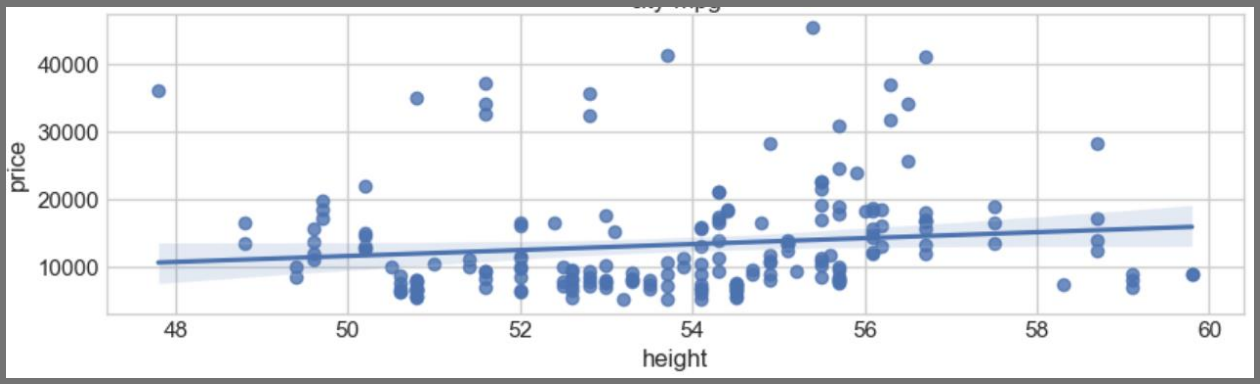


Figure 8







In the first plot, as the engine size increases, so does the price: This indicates a positive direct correlation between these two variables. Since the regression line is almost a perfect diagonal line, engine size appears to be a pretty good price predictor.

In the second plot, as city-mpg goes up, the price goes down: This indicates an inverse/negative relationship between these two variables. City mpg could potentially be a predictor of price.

In the thirth plot,height does not seem like a good predictor of the price at all since the regression line is close to horizontal. Also, the data points are very scattered and far from the fitted line, showing lots of variability. Therefore, it's not a reliable variable.