🚗 Electric Vehicle Sales Data Analysis Summary

Internship Project – Unified Mentor (Data Analyst Role)

Objective: To analyze and interpret **Electric Vehicle (EV) sales data** across Indian states using data analytics and visualization techniques. The goal was to identify state-wise adoption patterns, study monthly trends, understand vehicle categories, generate actionable insights, and highlight the rapid growth of EVs in India.

Tools & Technologies Used:

- Python (Pandas, NumPy, Seaborn, Matplotlib) & Jupyter Notebook
- Data Visualization & AI-powered step-by-step guidance for EDA

Key Steps & Work Done:

Data Collection & Cleaning:

- Loaded and inspected the EV sales dataset.
- Converted and standardized date formats; extracted day, month, and year.
- Handled missing values and ensured consistency in month names.

Exploratory Data Analysis (EDA):

- Analyzed monthly EV sales trends across India.
- Identified top 10 states by total EV sales.
- Visualized EV sales distribution by vehicle category and vehicle type.

Advanced Insights (Optional Forecasting Scope):

- Designed plots to show adoption patterns across states and months.
- Discussed potential for forecasting EV sales using time-series models (Prophet/ARIMA) as a future enhancement.

Insights:

- **Uttar Pradesh** leads EV adoption with **7.3L+ sales**, followed by Maharashtra, Karnataka, and Delhi.
- EV adoption shows a **positive upward trend month-over-month**, indicating increasing market demand.
- **Two-wheelers dominate EV sales**, aligning with affordability and consumer usage patterns.

Conclusion:

This project highlighted **India's growing EV adoption patterns**, providing valuable insights for policymakers and automotive stakeholders.

By analyzing **state-wise trends, category performance, and monthly growth**, the project showcased the ability to transform raw datasets into **actionable business insights**.

This internship project strengthened my expertise in **Python-based data analysis, visualization, and insight generation**, while showcasing my ability to work on **real-world datasets**.

```
import pandas as pd

file = r"C:\Users\ANKITA\Downloads\Electric Vehicle Sales by State in
India.csv"
df = pd.read_csv(file)
df.head()
```

```
     Year Month_Name       Date            State
Vehicle_Class  \
0  2014.0         jan   1/1/2014   Andhra Pradesh       ADAPTED VEHICLE

1  2014.0         jan  1/1/2014  Andhra Pradesh  AGRICULTURAL TRACTOR

2  2014.0         jan  1/1/2014  Andhra Pradesh              AMBULANCE

3  2014.0         jan  1/1/2014  Andhra Pradesh   ARTICULATED VEHICLE

4  2014.0         jan  1/1/2014  Andhra Pradesh                   BUS


  Vehicle_Category Vehicle_Type   EV_Sales_Quantity
0           Others        Others                0.0
1           Others        Others                0.0
2           Others        Others                0.0
3           Others        Others                0.0
4              Bus           Bus                0.0
```

```
df.isnull().sum()
```

```
Year                  0
Month_Name            0
Date                  0
State                 0
Vehicle_Class         0
Vehicle_Category      0
Vehicle_Type          0
EV_Sales_Quantity     0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96845 entries, 0 to 96844
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Year                96845 non-null  float64
 1   Month_Name          96845 non-null  object
 2   Date                96845 non-null  object
 3   State               96845 non-null  object
 4   Vehicle_Class       96845 non-null  object
```

```
 5   Vehicle_Category    96845 non-null  object
 6   Vehicle_Type        96845 non-null  object
 7   EV_Sales_Quantity  96845 non-null  float64
dtypes: float64(2), object(6)
memory usage: 5.9+ MB
```

```
df.describe()
```

```
                Year   EV_Sales_Quantity
count   96845.000000        96845.000000
mean     2018.622768           37.108896
std         2.895581          431.566675
min      2014.000000            0.000000
25%      2016.000000            0.000000
50%      2019.000000            0.000000
75%      2021.000000            0.000000
max      2024.000000        20584.000000
```

```
df.duplicated().sum()
```

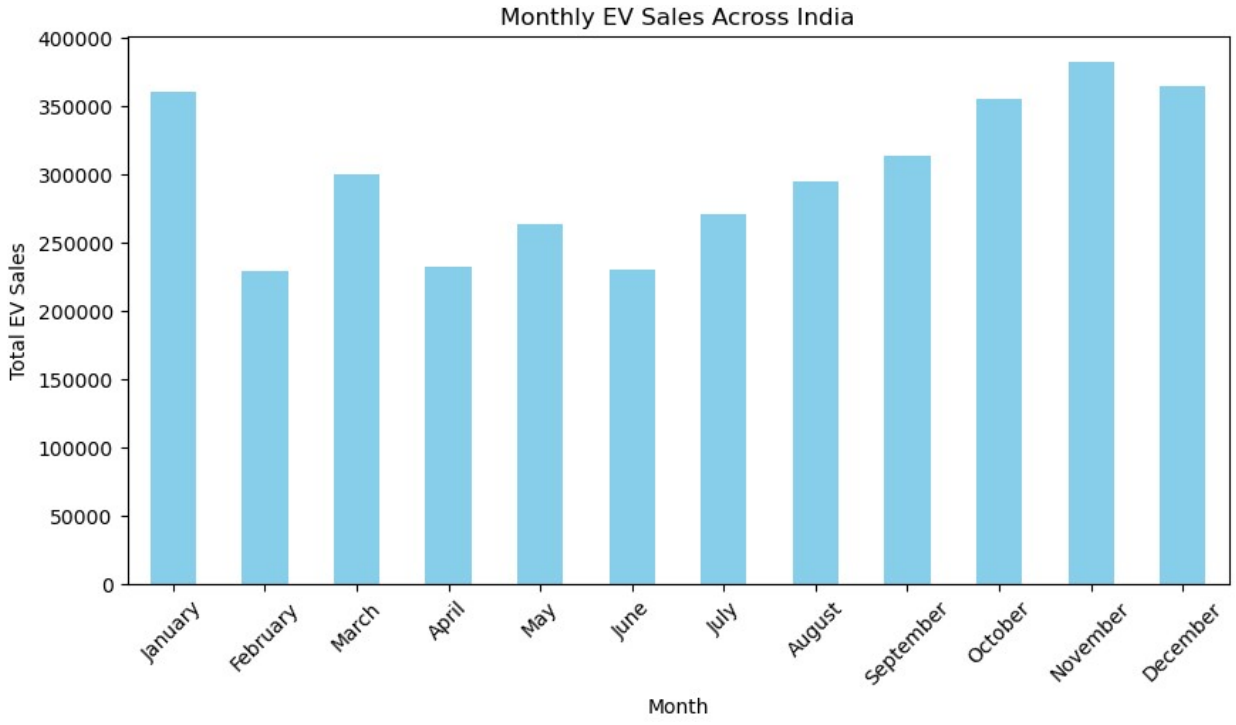
```
0
```

```
df['Date'] = pd.to_datetime(df['Date'], format="%d/%m/%Y",
errors='coerce')
df.head()
```

```
     Year Month Name        Date            State          Vehicle Class
\
0  2014.0         jan 2014-01-01  Andhra Pradesh        ADAPTED VEHICLE

1  2014.0         jan 2014-01-01  Andhra Pradesh  AGRICULTURAL TRACTOR

2  2014.0         jan 2014-01-01  Andhra Pradesh               AMBULANCE

3  2014.0         jan 2014-01-01  Andhra Pradesh    ARTICULATED VEHICLE

4  2014.0         jan 2014-01-01  Andhra Pradesh                    BUS


  Vehicle Category Vehicle Type   EV_Sales_Quantity
0           Others       Others                 0.0
1           Others       Others                 0.0
2           Others       Others                 0.0
3           Others       Others                 0.0
4              Bus          Bus                 0.0
```

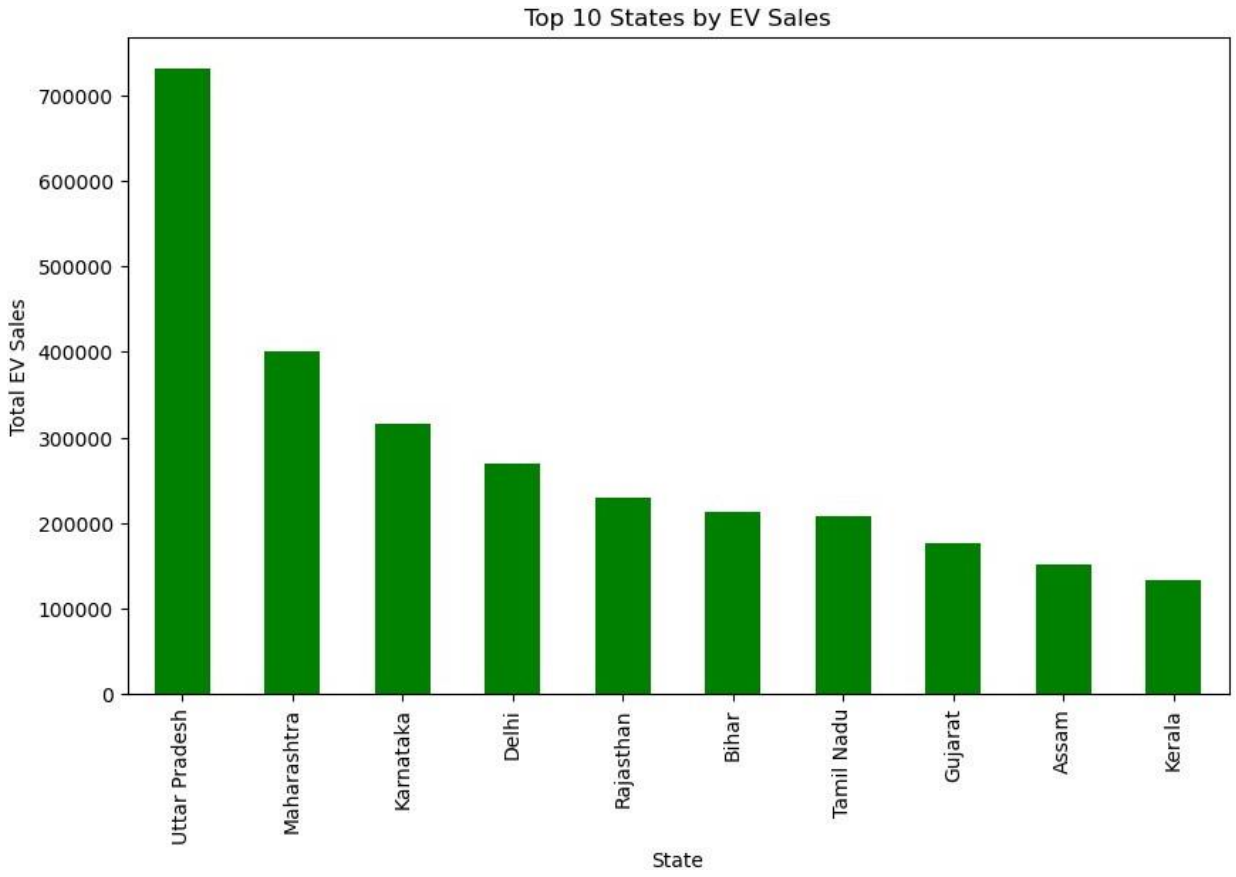
```
df['day'] = pd.to_datetime(df['Date'], dayfirst=True).dt.day
df['month'] = pd.to_datetime(df['Date'], dayfirst=True).dt.month
df['year'] = pd.to_datetime(df['Date'], dayfirst=True).dt.year

df['Month_Name'] = df['Month_Name'].str.strip().str.capitalize()
month_map = {
```

```
    'Jan': 'January', 'Feb': 'February', 'Mar': 'March', 'Apr':
'April',
    'May': 'May', 'Jun': 'June', 'Jul': 'July', 'Aug': 'August',
    'Sep': 'September', 'Oct': 'October', 'Nov': 'November', 'Dec':
'December'
}
df['Month_Name'] = df['Month_Name'].replace(month_map)

monthly_sales = (
    df.groupby('Month_Name')['EV_Sales_Quantity']
    .sum()
    .reindex([
        'January','February','March','April','May','June',
        'July','August','September','October','November','December'
    ])
)
plt.figure(figsize=(10,5))
monthly_sales.plot(kind='bar', color='skyblue')
plt.title("Monthly EV Sales Across India")
plt.xlabel("Month")
plt.ylabel("Total EV Sales")
plt.xticks(rotation=45)
plt.show()
```

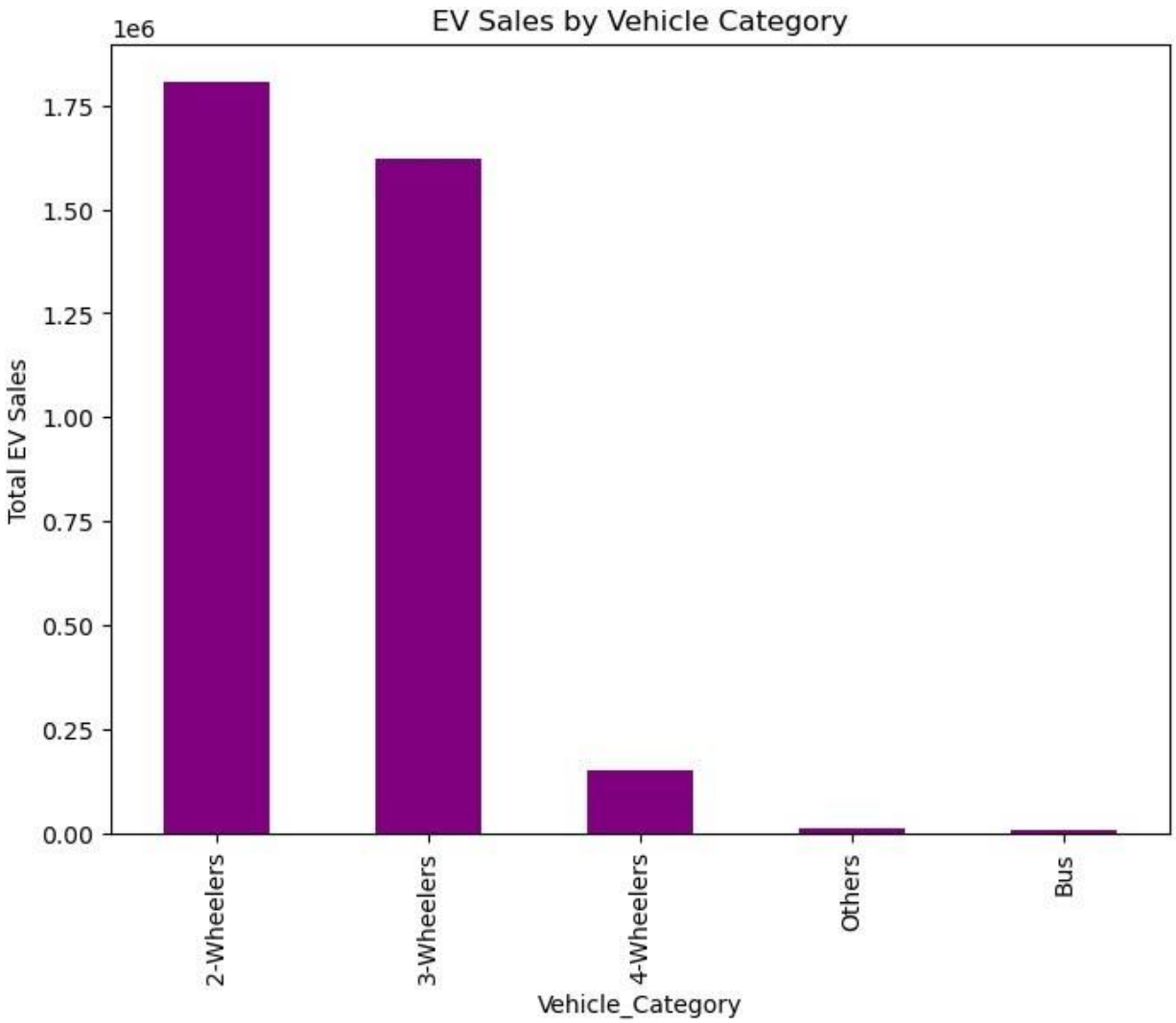



```
state_sales = df.groupby('State')
['EV_Sales_Quantity'].sum().sort_values(ascending=False).head(10)
```

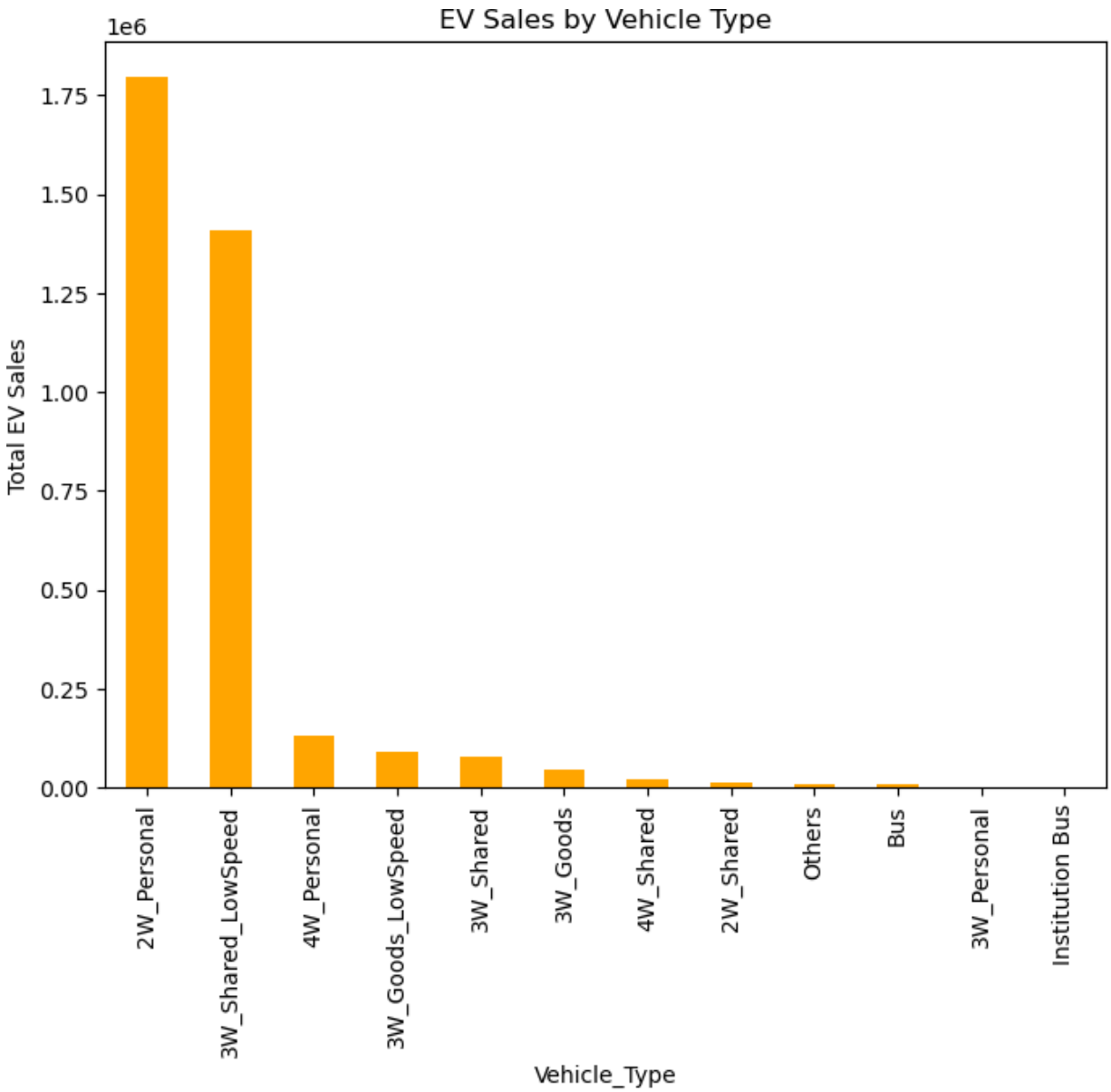
```
plt.figure(figsize=(10,6))
state_sales.plot(kind='bar', color='green')
plt.title("Top 10 States by EV Sales")
plt.ylabel("Total EV Sales")
plt.show()
```



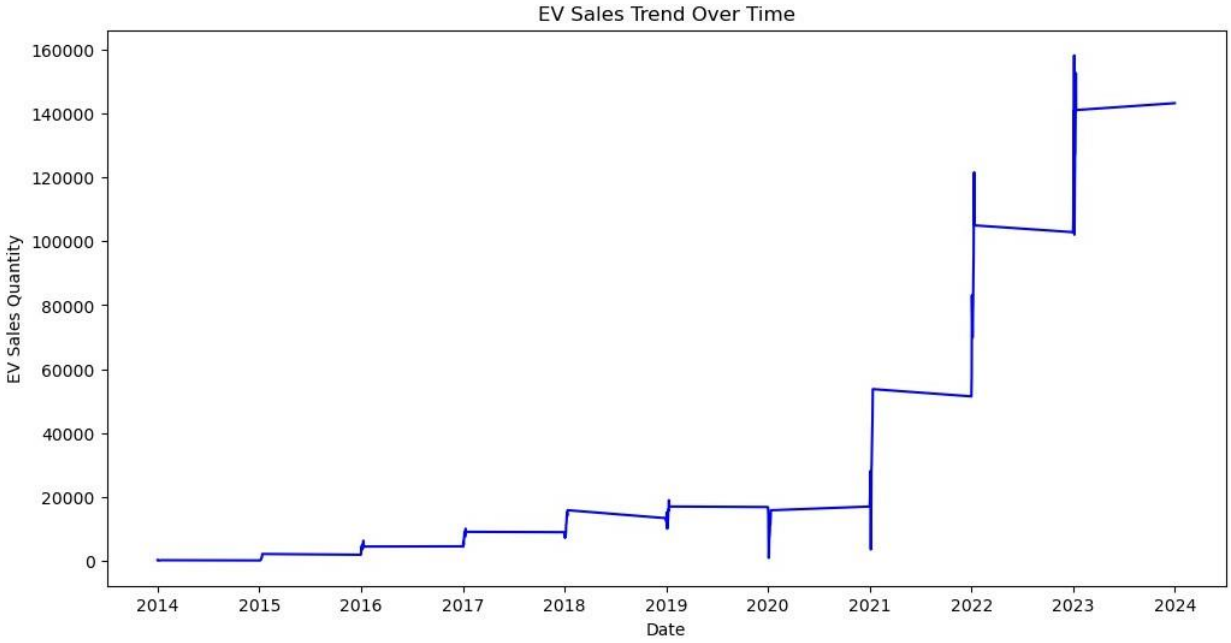

```
category_sales = df.groupby('Vehicle_Category')
['EV_Sales_Quantity'].sum().sort_values(ascending=False)

plt.figure(figsize=(8,6))
category_sales.plot(kind='bar', color='purple')
plt.title("EV Sales by Vehicle Category")
plt.ylabel("Total EV Sales")
plt.show()
```




```
type_sales = df.groupby('Vehicle_Type')
['EV_Sales_Quantity'].sum().sort_values(ascending=False)

plt.figure(figsize=(8,6))
type_sales.plot(kind='bar', color='orange')
plt.title("EV Sales by Vehicle Type")
plt.ylabel("Total EV Sales")
plt.show()
```




```
plt.figure(figsize=(12,6))
plt.plot(ts['Date'], ts['EV_Sales_Quantity'], color='blue')
plt.title("EV Sales Trend Over Time")
plt.xlabel("Date")
plt.ylabel("EV Sales Quantity")
plt.show()
```


EV Sales Trend Over Time
EV Sales Quantity
Date
0
20000
40000
60000
80000
100000
120000
140000
160000
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024