

```

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Replace with your dataset path

file_path = '//content/drive/MyDrive/dataseat/bmw_global_sales_2018_2025.csv'

```

```
df = pd.read_csv(file_path)
```

```
# Show first rows
```

```
print("Dataset Preview:")
```

```
print(df.head())
```

Dataset Preview:

	Year	Month	Region	Model	Units_Sold	Avg_Price_EUR	Revenue_EUR \
0	2018	1	Europe	3 Series	7822	47482	371404204
1	2018	1	Europe	5 Series	10280	61685	634121800
2	2018	1	Europe	X3	3105	58433	181434465
3	2018	1	Europe	X5	7420	67955	504226100
4	2018	1	Europe	X7	8474	92300	782150200

	BEV_Share	Premium_Share	GDP_Growth	Fuel_Price_Index
0	0.011	19.12	3.5	1.0
1	0.019	19.12	3.5	1.0
2	0.022	19.12	3.5	1.0
3	0.021	19.12	3.5	1.0
4	0.035	19.12	3.5	1.0

```

print("\nMissing values before handling:")
print(df.isnull().sum())

# Fill numeric columns with mean
numeric_cols = df.select_dtypes(include=np.number).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Fill categorical columns with mode
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)

print("\nMissing values after handling:")
print(df.isnull().sum())

```

Missing values before handling:

Year	0
Month	0
Region	0
Model	0
Units_Sold	0
Avg_Price_EUR	0
Revenue_EUR	0
BEV_Share	0

```
Premium_Share    0
GDP_Growth       0
Fuel_Price_Index  0
dtype: int64
```

Missing values after handling:

```
Year            0
Month           0
Region          0
Model           0
Units_Sold      0
Avg_Price_EUR   0
Revenue_EUR     0
BEV_Share       0
Premium_Share   0
GDP_Growth      0
Fuel_Price_Index 0
dtype: int64
```

/tmp/ipykernel_186/3405905384.py:11: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df[col].fillna(df[col].mode()[0], inplace=True)

for col in numeric_cols:

    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)

    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

print("\nOutliers handled successfully")
```

Outliers handled successfully

Histogram for numeric columns

```
df[numeric_cols].hist(figsize=(10,8))
plt.suptitle("Histogram of Numerical Features")
plt.show()
```

Correlation Heatmap

```
plt.figure(figsize=(8,6))
sns.heatmap(df[numeric_cols].corr(), annot=True, cmap="coolwarm")
```

```
plt.title("Correlation Heatmap")
```

```
plt.show()
```

```
# Boxplot to visualize outliers
```

```
plt.figure(figsize=(10,6))
```

```
sns.boxplot(data=df[numeric_cols])
```

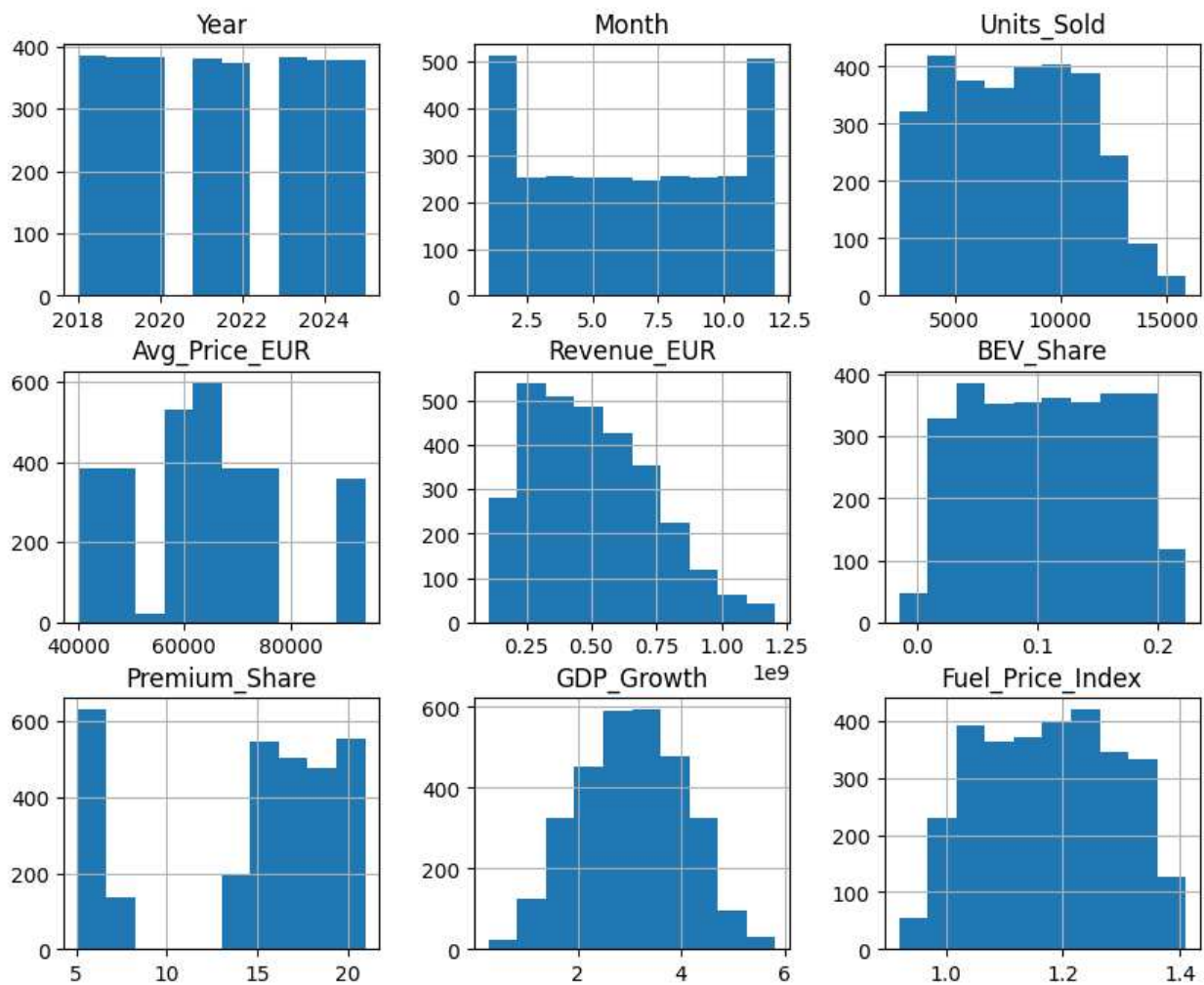
```
plt.title("Boxplot of Numerical Features")
```

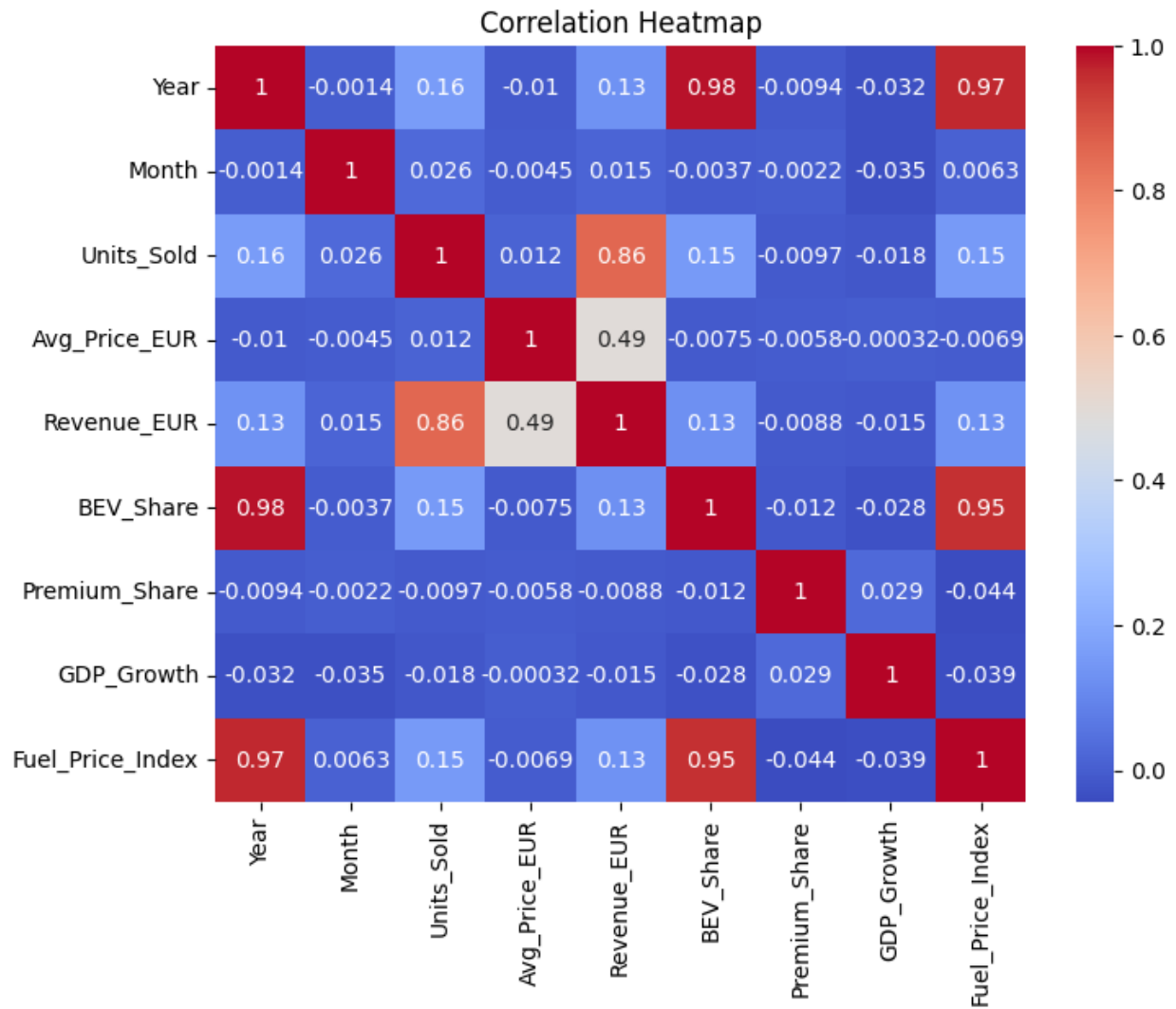
```
plt.xticks(rotation=45)
```

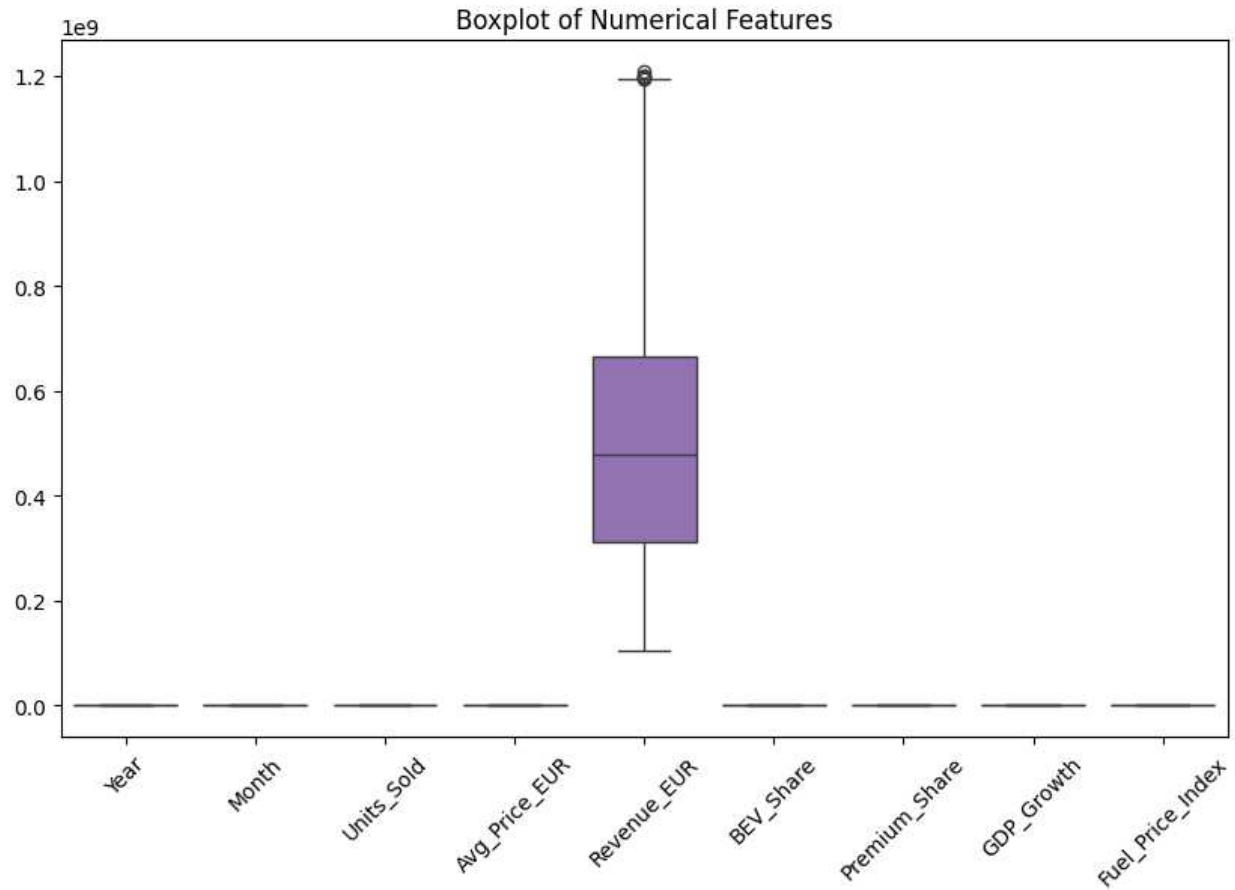
```
plt.show()
```

```
print("\nData visualization completed successfully.")
```

Histogram of Numerical Features







Data visualization completed successfully.

Name : Redhwan Murshed Hamoud Moqbel

ID Number: 261003286

Data set was used : <https://www.kaggle.com/dmahajanbe23/bmw-global-automotive-sales?resource=download>