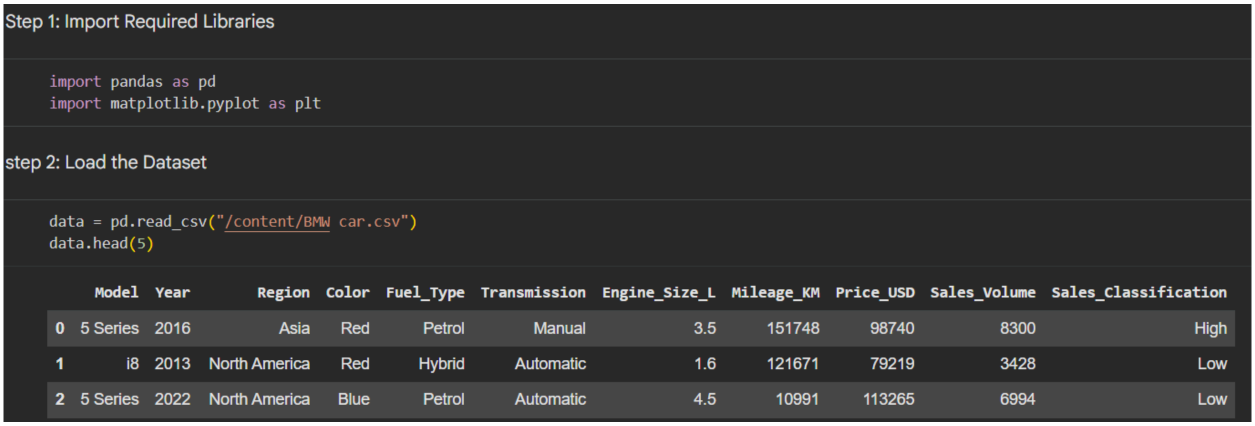
Task 2

Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.



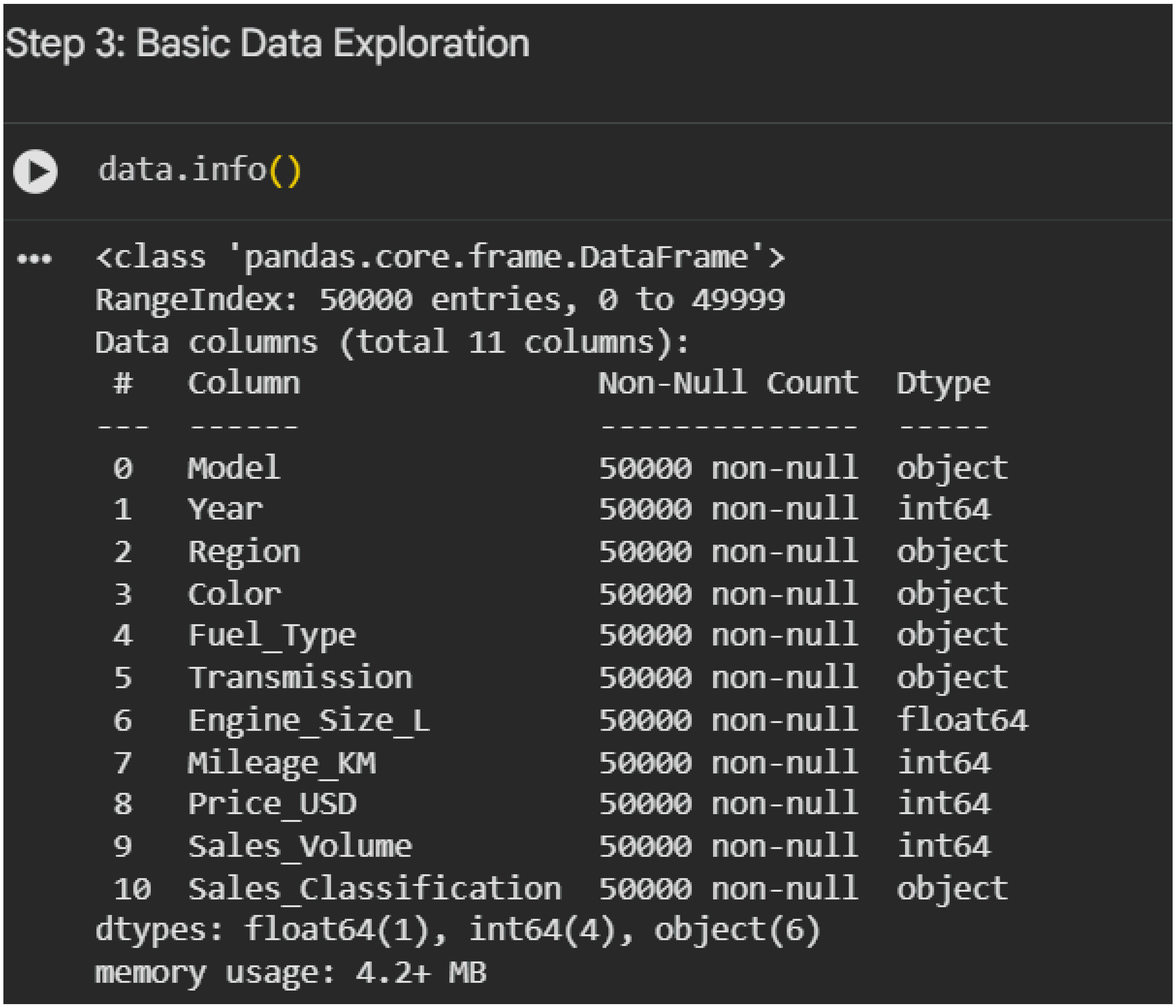
Step 1: Import Required Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
```

step 2: Load the Dataset

```
data = pd.read_csv("/content/BMW car.csv")
data.head(5)
```

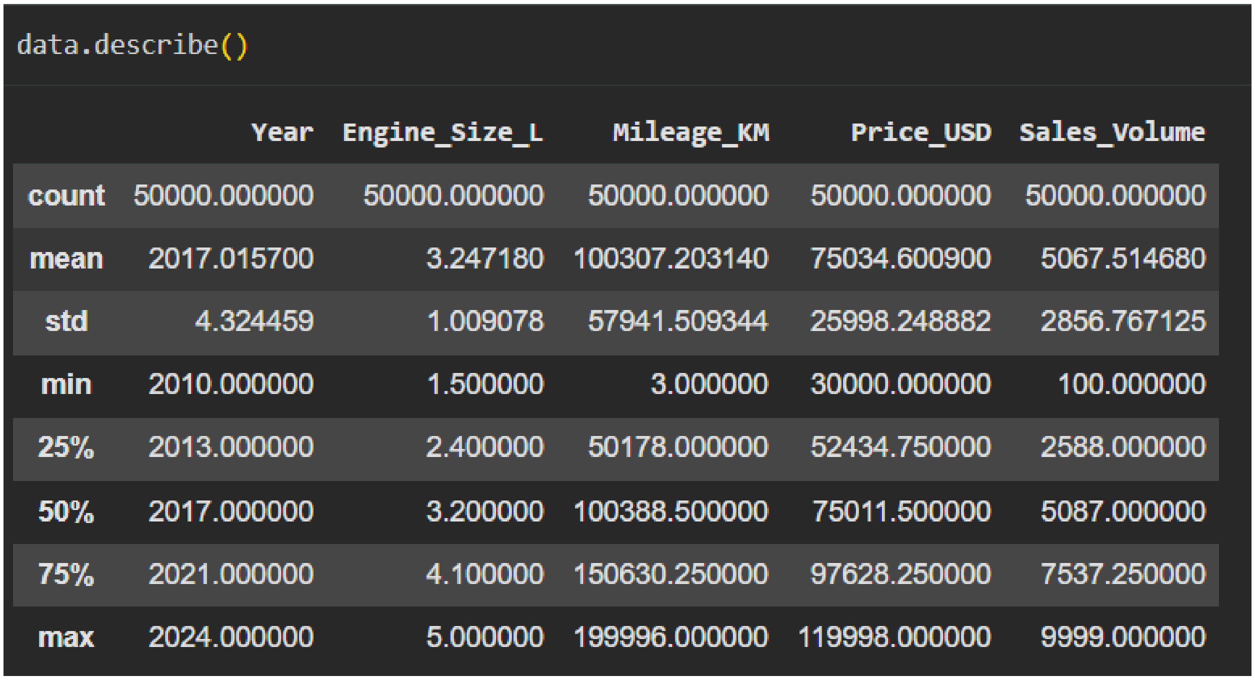
	Model	Year	Region	Color	Fuel_Type	Transmission	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume	Sales_Classification
0	5 Series	2016	Asia	Red	Petrol	Manual	3.5	151748	98740	8300	High
1	i8	2013	North America	Red	Hybrid	Automatic	1.6	121671	79219	3428	Low
2	5 Series	2022	North America	Blue	Petrol	Automatic	4.5	10991	113265	6994	Low



Step 3: Basic Data Exploration

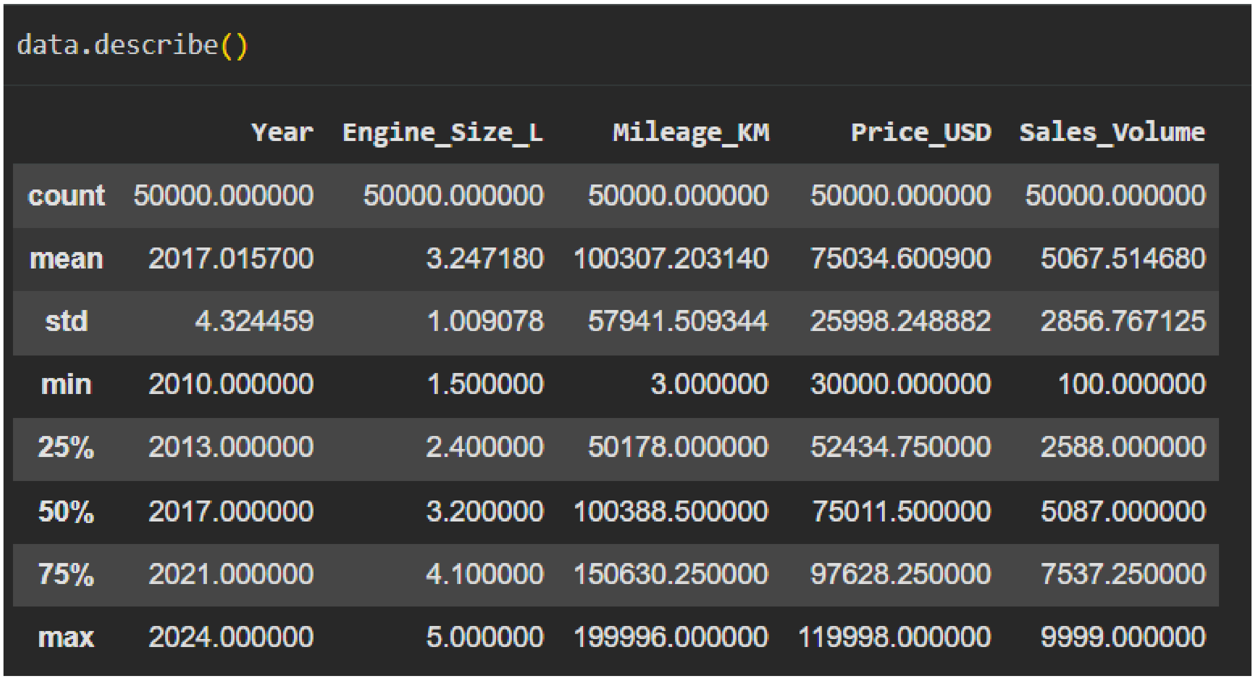
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Model                 50000 non-null  object 
 1   Year                  50000 non-null  int64  
 2   Region                50000 non-null  object 
 3   Color                 50000 non-null  object 
 4   Fuel_Type             50000 non-null  object 
 5   Transmission          50000 non-null  object 
 6   Engine_Size_L         50000 non-null  float64
 7   Mileage_KM            50000 non-null  int64  
 8   Price_USD             50000 non-null  int64  
 9   Sales_Volume          50000 non-null  int64  
 10  Sales_Classification  50000 non-null  object 
dtypes: float64(1), int64(4), object(6)
memory usage: 4.2+ MB
```



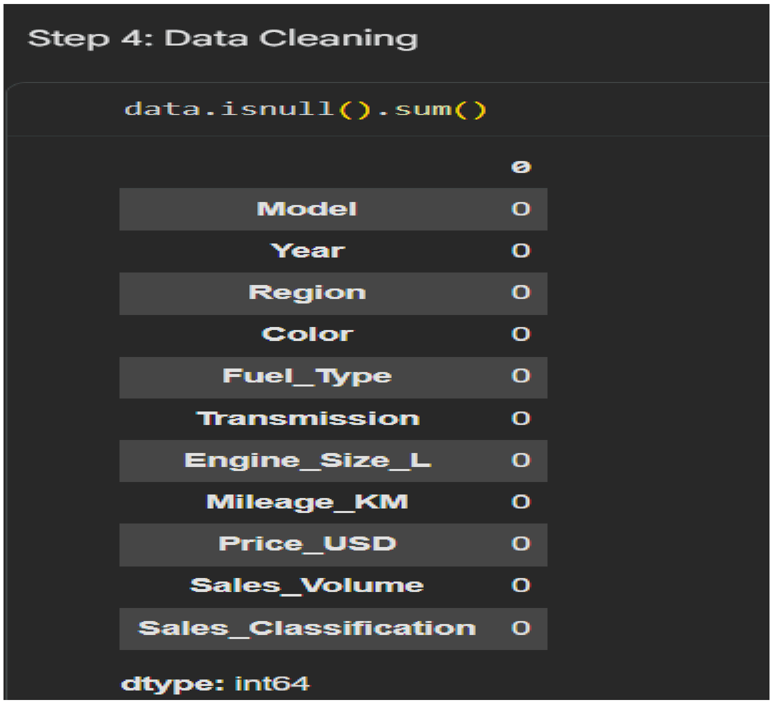
```
data.describe()
```

	Year	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	2017.015700	3.247180	100307.203140	75034.600900	5067.514680
std	4.324459	1.009078	57941.509344	25998.248882	2856.767125
min	2010.000000	1.500000	3.000000	30000.000000	100.000000
25%	2013.000000	2.400000	50178.000000	52434.750000	2588.000000
50%	2017.000000	3.200000	100388.500000	75011.500000	5087.000000
75%	2021.000000	4.100000	150630.250000	97628.250000	7537.250000
max	2024.000000	5.000000	199996.000000	119998.000000	9999.000000



```python
data.describe()
```

```
              Year  Engine_Size_L     Mileage_KM      Price_USD  Sales_Volume
count  50000.000000   50000.000000   50000.000000   50000.000000  50000.000000
mean    2017.015700       3.247180  100307.203140   75034.600900   5067.514680
std        4.324459       1.009078   57941.509344   25998.248882   2856.767125
min     2010.000000       1.500000       3.000000   30000.000000    100.000000
25%     2013.000000       2.400000   50178.000000   52434.750000   2588.000000
50%     2017.000000       3.200000  100388.500000   75011.500000   5087.000000
75%     2021.000000       4.100000  150630.250000   97628.250000   7537.250000
max     2024.000000       5.000000  199996.000000  119998.000000   9999.000000
```

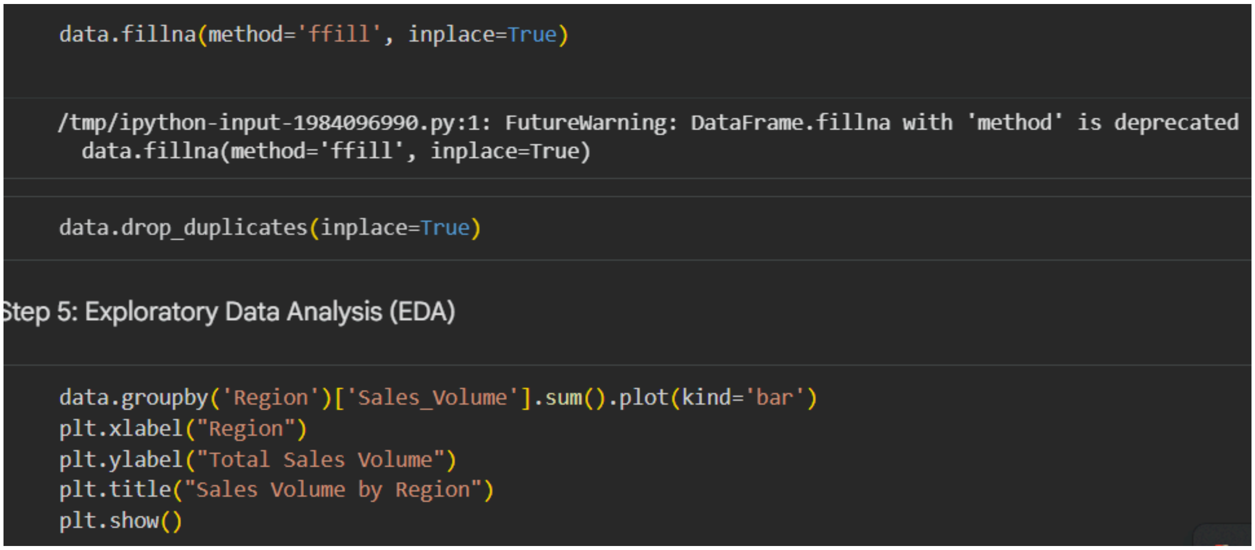


Step 4: Data Cleaning

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

```
                        0
               Model    0
                Year    0
              Region    0
               Color    0
           Fuel_Type    0
        Transmission    0
       Engine_Size_L    0
          Mileage_KM    0
           Price_USD    0
        Sales_Volume    0
Sales_Classification    0

dtype: int64
```



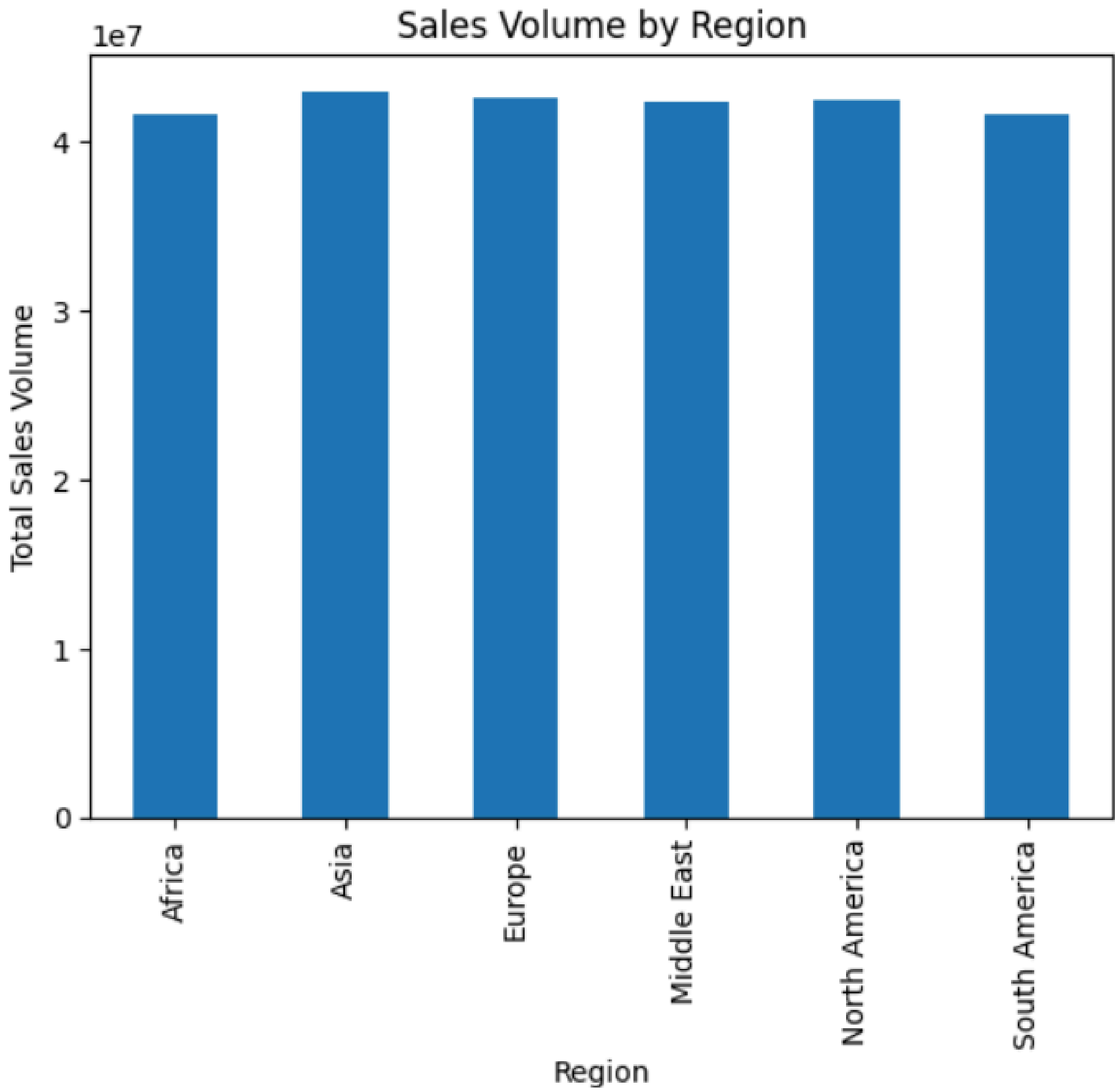
```
data.fillna(method='ffill', inplace=True)
```

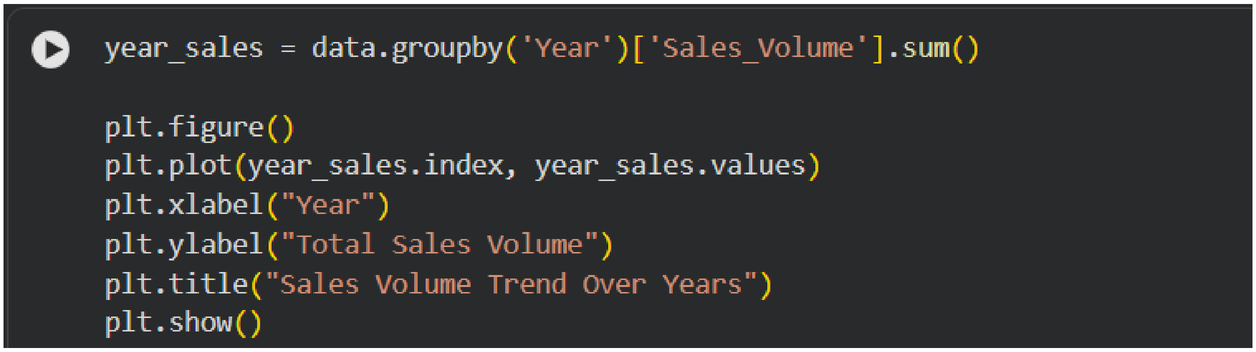
```
/tmp/ipython-input-1984096990.py:1: FutureWarning: DataFrame.fillna with 'method' is deprecated
  data.fillna(method='ffill', inplace=True)
```

```
data.drop_duplicates(inplace=True)
```

Step 5: Exploratory Data Analysis (EDA)

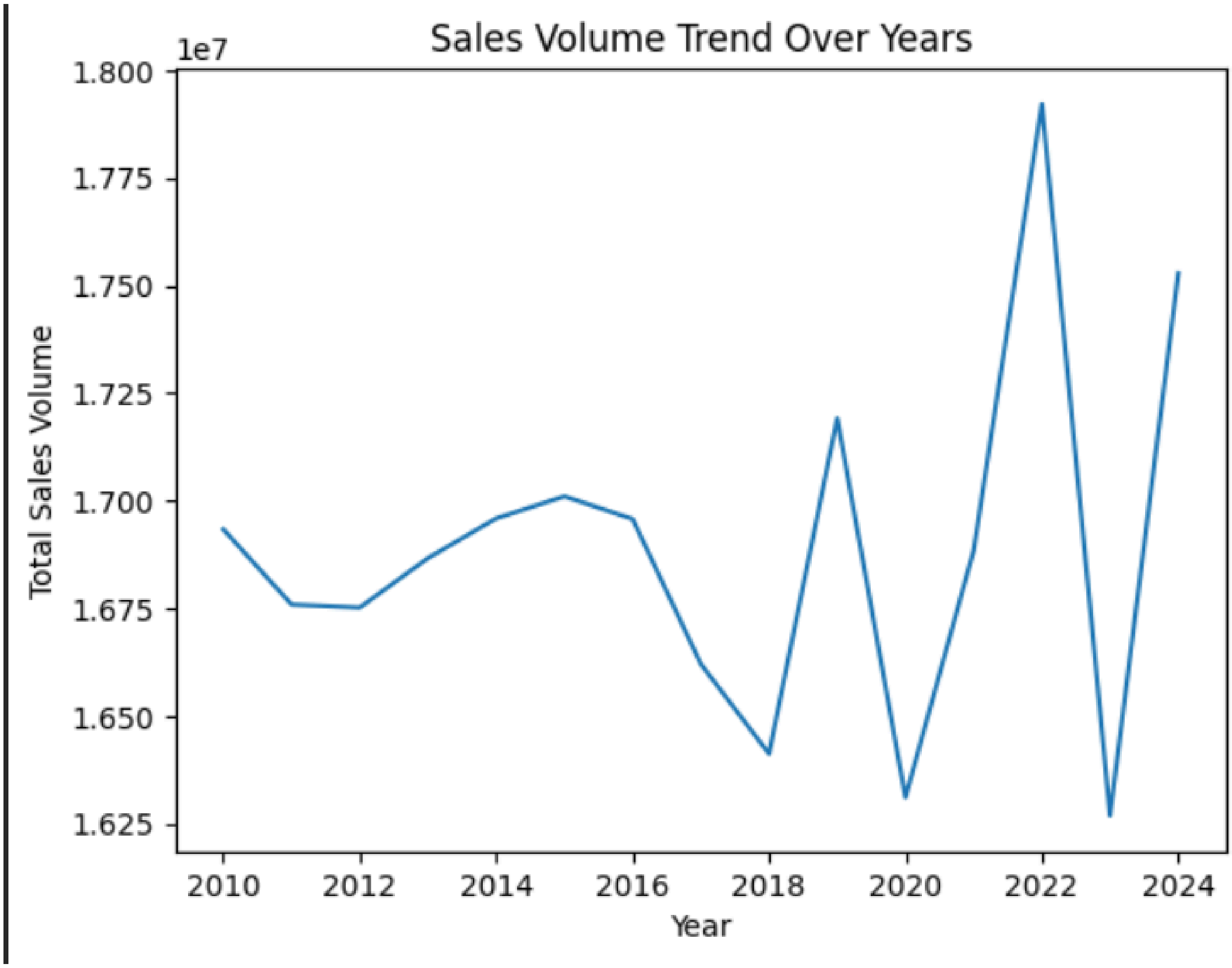
```
data.groupby('Region')['Sales_Volume'].sum().plot(kind='bar')
plt.xlabel("Region")
plt.ylabel("Total Sales Volume")
plt.title("Sales Volume by Region")
plt.show()
```

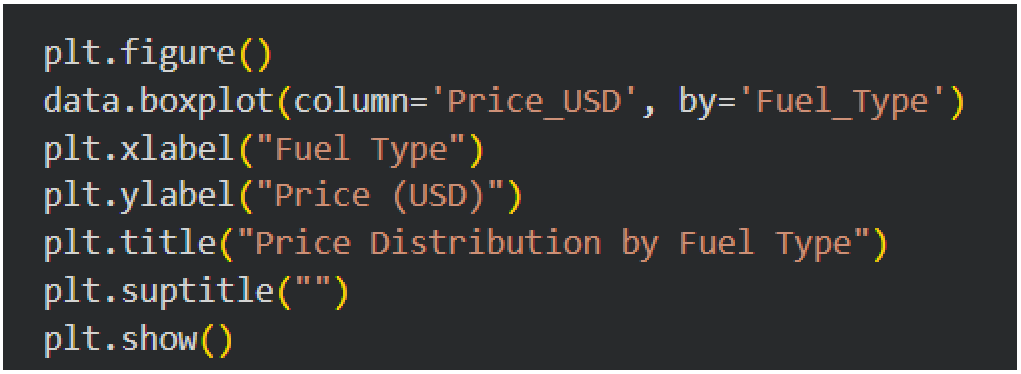




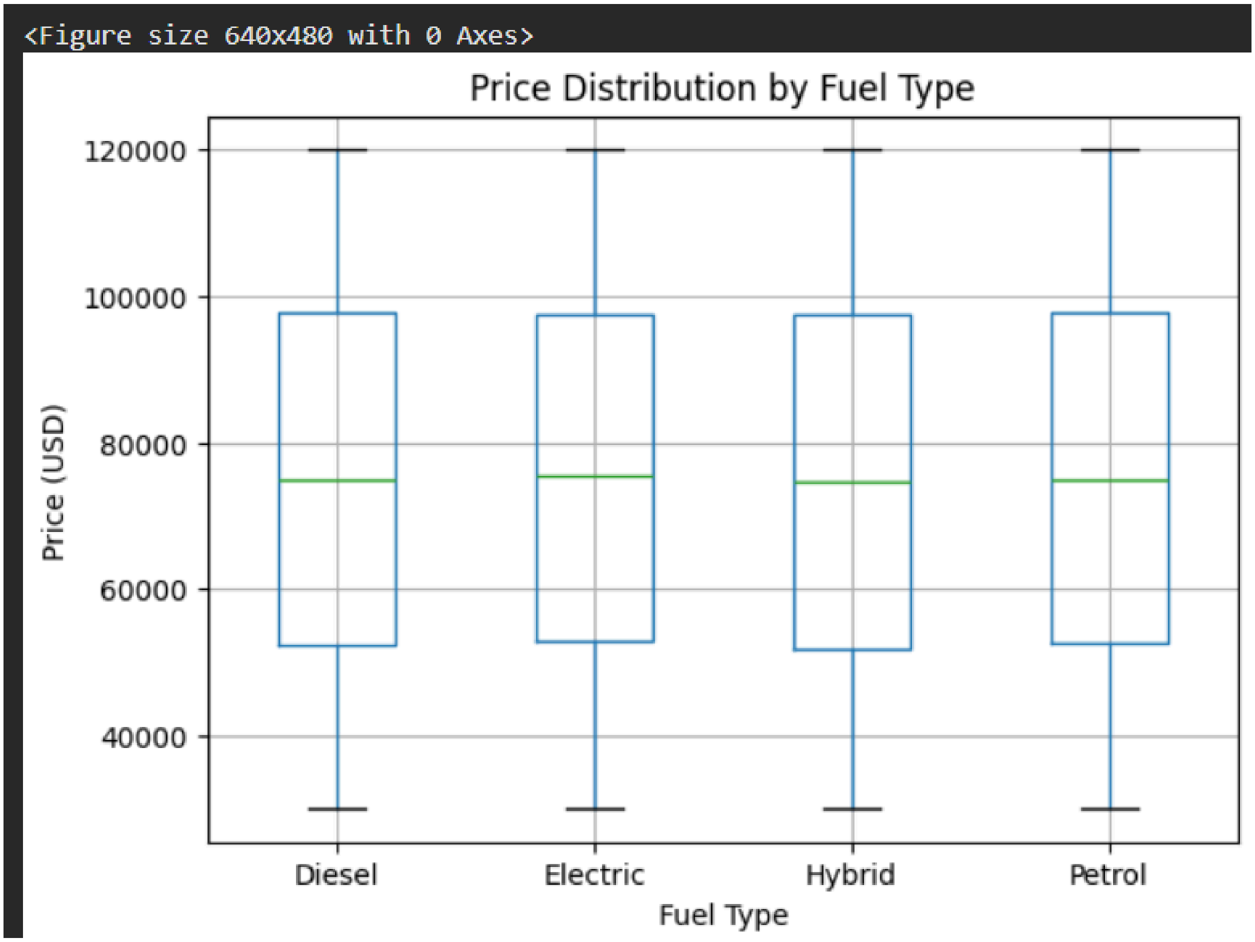
```
year_sales = data.groupby('Year')['Sales_Volume'].sum()

plt.figure()
plt.plot(year_sales.index, year_sales.values)
plt.xlabel("Year")
plt.ylabel("Total Sales Volume")
plt.title("Sales Volume Trend Over Years")
plt.show()
```





```
plt.figure()
data.boxplot(column='Price_USD', by='Fuel_Type')
plt.xlabel("Fuel Type")
plt.ylabel("Price (USD)")
plt.title("Price Distribution by Fuel Type")
plt.suptitle("")
plt.show()
```



```
<Figure size 640x480 with 0 Axes>
```

Price Distribution by Fuel Type
120000
100000
80000
60000
40000
Price (USD)
Diesel
Electric
Hybrid
Petrol
Fuel Type