Car Dekho.com : EDA(Exploratiry Data Analysis)

Project Objective:

The primary objective of this project was to perform an in-depth Exploratory Data Analysis (EDA) on the CarDekho used-car dataset to understand market trends, buyer preferences, pricing patterns, and inventory distribution. The goal was to uncover meaningful insights that can help improve decision-making, optimize inventory strategy, enhance pricing models, and identify opportunities for increasing sales and customer satisfaction on the CarDekho platform.

Step 1 : Import Required Modules

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import warnings
warnings.filterwarnings('ignore')

print('All Modules Done')
```

```
All Modules Done
```

```
pip install kagglehub
```

```
Requirement already satisfied: kagglehub in c:\users\saurabh\
anaconda3\lib\site-packages (0.3.13)
Requirement already satisfied: packaging in c:\users\saurabh\
anaconda3\lib\site-packages (from kagglehub) (24.1)
Requirement already satisfied: pyyaml in c:\users\saurabh\anaconda3\
lib\site-packages (from kagglehub) (6.0.1)
Requirement already satisfied: requests in c:\users\saurabh\anaconda3\
lib\site-packages (from kagglehub) (2.32.3)
Requirement already satisfied: tqdm in c:\users\saurabh\anaconda3\lib\
site-packages (from kagglehub) (4.66.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
saurabh\anaconda3\lib\site-packages (from requests->kagglehub) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\saurabh\
anaconda3\lib\site-packages (from requests->kagglehub) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\saurabh\
anaconda3\lib\site-packages (from requests->kagglehub) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\saurabh\
anaconda3\lib\site-packages (from requests->kagglehub) (2025.8.3)
Requirement already satisfied: colorama in c:\users\saurabh\anaconda3\
lib\site-packages (from tqdm->kagglehub) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

```python
# Import the Car Dekho Data from Kaggle

import kagglehub

# Download latest version
path = kagglehub.dataset_download("manishkr1754/cardekho-used-car-
data")

print("Path to dataset files:", path)
```

```
Path to dataset files: C:\Users\Saurabh\.cache\kagglehub\datasets\
manishkr1754\cardekho-used-car-data\versions\2
```

```python
# check no. of files are in path and create file name
file_name = os.listdir(path)[0]
file_name = path + '/' + file_name
print(file_name)
```

```
C:\Users\Saurabh\.cache\kagglehub\datasets\manishkr1754\cardekho-used-
car-data\versions\2/cardekho_dataset.csv
```

```python
# read the csv files

df = pd.read_csv(file_name)
print('Done')
```

```
Done
```

Step 2 : EDA(Exploratiry Data Analysis)

```
#2.1 head
df.head()
```

```
   Unnamed: 0       car_name    brand     model  vehicle_age
km_driven  \
0           0    Maruti Alto   Maruti      Alto            9
120000
1           1  Hyundai Grand  Hyundai     Grand            5
20000
2           2    Hyundai i20  Hyundai       i20           11
60000
3           3    Maruti Alto   Maruti      Alto            9
37000
4           4  Ford Ecosport     Ford  Ecosport            6
30000

  seller_type fuel_type transmission_type  mileage  engine  max_power
seats  \
0  Individual    Petrol            Manual    19.70     796      46.30
5
1  Individual    Petrol            Manual    18.90    1197      82.00
5
2  Individual    Petrol            Manual    17.00    1197      80.00
5
3  Individual    Petrol            Manual    20.92     998      67.10
5
4      Dealer    Diesel            Manual    22.77    1498      98.59
5

   selling_price
0         120000
1         550000
2         215000
3         226000
4         570000
```

```
#2.2 Name of all columns
df.columns
```

```
Index(['Unnamed: 0', 'car_name', 'brand', 'model', 'vehicle_age',
'km_driven',
       'seller_type', 'fuel_type', 'transmission_type', 'mileage',
'engine',
       'max_power', 'seats', 'selling_price'],
      dtype='object')
```

```
# 2.3 Drop Unnamed Columns
df1 = df.drop(columns = ['Unnamed: 0'])
```

```
df1
```

```
                car_name     brand     model  vehicle_age  km_driven  \
0            Maruti Alto    Maruti      Alto            9     120000
1          Hyundai Grand   Hyundai     Grand            5      20000
2            Hyundai i20   Hyundai       i20           11      60000
3            Maruti Alto    Maruti      Alto            9      37000
4          Ford Ecosport      Ford  Ecosport            6      30000
...                  ...       ...       ...          ...        ...
15406        Hyundai i10   Hyundai       i10            9      10723
15407      Maruti Ertiga    Maruti    Ertiga            2      18000
15408        Skoda Rapid     Skoda     Rapid            6      67000
15409  Mahindra XUV500  Mahindra    XUV500            5    3800000
15410         Honda City     Honda      City            2      13000

      seller_type fuel_type transmission_type  mileage  engine
max_power  \
0      Individual    Petrol            Manual    19.70     796
46.30
1      Individual    Petrol            Manual    18.90    1197
82.00
2      Individual    Petrol            Manual    17.00    1197
80.00
3      Individual    Petrol            Manual    20.92     998
67.10
4          Dealer    Diesel            Manual    22.77    1498
98.59
...           ...       ...               ...      ...     ...
...
15406      Dealer    Petrol            Manual    19.81    1086
68.05
15407      Dealer    Petrol            Manual    17.50    1373
91.10
15408      Dealer    Diesel            Manual    21.14    1498
103.52
15409      Dealer    Diesel            Manual    16.00    2179
140.00
15410      Dealer    Petrol         Automatic    18.00    1497
117.60

       seats  selling_price
0          5         120000
1          5         550000
2          5         215000
3          5         226000
4          5         570000
...      ...            ...
15406      5         250000
15407      7         925000
15408      5         425000
```

```
15409      7        1225000
15410      5        1200000

[15411 rows x 13 columns]
```

```
#2.4 checking the information od dataset
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15411 entries, 0 to 15410
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   car_name           15411 non-null  object 
 1   brand              15411 non-null  object 
 2   model              15411 non-null  object 
 3   vehicle_age        15411 non-null  int64  
 4   km_driven          15411 non-null  int64  
 5   seller_type        15411 non-null  object 
 6   fuel_type          15411 non-null  object 
 7   transmission_type  15411 non-null  object 
 8   mileage            15411 non-null  float64
 9   engine             15411 non-null  int64  
 10  max_power          15411 non-null  float64
 11  seats              15411 non-null  int64  
 12  selling_price      15411 non-null  int64  
dtypes: float64(2), int64(5), object(6)
memory usage: 1.5+ MB
```

```
# 2.5 Last 5 rows of dataset
df.tail()
```

```
       Unnamed: 0         car_name     brand   model  vehicle_age
km_driven  \
15406       19537      Hyundai i10   Hyundai     i10            9
10723
15407       19540    Maruti Ertiga    Maruti  Ertiga            2
18000
15408       19541      Skoda Rapid     Skoda   Rapid            6
67000
15409       19542  Mahindra XUV500  Mahindra  XUV500            5
3800000
15410       19543       Honda City     Honda    City            2
13000

      seller_type fuel_type transmission_type  mileage  engine
max_power  \
15406      Dealer    Petrol            Manual    19.81    1086
68.05
15407      Dealer    Petrol            Manual    17.50    1373
```

```
91.10
15408       Dealer    Diesel            Manual    21.14    1498
103.52
15409       Dealer    Diesel            Manual    16.00    2179
140.00
15410       Dealer    Petrol         Automatic    18.00    1497
117.60

       seats  selling_price
15406      5         250000
15407      7         925000
15408      5         425000
15409      7        1225000
15410      5        1200000
```

```
# 2.6 check the na values
df1.isna().sum()
```

```
car_name             0
brand                0
model                0
vehicle_age          0
km_driven            0
seller_type          0
fuel_type            0
transmission_type    0
mileage              0
engine               0
max_power            0
seats                0
selling_price        0
dtype: int64
```

```
# 2.7 describe the dataset
df.describe().round(2)
```

```
       Unnamed: 0  vehicle_age   km_driven   mileage    engine
max_power  \
count    15411.00     15411.00    15411.00  15411.00  15411.00
15411.00
mean      9811.86         6.04    55616.48     19.70   1486.06
100.59
std       5643.42         3.01    51618.55      4.17    521.11
42.97
min          0.00         0.00      100.00      4.00    793.00
38.40
25%       4906.50         4.00    30000.00     17.00   1197.00
74.00
50%       9872.00         6.00    50000.00     19.67   1248.00
88.50
```

```
75%      14668.50         8.00    70000.00      22.70   1582.00
117.30
max      19543.00        29.00  3800000.00      33.54   6592.00
626.00

            seats  selling_price
count  15411.00       15411.00
mean       5.33      774971.12
std        0.81      894128.36
min        0.00       40000.00
25%        5.00      385000.00
50%        5.00      556000.00
75%        5.00      825000.00
max        9.00    39500000.00
```

```python
df.describe(include='object')
```

```
             car_name   brand  model seller_type fuel_type
transmission_type
count           15411   15411  15411       15411     15411
15411
unique            121      32    120           3         5
2
top      Hyundai i20  Maruti    i20      Dealer    Petrol
Manual
freq              906    4992    906        9539      7643
12225
```

```python
df.describe(include='all').round()
```

```
        Unnamed: 0     car_name   brand  model  vehicle_age  km_driven
\
count      15411.0        15411   15411  15411      15411.0    15411.0

unique         NaN          121      32    120          NaN        NaN

top            NaN  Hyundai i20  Maruti    i20          NaN        NaN

freq           NaN          906    4992    906          NaN        NaN

mean        9812.0          NaN     NaN    NaN          6.0    55616.0

std         5643.0          NaN     NaN    NaN          3.0    51619.0

min            0.0          NaN     NaN    NaN          0.0      100.0

25%         4906.0          NaN     NaN    NaN          4.0    30000.0

50%         9872.0          NaN     NaN    NaN          6.0    50000.0

75%        14668.0          NaN     NaN    NaN          8.0    70000.0
```

```
max        19543.0          NaN      NaN     NaN           29.0  3800000.0


       seller_type fuel_type transmission_type  mileage   engine
max_power  \
count        15411     15411             15411  15411.0  15411.0
15411.0
unique           3         5                 2      NaN      NaN
NaN
top         Dealer    Petrol            Manual      NaN      NaN
NaN
freq          9539      7643             12225      NaN      NaN
NaN
mean           NaN       NaN               NaN     20.0   1486.0
101.0
std            NaN       NaN               NaN      4.0    521.0
43.0
min            NaN       NaN               NaN      4.0    793.0
38.0
25%            NaN       NaN               NaN     17.0   1197.0
74.0
50%            NaN       NaN               NaN     20.0   1248.0
88.0
75%            NaN       NaN               NaN     23.0   1582.0
117.0
max            NaN       NaN               NaN     34.0   6592.0
626.0

          seats  selling_price
count   15411.0        15411.0
unique      NaN            NaN
top         NaN            NaN
freq        NaN            NaN
mean        5.0       774971.0
std         1.0       894128.0
min         0.0        40000.0
25%         5.0       385000.0
50%         5.0       556000.0
75%         5.0       825000.0
max         9.0     39500000.0
```

Value Counts

```
# 2.8 Dividing the dataset into two parts Numerical & Categorical
# Display numerical & categarical columns from dataset
```

```
num_col = df1.select_dtypes('number').columns
cat_col = df1.select_dtypes('object').columns

print(num_col) # Columns which have numeric values

Index(['vehicle_age', 'km_driven', 'mileage', 'engine', 'max_power',
'seats',
       'selling_price'],
      dtype='object')

print(cat_col) # Columns which have Object/String values

Index(['car_name', 'brand', 'model', 'seller_type', 'fuel_type',
       'transmission_type'],
      dtype='object')
```

Step 3 : Univariate

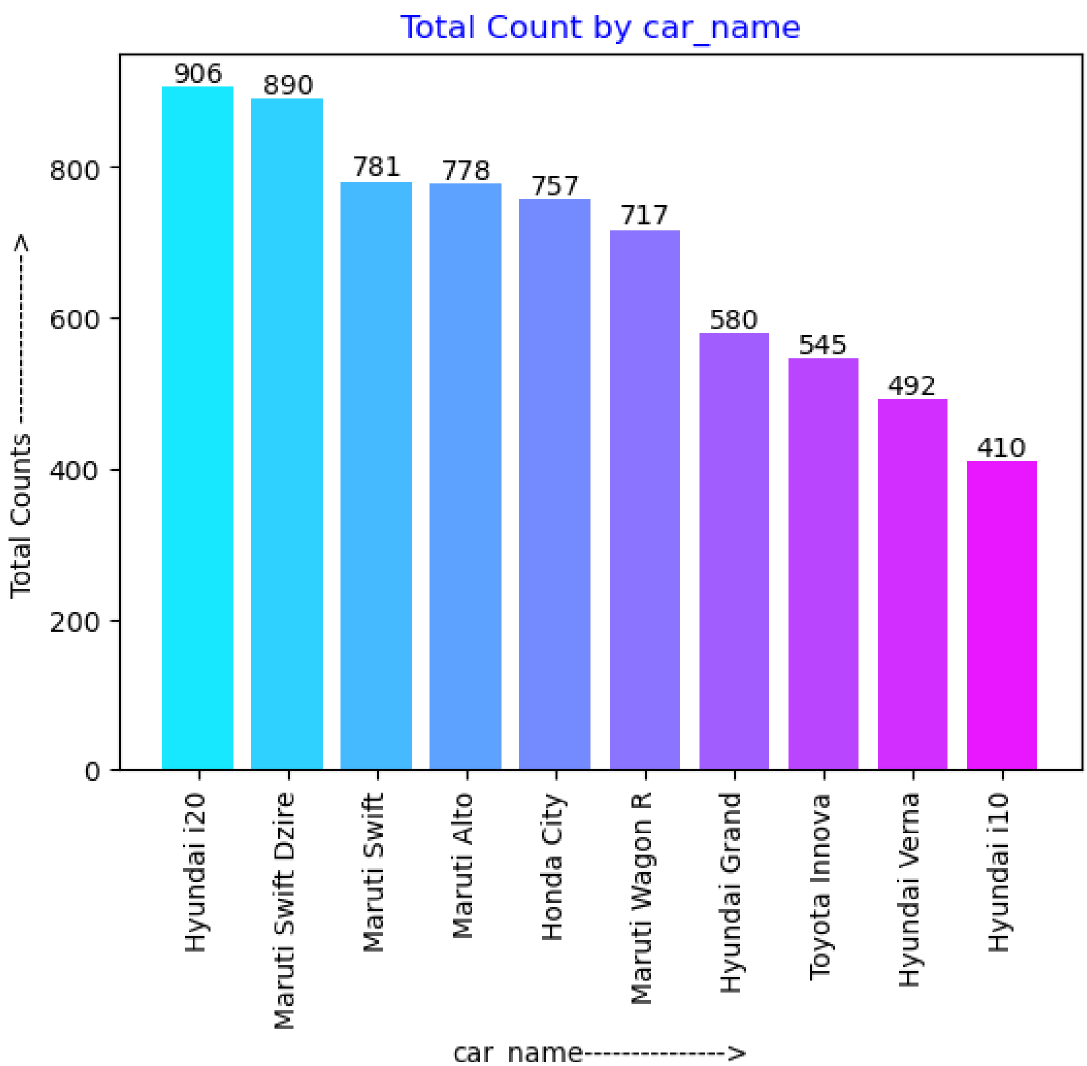
Univariate analysis means analyzing one variable (one column) at a time.

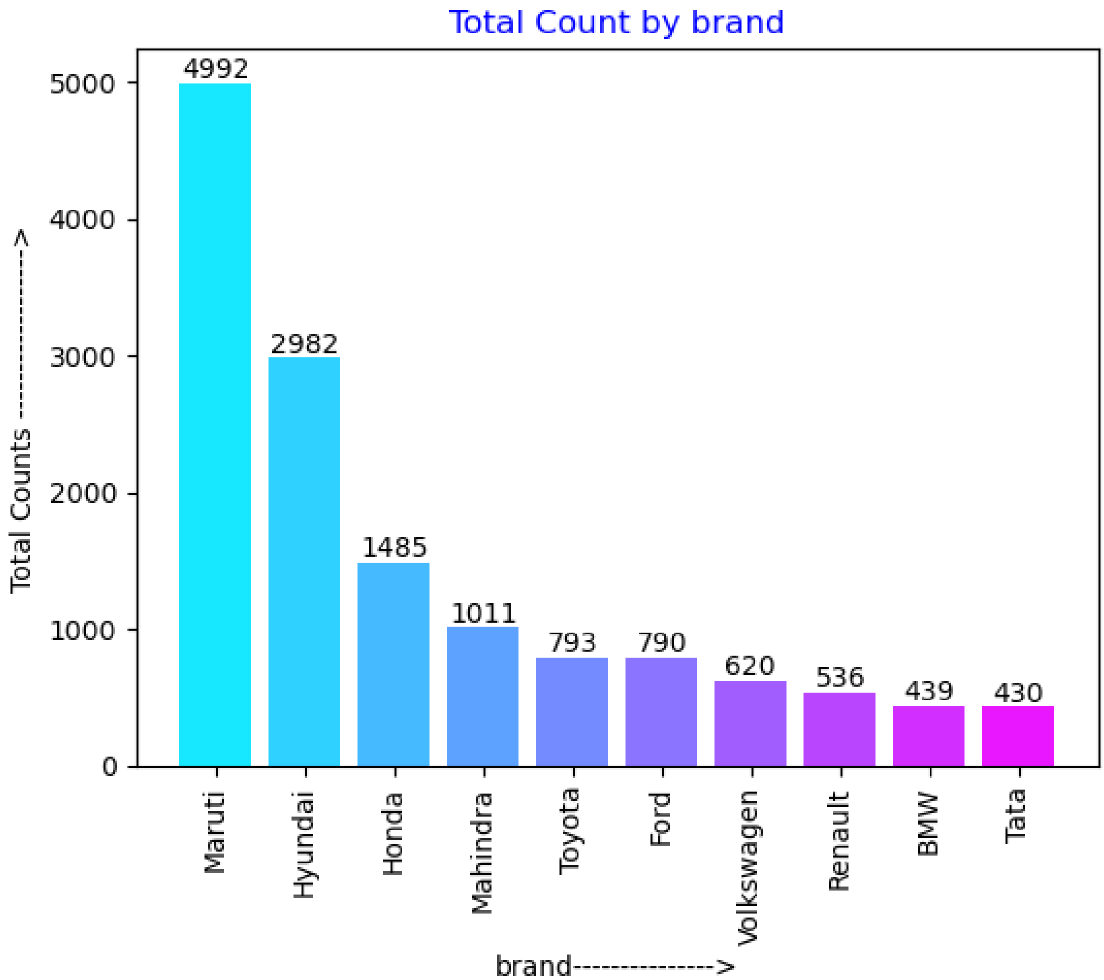
Goal: understand its distribution, central values, and spread

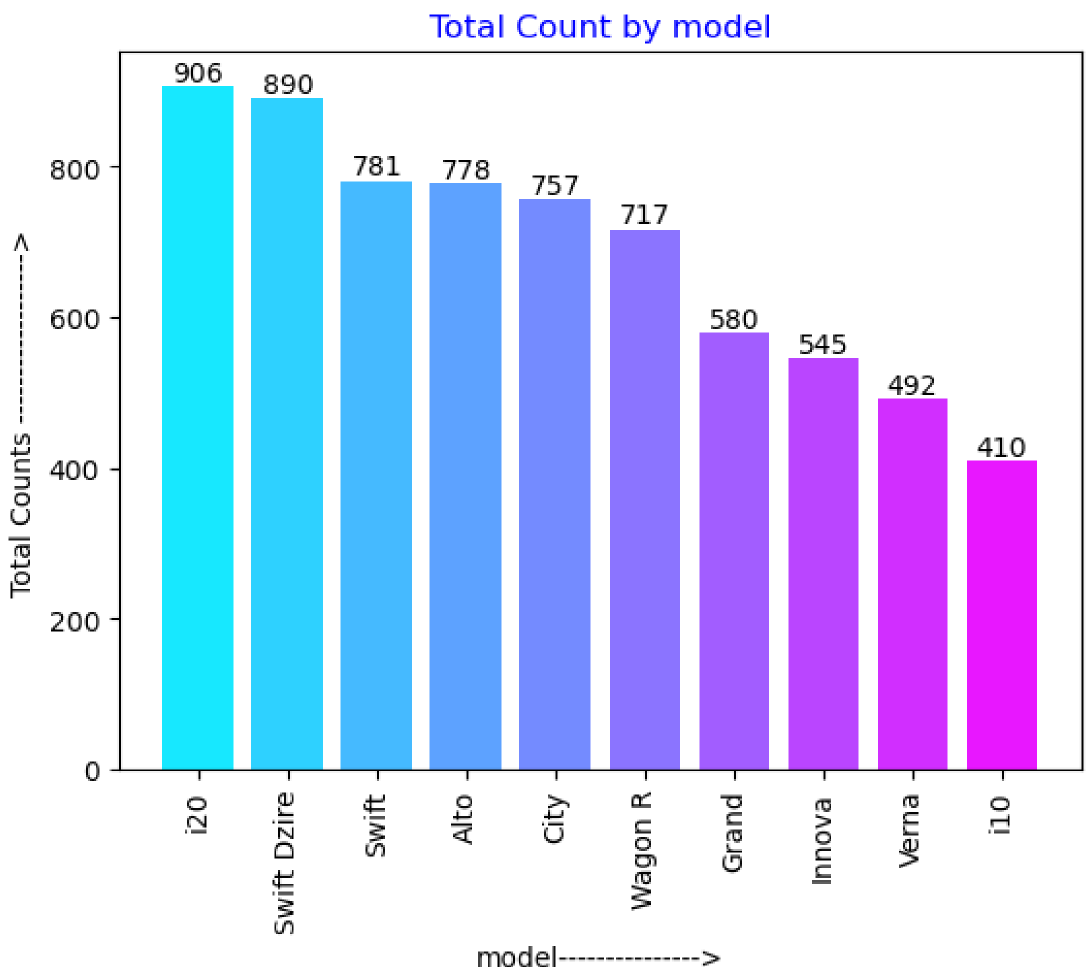
```
# 3.1 display TOP 10 data count  according its  columns data analysis
for i in cat_col:
    print(f'{i} column Analysis >>',df1[i].value_counts().head(10))
    print('-------------------------------------')
    print(end='\n'*2)

# Note : value_count always perform on categorical data.

car_name column Analysis >> car_name
Hyundai i20           906
Maruti Swift Dzire    890
Maruti Swift          781
Maruti Alto           778
Honda City            757
Maruti Wagon R        717
Hyundai Grand         580
Toyota Innova         545
Hyundai Verna         492
Hyundai i10           410
Name: count, dtype: int64
-------------------------------------


brand column Analysis >> brand
Maruti        4992
```

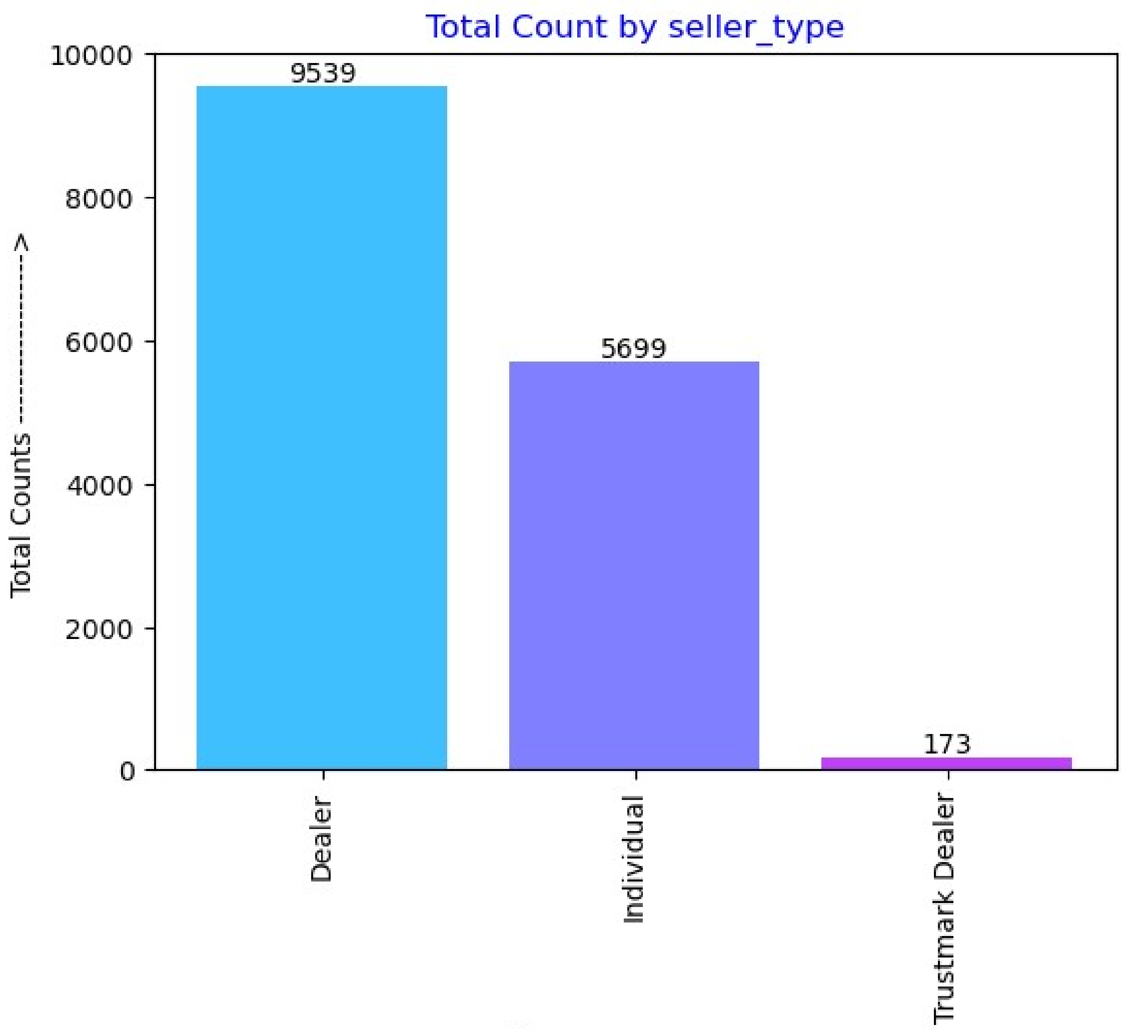
```
Hyundai       2982
Honda         1485
Mahindra      1011
Toyota         793
Ford           790
Volkswagen     620
Renault        536
BMW            439
Tata           430
Name: count, dtype: int64
------------------------------------


model column Analysis >> model
i20            906
Swift Dzire    890
Swift          781
Alto           778
City           757
Wagon R        717
Grand          580
Innova         545
Verna          492
i10            410
Name: count, dtype: int64
------------------------------------


seller_type column Analysis >> seller_type
Dealer              9539
Individual          5699
Trustmark Dealer     173
Name: count, dtype: int64
------------------------------------


fuel_type column Analysis >> fuel_type
Petrol      7643
Diesel      7419
CNG          301
LPG           44
Electric       4
Name: count, dtype: int64
------------------------------------


transmission_type column Analysis >> transmission_type
Manual       12225
Automatic     3186
Name: count, dtype: int64
```

```
----------------------------------



# 3.2 create bar chart using TOP 10 data count  according its columns wise data analysis

for i in cat_col:
    x=  df1[i].value_counts().head(10).index
    y= df1[i].value_counts().head(10).values
    chart = plt.bar(x,y,color = sns.color_palette('cool',len(x)))
    plt.bar_label(chart)
    plt.title(f'Total Count by {i}',color= 'b')
    plt.xlabel(f'{i}--------------->')
    plt.ylabel('Total Counts ------------------>')
    plt.xticks(rotation = 90)
    plt.show()
```

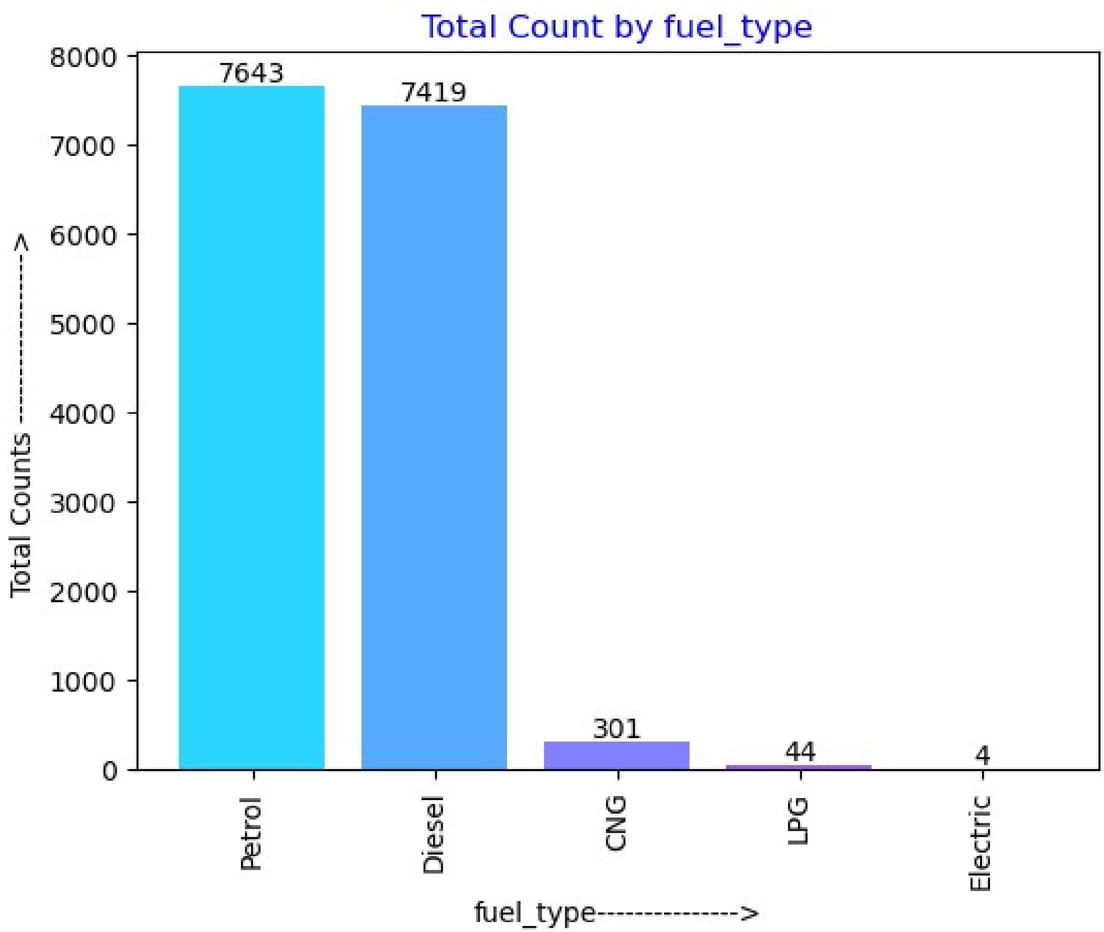


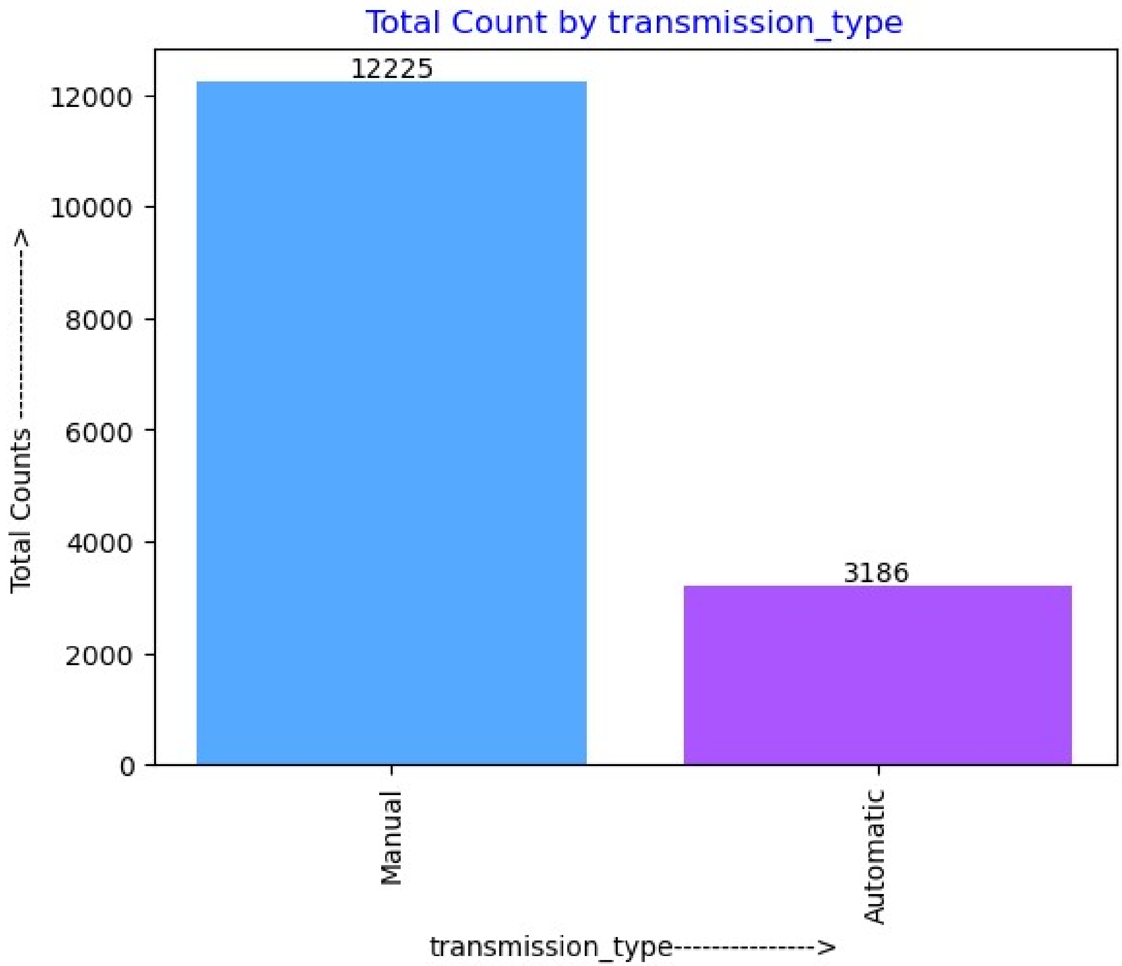

Total Count by brand
Total Counts ------------------->
brand-------------------->
5000
4000
3000
2000
1000
0
4992
2982
1485
1011
793
790
620
536
439
430
Maruti
Hyundai
Honda
Mahindra
Toyota
Ford
Volkswagen
Renault
BMW
Tata


Total Count by model
906
890
781
778
757
717
580
545
492
410
800
600
400
200
0
Total Counts ------------>
i20
Swift Dzire
Swift
Alto
City
Wagon R
Grand
Innova
Verna
i10
model------------->

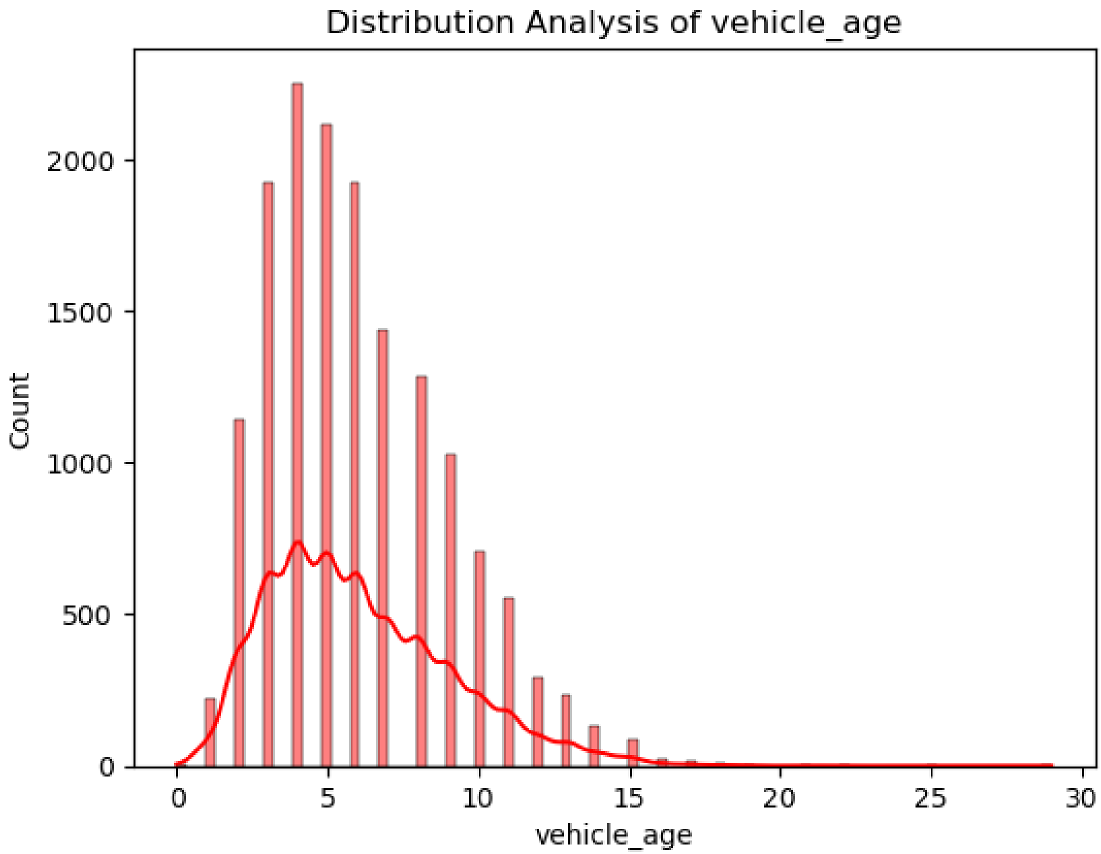


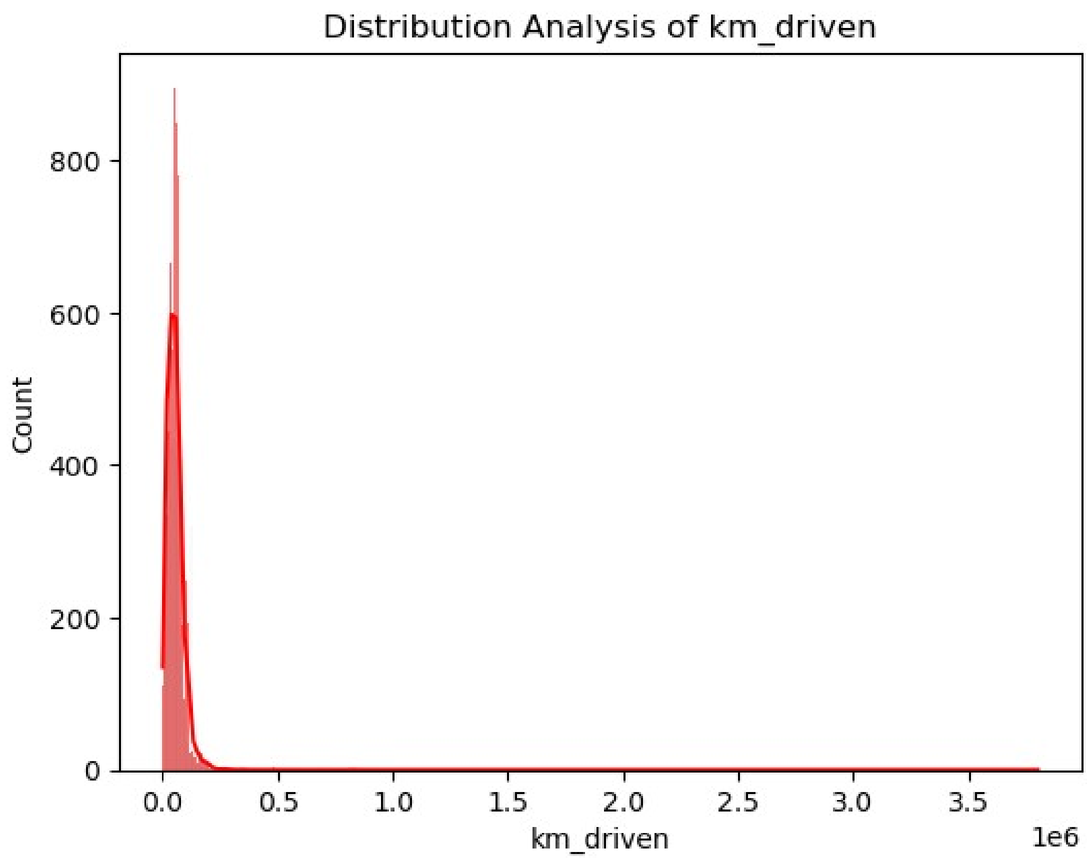
Total Count by seller_type
10000
8000
6000
4000
2000
0
9539
5699
173
Total Counts ------------------>
Dealer
Individual
Trustmark Dealer
seller_type--------------->

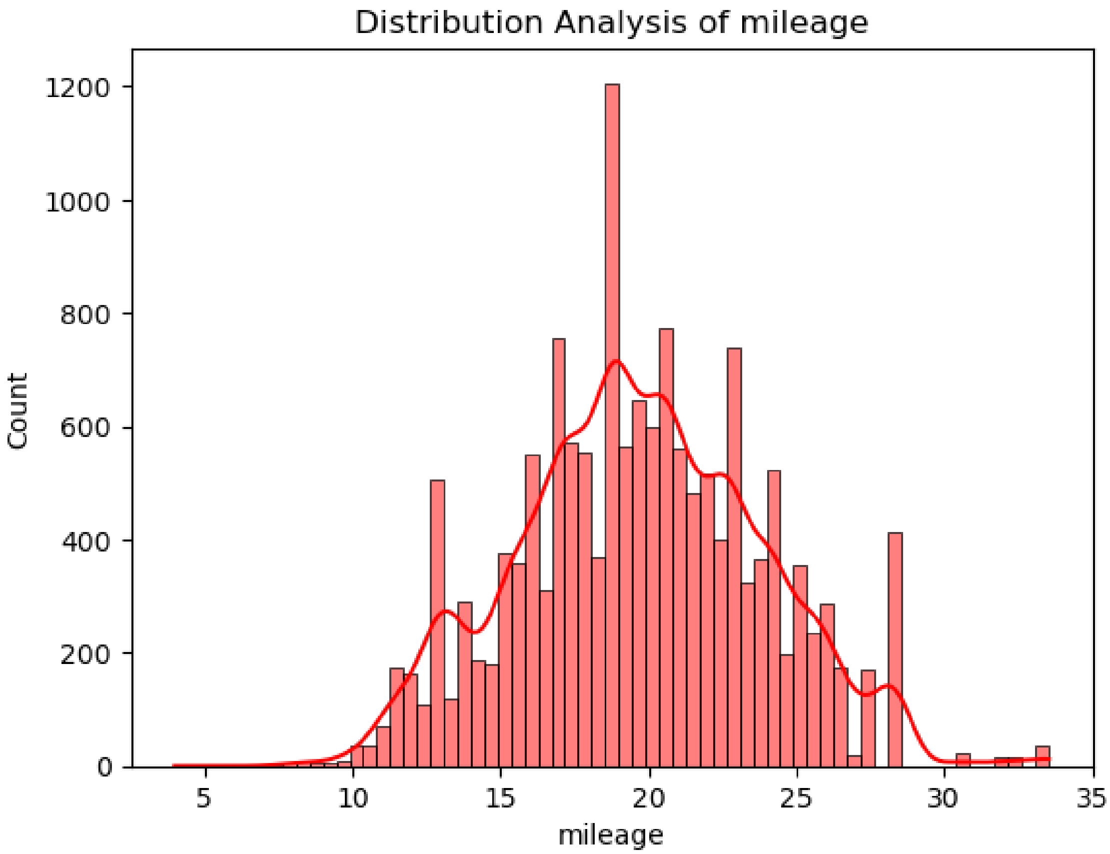

Total Count by fuel_type
7643
7419
301
44
4
8000
7000
6000
5000
4000
3000
2000
1000
0
Total Counts ------------------>
Petrol
Diesel
CNG
LPG
Electric
fuel_type-------------->

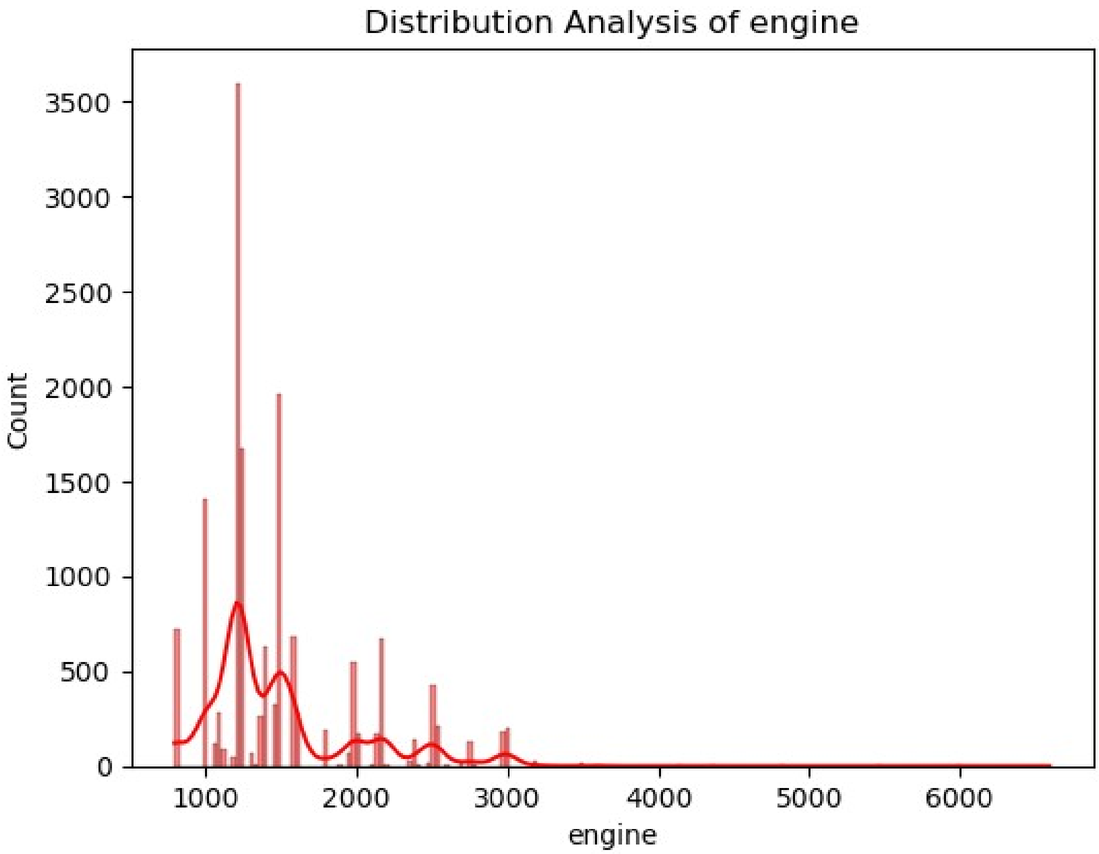


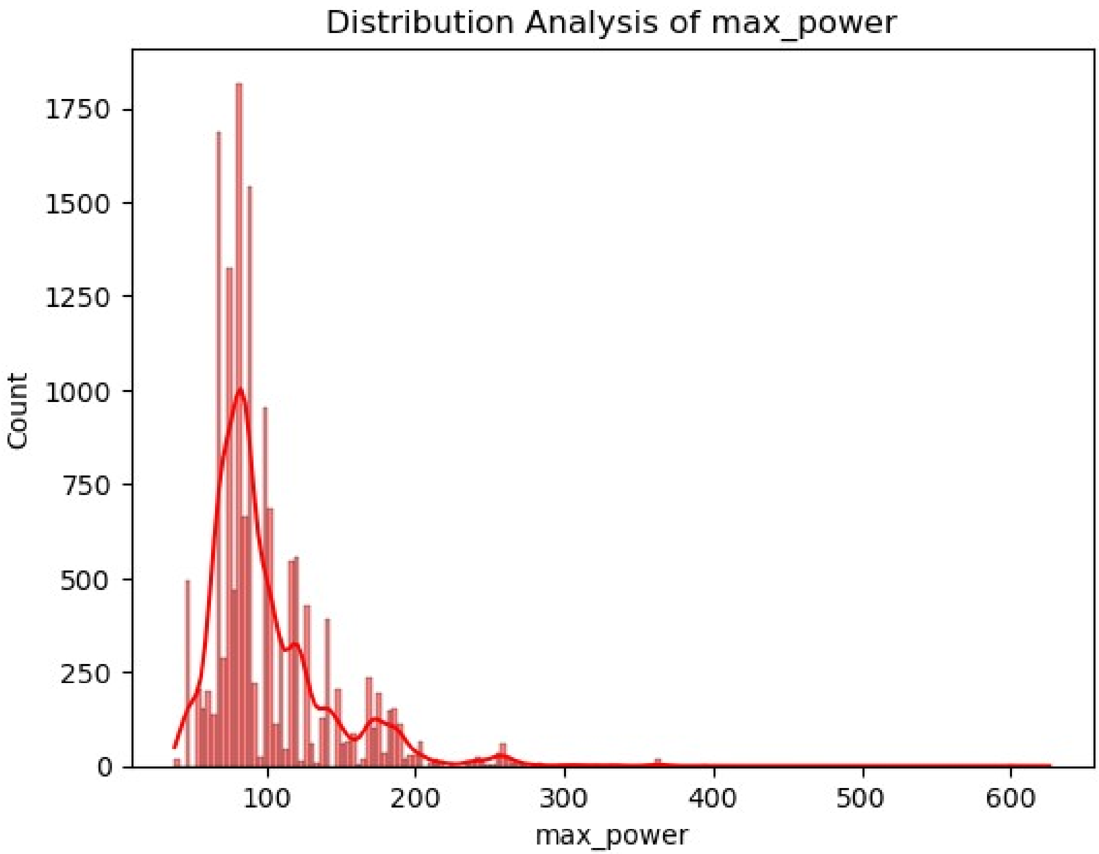

```
#  3.3
for i in num_col:
    plt.title(f'Distribution Analysis of {i}')
    sns.histplot(data = df1, x = i , color = 'r' , kde = True)
    plt.show()
```

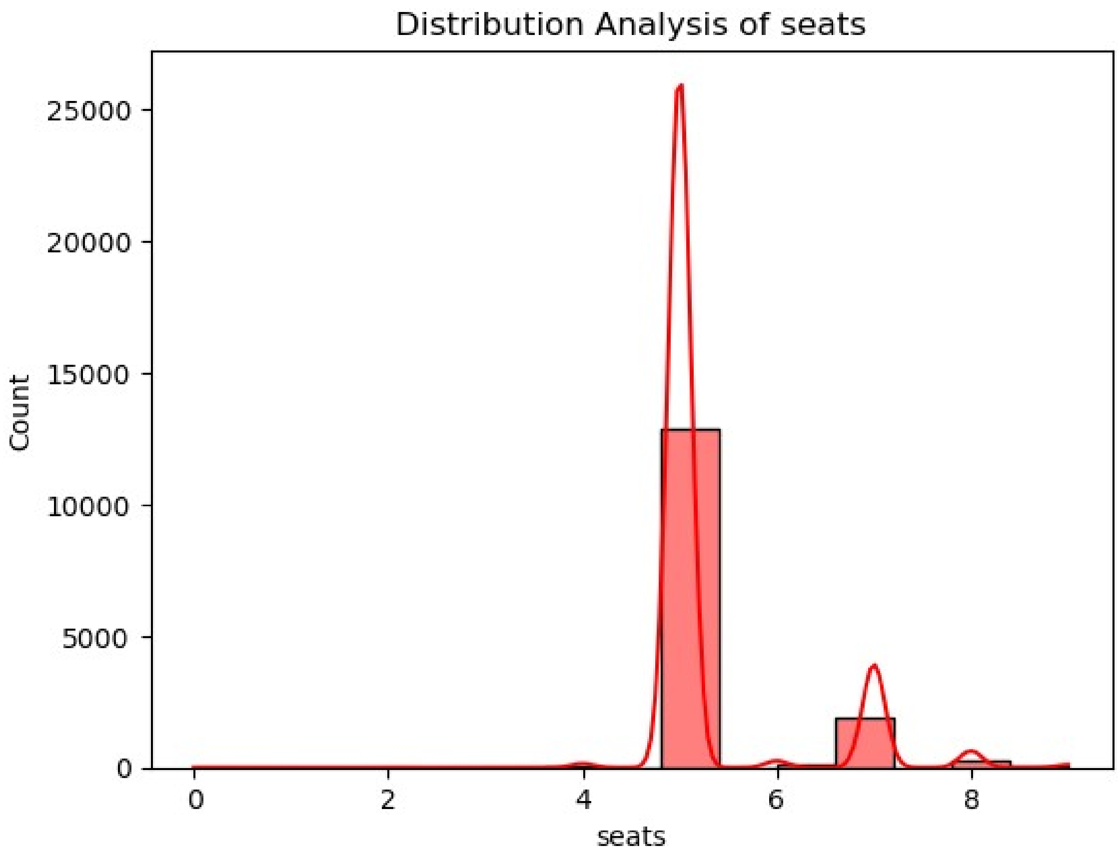

Distribution Analysis of vehicle_age
Count
0
500
1000
1500
2000
0
5
10
15
20
25
30
vehicle_age


Distribution Analysis of km_driven
Count
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
km_driven
1e6

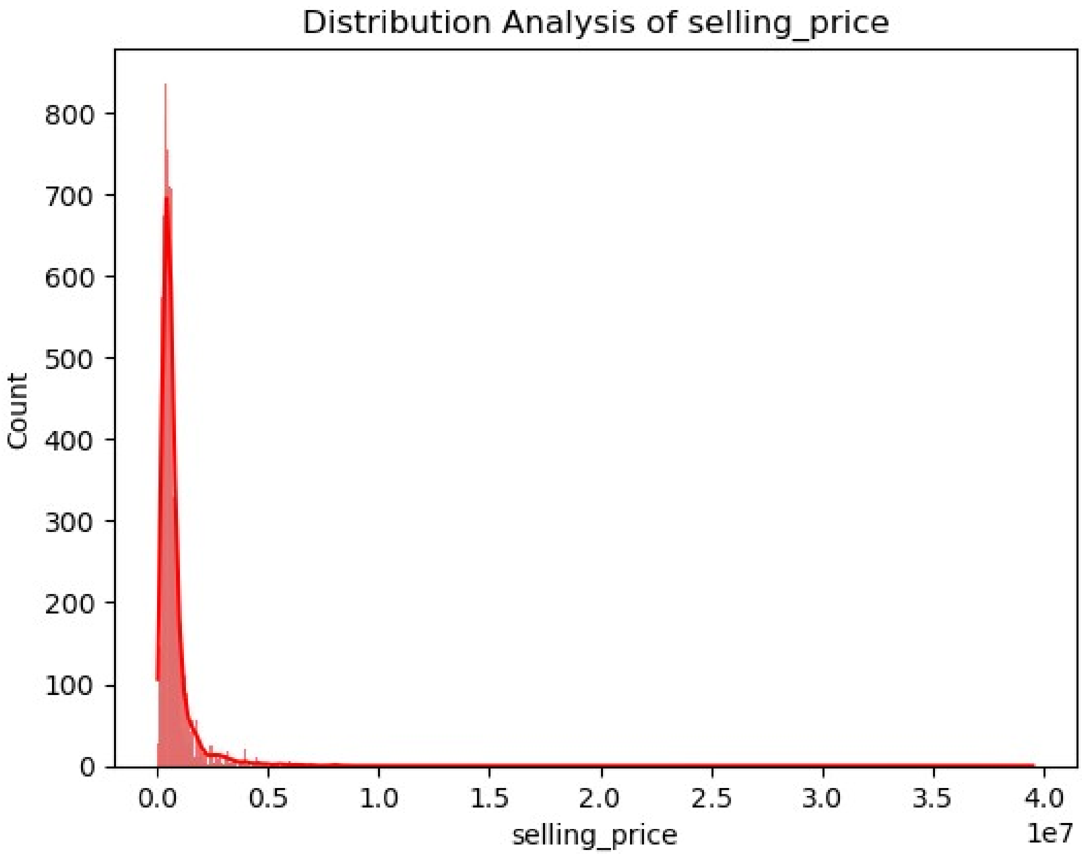

Distribution Analysis of mileage
Count
mileage
0
200
400
600
800
1000
1200
5
10
15
20
25
30
35


Distribution Analysis of engine
Count
3500
3000
2500
2000
1500
1000
500
0
1000
2000
3000
4000
5000
6000
engine


Distribution Analysis of max_power
Count
1750
1500
1250
1000
750
500
250
0
100
200
300
400
500
600
max_power



Distribution Analysis of seats
Count
25000
20000
15000
10000
5000
0
0
2
4
6
8
seats




```
# Which one is the most cheapest car ??

df1[df1['selling_price']   == df1['selling_price'].min()]

            car_name   brand    model  vehicle_age  km_driven
seller_type  \
7607  Maruti Wagon R  Maruti  Wagon R           21      80000
Individual

     fuel_type transmission_type  mileage  engine  max_power  seats  \
7607    Petrol            Manual     18.9     998       67.1      5

      selling_price
7607          40000

#  Which one is the most expensive car ??

df[df1['selling_price'] == df1['selling_price'].max()]

      Unnamed: 0           car_name    brand      model
vehicle_age  \
3799        4845  Ferrari GTC4Lusso  Ferrari  GTC4Lusso            2
```

```
      km_driven seller_type fuel_type transmission_type  mileage
engine  \
3799       3800      Dealer    Petrol         Automatic      4.0
3855

      max_power  seats  selling_price
3799      601.0      4       39500000
```

```
# Top 10 most expensive car ?

df1.sort_values('selling_price',ascending=False).head(10)
```

```
                   car_name          brand        model  vehicle_age
\
3799       Ferrari GTC4Lusso        Ferrari    GTC4Lusso            2

10969      Rolls-Royce Ghost    Rolls-Royce        Ghost            4

1172     Bentley Continental        Bentley  Continental            9

9722   Mercedes-Benz S-Class  Mercedes-Benz      S-Class            3

9364         Porsche Cayenne        Porsche      Cayenne            4

10989  Mercedes-Benz S-Class  Mercedes-Benz      S-Class            2

1888   Mercedes-Benz S-Class  Mercedes-Benz      S-Class            5

11000        Land Rover Rover     Land Rover        Rover            4

8439                   BMW 7            BMW            7            3

3096                   BMW 7            BMW            7            3


       km_driven seller_type fuel_type transmission_type  mileage
engine  \
3799        3800      Dealer    Petrol         Automatic     4.00
3855
10969       5000  Individual    Petrol         Automatic    10.20
6592
1172        9000      Dealer    Petrol         Automatic     9.50
5998
9722        4000      Dealer    Petrol         Automatic     7.81
4663
9364       24000      Dealer    Petrol         Automatic    12.50
3604
10989      18000      Dealer    Petrol         Automatic     7.81
2996
1888       41000      Dealer    Petrol         Automatic     7.81
5461
```

```
11000        9500      Dealer     Diesel         Automatic    12.65
2993
8439        19000      Dealer     Diesel         Automatic    17.66
2993
3096        50000  Individual     Diesel         Automatic    16.77
2993

       max_power  seats  selling_price
3799      601.00      4       39500000
10969     563.00      4       24200000
1172      626.00      4       14500000
9722      459.00      4       13000000
9364      440.00      5       11100000
10989     362.07      5       11000000
1888      362.90      5       11000000
11000     296.00      5        9200000
8439      355.37      4        8500000
3096      261.49      5        8500000
```

```
# Top 20 most expensive car count ?

df1.sort_values(['selling_price'],ascending=False).head(20)
['brand'].value_counts()
```

```
brand
BMW              6
Mercedes-Benz    5
Bentley          2
Volvo            2
Ferrari          1
Rolls-Royce      1
Porsche          1
Land Rover       1
Lexus            1
Name: count, dtype: int64
```

```
#3.4 Top 10 least expensive car ?
df1.sort_values(['selling_price']).head(10)
```

```
             car_name    brand    model  vehicle_age  km_driven
seller_type  \
7607   Maruti Wagon R   Maruti  Wagon R           21      80000
Individual
13676     Maruti Alto   Maruti     Alto           17     110000
Individual
12298  Maruti Wagon R   Maruti  Wagon R           17      50000
Individual
3787      Maruti Alto   Maruti     Alto           19     120000
Individual
7361       Honda City    Honda     City           19     110000
```

```
Individual
7930   Maruti Wagon R   Maruti  Wagon R           15      45000
Dealer
11190   Maruti Baleno   Maruti   Baleno           15      60500
Dealer
9045   Hyundai Santro  Hyundai   Santro           14      50000
Individual
2596      Maruti Alto   Maruti     Alto           29      22612
Dealer
2966      Maruti Alto   Maruti     Alto           15      80000
Individual

      fuel_type transmission_type  mileage  engine  max_power
seats  \
7607     Petrol            Manual    18.90     998      67.10      5

13676    Petrol            Manual    19.70     796      46.30      5

12298    Petrol            Manual    18.90     998      67.10      5

3787     Petrol            Manual    18.90    1061      47.00      5

7361     Petrol            Manual    13.00    1493     100.00      5

7930     Petrol            Manual    21.79     998      67.05      5

11190    Petrol            Manual    15.40    1590      94.00      5

9045     Petrol            Manual    17.80    1086      63.00      5

2596     Petrol            Manual    22.05     796      47.30      5

2966     Petrol            Manual    19.70     796      46.30      5


       selling_price
7607           40000
13676          45000
12298          50000
3787           50000
7361           50000
7930           55000
11190          60000
9045           60000
2596           60000
2966           60000
```

```
#3.5 Top 20 lest expensive car count ?

df1.sort_values('selling_price').head(20)['brand'].value_counts()
```

```
brand
Maruti     15
Honda       3
Hyundai     1
Tata        1
Name: count, dtype: int64
```

```
#3.6 Give the suggestion of list of cars according Customer's
requirements??
# budget => 10-15 lakh ,transmission type => manual , year = 3-5 ,
Brand => Mahindra


customer_df = df1[(df1['selling_price'] <=1500000) &
(df1['transmission_type'] == 'Manual') & (df1['vehicle_age'] <= 5) &
(df1['brand'] == 'Mahindra')]
customer_df
```

```
               car_name     brand    model  vehicle_age  km_driven  \
40      Mahindra Bolero  Mahindra   Bolero            1      40000
41      Mahindra KUV100  Mahindra   KUV100            3      17000
54     Mahindra Scorpio  Mahindra  Scorpio            4      50000
56     Mahindra Marazzo  Mahindra  Marazzo            2      36000
75     Mahindra Scorpio  Mahindra  Scorpio            5      44000
...                 ...       ...      ...          ...        ...
15224     Mahindra Thar  Mahindra     Thar            2      12551
15227      Mahindra KUV  Mahindra      KUV            3      50000
15313   Mahindra XUV500  Mahindra   XUV500            2      15000
15381     Mahindra Thar  Mahindra     Thar            4      43000
15409   Mahindra XUV500  Mahindra   XUV500            5    3800000

      seller_type fuel_type transmission_type  mileage  engine
max_power  \
40     Individual    Diesel            Manual    21.00    1498
74.96
41     Individual    Petrol            Manual    18.15    1198
82.00
54     Individual    Diesel            Manual    15.40    1997
120.00
56     Individual    Diesel            Manual    17.30    1497
121.00
75         Dealer    Diesel            Manual    15.40    2179
120.00
...           ...       ...               ...      ...     ...
...
15224      Dealer    Diesel            Manual    16.55    2498
105.00
15227  Individual       CNG            Manual    18.15    1198
82.00
15313  Individual    Diesel            Manual    15.10    2179
```

```
152.87
15381      Dealer    Diesel            Manual    16.55    2498
105.00
15409      Dealer    Diesel            Manual    16.00    2179
140.00

       seats  selling_price
40         7         850000
41         5         550000
54         7        1150000
56         7         990000
75         7        1050000
...      ...            ...
15224      6        1025000
15227      6         400000
15313      7        1250000
15381      6         795000
15409      7        1225000

[404 rows x 13 columns]

#3.7 Sort the customer's_df in ascending order according Vehicle age &
selling price
customer_df.sort_values(by = ['vehicle_age','selling_price'])

               car_name     brand    model  vehicle_age  km_driven  \
7789       Mahindra KUV  Mahindra      KUV            0      30000
3693    Mahindra Bolero  Mahindra   Bolero            1      10000
8434    Mahindra Bolero  Mahindra   Bolero            1      15000
2896   Mahindra Scorpio  Mahindra  Scorpio            1      15000
9039    Mahindra KUV100  Mahindra   KUV100            1      35000
...                 ...       ...      ...          ...        ...
4325   Mahindra Scorpio  Mahindra  Scorpio            5      65273
12925   Mahindra XUV500  Mahindra   XUV500            5      72000
6454   Mahindra Scorpio  Mahindra  Scorpio            5      86613
5704    Mahindra XUV500  Mahindra   XUV500            5     129615
3780    Mahindra XUV500  Mahindra   XUV500            5      42485

            seller_type fuel_type transmission_type  mileage
engine  \
7789         Individual    Petrol            Manual    18.15    1198

3693         Individual    Diesel            Manual    21.00    1498

8434         Individual    Diesel            Manual    21.00    1498

2896         Individual    Diesel            Manual    15.40    2523

9039         Individual    Diesel            Manual    25.32    1198
```

```
...                   ...       ...                 ...      ...     ...

4325   Trustmark Dealer    Diesel              Manual    15.40    2179

12925            Dealer    Diesel              Manual    16.00    2179

6454             Dealer    Diesel              Manual    15.40    2179

5704             Dealer    Diesel              Manual    16.00    2179

3780             Dealer    Diesel              Manual    16.00    2179


       max_power  seats  selling_price
7789       82.00      6         400000
3693       74.96      7         600000
8434       74.96      7         650000
2896       75.00      7         675000
9039       77.00      6         700000
...          ...    ...            ...
4325      120.00      7        1240000
12925     140.00      7        1245000
6454      120.00      7        1275000
5704      140.00      7        1300000
3780      140.00      7        1350000

[404 rows x 13 columns]
```

```python
customer_df.sort_values(by =
['vehicle_age','selling_price'],ascending=False)
```

```
               car_name     brand    model  vehicle_age  km_driven  \
3780    Mahindra XUV500  Mahindra   XUV500            5      42485
5704    Mahindra XUV500  Mahindra   XUV500            5     129615
6454   Mahindra Scorpio  Mahindra  Scorpio            5      86613
12925   Mahindra XUV500  Mahindra   XUV500            5      72000
4325   Mahindra Scorpio  Mahindra  Scorpio            5      65273
...                 ...       ...      ...          ...        ...
9039    Mahindra KUV100  Mahindra   KUV100            1      35000
2896   Mahindra Scorpio  Mahindra  Scorpio            1      15000
8434    Mahindra Bolero  Mahindra   Bolero            1      15000
3693    Mahindra Bolero  Mahindra   Bolero            1      10000
7789       Mahindra KUV  Mahindra      KUV            0      30000

            seller_type fuel_type transmission_type  mileage
engine  \
3780             Dealer    Diesel              Manual    16.00    2179

5704             Dealer    Diesel              Manual    16.00    2179

6454             Dealer    Diesel              Manual    15.40    2179
```

```
12925            Dealer    Diesel           Manual    16.00    2179

4325   Trustmark Dealer    Diesel           Manual    15.40    2179

...                 ...       ...              ...      ...     ...

9039         Individual    Diesel           Manual    25.32    1198

2896         Individual    Diesel           Manual    15.40    2523

8434         Individual    Diesel           Manual    21.00    1498

3693         Individual    Diesel           Manual    21.00    1498

7789         Individual    Petrol           Manual    18.15    1198


       max_power  seats  selling_price
3780      140.00      7        1350000
5704      140.00      7        1300000
6454      120.00      7        1275000
12925     140.00      7        1245000
4325      120.00      7        1240000
...          ...    ...            ...
9039       77.00      6         700000
2896       75.00      7         675000
8434       74.96      7         650000
3693       74.96      7         600000
7789       82.00      6         400000

[404 rows x 13 columns]
```

Step 4 : Bivariate Analysis

Bivariate Analysis means analyzing the relationship between two variables at the same time.

- “Bi” = two, “variate” = variables.

Goal:

- Check how two columns are related
- Identify patterns, correlations, or differences

Types of Bivariate Analysis

- 1Numerical vs Numerical ==> vehicle age , km_driven

- **2** Categorical vs Numerical ==> brand , vehicle age
- **3** Categorical vs Categorical ==> brand , fuel_type

```python
# 4.1 categorical vs numerical

# give me minimum , maximum , average selling price of each brand car

df1.groupby('brand')
['selling_price'].agg(['min','max','mean']).round(2).reset_index()
```

	brand	min	max	mean
0	Audi	750000	6800000	1966864.58
1	BMW	465000	8500000	2693826.88
2	Bentley	5200000	14500000	9266666.67
3	Datsun	170000	650000	320517.65
4	Ferrari	39500000	39500000	39500000.00
5	Force	700000	700000	700000.00
6	Ford	130000	3200000	645224.05
7	Honda	50000	3200000	617756.90
8	Hyundai	60000	2600000	576153.92
9	ISUZU	1895000	1900000	1897500.00
10	Isuzu	1050000	2300000	1355000.00
11	Jaguar	1299000	6300000	2643033.90
12	Jeep	800000	5600000	1795804.88
13	Kia	1080000	3525000	1735250.00
14	Land Rover	1275000	9200000	3823901.96
15	Lexus	3990000	8000000	5146500.00
16	MG	1488000	2075000	1752947.37
17	Mahindra	100000	2950000	787455.00
18	Maruti	40000	1225000	487089.32
19	Maserati	6000000	6200000	6100000.00
20	Mercedes-AMG	5100000	5100000	5100000.00
21	Mercedes-Benz	315000	13000000	2480741.84
22	Mini	1290000	3875000	2182647.06
23	Nissan	440000	1450000	955363.64
24	Porsche	2000000	11100000	5161190.48
25	Renault	200000	1155000	440985.07
26	Rolls-Royce	24200000	24200000	24200000.00
27	Skoda	200000	3550000	784089.82
28	Tata	70000	1750000	683534.88
29	Toyota	265000	3650000	1371316.52
30	Volkswagen	173000	1250000	516546.77
31	Volvo	1200000	8195000	3729700.00

```python
# 4.2
def brand_sellingprice_details(brand):
    details = df1.groupby('brand')
['selling_price'].agg(['min','max','mean']).round().reset_index()
    return details[details['brand'] == brand]

brand_sellingprice_details('Audi')
```

```
  brand     min      max       mean
0  Audi  750000  6800000  1966865.0
```

```
#4.3 categorical vs numerical

# give me minimum , maximum , average vehicle_age of each brand car

def brand_vehicle_age_details(brand):
    details = df1.groupby('brand')
['vehicle_age'].agg(['min','max','mean']).round().reset_index()
    return details[details['brand']==brand]


brand_vehicle_age_details('Audi')
```

```
  brand  min  max  mean
0  Audi    1   12   7.0
```

```
df.sample()
```

```
      Unnamed: 0       car_name   brand   model  vehicle_age
km_driven  \
2354        3005  Toyota Innova  Toyota  Innova           10
118000

     seller_type fuel_type transmission_type  mileage  engine
max_power  \
2354      Dealer    Diesel            Manual     12.8    2494
102.0

      seats  selling_price
2354      8         721000
```

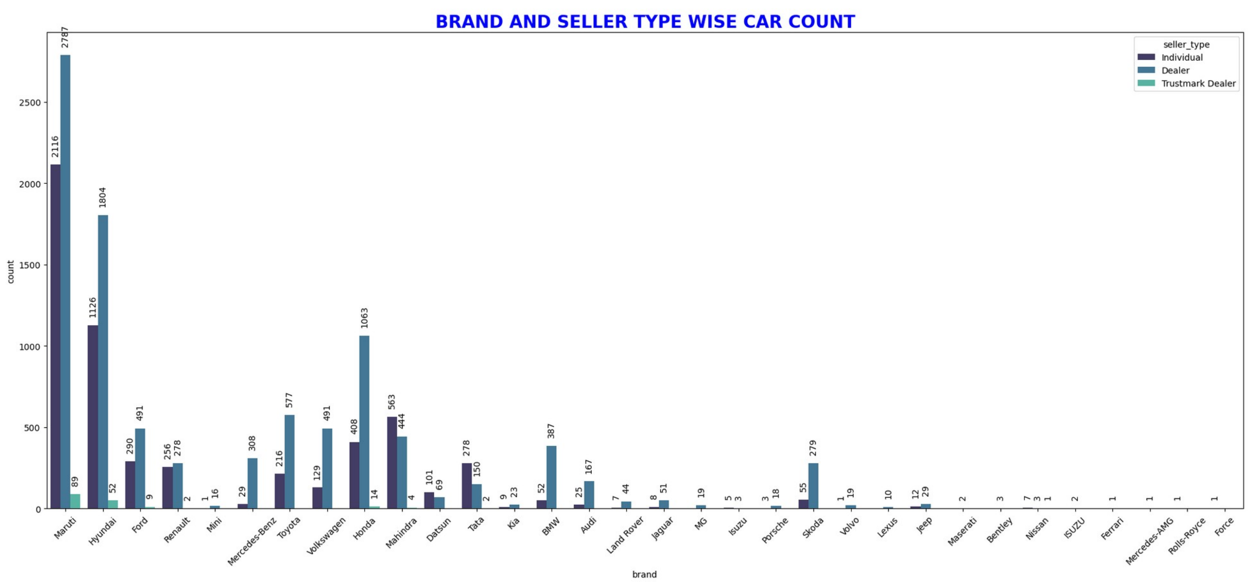
```
#4.4 give me minimum , maximum , average km_driven of each  car_name

def car_km_driven(car_name):
    details = df1.groupby('car_name')
['km_driven'].agg(['min','max','mean']).reset_index().round()
    return details[details['car_name'] == car_name]

car_km_driven('Maruti Ciaz')
```

```
       car_name   min     max     mean
68  Maruti Ciaz  1685  480000  49528.0
```

```
#4.4 categorical vs categorical
# give the all brand , seller type and count of cars

tempdf = df1.groupby(['brand','seller_type'])
['car_name'].count().reset_index()
tempdf.columns=['Brand','Seller Type','Car Count']
tempdf
```

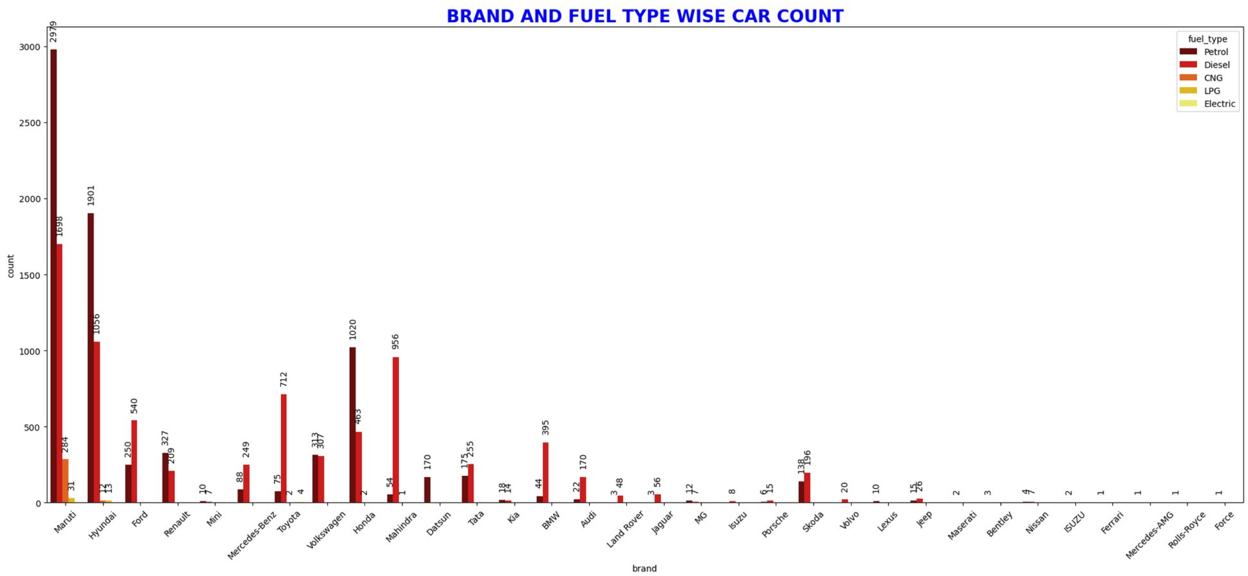
```
         Brand Seller Type  Car Count
0         Audi      Dealer        167
1         Audi  Individual         25
2          BMW      Dealer        387
3          BMW  Individual         52
4      Bentley      Dealer          3
..         ...         ...        ...
58      Toyota  Individual        216
59  Volkswagen      Dealer        491
60  Volkswagen  Individual        129
61       Volvo      Dealer         19
62       Volvo  Individual          1

[63 rows x 3 columns]
```

```python
#4.5 Create a chart that display Brand and Seller type wise car count

plt.figure(figsize=(25,10))
plt.title('BRAND AND SELLER TYPE WISE CAR COUNT',color = 'b',fontsize
= 20 , fontweight = 'bold')
chart = sns.countplot(data=df1, x = 'brand' ,hue ='seller_type' ,
palette=sns.color_palette('mako',3))
for i in chart.containers:
    plt.bar_label(i,rotation = 90 , padding = 10)
plt.xticks(rotation=45)
plt.show()
```

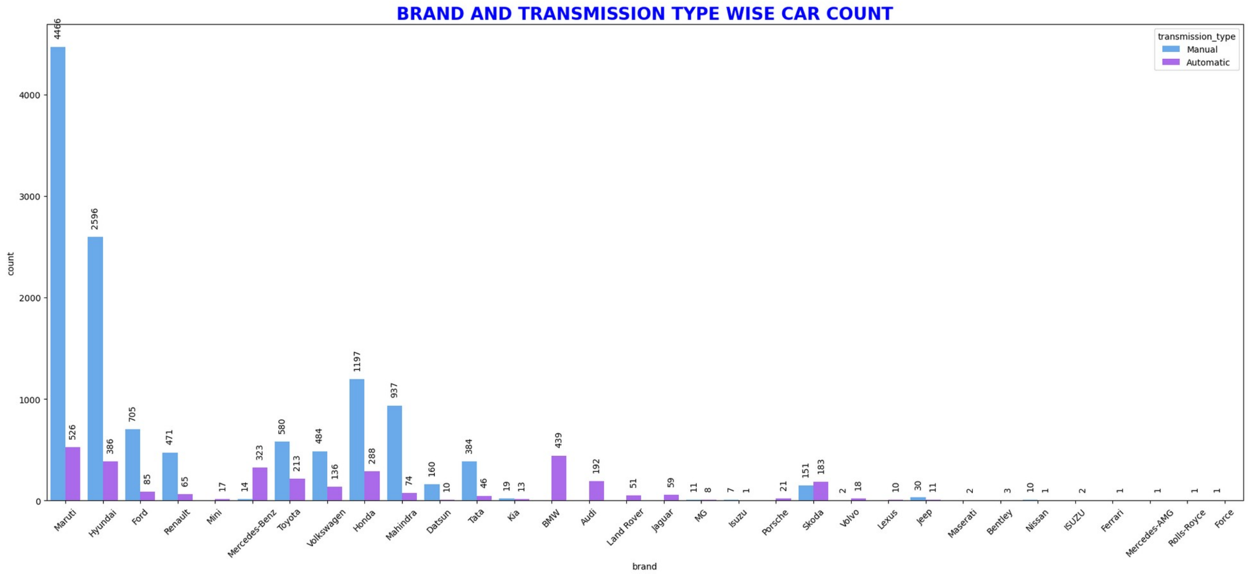



```python
#4.6 Create a chart that display Brand and fuel type wise car count

plt.figure(figsize=(25,10))
plt.title('BRAND AND FUEL TYPE WISE CAR COUNT',color = 'b',fontsize =
20 , fontweight = 'bold')
```

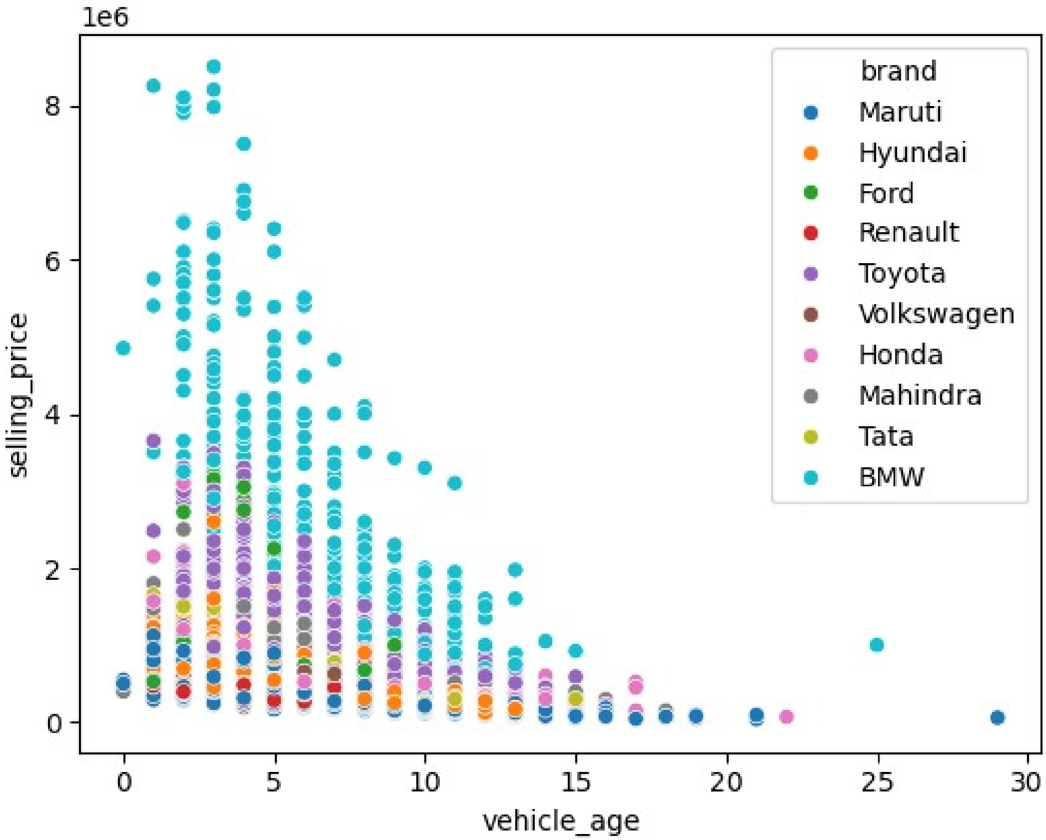
```
chart = sns.countplot(data=df1, x = 'brand' ,hue ='fuel_type' ,
palette=sns.color_palette('hot',5))
for i in chart.containers:
    plt.bar_label(i,rotation = 90 , padding = 10)
plt.xticks(rotation=45)
plt.show()
```



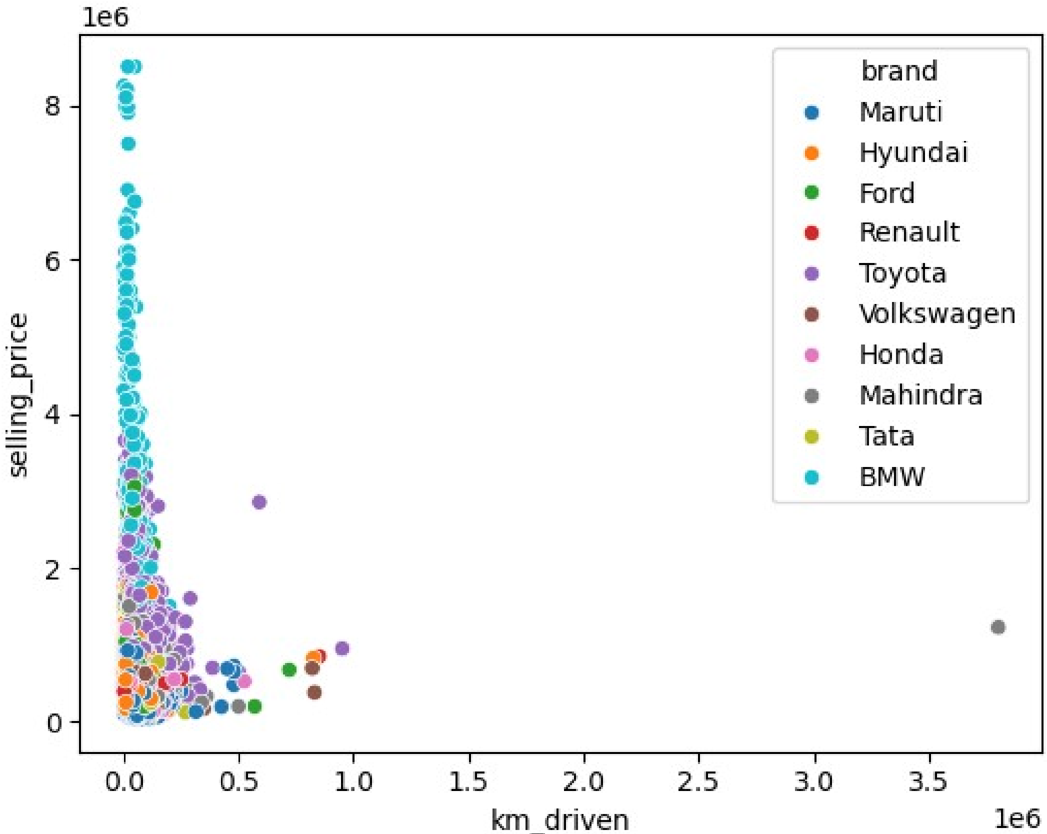

```
#4.7 Create a chart that display Brand and transmisssion type wise car
count

plt.figure(figsize=(25,10))
plt.title('BRAND AND TRANSMISSION TYPE WISE CAR COUNT',color =
'b',fontsize = 20 , fontweight = 'bold')
chart = sns.countplot(data=df1, x = 'brand' ,hue
='transmission_type' , palette=sns.color_palette('cool',2))
for i in chart.containers:
    plt.bar_label(i,rotation = 90 , padding = 10)
plt.xticks(rotation=45)
plt.show()
```



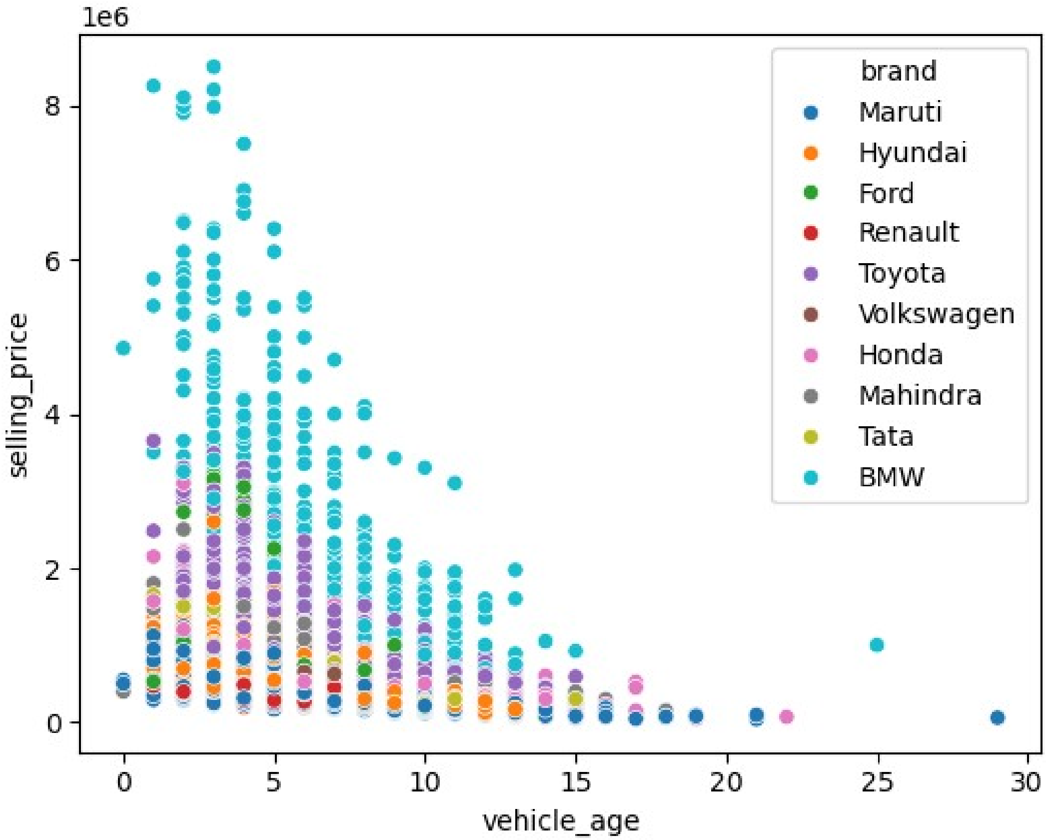

```
#4.8 CREATE A SCATTER PLOT OF TOP 10 BRAND

top_10_brand = df1['brand'].value_counts().head(10).index
top_10_car_df = df1[df1['brand'].isin(top_10_brand)]
sns.scatterplot(data = top_10_car_df, x
='vehicle_age',y='selling_price',hue='brand')
plt.show()
```



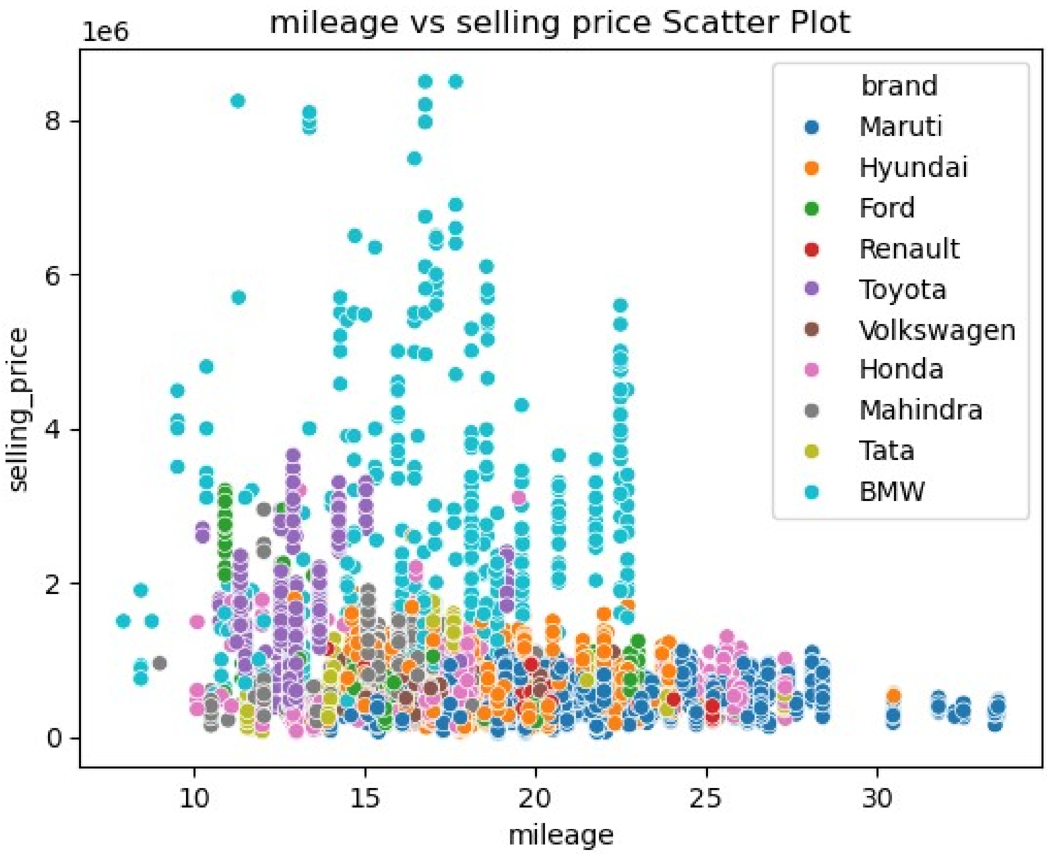

```
#4.9 CREATE A SCATTER PLOT OF TOP 10 BRAND

top_10_brand = df1['brand'].value_counts().head(10).index
top_10_car_df = df1[df1['brand'].isin(top_10_brand)]
sns.scatterplot(data = top_10_car_df, x
='km_driven',y='selling_price',hue='brand')
plt.show()
```

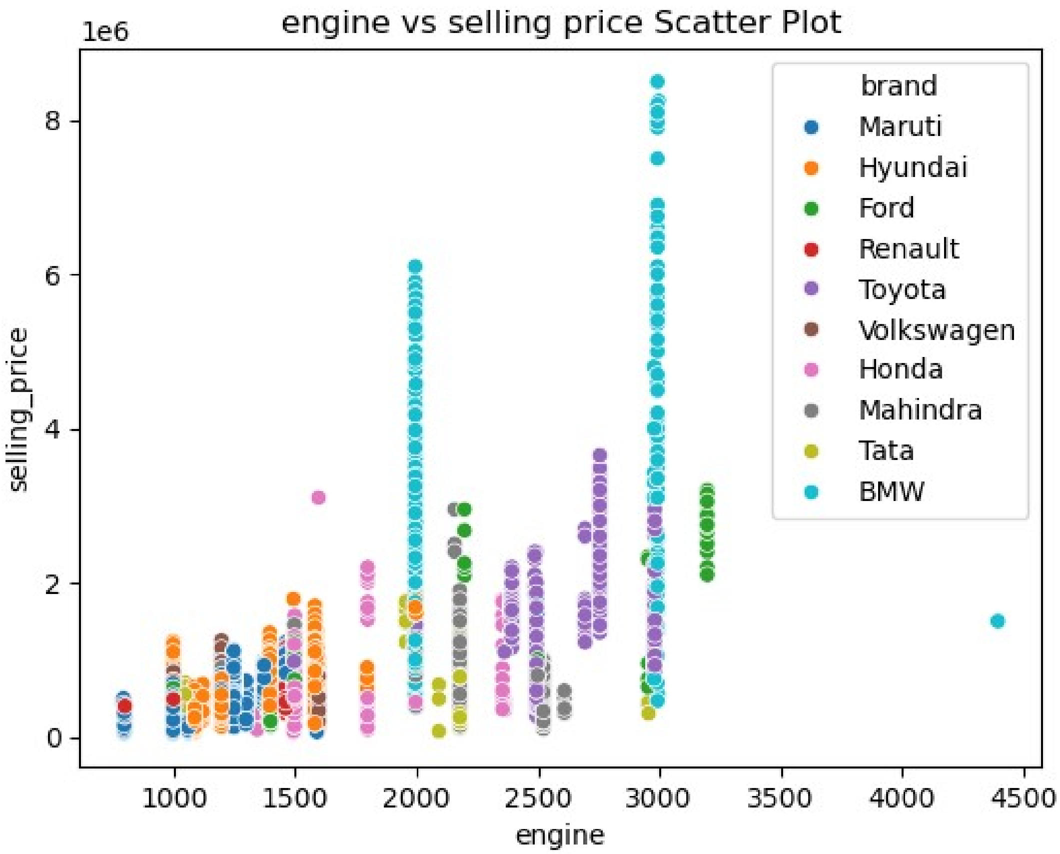



```
#4.10 CREATE A SCATTER PLOT OF TOP 10 BRAND

top_10_brand = df1['brand'].value_counts().head(10).index
top_10_car_df = df1[df1['brand'].isin(top_10_brand)]
sns.scatterplot(data = top_10_car_df, x
='vehicle_age',y='selling_price',hue='brand')
plt.show()
```

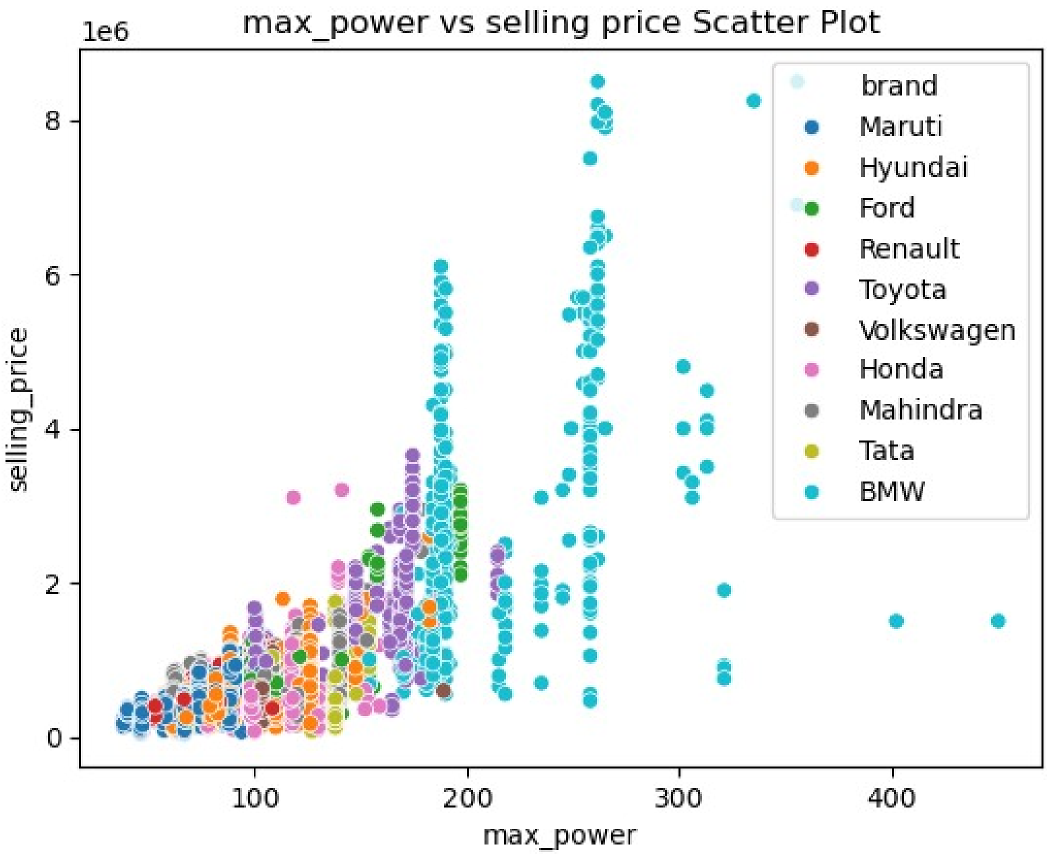


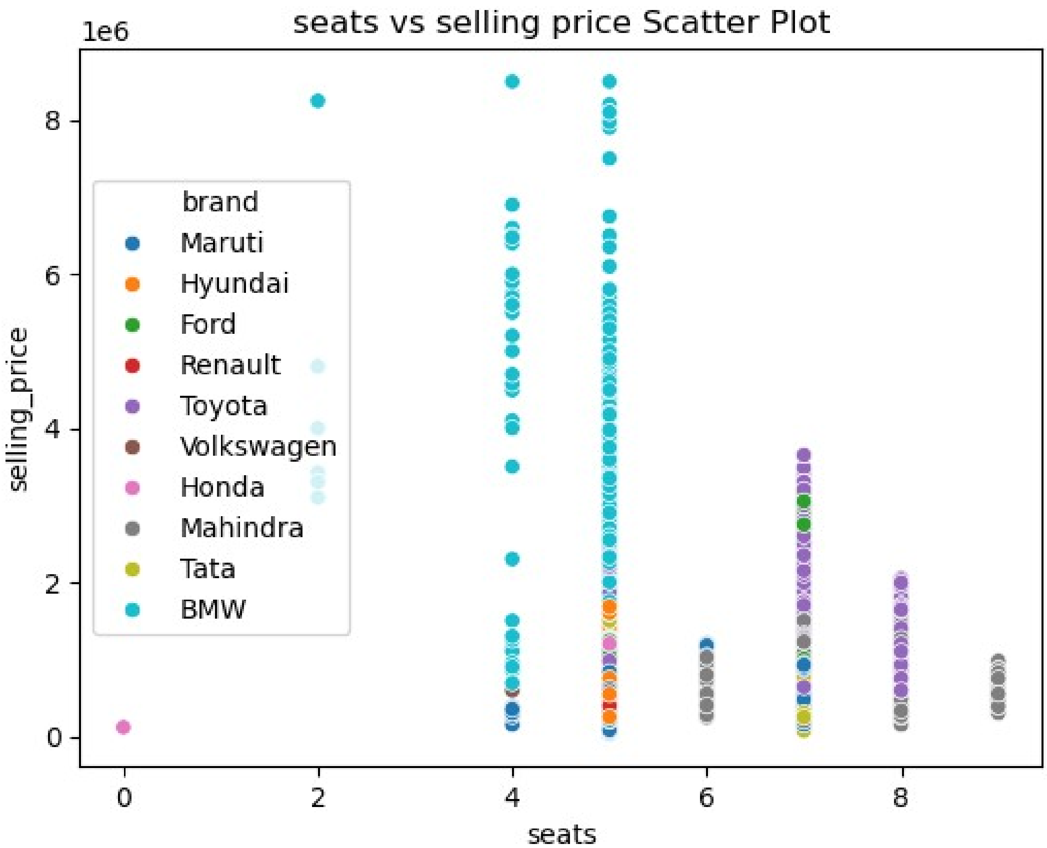

```
#4.11

imp_num_cols = df1.columns[-5:-1]
for i in imp_num_cols:
    plt.title(f'{i} vs selling price Scatter Plot')

sns.scatterplot(data=top_10_car_df,x=i,y='selling_price',hue='brand')
    plt.show()
```


mileage vs selling price Scatter Plot
1e6
8
6
4
2
0
selling_price
10
15
20
25
30
mileage
brand
Maruti
Hyundai
Ford
Renault
Toyota
Volkswagen
Honda
Mahindra
Tata
BMW



engine vs selling price Scatter Plot
1e6
selling_price
engine
brand
Maruti
Hyundai
Ford
Renault
Toyota
Volkswagen
Honda
Mahindra
Tata
BMW
0
2
4
6
8
1000
1500
2000
2500
3000
3500
4000
4500


max_power vs selling price Scatter Plot
1e6
selling_price
max_power
0
2
4
6
8
100
200
300
400
brand
Maruti
Hyundai
Ford
Renault
Toyota
Volkswagen
Honda
Mahindra
Tata
BMW




```
#4.12 list of cars which have 0 seats??
df1[df1['seats'] == 0]

            car_name   brand  model  vehicle_age  km_driven seller_type  \
3217     Honda City   Honda   City           18      40000  Individual

12619  Nissan Kicks  Nissan  Kicks            2      10000  Individual


      fuel_type transmission_type  mileage  engine  max_power  seats  \
3217     Petrol            Manual    13.00    1493     100.00      0

12619    Diesel            Manual    19.39    1461     108.49      0


       selling_price
3217          115000
12619        1154000

print('Done')

Done
```

Step 5 : Multivariate Analysis

Multivariate Analysis means analyzing three or more variables together to understand complex relationships.

- “Multi” = many variables

Types of Univariate Analysis

- 1 Numerical vs Numerical vs Numerical ==> vehicle age , km_driven , milege
- 2 Categorical vs Categorical vs Numerical ==> brand ,model , vehicle age
- 3 Categorical vs Categorical vs Categorical ==> brand , fuel_type , seller_type
- etc......

```
pd.set_option('display.max_rows',100)
df1.groupby(['brand','seller_type'])
['selling_price'].agg(['min','max','mean','count']).round(2).reset_ind
ex()
```

	brand	seller_type	min	max	mean	count
0	Audi	Dealer	750000	6800000	1994467.07	167
1	Audi	Individual	857000	4200000	1782480.00	25
2	BMW	Dealer	465000	8500000	2686912.14	387
3	BMW	Individual	550000	8500000	2745288.46	52
4	Bentley	Dealer	5200000	14500000	9266666.67	3
5	Datsun	Dealer	215000	530000	298739.13	69
6	Datsun	Individual	170000	650000	335396.04	101
7	Ferrari	Dealer	39500000	39500000	39500000.00	1
8	Force	Individual	700000	700000	700000.00	1
9	Ford	Dealer	139000	3200000	682539.71	491
10	Ford	Individual	130000	3200000	582051.72	290
11	Ford	Trustmark Dealer	525000	985000	645000.00	9
12	Honda	Dealer	90000	3200000	627442.14	1063
13	Honda	Individual	50000	3100000	591970.59	408

14	Honda	Trustmark Dealer	385000	885000	633857.14	14
15	Hyundai	Dealer	90000	1790000	576812.64	1804
16	Hyundai	Individual	60000	2600000	572446.71	1126
17	Hyundai	Trustmark Dealer	260000	1150000	633576.92	52
18	ISUZU	Dealer	1895000	1900000	1897500.00	2
19	Isuzu	Dealer	1290000	2300000	1680000.00	3
20	Isuzu	Individual	1050000	1250000	1160000.00	5
21	Jaguar	Dealer	1299000	5000000	2545862.75	51
22	Jaguar	Individual	1900000	6300000	3262500.00	8
23	Jeep	Dealer	800000	5600000	1857137.93	29
24	Jeep	Individual	1472000	2050000	1647583.33	12
25	Kia	Dealer	1140000	3525000	1846260.87	23
26	Kia	Individual	1080000	1750000	1451555.56	9
27	Land Rover	Dealer	1275000	9200000	3864068.18	44
28	Land Rover	Individual	1800000	6200000	3571428.57	7
29	Lexus	Dealer	3990000	8000000	5146500.00	10
30	MG	Dealer	1488000	2075000	1752947.37	19
31	Mahindra	Dealer	245000	2950000	854186.94	444
32	Mahindra	Individual	100000	2400000	732891.65	563
33	Mahindra	Trustmark Dealer	475000	1575000	1060000.00	4
34	Maruti	Dealer	55000	1225000	511404.74	2787
35	Maruti	Individual	40000	1100000	454393.14	2116
36	Maruti	Trustmark Dealer	210000	990000	503022.47	89
37	Maserati	Dealer	6000000	6200000	6100000.00	2
38	Mercedes-AMG	Dealer	5100000	5100000	5100000.00	

```
1
39  Mercedes-Benz            Dealer    315000  13000000   2483214.29
308
40  Mercedes-Benz        Individual    660000   5700000   2454482.76
29
41           Mini            Dealer   1290000   3875000   2119062.50
16
42           Mini        Individual   3200000   3200000   3200000.00
1
43         Nissan            Dealer    440000    890000    641666.67
3
44         Nissan        Individual    850000   1450000   1100571.43
7
45         Nissan  Trustmark Dealer    880000    880000    880000.00
1
46        Porsche            Dealer   2575000  11100000   5476944.44
18
47        Porsche        Individual   2000000   5000000   3266666.67
3
48        Renault            Dealer    225000   1155000    441482.01
278
49        Renault        Individual    200000   1150000    441761.72
256
50        Renault  Trustmark Dealer    265000    280000    272500.00
2
51    Rolls-Royce        Individual  24200000  24200000  24200000.00
1
52          Skoda            Dealer    200000   3550000    813749.10
279
53          Skoda        Individual    235000   2100000    633636.36
55
54           Tata            Dealer    105000   1750000    765700.00
150
55           Tata        Individual     70000   1750000    640917.27
278
56           Tata  Trustmark Dealer    440000    450000    445000.00
2
57         Toyota            Dealer    265000   3650000   1410672.44
577
58         Toyota        Individual    300000   3300000   1266185.19
216
59     Volkswagen            Dealer    199000   1250000    528796.33
491
60     Volkswagen        Individual    173000    975000    469922.48
129
61          Volvo            Dealer   1200000   8195000   3697052.63
19
62          Volvo        Individual   4350000   4350000   4350000.00
1
```

```
pd.set_option('display.max_rows',150)
df1.groupby(['brand','model'])
[['selling_price','km_driven']].agg(['min','max','count','mean']).rese
t_index().round(2)
```

```
            brand         model selling_price                 \
                                          min       max count   
mean  
0            Audi            A4        750000   4200000    99   
1566747.47  
1            Audi            A6        857000   4600000    64   
2076875.00  
2            Audi            A8       2200000   5500000     6   
3291666.67  
3            Audi            Q7       1000000   6800000    23   
3037391.30  
4             BMW             3        550000   4500000   152   
1786302.63  
5             BMW             5        550000   5800000   118   
2903847.46  
6             BMW             6       1500000   6500000    18   
5099333.33  
7             BMW             7        465000   8500000    37   
3782648.65  
8             BMW            X1        700000   3650000    54   
2064351.85  
9             BMW            X3       1275000   6100000    23   
2521521.74  
10            BMW            X4       4950000   6500000     6   
5570000.00  
11            BMW            X5       1375000   8100000    25   
4276320.00  
12            BMW            Z4       3100000   8250000     6   
4479000.00  
13        Bentley   Continental       5200000  14500000     3   
9266666.67  
14         Datsun            GO        180000    650000    85   
356752.94  
15         Datsun        RediGO        170000    425000    75   
279360.00  
16         Datsun       redi-GO        249000    435000    10   
321200.00  
17        Ferrari     GTC4Lusso      39500000  39500000     1   
39500000.00  
18          Force        Gurkha        700000    700000     1   
700000.00  
19           Ford        Aspire        340000    845000    65   
547307.69  
20           Ford      Ecosport        350000   1245000   374   
```

						706227.27
21	Ford	Endeavour	300000	3200000	47	2153872.34
22	Ford	Figo	130000	650000	271	316601.48
23	Ford	Freestyle	570000	880000	33	696727.27
24	Honda	Amaze	265000	935000	362	516743.09
25	Honda	CR	890000	1775000	3	1471666.67
26	Honda	CR-V	355000	3200000	28	1247678.57
27	Honda	City	50000	1570000	757	625428.01
28	Honda	Civic	89000	2200000	59	570627.12
29	Honda	Jazz	225000	1010000	175	583754.29
30	Honda	WR-V	548000	1010000	101	808762.38
31	Hyundai	Aura	900000	900000	1	900000.00
32	Hyundai	Creta	695000	1575000	336	1025970.24
33	Hyundai	Elantra	525000	1890000	50	998380.00
34	Hyundai	Grand	210000	800000	580	474451.72
35	Hyundai	Santro	60000	650000	139	295719.42
36	Hyundai	Tucson	1500000	2600000	10	1780900.00
37	Hyundai	Venue	835000	1240000	58	1034103.45
38	Hyundai	Verna	120000	1590000	492	653465.45
39	Hyundai	i10	100000	500000	410	279175.61
40	Hyundai	i20	150000	975000	906	543603.75
41	ISUZU	MUX	1895000	1900000	2	1897500.00
42	Isuzu	D-Max	1050000	1450000	7	1220000.00
43	Isuzu	MUX	2300000	2300000	1	2300000.00
44	Jaguar	F-PACE	4300000	6300000	3	5200000.00

45	Jaguar	XE	2700000	3800000	4	3162500.00
46	Jaguar	XF	1299000	4550000	52	2455557.69
47	Jeep	Compass	800000	2750000	39	1619307.69
48	Jeep	Wrangler	4875000	5600000	2	5237500.00
49	Kia	Carnival	3000000	3525000	4	3243750.00
50	Kia	Seltos	1080000	1950000	28	1519750.00
51	Land Rover	Rover	1275000	9200000	51	3823901.96
52	Lexus	ES	3990000	5375000	6	4394166.67
53	Lexus	NX	4500000	6400000	2	5450000.00
54	Lexus	RX	6200000	8000000	2	7100000.00
55	MG	Hector	1488000	2075000	19	1752947.37
56	Mahindra	Alturas	2400000	2950000	3	2616666.67
57	Mahindra	Bolero	100000	950000	211	524436.02
58	Mahindra	KUV	275000	729000	66	431272.73
59	Mahindra	KUV100	245000	786000	27	525777.78
60	Mahindra	Marazzo	600000	1476000	24	1093375.00
61	Mahindra	Scorpio	145000	1550000	273	781516.48
62	Mahindra	Thar	153000	1025000	62	776177.42
63	Mahindra	XUV300	800000	1190000	15	998600.00
64	Mahindra	XUV500	450000	1899000	330	1006830.30
65	Maruti	Alto	45000	485000	778	245452.44
66	Maruti	Baleno	60000	910000	364	646307.69
67	Maruti	Celerio	240000	595000	237	434050.63
68	Maruti	Ciaz	449000	1100000	346	715239.88
69	Maruti	Dzire LXI	385000	500000	2	

```
442500.00
70          Maruti     Dzire VXI       180000    790000    17
502588.24
71          Maruti     Dzire ZXI       440000    760000     4
550000.00
72          Maruti          Eeco       130000    490000   125
334872.00
73          Maruti        Ertiga       350000   1100000   343
719860.06
74          Maruti         Ignis       415000    700000    73
532602.74
75          Maruti      S-Presso       400000    550000    13
463230.77
76          Maruti         Swift       120000    875000   781
471736.24
77          Maruti   Swift Dzire       165000    925000   890
525888.76
78          Maruti        Vitara       525000   1225000   295
830596.61
79          Maruti       Wagon R        40000    625000   717
307390.34
80          Maruti           XL6      1000000   1200000     7
1113571.43
81        Maserati        Ghibli      6200000   6200000     1
6200000.00
82        Maserati  Quattroporte      6000000   6000000     1
6000000.00
83    Mercedes-AMG             C      5100000   5100000     1
5100000.00
84   Mercedes-Benz       C-Class       425000   4200000   118
1676550.85
85   Mercedes-Benz           CLS       700000   7500000     9
3433333.33
86   Mercedes-Benz       E-Class       315000   5743000   125
2021856.00
87   Mercedes-Benz      GL-Class      1690000   7595000    36
3929444.44
88   Mercedes-Benz           GLS      5375000   8000000    12
6781083.33
89   Mercedes-Benz       S-Class       625000  13000000    37
3559783.78
90            Mini        Cooper      1290000   3875000    17
2182647.06
91          Nissan         Kicks       850000   1450000     8
1046750.00
92          Nissan       X-Trail       440000   1100000     3
711666.67
93         Porsche       Cayenne      2000000  11100000    16
5077500.00
```

94	Porsche	Macan	5975000	5995000	2	5985000.00
95	Porsche	Panamera	3800000	6500000	3	5058333.33
96	Renault	Duster	295000	1155000	218	567389.91
97	Renault	KWID	200000	550000	306	342558.82
98	Renault	Triber	550000	800000	12	654500.00
99	Rolls-Royce	Ghost	24200000	24200000	1	24200000.00
100	Skoda	Octavia	200000	2375000	59	1247627.12
101	Skoda	Rapid	225000	1195000	182	565895.60
102	Skoda	Superb	235000	3550000	93	917021.51
103	Tata	Altroz	730000	730000	1	730000.00
104	Tata	Harrier	1228000	1750000	21	1618523.81
105	Tata	Hexa	800000	1600000	41	1284170.73
106	Tata	Nexon	550000	1030000	85	805364.71
107	Tata	Safari	70000	1270000	100	520840.00
108	Tata	Tiago	285000	665000	145	452048.28
109	Tata	Tigor	395000	740000	37	553054.05
110	Toyota	Camry	345000	2400000	36	1614277.78
111	Toyota	Fortuner	723000	3650000	187	1947529.41
112	Toyota	Glanza	747000	915000	8	829250.00
113	Toyota	Innova	265000	2350000	545	1176111.93
114	Toyota	Yaris	890000	1300000	17	1031588.24
115	Volkswagen	Polo	173000	975000	373	513222.52
116	Volkswagen	Vento	200000	1250000	247	521566.80
117	Volvo	S90	3650000	4750000	4	4187500.00
118	Volvo	XC	1200000	8195000	7	

```
4099285.71
119          Volvo          XC60       1400000   1825000      5
1645000.00
120          Volvo          XC90       4100000   6975000      4
5231000.00

    km_driven
          min       max count        mean
0       10950   130000    99   58898.28
1        3000   110000    64   53239.89
2       25000   131473     6   60245.50
3        4000   155000    23   71898.17
4        2000   158000   152   57019.20
5        2000   142500   118   47416.80
6        2300    65000    18   19451.22
7       16000    99900    37   51720.84
8       10000   210000    54   53345.24
9        8620   200000    23   64139.57
10       7099    14000     6    9658.17
11       7000    96714    25   52499.12
12       2000    33000     6   18003.67
13       9000    37500     3   25500.00
14       1041   450000    85   39978.71
15       1001    80000    75   26435.08
16       2300    50500    10   22978.70
17       3800     3800     1    3800.00
18      60000    60000     1   60000.00
19       5000   130000    65   47530.11
20       4000   720000   374   54960.41
21      11387   132000    47   58577.94
22       8073   570000   271   68859.37
23       3957    63001    33   25181.64
24       1800   208000   362   49416.57
25      34000    75000     3   54666.67
26      13868   155000    28   75066.57
27       2000   233000   757   57874.60
28       1900   180000    59   63943.31
29       4065   525000   175   46305.09
30       5000   127991   101   35687.76
31       4500     4500     1    4500.00
32       1470   825000   336   54923.95
33      10000   130600    50   57381.46
34       1000   198000   580   40627.75
35        100   174926   139   47708.72
36      12000   120000    10   59932.50
37        581    40000    58   16447.60
38       3700   225000   492   55995.37
39       5000   190000   410   57132.45
40       1493   220000   906   50608.13
```

41	47000	65029	2	56014.50
42	51000	120000	7	82161.71
43	34000	34000	1	34000.00
44	10000	30000	3	20666.67
45	15000	91795	4	39698.75
46	7600	110000	52	45926.02
47	4000	70000	39	33268.10
48	32000	40000	2	36000.00
49	3000	14000	4	8200.00
50	1200	25000	28	9406.21
51	4000	175000	51	67092.90
52	14000	34000	6	27666.67
53	7622	44000	2	25811.00
54	17272	52000	2	34636.00
55	1000	35803	19	11538.68
56	13000	36000	3	20677.67
57	2000	500000	211	87702.72
58	5000	100000	66	43022.74
59	5000	80000	27	31359.04
60	5000	100000	24	36112.79
61	6345	230000	273	80253.52
62	5000	84000	62	35555.90
63	2000	30000	15	15158.27
64	2200	3800000	330	80382.57
65	1200	425785	778	46883.66
66	1001	479000	364	35189.37
67	2000	110000	237	37323.81
68	1685	480000	346	49528.22
69	10000	50000	2	30000.00
70	1902	91200	17	34259.00
71	18767	66250	4	44009.50
72	1500	203125	125	42002.81
73	2100	197000	343	55612.45
74	2300	80000	73	24843.97
75	1000	12000	13	5430.77
76	1332	275000	781	59113.18
77	1000	250000	890	62227.83
78	1000	480000	295	49305.18
79	2000	315000	717	52994.70
80	4500	23000	7	12953.43
81	15000	15000	1	15000.00
82	9500	9500	1	9500.00
83	24000	24000	1	24000.00
84	6577	160000	118	55215.63
85	4800	103000	9	35644.44
86	1198	1325000	125	69517.71
87	4000	186900	36	59747.25
88	10000	71000	12	32416.67
89	4000	147500	37	52046.32

```
90        6000    70000    17   32210.71
91        4000    40000     8   13048.38
92       95000   110000     3  104000.00
93       24000   126000    16   67384.12
94       30350    34000     2   32175.00
95       25000    33000     3   27666.67
96        7131   850000   218   78018.49
97        1100   110000   306   27503.43
98        1784    35000    12   12182.75
99        5000     5000     1    5000.00
100       3700   300000    59   71198.54
101       8500   675000   182   65596.80
102       8000   160000    93   61654.89
103       3800     3800     1    3800.00
104       4000    68000    21   17940.48
105      10000   186000    41   52041.83
106       3500    95000    85   32526.15
107      30000   270000   100   92685.23
108       1000   110000   145   32020.25
109       5000    95000    37   37183.00
110      32000   138000    36   70570.81
111       1677   590000   187   90764.71
112       2000    25637     8   10517.12
113       6006   950000   545   96895.21
114       5000    41000    17   21583.18
115       7000   820000   373   58451.47
116       1500   830000   247   69110.28
117      18000    40000     4   25850.00
118       9000   122000     7   55642.00
119      70252   124000     5  101650.40
120      25500    85000     4   55875.00
```

```
def brand_model_analysis(brand):
    '''User must give the brand name
    ex:Audi'''
    temp_df = df1.groupby(['brand','model'])
['selling_price'].agg(['min','max','mean','count']).reset_index().roun
d(2)
    result_df = temp_df[temp_df['brand'] == brand]

    total_model_count = result_df['model'].count()
    total_no_of_cars = result_df['count'].sum()
    avg_price  = result_df['mean'].mean()
    cheapest_car_df = result_df[result_df['min'] ==
result_df['min'].min()]
    expensive_car_df= result_df[result_df['max'] ==
result_df['max'].max()]
    cheapest_car_name = cheapest_car_df['model'].values[0]
    expensive_car_name = expensive_car_df['model'].values[0]
```

```
    min_price_range,max_price_range =
cheapest_car_df['min'].min(),expensive_car_df['max'].max()

    print(f'''
    □{brand} car brand stats:
    □Total no of cars: {total_no_of_cars}
    ⚙Total Unique Models: {total_model_count}
    □Cheapest Model: {cheapest_car_name} Rs: {min_price_range}
    □Expensive Model: {expensive_car_name} Rs: {max_price_range }
    □On an Average Price of {brand} car\'s: {round(avg_price,2)}
    ''')
    return result_df


brand_model_analysis('BMW')
```

```
    □BMW car brand stats:
    □Total no of cars: 439
    ⚙Total Unique Models: 9
    □Cheapest Model: 7 Rs: 465000
    □Expensive Model: 7 Rs: 8500000
    □On an Average Price of BMW car's: 3609258.41
```

	brand	model	min	max	mean	count
4	BMW	3	550000	4500000	1786302.63	152
5	BMW	5	550000	5800000	2903847.46	118
6	BMW	6	1500000	6500000	5099333.33	18
7	BMW	7	465000	8500000	3782648.65	37
8	BMW	X1	700000	3650000	2064351.85	54
9	BMW	X3	1275000	6100000	2521521.74	23
10	BMW	X4	4950000	6500000	5570000.00	6
11	BMW	X5	1375000	8100000	4276320.00	25
12	BMW	Z4	3100000	8250000	4479000.00	6

```
brand_model_analysis('Maruti')
```

```
    □Maruti car brand stats:
    □Total no of cars: 4992
    ⚙Total Unique Models: 16
    □Cheapest Model: Wagon R Rs: 40000
    □Expensive Model: Vitara Rs: 1225000
    □On an Average Price of Maruti car's: 552242.99
```

	brand	model	min	max	mean	count
65	Maruti	Alto	45000	485000	245452.44	778
66	Maruti	Baleno	60000	910000	646307.69	364
67	Maruti	Celerio	240000	595000	434050.63	237

68	Maruti	Ciaz	449000	1100000	715239.88	346
69	Maruti	Dzire LXI	385000	500000	442500.00	2
70	Maruti	Dzire VXI	180000	790000	502588.24	17
71	Maruti	Dzire ZXI	440000	760000	550000.00	4
72	Maruti	Eeco	130000	490000	334872.00	125
73	Maruti	Ertiga	350000	1100000	719860.06	343
74	Maruti	Ignis	415000	700000	532602.74	73
75	Maruti	S-Presso	400000	550000	463230.77	13
76	Maruti	Swift	120000	875000	471736.24	781
77	Maruti	Swift Dzire	165000	925000	525888.76	890
78	Maruti	Vitara	525000	1225000	830596.61	295
79	Maruti	Wagon R	40000	625000	307390.34	717
80	Maruti	XL6	1000000	1200000	1113571.43	7

```
pivot_table_df = df1.pivot_table(values='selling_price',index =
'brand',columns = 'vehicle_age',aggfunc = 'mean').round()
```

```
pivot_table_df.to_html('brand vs age.html')
print('Done')
```

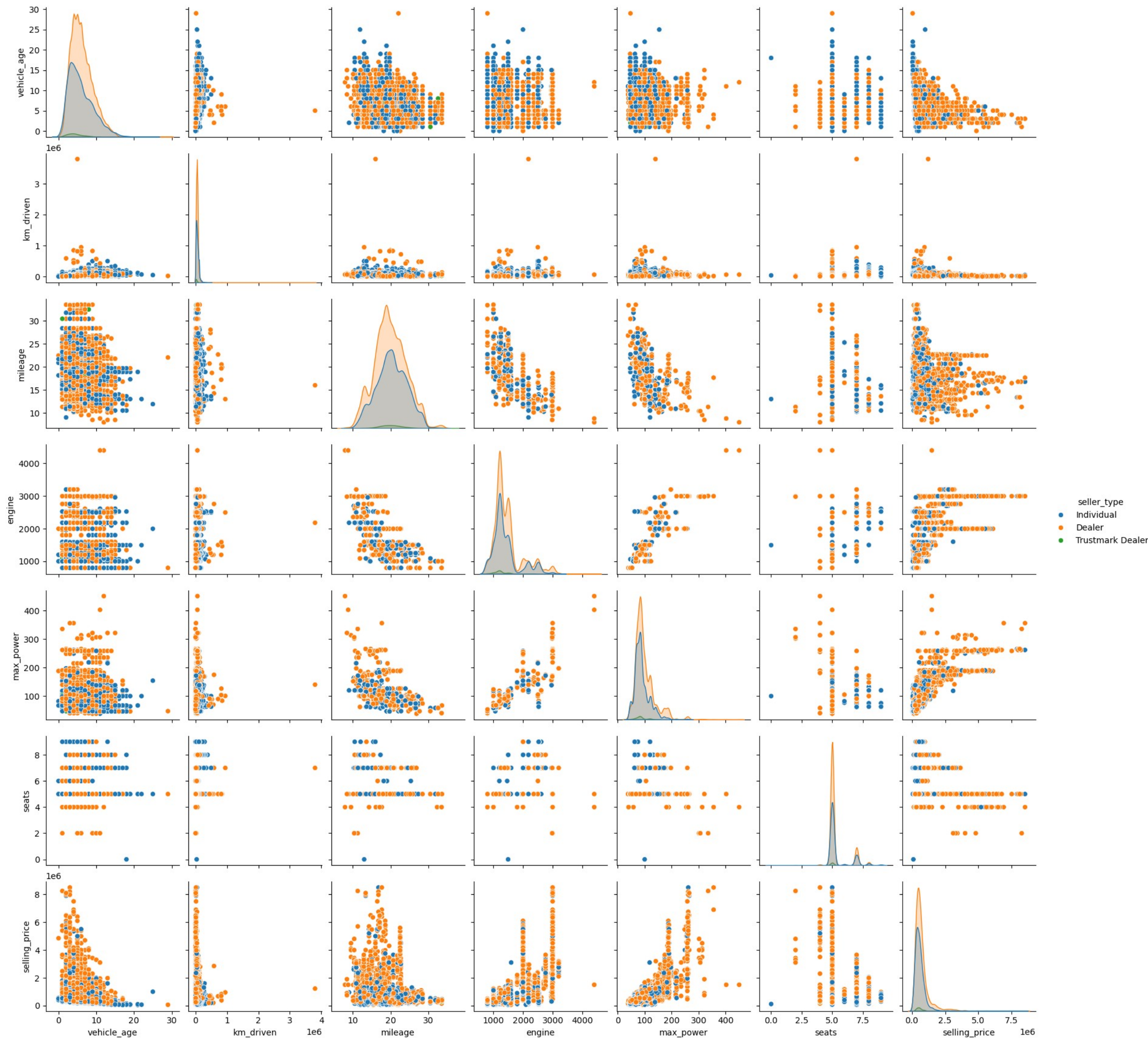
```
Done
```

```
pivot_table_df2 = df1.pivot_table(values='selling_price' , index =
['brand' , 'fuel_type'],columns = 'vehicle_age' , aggfunc =
'mean').round()
```

```
pivot_table_df2.to_html('brand vs fuel_type vs age.html')
print('done')
```

```
done
```

```
sns.pairplot(data = top_10_car_df , hue = 'seller_type')
plt.show()
```

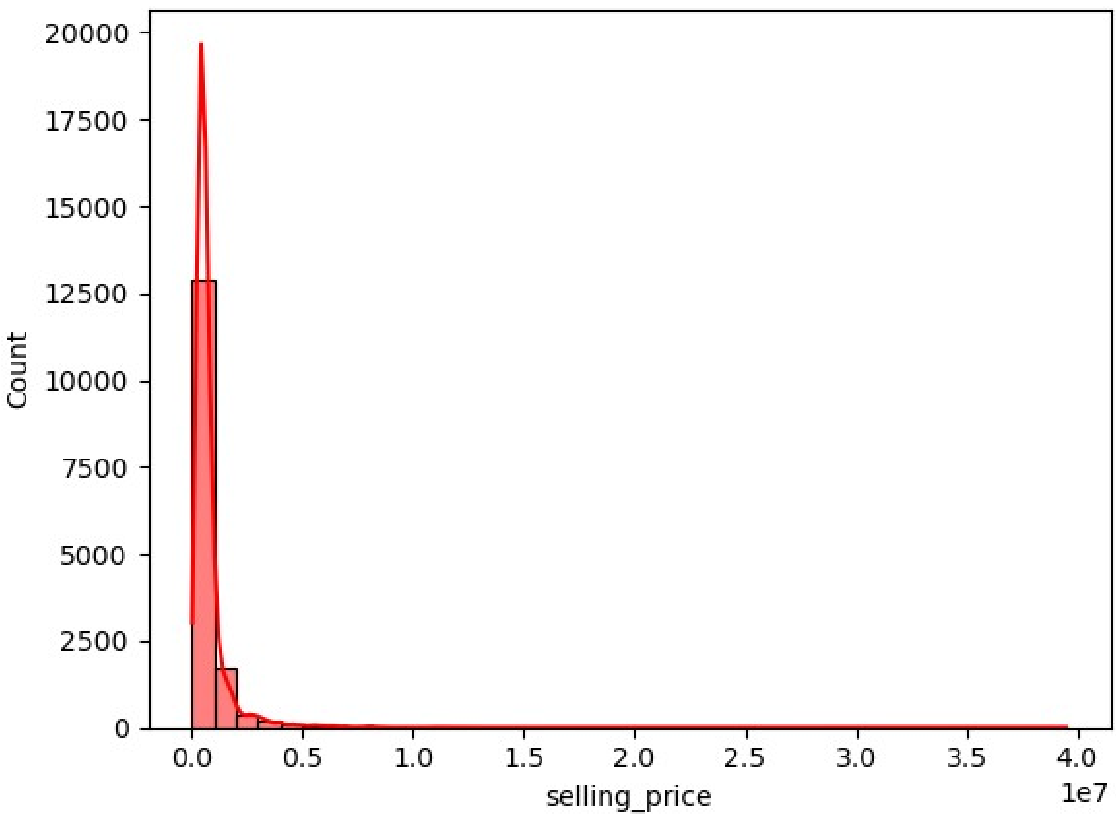


� 15 Python EDA Questions

-
  a. What is the distribution of selling prices across all car listings?
-
  a. How does the average selling price vary by car brand?
-
  a. What is the relationship between vehicle age and selling price?
-
  a. How does km_driven correlate with selling_price?
-
  a. What is the count of cars available by fuel_type (Petrol, Diesel,etc.)?
-

- a. What is the distribution of transmission_type among listed cars?
- a. Which seller_type (Dealer/Individual) lists the highest number of cars?
- a. How do mileage and max_power relate to each other across models?
- a. Are there any noticeable outliers in engine capacity or selling_price?
- a. Which car models have the highest average km_driven?
- a. How does selling_price vary across different transmission types?
- a. Which factors (vehicle_age, km_driven, engine, mileage,max_power) have the strongest correlation with selling_price?
- a. What is the distribution of seats available in the dataset?
- a. Which brand and model combinations offer the best mileage on average?
- a. Are there significant differences in selling_price between petrol and diesel vehicles?

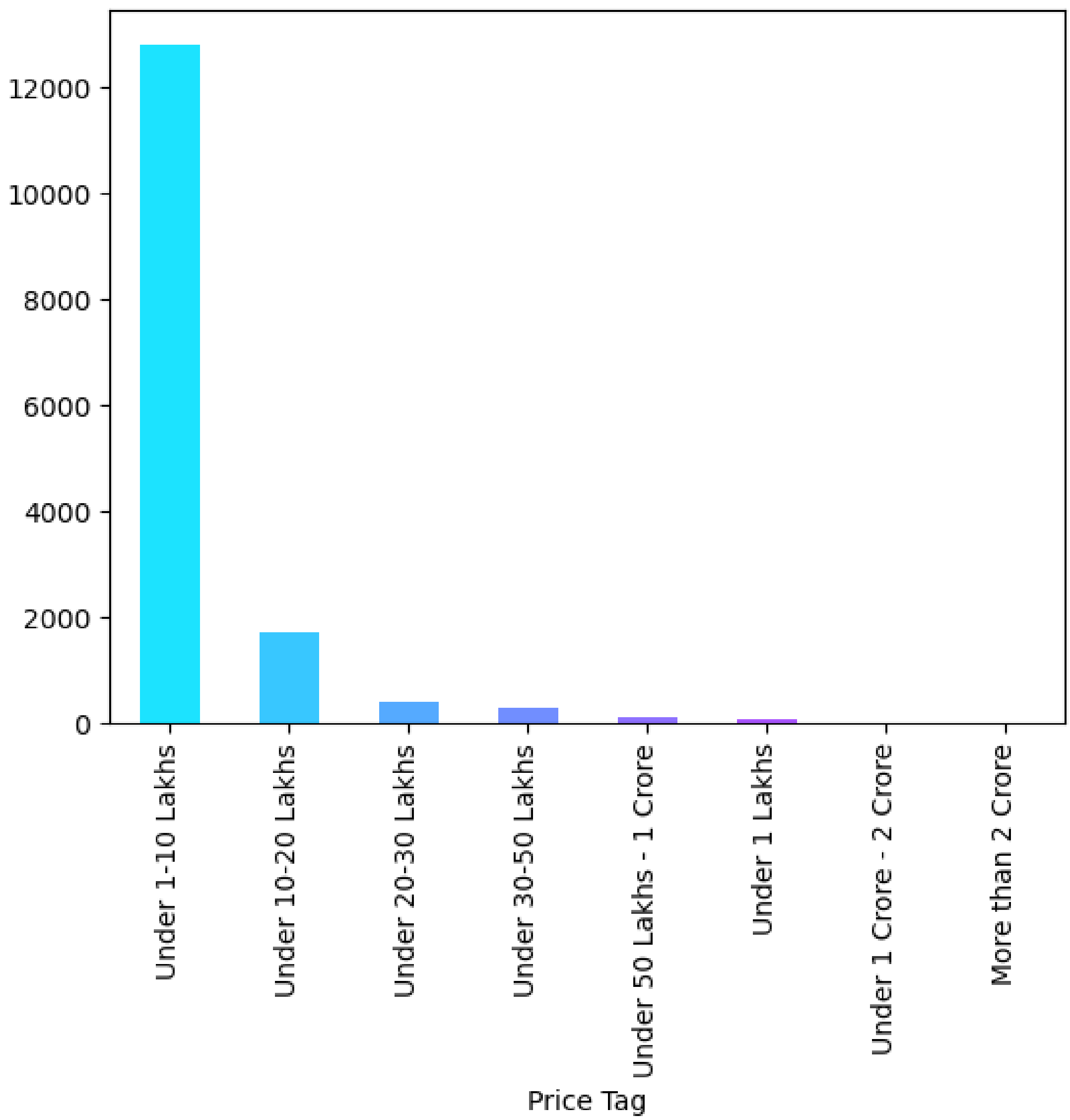
```
# 1. What is the distribution of selling prices across all car listings?

sns.histplot(df1['selling_price'],kde =True , color = 'r' , bins = range(10000,40000000,1000000))
plt.show()
```




```
price_tag = []
for i in df1['selling_price']:
    if i<=100000:
        price_tag.append('Under 1 Lakhs')
    elif i <= 1000000:
        price_tag.append('Under 1-10 Lakhs')
    elif i <= 2000000:
        price_tag.append('Under 10-20 Lakhs')
    elif i <= 3000000:
        price_tag.append('Under 20-30 Lakhs')
    elif i <= 5000000:
        price_tag.append('Under 30-50 Lakhs')
    elif i <= 10000000:
        price_tag.append('Under 50 Lakhs - 1 Crore')
    elif i <= 20000000:
        price_tag.append('Under 1 Crore - 2 Crore')
    else:
        price_tag.append('More than 2 Crore')
df1['Price Tag'] = price_tag

df1['Price Tag'].value_counts()

Price Tag
Under 1-10 Lakhs            12808
```

```
Under 10-20 Lakhs              1721
Under 20-30 Lakhs               395
Under 30-50 Lakhs               278
Under 50 Lakhs - 1 Crore        109
Under 1 Lakhs                    93
Under 1 Crore - 2 Crore           5
More than 2 Crore                 2
Name: count, dtype: int64
```

```
df1['Price
Tag'].value_counts().plot(kind='bar',color=sns.color_palette('cool',8)
)
plt.show()
```



```python
# 2. How does the average selling price vary by car brand?

temp_df = df1.groupby('brand')
['selling_price'].mean().reset_index().round(1)
temp_df.sort_values(by = 'selling_price' , ascending = False)
```

	brand	selling_price
4	Ferrari	39500000.0
26	Rolls-Royce	24200000.0
2	Bentley	9266666.7
19	Maserati	6100000.0
24	Porsche	5161190.5
15	Lexus	5146500.0
20	Mercedes-AMG	5100000.0
14	Land Rover	3823902.0
31	Volvo	3729700.0
1	BMW	2693826.9
11	Jaguar	2643033.9
21	Mercedes-Benz	2480741.8
22	Mini	2182647.1
0	Audi	1966864.6
9	ISUZU	1897500.0
12	Jeep	1795804.9
16	MG	1752947.4
13	Kia	1735250.0
29	Toyota	1371316.5
10	Isuzu	1355000.0
23	Nissan	955363.6
17	Mahindra	787455.0
27	Skoda	784089.8
5	Force	700000.0
28	Tata	683534.9
6	Ford	645224.1
7	Honda	617756.9
8	Hyundai	576153.9
30	Volkswagen	516546.8
18	Maruti	487089.3
25	Renault	440985.1
3	Datsun	320517.6

```python
# 3.What is the relationship between vehicle age and selling price?
df1[['vehicle_age','selling_price']].corr()
```

	vehicle_age	selling_price
vehicle_age	1.000000	-0.241851
selling_price	-0.241851	1.000000

```python
# 4.How does km_driven correlate with selling_price?

df1[['km_driven','selling_price']].corr()
```

```
               km_driven  selling_price
km_driven        1.00000       -0.08003
selling_price   -0.08003        1.00000
```

```
# 5.What is the count of cars available by fuel_type (Petrol,
Diesel,etc.)?

df1['fuel_type'].value_counts()
```

```
fuel_type
Petrol      7643
Diesel      7419
CNG          301
LPG           44
Electric       4
Name: count, dtype: int64
```

```
#6.What is the distribution of transmission_type among listed cars?
df1['transmission_type'].value_counts()
```

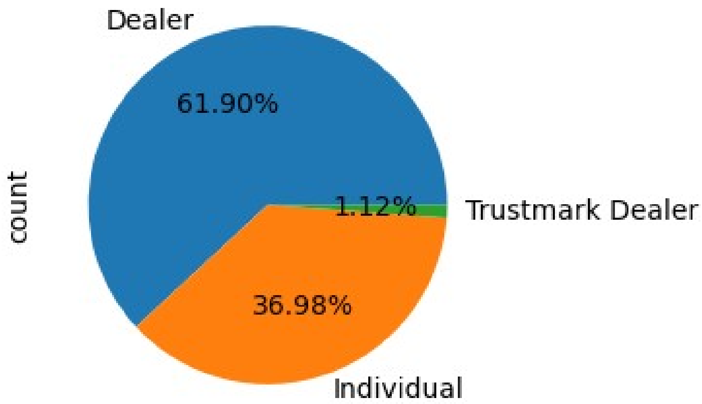
```
transmission_type
Manual       12225
Automatic     3186
Name: count, dtype: int64
```

```
#7. Which Seller_type(Dealer/Individual) lists the highest number of
cars ?
df1['seller_type'].value_counts()
```

```
seller_type
Dealer              9539
Individual          5699
Trustmark Dealer     173
Name: count, dtype: int64
```

```
df1['seller_type'].value_counts().plot(kind='pie',autopct = '%.2f%
%',figsize = (3,3))
```

```
<Axes: ylabel='count'>
```

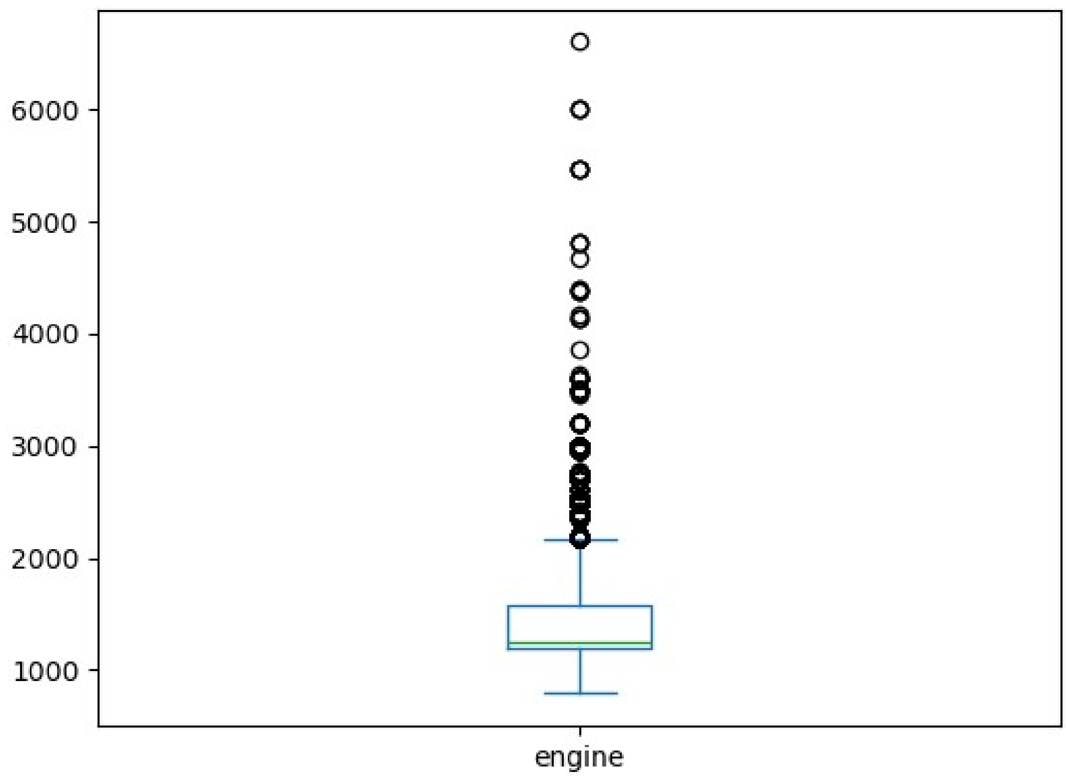



```
#8. How do mileage and max_power relate to each other across models?
df1[['mileage','max_power']].corr()
```

	mileage	max_power
mileage	1.000000	-0.533128
max_power	-0.533128	1.000000

```
#9. Are there any noticeable outliers in engine capacity or
selling_price?
df1['engine'].plot(kind='box')
```

```
<Axes: >
```

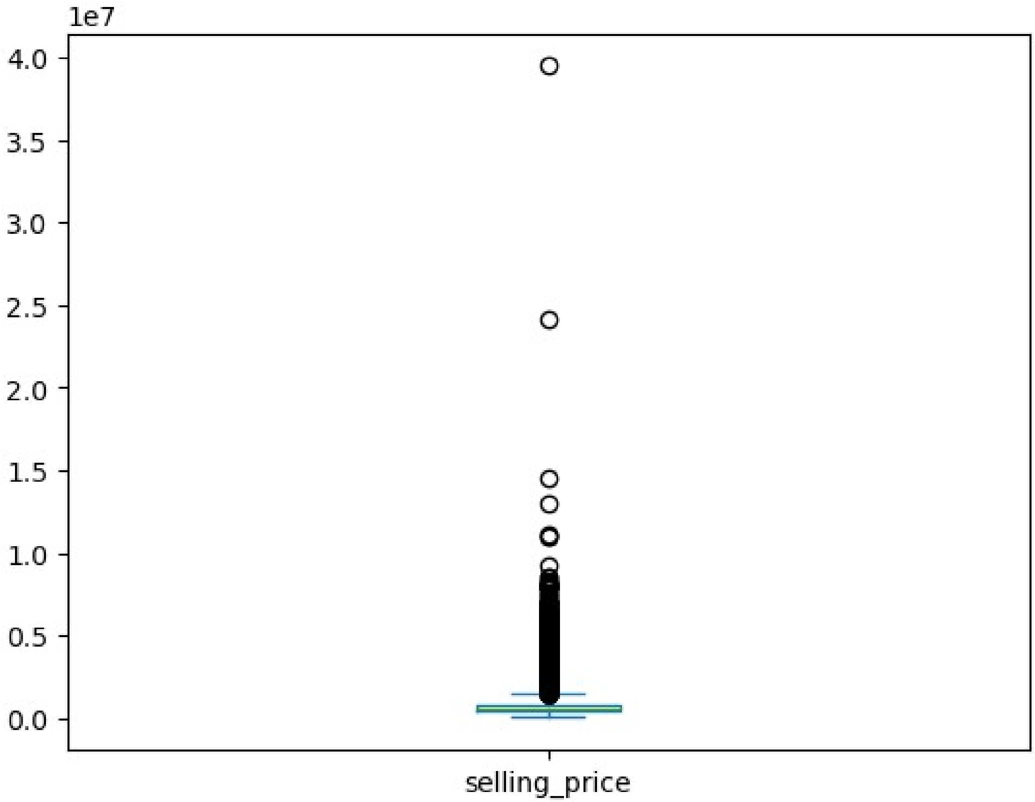



```
df1['engine'].max()
```

```
6592
```

```
df1['selling_price'].plot(kind='box')
```

```
<Axes: >
```



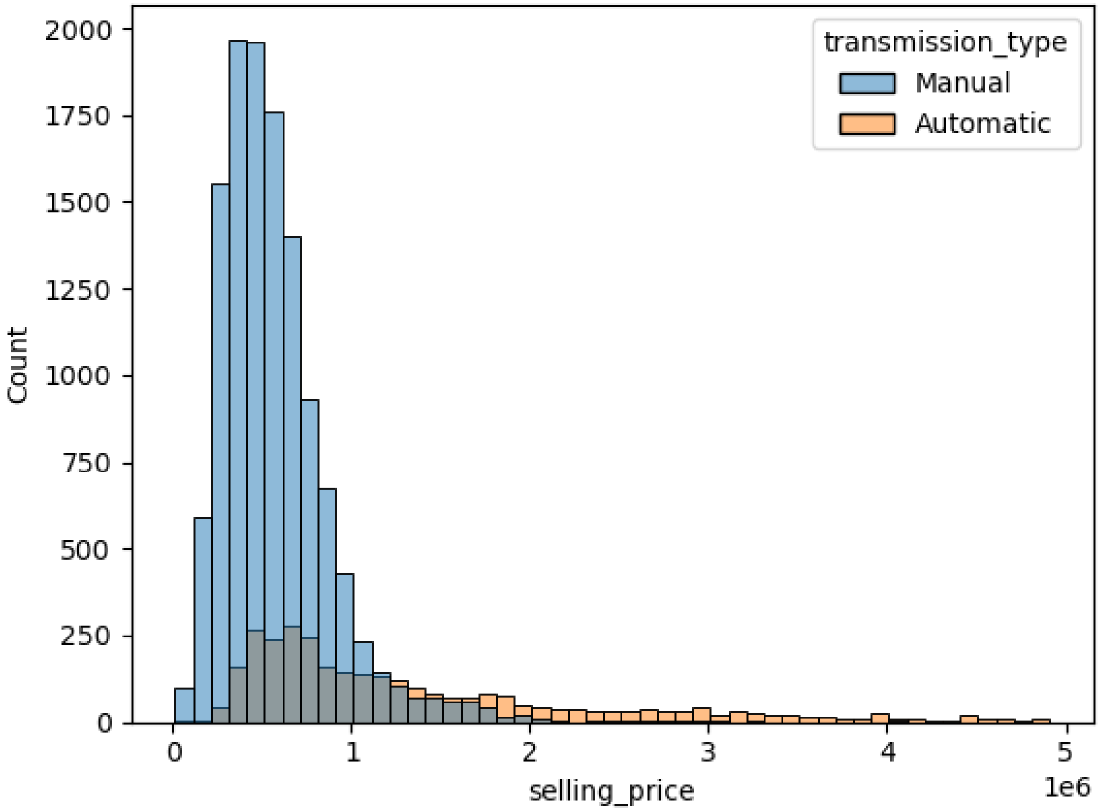

```
# Which car models have the highest average km_driven?
temp_df = df1.groupby('model')
['km_driven'].mean().round(2).reset_index()
temp_df.sort_values('km_driven',ascending=False).head(1)
```

```
       model  km_driven
102  X-Trail   104000.0
```

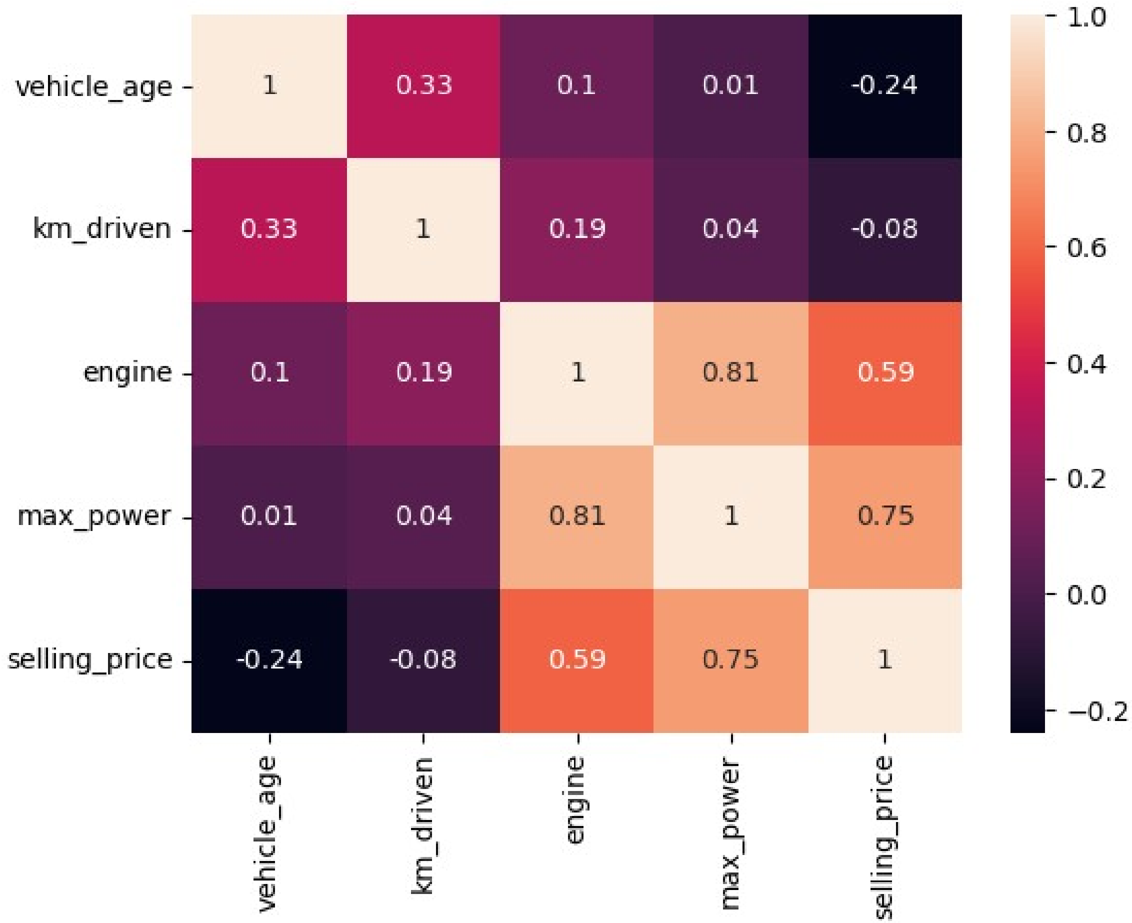
```
df1[df1['model'] == 'X-Trail']['brand'].unique()
```

```
array(['Nissan'], dtype=object)
```

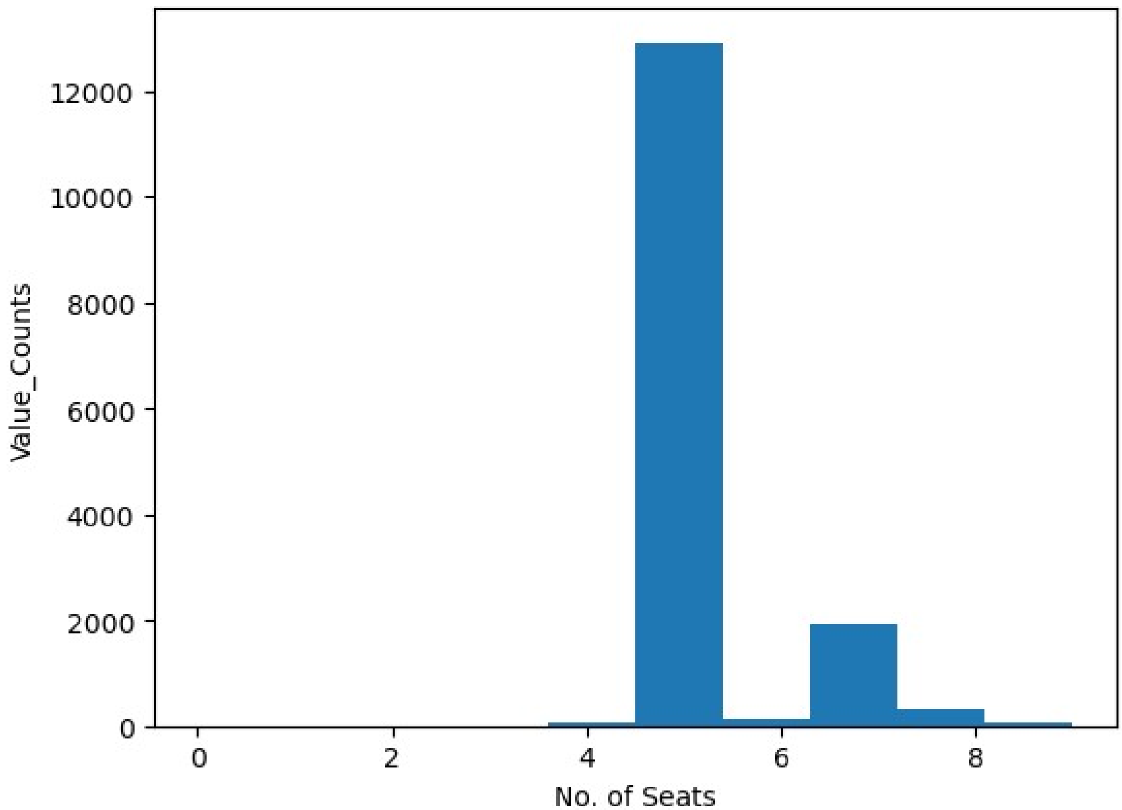
```
# 11. How does selling_price vary across different transmission types?
temp_df = df1[['transmission_type','selling_price']]
sns.histplot(data = temp_df , x ='selling_price',hue =
'transmission_type',bins = range(10000,5000000,100000))
plt.show()
```




```
# 12. Which factors (vehicle_age, km_driven, engine, mileage,
max_power) have the strongest correlation with selling_price?
temp_corr =
df1[['vehicle_age','km_driven','engine','max_power','selling_price']].
corr().round(2)
sns.heatmap(temp_corr,annot = True)
plt.show()
```




```
# 13.What is the distribution of seats available in the dataset?
df1['seats'].plot(kind = 'hist',xlabel = 'No. of Seats',ylabel
='Value_Counts')
plt.show()
```




```
#14. Which brand and model combinations offer the best mileage on average?
df1.groupby(['brand','model'])['mileage'].mean().sort_values(ascending
= False).head(5)
```

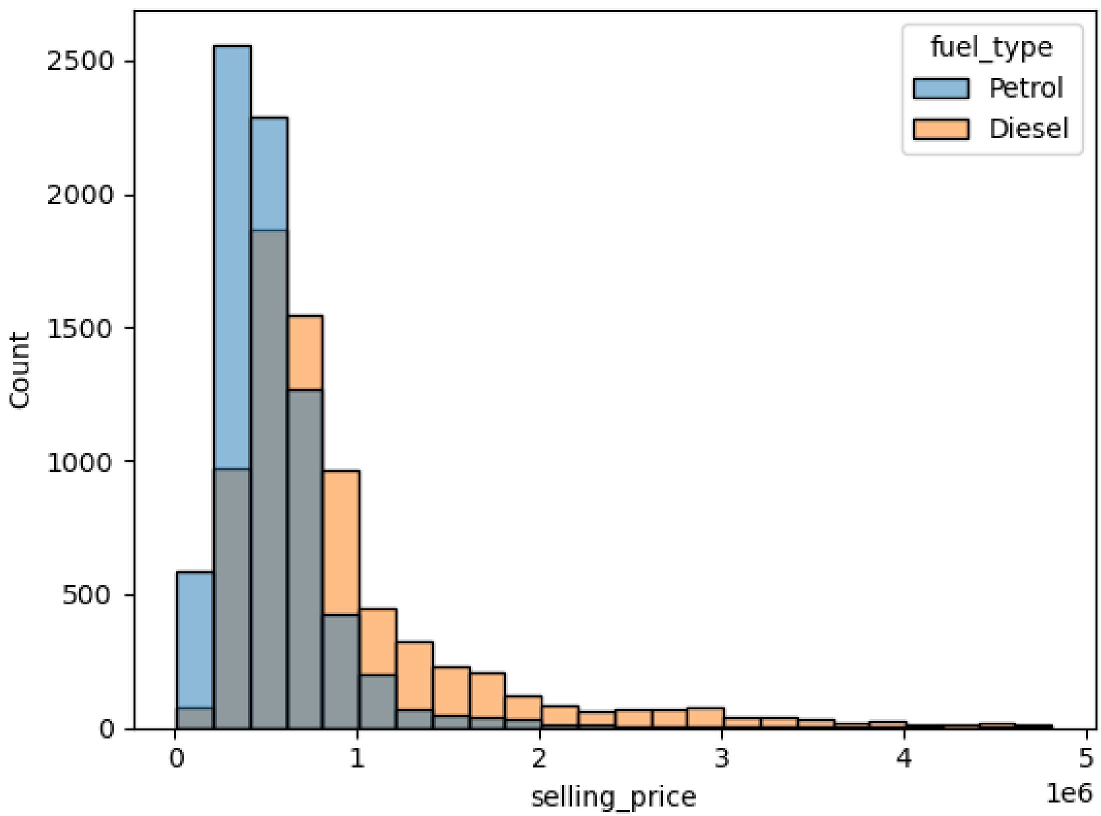
```
brand    model
Tata     Tiago       24.625103
Maruti   Ciaz        24.289046
         Vitara      24.231932
Renault  KWID        24.037810
Maruti   Celerio     23.703502
Name: mileage, dtype: float64
```

```
#15. Are there significant differences in selling_price between petrol and diesel vehicles?

df1['fuel_type'].unique()
```

```
array(['Petrol', 'Diesel', 'CNG', 'LPG', 'Electric'], dtype=object)
```

```
petrol_diesel_df = df1[df1['fuel_type'].isin(['Diesel','Petrol'])]
```

```
sns.histplot(data=petrol_diesel_df,x =
'selling_price',hue='fuel_type',bins=range(10000,5000000,200000))
plt.show()
```



Findings and Insights

Insights from EDA Analysis:

1. **Top Brands Dominating Inventory**:
   - Maruti, Hyundai, and Honda account for approximately 65-70% of the overall car inventory.
2. **Most Demanded Models**:
   - Popular models from Maruti and Hyundai, such as the i20, Swift, Dzire, and Alto, are the most in-demand on the platform.
3. **Dealer Partnerships**:
   - CarDekho has a stronger tie-up with dealers (including Trustmark dealers), followed by individual sellers.
4. **Fuel Type Distribution**:
   - Petrol and Diesel cars make up the majority of the inventory, while Electric and LPG vehicles remain relatively underrepresented.
5. **Transmission Type**:
   - Over 80% of the cars listed have manual transmissions.
6. **Vehicle Age and Value**:
   - The majority of vehicles fall within the 2-10 year age range. As cars age, their quantity and selling price generally decrease.

7. **Selling Patterns**:
   - Cars are more likely to be sold when the odometer reading is under 25,000 km.
8. **Mileage Trends**:
   - Most second-hand cars have a mileage range between 12,000 km and 25,000 km, indicating that many are in good condition.
9. **Engine Capacity**:
   - A majority of cars have mid-range engine capacities, suggesting that most buyers and sellers are focused on passenger vehicles rather than sports or luxury cars.
10. **Affordability for Passengers**:
    - Approximately 90-95% of cars are within a budget of up to Rs.25 lakhs, making them accessible to the average consumer.
11. **Luxury and Sports Cars**:
    - The top 20 most expensive cars fall into the sports or luxury category, with BMW, Benz, and Volvo leading the pack.
12. **Cost-Effective Brands**:
    - Maruti and Hyundai are the dominant cost-effective brands in the inventory.
13. **Luxury Cars and Transmission Type**:
    - Sports and luxury cars are more likely to feature automatic transmissions.
14. **Value Retention in Luxury Cars**:
    - Despite higher mileage, sports and luxury cars maintain higher selling prices, reflecting their strong brand value.
15. **Price Range Distribution**:
    - About 83% (12,000 cars) of the listings are priced below Rs.10 lakhs, with a significant portion falling in the Rs.1-10 lakh range.

Suggestions:

1. **Expand Luxury and Sports Car Listings**:
   - Given the strong brand value retention, adding more sports and luxury cars could attract buyers, especially as their prices remain stable even with higher mileage.
2. **Increase Inventory of Popular Models**:
   - To boost sales and revenue, it is recommended to include newer models from in-demand brands such as Maruti, Hyundai, and Honda.
3. **Diversify with Electric Vehicles**:
   - In line with industry trends and growing demand for electric vehicles, CarDekho should consider increasing its inventory of EVs.
4. **Address Data Outliers**:
   - The engine capacity and selling price data show outliers that could skew future price predictions. These should be addressed to improve the accuracy of pricing models.
5. **Leverage Key Predictive Features**:
   - Engine capacity and maximum power are highly correlated with selling prices and can be valuable predictors for sales forecasting.

Final Conclusion:

The EDA reveals that CarDekho's inventory is dominated by popular and affordable brands like Maruti, Hyundai, and Honda, with most cars falling in the ₹1–10 lakh range and featuring manual transmission. Buyers prefer cars with lower mileage and mid-range engine capacities, while electric vehicles remain underrepresented despite rising market demand. Sports and luxury cars, though fewer in number, retain strong resale value even with higher mileage.

Overall, CarDekho can improve its market reach by expanding its luxury and EV segments, adding more high-demand newer models, and using key predictors like engine capacity and max power to enhance pricing accuracy and sales forecasting.

Design and Developed by: Saurabh Kumar(Data Analyst)

Connect With Me