PDS-Assignment2

a) Missing values were identified in all columns of the train dataset. Imputation was performed by replacing them with either mean, median, or mode, depending on the nature of the data and the specific column.

```
[5847 rows x 14 columns]
```

In [6]:
```
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 Unnamed: 0             0
Name                    0
Location                0
Year                    0
Kilometers_Driven       0
Fuel_Type               0
Transmission            0
Owner_Type              0
Mileage                 2
Engine                 36
Power                  36
Seats                  38
New_Price            5032
Price                   0
dtype: int64
```

In [7]:
```
df['Mileage'] = df['Mileage'].str.extract('(\d+\.\d+)').astype(float)
df['Mileage'].fillna(df['Mileage'].median(), inplace=True)
```

In [8]:
```
df['Engine'] = df['Engine'].str.extract('(\d+)').astype(float)

df['Power'] = df['Power'].str.extract('(\d+\.\d+)').astype(float)
```

5847 rows × 13 columns

In [13]:
```
df.drop('Unnamed: 0', axis=1, inplace=True)
```

In [15]:
```
df.head()
```

Out[15]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.19	1199.0	88.70	5.0	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50

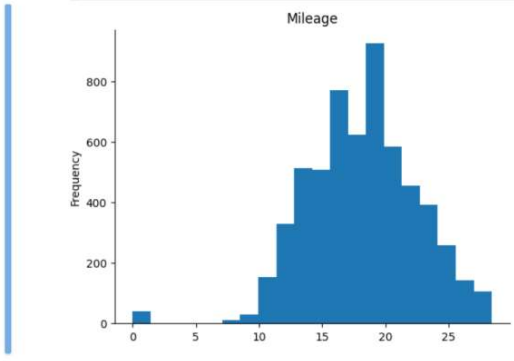
In [14]:
```
df.to_csv("clean_data.csv")
```

5847 rows × 13 columns

In [6]:
```
from matplotlib import pyplot as plt
import seaborn as sns
import pandas as pd
plt.subplots(figsize=(8, 8))
df_2dhist = pd.DataFrame({
    x_label: grp['Transmission'].value_counts()
    for x_label, grp in df.groupby('Fuel_Type')
})
sns.heatmap(df_2dhist, cmap='viridis')
plt.xlabel('Fuel_Type')
_ = plt.ylabel('Transmission')
```

b) Units were removed from certain attributes in the train dataset, retaining only the numerical values. This involved removing "kmpl" from "Mileage," "CC" from "Engine," "bhp" from "Power," and "lakh" from "New_price."

```
In [7]: from matplotlib import pyplot as plt
        df['Mileage'].plot(kind='hist', bins=20, title='Mileage')
        plt.gca().spines[['top', 'right',]].set_visible(False)
```



Question C

c) Categorical variables "Fuel_Type" and "Transmission" were transformed into numerical one-hot encoded values, facilitating further analysis and modeling.

0 5 10 15 20 25

Question C

```
In [8]: df = pd.get_dummies(df, columns=['Fuel_Type', 'Transmission'])
In [9]: df
```

Out[9]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Type_Die
0	0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	
1	1	Honda Jazz V	Chennai	2011	46000	First	18.19	1199.0	88.70	5.0	4.50	
2	2	Maruti	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	

d) An additional feature was created and added to the train dataset. Specifically, the current age of each car was calculated by subtracting the "Year" value from the current year, providing valuable information for analysis.

Question D

In [10]:
```
from datetime import datetime
# Calculate current year
current_year = datetime.now().year

# Calculate the age of the car
df['Age'] = current_year - df['Year']
```

In [11]:
```
df
```

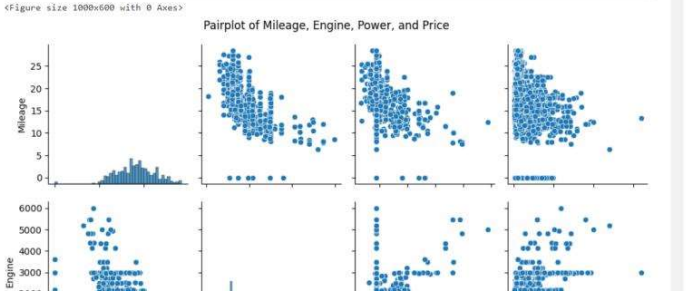
Out[11]:

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Type_Die
0	0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	
1	1	Honda Jazz V	Chennai	2011	46000	First	18.19	1199.0	88.70	5.0	4.50	

e) Various data manipulation operations were performed on the train dataset, including select, filter, rename, mutate, arrange, and summarize with group by operations. These operations allowed for the extraction of specific columns, filtering of rows based on conditions, renaming of columns, creation of new columns, sorting of data, and aggregation of information by groups for summary analysis.

```
sns.pairplot(df[['Mileage', 'Engine', 'Power', 'Price_in_lakhs']])
plt.suptitle('Pairplot of Mileage, Engine, Power, and Price', y=1.02)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



In [22]:
```
plt.figure(figsize=(12, 8))
sns.violinplot(x='Location', y='Price_in_lakhs', data=df)
plt.title('Price Distribution by Location')
plt.xlabel('Location')
plt.ylabel('Price (in Lakhs)')
plt.xticks(rotation=45)
plt.show()
```

[5847 rows x 19 columns]

In [24]:
```
# Grouping by Fuel Type and Calculating Mean Price:
fuel_type_price = df.groupby('Fuel_Type_Diesel')['Price_in_lakhs'].mean()
print(fuel_type_price)
```

```
Fuel_Type_Diesel
0    5.761988e+05
1    1.296069e+06
Name: Price_in_lakhs, dtype: float64
```

In [25]:
```
# Grouping by Transmission Type and Finding Maximum Power
transmission_power_max = df.groupby('Transmission_Manual')['Power'].max()
print(transmission_power_max)
```

```
Transmission_Manual
0    488.1
1    199.3
Name: Power, dtype: float64
```

In [26]:
```
import matplotlib.pyplot as plt
grouped_cars_by_location = df.groupby('Location').size().reset_index(name='Number_of_Cars')

print(grouped_cars_by_location)
```

```
      Location  Number_of_Cars
0    Ahmedabad             218
1    Bangalore             352
2      Chennai             476
9       Mumbai             762
10        Pune             590
```

In [28]:
```
# Plotting the bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x='Location', y='Number_of_Cars', data=grouped_cars_by_location)
plt.title('Number of Cars by Location')
plt.xlabel('Location')
plt.ylabel('Number of Cars')
plt.xticks(rotation=45)
plt.show()
```