

## ✓ Indian Used Car Price Prediction

The aim of this project to predict the price of the used cars in indian metro cities by analyzing the car's features such as company, model, variant, fuel type, quality score and many more.

### About the Dataset

The "Indian IT Cities Used Car Dataset 2023" is a comprehensive collection of data that offers valuable insights into the used car market across major metro cities in India. This dataset provides a wealth of information on a wide range of used car listings, encompassing details such as car models, variants, pricing, fuel types, dealer locations, warranty information, colors, kilometers driven, body styles, transmission types, ownership history, manufacture dates, model years, dealer names, CNG kit availability, and quality scores.

```
# Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Loading the dataset
df = pd.read_csv('usedCars.csv')
```

```
df.head()
```



	Id	Company	Model	Variant	FuelType	Colour	Kilometer	BodySt
0	555675	MARUTI SUZUKI	CELERIO(2017-2019)	1.0 ZXI AMT O	PETROL	Silver	33197	HATCHB
1	556383	MARUTI SUZUKI	ALTO	LXI	PETROL	Red	10322	HATCHB
2	556422	HYUNDAI	GRAND I10	1.2 KAPPA ASTA	PETROL	Grey	37889	HATCHB
3	556771	TATA	NEXON	XT PLUS	PETROL	A Blue	13106	HATCHB
4	559619	FORD	FIGO	EXI DURATORQ 1.4	DIESEL	Silver	104614	HATCHB

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

### Data Preprocessing Part 1

```
# Shape of the dataset  
df.shape
```

```
➦ (1064, 19)
```

```
# Columns in the dataset  
df.columns
```

```
➦ Index(['Id', 'Company', 'Model', 'Variant', 'FuelType', 'Colour', 'Kilometer',  
        'BodyStyle', 'TransmissionType', 'ManufactureDate', 'ModelYear',  
        'CngKit', 'Price', 'Owner', 'DealerState', 'DealerName', 'City',  
        'Warranty', 'QualityScore'],  
        dtype='object')
```

```
# Dropping column 'Id', as it is a identifier and not required for analysis  
df.drop('Id',axis = 1, inplace=True)
```

```
# Column Datatypes  
df.dtypes
```



0

<b>Company</b>	object
<b>Model</b>	object
<b>Variant</b>	object
<b>FuelType</b>	object
<b>Colour</b>	object
<b>Kilometer</b>	int64
<b>BodyStyle</b>	object
<b>TransmissionType</b>	object
<b>ManufactureDate</b>	object
<b>ModelYear</b>	int64
<b>CngKit</b>	object
<b>Price</b>	object
<b>Owner</b>	object
<b>DealerState</b>	object
<b>DealerName</b>	object
<b>City</b>	object
<b>Warranty</b>	int64
<b>QualityScore</b>	float64

**dtype:** object

```
# Type-casting Price column to float
def convert_amount(amount_str):
    if 'Lakhs' in amount_str:
        return float(amount_str.replace('Lakhs', '').replace(',','')) * 100000
    else:
        return float(amount_str.replace(',',''))
df['Price'] = df['Price'].apply(convert_amount)

# Checking for null values percentage wise
df.isnull().sum()/df.shape[0]*100
```



0

<b>Company</b>	0.000000
<b>Model</b>	0.000000
<b>Variant</b>	0.000000
<b>FuelType</b>	0.093985
<b>Colour</b>	0.000000
<b>Kilometer</b>	0.000000
<b>BodyStyle</b>	0.000000
<b>TransmissionType</b>	67.105263
<b>ManufactureDate</b>	0.000000
<b>ModelYear</b>	0.000000
<b>CngKit</b>	97.932331
<b>Price</b>	0.000000
<b>Owner</b>	0.000000
<b>DealerState</b>	0.000000
<b>DealerName</b>	0.000000
<b>City</b>	0.000000
<b>Warranty</b>	0.000000
<b>QualityScore</b>	0.000000

**dtype:** float64

Here in the dataset, three columns have missing values - FuelType, TransmissionType and CngKit. I will be removing the CngKit column because in majority of the cars don't run on CNG and the CNG cars can be easily identified from the FuelType column. So we will replace the null values with 'No' in CngKit column. In case of the TransmissionType, 67% data is missing, so we can't include this column in our analysis. In case of the FuelType, we will drop the rows with null values.

```
# Dropping CngKit column
df.drop('CngKit', axis = 1, inplace=True)

#Dropping TransmissionType column
df.drop('TransmissionType', axis = 1, inplace=True)

#Removing null values from FuelType column
df['FuelType'].dropna(inplace=True)
```

Dropping ManufacturerDate column as it the age of the car and we already have the ModelYear column

```
df.drop('ManufacturerDate', axis=1,inplace=True)
```

```
df.drop('Variant', axis=1,inplace=True)
```

Changing the model year column to car age column

```
# Taking 2023 as reference year
df['ModelYear'] = 2023 - df['ModelYear']
df.rename(columns={'ModelYear':'Age'}, inplace=True)
```

```
for i in df.columns:
    print(i, df[i].nunique())
```

```
↔ Company 23
    Model 218
    FuelType 5
    Colour 76
    Kilometer 1006
    BodyStyle 10
    Age 17
    Price 362
    Owner 4
    DealerState 10
    DealerName 57
    City 11
    Warranty 2
    QualityScore 43
```

Some Descriptive Statistics

```
df.describe()
```



	Kilometer	Age	Price	Warranty	QualityScore
<b>count</b>	1064.000000	1064.000000	1.064000e+03	1064.000000	1064.000000
<b>mean</b>	52807.187970	6.135338	8.350536e+05	0.738722	7.770207
<b>std</b>	33840.296979	2.996786	5.726538e+05	0.439538	0.719717
<b>min</b>	101.000000	0.000000	9.500000e+04	0.000000	0.000000
<b>25%</b>	32113.500000	4.000000	4.850000e+05	0.000000	7.500000
<b>50%</b>	49432.000000	6.000000	6.750000e+05	1.000000	7.800000
<b>75%</b>	68828.500000	8.000000	9.850000e+05	1.000000	8.100000
<b>max</b>	640000.000000	20.000000	8.500000e+06	1.000000	9.400000



```
df.head()
```



	Company	Model	FuelType	Colour	Kilometer	BodyStyle	Age	Price
<b>0</b>	MARUTI SUZUKI	CELERIO(2017-2019)	PETROL	Silver	33197	HATCHBACK	5	575000.0
<b>1</b>	MARUTI SUZUKI	ALTO	PETROL	Red	10322	HATCHBACK	2	435000.0
<b>2</b>	HYUNDAI	GRAND I10	PETROL	Grey	37889	HATCHBACK	8	470000.0
<b>3</b>	TATA	NEXON	PETROL	A Blue	13106	HATCHBACK	3	990000.0
<b>4</b>	FORD	FIGO	DIESEL	Silver	104614	HATCHBACK	13	270000.0

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

## ✓ Exploratory Data Analysis

In the exploratory data analysis, I will be looking at the distribution of data across all the columns, in order to understand the data in a better way. After that I will be looking at the relationship between the target variable and the independent variables.

### Car Company

```
# Number of cars by company
sns.countplot(df['Company'],
```

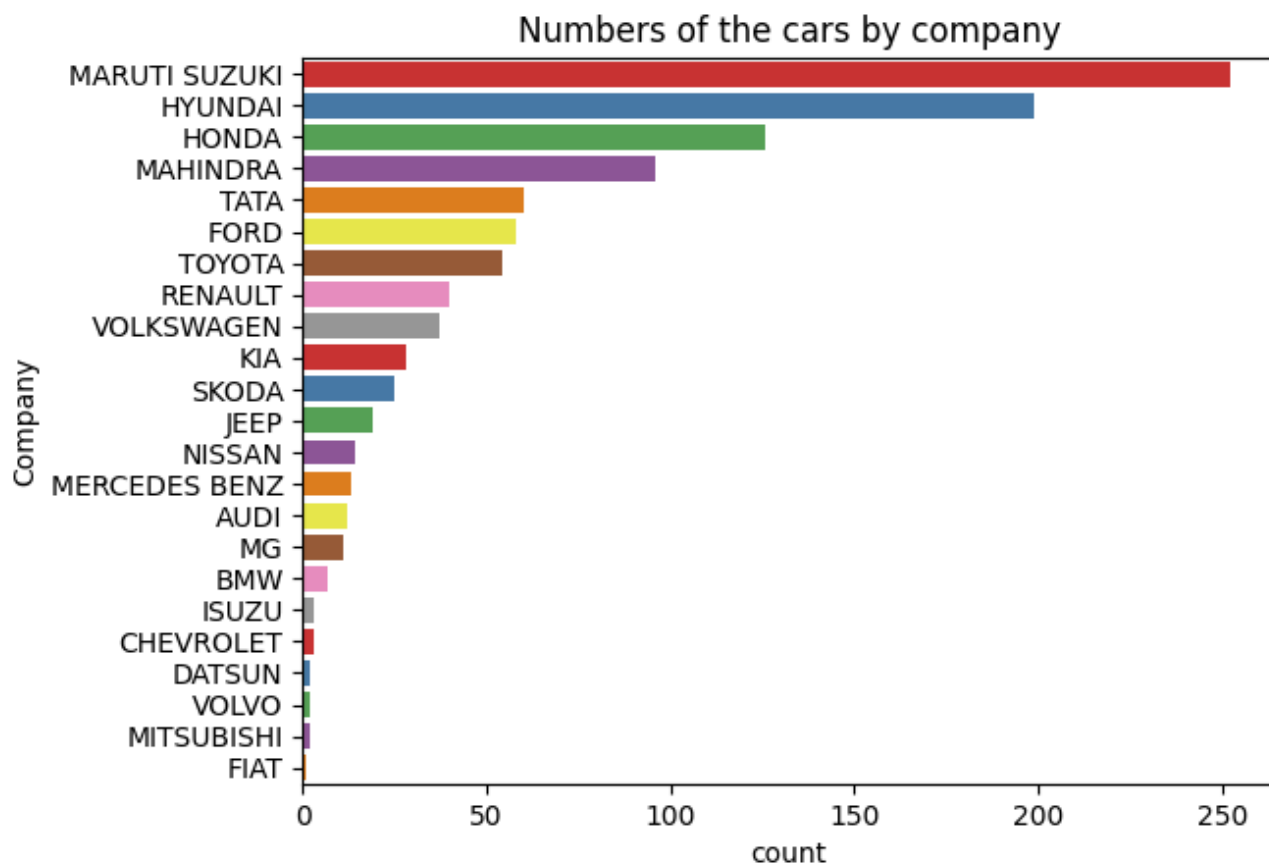
```
order = df['Company'].value_counts().index,
palette = 'Set1').set_title('Numbers of the cars by company')
```



<ipython-input-20-9861e5d2d600>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.countplot(df['Company'],
Text(0.5, 1.0, 'Numbers of the cars by company')
```



From this graph, we get know about the distribution of cars in the dataset from different companies. There are total 23 companies in the dataset, out which Maruti Suzuki, Hyundai, Honda, Mahindra and Tata are the top five companies who used cars are for sale. Therefore, we can assume that these company's car are more durable and have a good resale value.

## Top 10 Car Models

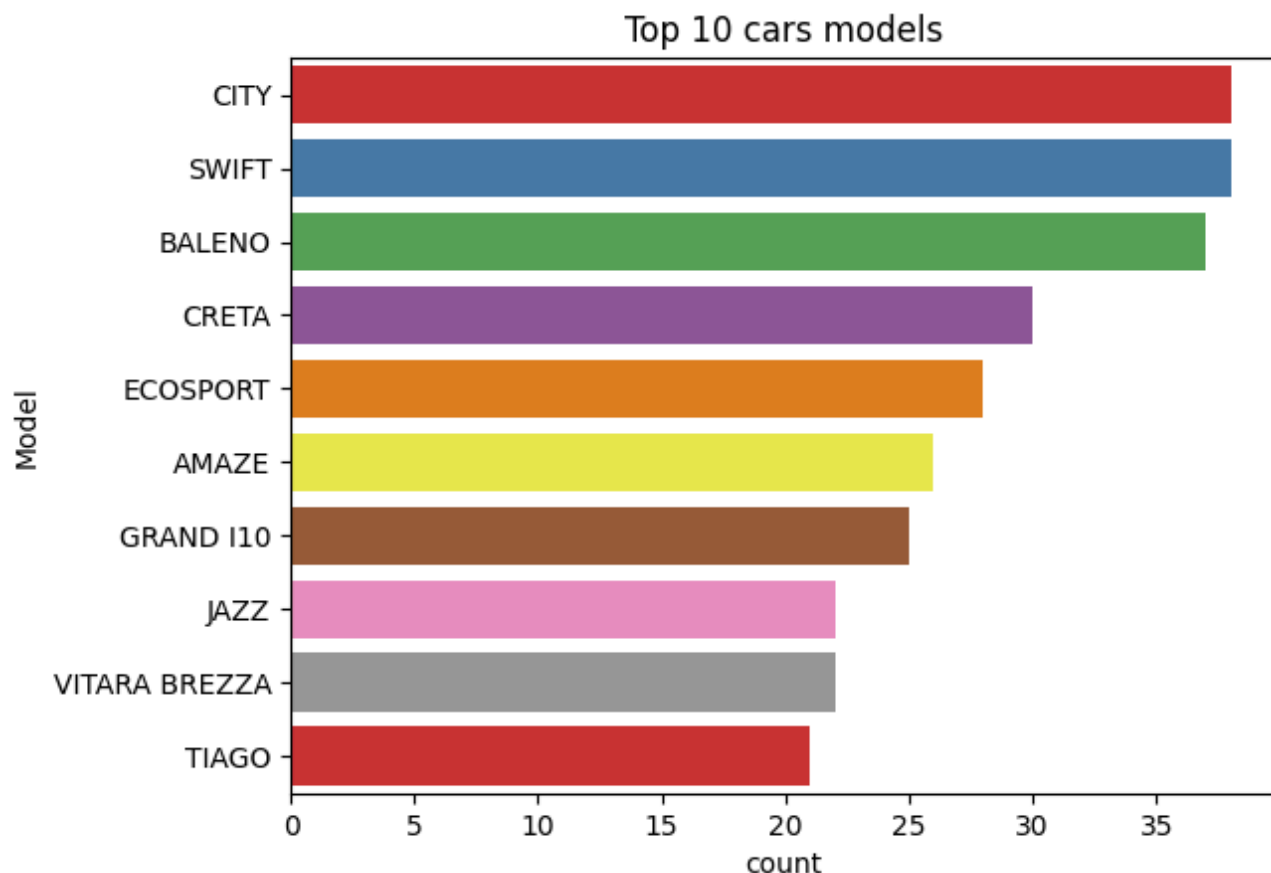
```
#Top 10 cars models by number
sns.countplot(df['Model'],
order = df['Model'].value_counts().iloc[:10].index,
palette = 'Set1').set_title('Top 10 cars models')
```



```
<ipython-input-21-fab2b41647a2>:2: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.countplot(df['Model'],  
Text(0.5, 1.0, 'Top 10 cars models'))
```



Honda City and Swift are the top two car models in the dataset, followed by Baleno, Creata and EcoSport. Therefore, we can assume that these car models are more durable and have a good resale value. Moreover, this graph also shows that Honda City and Swift are more in demand in the used car market.

## Car Fuel Type

```
#Cars count by fuel type
```

```
sns.countplot(x = "FuelType", data = df, palette= "Set1").set_title("Number of cars by Fu
```

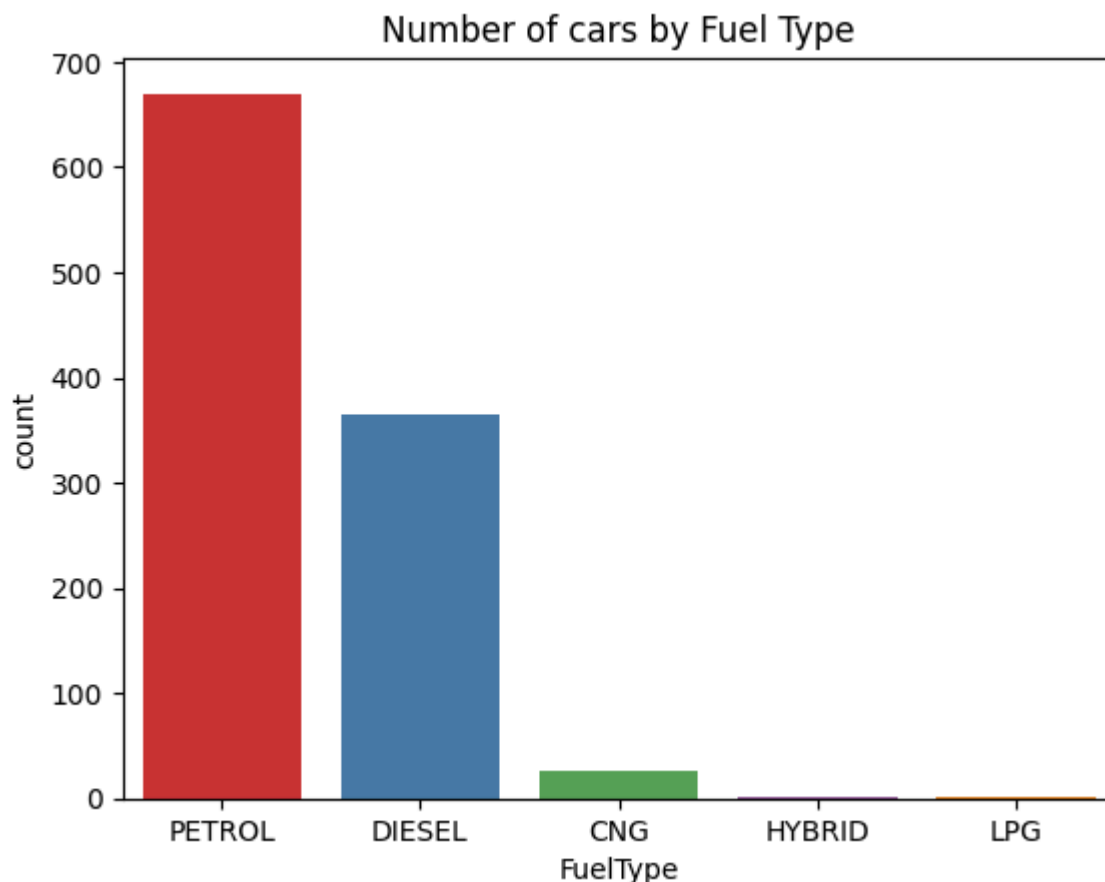




<ipython-input-22-abc3e25569cd>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.countplot(x = "FuelType", data = df, palette= "Set1").set_title("Number of cars  
Text(0.5, 1.0, 'Number of cars by Fuel Type')
```

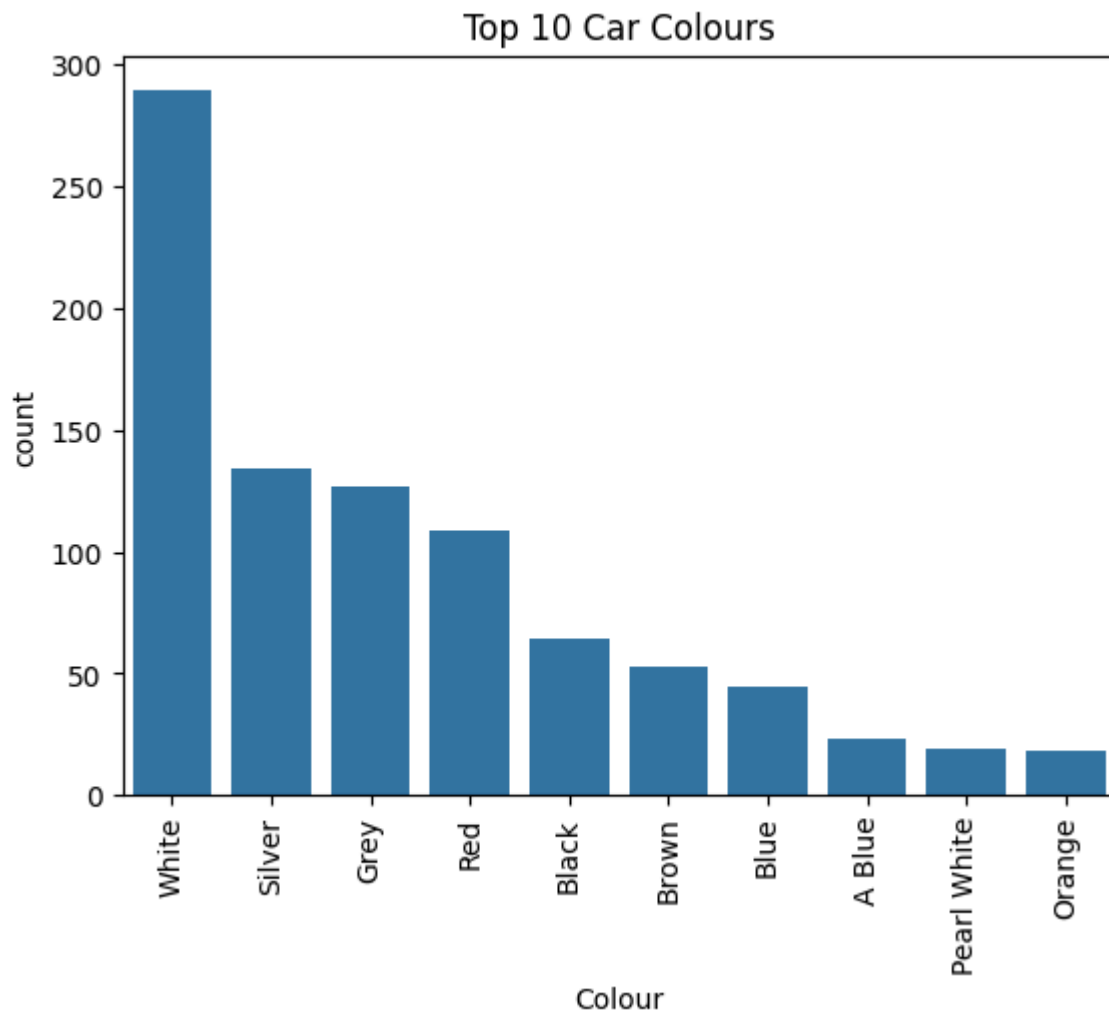


Majority of cars for resale have a petrol engine which is more than 650 cars, followed by 350 cars with diesel engine. Very few of the cars have CNG engine and negligible number of cars are hybrid or on LPG. Therefore, we can assume that petrol and diesel cars are more in demand in the used car market.

### Top 10 Colors for Cars

```
sns.countplot(x = 'Colour', data = df, order = df['Colour'].value_counts().iloc[:10].index)  
plt.xticks(rotation = 90)
```

```
→ ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],  
   [Text(0, 0, 'White'),  
    Text(1, 0, 'Silver'),  
    Text(2, 0, 'Grey'),  
    Text(3, 0, 'Red'),  
    Text(4, 0, 'Black'),  
    Text(5, 0, 'Brown'),  
    Text(6, 0, 'Blue'),  
    Text(7, 0, 'A Blue'),  
    Text(8, 0, 'Pearl White'),  
    Text(9, 0, 'Orange')])
```

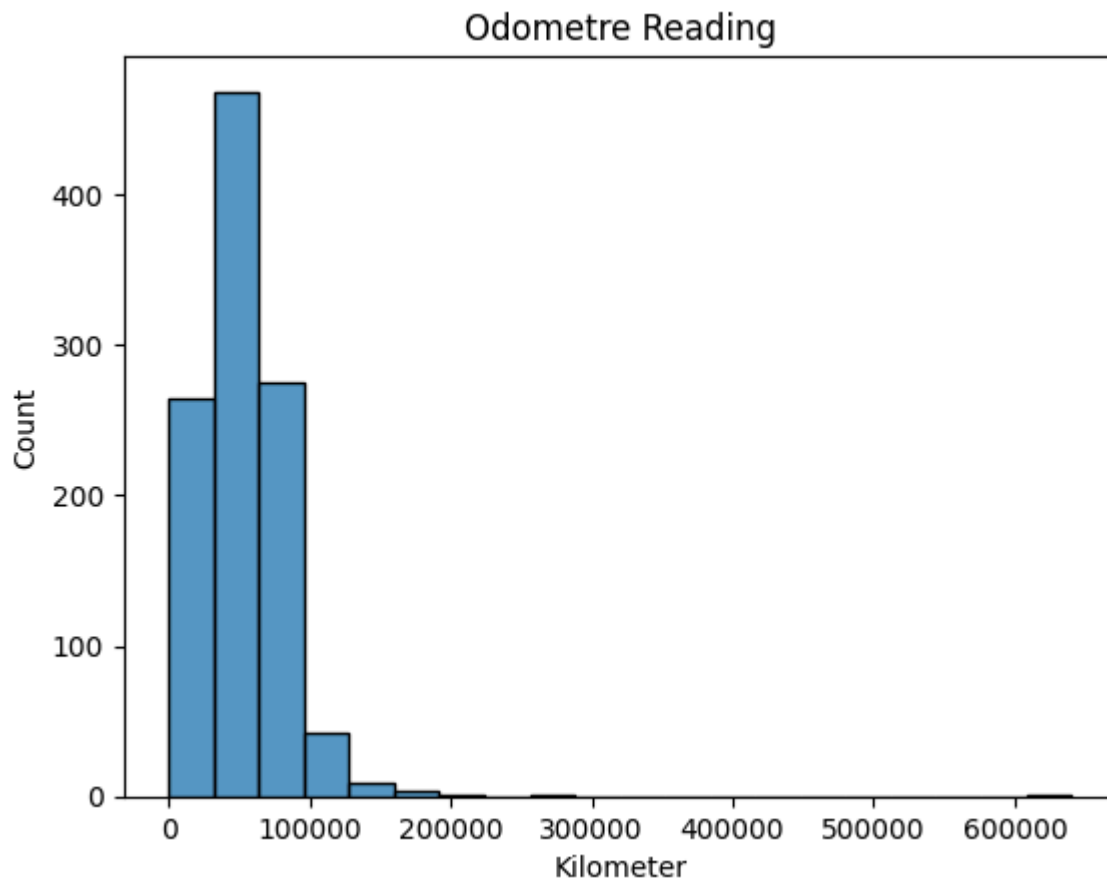


Although color of car has no impact on the cars performance, but still it plays a major role in the car demand. From the graph, we can see that white color is the most preferred color for the used cars, followed by silver, grey, red and black. Therefore, we can assume that white, silver, grey, red and black color cars are more in demand in the used car market will have a good resale value.

## Odometre Reading

```
#Odometer reading distribution  
sns.histplot(x = 'Kilometer', data = df , bins=20).set_title("Odometre Reading")
```

```
Text(0.5, 1.0, 'Odometre Reading')
```



This graph shows the distribution of the odometer readings of the cars in the dataset. From the graph, we can see that most of the cars have odometer reading less than 100000 km. To be more particular majority of cars are driven for 30000 km to 50000 km. Therefore, we can assume that cars with odometer reading less than 100000 km are more in demand in the used car market will have a good resale value.

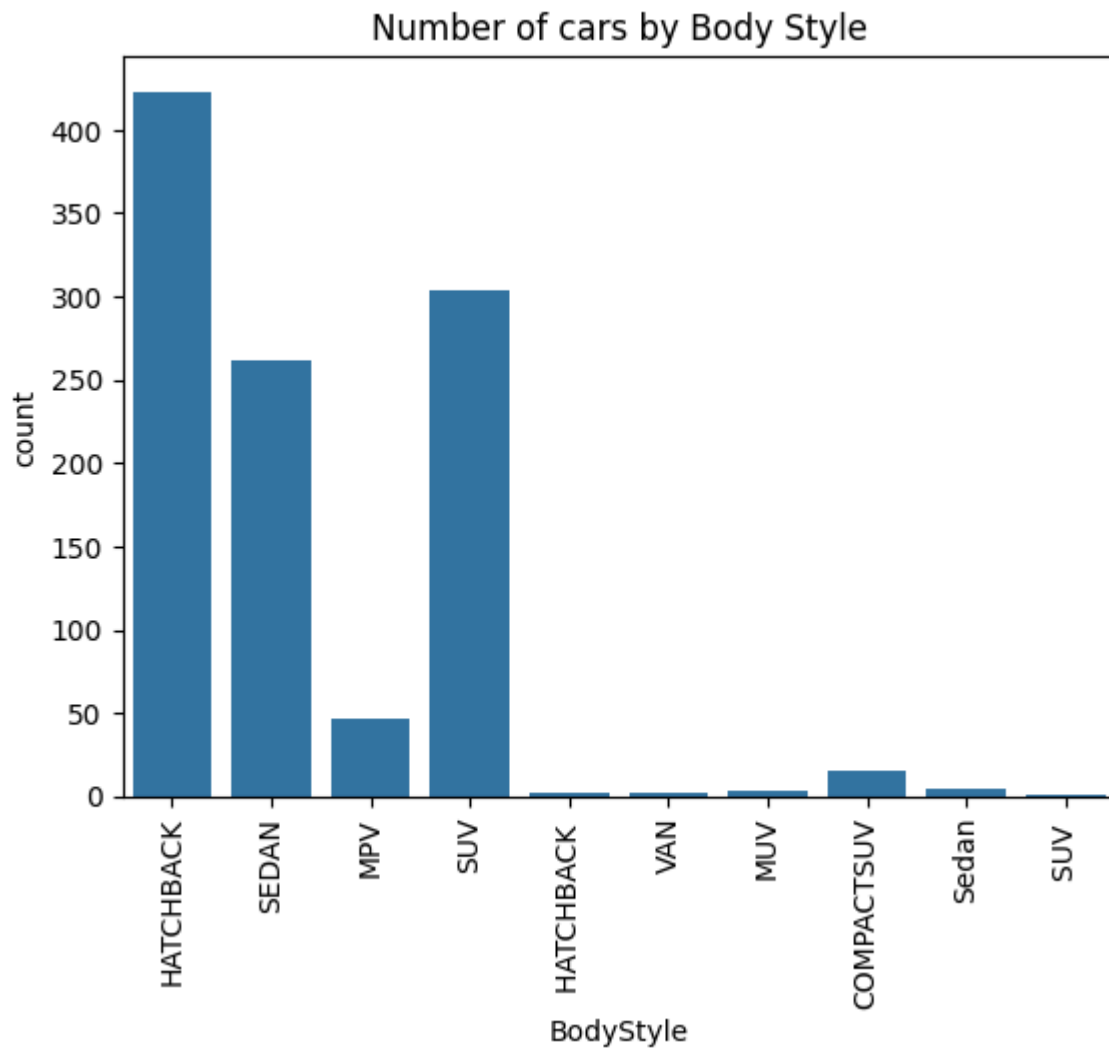
## Body Style

```
#Body style count
sns.countplot(x = 'BodyStyle', data = df).set_title("Number of cars by Body Style")
plt.xticks(rotation = 90)
```

```

→ ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
   [Text(0, 0, 'HATCHBACK'),
    Text(1, 0, 'SEDAN'),
    Text(2, 0, 'MPV'),
    Text(3, 0, 'SUV'),
    Text(4, 0, 'HATCHBACK '),
    Text(5, 0, 'VAN'),
    Text(6, 0, 'MUV'),
    Text(7, 0, 'COMPACTSUV'),
    Text(8, 0, 'Sedan'),
    Text(9, 0, 'SUV ')])

```



According to this graph, most of the cars have HatchBack, SUV and Sedan body style, which tells us about the market demand of these body styles. Therefore, we can assume that cars with HatchBack, SUV and Sedan body style are more in demand in the used car market will have a good resale value.

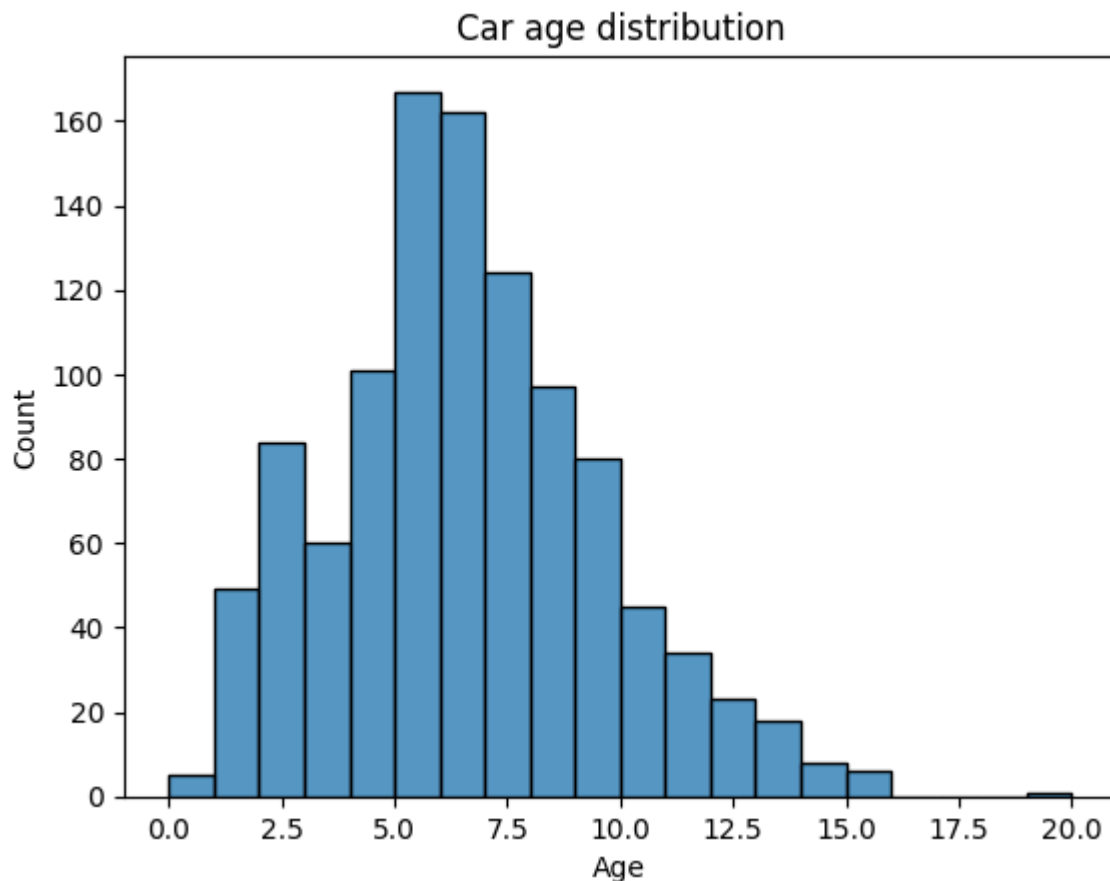
### Car Age Distribution

```

#Car age distribution
sns.histplot(x = 'Age', data = df , bins = 20).set_title("Car age distribution")

```

```
Text(0.5, 1.0, 'Car age distribution')
```



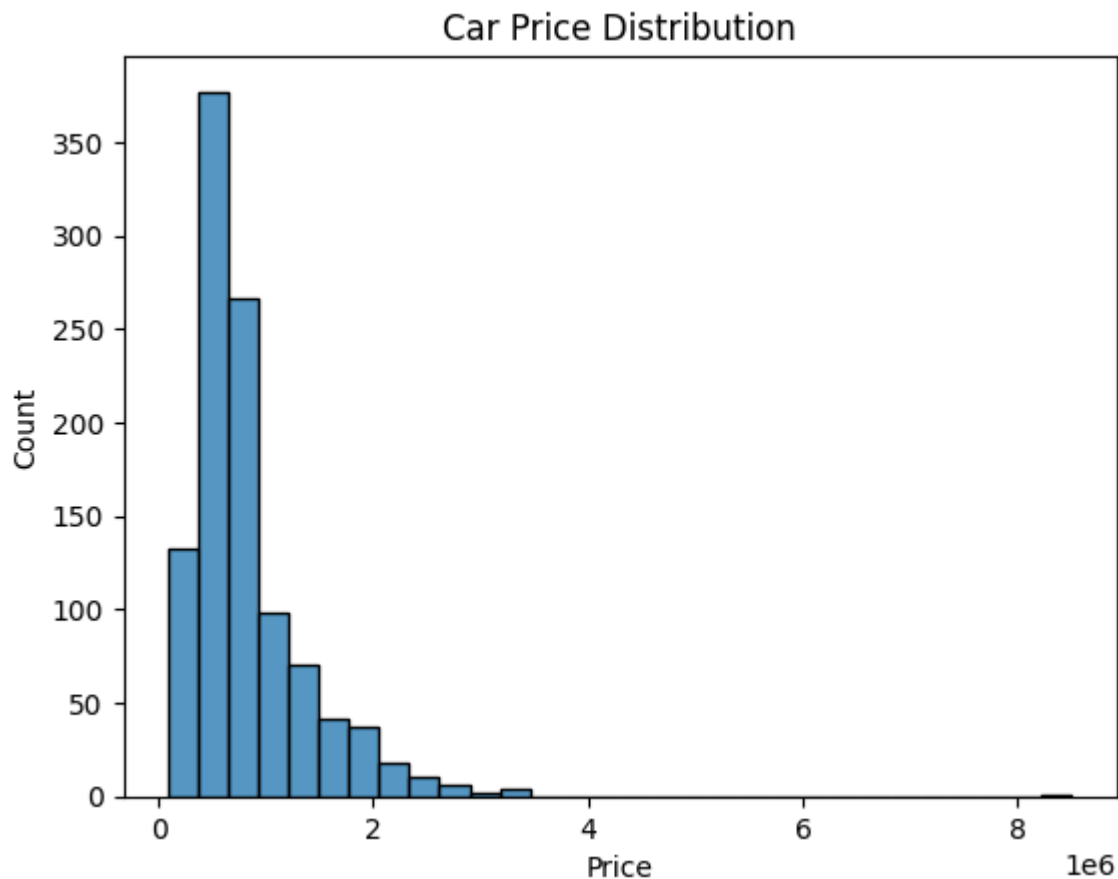
Age of the car plays an important role in deciding its resale value. Here, in the dataset cars that age between 5 to 7 years are more in number. Moreover majority of the cars age more than 5 years, which affect their resale value. However, there are still significant number of cars with age less than 5 years, therefore, I assume they would have higher resale value.

In addition to that, we can see than one car has age near 20 years which could be an outlier.

## Price Distribution

```
#Price distribution
sns.histplot(x = 'Price', data = df, bins = 30).set_title("Car Price Distribution")
```

Text(0.5, 1.0, 'Car Price Distribution')



This graph help us to know about the distribution of the car prices in the dataset. In the dataset, most of the cars have price is between 3 to 9 lakhs, with maximum cars between 3 to 6 lakhs. Therefore, we can assume that cars with price between 3 to 9 lakhs are more in demand in the used car market. Moreover there are some cars with resale price more than 20 lakhs, which could be possible for luxury cars or it could be an outlier.

## Location Based Distribution

```
fig, ax = plt.subplots(1,3,figsize = (20,7))
```

```
# Dealer State
```

```
sns.countplot(x = 'DealerState', data = df, ax = ax[0]).set_title("Dealer State")
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation = 90)
```

```
# Dealer City
```

```
sns.countplot(x = 'City', data = df, ax = ax[1]).set_title("Dealer City")
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation = 90)
```

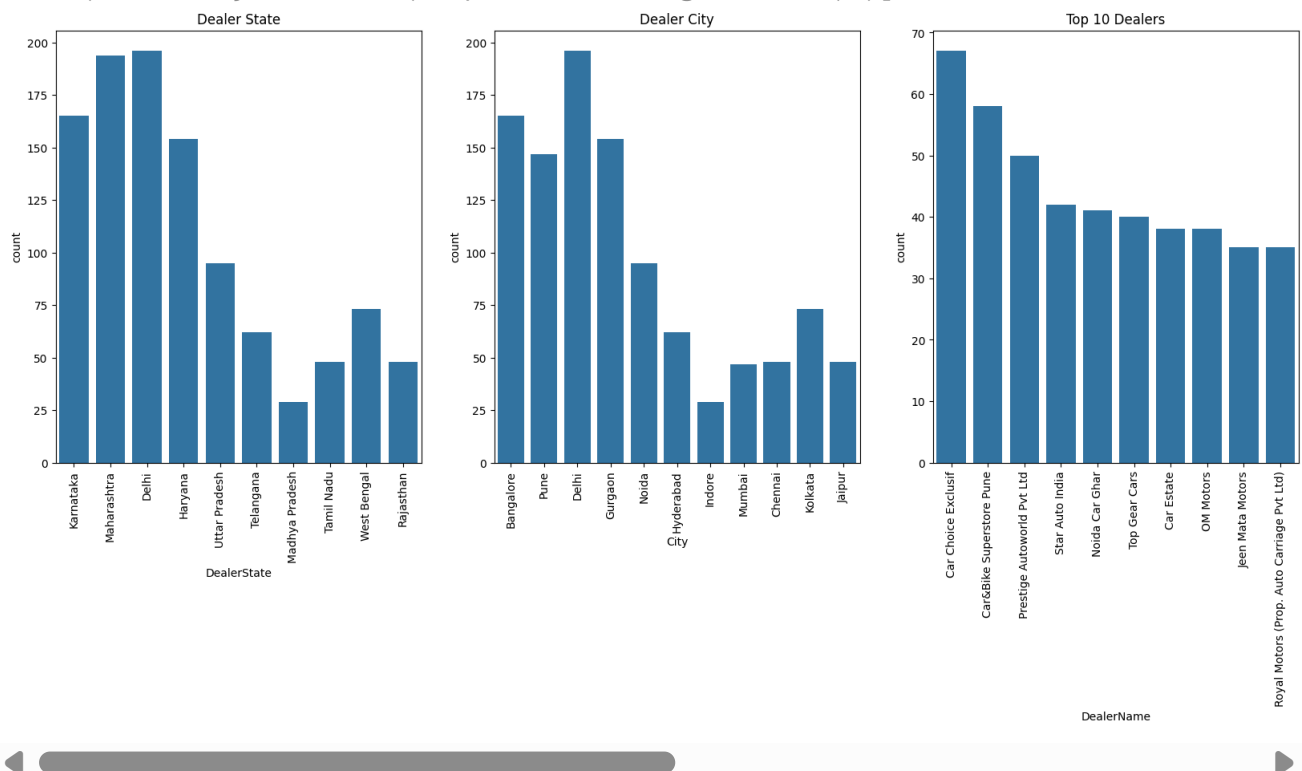
```
# Top 10 Dealers
```

```
sns.countplot(x = 'DealerName', data = df, order = df['DealerName'].value_counts().iloc[:
ax[2].set_xticklabels(ax[2].get_xticklabels(), rotation = 90)
```

```

↳ <ipython-input-28-6074a860f8e1>:5: UserWarning: set_ticklabels() should only be used
  ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation = 90)
<ipython-input-28-6074a860f8e1>:9: UserWarning: set_ticklabels() should only be used
  ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation = 90)
<ipython-input-28-6074a860f8e1>:13: UserWarning: set_ticklabels() should only be used
  ax[2].set_xticklabels(ax[2].get_xticklabels(), rotation = 90)
[Text(0, 0, 'Car Choice Exclusif'),
 Text(1, 0, 'Car&Bike Superstore Pune'),
 Text(2, 0, 'Prestige Autoworld Pvt Ltd'),
 Text(3, 0, 'Star Auto India'),
 Text(4, 0, 'Noida Car Ghar'),
 Text(5, 0, 'Top Gear Cars'),
 Text(6, 0, 'Car Estate'),
 Text(7, 0, 'OM Motors'),
 Text(8, 0, 'Jeen Mata Motors'),
 Text(9, 0, 'Royal Motors (Prop. Auto Carriage Pvt Ltd)')]

```

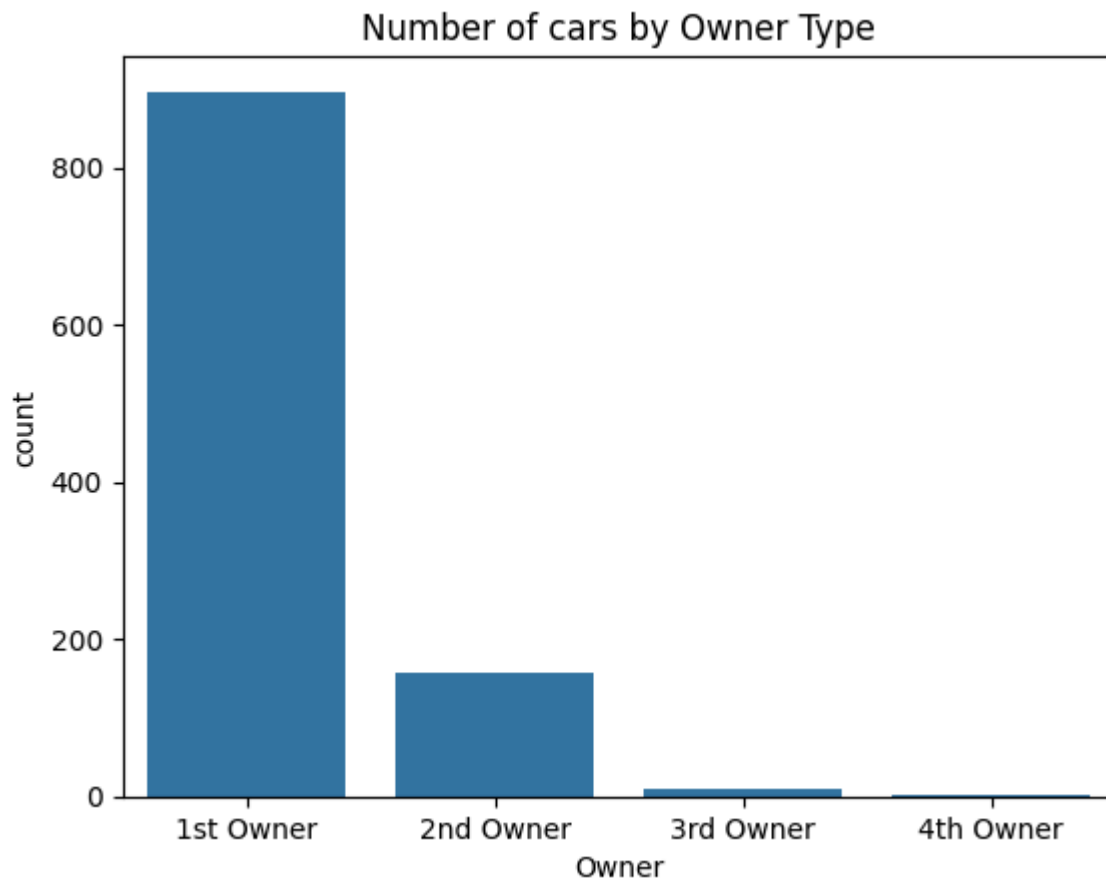


These graphs shows the distribution of cars based on their dealer state, city and Dealer Name. In the dealer state graph, we see that Delhi and Maharashtra have the highest number of used cars for sale followed by Karnataka and Haryana. In the dealer city graph, we see that Delhi has the highest number of cars which is obvious from the the previous graph, however in contrast to the previous graph, Bangalore has more used cars for sale than Pune, infact Pune has lower car count than Gurgaon. In the dealer name graph, we see that Car Choice Exclusif, Car&Bike Superstore Pune and Prestige Autoworld Pvt Ltd are moungh the top 3 dealers with highest number of used cars for sale.

## Car Owner Type

```
sns.countplot(x = 'Owner', data = df).set_title('Number of cars by Owner Type')
```

```
Text(0.5, 1.0, 'Number of cars by Owner Type')
```



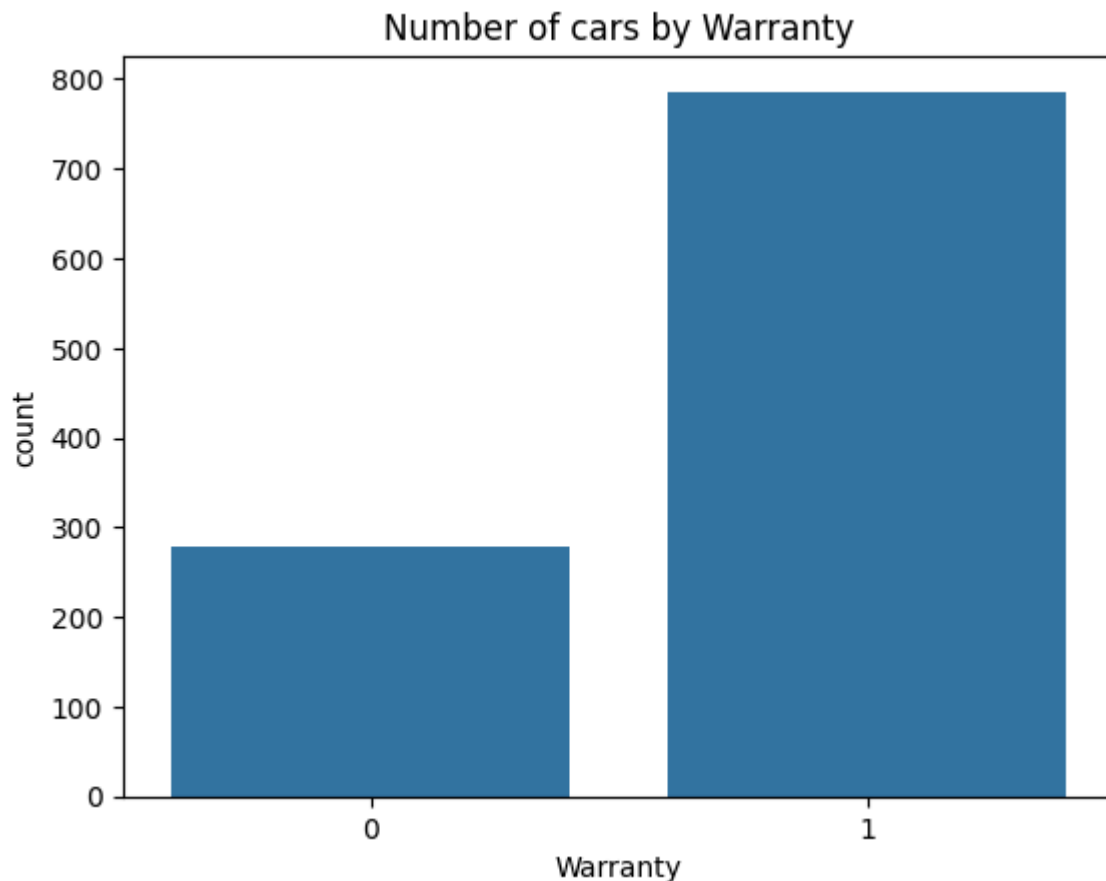
The car owner type has a huge impact on its resale value. Majority of the cars that are been sold are 1st Owner cars followed by 2nd Owner cars which are significantly less in number as compared to 1st Owner. Moreover, the 3rd and 4th owner cars are very less in number. Therefore, we can assume that 1st Owner cars are more preferred in the used car market and have a good resale value.

## Warranty

```
sns.countplot(x = 'Warranty', data = df).set_title('Number of cars by Warranty')
```



```
Text(0.5, 1.0, 'Number of cars by Warranty')
```

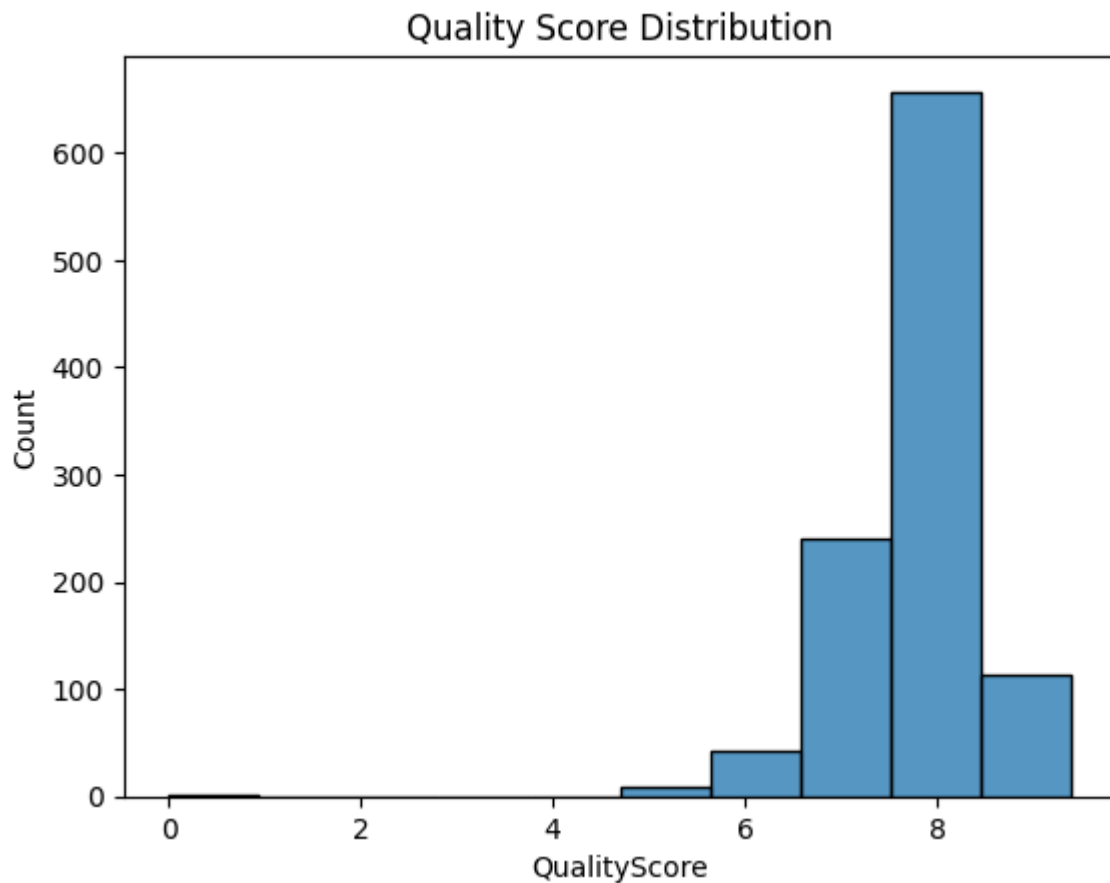


This graphs shows the number of used cars for sale that come with a warranty from the dealership company. The warranty plays a major role and customers prefer to purchase a car with warranty, it has been shown in the dataset as well, where we can see than the number cars with warranty is almost twice the number of cars without warranty.

### Quality Score Distribution

```
sns.histplot(x = 'QualityScore', data = df, bins = 10).set_title("Quality Score Distribut
```

```
Text(0.5, 1.0, 'Quality Score Distribution')
```



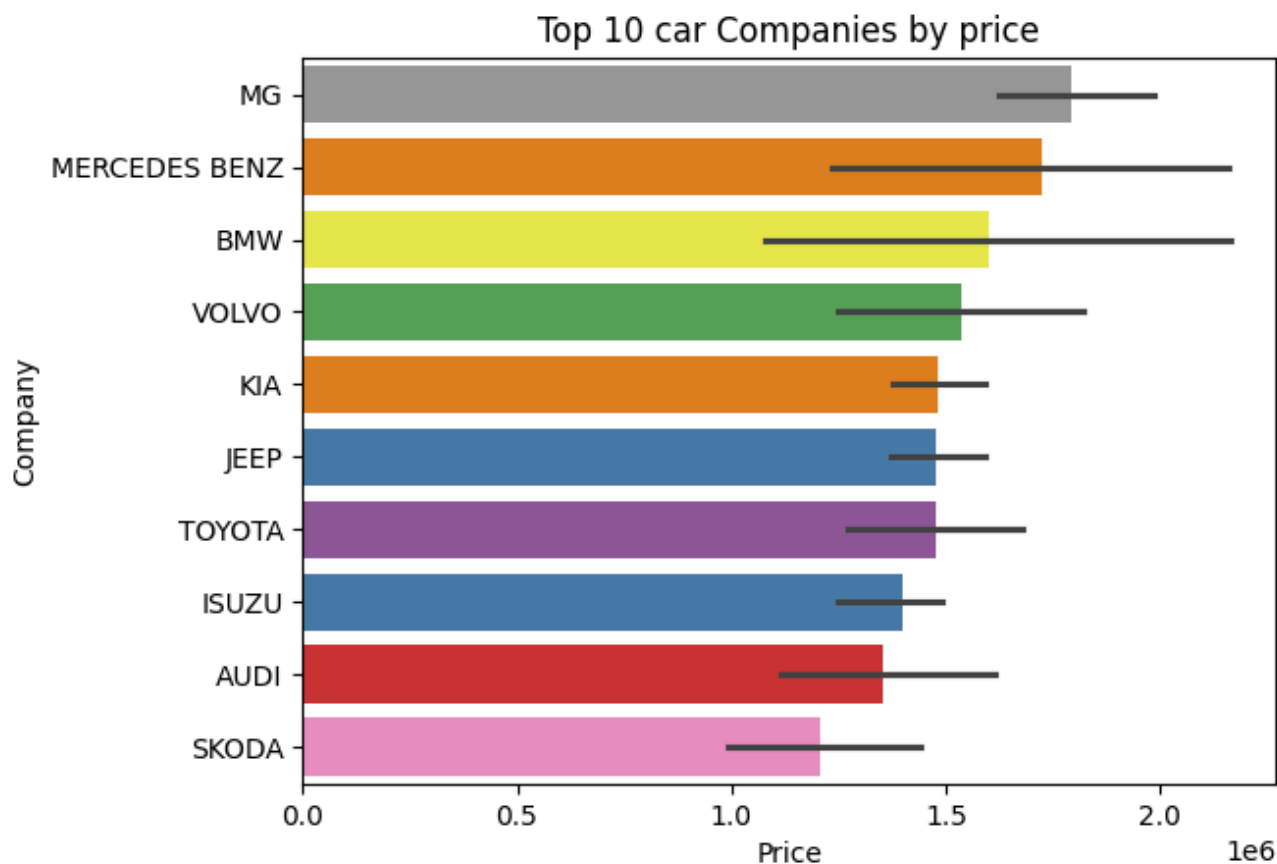
Quality score is an important feature which has a huge impact on the car sales and its preference by the customers. Cars with higher quality scores tend to have a much higher resale value and are more preferred by the customers. In the dataset, most of the cars have a decent quality score between 7-8, which highlights that the cars are thoroughly checked before being sold in the used car market. However, there are some cars with quality score less than 5, which could be due to the fact that they are not in good condition or they are very old.

**Till now, I have visualized the distribution of the data and got a better understanding of the data. Now, I will be looking at the relationship between the Car Price and the independent variables.**

### Top 10 Cars Companies by Price

```
# Top 10 car companies by price
sns.barplot(y = 'Company', x = 'Price', data = df, order = df.groupby('Company')['Price'])
```

```
Text(0.5, 1.0, 'Top 10 car Companies by price')
```



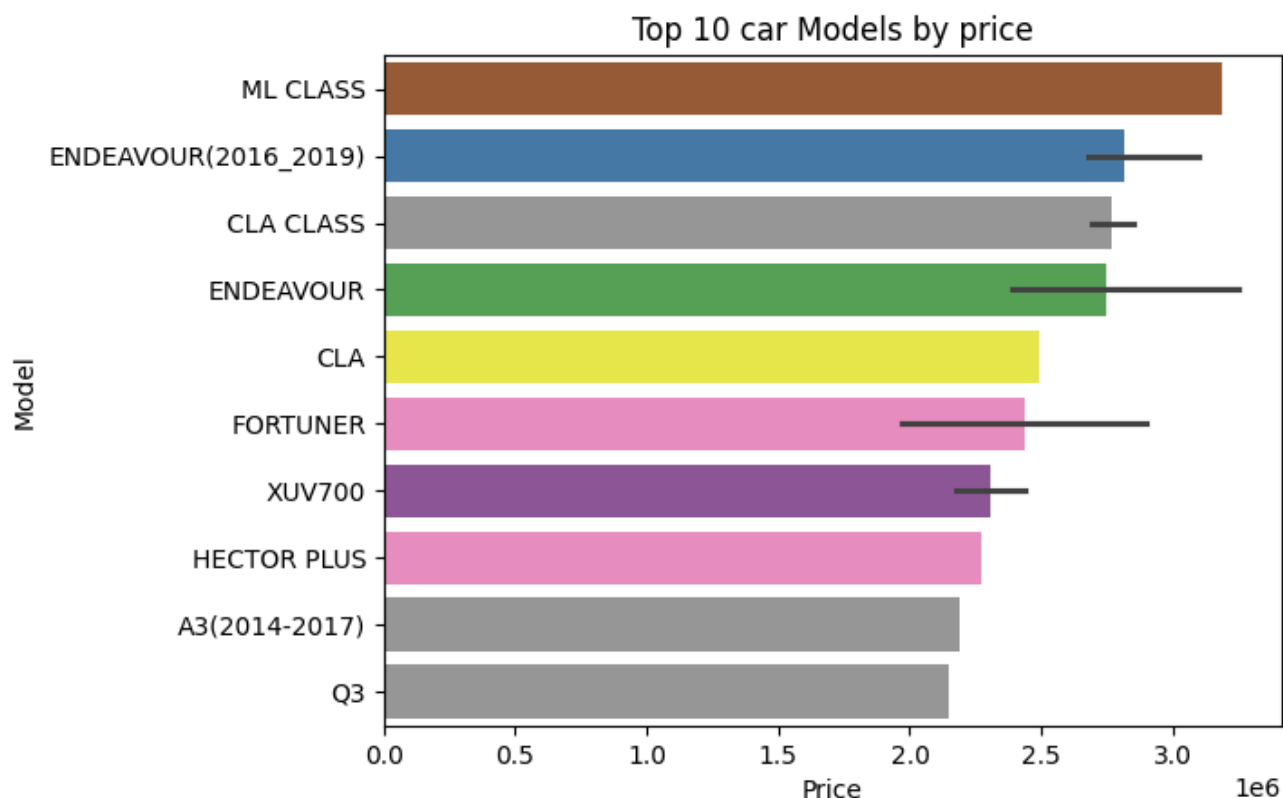
This graphs highlights the top 10 car companies in the dataset with the highest resale value. The MG, Mercedes Benz and BMW are the top 3 car companies with the highest resale value, since these are luxury car companies. The list also includes Volvo. followed by KIA, Jeep and Toyota. Surprisingly Audi has much lower resale price has compared to the other luxury car companies which might be due to other features.

Moreover, my previous hypothesis, about the car companies -Maruti Suzuki, Hyundai, Honda, Mahindra and Tata, was wrong as they are not in the top 10 list. This means that these companies cars are in greater number due to their demand because of low price

### Top 10 Car Models by Price

```
#Top 10 car models by price
sns.barplot(y = 'Model', x = 'Price', data = df, order = df.groupby('Model')['Price'].mea
```

```
Text(0.5, 1.0, 'Top 10 car Models by price')
```



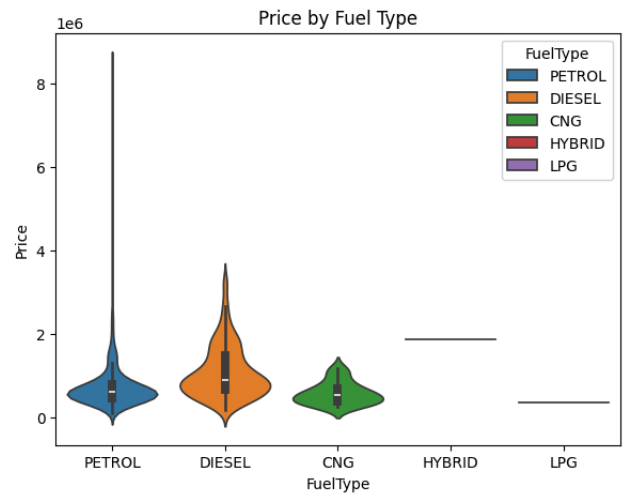
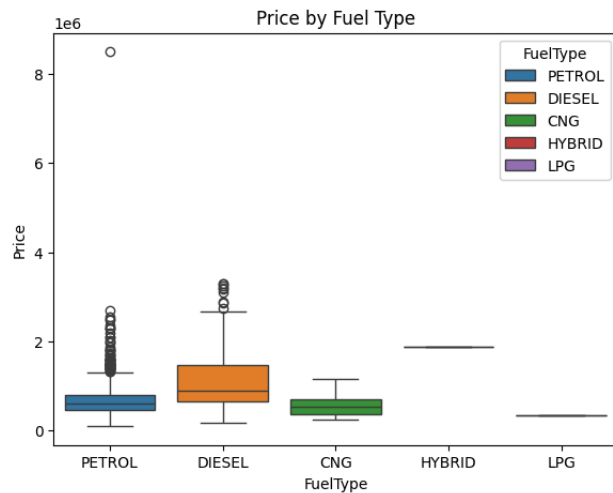
This graph shows the relation between the car model and its resale value and we can see that it shows similarity with the previous graph. The car models - ML Class, Endeavour(2016\_2019), CLA class are the top three models with highest resale value, followed by CLA, Fortuner and XUV700. Like the previous graph, the Audi model A3 is at the 9th position with a much lower resale value as compared to the other models.

In the car model also my hypothesis was wrong as I assumed that Honda City and Swift are the top two car models in the dataset, followed by Baleno, Creta and EcoSport. Therefore, we came to know that these cars are in higher number due to their high demand because of low price.

## Car Fuel Type and Price

```
fig, ax = plt.subplots(1,2,figsize=(15,5))
sns.boxplot(x = 'FuelType', y = 'Price', data = df, ax = ax[0], hue = 'FuelType').set_title('Boxplot of Price by Fuel Type')
sns.violinplot(x = 'FuelType', y = 'Price', data = df, ax = ax[1], hue = 'FuelType').set_title('Violinplot of Price by Fuel Type')
```

```
Text(0.5, 1.0, 'Price by Fuel Type')
```



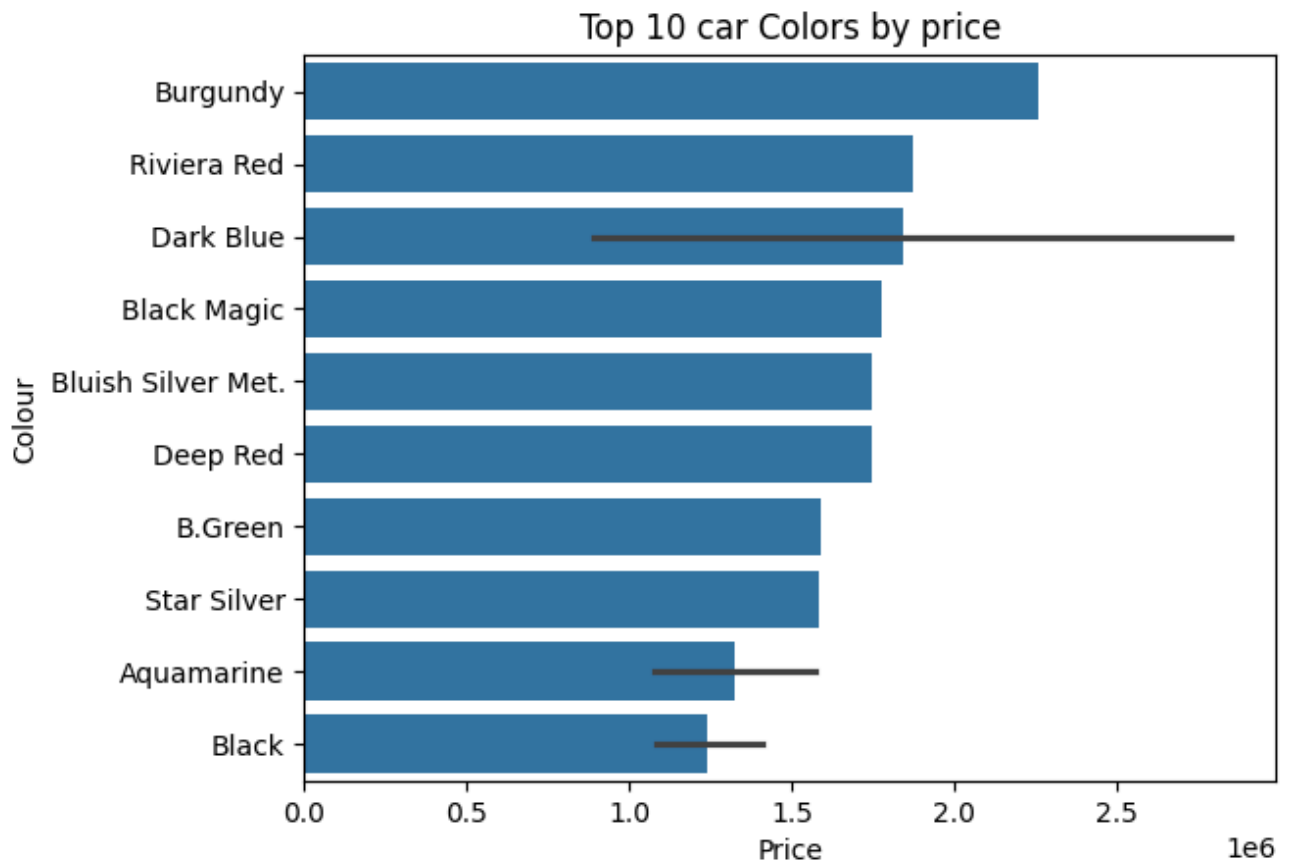
The above plots visualize the relationship between the car fuel type and its resale value. In the boxplot we can see that cars with diesel fuel type have higher resale value than petrol and CNG and LPG. In the violin plot, we can see that the distribution of the price for diesel cars is more concentrated between 10 to 20 lakh as compared to Petrol. From this it is cleared that, customers prefer petrol and diesel car than other fuel type and the diesel cars are more in demand in the used car market.

## Top 10 Car Colors by Price

```
#Top 10 car colors by price
```

```
sns.barplot(y = 'Colour', x = 'Price', data = df, order = df.groupby('Colour')['Price'].n
```

```
Text(0.5, 1.0, 'Top 10 car Colors by price')
```



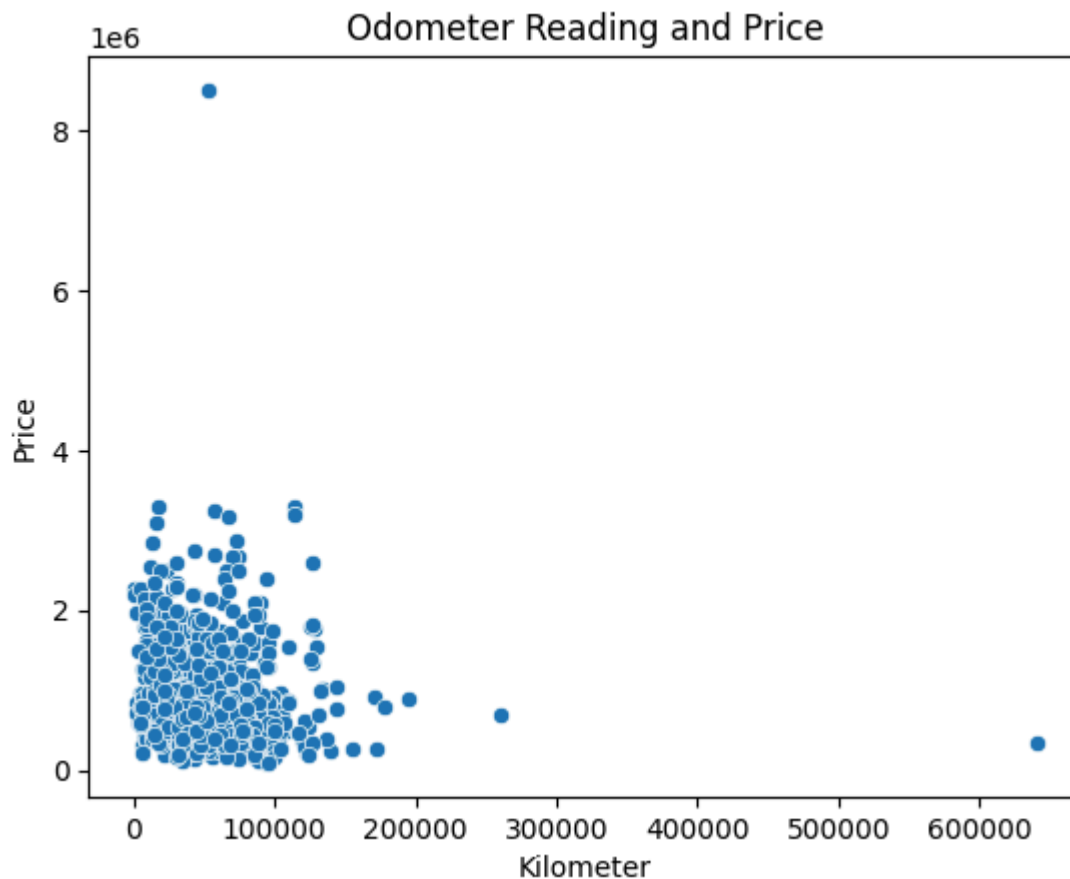
The cars with colors like Burgundy, Riviera Red and Dark Blue have higher resale value as compared to other colors. This shows that color of the car does matter and plays a major role in the resale value of the car.

Moreover, we also came to know that exotic colors have more price but they are not in demand in the used car market.

### Odometer Reading and Price

```
sns.scatterplot(x = 'Kilometer', y = 'Price', data = df).set_title('Odometer Reading and
```

```
Text(0.5, 1.0, 'Odometer Reading and Price')
```



In the scatter plot we can see that the data is concentrated near the origin, which means that most of the cars have odometer reading less than 100,000 km. In addition to that, the cars with less odometer reading show higher resale value, and as the odometer reading increases, the resale value decreases. Therefore, my hypothesis was correct that cars with odometer reading less than 100,000 km are more in demand in the used car market and will have a good resale value.

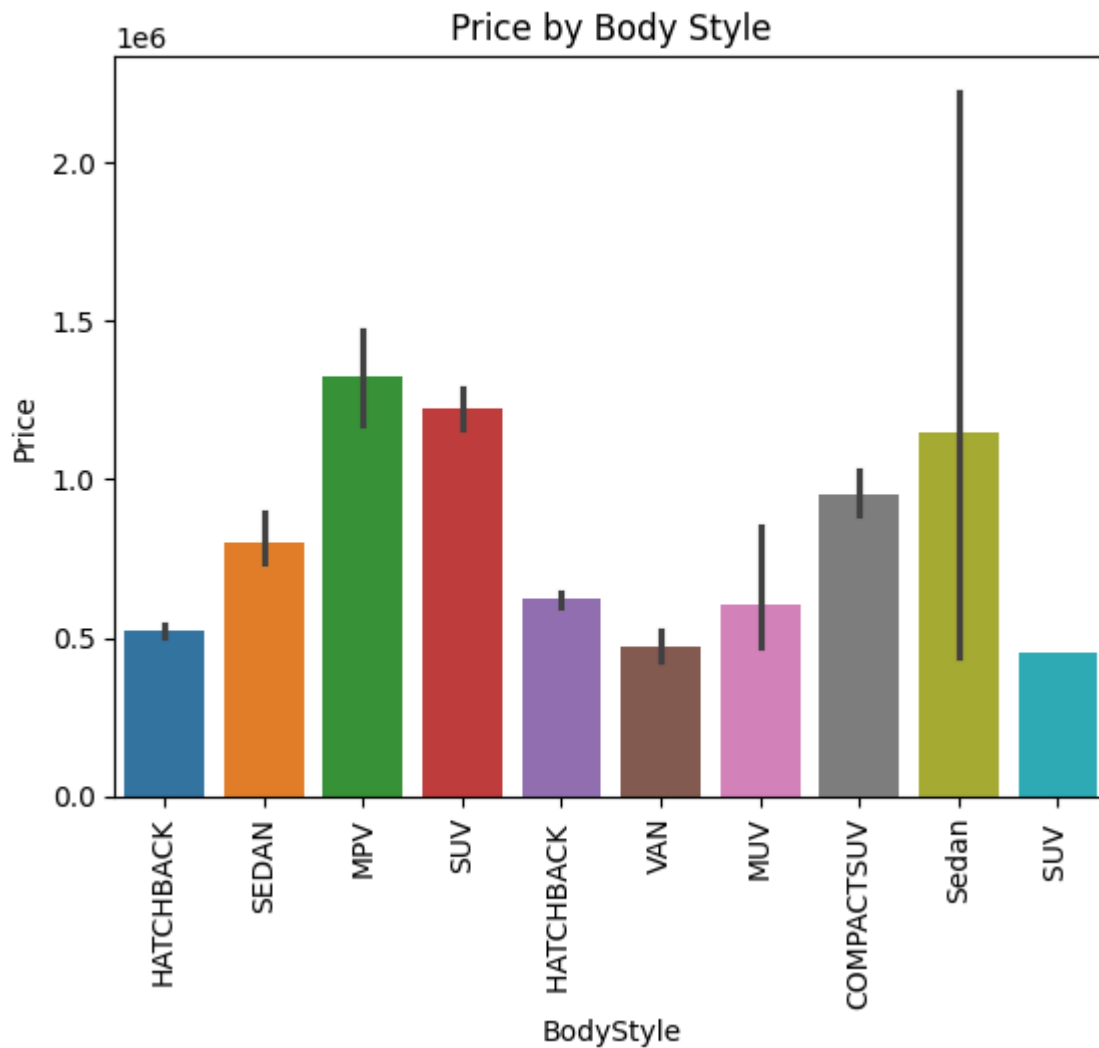
## Body Style and Price

```
sns.barplot(x = 'BodyStyle', y = 'Price', data = df, hue = 'BodyStyle').set_title('Price  
plt.xticks(rotation = 90)
```

```

→ ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
   [Text(0, 0, 'HATCHBACK'),
    Text(1, 0, 'SEDAN'),
    Text(2, 0, 'MPV'),
    Text(3, 0, 'SUV'),
    Text(4, 0, 'HATCHBACK '),
    Text(5, 0, 'VAN'),
    Text(6, 0, 'MUV'),
    Text(7, 0, 'COMPACTSUV'),
    Text(8, 0, 'Sedan'),
    Text(9, 0, 'SUV ')])

```



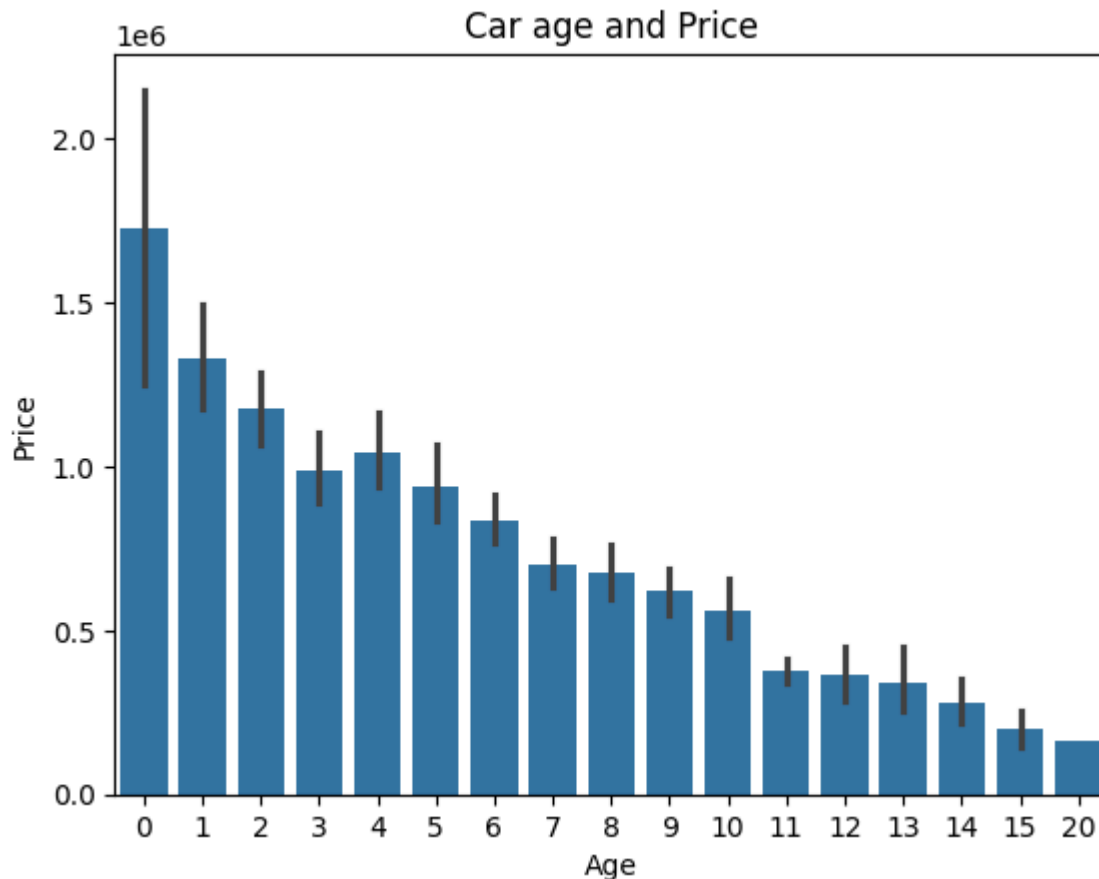
MPV, SUV and Sedan are the top 3 car body styles with the highest resale value. Therefore, we can assume that these body styles are more preferred in the used car market and have a good resale value. This also shows that my assumption was correct however, the Hatchback body style cars despite being in majority have lower resale value.

### Car Age and Price

```
sns.barplot(x = 'Age', y = 'Price', data = df).set_title('Car age and Price')
```



Text(0.5, 1.0, 'Car age and Price')



As we discussed earlier, age is a key determinant for a car's resale value and this graph clearly visualizes the relation of the age with car price. The cars with age less than a year has then highest price and as the age increases the prices decreases gradually. Therefore, my hypothesis was correct that cars with age less than 5 years have higher resale value.

### Location based Price Distribution

```
fig, ax = plt.subplots(1,3,figsize=(20,7))
```

```
#Dealer State
```

```
sns.violinplot(x = 'DealerState', y = 'Price', data = df, ax = ax[0], hue = 'DealerState')
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation = 90)
```

```
#City
```

```
sns.violinplot(x = 'City', y = 'Price', data = df, ax = ax[1], hue = 'City').set_title('City')
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation = 90)
```

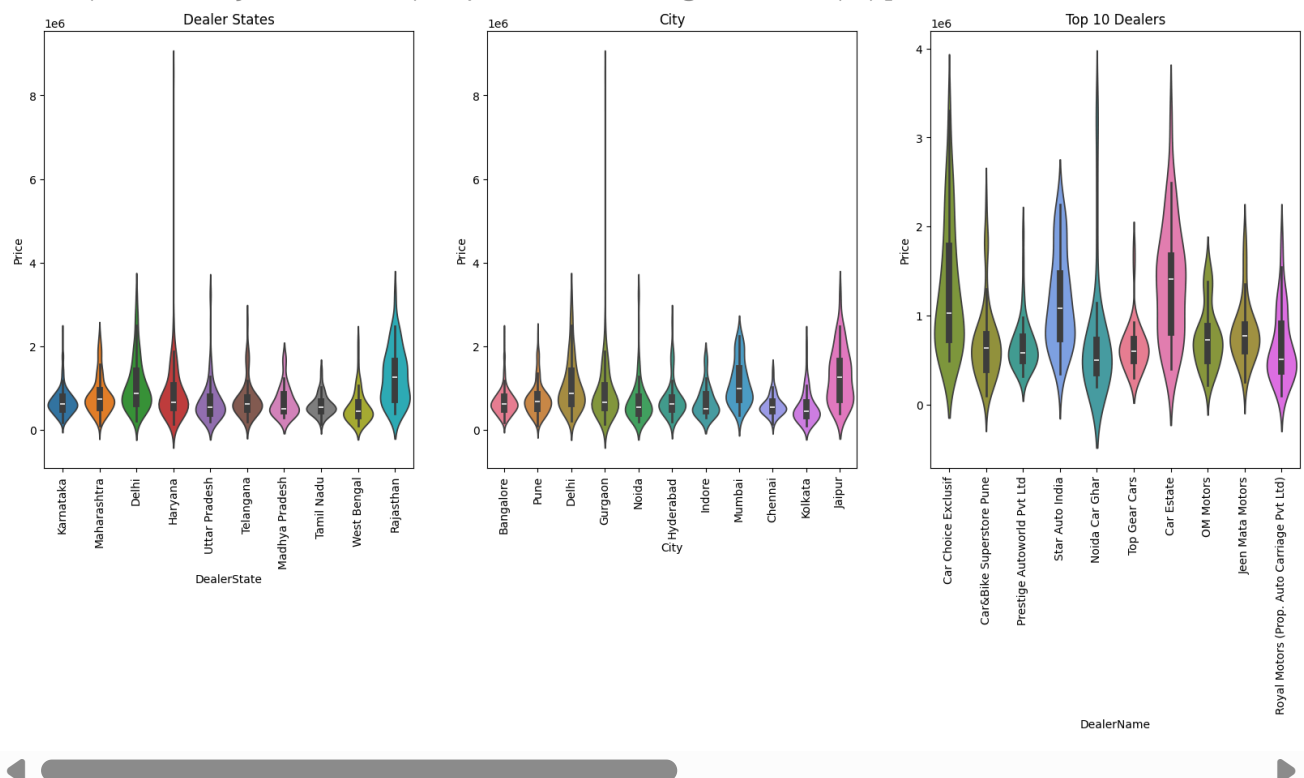
```
#top 10 dealers
```

```
sns.violinplot(x = 'DealerName', y = 'Price', data = df, order = df['DealerName'].value_counts().index)
ax[2].set_xticklabels(ax[2].get_xticklabels(), rotation = 90)
```

```

↳ <ipython-input-39-5e72a3224be2>:5: UserWarning: set_ticklabels() should only be used
  ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation = 90)
<ipython-input-39-5e72a3224be2>:9: UserWarning: set_ticklabels() should only be used
  ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation = 90)
<ipython-input-39-5e72a3224be2>:13: UserWarning: set_ticklabels() should only be used
  ax[2].set_xticklabels(ax[2].get_xticklabels(), rotation = 90)
[Text(0, 0, 'Car Choice Exclusif'),
 Text(1, 0, 'Car&Bike Superstore Pune'),
 Text(2, 0, 'Prestige Autoworld Pvt Ltd'),
 Text(3, 0, 'Star Auto India'),
 Text(4, 0, 'Noida Car Ghar'),
 Text(5, 0, 'Top Gear Cars'),
 Text(6, 0, 'Car Estate'),
 Text(7, 0, 'OM Motors'),
 Text(8, 0, 'Jeen Mata Motors'),
 Text(9, 0, 'Royal Motors (Prop. Auto Carriage Pvt Ltd)')]

```

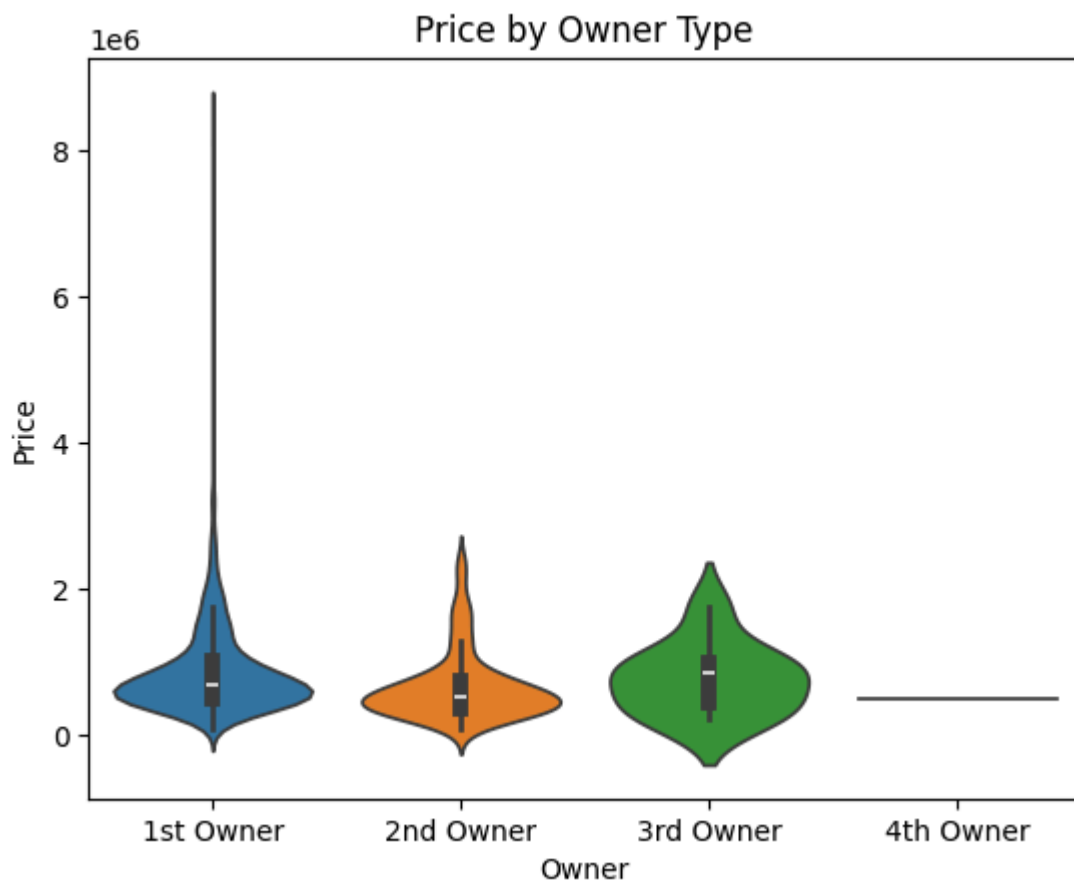


In the above graph we can see the price distribution based on the state, city and the dealer name. In the state graph, we can see that the cars in Rajasthan have the highest price followed by Delhi. Moreover, there are some outliers in the graph which are visible from the violinplot where there is strong peak in case of Haryana. In the city graph, we can see that the cars in Jaipur have the highest price followed by Mumbai and Delhi. Moreover, there are some outliers in the graph which are visible from the violinplot where there is strong peak in case of Gurgaon. In the dealer name graph, we can see the top 10 dealers along with their price distribution. Here, Car Estate has the highest price followed by Star Auto India and Car Choice. Moreover, there are some outliers in the graph which are visible from the violinplot where there is strong peak in case of Noida Car Ghar.

## Car Owner Type and Price

```
sns.violinplot(x = 'Owner', y = 'Price', data = df, hue = 'Owner').set_title('Price by Ow
```

```
↵↵ Text(0.5, 1.0, 'Price by Owner Type')
```

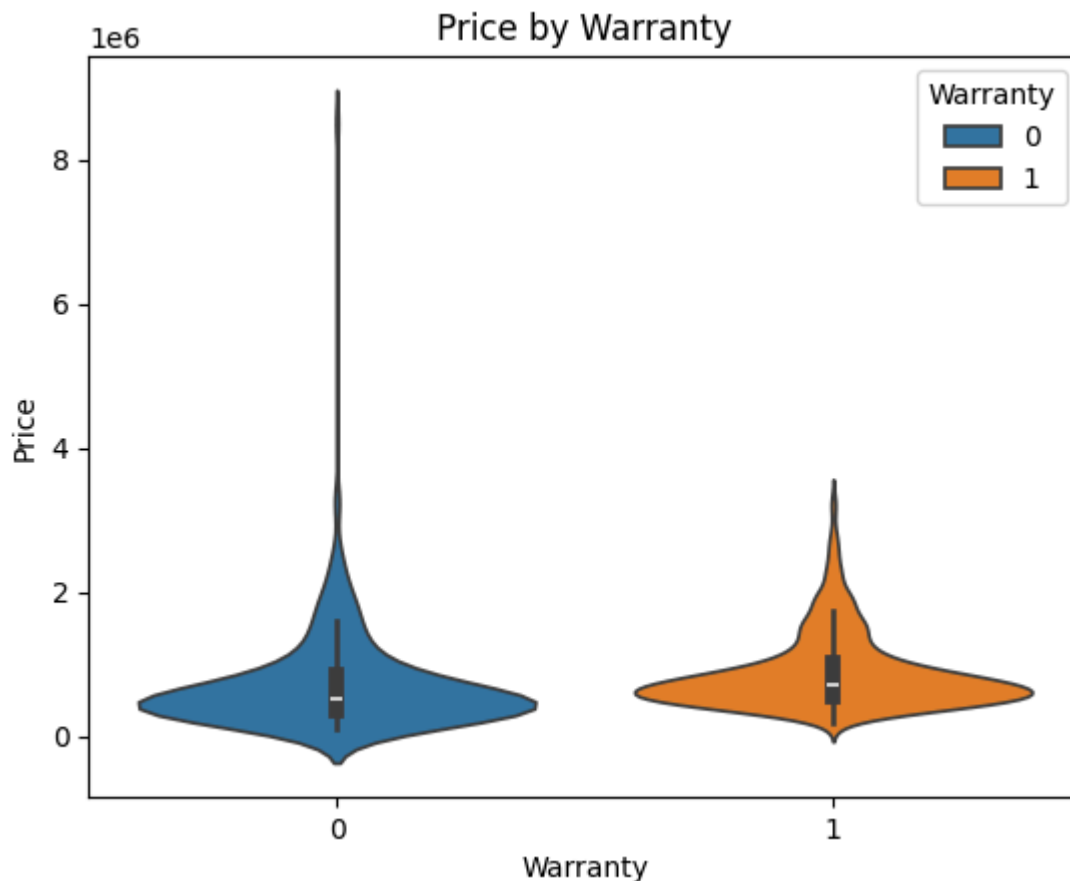


The graph shows the price distribution with respect to the car owner type. The cars with 1st owner have the highest price which is obvious as they are new cars. However, the 3rd Owner type cars despite being less in number have higher price than 2nd Owner type cars, which is not obvious. Therefore, we can assume that 3rd Owner type cars having higher price could be some luxury or vintage cars.

## Warranty and Price

```
sns.violinplot(x = 'Warranty', y = 'Price', data = df, hue = 'Warranty').set_title('Price
```

```
Text(0.5, 1.0, 'Price by Warranty')
```



Here, we can see some change in the violinplot of the cars with and without warranty. The cars with warranty tends to have slightly higher price than the cars without warranty. Therefore, we can assume that cars with warranty are more preferred in the used car market and have a good resale value.

### Quality Score and Price

```
sns.scatterplot(x = 'QualityScore', y = 'Price', data = df).set_title('Quality Score and
```

```
Text(0.5, 1.0, 'Quality Score and Price')
```



We can see a very high concentration near the quality score 7 and above having much higher price than the cars with quality score less than 7. Therefore, we can assume that cars with quality score 7 and above are more preferred in the used car market and have a good resale value.

## ✓ Data Preprocessing Part 2

Dropping column car model because, it has too many unique values and it will increase the dimensionality of the dataset.

```
df.drop('Model', axis = 1, inplace = True)
```

### Label Encoding

```
#columns for label encoding
cols = df.select_dtypes(include=['object']).columns

from sklearn.preprocessing import LabelEncoder
#Label encoding object
le = LabelEncoder()
```

```
#label encoding for object type columns
for i in cols:
    le.fit(df[i])
    df[i] = le.transform(df[i])
    print(i, df[i].unique())
```

```
⇒ Company [12  7 19  5 13 21 11  6 17 16  9  4 20 10  1  3 18 14  0  8 22 15  2]
FuelType [4 1 0 2 5 3]
Colour [61 56 34  0  9 11 66 47 49 38 14 71 72 30 74 52 39 28 60  7 54 62 40 13
 20 70 63 12 24 23 35 26 29 15 31  1 68  4  8 73 22 44 57 65 42 50 32 64
 19 43 46 33 16 27 53 25 10 69 51 17  6 48 59 58  5  3 18 45 67 36 21 55
  2 37 75 41]
BodyStyle [1 5 3 6 2 9 4 0 8 7]
Owner [0 1 2 3]
DealerState [2 4 0 1 8 7 3 6 9 5]
DealerName [52 38  4  1 56 29  0 34 47 51 11 21  9 10 43 33  7 16  5 12 42 17 27 50
 45  6 20 36 23 41 32 31 18  2 48 15 54 40 55 13 49 25 35 46 24 14 44 19
 39 28 26  3 53 30  8 22 37]
City [ 0 10  2  3  9  4  5  8  1  7  6]
```

## Outlier Removal

```
#Using IQRS to remove outliers
```

```
#columns for outlier removal
cols = df.select_dtypes(include=['int64','float64']).columns
Q1 = df[cols].quantile(0.25)
Q3 = df[cols].quantile(0.75)
```

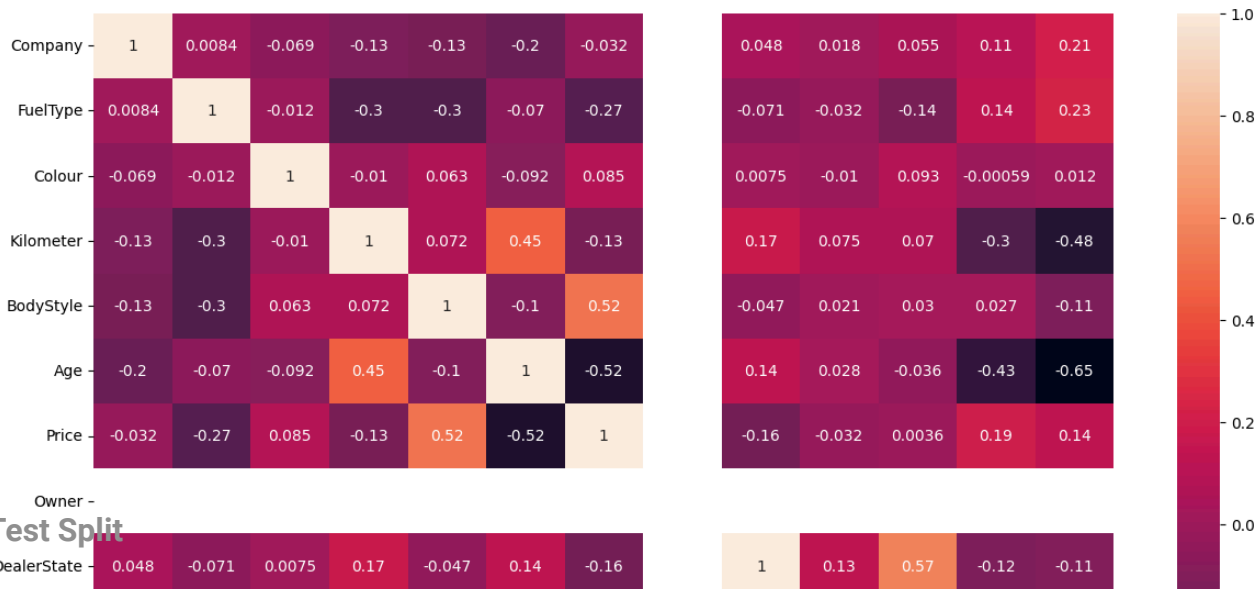
```
IQR = Q3 - Q1
```

```
#Removing outliers
```

```
df = df[~((df[cols] < (Q1 - 1.5 * IQR)) |(df[cols] > (Q3 + 1.5 * IQR))).any(axis=1)]
```

## Correlation Matrix Heatmap

```
plt.figure(figsize=(15,10))
sns.heatmap(df.corr(), annot=True)
```

 <Axes: >


## Train-Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.drop('Price',axis=1), df['Price'],
```

## Model Building

I will be using the following regression models:

- Decision Tree Regressor
- Random Forest Regressor
- Ridge Regressor

## Decision Tree Regressor

```
from sklearn.tree import DecisionTreeRegressor
#Decision Tree Regressor Object
dtr = DecisionTreeRegressor()
```

## Hyperparameter Tuning

```
from sklearn.model_selection import GridSearchCV
```

```
#parameters for grid search
```