Implementation & Results:-

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sb
```

```python
df = pd.read_csv('/content/automobile.csv')
df.head()
```

- **automobile.csv**(text/csv) - 35648 bytes, last modified: 10/25/2024 - 100% done
  Saving automobile.csv to automobile (1).csv

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	city-L/100km	ho
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27	13495.0	11.190476	
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27	16500.0	11.190476	
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...	9.0	154.0	5000.0	19	26	16500.0	12.368421	
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...	10.0	102.0	5500.0	24	30	13950.0	9.791667	
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...	8.0	115.0	5500.0	18	22	17450.0	13.055556	

5 rows × 29 columns

Value Counts :- Value counts is a method used to summarize the frequency of each unique value in a dataset. It provides a count of how many times each value appears, which is particularly useful for categorical data analysis.

```python
df['drive-wheels'].value_counts()
```
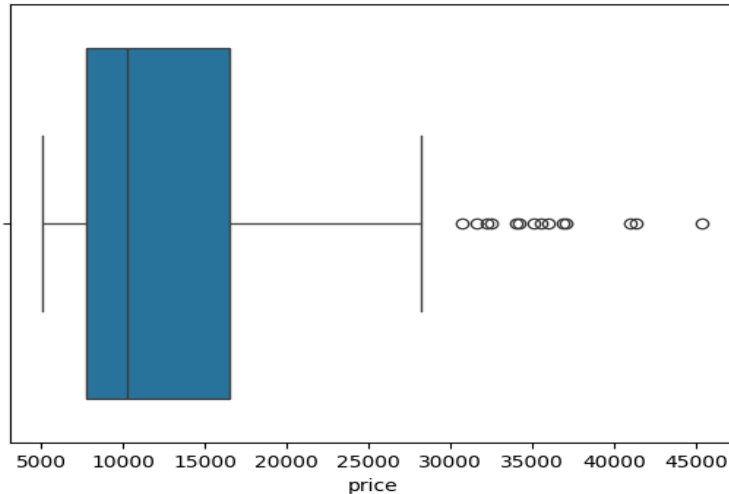
```
               count
drive-wheels
       fwd      118
       rwd       75
       4wd        8

dtype: int64
```

Box Plot:- A box plot (or whisker plot) is a standardized way of displaying the distribution of data based on a five-number summary: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. It visually shows the spread and identifies potential outliers in the dataset.

```python
sb.boxplot(x = df['price'])
```

Scatterplot :- A scatterplot is a graph that displays values for typically two variables for a set of data. Each point represents an observation in the dataset, with one variable determining the position on the x-axis and the other on the y-axis. Scatterplots are useful for identifying correlations or relationships between variables.

sb.scatterplot(x = df['horsepower'], y = df['price'])


```
<Axes: xlabel='horsepower', ylabel='price'>
```
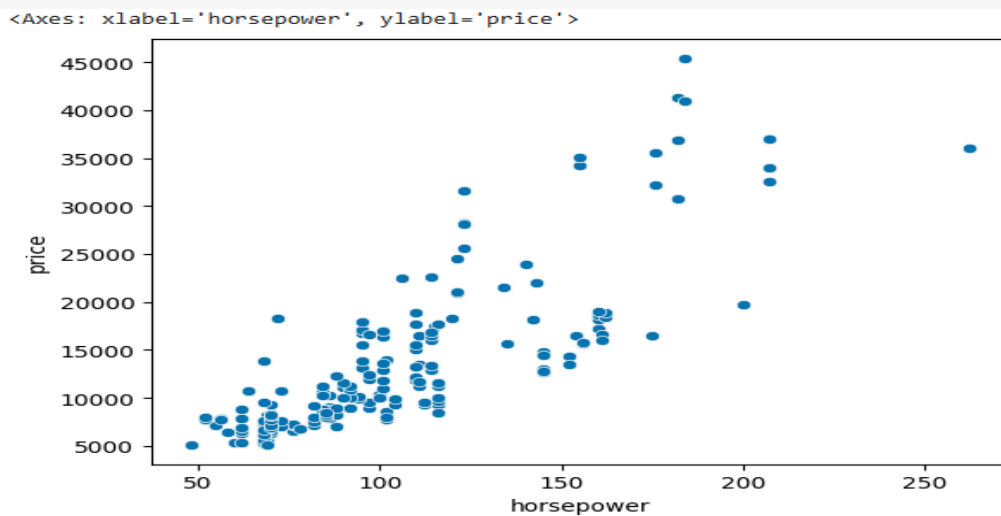
sb.scatterplot(x = df['highway-mpg'], y = df['price'])

```
<Axes: xlabel='highway-mpg', ylabel='price'>
```
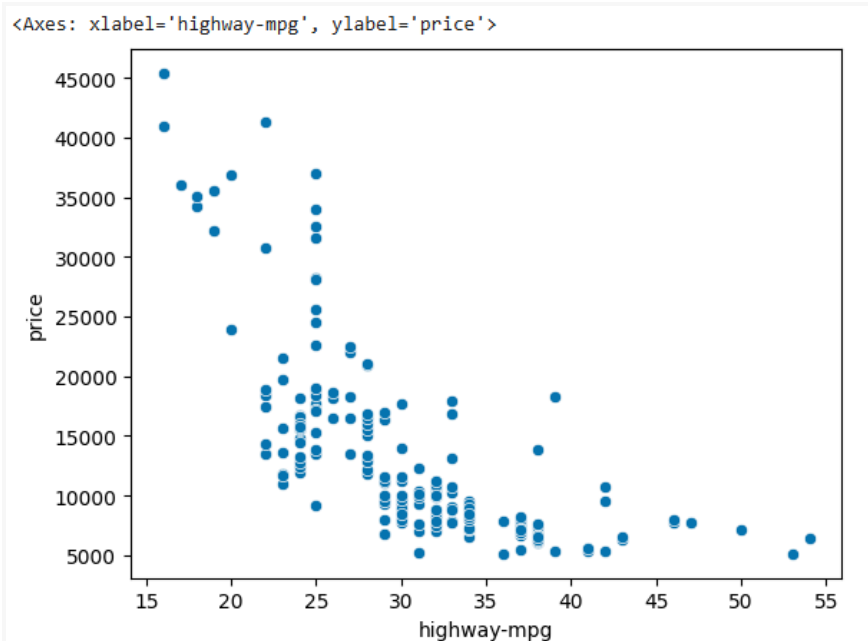

Group By :- The group by function is used in data analysis to aggregate and summarize data. It allows you to group a dataset by one or more columns and then perform calculations (such as mean, sum, or count) on each group, providing insights into patterns and trends within the data.

df_test = df[['drive-wheels', 'body-style', 'price']]

df_test.head()

df_grp = df_test.groupby(['drive-wheels', 'body-style'], as_index=False).mean()

Df_grp

	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

Pivot Table :- A pivot table is a data processing tool used to summarize and analyze data. It allows you to reorganize and group data to extract meaningful insights by aggregating

information using functions like sum, average, or count across different dimensions and categories.
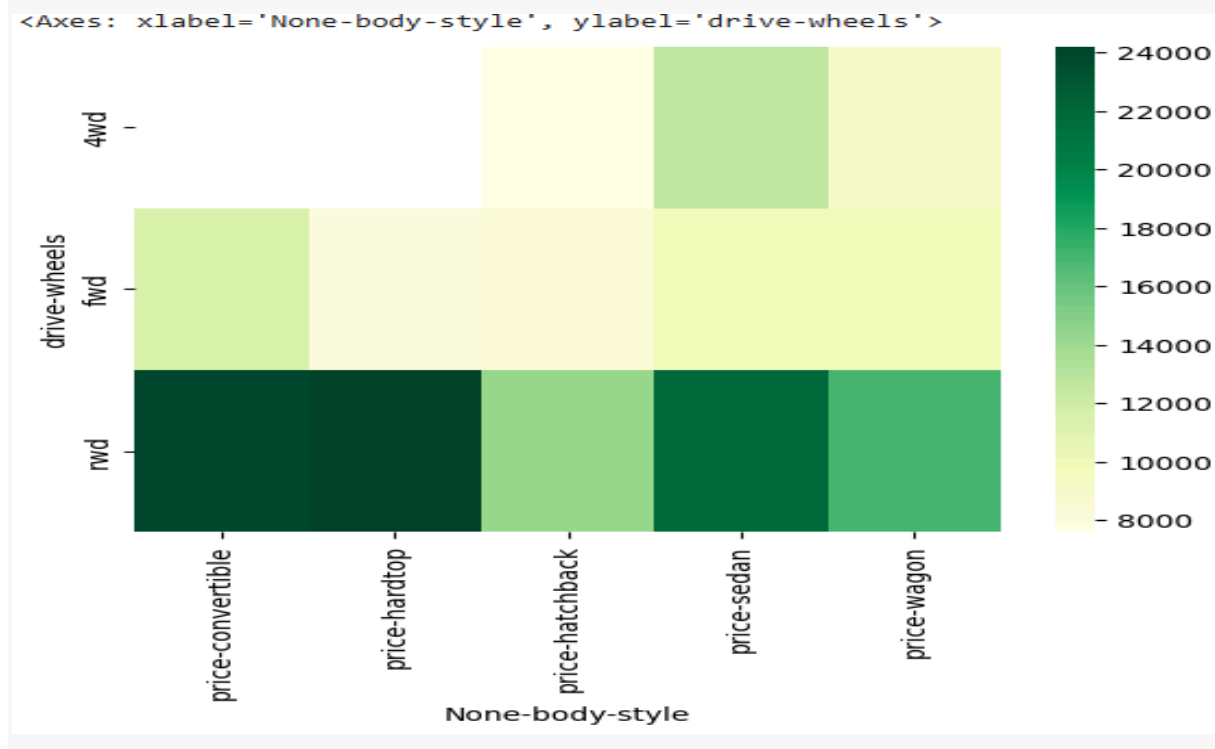
df_pivot = df_grp.pivot(index = 'drive-wheels', columns = 'body-style')

Df_pivot

	price				
body-style	convertible	hardtop	hatchback	sedan	wagon
drive-wheels					
4wd	NaN	NaN	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

Heat Map:- A heat map is a data visualization technique that uses color to represent the values of a matrix or a dataset. It is often used to illustrate the magnitude of values in a two-dimensional space, highlighting areas of higher or lower values, making it easier to spot patterns, correlations, or areas needing attention.

sb.heatmap(df_pivot, cmap= 'YlGn')