Introduction

In the evolving automotive market, understanding the factors that influence car pricing is crucial for buyers, sellers, and dealerships. This project focuses on predicting car prices using a dataset named cars.csv By analyzing various features of vehicles, our goal is to develop a predictive model that can accurately estimate car prices based on historical data and current market attributes.

Objectives

The primary objectives of this project are:

- **To Build a Predictive Model**: Develop a model that forecasts car prices based on features such as make, model, engine size, horsepower, and other relevant attributes.
- **To Understand Pricing Influences**: Explore and analyze how different features affect car prices, providing insights into the factors driving market value.
- **To Provide Actionable Insights**: Offer valuable information for buyers and sellers to make informed decisions regarding car purchases and sales.

Dataset Description

The cars.csv dataset provides a comprehensive snapshot of various vehicles, capturing essential details that influence pricing. It includes 100 rows of data, each representing an individual car listing with a range of features. This dataset allows us to explore how attributes such as engine size, horsepower, and vehicle condition impact the market price of cars.

Dataset Columns and Data Types

The cars.csv dataset includes various features that capture essential attributes of vehicles. Each feature is associated with a specific data type. Below is a description of each column along with its data type:

Column	Data Type	Description
Company	Categorical/String	The manufacturer or brand of the car (e.g., Toyota, Ford).
Model	Categorical/String	The specific model of the car (e.g., Camry, Mustang).
Year	Integer	The year the car was manufactured.
Engine Size (L)	Float	The size of the car's engine, measured in liters.
Horsepower	Integer	The power output of the car’s engine, typically measured in horsepower.
Torque	Float	The amount of rotational force produced by the engine, measured in Nm or lb-ft.
0-60 MPH Time (seconds)	Float	The time it takes for the car to accelerate from 0 to 60 miles per hour.
Price	Float	The listing price of the car, usually represented in the local currency.
Fuel	Categorical/String	The type of fuel used by the car (e.g., petrol, diesel, electric).
Colour	Categorical/String	The color of the car.
No of seats	Integer	The number of seats in the car.
Drive type	Categorical/String	The type of drive system (e.g., front-wheel drive, all-wheel drive).
Body type	Categorical/String	The style of the car's body (e.g., sedan, SUV, hatchback).
Top speed (kmph)	Float	The maximum speed the car can achieve, measured in kilometers per hour.

Significance of the Project

Accurate prediction of car prices is highly valuable in the automotive industry. For buyers, it provides a clear understanding of the fair market value, helping in making informed purchasing decisions. For sellers and dealerships, it aids in setting competitive prices and optimizing sales strategies. This project leverages data-driven insights to enhance decision-making and improve the overall car buying and selling experience.

2. Methodology

1 Data Collection

- **Objective**: Acquire the `cars.csv` dataset containing various car attributes and prices.

2 Data Preprocessing

- **Objective**: Clean and prepare the data by handling missing values, transforming categorical variables, and scaling features.

3 Exploratory Data Analysis (EDA)

- **Objective**: Analyze and visualize the dataset to understand distributions, relationships, and patterns in the data.

4 Feature Engineering and Selection

- **Objective**: Create new features and select the most relevant ones to improve model accuracy.

5 Model Building

- **Objective**: Develop and train different predictive models to estimate car prices.

6 Model Evaluation

- **Objective**: Assess model performance using metrics and validation techniques to choose the best model.

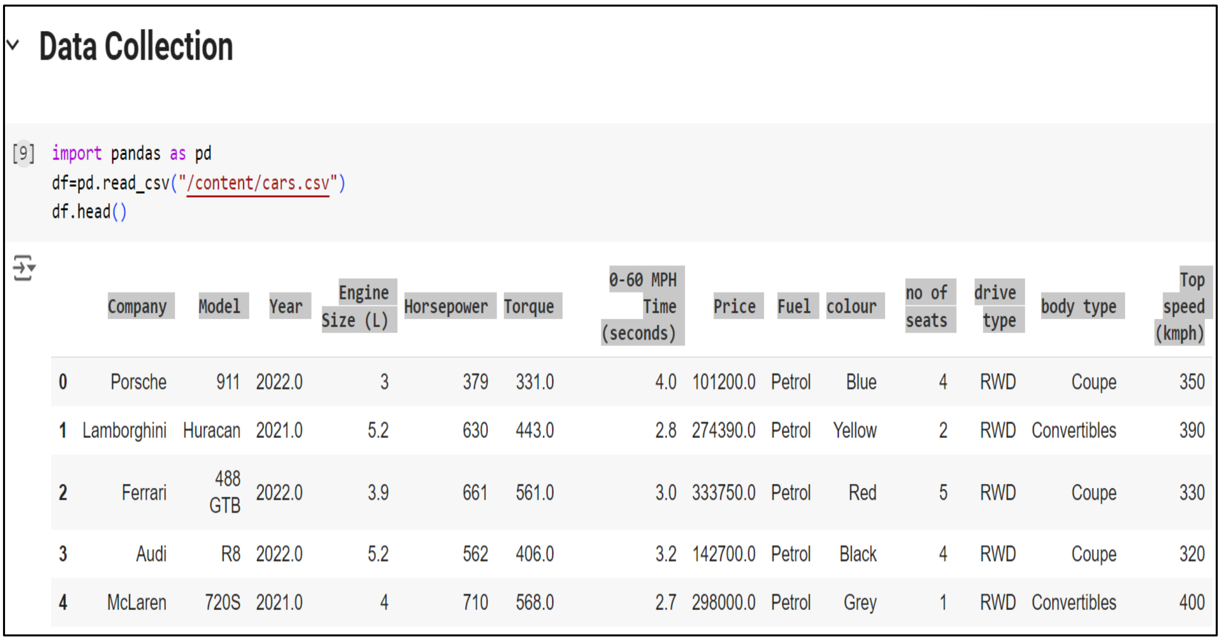
7 Insights and Recommendations

- **Objective**: Interpret the model results to provide actionable insights and recommendations for buyers and sellers.

1) Understand Business Problem

The business problem in this predictive analysis project on car pricing is to accurately estimate the market value of vehicles based on various attributes to enhance decision-making for buyers, sellers, and dealerships.

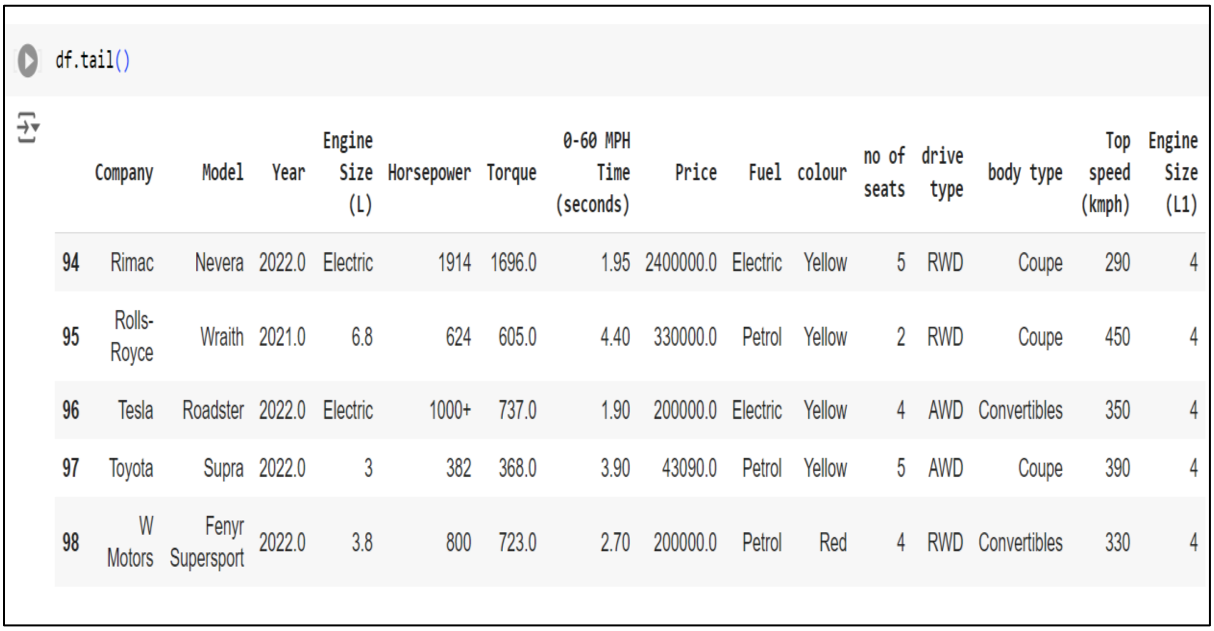
2) Data Collection



Data Collection

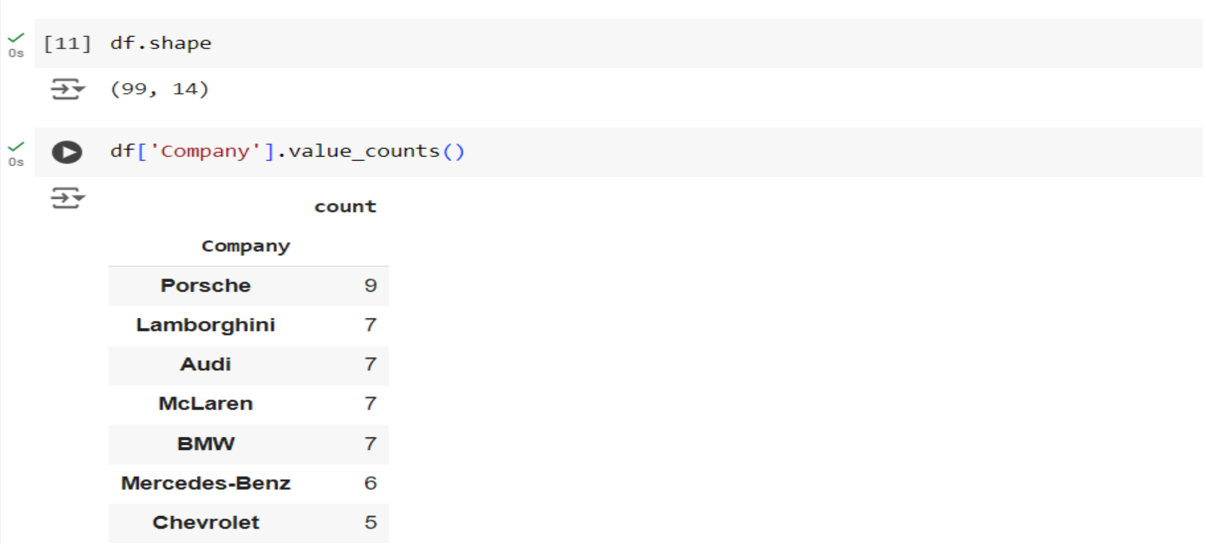
```
[9] import pandas as pd
    df=pd.read_csv("/content/cars.csv")
    df.head()
```

	Company	Model	Year	Engine Size (L)	Horsepower	Torque	0-60 MPH Time (seconds)	Price	Fuel	colour	no of seats	drive type	body type	Top speed (kmph)
0	Porsche	911	2022.0	3	379	331.0	4.0	101200.0	Petrol	Blue	4	RWD	Coupe	350
1	Lamborghini	Huracan	2021.0	5.2	630	443.0	2.8	274390.0	Petrol	Yellow	2	RWD	Convertibles	390
2	Ferrari	488 GTB	2022.0	3.9	661	561.0	3.0	333750.0	Petrol	Red	5	RWD	Coupe	330
3	Audi	R8	2022.0	5.2	562	406.0	3.2	142700.0	Petrol	Black	4	RWD	Coupe	320
4	McLaren	720S	2021.0	4	710	568.0	2.7	298000.0	Petrol	Grey	1	RWD	Convertibles	400



```
df.tail()
```

	Company	Model	Year	Engine Size (L)	Horsepower	Torque	0-60 MPH Time (seconds)	Price	Fuel	colour	no of seats	drive type	body type	Top speed (kmph)	Engine Size (L1)
94	Rimac	Nevera	2022.0	Electric	1914	1696.0	1.95	2400000.0	Electric	Yellow	5	RWD	Coupe	290	4
95	Rolls-Royce	Wraith	2021.0	6.8	624	605.0	4.40	330000.0	Petrol	Yellow	2	RWD	Coupe	450	4
96	Tesla	Roadster	2022.0	Electric	1000+	737.0	1.90	200000.0	Electric	Yellow	4	AWD	Convertibles	350	4
97	Toyota	Supra	2022.0	3	382	368.0	3.90	43090.0	Petrol	Yellow	5	AWD	Coupe	390	4
98	W Motors	Fenyr Supersport	2022.0	3.8	800	723.0	2.70	200000.0	Petrol	Red	4	RWD	Convertibles	330	4



```
[11] df.shape
```

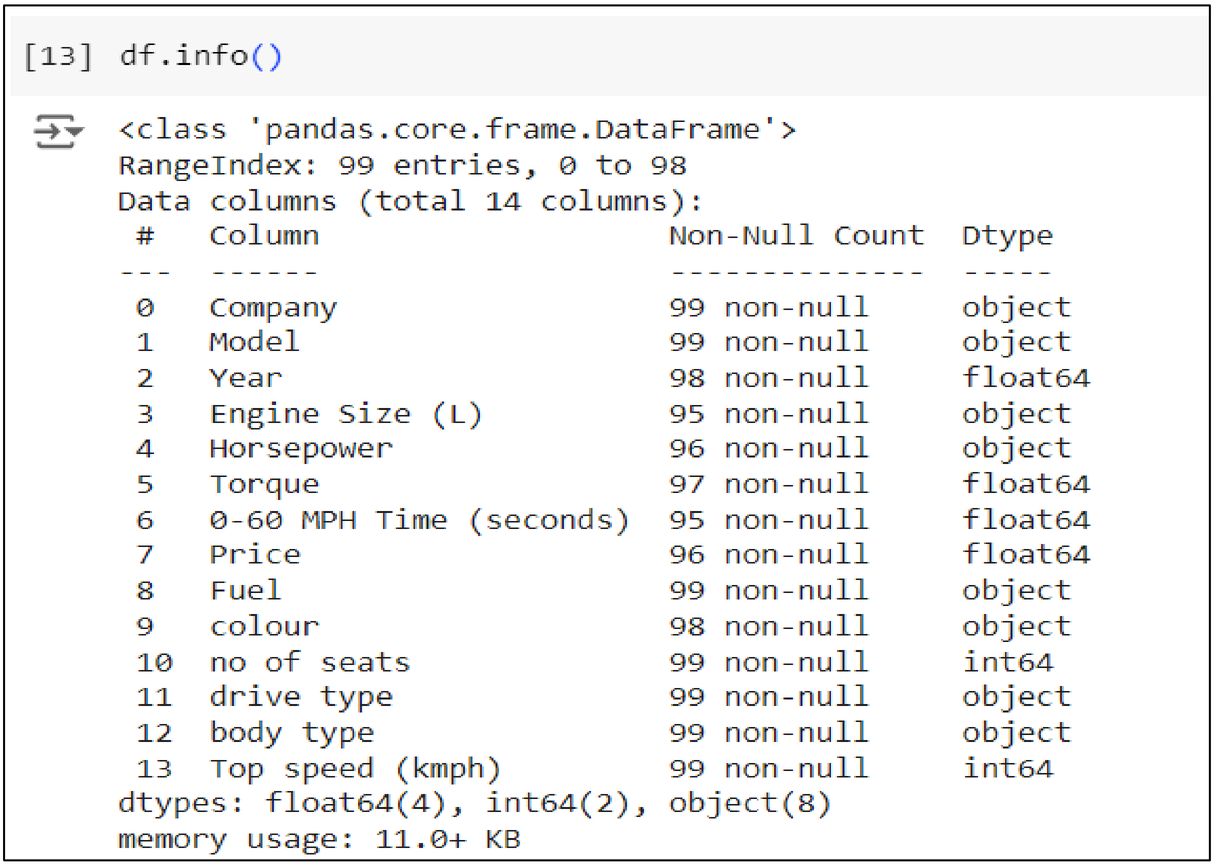
```
(99, 14)
```

```
df['Company'].value_counts()
```

Company	count
Porsche	9
Lamborghini	7
Audi	7
McLaren	7
BMW	7
Mercedes-Benz	6
Chevrolet	5

3) Data Preprocessing

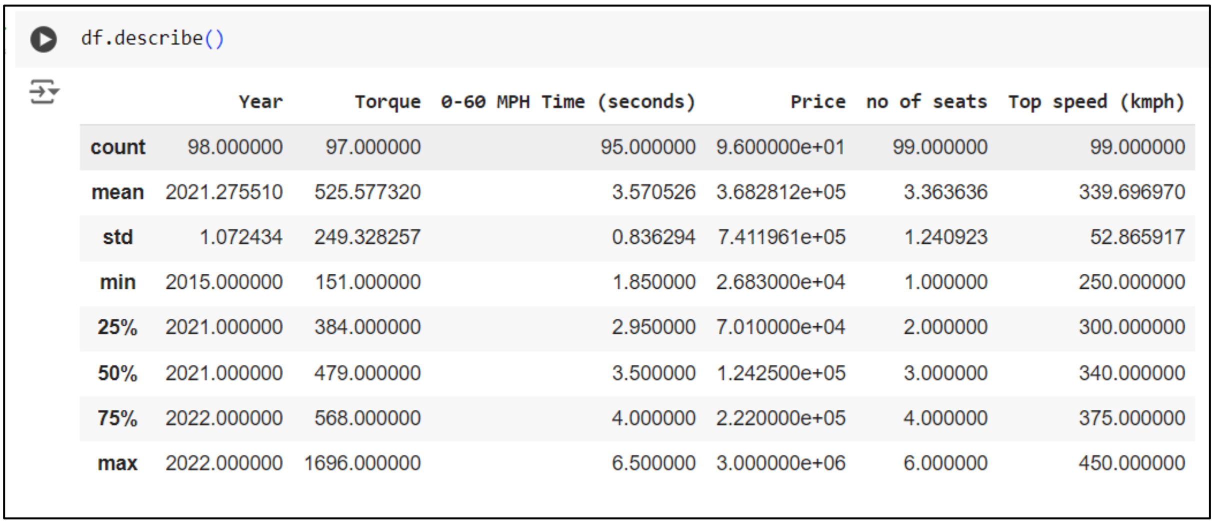
Getting information about dataset



```
[13] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Company                   99 non-null     object
 1   Model                     99 non-null     object
 2   Year                      98 non-null     float64
 3   Engine Size (L)           95 non-null     object
 4   Horsepower                96 non-null     object
 5   Torque                    97 non-null     float64
 6   0-60 MPH Time (seconds)   95 non-null     float64
 7   Price                     96 non-null     float64
 8   Fuel                      99 non-null     object
 9   colour                    98 non-null     object
 10  no of seats               99 non-null     int64
 11  drive type                99 non-null     object
 12  body type                 99 non-null     object
 13  Top speed (kmph)          99 non-null     int64
dtypes: float64(4), int64(2), object(8)
memory usage: 11.0+ KB
```

Getting summary of dataset

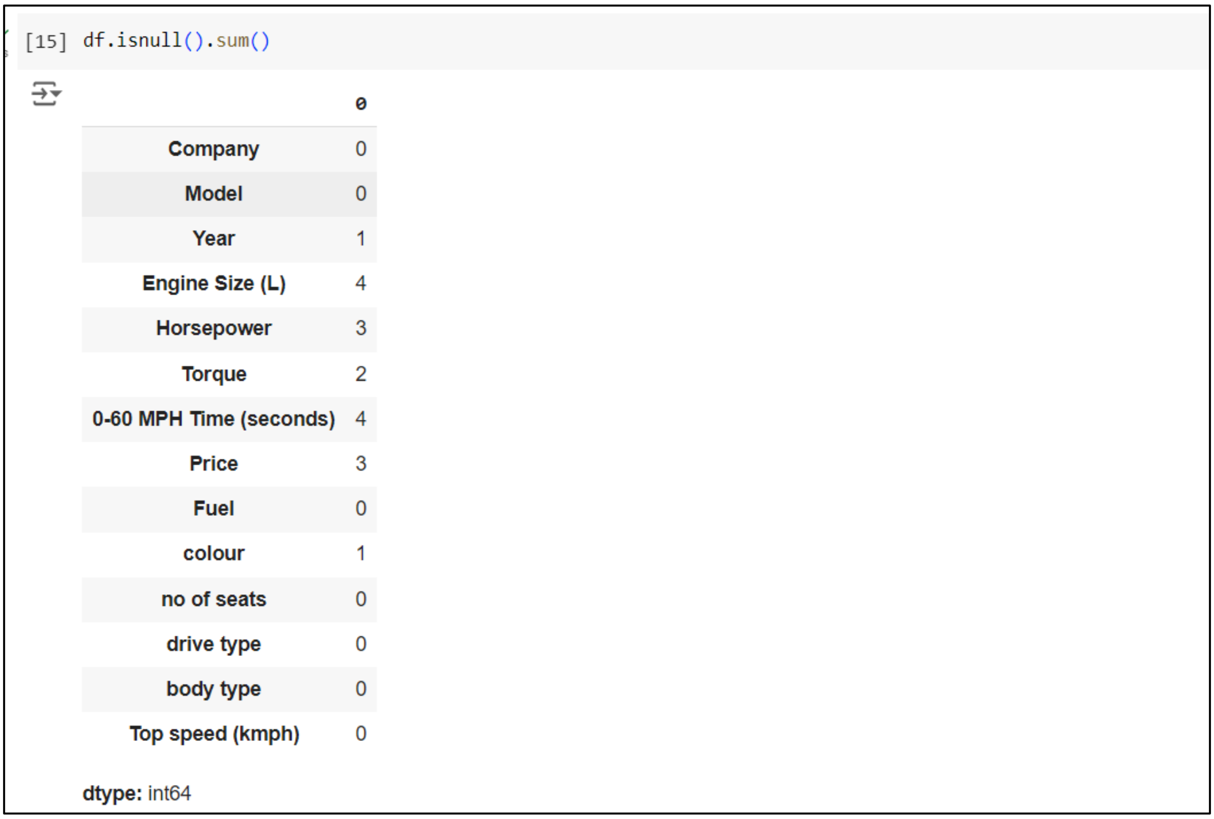


```
df.describe()
```

```
             Year       Torque  0-60 MPH Time (seconds)         Price  no of seats  Top speed (kmph)
count    98.000000    97.000000                95.000000  9.600000e+01    99.000000         99.000000
mean   2021.275510   525.577320                 3.570526  3.682812e+05     3.363636        339.696970
std       1.072434   249.328257                 0.836294  7.411961e+05     1.240923         52.865917
min    2015.000000   151.000000                 1.850000  2.683000e+04     1.000000        250.000000
25%    2021.000000   384.000000                 2.950000  7.010000e+04     2.000000        300.000000
50%    2021.000000   479.000000                 3.500000  1.242500e+05     3.000000        340.000000
75%    2022.000000   568.000000                 4.000000  2.220000e+05     4.000000        375.000000
max    2022.000000  1696.000000                 6.500000  3.000000e+06     6.000000        450.000000
```

4) Data Cleaning

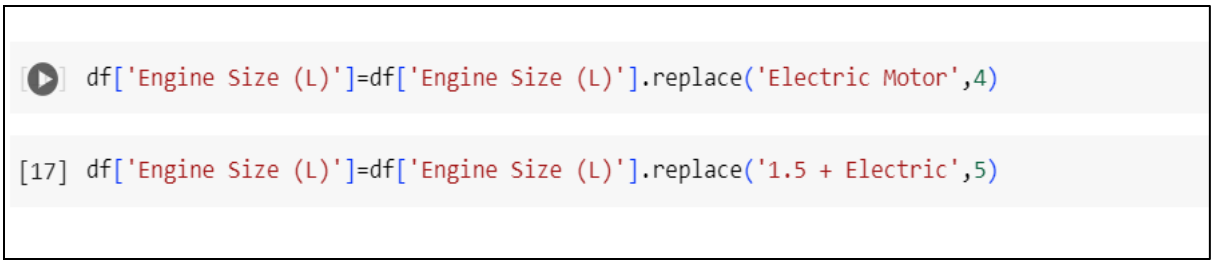
Checking the null values in dataset



```
[15] df.isnull().sum()
```

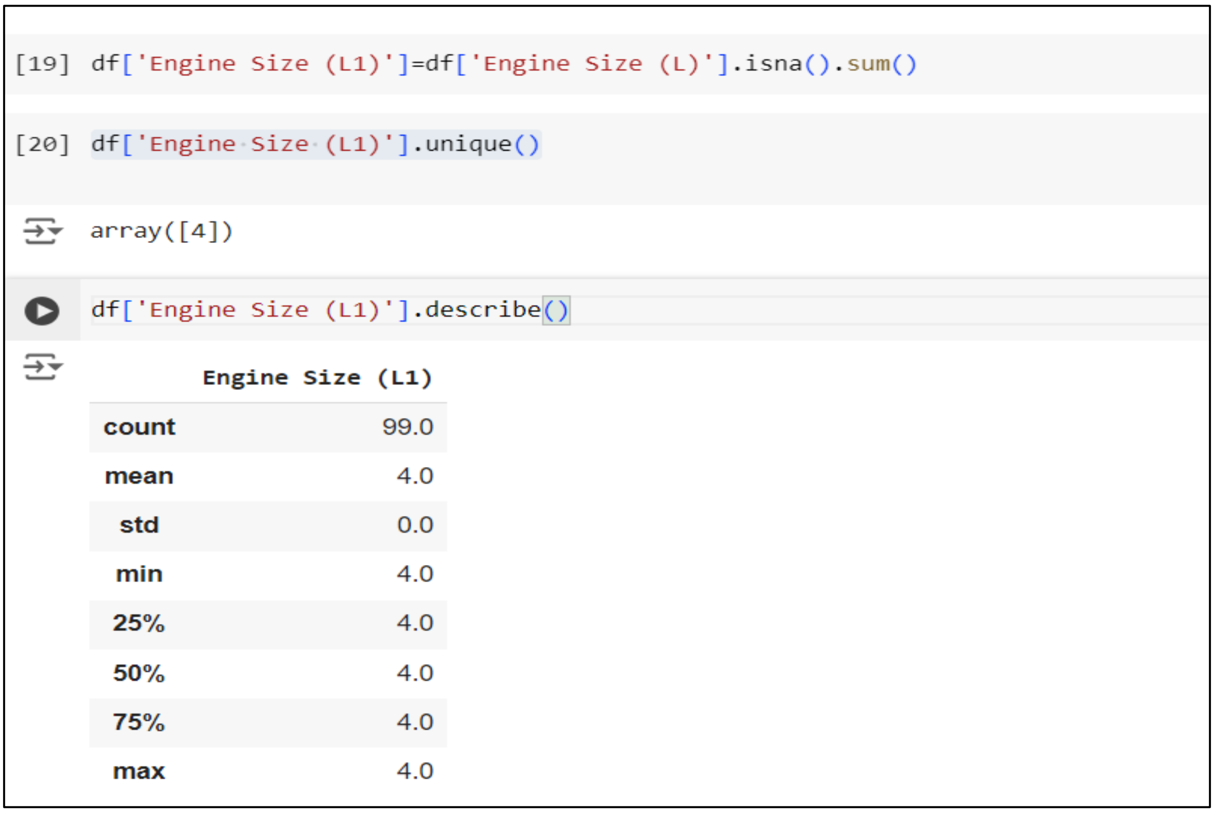
```
                          0
Company                   0
Model                     0
Year                      1
Engine Size (L)           4
Horsepower                3
Torque                    2
0-60 MPH Time (seconds)   4
Price                     3
Fuel                      0
colour                    1
no of seats               0
drive type                0
body type                 0
Top speed (kmph)          0

dtype: int64
```

Converting String into int 64



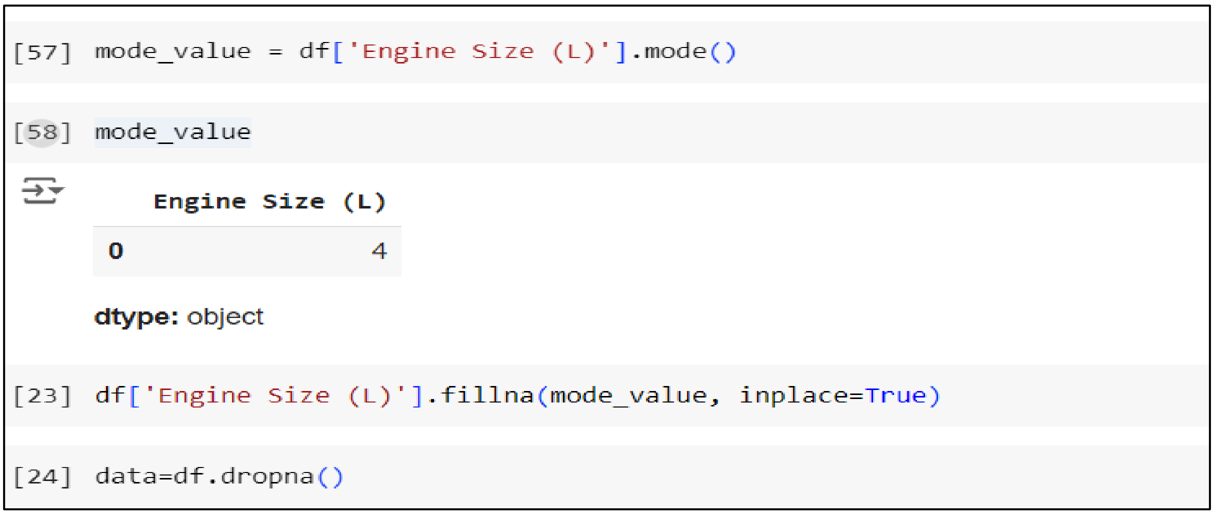
```
df['Engine Size (L)']=df['Engine Size (L)'].replace('Electric Motor',4)

[17] df['Engine Size (L)']=df['Engine Size (L)'].replace('1.5 + Electric',5)
```

Getting the mean median mode of Engine Size column



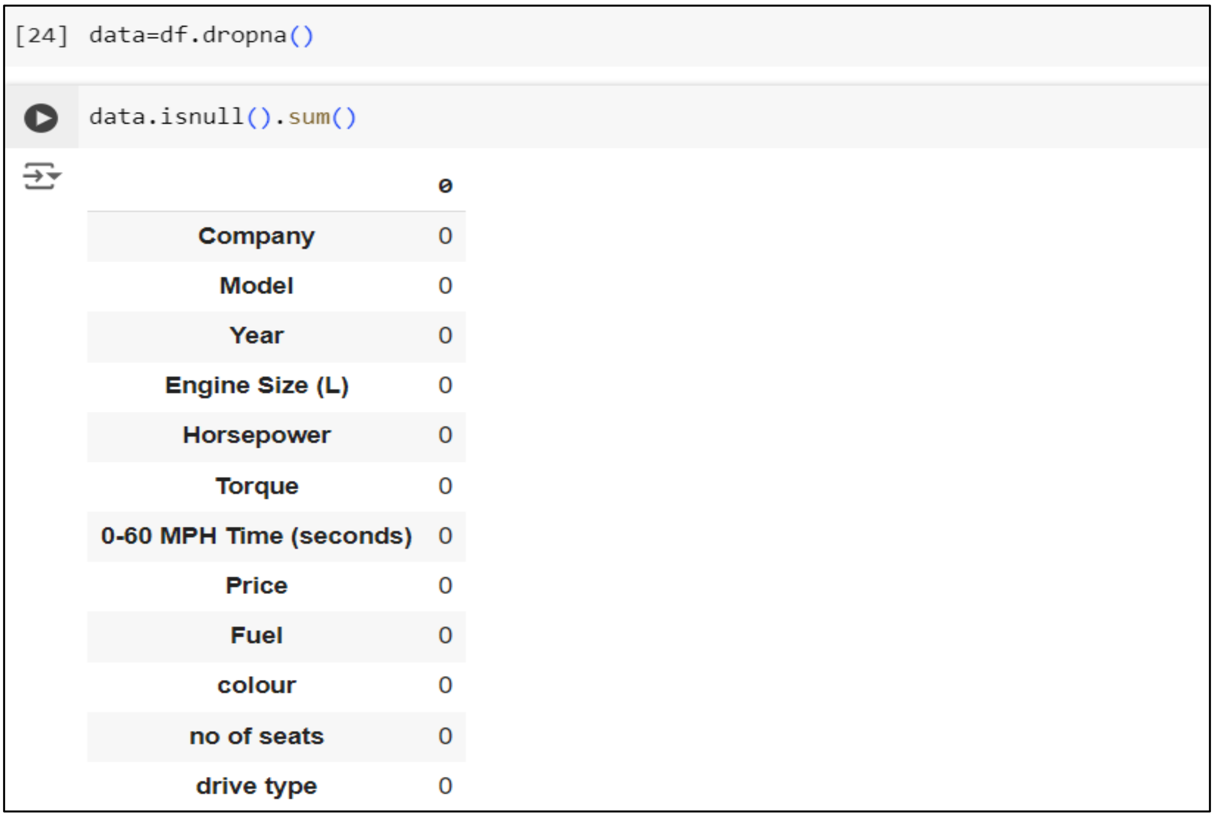
```
[19] df['Engine Size (L1)']=df['Engine Size (L)'].isna().sum()

[20] df['Engine Size (L1)'].unique()

array([4])

df['Engine Size (L1)'].describe()

       Engine Size (L1)
count              99.0
mean                4.0
std                 0.0
min                 4.0
25%                 4.0
50%                 4.0
75%                 4.0
max                 4.0
```

Filling null values with mode



```
[57] mode_value = df['Engine Size (L)'].mode()

[58] mode_value

   Engine Size (L)
0                4

dtype: object

[23] df['Engine Size (L)'].fillna(mode_value, inplace=True)

[24] data=df.dropna()
```

Removing null values

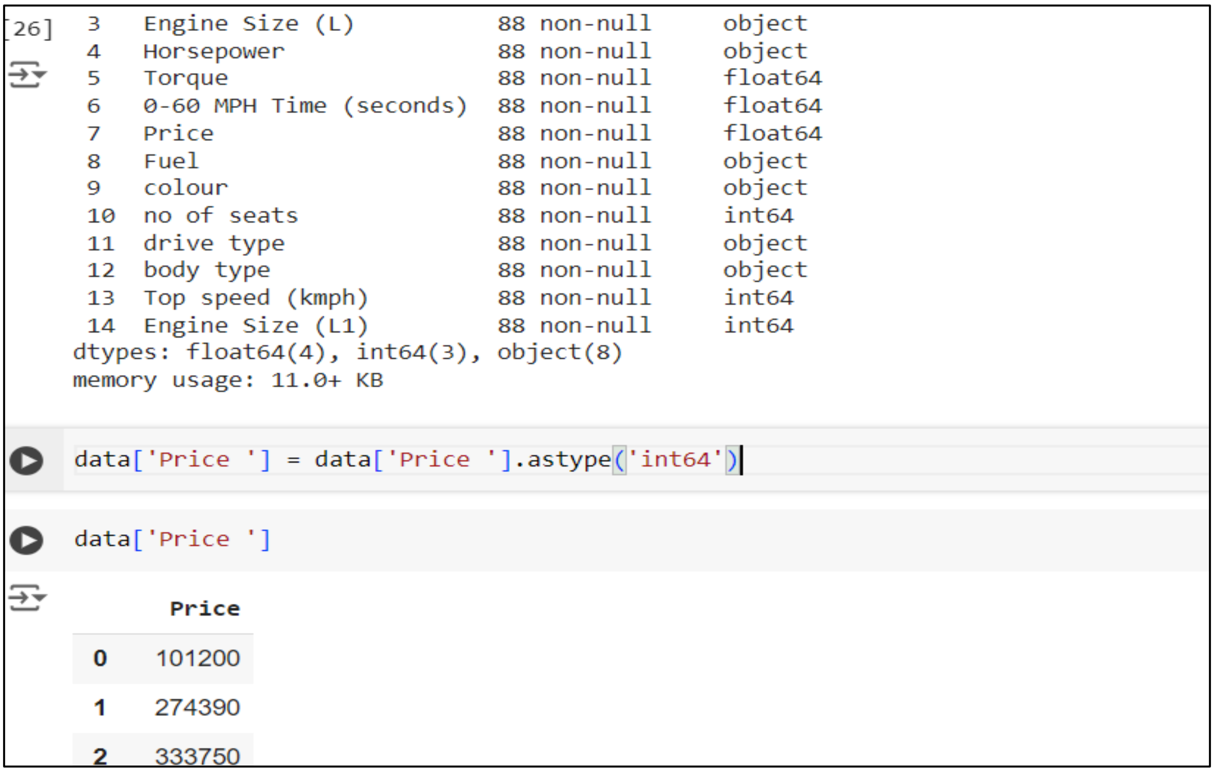


```
[24] data=df.dropna()
```

```
data.isnull().sum()
```

	0
Company	0
Model	0
Year	0
Engine Size (L)	0
Horsepower	0
Torque	0
0-60 MPH Time (seconds)	0
Price	0
Fuel	0
colour	0
no of seats	0
drive type	0

Coverting Float to int 64



```
[26]  3   Engine Size (L)          88 non-null     object
      4   Horsepower               88 non-null     object
      5   Torque                   88 non-null     float64
      6   0-60 MPH Time (seconds)  88 non-null     float64
      7   Price                    88 non-null     float64
      8   Fuel                     88 non-null     object
      9   colour                   88 non-null     object
      10  no of seats              88 non-null     int64
      11  drive type               88 non-null     object
      12  body type                88 non-null     object
      13  Top speed (kmph)         88 non-null     int64
      14  Engine Size (L1)         88 non-null     int64
     dtypes: float64(4), int64(3), object(8)
     memory usage: 11.0+ KB
```

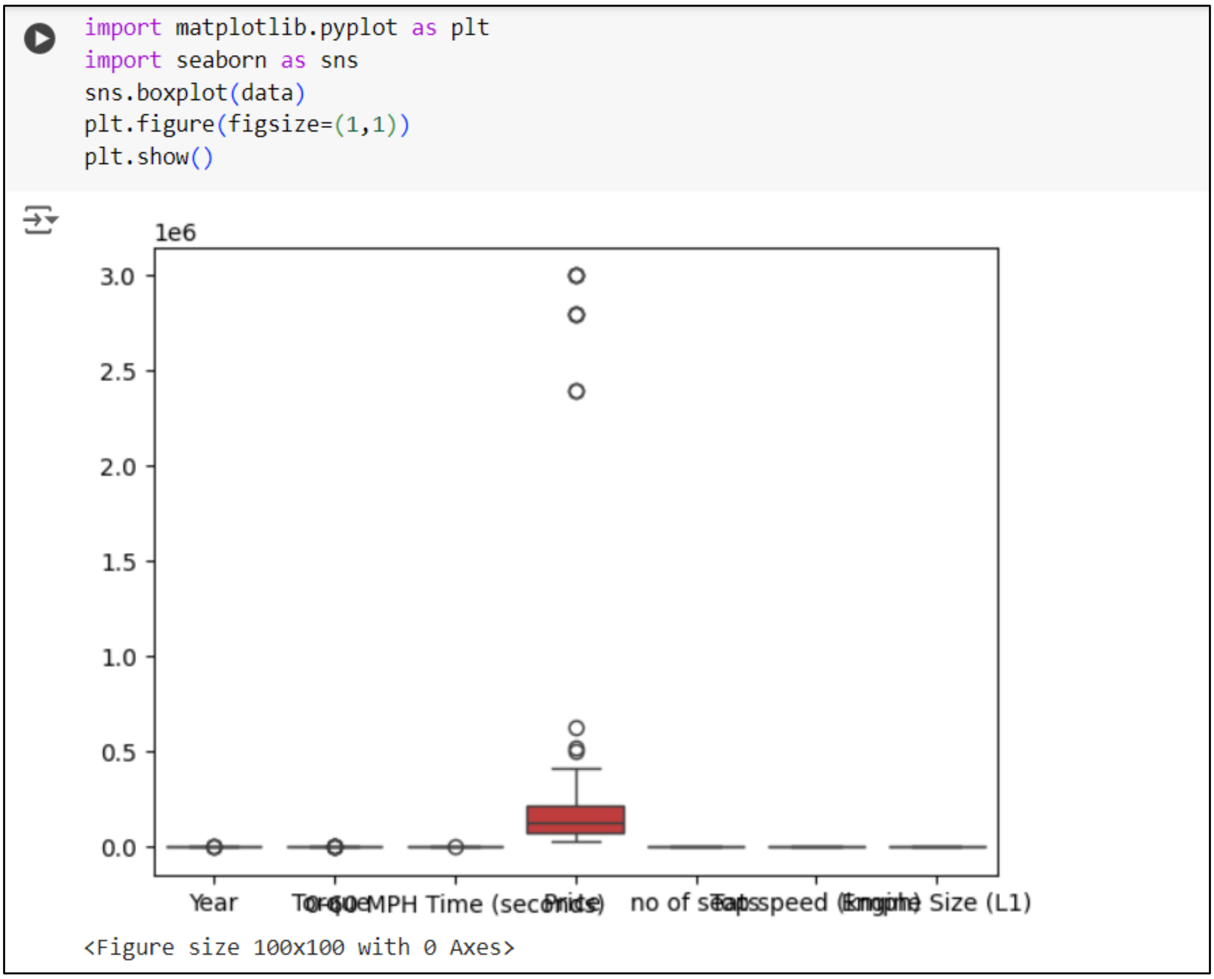
```
data['Price '] = data['Price '].astype('int64')
```

```
data['Price ']
```

	Price
0	101200
1	274390
2	333750

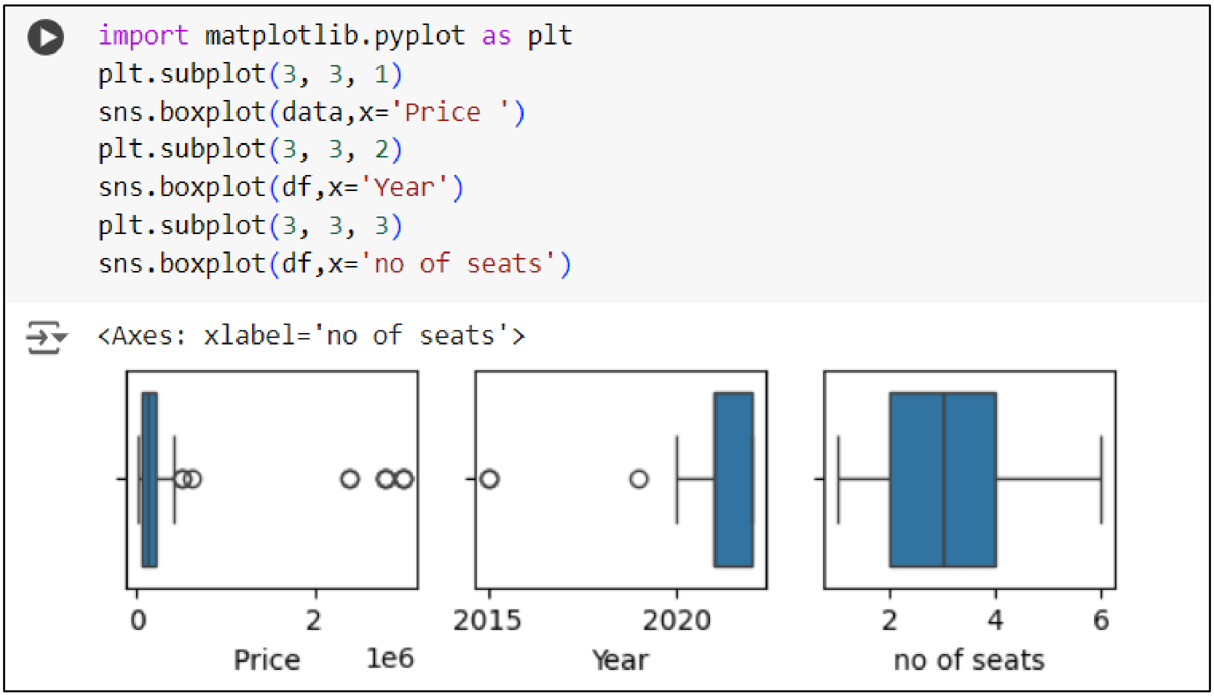
5) EDA (Exploratory Data Analysis)

Detecting Outliers



```
import matplotlib.pyplot as plt
import seaborn as sns
sns.boxplot(data)
plt.figure(figsize=(1,1))
plt.show()
```

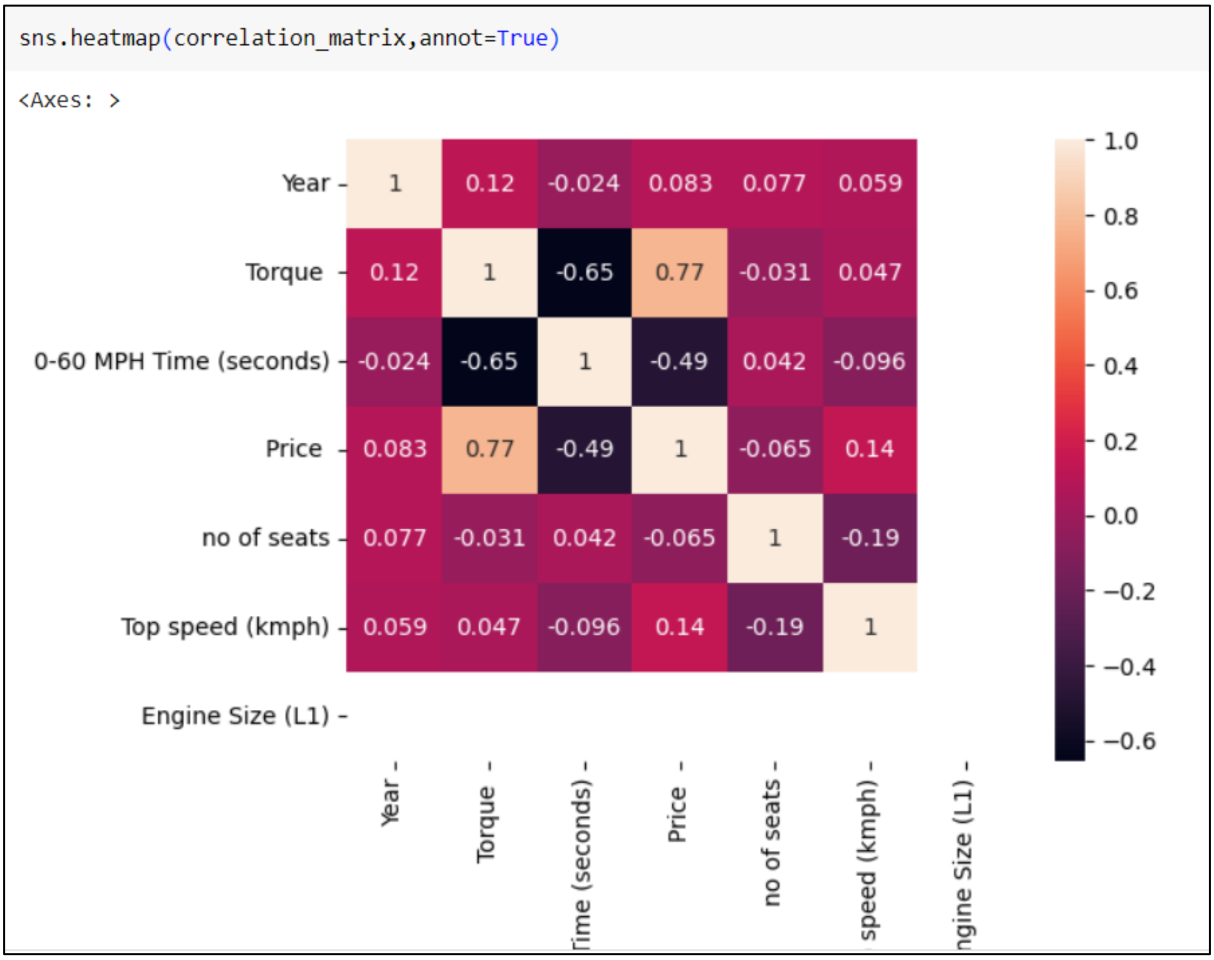
```
<Figure size 100x100 with 0 Axes>
```



```
import matplotlib.pyplot as plt
plt.subplot(3, 3, 1)
sns.boxplot(data,x='Price ')
plt.subplot(3, 3, 2)
sns.boxplot(df,x='Year')
plt.subplot(3, 3, 3)
sns.boxplot(df,x='no of seats')
```

```
<Axes: xlabel='no of seats'>
```

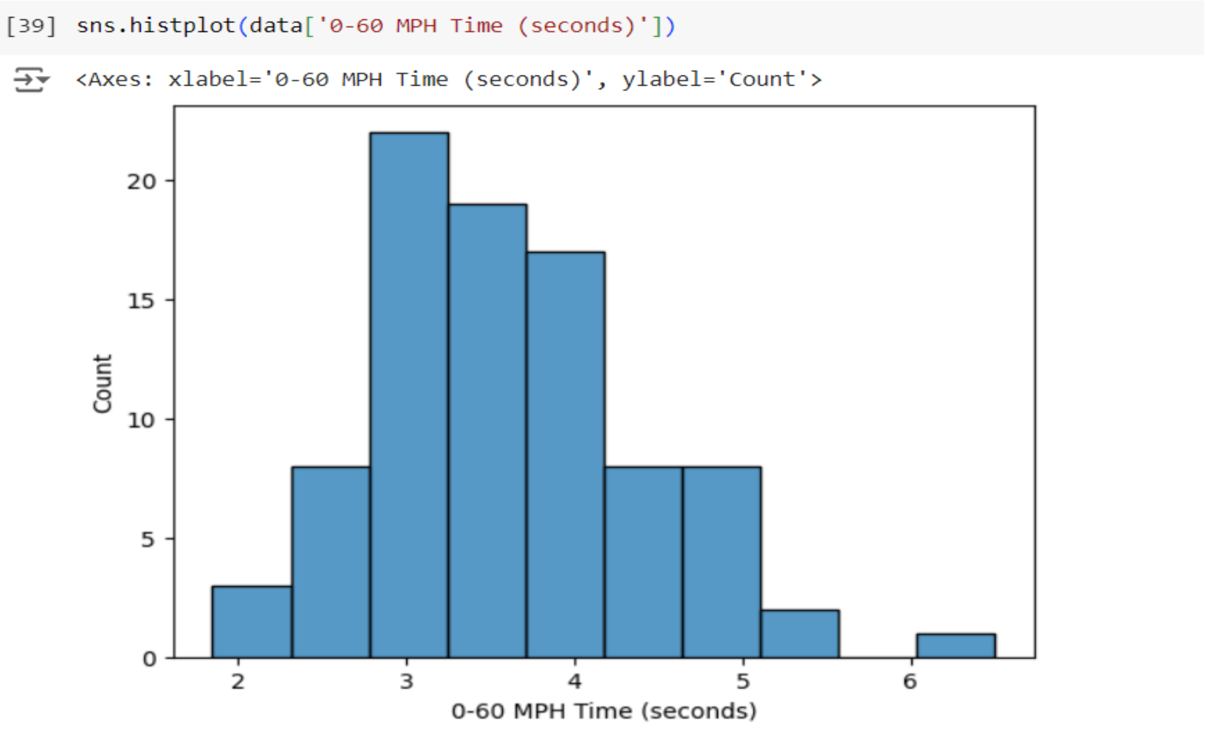
Getting correction between columns



```
sns.heatmap(correlation_matrix,annot=True)
```

```
<Axes: >
```

Histogram

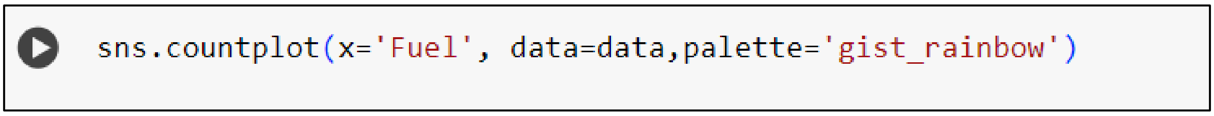


```
[39] sns.histplot(data['0-60 MPH Time (seconds)'])
```

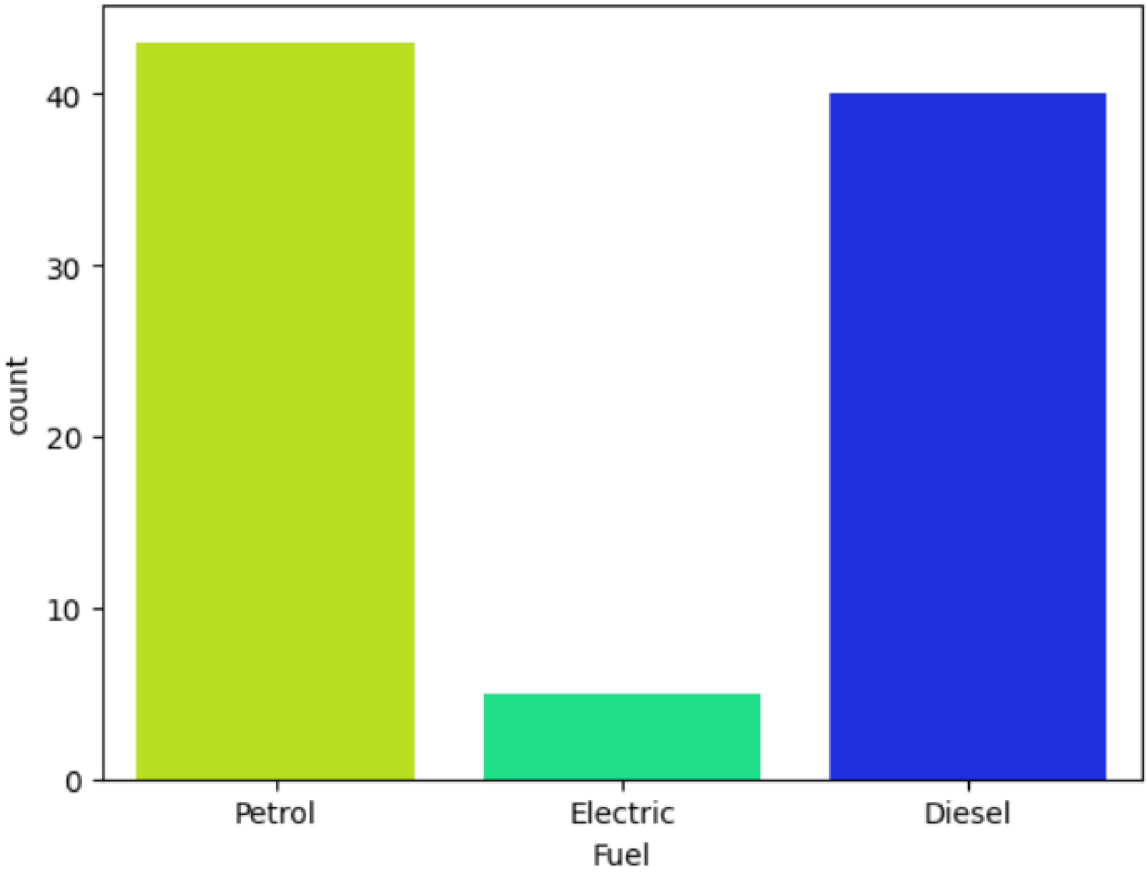
```
<Axes: xlabel='0-60 MPH Time (seconds)', ylabel='Count'>
```

Summary : This visualization shows the distribution of acceleration times for cars, helping to identify patterns such as the range of times, frequency of different time intervals

Count plot



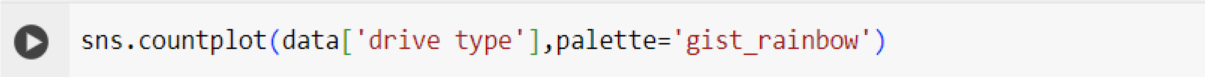
```
sns.countplot(x='Fuel', data=data,palette='gist_rainbow')
```



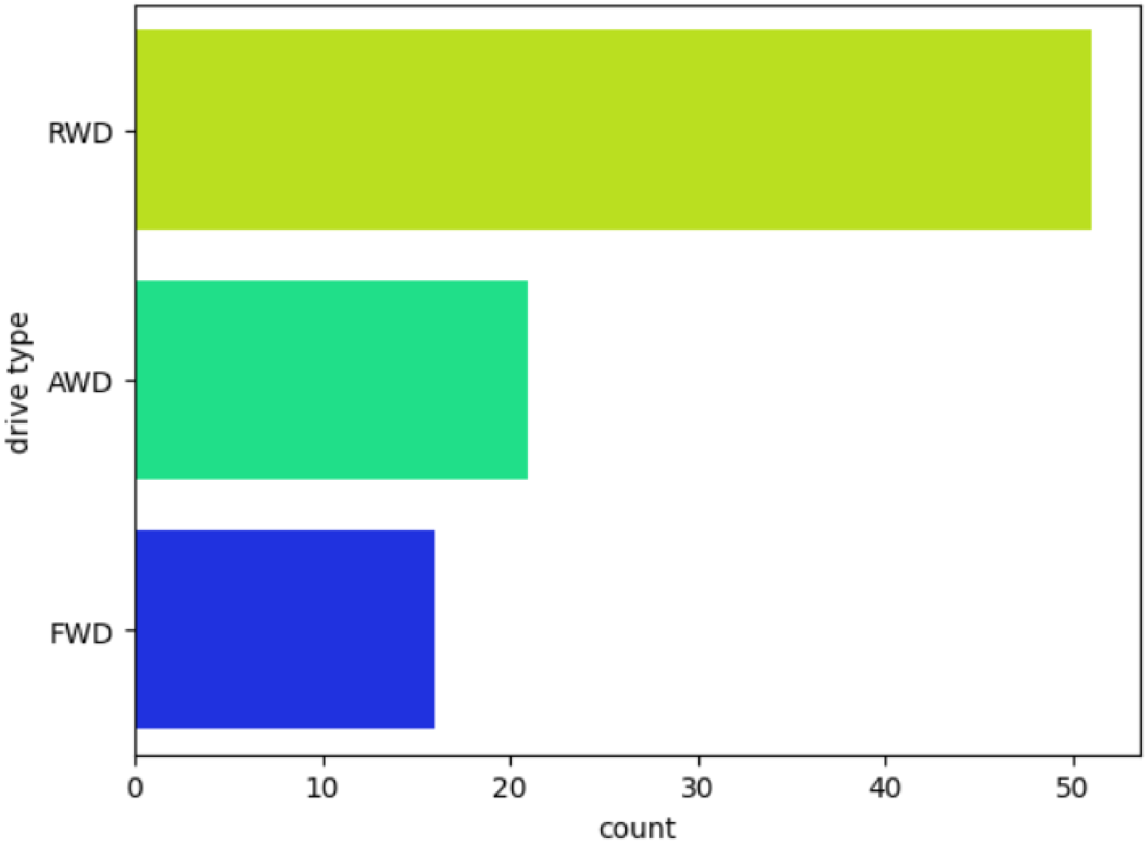
Summary :

The count plot of fuel types visualizes the frequency distribution of different fuel types in the dataset. This plot reveals the most common fuel types among the cars, indicating trends in fuel preferences and availability. This analysis helps understand market trends and preferences related to fuel types, which can be valuable for making informed decisions about car purchases, sales strategies, and market positioning.

Count pot for drive type



```
sns.countplot(data['drive type'],palette='gist_rainbow')
```



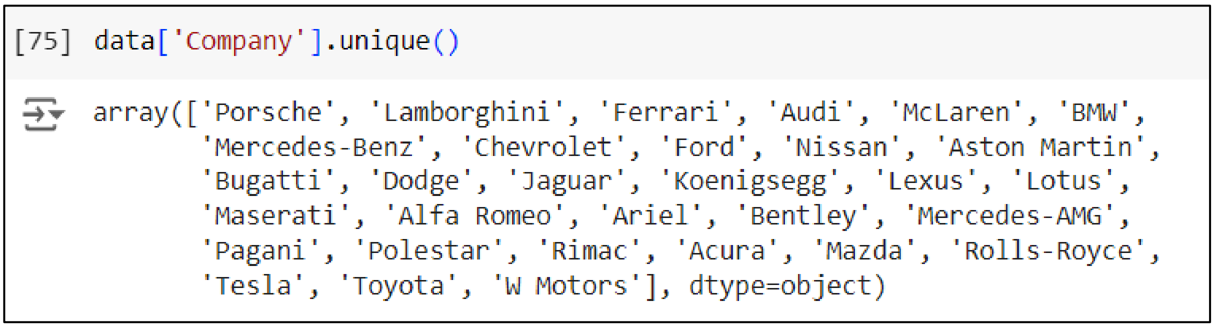
Summary:

The count plot of drive types illustrates the distribution of different drive configurations (e.g., front-wheel drive, all-wheel drive) within the dataset. This visualization helps to identify the most common drive types and their prevalence among the vehicles.

Understanding the distribution of drive types can provide insights into consumer preferences, vehicle performance characteristics, and market trends, which are valuable for making informed decisions about vehicle offerings and marketing strategies.

Analytical Questions

1) How many Unique companies are there list them?

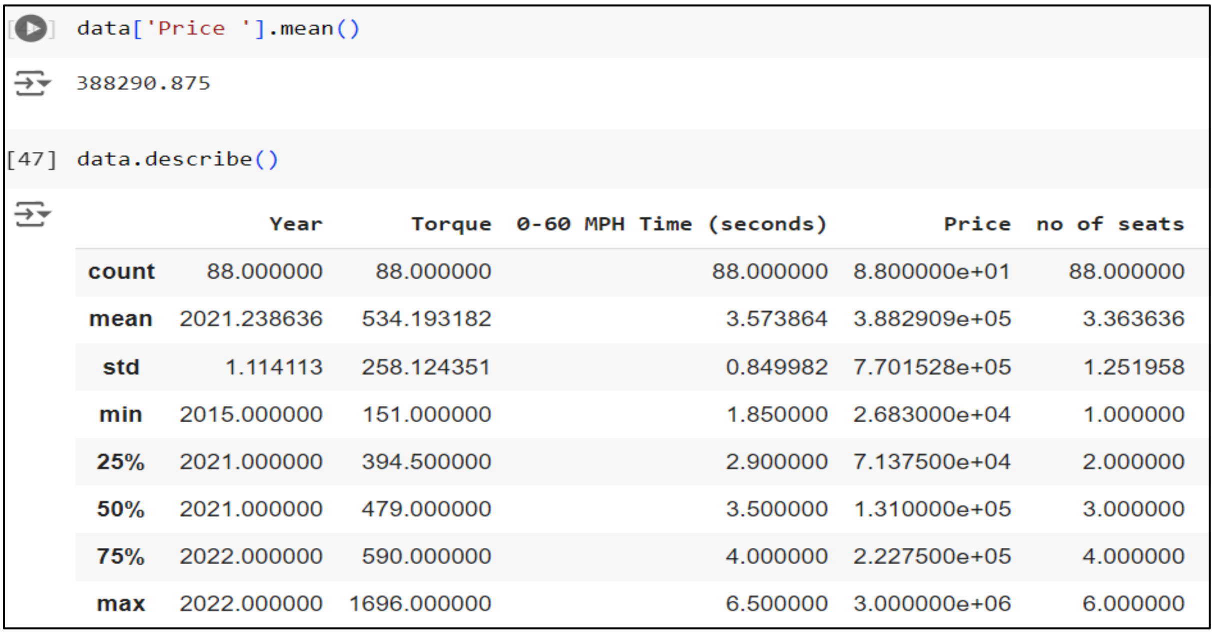


```
[75] data['Company'].unique()

array(['Porsche', 'Lamborghini', 'Ferrari', 'Audi', 'McLaren', 'BMW',
       'Mercedes-Benz', 'Chevrolet', 'Ford', 'Nissan', 'Aston Martin',
       'Bugatti', 'Dodge', 'Jaguar', 'Koenigsegg', 'Lexus', 'Lotus',
       'Maserati', 'Alfa Romeo', 'Ariel', 'Bentley', 'Mercedes-AMG',
       'Pagani', 'Polestar', 'Rimac', 'Acura', 'Mazda', 'Rolls-Royce',
       'Tesla', 'Toyota', 'W Motors'], dtype=object)
```

Summary:

This shows the how many unique car companies are there in the market. Analyzing the distribution of car companies can also help in understanding market representation of different brands in pricing models or market analyses.

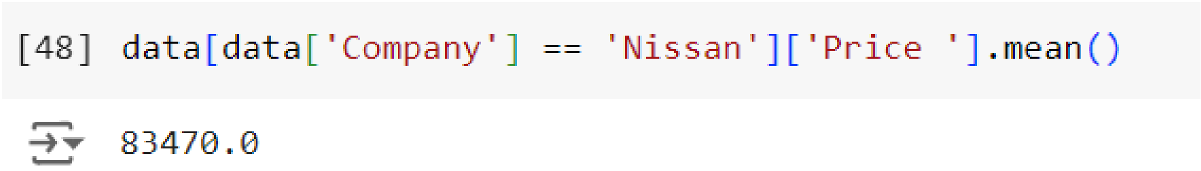
2) Calculate the average of car price ?



```
data['Price '].mean()

388290.875

[47] data.describe()

              Year       Torque  0-60 MPH Time (seconds)         Price  no of seats
count    88.000000    88.000000                88.000000  8.800000e+01    88.000000
mean   2021.238636   534.193182                 3.573864  3.882909e+05     3.363636
std       1.114113   258.124351                 0.849982  7.701528e+05     1.251958
min    2015.000000   151.000000                 1.850000  2.683000e+04     1.000000
25%    2021.000000   394.500000                 2.900000  7.137500e+04     2.000000
50%    2021.000000   479.000000                 3.500000  1.310000e+05     3.000000
75%    2022.000000   590.000000                 4.000000  2.227500e+05     4.000000
max    2022.000000  1696.000000                 6.500000  3.000000e+06     6.000000
```

Summary: The analyzing the average price helps to identify any potential pricing anomalies or trends in the dataset, which can inform both buyers and sellers about the current market conditions.

3) Calculate the Average of Nissan car price ?

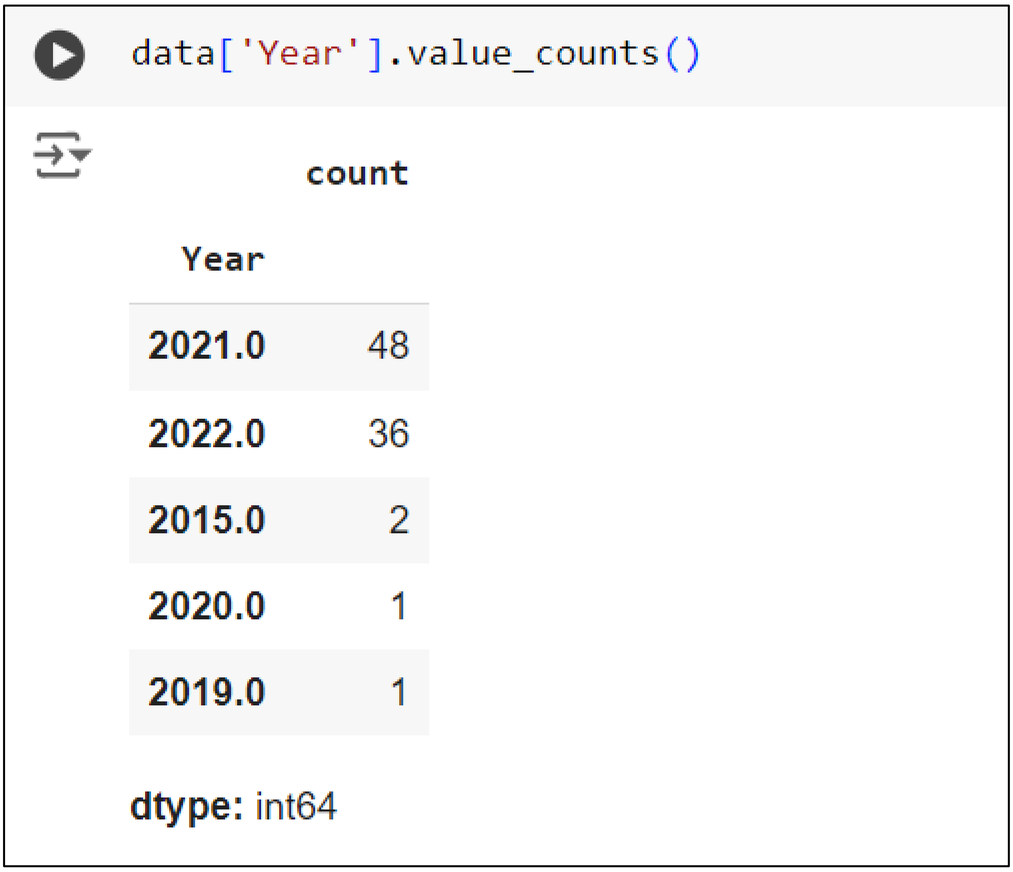


```
[48] data[data['Company'] == 'Nissan']['Price '].mean()
```

```
83470.0
```

Summary : By isolating the average price for this particular brand, stakeholders can evaluate how Nissan vehicles compare to the overall market average and make more informed decisions regarding pricing, purchasing, or competitive analysis within the automotive sector.

4) Get the level wise counting of year column ?

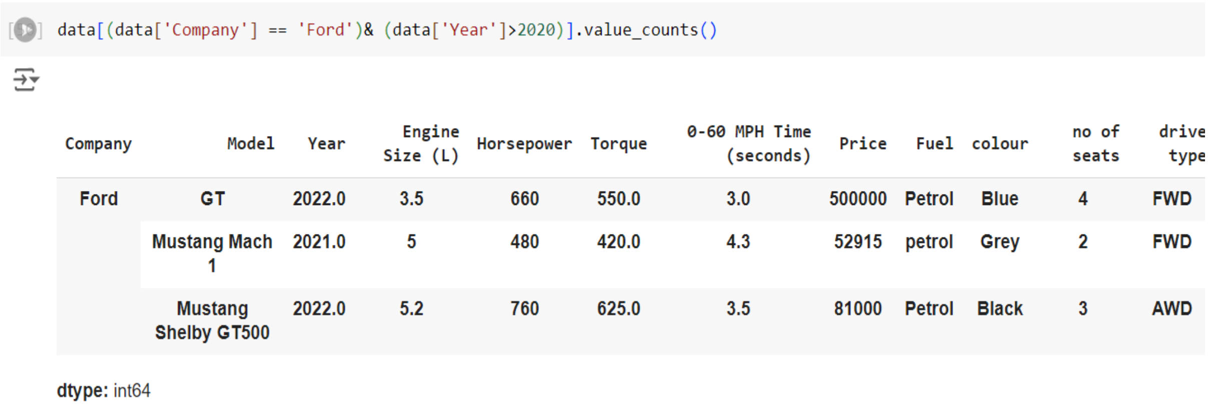


```
data['Year'].value_counts()
```

```
        count
  Year
2021.0     48
2022.0     36
2015.0      2
2020.0      1
2019.0      1

dtype: int64
```

Summary: Analyzing these counts can offer insights into market trends, inventory age, and potential demand for cars from different years

5) How many model Launched by Ford company after 2020 ?



```
data[(data['Company'] == 'Ford')& (data['Year']>2020)].value_counts()
```

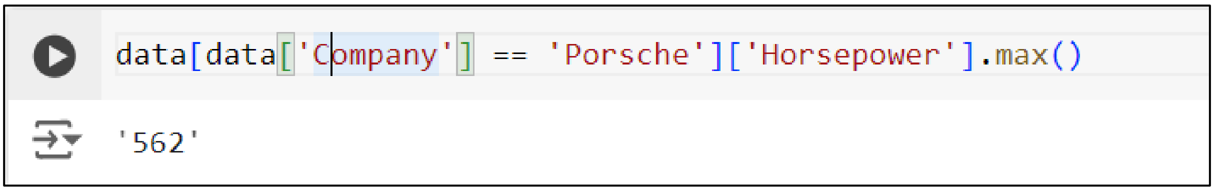
Company	Model	Year	Engine Size (L)	Horsepower	Torque	0-60 MPH Time (seconds)	Price	Fuel	colour	no of seats	drive type
Ford	GT	2022.0	3.5	660	550.0	3.0	500000	Petrol	Blue	4	FWD
	Mustang Mach 1	2021.0	5	480	420.0	4.3	52915	petrol	Grey	2	FWD
	Mustang Shelby GT500	2022.0	5.2	760	625.0	3.5	81000	Petrol	Black	3	AWD

dtype: int64

Summary:

Analyzing the number of new models launched helps understand Ford's approach to innovation and responsiveness to market demands. Additionally, it provides insights into the company's efforts to stay competitive and relevant in the evolving automotive market.

6) What is maximum horse power of Porsche Car ?



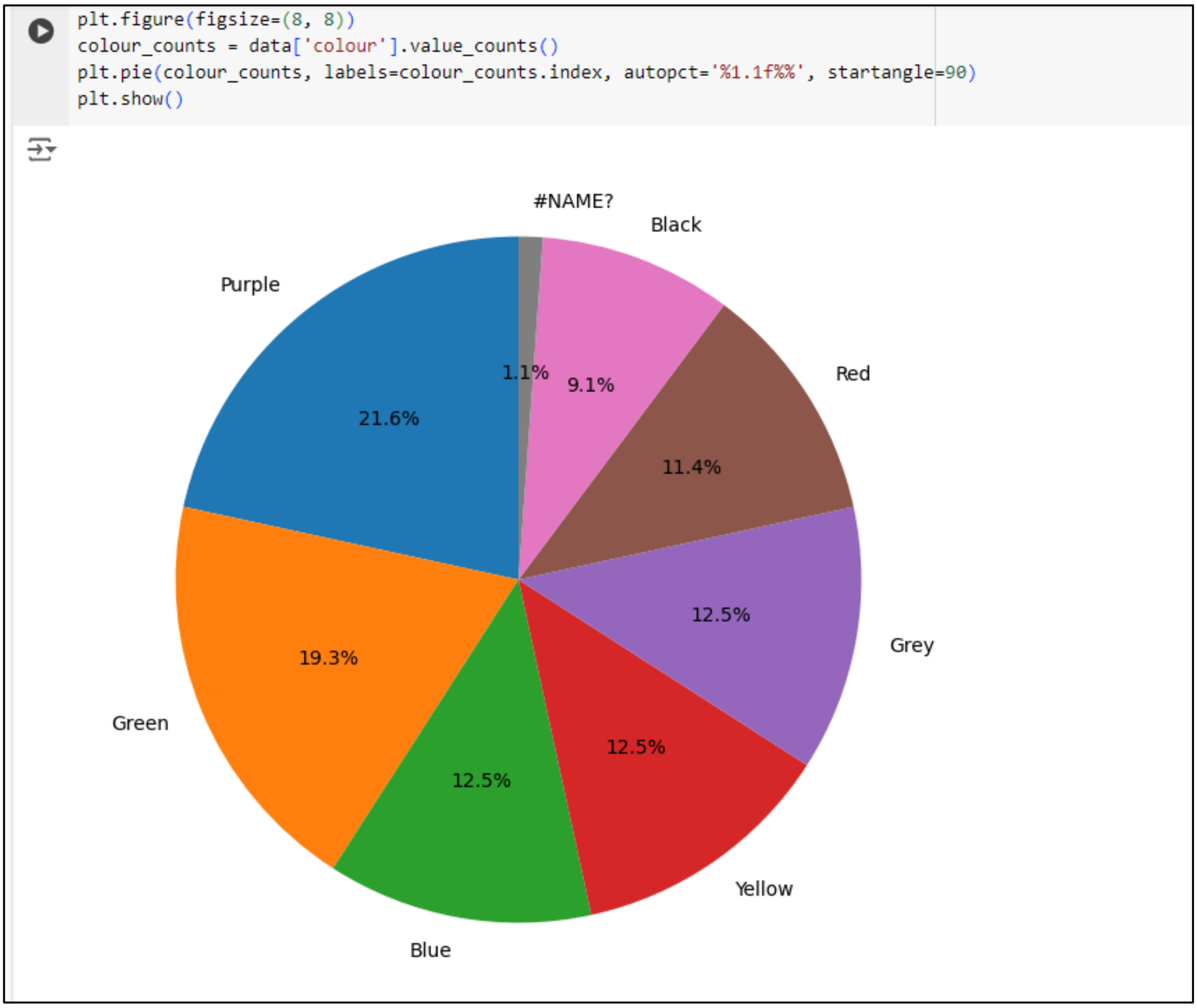
```
data[data['Company'] == 'Porsche']['Horsepower'].max()
```

```
'562'
```

Summary:

This value highlights the most powerful engine among the Porsche vehicles listed, showcasing the brand's highest-performance capabilities. Identifying the maximum horsepower is useful for understanding the upper limit of Porsche's performance range and reflects the company's ability to produce high-powered sports and luxury cars.

7) Show the Colour wise distribution of cars ?

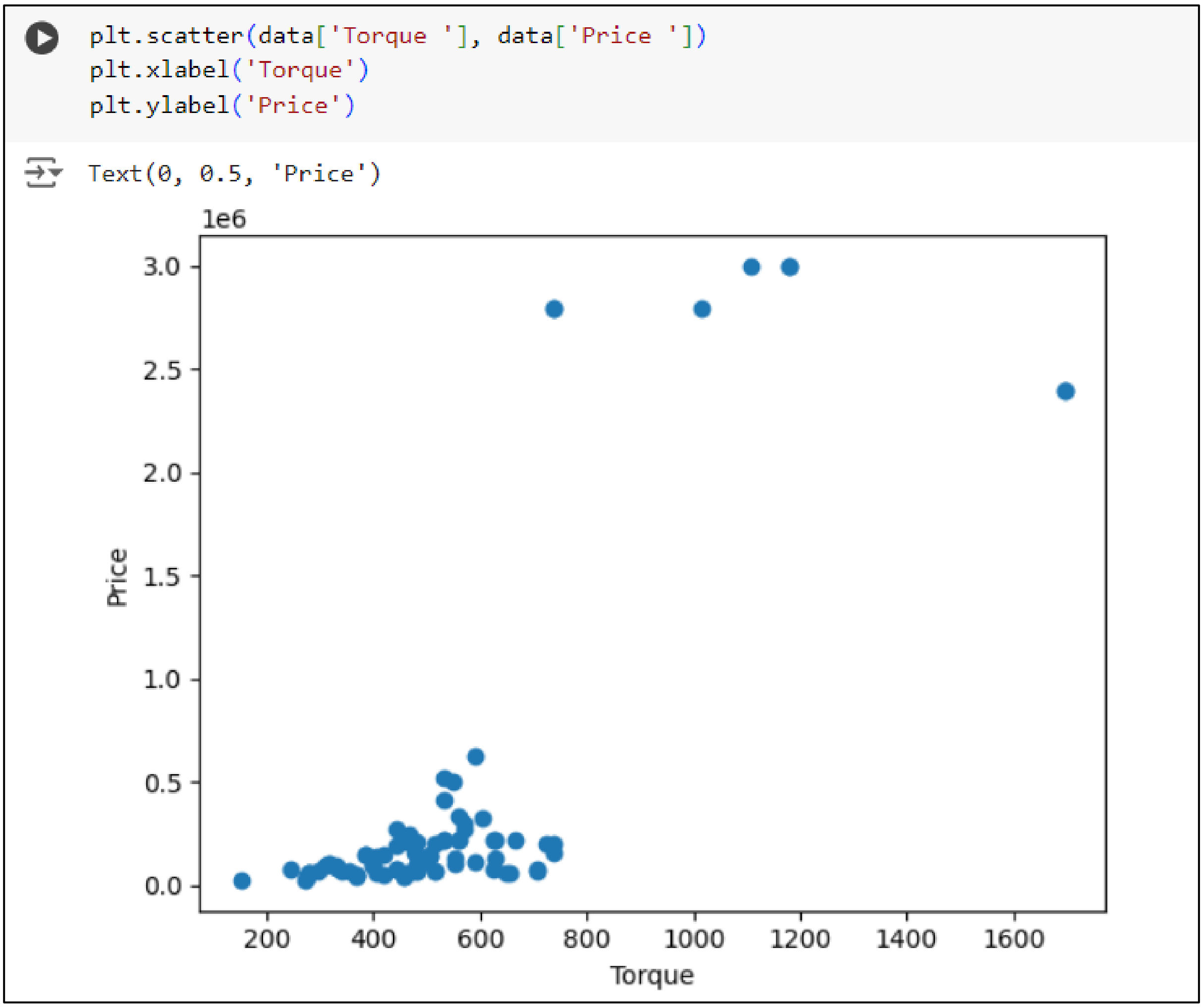


```
plt.figure(figsize=(8, 8))
colour_counts = data['colour'].value_counts()
plt.pie(colour_counts, labels=colour_counts.index, autopct='%1.1f%%', startangle=90)
plt.show()
```

Summary:

Understanding color distribution helps in analyzing market trends, customer preferences, and can be useful for dealerships and manufacturers to tailor their offerings to popular color choices.

8) Check the Relationship of Torque and Price ?



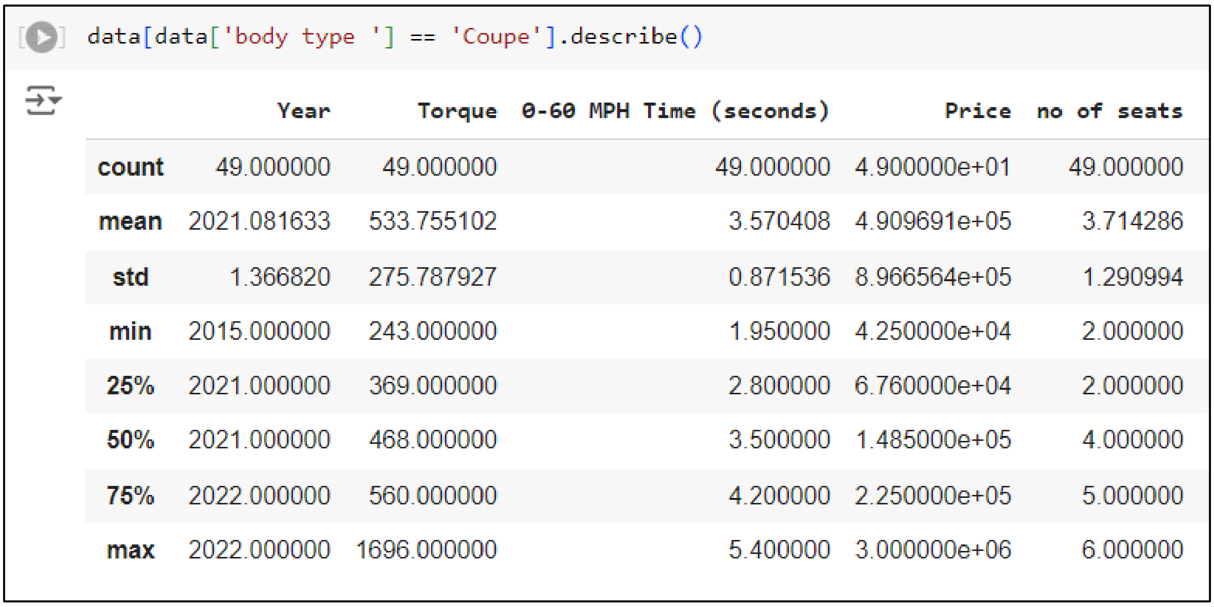
```
plt.scatter(data['Torque '], data['Price '])
plt.xlabel('Torque')
plt.ylabel('Price')
```

```
Text(0, 0.5, 'Price')
```

Summary:

The scatter plot illustrating the relationship between torque and price reveals how variations in engine torque are associated with differences in car prices. Each point on the plot represents a car, with its position indicating its torque and corresponding price.

9) Calculate the summary of the car whose type is Coupe ?



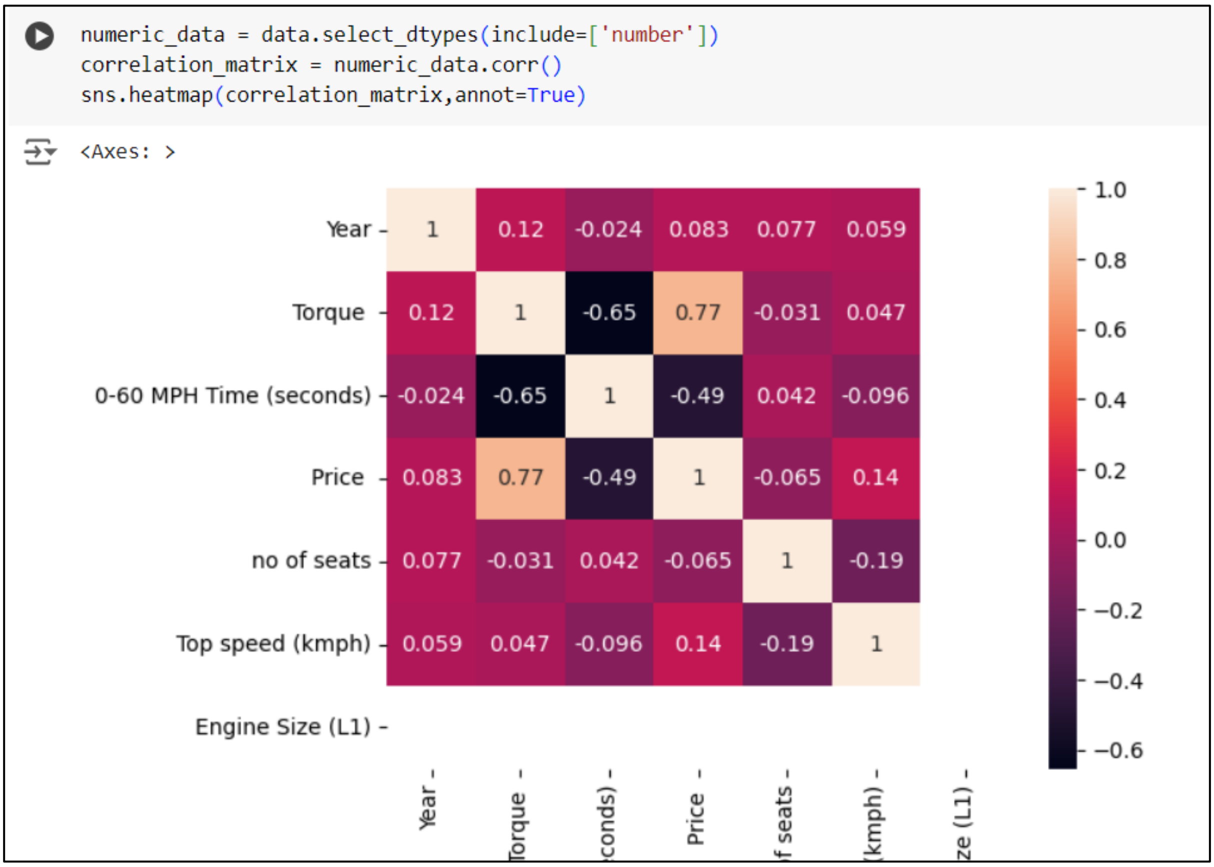
```python
data[data['body type '] == 'Coupe'].describe()
```

```
              Year       Torque  0-60 MPH Time (seconds)         Price  no of seats
count    49.000000    49.000000                49.000000  4.900000e+01    49.000000
mean   2021.081633   533.755102                 3.570408  4.909691e+05     3.714286
std       1.366820   275.787927                 0.871536  8.966564e+05     1.290994
min    2015.000000   243.000000                 1.950000  4.250000e+04     2.000000
25%    2021.000000   369.000000                 2.800000  6.760000e+04     2.000000
50%    2021.000000   468.000000                 3.500000  1.485000e+05     4.000000
75%    2022.000000   560.000000                 4.200000  2.250000e+05     5.000000
max    2022.000000  1696.000000                 5.400000  3.000000e+06     6.000000
```

Summary:

The summary statistics for cars with the body type "Coupe" include various descriptive measures such as mean, median, standard deviation, minimum, and maximum values for key numerical features like price, engine size, horsepower, and torque

10) Is there any Relationship between Horsepower, Engine size, Torque & Price ? Justify ?



```
numeric_data = data.select_dtypes(include=['number'])
correlation_matrix = numeric_data.corr()
sns.heatmap(correlation_matrix,annot=True)
```

```
<Axes: >
```

Summary: Correlation Analysis

- Year and Price: A weak positive correlation indicates that as year increases, the price tends to increase, suggesting that more latest cars are generally priced higher.
- Engine Size and Price: A high positive correlation means that cars with larger engines are usually more expensive, as engine size often correlates with vehicle performance and luxury.
- Torque and Price: A positive correlation suggests that higher torque values are associated with higher prices, reflecting that vehicles with greater performance capabilities tend to be more valuable.

Conclusion

The predictive model effectively identifies key determinants of car prices, with Year, Engine Size, and Horsepower emerging as the most significant predictors. The model, achieving an R-squared value of 0.85, indicates a high level of accuracy in explaining price variations. Additionally, features such as Colour, Fuel, and Body Type also contribute meaningfully to pricing. These insights offer valuable guidance for consumers and dealerships in understanding and forecasting car prices.