In [ ]:
```
EV ANALYTICS BY ADITYA ANSUMAN ROUTRAY
```

'''Few Key Points:

1. Fiscal Year: The fiscal year is a one-year period used for financial reporting and budgeting, starting on April 1st and ending on March 31st of the following year in India.
2. Penetration Rate: This metric represents the percentage of total vehicles that are electric within a specific region or category. It is calculated as: Penetration Rate = (Electric Vehicles Sold / Total Vehicles Sold) * 100
   This indicates the adoption level of electric vehicles.
3. Compound Annual Growth Rate (CAGR): CAGR measures the mean annual growth rate over a specified period longer than one year. It is calculated as: CAGR = [(Ending Value / Beginning Value) ** 1/n] -1 '''

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import spearmanr
```

In [4]:
```
df1=pd.read_csv(r"C:\Users\adian\Downloads\study\python\EV project\datasets\dim_dat
```

In [5]:
```
df2=pd.read_csv(r"C:\Users\adian\Downloads\study\python\EV project\datasets\electri
```

In [3]:
```
df3=pd.read_csv(r"C:\Users\adian\Downloads\study\python\EV project\datasets\electri
```

In [ ]:
```
# knowing the datasets
```

In [390…
```
df1.shape
```

Out[390…
```
(36, 3)
```

In [389…
```
df2.shape
```

Out[389…
```
(816, 4)
```

In [391…
```
df3.shape
```

Out[391…
```
(2445, 5)
```

In [122…
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   date         36 non-null     object
 1   fiscal_year  36 non-null     int64 
 2   quarter      36 non-null     object
dtypes: int64(1), object(2)
memory usage: 992.0+ bytes
```

In [392…
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   date                   816 non-null    object
 1   vehicle_category       816 non-null    object
 2   maker                  816 non-null    object
 3   electric_vehicles_sold  816 non-null    int64 
dtypes: int64(1), object(3)
memory usage: 25.6+ KB
```

In [124…
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2445 entries, 0 to 2444
Data columns (total 5 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   date                   2445 non-null   object
 1   state                  2445 non-null   object
 2   vehicle_category       2445 non-null   object
 3   electric_vehicles_sold  2445 non-null   int64 
 4   total_vehicles_sold    2445 non-null   int64 
dtypes: int64(2), object(3)
memory usage: 95.6+ KB
```

In [150…
```
df1.head()
```

Out[150…

	date	fiscal_year	quarter
0	1-Apr-21	2022	Q1
1	1-May-21	2022	Q1
2	1-Jun-21	2022	Q1
3	1-Jul-21	2022	Q2
4	1-Aug-21	2022	Q2

In [151…
```
df2.tail()
```

Out[151...

	date	vehicle_category	maker	electric_vehicles_sold
811	2024-03-01	2-Wheelers	BGAUSS	3070
812	2024-03-01	2-Wheelers	BATTRE ELECTRIC	625
813	2024-03-01	2-Wheelers	KINETIC GREEN	3915
814	2024-03-01	2-Wheelers	REVOLT	585
815	2024-03-01	2-Wheelers	OTHERS	10579

In [152...
```
df3.tail()
```

Out[152...

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
2440	2024-03-01	Mizoram	2-Wheelers	58	1932
2441	2024-03-01	DNH and DD	2-Wheelers	25	780
2442	2024-03-01	Manipur	2-Wheelers	13	1394
2443	2024-03-01	Andaman & Nicobar Island	2-Wheelers	2	447
2444	2024-03-01	Nagaland	2-Wheelers	2	1180

In [154...
```
df1.to_csv('opdf1.csv',index=False)
df2.to_csv('opdf2.csv',index=False)
df3.to_csv('opdf3.csv',index=False)
```

In [47]:
```
pd.options.display.float_format = '{:.2f}'.format
```

In [26]:
```
# find null values if any
df1.isnull().sum()
```

Out[26]:
```
date           0
fiscal_year    0
quarter        0
dtype: int64
```

In [27]:
```
df2.isnull().sum()
```

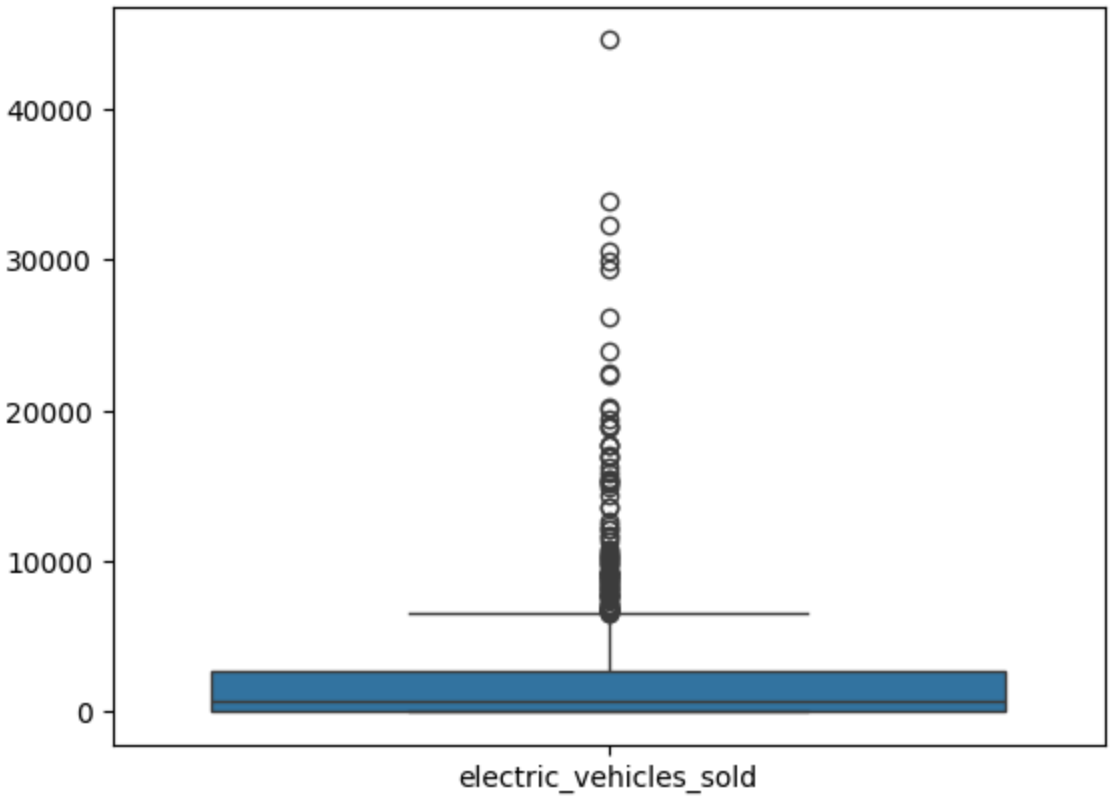
Out[27]:
```
date                   0
vehicle_category       0
maker                  0
electric_vehicles_sold 0
dtype: int64
```

In [28]:
```
df3.isnull().sum()
```

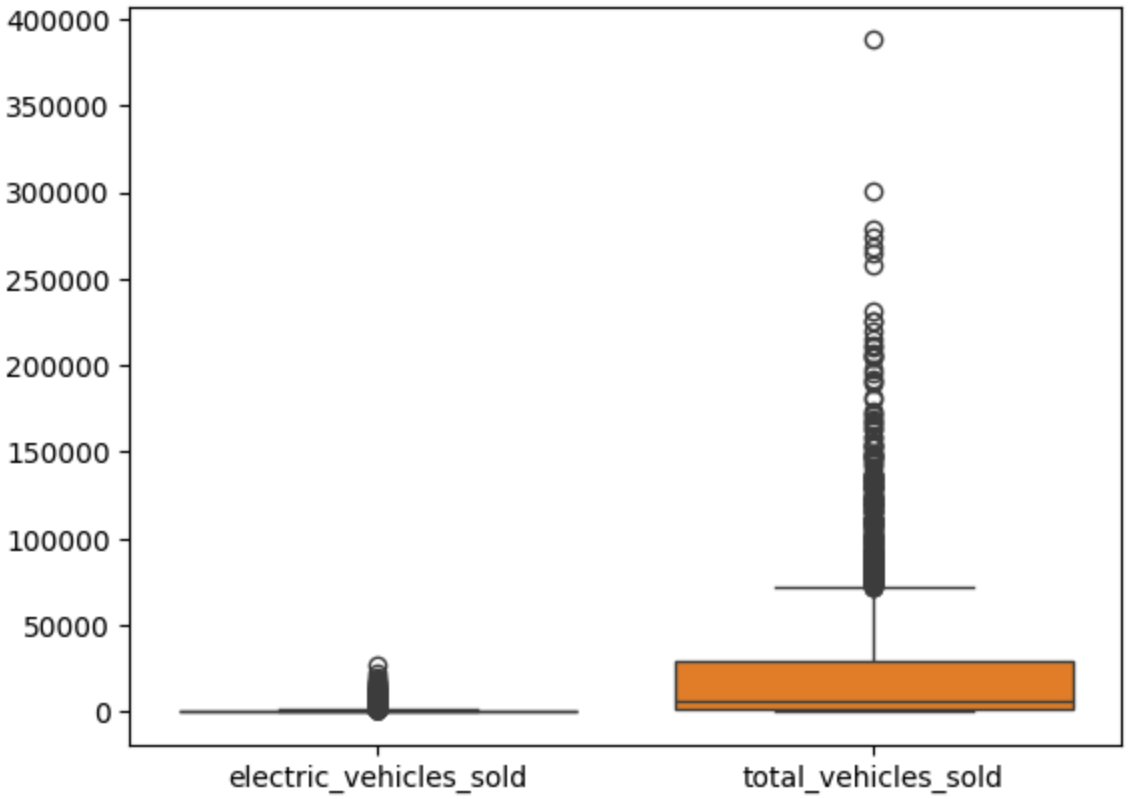
Out[28]:
```
date                    0
state                   0
vehicle_category        0
electric_vehicles_sold  0
total_vehicles_sold     0
dtype: int64
```

In [34]:
```
# finding out outliers
```

In [36]:
```
sns.boxplot(data=df2)
plt.show()
```



In [37]:
```
sns.boxplot(data=df3)
plt.show()
```



all values are important

In [38]:
```
                                                    #preliminary research questions
```

In [39]:
```
#1.List the top 3 and bottom 3 makers for the fiscal years 2023 and 2024 in terms o
```

convert the date column to datetime format

In [52]:
```
df2['date'] = pd.to_datetime(df2['date'])
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\1196262380.py:1: UserWarning: Coul
d not infer format, so each element will be parsed individually, falling back to `da
teutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df2['date'] = pd.to_datetime(df2['date'])
```

In [53]:
```
df2["vehicle_category"].value_counts()
```

Out[53]:
```
vehicle_category
2-Wheelers    456
4-Wheelers    360
Name: count, dtype: int64
```

In [54]:
```
df2wh_23=df2[(df2["vehicle_category"]=="2-Wheelers") & (df2['date']>'2022-03-31') &
df2wh_24=df2[(df2["vehicle_category"]=="2-Wheelers") & (df2['date']>'2023-03-31') &
```

In [55]:
```
total_sales_2023=df2wh_23.groupby("maker")["electric_vehicles_sold"].sum().sort_val
```

In [56]:
```
top3_2023=total_sales_2023.head(3)
bottom3_2023=total_sales_2023.tail(3)
```

In [57]:
```
top3_2023
```

Out[57]:
```
maker
OLA ELECTRIC     152583
OKINAWA           96945
HERO ELECTRIC     88993
Name: electric_vehicles_sold, dtype: int64
```

In [58]:
```
bottom3_2023
```

Out[58]:
```
maker
PURE EV     11556
BEING       11018
JITENDRA     8563
Name: electric_vehicles_sold, dtype: int64
```

In [59]:
```
total_sales_2024=df2wh_24.groupby("maker")["electric_vehicles_sold"].sum().sort_val
```

In [60]:
```
top3_2024=total_sales_2024.head(3)
bottom3_2024=total_sales_2024.tail(3)
```

In [61]:
```
top3_2024
```

Out[61]:
```
maker
OLA ELECTRIC    322489
TVS             180743
ATHER           107552
Name: electric_vehicles_sold, dtype: int64
```

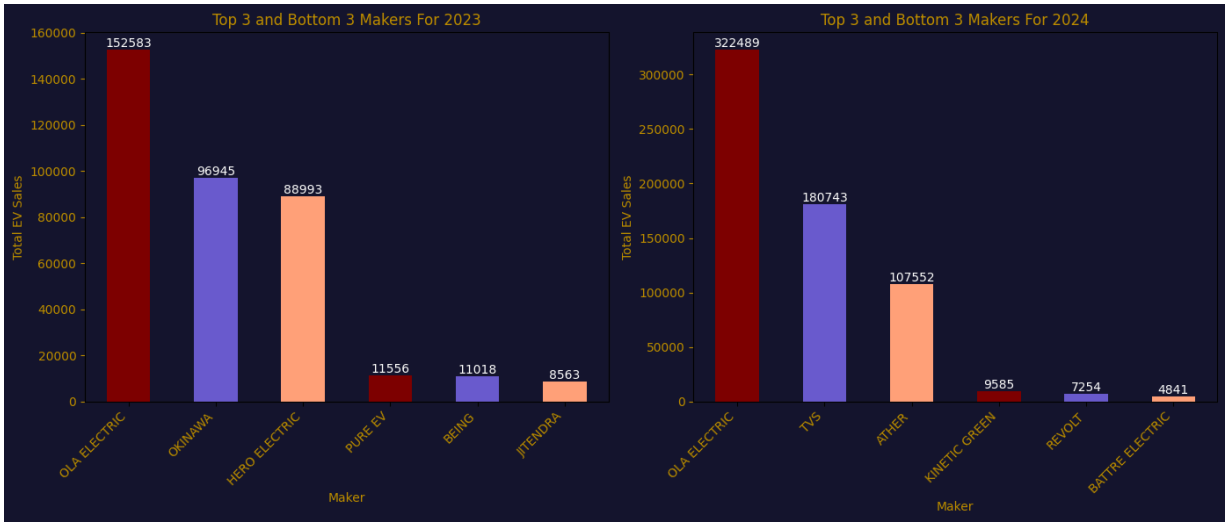
In [62]:
```
bottom3_2024
```

Out[62]:
```
maker
KINETIC GREEN      9585
REVOLT             7254
BATTRE ELECTRIC    4841
Name: electric_vehicles_sold, dtype: int64
```

In [63]:
```
top3_2023.to_csv('top3_2023.csv')
top3_2024.to_csv('top3_2024.csv')
bottom3_2023.to_csv('bottom3_2023.csv')
bottom3_2024.to_csv('bottom3_2024.csv')
```

In [567…
```
# Increase figure size to accommodate side-by-side subplots
fig, axs = plt.subplots(1, 2, figsize=(14, 6))  # 1 row, 2 columns
# Set figure and axes background colors
fig.patch.set_facecolor('#18182F')
axs[0].set_facecolor('#18182F')
axs[1].set_facecolor('#18182F')
# Plot for 2023 data
combined_2023 = pd.concat([top3_2023, bottom3_2023])
combined_2023.plot(kind='bar', color=['maroon','slateblue','#FFA07A'], ax=axs[0])

# Add data labels to 2023 plot (as whole numbers)
for index, value in enumerate(combined_2023):
    axs[0].text(index, value, str(int(value)), color='white', ha='center', va='bott

# Customize 2023 plot
```

```
axs[0].set_title('Top 3 and Bottom 3 Makers For 2023', color='#BF9000')
axs[0].set_xlabel('Maker', color='#BF9000')
axs[0].set_ylabel('Total EV Sales', color='#BF9000')
axs[0].set_xticks(range(len(combined_2023)))
axs[0].set_xticklabels(combined_2023.index, rotation=45, ha='right', fontsize=10, c
axs[0].tick_params(axis='y', colors='#BF9000')

# Plot for 2024 data
combined_2024 = pd.concat([top3_2024, bottom3_2024])
combined_2024.plot(kind='bar', color=['maroon','slateblue','#FFA07A'], ax=axs[1])

# Add data labels to 2024 plot (as whole numbers)
for index, value in enumerate(combined_2024):
    axs[1].text(index, value, str(int(value)), color='white', ha='center', va='bott

# Customize 2024 plot
axs[1].set_title('Top 3 and Bottom 3 Makers For 2024', color='#BF9000')
axs[1].set_xlabel('Maker', color='#BF9000')
axs[1].set_ylabel('Total EV Sales', color='#BF9000')
axs[1].set_xticks(range(len(combined_2024)))
axs[1].set_xticklabels(combined_2024.index, rotation=45, ha='right', fontsize=10, c
axs[1].tick_params(axis='y', colors='#BF9000')

# Adjust layout to avoid overlap
plt.tight_layout()

plt.show()
```



In [120…
```
#2.Identify the top 5 states with the highest penetration rate in 2-wheeler and 4-w
#Penetration Rate =  (Electric Vehicles Sold / Total Vehicles Sold) * 100
```

In [64]:
```
df3.head()
```

Out[64]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	1-Apr-21	Sikkim	2-Wheelers	0	398
1	1-Apr-21	Sikkim	4-Wheelers	0	361
2	1-May-21	Sikkim	2-Wheelers	0	113
3	1-May-21	Sikkim	4-Wheelers	0	98
4	1-Jun-21	Sikkim	2-Wheelers	0	229

In [66]:
```
df3['date'] = pd.to_datetime(df3['date'])
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\578405167.py:1: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df3['date'] = pd.to_datetime(df3['date'])
```

In [67]:
```
fy24_df3=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [68]:
```
fy24_df3
```

Out[68]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
48	2023-04-01	Sikkim	2-Wheelers	0	465
49	2023-04-01	Sikkim	4-Wheelers	0	439
50	2023-05-01	Sikkim	2-Wheelers	0	507
51	2023-05-01	Sikkim	4-Wheelers	0	448
52	2023-06-01	Sikkim	2-Wheelers	0	517
...	...	...	...	...	...
2440	2024-03-01	Mizoram	2-Wheelers	58	1932
2441	2024-03-01	DNH and DD	2-Wheelers	25	780
2442	2024-03-01	Manipur	2-Wheelers	13	1394
2443	2024-03-01	Andaman & Nicobar Island	2-Wheelers	2	447
2444	2024-03-01	Nagaland	2-Wheelers	2	1180

814 rows × 5 columns

In [69]:
```
fy24_df3_2wh=fy24_df3[fy24_df3['vehicle_category']=='2-Wheelers']
fy24_df3_4wh=fy24_df3[fy24_df3['vehicle_category']=='4-Wheelers']
```

In [70]:
```
a2=fy24_df3_2wh.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].
a4=fy24_df3_4wh.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].
```

In [73]:
```
pd.options.display.float_format = '{:.2f}'.format
```

In [74]:
```
a2['penetration_rate']=100*(a2['electric_vehicles_sold']/a2['total_vehicles_sold'])
```

In [75]:
```
a4['penetration_rate']=100*(a4['electric_vehicles_sold']/a4['total_vehicles_sold'])
```

In [76]:
```
top5_2wh=a2.sort_values(by="penetration_rate",ascending=False).head()
top5_4wh=a4.sort_values(by="penetration_rate",ascending=False).head()
```

In [77]:
```
top5_2wh
```

Out[77]:

state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
Goa	9768	54290	17.99
Kerala	64769	478887	13.52
Karnataka	148111	1279767	11.57
Maharashtra	183052	1817343	10.07
Delhi	38094	405218	9.40

In [78]:
```
top5_4wh
```

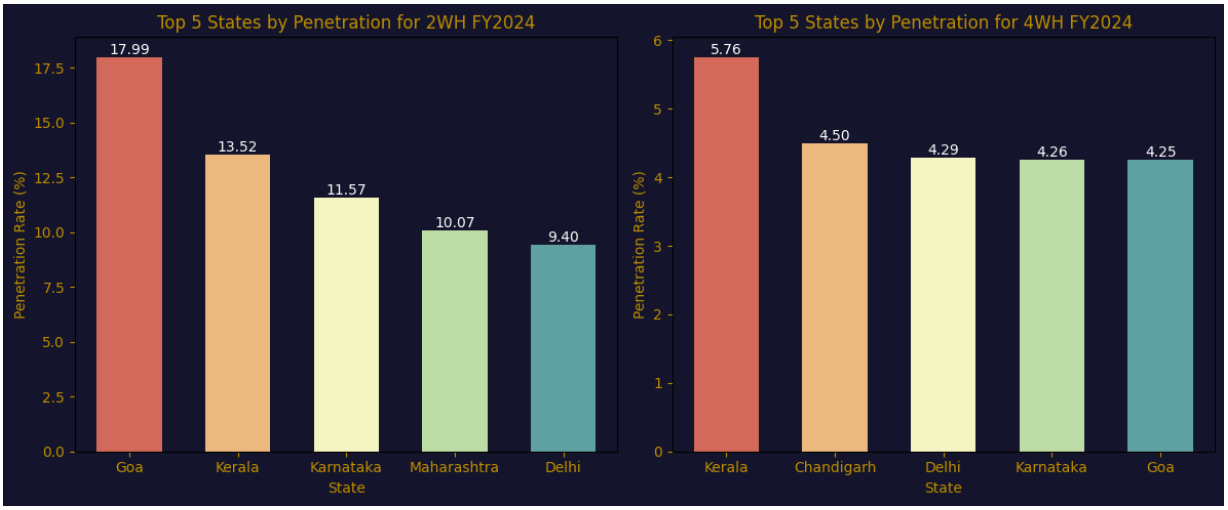
Out[78]:

state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
Kerala	9169	159227	5.76
Chandigarh	1020	22651	4.50
Delhi	8630	201130	4.29
Karnataka	12878	302221	4.26
Goa	1031	24234	4.25

In [79]:
```
top5_2wh.to_csv('top5_2wh.csv')
top5_4wh.to_csv('top5_4wh.csv')
```

In [571…
```
fig, axs = plt.subplots(1, 2, figsize=(12, 5))
fig.patch.set_facecolor('#18182F')  # Dark blue background

# 2-Wheelers Plot
ax1 = sns.barplot(x='state', y='penetration_rate', data=top5_2wh.head(5),hue='state
ax1.set_facecolor('#18182F')  # Bar plot background
for bar in ax1.patches:
    ax1.text(
        bar.get_x() + bar.get_width() / 2,  # X coordinate (center of the bar)
        bar.get_height(),                   # Y coordinate (height of the bar)
        '{:.2f}'.format(bar.get_height()),  # Label text (formatted value)
        ha='center', va='bottom',            # Align the text in the center horizon
        color='white'                       # Set label color to white
    )
ax1.set_title('Top 5 States by Penetration for 2WH FY2024', color='#BF9000')  # Tit
ax1.set_xlabel('State', color='#BF9000')  # X-axis label color
ax1.set_ylabel('Penetration Rate (%)', color='#BF9000')  # Y-axis label color
ax1.tick_params(axis='x', colors='#BF9000')  # X ticks color
ax1.tick_params(axis='y', colors='#BF9000')  # Y ticks color

# 4-Wheelers Plot
ax2 = sns.barplot(x='state', y='penetration_rate', data=top5_4wh.head(5),hue='state
ax2.set_facecolor('#18182F')  # Bar plot background
```

```
for bar in ax2.patches:
    ax2.text(
        bar.get_x() + bar.get_width() / 2,  # X coordinate (center of the bar)
        bar.get_height(),                   # Y coordinate (height of the bar)
        '{:.2f}'.format(bar.get_height()),  # Label text (formatted value)
        ha='center', va='bottom',            # Align the text in the center horizon
       color='white'                        # Set label color to white
    )
ax2.set_title('Top 5 States by Penetration for 4WH FY2024',color='#BF9000')  # Titl
ax2.set_xlabel('State', color='#BF9000')  # X-axis label color
ax2.set_ylabel('Penetration Rate (%)', color='#BF9000')  # Y-axis label color
ax2.tick_params(axis='x', colors='#BF9000')  # X ticks color
ax2.tick_params(axis='y', colors='#BF9000')  # Y ticks color

# Adjust layout for better spacing
plt.tight_layout()
plt.show()
```



In [104…
```
# 3.List the states with negative penetration (decline) in EV sales from 2022 to 20
```

In [442…
```
df3
```

Out[442...

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	2021-04-01	Sikkim	2-Wheelers	0	398
1	2021-04-01	Sikkim	4-Wheelers	0	361
2	2021-05-01	Sikkim	2-Wheelers	0	113
3	2021-05-01	Sikkim	4-Wheelers	0	98
4	2021-06-01	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
2440	2024-03-01	Mizoram	2-Wheelers	58	1932
2441	2024-03-01	DNH and DD	2-Wheelers	25	780
2442	2024-03-01	Manipur	2-Wheelers	13	1394
2443	2024-03-01	Andaman & Nicobar Island	2-Wheelers	2	447
2444	2024-03-01	Nagaland	2-Wheelers	2	1180

2445 rows × 5 columns

In [443...
```
df3['date'] = pd.to_datetime(df3['date'], format='%Y-%m-%d')
```

In [444...
```
fy22_df3=df3[(df3['date']>'2021-03-31') & (df3['date']<'2022-04-01')]
```

In [445...
```
fy24_df3=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [446...
```
ev_sales_22=fy22_df3.groupby('state')['electric_vehicles_sold'].sum()
```

In [447...
```
ev_sales_24=fy24_df3.groupby('state')['electric_vehicles_sold'].sum()
```

In [448...
```
a22=fy22_df3.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].sum
a24=fy24_df3.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].sum
```

In [449...
```
a22.rename(columns={"electric_vehicles_sold":"EV_sold_2022","total_vehicles_sold":"
a24.rename(columns={"electric_vehicles_sold":"EV_sold_2024","total_vehicles_sold":"
```

In [450...
```
a=pd.concat([a22,a24],axis=1,join="inner")
```

In [451…

```
a
```

Out[451...

state	EV_sold_2022	TV_sold_2022	EV_sold_2024	TV_sold_2024
Andaman & Nicobar Island	22	5148	35	7203
Andhra Pradesh	13928	772748	33183	782865
Arunachal Pradesh	0	19929	31	27892
Assam	730	379450	3497	547626
Bihar	4829	892873	15069	1132703
Chandigarh	411	36954	2877	45147
Chhattisgarh	4534	390272	28540	503068
DNH and DD	35	12413	198	16400
Delhi	16535	401540	46724	606348
Goa	1778	48372	10799	78524
Gujarat	18026	1094872	84359	1590987
Haryana	5926	528591	11793	732029
Himachal Pradesh	443	98266	1048	117084
Jammu and Kashmir	1434	133943	2283	139359
Jharkhand	2713	411613	7830	495011
Karnataka	43111	1007894	160989	1581988
Kerala	13639	689575	73938	638114
Ladakh	12	2911	31	3206
Madhya Pradesh	7916	967179	43223	1286182
Maharashtra	48374	1667002	197169	2293994
Manipur	25	36129	126	18422
Meghalaya	4	22193	133	36628
Mizoram	0	19439	275	27422
Nagaland	1	12852	9	16972
Odisha	9498	479527	39118	618149
Puducherry	734	42945	3098	57692
Punjab	4528	443232	11198	574486
Rajasthan	20087	880985	66444	1300476
Sikkim	0	8897	0	10518

state	EV_sold_2022	TV_sold_2022	EV_sold_2024	TV_sold_2024
Tamil Nadu	36863	1345017	94314	1716940
Tripura	28	37735	304	46447
Uttar Pradesh	10222	2497288	57758	2932347
Uttarakhand	2079	173331	6336	233111
West Bengal	2685	860709	16864	961909

In [452…
```
a['pen22']=100*(a['EV_sold_2022']/a['TV_sold_2022'])
a['pen24']=100*(a['EV_sold_2024']/a['TV_sold_2024'])
```

In [453…
```
a['pen_growth']=a['pen24']-a['pen22']
```

In [454…
```
a['sales_growth']=a['EV_sold_2024']-a['EV_sold_2022']
```

In [455…
```
a_final=a[['sales_growth','pen_growth']]
```

In [456…
```
a_final
```

Out[456...

state	sales_growth	pen_growth
Andaman & Nicobar Island	13	0.06
Andhra Pradesh	19255	2.44
Arunachal Pradesh	31	0.11
Assam	2767	0.45
Bihar	10240	0.79
Chandigarh	2466	5.26
Chhattisgarh	24006	4.51
DNH and DD	163	0.93
Delhi	30189	3.59
Goa	9021	10.08
Gujarat	66333	3.66
Haryana	5867	0.49
Himachal Pradesh	605	0.44
Jammu and Kashmir	849	0.57
Jharkhand	5117	0.92
Karnataka	117878	5.90
Kerala	60299	9.61
Ladakh	19	0.55
Madhya Pradesh	35307	2.54
Maharashtra	148795	5.69
Manipur	101	0.61
Meghalaya	129	0.35
Mizoram	275	1.00
Nagaland	8	0.05
Odisha	29620	4.35
Puducherry	2364	3.66
Punjab	6670	0.93
Rajasthan	46357	2.83
Sikkim	0	0.00

state	sales_growth	pen_growth
Tamil Nadu	57451	2.75
Tripura	276	0.58
Uttar Pradesh	47536	1.56
Uttarakhand	4257	1.52
West Bengal	14179	1.44

In [465…
```
a_final.to_csv('a_final.csv')
```

There are no negative penetration or sales from 2022 to 2024

In [ ]:
```
#4.What are the quarterly trends based on sales volume for the top 5 EV makers (4-w
```

In [481…
```
df2["vehicle_category"].value_counts()
```

Out[481…
```
vehicle_category
2-Wheelers    456
4-Wheelers    360
Name: count, dtype: int64
```

In [482…
```
IVD=df2[df2["vehicle_category"]=="4-Wheelers"]
```

In [483…
```
IVD
```

Out[483…

	date	vehicle_category	maker	electric_vehicles_sold
9	01-Apr-21	4-Wheelers	BYD India	0
10	01-Apr-21	4-Wheelers	PCA Automobiles	0
11	01-Apr-21	4-Wheelers	BMW India	0
12	01-Apr-21	4-Wheelers	Volvo Auto India	0
13	01-Apr-21	4-Wheelers	KIA Motors	0
...	...	...	...	...
364	01-Mar-24	4-Wheelers	PCA Automobiles	130
365	01-Mar-24	4-Wheelers	BMW India	55
366	01-Mar-24	4-Wheelers	Mercedes -Benz AG	31
367	01-Mar-24	4-Wheelers	Volvo Auto India	39
368	01-Mar-24	4-Wheelers	KIA Motors	26

360 rows × 4 columns

In [484…
```
IVD['date'] = pd.to_datetime(IVD['date'], format='%d-%b-%y')
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\3983220184.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  IVD['date'] = pd.to_datetime(IVD['date'], format='%d-%b-%y')
```

In [485…
```
IVD['date'] = IVD['date'].dt.strftime('%Y-%m-%d')
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\2308990861.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  IVD['date'] = IVD['date'].dt.strftime('%Y-%m-%d')
```

In [486…
```
fy22_24_IVD=IVD[(IVD['date']>'2021-03-31') & (IVD['date']<'2024-04-01')]
```

In [487…
```
fy22_24_IVD
```

Out[487…

	date	vehicle_category	maker	electric_vehicles_sold
9	2021-04-01	4-Wheelers	BYD India	0
10	2021-04-01	4-Wheelers	PCA Automobiles	0
11	2021-04-01	4-Wheelers	BMW India	0
12	2021-04-01	4-Wheelers	Volvo Auto India	0
13	2021-04-01	4-Wheelers	KIA Motors	0
...	...	...	...	...
364	2024-03-01	4-Wheelers	PCA Automobiles	130
365	2024-03-01	4-Wheelers	BMW India	55
366	2024-03-01	4-Wheelers	Mercedes -Benz AG	31
367	2024-03-01	4-Wheelers	Volvo Auto India	39
368	2024-03-01	4-Wheelers	KIA Motors	26

360 rows × 4 columns

In [488…
```
Q1fy22=IVD[(IVD['date']>'2021-03-31') & (IVD['date']<'2021-06-01')]
Q2fy22=IVD[(IVD['date']>'2021-05-31') & (IVD['date']<'2021-09-01')]
Q3fy22=IVD[(IVD['date']>'2021-08-31') & (IVD['date']<'2022-01-01')]
Q4fy22=IVD[(IVD['date']>'2021-12-31') & (IVD['date']<'2022-04-01')]
```

In [489…
```
Q1fy23=IVD[(IVD['date']>'2022-03-31') & (IVD['date']<'2022-06-01')]
Q2fy23=IVD[(IVD['date']>'2022-05-31') & (IVD['date']<'2022-09-01')]
Q3fy23=IVD[(IVD['date']>'2022-08-31') & (IVD['date']<'2023-01-01')]
Q4fy23=IVD[(IVD['date']>'2022-12-31') & (IVD['date']<'2023-04-01')]
```

In [490…
```
Q1fy24=IVD[(IVD['date']>'2023-03-31') & (IVD['date']<'2023-06-01')]
Q2fy24=IVD[(IVD['date']>'2023-05-31') & (IVD['date']<'2023-09-01')]
Q3fy24=IVD[(IVD['date']>'2023-08-31') & (IVD['date']<'2024-01-01')]
Q4fy24=IVD[(IVD['date']>'2023-12-31') & (IVD['date']<'2024-04-01')]
```

In [491…
```
Top5_maker = fy22_24_IVD.groupby("maker")["electric_vehicles_sold"].sum().sort_valu
```

Use Top5_maker to filter and group in all quarters top5_sales_Q1fy22 = Q1fy22[Q1fy22['maker'].isin(Top5_maker.index)] Q1fy22_top5 = top5_sales_Q1fy22.groupby("maker")["electric_vehicles_sold"].sum() we can combine these two to write Q1fy22_top5 = Q1fy22[Q1fy22['maker'].isin(Top5_maker.index)].groupby("maker")["electric_vehicles_sold"].sum()

In [492…
```
Top5_maker
```

Out[492…
```
maker
Tata Motors            88935
Mahindra & Mahindra    41193
MG Motor               13753
BYD India               2419
Hyundai Motor           2076
Name: electric_vehicles_sold, dtype: int64
```

In [493…
```
Q1fy22_top5 = Q1fy22[Q1fy22['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q2fy22_top5 = Q2fy22[Q2fy22['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q3fy22_top5 = Q3fy22[Q3fy22['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q4fy22_top5 = Q4fy22[Q4fy22['maker'].isin(Top5_maker.index)].groupby("maker")["elec

Q1fy23_top5 = Q1fy23[Q1fy23['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q2fy23_top5 = Q2fy23[Q2fy23['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q3fy23_top5 = Q3fy23[Q3fy23['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q4fy23_top5 = Q4fy23[Q4fy23['maker'].isin(Top5_maker.index)].groupby("maker")["elec

Q1fy24_top5 = Q1fy24[Q1fy24['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q2fy24_top5 = Q2fy24[Q2fy24['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q3fy24_top5 = Q3fy24[Q3fy24['maker'].isin(Top5_maker.index)].groupby("maker")["elec
Q4fy24_top5 = Q4fy24[Q4fy24['maker'].isin(Top5_maker.index)].groupby("maker")["elec
```

In [494…
```
Q1fy22_top5
```

Out[494…
```
maker
BYD India                0
Hyundai Motor           15
MG Motor               157
Mahindra & Mahindra    249
Tata Motors            459
Name: electric_vehicles_sold, dtype: int64
```

In [495…
```
Qtr_top5_maker = pd.concat([Q1fy22_top5, Q2fy22_top5, Q3fy22_top5, Q4fy22_top5,
                            Q1fy23_top5, Q2fy23_top5, Q3fy23_top5, Q4fy23_top5,
                            Q1fy24_top5, Q2fy24_top5, Q3fy24_top5, Q4fy24_top5],
                            axis=1)
```

In [496…
```
Qtr_top5_maker.columns = ['Q1fy22', 'Q2fy22', 'Q3fy22', 'Q4fy22',
                          'Q1fy23', 'Q2fy23', 'Q3fy23', 'Q4fy23',
                          'Q1fy24', 'Q2fy24', 'Q3fy24', 'Q4fy24']
```

In [497…
```
Qtr_top5_maker
```

Out[497…

maker	Q1fy22	Q2fy22	Q3fy22	Q4fy22	Q1fy23	Q2fy23	Q3fy23	Q4fy23	Q1fy24
BYD India	0	0	1	32	45	101	151	623	266
Hyundai Motor	15	38	31	26	36	137	248	155	171
MG Motor	157	663	674	153	358	601	1372	946	616
Mahindra & Mahindra	249	481	1659	1653	1233	2738	4591	5243	8337
Tata Motors	459	1852	4563	5834	3506	6364	8648	9528	3576

In [28]:
```
Qtr_top5_maker.to_csv('Qtr_top5_maker.csv')
```

In [498…
```
Qtr_top5_maker_t = Qtr_top5_maker.transpose()
```

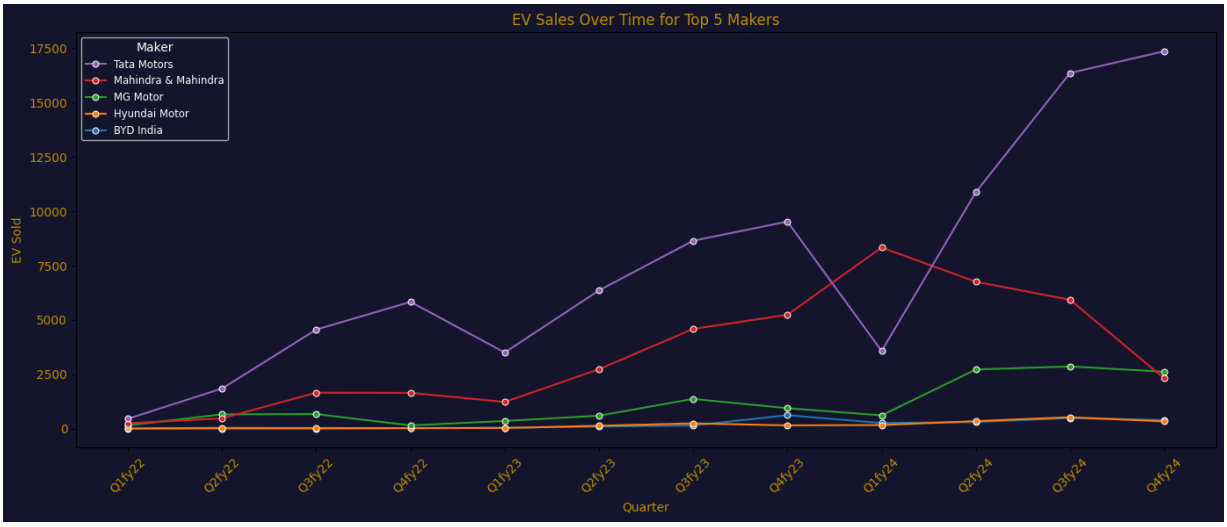
In [30]:
```
Qtr_top5_maker_t
```

Out[30]:

maker	BYD India	Hyundai Motor	MG Motor	Mahindra & Mahindra	Tata Motors
Q1fy22	0	15	157	249	459
Q2fy22	0	38	663	481	1852
Q3fy22	1	31	674	1659	4563
Q4fy22	32	26	153	1653	5834
Q1fy23	45	36	358	1233	3506
Q2fy23	101	137	601	2738	6364
Q3fy23	151	248	1372	4591	8648
Q4fy23	623	155	946	5243	9528
Q1fy24	266	171	616	8337	3576
Q2fy24	309	350	2725	6761	10882
Q3fy24	491	531	2866	5932	16362
Q4fy24	400	338	2622	2316	17361

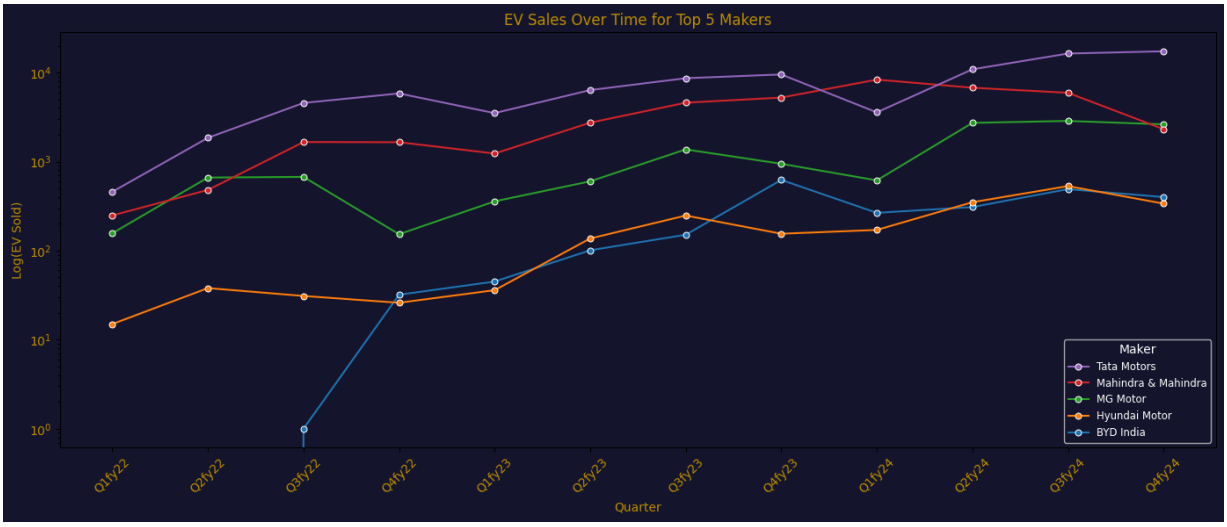
In [573…

```
plt.figure(figsize=(14, 6), facecolor='#18182F')

ax = plt.gca()
ax.set_facecolor('#18182F')

for maker in Qtr_top5_maker_t.columns:
    sns.lineplot(x=Qtr_top5_maker_t.index, y=Qtr_top5_maker_t[maker], label=maker,

plt.title('EV Sales Over Time for Top 5 Makers', color='#BF9000')
plt.xlabel('Quarter',color='#BF9000')
plt.ylabel('EV Sold',color='#BF9000')

plt.xticks(rotation=45, color='#BF9000')
plt.yticks(color='#BF9000')

#Legend editing
handles, labels = plt.gca().get_legend_handles_labels()
order = [4,3,2,1,0]
legend = plt.legend([handles[i] for i in order], [labels[i] for i in order],title='
for text in legend.get_texts():
    text.set_color('white')
legend.get_title().set_color('white')

plt.tight_layout()
plt.show()
```



In [579…

```
plt.figure(figsize=(14, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')

for maker in Qtr_top5_maker_t.columns:
    sns.lineplot(x=Qtr_top5_maker_t.index, y=Qtr_top5_maker_t[maker], label=maker,
ax.set_yscale('log')
plt.title('EV Sales Over Time for Top 5 Makers', color='#BF9000')
plt.xlabel('Quarter', color='#BF9000')
plt.ylabel('Log(EV Sold)', color='#BF9000')

plt.xticks(rotation=45, color='#BF9000')
plt.yticks(color='#BF9000')

#Legend editing
handles, labels = plt.gca().get_legend_handles_labels()
order = [4,3,2,1,0]
legend = plt.legend([handles[i] for i in order], [labels[i] for i in order],title='
for text in legend.get_texts():
    text.set_color('white')
legend.get_title().set_color('white')

plt.tight_layout()
plt.show()
```



Tata motors have the highest EV sales followed by Mahindra & Mahindra(exception: Q1fy24) while BYD India and Hyundai motors lag in this race.The only consistency by all the makers is in sales from Q1FY22 to Q2Fy22 where everyone's sale have increased.

In [516…
```
#5.How do the EV sales and penetration rates in Delhi compare to Karnataka for 2024
```

In [502…
```
df3['date'] = pd.to_datetime(df3['date'], format='%d-%b-%y')
```

In [503…
```
df3['date'] = pd.to_datetime(df3['date'], format='%Y-%m-%d')
```

In [504…
```
df3_2024=df3[(df3['date']>'2023-12-31') & (df3['date']<'2024-04-01')]
```

In [505…
```
df3_2024_filtered = df3_2024[df3_2024['state'].isin(['Karnataka', 'Delhi'])]
result = df3_2024_filtered.groupby('state')[['electric_vehicles_sold','total_vehicl
```

In [506…
```
result
```

Out[506…

state	electric_vehicles_sold	total_vehicles_sold
Delhi	13277	155839
Karnataka	51247	430905

In [507…
```
result['penetration_rate']=(result['electric_vehicles_sold']/result['total_vehicles
```

In [508…
```
result
```

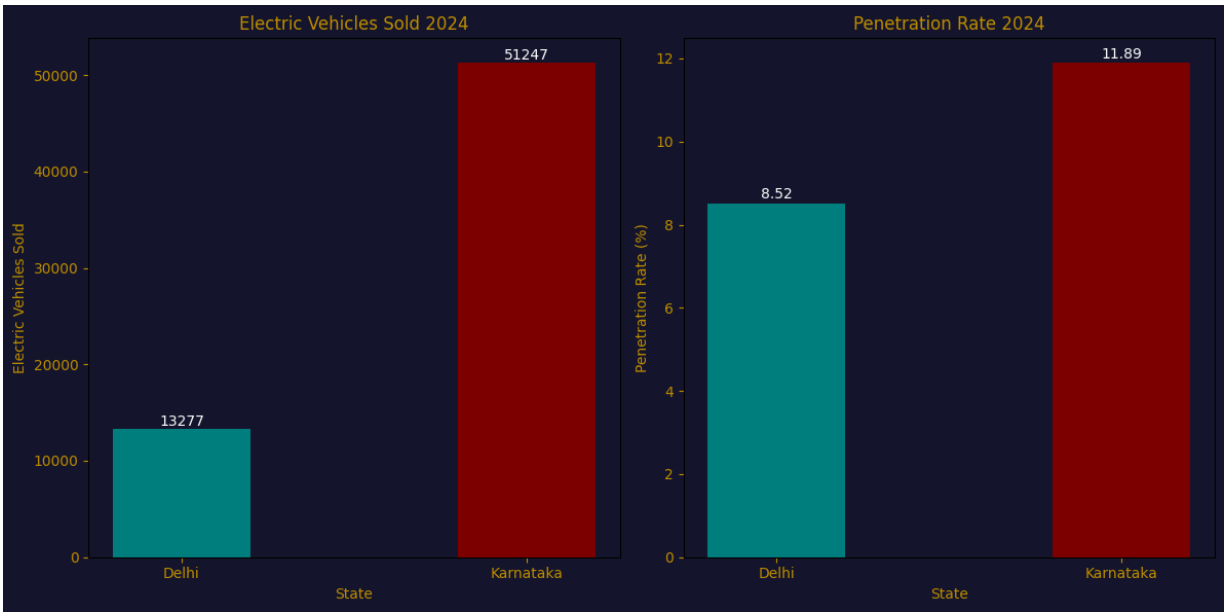
Out[508…

state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
Delhi	13277	155839	8.52
Karnataka	51247	430905	11.89

In [88]:
```
result.to_csv('output5.csv')
```

In [509…
```
data = {
    'state': ['Delhi', 'Karnataka'],
    'electric_vehicles_sold': [13277, 51247],
    'penetration': [8.52, 11.89]
}
df = pd.DataFrame(data)
```

In [581…
```
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
fig.patch.set_facecolor('#18182F')  # Set figure background color

# Plot for Electric Vehicles Sold
axes[0].bar(df['state'], df['electric_vehicles_sold'], color=['teal', 'maroon'], wi
axes[0].set_title('Electric Vehicles Sold 2024', color='#BF9000')  # Title color
axes[0].set_xlabel('State', color='#BF9000')  # X-axis label color
axes[0].set_ylabel('Electric Vehicles Sold', color='#BF9000')  # Y-axis label color
axes[0].set_facecolor('#18182F')  # Set subplot background color

# Add data labels to the first plot
for i, value in enumerate(df['electric_vehicles_sold']):
    axes[0].text(i, value + 1, str(value), ha='center', va='bottom', fontsize=10, c

# Plot for Penetration Rate
axes[1].bar(df['state'], df['penetration'], color=['teal', 'maroon'], width=0.4)
axes[1].set_title('Penetration Rate 2024', color='#BF9000')  # Title color
axes[1].set_xlabel('State',color='#BF9000')  # X-axis label color
axes[1].set_ylabel('Penetration Rate (%)', color='#BF9000')  # Y-axis label color
axes[1].set_facecolor('#18182F')  # Set subplot background color

# Add data labels to the second plot
for i, value in enumerate(df['penetration']):
    axes[1].text(i, value + 0.03, str(value), ha='center', va='bottom', fontsize=10

# Customize ticks color for better visibility
for ax in axes:
    ax.tick_params(axis='x', colors='#BF9000')  # X-axis ticks color
    ax.tick_params(axis='y', colors='#BF9000')  # Y-axis ticks color

# Show the plots
plt.tight_layout()  # Adjust layout for better spacing
plt.show()
```



In [576…
```
#6.List down the compounded annual growth rate (CAGR) in 4-wheeler units for the to
#CAGR = [(Ending Value / Beginning Value) ** 1/n] -1
```

In [92]:
```
IVD=df2[df2["vehicle_category"] == "4-Wheelers"]
```

In [95]:
```
IVD['date'] = pd.to_datetime(IVD['date'], format='%Y-%m-%d')
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\630512450.py:1: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  IVD['date'] = pd.to_datetime(IVD['date'], format='%Y-%m-%d')
```

In [96]:
```
IVD.loc[:, 'date'] = pd.to_datetime(IVD['date'], format='%d-%b-%y')
```

In [97]:
```
IVD.loc[:, 'date'] = pd.to_datetime(IVD['date'], format='%Y-%m-%d')
```

In [98]:
```
IVD
```

Out[98]:

	date	vehicle_category	maker	electric_vehicles_sold
9	2021-04-01	4-Wheelers	BYD India	0
10	2021-04-01	4-Wheelers	PCA Automobiles	0
11	2021-04-01	4-Wheelers	BMW India	0
12	2021-04-01	4-Wheelers	Volvo Auto India	0
13	2021-04-01	4-Wheelers	KIA Motors	0
...	...	...	...	...
364	2024-03-01	4-Wheelers	PCA Automobiles	130
365	2024-03-01	4-Wheelers	BMW India	55
366	2024-03-01	4-Wheelers	Mercedes -Benz AG	31
367	2024-03-01	4-Wheelers	Volvo Auto India	39
368	2024-03-01	4-Wheelers	KIA Motors	26

360 rows × 4 columns

In [99]:
```
IVD_fy22=IVD[(IVD['date']>'2021-03-31') & (IVD['date']<'2022-04-01')]
IVD_fy23=IVD[(IVD['date']>'2022-03-31') & (IVD['date']<'2023-04-01')]
IVD_fy24=IVD[(IVD['date']>'2023-03-31') & (IVD['date']<'2024-04-01')]
```

In [100…
```
IVD_fy22
```

Out[100…

	date	vehicle_category	maker	electric_vehicles_sold
9	2021-04-01	4-Wheelers	BYD India	0
10	2021-04-01	4-Wheelers	PCA Automobiles	0
11	2021-04-01	4-Wheelers	BMW India	0
12	2021-04-01	4-Wheelers	Volvo Auto India	0
13	2021-04-01	4-Wheelers	KIA Motors	0
...	...	...	...	...
344	2022-03-01	4-Wheelers	BYD India	14
345	2022-03-01	4-Wheelers	Hyundai Motor	14
346	2022-03-01	4-Wheelers	BMW India	7
347	2022-03-01	4-Wheelers	Mercedes -Benz AG	5
348	2022-03-01	4-Wheelers	Volvo Auto India	4

120 rows × 4 columns

In [101…
```
IVD_fy22_24=IVD[(IVD['date']>'2021-03-31') & (IVD['date']<'2024-04-01')]
```

In [102…
```
Top5_maker =IVD_fy22_24.groupby("maker")["electric_vehicles_sold"].sum().sort_value
```

In [103…
```
Top5_maker
```

Out[103…
```
maker
Tata Motors             88935
Mahindra & Mahindra     41193
MG Motor                13753
BYD India                2419
Hyundai Motor            2076
Name: electric_vehicles_sold, dtype: int64
```

In [104…
```
top5_22=IVD_fy22[IVD_fy22['maker'].isin(Top5_maker.index)].groupby("maker")["electr
top5_23=IVD_fy23[IVD_fy23['maker'].isin(Top5_maker.index)].groupby("maker")["electr
top5_24=IVD_fy24[IVD_fy24['maker'].isin(Top5_maker.index)].groupby("maker")["electr
```

In [105…
```
top5_22
```

Out[105…
```
maker
BYD India                  33
Hyundai Motor             110
MG Motor                 1647
Mahindra & Mahindra      4042
Tata Motors             12708
Name: electric_vehicles_sold, dtype: int64
```

In [106…
```
# change column name in pandas series
top5_22.name = "EV_sold_fy22"
top5_23.name=  "EV_sold_fy23"
top5_24.name = "EV_sold_fy24"
```

In [107…
```
x=pd.concat([top5_22,top5_23,top5_24],axis=1,join="inner")
```

In [108…
```
x
```

Out[108…

maker	EV_sold_fy22	EV_sold_fy23	EV_sold_fy24
BYD India	33	920	1466
Hyundai Motor	110	576	1390
MG Motor	1647	3277	8829
Mahindra & Mahindra	4042	13805	23346
Tata Motors	12708	28046	48181

In [109…
```
x['CAGR_22_24']=x['EV_sold_fy24']/x['EV_sold_fy22']**(1/2)-1
```

In [110…
```
x
```

Out[110…

maker	EV_sold_fy22	EV_sold_fy23	EV_sold_fy24	CAGR_22_24
BYD India	33	920	1466	254.20
Hyundai Motor	110	576	1390	131.53
MG Motor	1647	3277	8829	216.55
Mahindra & Mahindra	4042	13805	23346	366.21
Tata Motors	12708	28046	48181	426.40

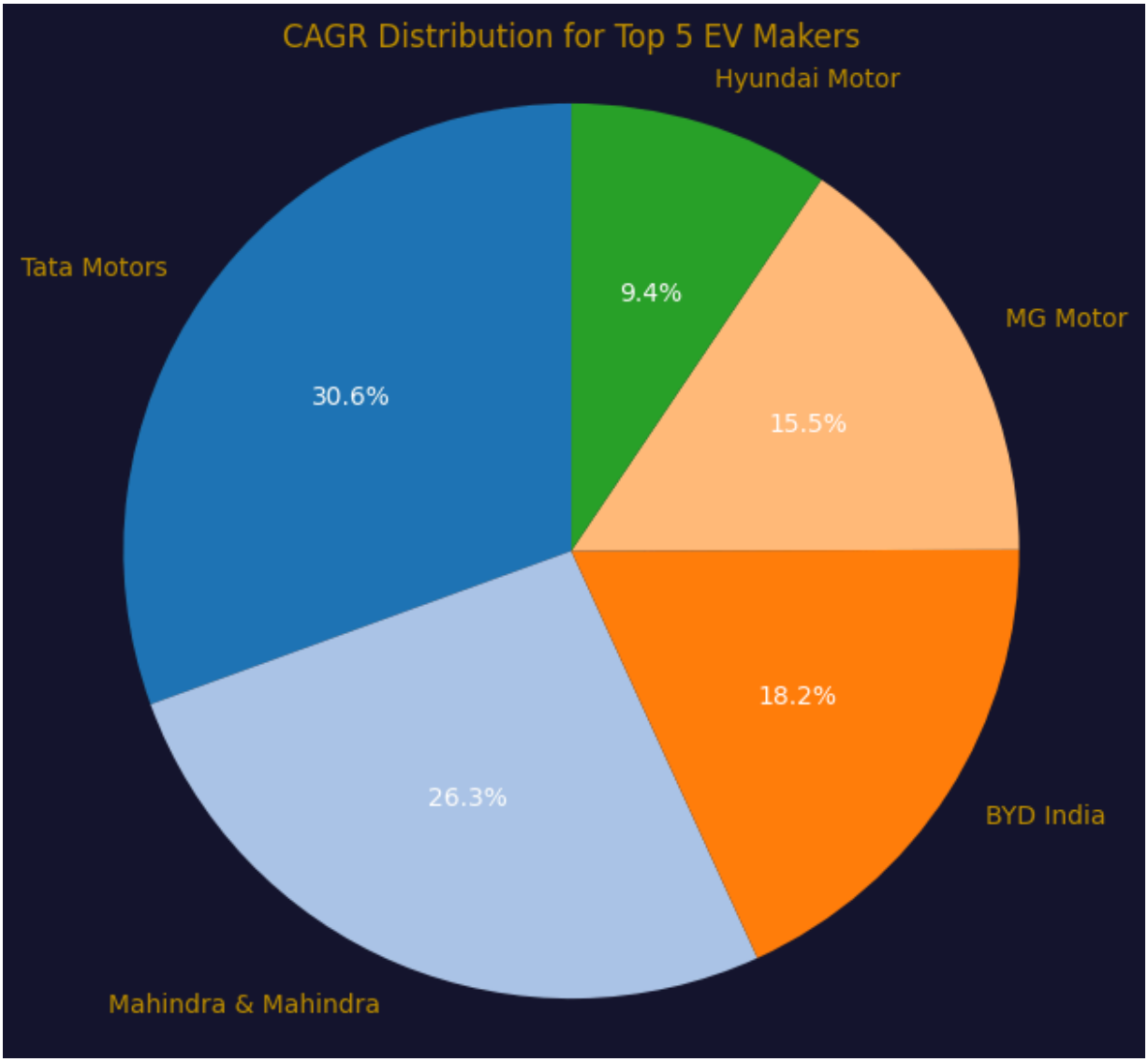
In [111…
```
y=x['CAGR_22_24'].sort_values(ascending=False)
```

In [112…
```
y
```

Out[112…
```
maker
Tata Motors           426.40
Mahindra & Mahindra   366.21
BYD India             254.20
MG Motor              216.55
Hyundai Motor         131.53
Name: CAGR_22_24, dtype: float64
```

In [113…
```
y.to_csv('output6.csv')
```

In [521…
```
labels = y.index  # Your labels
sizes = y.values  # Your sizes
colors1 = ['maroon', 'teal', 'yellow', 'violet', 'green']

# Create a pie chart with a specified figure size and background color
plt.figure(figsize=(7, 7), facecolor='#18182F')  # Set figure background color
wedges, texts, autotexts = plt.pie(sizes, labels=labels, colors=colors1, autopct='%
# Equal aspect ratio ensures the pie chart is drawn as a circle.
plt.axis('equal')

# Set title and adjust color
plt.title('CAGR Distribution for Top 5 EV Makers', color='#BF9000')  # Title color

# Set the percentage text color to white
for autotext in autotexts:
    autotext.set_color('white')  # Set percentage labels to white

# Set slice labels (names of pie sections) to #BF9000
for text in texts:
    text.set_color('#BF9000')  # Set pie slice labels to golden color

# Show the plot
plt.show()
```



In [743…
```
#7.List down the top 10 states that had the highest compounded annual growth rate (
```

In [129…
```
df3['date']= pd.to_datetime(df3['date'], format='%Y-%m-%d')
```

In [130…
```
fy22=df3[(df3['date']>'2021-03-31') & (df3['date']<'2022-04-01')]
fy24=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [131…
```
fy22
```

Out[131…

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	2021-04-01	Sikkim	2-Wheelers	0	398
1	2021-04-01	Sikkim	4-Wheelers	0	361
2	2021-05-01	Sikkim	2-Wheelers	0	113
3	2021-05-01	Sikkim	4-Wheelers	0	98
4	2021-06-01	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
1745	2022-03-01	Tamil Nadu	2-Wheelers	7708	124272
1746	2022-03-01	Tripura	2-Wheelers	18	3504
1747	2022-03-01	Uttar Pradesh	2-Wheelers	1986	180927
1748	2022-03-01	Uttarakhand	2-Wheelers	435	11692
1749	2022-03-01	West Bengal	2-Wheelers	626	73783

815 rows × 5 columns

In [132…
```
fy22=fy22.groupby('state')['total_vehicles_sold'].sum()
fy24=fy24.groupby('state')['total_vehicles_sold'].sum()
```

In [216…
```
type(fy22)
```

Out[216… pandas.core.series.Series

In [133…
```
fy22.name="total_vehicles_sold_22"
fy24.name="total_vehicles_sold_24"
```

In [134…
```
x=pd.concat([fy22,fy24],axis=1,join='inner')
```

In [135…
```
x['CAGR']=(x['total_vehicles_sold_24']/x['total_vehicles_sold_22'])**(1/2)-1
```

In [136…
```
res=x['CAGR'].sort_values(ascending=False).head(10)
```

In [137…
```
res
```

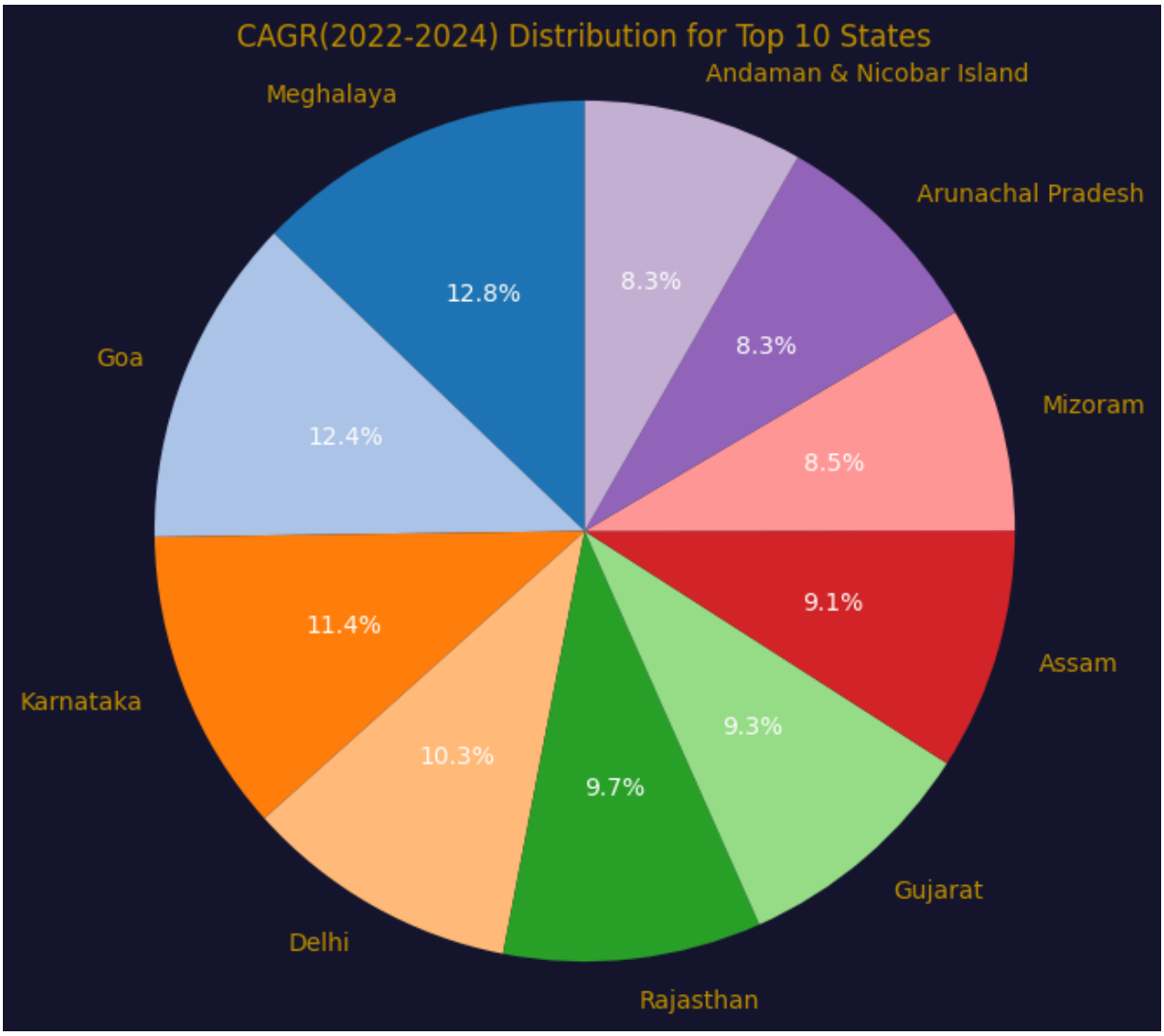
Out[137…
```
state
Meghalaya                  0.28
Goa                        0.27
Karnataka                  0.25
Delhi                      0.23
Rajasthan                  0.21
Gujarat                    0.21
Assam                      0.20
Mizoram                    0.19
Arunachal Pradesh          0.18
Andaman & Nicobar Island   0.18
Name: CAGR, dtype: float64
```

In [222…
```
type(res)
```

Out[222…
```
pandas.core.series.Series
```

In [138…
```
res.to_csv('output7.csv')
```

In [519…
```
labels = res.index  # Your labels
sizes = res.values  # Your sizes

# Use a colormap to generate colors automatically
colors1=['#FFA07A','teal','maroon','#E4C200','blue','red','green','slateblue','oran

# Create a pie chart with a specified figure size and background color
plt.figure(figsize=(7, 7), facecolor='#18182F')  # Set figure background color
wedges, texts, autotexts = plt.pie(sizes, labels=labels, colors=colors2, autopct='%

# Equal aspect ratio ensures the pie chart is drawn as a circle.
plt.axis('equal')
plt.title('CAGR(2022-2024) Distribution for Top 10 States', color='#BF9000')

# Set the percentage text color to white
for autotext in autotexts:
    autotext.set_color('white')  # Set percentage labels to white

# Set slice labels (names of pie sections) to #BF9000
for text in texts:
    text.set_color('#BF9000')  # Set pie slice labels to golden color

plt.show()
```



In [787…
```
#8.What are the peak and low season months for EV sales based on the data from 2022
```

In [171…
```
df2
```

Out[171…

	date	vehicle_category	maker	electric_vehicles_sold
0	2021-04-01	2-Wheelers	OLA ELECTRIC	0
1	2022-04-01	2-Wheelers	OKAYA EV	0
2	2021-05-01	2-Wheelers	OLA ELECTRIC	0
3	2021-06-01	2-Wheelers	OLA ELECTRIC	0
4	2021-07-01	2-Wheelers	OLA ELECTRIC	0
...	...	...	...	...
811	2024-03-01	2-Wheelers	BGAUSS	3070
812	2024-03-01	2-Wheelers	BATTRE ELECTRIC	625
813	2024-03-01	2-Wheelers	KINETIC GREEN	3915
814	2024-03-01	2-Wheelers	REVOLT	585
815	2024-03-01	2-Wheelers	OTHERS	10579

816 rows × 4 columns

In [172…
```
df2['date'] = pd.to_datetime(df2['date'], format='%Y-%m-%d')
```

In [173…
```
# empty dictionary creation to store month and total_ev_sold as key:value pair
month_tot_22 = {}
month_tot_23 = {}
month_tot_24 = {}
```

In [174…
```
# totals for each month of 2022
month_tot_22['Apr']=df2[(df2['date']>'2021-03-31') & (df2['date']<'2021-05-01')]['e
month_tot_22['May']=df2[(df2['date']>'2021-04-30') & (df2['date']<'2021-06-01')]['e
month_tot_22['Jun']=df2[(df2['date']>'2021-05-31') & (df2['date']<'2021-07-01')]['e
month_tot_22['Jul']=df2[(df2['date']>'2021-06-30') & (df2['date']<'2021-08-01')]['e
month_tot_22['Aug']=df2[(df2['date']>'2021-07-31') & (df2['date']<'2021-09-01')]['e
month_tot_22['Sep']=df2[(df2['date']>'2021-08-31') & (df2['date']<'2021-10-01')]['e
month_tot_22['Oct']=df2[(df2['date']>'2021-09-30') & (df2['date']<'2021-11-01')]['e
month_tot_22['Nov']=df2[(df2['date']>'2021-10-31') & (df2['date']<'2021-12-01')]['e
month_tot_22['Dec']=df2[(df2['date']>'2021-11-30') & (df2['date']<'2022-01-01')]['e
month_tot_22['Jan']=df2[(df2['date']>'2021-12-31') & (df2['date']<'2022-02-01')]['e
month_tot_22['Feb']=df2[(df2['date']>'2022-01-31') & (df2['date']<'2022-03-01')]['e
month_tot_22['Mar']=df2[(df2['date']>'2022-02-28') & (df2['date']<'2022-04-01')]['e
```

In [175…
```
# totals for each month of 2023
month_tot_23['Apr']=df2[(df2['date']>'2022-03-31') & (df2['date']<'2022-05-01')]['e
month_tot_23['May']=df2[(df2['date']>'2022-04-30') & (df2['date']<'2022-06-01')]['e
month_tot_23['Jun']=df2[(df2['date']>'2022-05-31') & (df2['date']<'2022-07-01')]['e
month_tot_23['Jul']=df2[(df2['date']>'2022-06-30') & (df2['date']<'2022-08-01')]['e
month_tot_23['Aug']=df2[(df2['date']>'2022-07-31') & (df2['date']<'2022-09-01')]['e
month_tot_23['Sep']=df2[(df2['date']>'2022-08-31') & (df2['date']<'2022-10-01')]['e
month_tot_23['Oct']=df2[(df2['date']>'2022-09-30') & (df2['date']<'2022-11-01')]['e
month_tot_23['Nov']=df2[(df2['date']>'2022-10-31') & (df2['date']<'2022-12-01')]['e
```

```
month_tot_23['Dec']=df2[(df2['date']>'2022-11-30') & (df2['date']<'2023-01-01')]['e
month_tot_23['Jan']=df2[(df2['date']>'2022-12-31') & (df2['date']<'2023-02-01')]['e
month_tot_23['Feb']=df2[(df2['date']>'2023-01-31') & (df2['date']<'2023-03-01')]['e
month_tot_23['Mar']=df2[(df2['date']>'2023-02-28') & (df2['date']<'2023-04-01')]['e
```

In [176…
```
# totals for each month of 2024
month_tot_24['Apr']=df2[(df2['date']>'2023-03-31') & (df2['date']<'2023-05-01')]['e
month_tot_24['May']=df2[(df2['date']>'2023-04-30') & (df2['date']<'2023-06-01')]['e
month_tot_24['Jun']=df2[(df2['date']>'2023-05-31') & (df2['date']<'2023-07-01')]['e
month_tot_24['Jul']=df2[(df2['date']>'2023-06-30') & (df2['date']<'2023-08-01')]['e
month_tot_24['Aug']=df2[(df2['date']>'2023-07-31') & (df2['date']<'2023-09-01')]['e
month_tot_24['Sep']=df2[(df2['date']>'2023-08-31') & (df2['date']<'2023-10-01')]['e
month_tot_24['Oct']=df2[(df2['date']>'2023-09-30') & (df2['date']<'2023-11-01')]['e
month_tot_24['Nov']=df2[(df2['date']>'2023-10-31') & (df2['date']<'2023-12-01')]['e
month_tot_24['Dec']=df2[(df2['date']>'2023-11-30') & (df2['date']<'2024-01-01')]['e
month_tot_24['Jan']=df2[(df2['date']>'2023-12-31') & (df2['date']<'2024-02-01')]['e
month_tot_24['Feb']=df2[(df2['date']>'2024-01-31') & (df2['date']<'2024-03-01')]['e
month_tot_24['Mar']=df2[(df2['date']>'2024-02-29') & (df2['date']<'2024-04-01')]['e
```

In [177…
```
month_tot_22
```

Out[177…
```
{'Apr': 6315,
 'May': 1499,
 'Jun': 5487,
 'Jul': 15794,
 'Aug': 17153,
 'Sep': 19351,
 'Oct': 22190,
 'Nov': 26159,
 'Dec': 29241,
 'Jan': 31672,
 'Feb': 38171,
 'Mar': 58118}
```

In [178…
```
month_tot_23
```

Out[178…
```
{'Apr': 55524,
 'May': 45373,
 'Jun': 47591,
 'Jul': 50010,
 'Aug': 55584,
 'Sep': 56828,
 'Oct': 81162,
 'Nov': 80752,
 'Dec': 68475,
 'Jan': 68116,
 'Feb': 70827,
 'Mar': 95126}
```

In [179…
```
month_tot_24
```

Out[179…
```
{'Apr': 72818,
 'May': 112997,
 'Jun': 53631,
 'Jul': 61622,
 'Aug': 69224,
 'Sep': 69793,
 'Oct': 81833,
 'Nov': 98285,
 'Dec': 82685,
 'Jan': 89311,
 'Feb': 89051,
 'Mar': 138343}
```

In [180…
```
# inserting values to the empty dictionary
month_tot_22={'Apr': 6315,
 'May': 1499,
 'Jun': 5487,
 'Jul': 15794,
 'Aug': 17153,
 'Sep': 19351,
 'Oct': 22190,
 'Nov': 26159,
 'Dec': 29241,
 'Jan': 31672,
 'Feb': 38171,
 'Mar': 58118}
```

In [181…
```
month_tot_23={'Apr': 55524,
 'May': 45373,
 'Jun': 47591,
 'Jul': 50010,
 'Aug': 55584,
 'Sep': 56828,
 'Oct': 81162,
 'Nov': 80752,
 'Dec': 68475,
 'Jan': 68116,
 'Feb': 70827,
 'Mar': 95126}
```

In [182…
```
month_tot_24={'Apr': 72818,
 'May': 112997,
 'Jun': 53631,
 'Jul': 61622,
 'Aug': 69224,
 'Sep': 69793,
 'Oct': 81833,
 'Nov': 98285,
 'Dec': 82685,
 'Jan': 89311,
 'Feb': 89051,
 'Mar': 138343}
```

In [183…
```
#creating dataframes from dictionary for easy data anlysis
df_2022=pd.DataFrame(list(month_tot_22.items()), columns=['Month', 'Tot_EV_sold'])
```

```
df_2023=pd.DataFrame(list(month_tot_23.items()), columns=['Month', 'Tot_EV_sold'])
df_2024=pd.DataFrame(list(month_tot_24.items()), columns=['Month', 'Tot_EV_sold'])
```

In [184…
```
# getting all information in one place
merged_df_2022_2023 = pd.merge(df_2022, df_2023, on='Month', how='outer', suffixes=
```

In [185…
```
merged_df_2022_2023
```

Out[185…

	Month	Tot_EV_sold_2022	Tot_EV_sold_2023
0	Apr	6315	55524
1	May	1499	45373
2	Jun	5487	47591
3	Jul	15794	50010
4	Aug	17153	55584
5	Sep	19351	56828
6	Oct	22190	81162
7	Nov	26159	80752
8	Dec	29241	68475
9	Jan	31672	68116
10	Feb	38171	70827
11	Mar	58118	95126

In [186…
```
final_merged_df = pd.merge(merged_df_2022_2023, df_2024, on='Month', how='outer')
```

In [187…
```
final_merged_df
```

Out[187…

	Month	Tot_EV_sold_2022	Tot_EV_sold_2023	Tot_EV_sold
0	Apr	6315	55524	72818
1	May	1499	45373	112997
2	Jun	5487	47591	53631
3	Jul	15794	50010	61622
4	Aug	17153	55584	69224
5	Sep	19351	56828	69793
6	Oct	22190	81162	81833
7	Nov	26159	80752	98285
8	Dec	29241	68475	82685
9	Jan	31672	68116	89311
10	Feb	38171	70827	89051
11	Mar	58118	95126	138343

In [203…

```
final_merged_df.rename(columns={'Tot_EV_sold': 'Tot_EV_sold_2024'}, inplace=True)
```

In [204…

```
final_merged_df
```

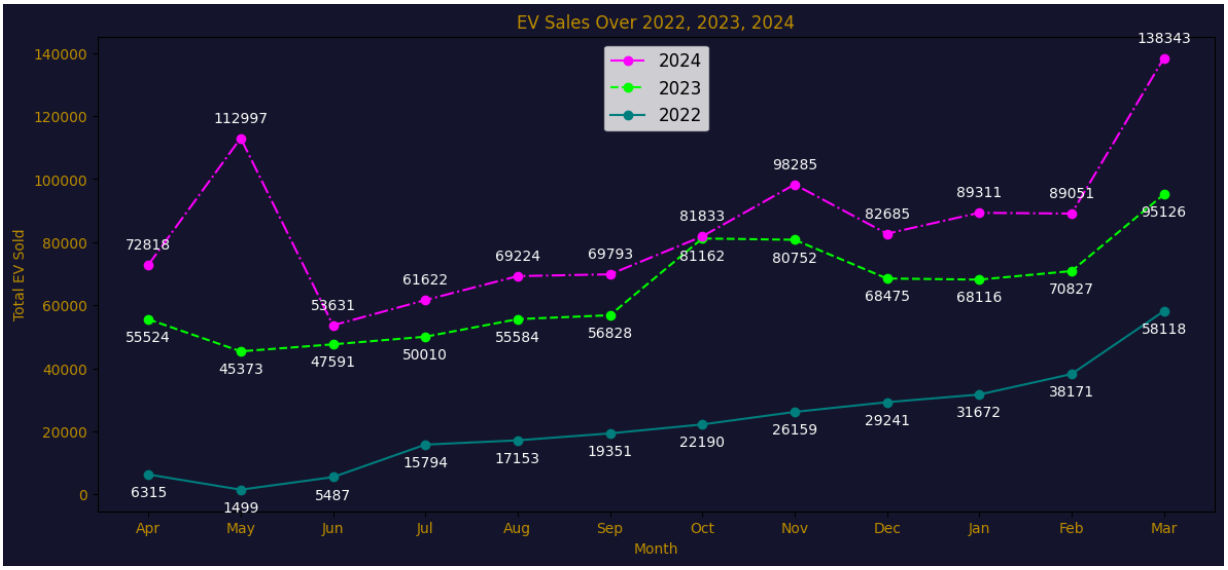
Out[204…

	Month	Tot_EV_sold_2022	Tot_EV_sold_2023	Tot_EV_sold_2024
0	Apr	6315	55524	72818
1	May	1499	45373	112997
2	Jun	5487	47591	53631
3	Jul	15794	50010	61622
4	Aug	17153	55584	69224
5	Sep	19351	56828	69793
6	Oct	22190	81162	81833
7	Nov	26159	80752	98285
8	Dec	29241	68475	82685
9	Jan	31672	68116	89311
10	Feb	38171	70827	89051
11	Mar	58118	95126	138343

In [527…

```
plt.figure(figsize=(14, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')
```

```
# Plot for 2022
plt.plot(df_2022['Month'], df_2022['Tot_EV_sold'], label='2022', marker='o', color=
for i, txt in enumerate(df_2022['Tot_EV_sold']):
    plt.text(df_2022['Month'].iloc[i], df_2022['Tot_EV_sold'].iloc[i]-7000 ,  # Adj
             f'{txt}', color='white', ha='center')

# Plot for 2023
plt.plot(df_2023['Month'], df_2023['Tot_EV_sold'], label='2023', marker='o', color=
for i, txt in enumerate(df_2023['Tot_EV_sold']):
    plt.text(df_2023['Month'].iloc[i], df_2023['Tot_EV_sold'].iloc[i]-7000,  # Adju
             f'{txt}', color='white', ha='center')

# Plot for 2024
plt.plot(df_2024['Month'], df_2024['Tot_EV_sold'], label='2024', marker='o', color=
for i, txt in enumerate(df_2024['Tot_EV_sold']):
    plt.text(df_2024['Month'].iloc[i], df_2024['Tot_EV_sold'].iloc[i] + 5000,  # Ad
             f'{txt}', color='white', ha='center')
# Customize labels, title, and ticks
plt.xlabel('Month', color='#BF9000')
plt.ylabel('Total EV Sold', color='#BF9000')
plt.title('EV Sales Over 2022, 2023, 2024', color='#BF9000')
# Customize the ticks and text color
plt.xticks(rotation=0, color='#BF9000')
plt.yticks(color='#BF9000')
#legend order handling
handles, labels = plt.gca().get_legend_handles_labels()
order = [2, 1, 0]  # List of indices to reorder
plt.legend([handles[i] for i in order], [labels[i] for i in order],facecolor='WHITE
plt.show()
```



Ev sales gradually increasing over years.Only Oct2022 and Oct2023 have nearly same EV sales. {Highest EV sales:Mar2024(138343),lowest EV sales:May2022(1499)}

9. What is the projected number of EV sales (including 2-wheelers and 4-wheelers) for

the top 10 states by penetration rate in 2030,

based on the compounded annual growth rate (CAGR) from previous years?

Answer approach: first top 10state by penetration rate is determined from df3. Then for 2022 ev_sales for those states are calculated , again same thing done for 2024 Then CAGR_22_24(CAGR for 3 years is calculated using formula, CAGR= (Ending_value/ Beginning_value)**(1/n)-1 where n=no.of years here n=2024-2022=2) Lastly,projected EV sales is calculated using formula, Projected Sales=Ending_value×(1+CAGR)**n where n=2030-2024=6) here, Ending_value=EV_sales_2024 Beginning_value=EV_sales_2022 CAGR=CAGR_22_24 projected EV sales=EV_sales_2030 for top 10 states by penetration

In [306…

```
df3
```

Out[306…

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	1-Apr-21	Sikkim	2-Wheelers	0	398
1	1-Apr-21	Sikkim	4-Wheelers	0	361
2	1-May-21	Sikkim	2-Wheelers	0	113
3	1-May-21	Sikkim	4-Wheelers	0	98
4	1-Jun-21	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
2440	1-Mar-24	Mizoram	2-Wheelers	58	1932
2441	1-Mar-24	DNH and DD	2-Wheelers	25	780
2442	1-Mar-24	Manipur	2-Wheelers	13	1394
2443	1-Mar-24	Andaman & Nicobar Island	2-Wheelers	2	447
2444	1-Mar-24	Nagaland	2-Wheelers	2	1180

2445 rows × 5 columns

In [309…
```
df3['date']=pd.to_datetime(df3['date'])
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_34120\3029150507.py:1: UserWarning: Coul
d not infer format, so each element will be parsed individually, falling back to `da
teutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df3['date']=pd.to_datetime(df3['date'])
```

In [312…
```
df3['date']=pd.to_datetime(df3['date'],format='%d-%m-%Y')
```

In [313…
```
df3['date']=pd.to_datetime(df3['date'],format='%Y-%m-%d')
```

In [315…
```
k=df3.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].sum()
```

In [316…
```
k['penetration_rate']=100*(k['electric_vehicles_sold']/k['total_vehicles_sold'])
```

In [317…
```
top10_state=k.groupby("state")["penetration_rate"].sum().sort_values(ascending=Fals
```

In [318…
```
top10_state=k.sort_values(by='penetration_rate',ascending=False).reset_index().head
```

In [319…
```
top10_state
```

Out[319…

	state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
0	Goa	19684	199970	9.84
1	Karnataka	312995	3994329	7.84
2	Delhi	107312	1588436	6.76
3	Kerala	137060	2064677	6.64
4	Maharashtra	396045	6101429	6.49
5	Odisha	78267	1688794	4.63
6	Rajasthan	150366	3307591	4.55
7	Gujarat	181389	4125551	4.40
8	Tamil Nadu	200062	4652363	4.30
9	Chandigarh	5279	130628	4.04

In [320…
```
df3_2022=df3[(df3['date']>= '2021-04-01') &  (df3['date']<'2022-04-01')]
df3_2024=df3[(df3['date']>= '2023-04-01') &  (df3['date']<'2024-04-01')]
```

In [321…
```
merge_2022= df3_2022.merge(top10_state[['state']], on='state',how='inner')
merge_2024=df3_2024.merge(top10_state[['state']], on='state',how='inner')
```

In [322…
```
ev_sales_2022 = merge_2022.groupby('state')['electric_vehicles_sold'].sum()
ev_sales_2024 = merge_2024.groupby('state')['electric_vehicles_sold'].sum()
```

In [323…
```
ev_sales_2022=ev_sales_2022.rename("ev_sales_2022")
ev_sales_2024=ev_sales_2024.rename("ev_sales_2024")
```

In [324…
```
ev_sales=pd.concat([ev_sales_2022,ev_sales_2024],axis=1)
```

In [325…
```
ev_sales
```

Out[325…

state	ev_sales_2022	ev_sales_2024
Chandigarh	411	2877
Delhi	16535	46724
Goa	1778	10799
Gujarat	18026	84359
Karnataka	43111	160989
Kerala	13639	73938
Maharashtra	48374	197169
Odisha	9498	39118
Rajasthan	20087	66444
Tamil Nadu	36863	94314

In [326…
```
ev_sales['CAGR_22_24']=(ev_sales['ev_sales_2024']/ev_sales['ev_sales_2022'])**(1/2)
```

In [327…
```
pd.options.display.float_format = '{:.2f}'.format
ev_sales
```

Out[327...

state	ev_sales_2022	ev_sales_2024	CAGR_22_24
Chandigarh	411	2877	1.65
Delhi	16535	46724	0.68
Goa	1778	10799	1.46
Gujarat	18026	84359	1.16
Karnataka	43111	160989	0.93
Kerala	13639	73938	1.33
Maharashtra	48374	197169	1.02
Odisha	9498	39118	1.03
Rajasthan	20087	66444	0.82
Tamil Nadu	36863	94314	0.60

Projected Sales=EV_Sales_2024×(1+CAGR)**n n=2030-2024=6

In [336...
```
ev_sales['ev_sales_2030']=ev_sales['ev_sales_2024']*((1+ev_sales['CAGR_22_24'])**6)
```

In [337...
```
ev_sales.sort_values(by="ev_sales_2030",ascending=False,inplace= True)
```

In [338...
```
ev_sales
```

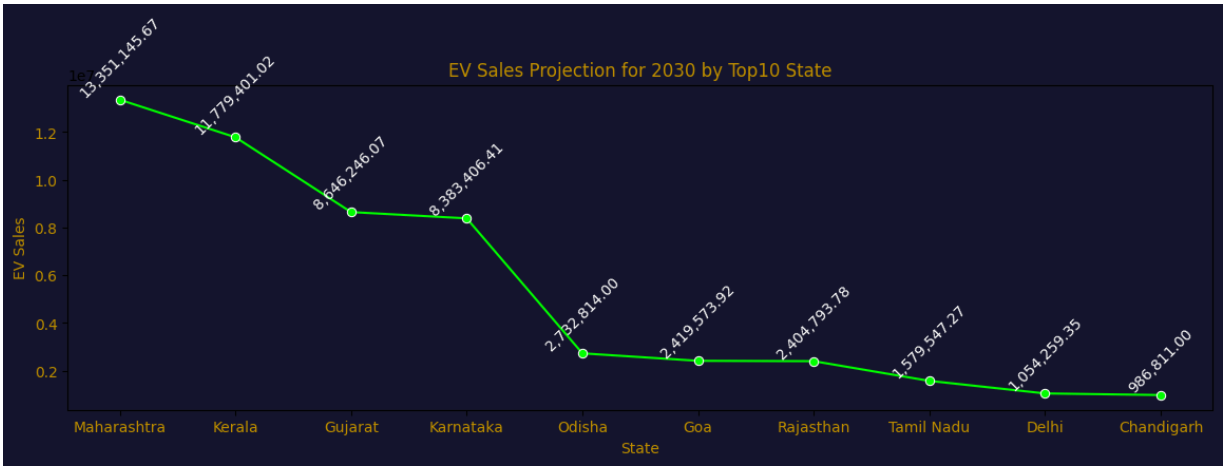
Out[338...

state	ev_sales_2022	ev_sales_2024	CAGR_22_24	ev_sales_2030
Maharashtra	48374	197169	1.02	13351145.67
Kerala	13639	73938	1.33	11779401.02
Gujarat	18026	84359	1.16	8646246.07
Karnataka	43111	160989	0.93	8383406.41
Odisha	9498	39118	1.03	2732814.00
Goa	1778	10799	1.46	2419573.92
Rajasthan	20087	66444	0.82	2404793.78
Tamil Nadu	36863	94314	0.60	1579547.27
Delhi	16535	46724	0.68	1054259.35
Chandigarh	411	2877	1.65	986811.00

In [298...
```
ev_sales.to_csv('output9.csv')
```

In [530…

```
plt.figure(figsize=(14, 4), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')

# Create the line plot using the index for the x-axis
sns.lineplot(ev_sales, x=ev_sales.index, y='ev_sales_2030', color='lime', marker='o

# labels above the points with two decimal points and rotate by 45 degrees
for i in range(len(ev_sales)):
    plt.text(ev_sales.index[i], ev_sales['ev_sales_2030'].iloc[i],
             f'{ev_sales["ev_sales_2030"].iloc[i]:,.2f}',
             color='white', ha='center', va='bottom', rotation=45)

# Set title and customize text color
plt.title('EV Sales Projection for 2030 by Top10 State', color='#BF9000')
plt.xlabel('State', color='#BF9000')
plt.ylabel('EV Sales', color='#BF9000')

# Customize tick labels
plt.xticks( color='#BF9000')
plt.yticks(color='#BF9000')
# Show the plot
plt.show()
```



Highest projected sales is for Maharastra and lowest is for Chandigarh.

In [107…

```
#10.Estimate the revenue growth rate of 4-wheeler and 2-wheelers EVs in India for 2
#Avg unit price for 2_wheelers=85000, 4_wheelers=1500000
```

In [340…

```
df3
```

Out[340...

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	2021-04-01	Sikkim	2-Wheelers	0	398
1	2021-04-01	Sikkim	4-Wheelers	0	361
2	2021-05-01	Sikkim	2-Wheelers	0	113
3	2021-05-01	Sikkim	4-Wheelers	0	98
4	2021-06-01	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
2440	2024-03-01	Mizoram	2-Wheelers	58	1932
2441	2024-03-01	DNH and DD	2-Wheelers	25	780
2442	2024-03-01	Manipur	2-Wheelers	13	1394
2443	2024-03-01	Andaman & Nicobar Island	2-Wheelers	2	447
2444	2024-03-01	Nagaland	2-Wheelers	2	1180

2445 rows × 5 columns

In [341...
```
df3['date'] = pd.to_datetime(df3['date'], format='%Y%m-%d')
```

In [342...
```
df3_2022= df3[(df3['date']>'2021-03-31') & (df3['date']<'2022-04-01')]
df3_2023=df3[(df3['date']>'2022-03-31') & (df3['date']<'2023-04-01')]
df3_2024=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [343...
```
Rev2w_2022 = df3_2022[df3_2022['vehicle_category'] =='2-Wheelers'].groupby('vehicle
Rev2w_2023 = df3_2023[df3_2023['vehicle_category'] == '2-Wheelers'].groupby('vehicl
Rev2w_2024 = df3_2024[df3_2024['vehicle_category'] == '2-Wheelers'].groupby('vehicl
```

In [344...
```
Rev4w_2022 = df3_2022[df3_2022['vehicle_category'] == '4-Wheelers'].groupby('vehicl
Rev4w_2023 = df3_2023[df3_2023['vehicle_category'] == '4-Wheelers'].groupby('vehicl
Rev4w_2024 = df3_2024[df3_2024['vehicle_category'] == '4-Wheelers'].groupby('vehicl
```

In [345...
```
Revenue2w={'Rev_2022':Rev2w_2022,'Rev_2023':Rev2w_2023,'Rev_2024':Rev2w_2024}
```

In [346...
```
two_wh=pd.DataFrame(Revenue2w)
```

In [347…
```
Revenue4w={'Rev_2022':Rev4w_2022,'Rev_2023':Rev4w_2023,'Rev_2024':Rev4w_2024}
```

In [348…
```
four_wh=pd.DataFrame(Revenue4w)
```

In [349…
```
x=pd.concat([two_wh,four_wh],join='inner')
```

In [350…
```
x
```

Out[350…

vehicle_category	Rev_2022	Rev_2023	Rev_2024
2-Wheelers	21468705000	61871755000	79278820000
4-Wheelers	27865500000	71197500000	130351500000

In [351…
```
x['growth_22_24']=100*(x['Rev_2024']-x['Rev_2022'])/x['Rev_2022']
```

In [352…
```
x
```

Out[352…

vehicle_category	Rev_2022	Rev_2023	Rev_2024	growth_22_24
2-Wheelers	21468705000	61871755000	79278820000	269.28
4-Wheelers	27865500000	71197500000	130351500000	367.79

In [353…
```
x['growth_23_24']=100*(x['Rev_2024']-x['Rev_2023'])/x['Rev_2023']
```

In [354…
```
x
```

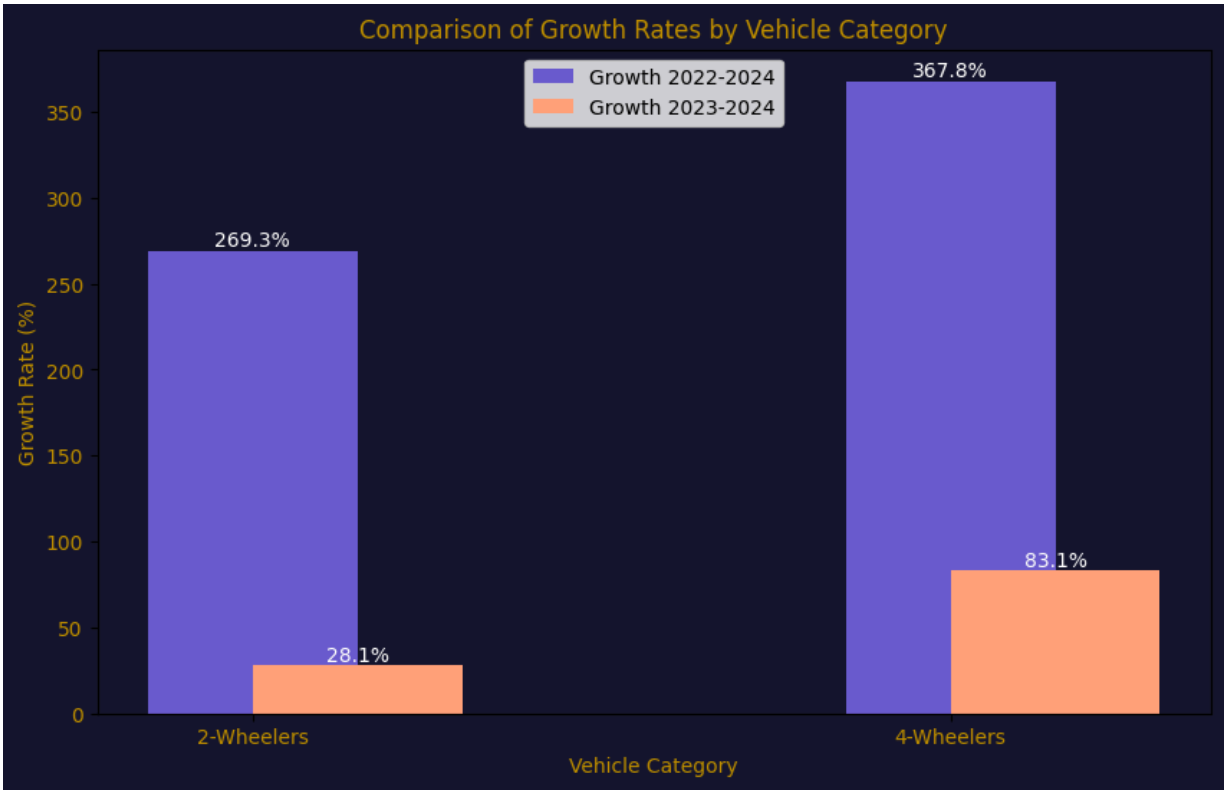
Out[354…

vehicle_category	Rev_2022	Rev_2023	Rev_2024	growth_22_24	growth_23_24
2-Wheelers	21468705000	61871755000	79278820000	269.28	28.13
4-Wheelers	27865500000	71197500000	130351500000	367.79	83.08

In [355…
```
x.to_csv('output10.csv')
```

In [532…
```
vehicle_category = ['2-Wheelers', '4-Wheelers']
growth_22_24 = [269.28, 367.79]
growth_23_24 = [28.13, 83.08]

fig, ax = plt.subplots(figsize=(10, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')

chat1=ax.bar(vehicle_category, growth_22_24, width=0.3, label='Growth 2022-2024', a
```

```
chat2=ax.bar(vehicle_category, growth_23_24, width=0.3, label='Growth 2023-2024', a

for bar in chat1:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.1f}%', ha='center

for bar in chat2:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.1f}%', ha='center

ax.set_xlabel('Vehicle Category',color='#BF9000')
ax.set_ylabel('Growth Rate (%)',color='#BF9000')
ax.set_title('Comparison of Growth Rates by Vehicle Category',color='#BF9000')
plt.xticks( color='#BF9000')
plt.yticks( color='#BF9000')
ax.legend(loc='upper center')
plt.show()
```



Secondary Research Questions

In [120…
```
#1.What are the primary reasons for customers choosing 4-wheeler EVs in 2023 and 20
```

In [53]:
```
df3
```

Out[53]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	1-Apr-21	Sikkim	2-Wheelers	0	398
1	1-Apr-21	Sikkim	4-Wheelers	0	361
2	1-May-21	Sikkim	2-Wheelers	0	113
3	1-May-21	Sikkim	4-Wheelers	0	98
4	1-Jun-21	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
2440	1-Mar-24	Mizoram	2-Wheelers	58	1932
2441	1-Mar-24	DNH and DD	2-Wheelers	25	780
2442	1-Mar-24	Manipur	2-Wheelers	13	1394
2443	1-Mar-24	Andaman & Nicobar Island	2-Wheelers	2	447
2444	1-Mar-24	Nagaland	2-Wheelers	2	1180

2445 rows × 5 columns

In [55]:
```
df3['date']=pd.to_datetime(df3['date'])
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_36344\3029150507.py:1: UserWarning: Coul
d not infer format, so each element will be parsed individually, falling back to `da
teutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df3['date']=pd.to_datetime(df3['date'])
```

In [56]:
```
Ndf3_2023=df3[(df3['date']>'2022-03-31') & (df3['date']<'2023-04-01')]
Ndf3_2024=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [57]:
```
Ndf3_2w_2023 = Ndf3_2023[Ndf3_2023['vehicle_category'] == '2-Wheelers'].groupby('ve
Ndf3_2w_2024 = Ndf3_2024[Ndf3_2024['vehicle_category'] == '2-Wheelers'].groupby('ve
```

```
Ndf3_4w_2023 = Ndf3_2023[Ndf3_2023['vehicle_category'] == '4-Wheelers'].groupby('ve
Ndf3_4w_2024 = Ndf3_2024[Ndf3_2024['vehicle_category'] == '4-Wheelers'].groupby('ve
```

In [58]:
```
Nx2wh={'Ndf3_2023':Ndf3_2w_2023,'Ndf3_2024':Ndf3_2w_2024}
Nx2wh=pd.DataFrame(Nx2wh)
```

In [59]:
```
Nx4wh={'Ndf3_2023':Ndf3_4w_2023,'Ndf3_2024':Ndf3_4w_2024}
Nx4wh=pd.DataFrame(Nx4wh)
```

In [60]:
```
Nx=pd.concat([Nx2wh,Nx4wh],join='inner')
```

In [61]:
```
Nx
```

Out[61]:

vehicle_category	Ndf3_2023	Ndf3_2024
2-Wheelers	727903	932692
4-Wheelers	47465	86901

In [62]:
```
Nx.to_csv('secop1.csv')
```

clearly customer are chosing more 2_wheeler EVs in 2023 and 2024 as compared to 4_wheeler Evs. source:https://e-amrit.niti.gov.in/ Lower running cost Lower maintianance cost Zero tailpipe emissions No noise polllution Ease of charging at home Tax and Financial benefits: FAME II benefits plus additional state wise incentives (Purchase Incentives: Direct discount provided to the user on the cost of the electric vehicle Coupons: Financial incentive where the amount is reimbursed later Interest Subventions:Discount offered on the interest rate while availing loan Road tax exemption:Road tax at the time of purchase is waived off Registration fee exemption:One-time registration fee applicable on new vehicle purchase is waived off Income tax benefit: Provided as a deduction on the tax amount payable by an individual to the government Scrapping incentives:Provided upon de-registering old Petrol and Diesel Vehicles Others: Incentives such as interest-free loans, top-up subsidies, special incentives on electric three-wheelers, etc. can also be availed)

In [130…
```
#2.How do government incentives and subsidies impact the adoption rates of 2-wheele
```

Information collected from "www.tatacapital.com" and "www.godigit.com" Features of Phase II of FAME(“Faster Adoption and Manufacturing of Electric and Hybrid Vehicles in India”) India Scheme:- FAME phase II valid till 31st Mar 2024 The second phase of Fame India Scheme stresses on electrification of public transportation and shared transportation. This phase gets budgetary support of ₹ 10,000 crores. Through this scheme, the concerned department aims to provide incentives to various categories of vehicles. These are- Electric Two-wheelers: 10 lakh.Registered electric two-wheelers will get an incentive of ₹ 20,000 each. Electric Four-wheelers: 35,000 electric 4-wheelers with ex-factory price of ₹ 15 lakh will get an incentive of ₹ 1.5 lakh each. Hybrid Four-wheelers: Through this scheme, the Government will provide ₹ 13,000 - ₹ 20,000 as an incentive to hybrid 4-wheelers with ex-factory price of ₹ 15 lakh. e-rickshaws: 5 lakh e-rickshaws (each) can avail ₹ 50,000 as incentives. e-buses: Nearly 8000 e-buses with a maximum ex-factory price of ₹ 2 crores will receive an incentive of ₹ 50 lakh each. Under the second phase of Fame India Scheme, the Government is hopeful of establishing 2700 charging stations in metros, smart cities, hilly states, million-plus cities across the country. The grid measurement will follow a 3 km x 3 km layout. The Government aims to cover highways as well and establish charging stations on both sides of the road with a gap of 25 km between two consecutive stations. Top EV-friendly States that Offer Best Incentives to Buyers: 1>Gujarat: Two-wheeler: Maximum up to Rs. 20,000 Three-wheeler: Benefits up to Rs. 50,000 Four-wheeler: Maximum up to Rs. 1.5 lakh 2>Maharashtra Two-wheeler: Maximum up to Rs. 25,000 Three-wheeler: Benefits up to Rs. 30,000 Four-wheeler: Maximum up to Rs. 2.5 lakh 3>Meghalaya Two-wheeler: Maximum up to Rs. 20,000 Three-wheeler: NA Four-wheeler: Maximum up to Rs. 60,000 4>Karnataka:Besides the FAME EV subsidy, Karnataka doesn’t extend any direct incentives to EV buyers. But it gives electric car subsidy to EV makers. 5>Andhra Pradesh: Andhra Pradesh also doesn’t directly offer electric car subsidy to buyers but it offers them an exemption from registration charges and road tax. 6>Telangana: Telangana offers EV buyers complete exemption from registration charges and road tax besides the FAME incentive. 7>Tamil Nadu: Tamil Nadu offers a 100% road tax waiver and zero registration charges for EV buyers. However, the state government is looking to reconsider its EV policy. 8>Kerala: EV buyers in Kerala can get a 50% discount on road tax for the first five years. The state also gives a subsidy to e-rickshaws ranging from 10,000 to 30,000.

In [268…
```
#Let's find out the impact of subsidies for two best states:Gujarat and Maharastra
```

In [63]:
```
df3['date'] = pd.to_datetime(df3['date'], format='%Y-%m-%d')
```

In [64]:
```
df3_2024=df3[(df3['date']>'2023-03-31') & (df3['date']<'2024-04-01')]
```

In [65]:
```
df3_2024.loc[:, 'month'] = df3_2024['date'].dt.strftime('%b')
```

```
C:\Users\adian\AppData\Local\Temp\ipykernel_36344\1349304033.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  df3_2024.loc[:, 'month'] = df3_2024['date'].dt.strftime('%b')
```

In [66]:
```
G=df3_2024[df3_2024['state']=='Gujarat']
```

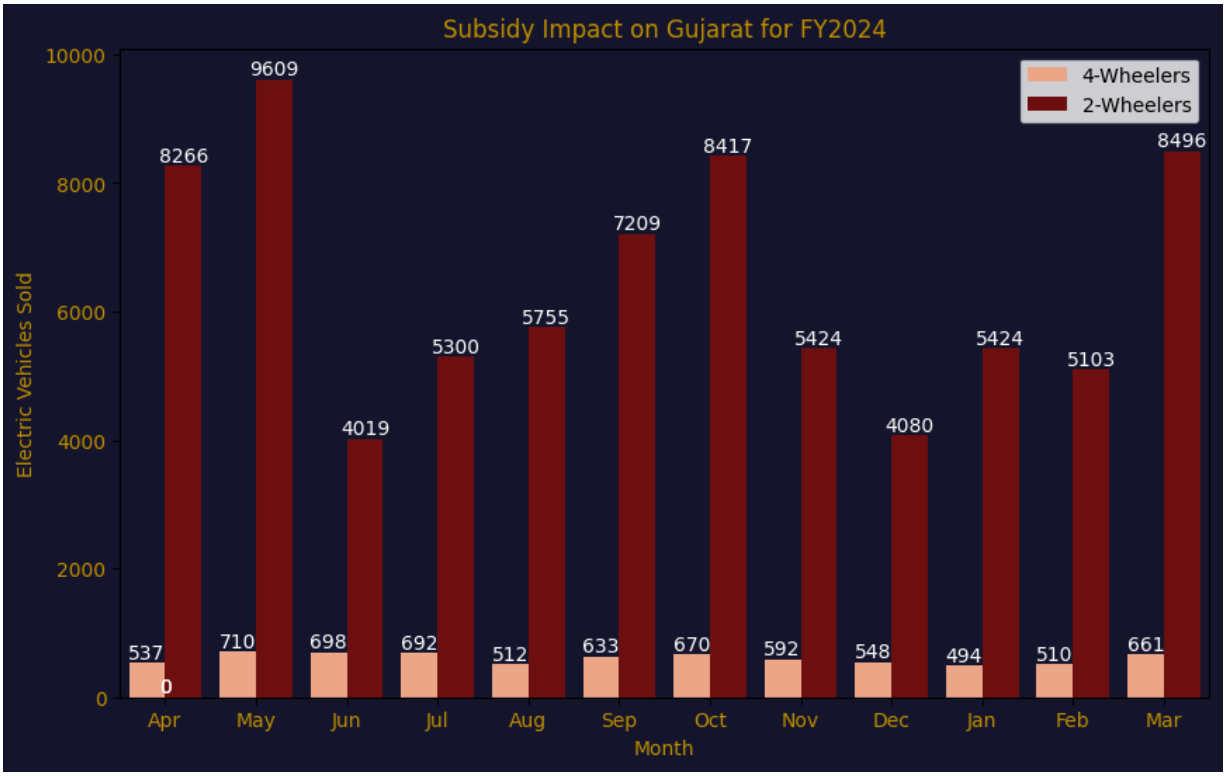
In [67]:
```
G
```

Out[67]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold	month
1118	2023-04-01	Gujarat	4-Wheelers	537	28451	Apr
1144	2023-05-01	Gujarat	4-Wheelers	710	25770	May
1171	2023-06-01	Gujarat	4-Wheelers	698	24829	Jun
1200	2023-07-01	Gujarat	4-Wheelers	692	27755	Jul
1229	2023-08-01	Gujarat	4-Wheelers	512	25572	Aug
1258	2023-09-01	Gujarat	4-Wheelers	633	36818	Sep
1286	2023-10-01	Gujarat	4-Wheelers	670	42365	Oct
1314	2023-11-01	Gujarat	4-Wheelers	592	32601	Nov
1344	2023-12-01	Gujarat	4-Wheelers	548	26023	Dec
1372	2024-01-01	Gujarat	4-Wheelers	494	33379	Jan
1399	2024-02-01	Gujarat	4-Wheelers	510	25758	Feb
1427	2024-03-01	Gujarat	4-Wheelers	661	25424	Mar
2100	2023-04-01	Gujarat	2-Wheelers	8266	89782	Apr
2129	2023-05-01	Gujarat	2-Wheelers	9609	87169	May
2159	2023-06-01	Gujarat	2-Wheelers	4019	80898	Jun
2188	2023-07-01	Gujarat	2-Wheelers	5300	98394	Jul
2217	2023-08-01	Gujarat	2-Wheelers	5755	87134	Aug
2246	2023-09-01	Gujarat	2-Wheelers	7209	129851	Sep
2274	2023-10-01	Gujarat	2-Wheelers	8417	149414	Oct

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold	month
2303	2023-11-01	Gujarat	2-Wheelers	5424	152556	Nov
2334	2023-12-01	Gujarat	2-Wheelers	4080	85848	Dec
2363	2024-01-01	Gujarat	2-Wheelers	5424	91933	Jan
2393	2024-02-01	Gujarat	2-Wheelers	5103	88441	Feb
2422	2024-03-01	Gujarat	2-Wheelers	8496	94822	Mar

In [72]:
```
custom_palette = ['#FFA07A', 'maroon']
plt.figure(figsize=(10, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')
ax = sns.barplot(x='month', y='electric_vehicles_sold', data=G, hue='vehicle_catego
plt.title('Subsidy Impact on Gujarat for FY2024',color='#BF9000')
plt.xlabel('Month', color='#BF9000')
plt.ylabel('Electric Vehicles Sold', color='#BF9000')

plt.xticks(color='#BF9000')
plt.yticks(color='#BF9000')

for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='bottom', color='white')
plt.legend(facecolor='white', edgecolor='#18182F',loc=1, fontsize=10)
plt.show()
```



From Jun2023 to oct 2023 sales have constantly increased for two wheelers FAME II subsidy is extended till 31st July 2024. So,we can expect more sales in upcoming months.

In [73]:
```
M=df3_2024[df3_2024['state']=='Maharashtra']
```

In [74]:
```
M
```

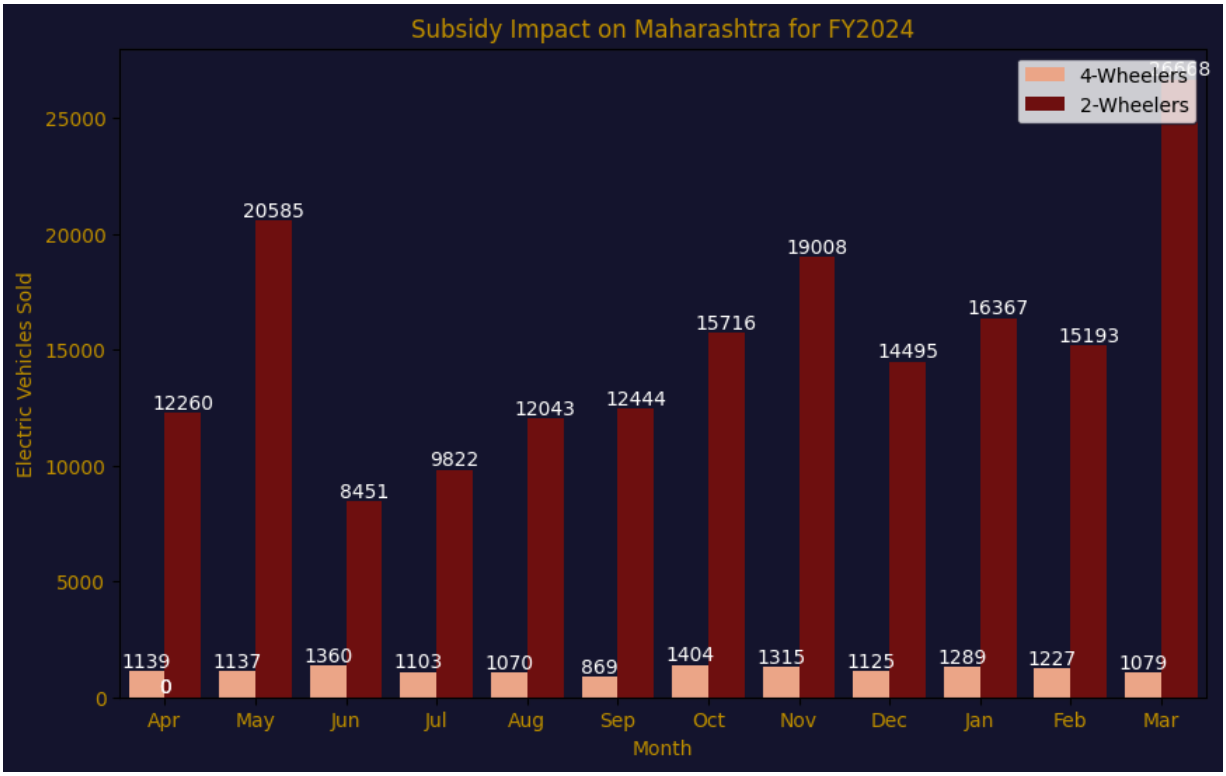
Out[74]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold	montl
1126	2023-04-01	Maharashtra	4-Wheelers	1139	35336	Ap
1152	2023-05-01	Maharashtra	4-Wheelers	1137	35179	Ma
1179	2023-06-01	Maharashtra	4-Wheelers	1360	36069	Ju
1208	2023-07-01	Maharashtra	4-Wheelers	1103	36082	Ju
1237	2023-08-01	Maharashtra	4-Wheelers	1070	40169	Au
1266	2023-09-01	Maharashtra	4-Wheelers	869	43063	Se
1294	2023-10-01	Maharashtra	4-Wheelers	1404	47126	Oc
1322	2023-11-01	Maharashtra	4-Wheelers	1315	48131	No
1352	2023-12-01	Maharashtra	4-Wheelers	1125	32212	De
1364	2024-01-01	Maharashtra	4-Wheelers	1289	50156	Ja
1392	2024-02-01	Maharashtra	4-Wheelers	1227	36302	Fe
1420	2024-03-01	Maharashtra	4-Wheelers	1079	36826	Ma
2108	2023-04-01	Maharashtra	2-Wheelers	12260	130977	Ap
2138	2023-05-01	Maharashtra	2-Wheelers	20585	126680	Ma
2168	2023-06-01	Maharashtra	2-Wheelers	8451	119809	Ju
2196	2023-07-01	Maharashtra	2-Wheelers	9822	126943	Ju
2225	2023-08-01	Maharashtra	2-Wheelers	12043	133378	Au
2254	2023-09-01	Maharashtra	2-Wheelers	12444	134556	Se
2282	2023-10-01	Maharashtra	2-Wheelers	15716	167065	Oc

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold	month
2311	2023-11-01	Maharashtra	2-Wheelers	19008	230768	No
2342	2023-12-01	Maharashtra	2-Wheelers	14495	174125	De
2355	2024-01-01	Maharashtra	2-Wheelers	16367	172830	Ja
2386	2024-02-01	Maharashtra	2-Wheelers	15193	147649	Fe
2415	2024-03-01	Maharashtra	2-Wheelers	26668	152563	Ma

In [77]:

```
custom_palette = ['#FFA07A', 'maroon']
plt.figure(figsize=(10, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')
ax = sns.barplot(x='month', y='electric_vehicles_sold', data=M, hue='vehicle_catego
plt.title('Subsidy Impact on Maharashtra for FY2024',color='#BF9000')
plt.xlabel('Month', color='#BF9000')
plt.ylabel('Electric Vehicles Sold', color='#BF9000')

plt.xticks(color='#BF9000')
plt.yticks(color='#BF9000')

for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='bottom', color='white')
plt.legend(facecolor='white', edgecolor='#18182F',loc=1, fontsize=10)
plt.show()
```



From Jun2023 to Nov 2023 2_wheelers sales wer growing. 4_wheelers sale is stagnant for Maharastra also. FAME II subsidy is extended till 31st July 2024. So,we can expect more sales in upcoming months.

In [133…
```
#3.How does the availability of charging stations infrastructure correlate with the
```

First we will find top5_states_EV_sales then top5_states_penetration_rate. Then we will make comparision of (No. of PCS with Ev sales) for top5_state_EV_sales And comparision of (No. of PCS with penetration rate) for top5_state_penetration_ratedata imported from https://www.data.gov.in/ [PCS= Public_EV_Charging_Stations] [Data collected till 31st mar2024]

In [6]:
```
df4=pd.read_csv(r"C:\Users\adian\Downloads\study\python\EV project\datasets\PCS_31M
```

In [7]:
```
df4
```

Out[7]:

	state	No. of PCS
0	Andaman & Nicobar Island	3
1	Andhra Pradesh	327
2	Arunachal Pradesh	9
3	Assam	86
4	Bihar	124
5	Chandigarh	12
6	Chhattisgarh	149
7	Delhi	1886
8	Goa	113
9	Gujarat	476
10	Haryana	377
11	Himachal Pradesh	44
12	Jammu and Kashmir	47
13	Jharkhand	135
14	Karnataka	5130
15	Kerala	958
16	Lakshadweep	1
17	Madhya Pradesh	341
18	Maharashtra	3083
19	Manipur	17
20	Meghalaya	21
21	Nagaland	6
22	Odisha	198
23	Puducherry	23
24	Punjab	158
25	Rajasthan	500
26	Sikkim	2
27	Tamil Nadu	643
28	Telangana	483
29	Tripura	18

	state	No. of PCS
30	DNH and DD	1
31	Uttar Pradesh	583
32	Uttarakhand	76
33	West Bengal	318
34	Mizoram	0
35	Ladakh	0

In [8]:
```
df4.head().to_csv('opdf4.csv')
```

In [9]:
```
df3
```

Out[9]:

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
0	1-Apr-21	Sikkim	2-Wheelers	0	398
1	1-Apr-21	Sikkim	4-Wheelers	0	361
2	1-May-21	Sikkim	2-Wheelers	0	113
3	1-May-21	Sikkim	4-Wheelers	0	98
4	1-Jun-21	Sikkim	2-Wheelers	0	229
...	...	...	...	...	...
2440	1-Mar-24	Mizoram	2-Wheelers	58	1932
2441	1-Mar-24	DNH and DD	2-Wheelers	25	780
2442	1-Mar-24	Manipur	2-Wheelers	13	1394
2443	1-Mar-24	Andaman & Nicobar Island	2-Wheelers	2	447
2444	1-Mar-24	Nagaland	2-Wheelers	2	1180

2445 rows × 5 columns

In [10]:
```
df=df3.groupby('state')[['electric_vehicles_sold','total_vehicles_sold']].sum()
```

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

Out[11]:

state	electric_vehicles_sold	total_vehicles_sold
Andaman & Nicobar Island	80	18885
Andhra Pradesh	77422	2283871
Arunachal Pradesh	33	71547
Assam	6418	1403271
Bihar	31019	3048373
Chandigarh	5279	130628
Chhattisgarh	53804	1334989
DNH and DD	355	43397
Delhi	107312	1588436
Goa	19684	199970
Gujarat	181389	4125551
Haryana	30797	1902768
Himachal Pradesh	2595	325366
Jammu and Kashmir	5971	414553
Jharkhand	18461	1364886
Karnataka	312995	3994329
Kerala	137060	2064677
Ladakh	68	10496
Madhya Pradesh	78979	3498698
Maharashtra	396045	6101429
Manipur	299	104513
Meghalaya	177	90183
Mizoram	340	71307
Nagaland	13	44092
Odisha	78267	1688794
Puducherry	5536	150798
Punjab	23833	1543962
Rajasthan	150366	3307591
Sikkim	0	29346

state	electric_vehicles_sold	total_vehicles_sold
Tamil Nadu	200062	4652363
Tripura	562	125216
Uttar Pradesh	95203	8127084
Uttarakhand	15127	622707
West Bengal	30560	2736176

In [12]:
```
df['penetration_rate']=(df['electric_vehicles_sold']/df['total_vehicles_sold'])*100
```

In [13]:
```
df
```

Out[13]:

state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
Andaman & Nicobar Island	80	18885	0.423617
Andhra Pradesh	77422	2283871	3.389946
Arunachal Pradesh	33	71547	0.046124
Assam	6418	1403271	0.457360
Bihar	31019	3048373	1.017559
Chandigarh	5279	130628	4.041247
Chhattisgarh	53804	1334989	4.030295
DNH and DD	355	43397	0.818029
Delhi	107312	1588436	6.755828
Goa	19684	199970	9.843477
Gujarat	181389	4125551	4.396722
Haryana	30797	1902768	1.618537
Himachal Pradesh	2595	325366	0.797563
Jammu and Kashmir	5971	414553	1.440347
Jharkhand	18461	1364886	1.352567
Karnataka	312995	3994329	7.835984
Kerala	137060	2064677	6.638326
Ladakh	68	10496	0.647866
Madhya Pradesh	78979	3498698	2.257383
Maharashtra	396045	6101429	6.491020
Manipur	299	104513	0.286089
Meghalaya	177	90183	0.196268
Mizoram	340	71307	0.476812
Nagaland	13	44092	0.029484
Odisha	78267	1688794	4.634491
Puducherry	5536	150798	3.671136
Punjab	23833	1543962	1.543626
Rajasthan	150366	3307591	4.546088
Sikkim	0	29346	0.000000

state	electric_vehicles_sold	total_vehicles_sold	penetration_rate
Tamil Nadu	200062	4652363	4.300223
Tripura	562	125216	0.448824
Uttar Pradesh	95203	8127084	1.171429
Uttarakhand	15127	622707	2.429232
West Bengal	30560	2736176	1.116887

In [17]:
```
x=pd.merge(df4,df,on='state',how='inner').reset_index(drop=True)
```

In [18]:
```
x.drop(columns='total_vehicles_sold', inplace=True)
```

In [119…
```
x
```

Out[119...

	state	No. of PCS	electric_vehicles_sold	penetration_rate
0	Andaman & Nicobar Island	3	80	0.423617
1	Andhra Pradesh	327	77422	3.389946
2	Arunachal Pradesh	9	33	0.046124
3	Assam	86	6418	0.457360
4	Bihar	124	31019	1.017559
5	Chandigarh	12	5279	4.041247
6	Chhattisgarh	149	53804	4.030295
7	Delhi	1886	107312	6.755828
8	Goa	113	19684	9.843477
9	Gujarat	476	181389	4.396722
10	Haryana	377	30797	1.618537
11	Himachal Pradesh	44	2595	0.797563
12	Jammu and Kashmir	47	5971	1.440347
13	Jharkhand	135	18461	1.352567
14	Karnataka	5130	312995	7.835984
15	Kerala	958	137060	6.638326
16	Madhya Pradesh	341	78979	2.257383
17	Maharashtra	3083	396045	6.491020
18	Manipur	17	299	0.286089
19	Meghalaya	21	177	0.196268
20	Nagaland	6	13	0.029484
21	Odisha	198	78267	4.634491
22	Puducherry	23	5536	3.671136
23	Punjab	158	23833	1.543626
24	Rajasthan	500	150366	4.546088
25	Sikkim	2	0	0.000000
26	Tamil Nadu	643	200062	4.300223
27	Tripura	18	562	0.448824
28	DNH and DD	1	355	0.818029
29	Uttar Pradesh	583	95203	1.171429

	state	No. of PCS	electric_vehicles_sold	penetration_rate
30	Uttarakhand	76	15127	2.429232
31	West Bengal	318	30560	1.116887
32	Mizoram	0	340	0.476812
33	Ladakh	0	68	0.647866

In [137…
```
x.to_csv('correlation.csv',index= False)
```

In [78]:
```
a=x.sort_values(by='electric_vehicles_sold', ascending=False).reset_index(drop=True
b=x.sort_values(by='penetration_rate', ascending=False).reset_index(drop=True).head
```

In [102…
```
a
```

Out[102…

	state	No. of PCS	electric_vehicles_sold	penetration_rate
0	Maharashtra	3083	396045	6.491020
1	Karnataka	5130	312995	7.835984
2	Tamil Nadu	643	200062	4.300223
3	Gujarat	476	181389	4.396722
4	Rajasthan	500	150366	4.546088

In [117…
```
a['state']
```

Out[117…
```
0    Maharashtra
1      Karnataka
2     Tamil Nadu
3        Gujarat
4      Rajasthan
Name: state, dtype: object
```

In [118…
```
b['state']
```

Out[118…
```
0           Goa
1     Karnataka
2         Delhi
3        Kerala
4    Maharashtra
Name: state, dtype: object
```

In [80]:
```
b
```

Out[80]:

	state	No. of PCS	electric_vehicles_sold	penetration_rate
0	Goa	113	19684	9.843477
1	Karnataka	5130	312995	7.835984
2	Delhi	1886	107312	6.755828
3	Kerala	958	137060	6.638326
4	Maharashtra	3083	396045	6.491020

In [90]:
```
a.to_csv('secop3a.csv',index=False)
b.to_csv('secop3b.csv',index=False)
```

In [105…
```
a.columns = a.columns.str.strip()
```

In [106…
```
spearman_corr_a = spearmanr(a['No. of PCS'], a['electric_vehicles_sold'])
```

In [111…
```
b.columns = b.columns.str.strip()
```

In [115…
```
spearman_corr_b = spearmanr(b['No. of PCS'], b['penetration_rate'])
```

In [113…
```
spearman_corr_a
```

Out[113…
```
SignificanceResult(statistic=0.7999999999999999, pvalue=0.10408803866182788)
```

In [116…
```
spearman_corr_b
```

Out[116…
```
SignificanceResult(statistic=-0.3, pvalue=0.6238376647810728)
```

since in both the cases the pvalue is more than o.o5, the statistics become insignificant. We can't draw any conclusions between these varaiables for top5 states only.The correlation statistics can be understood using only more data. so, we will be using dataframe x(all states data) we will try to find spearman correlation among ['No. of PCS', 'electric_vehicles_sold', 'penetration_rate'] for all states

In [129…
```
x.columns = x.columns.str.strip()
```

In [126…
```
data = x[['No. of PCS', 'electric_vehicles_sold', 'penetration_rate']]
```

In [127…
```
correlation_matrix = data.corr(method='spearman')     # since data distribution is
```

In [128…
```
correlation_matrix
```

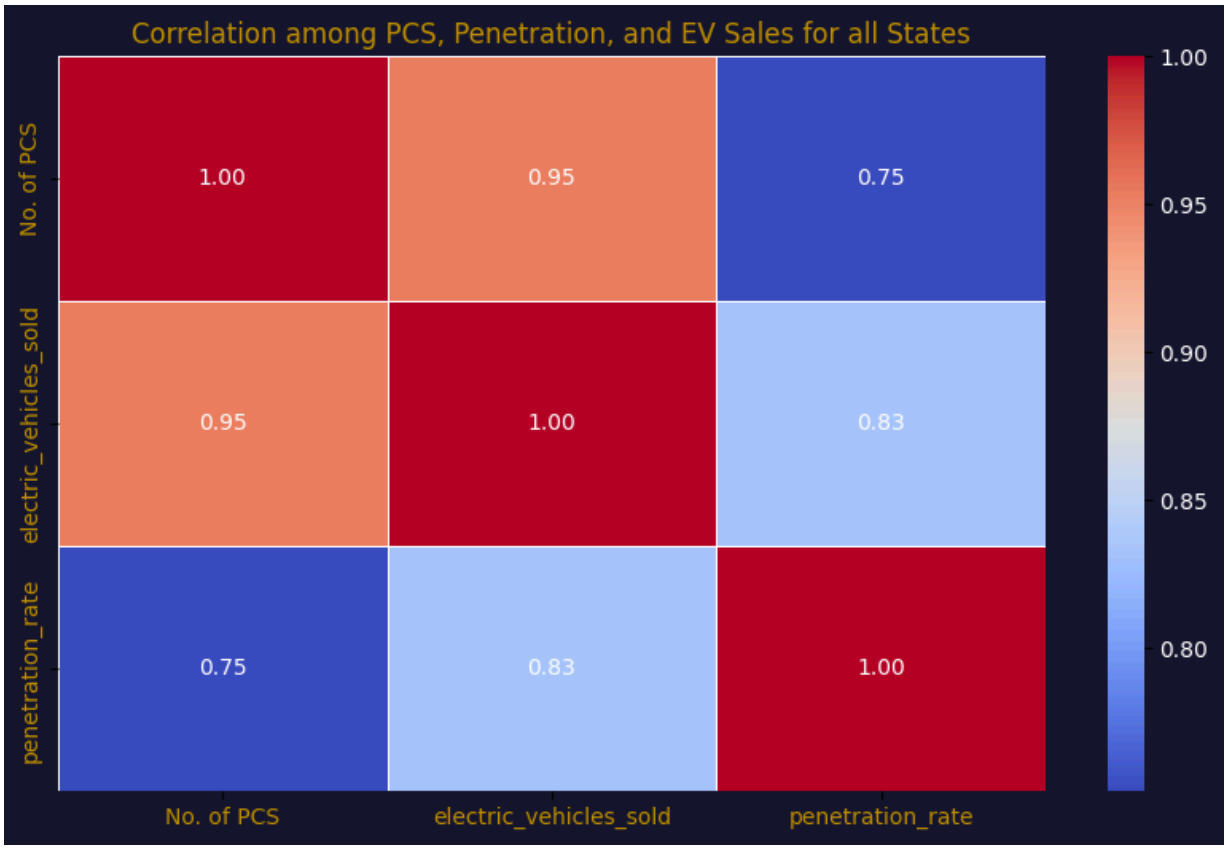
Out[128…

	No. of PCS	electric_vehicles_sold	penetration_rate
No. of PCS	1.000000	0.953320	0.751471
electric_vehicles_sold	0.953320	1.000000	0.833766
penetration_rate	0.751471	0.833766	1.000000

In [ ]:
```

```

In [ ]:

In [130…
```
plt.figure(figsize=(10, 6), facecolor='#18182F')
ax = plt.gca()
ax.set_facecolor('#18182F')
ax=sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5, fmt
plt.title('Correlation among PCS, Penetration, and EV Sales for all States',color='
plt.xticks(color='#BF9000')
plt.yticks(color='#BF9000')
colorbar = ax.collections[0].colorbar  # Access the color bar
colorbar.ax.tick_params(labelcolor='white')  # Set tick labels to white
plt.show()
```



In [ ]:
```
No. of PCS and electric_vehicles_sold (0.953320)
No. of PCS and penetration_rate (0.751471)
electric_vehicles_sold and penetration_rate (0.833766)

All the correlations are positive, indicating that increases in one variable are as
Significance of Charging Infrastructure: The strong correlations highlight the impo
in influencing both sales of electric vehicles and their penetration rates in the m
```

In [145…
```
#4.Who should be the brand ambassador if AtliQ Motors launches their EV/Hybrid vehi
```

Atliq Motor has the option to hire one of two youth icons, both renowned for their impressive bike collections: MS Dhoni and John Abraham. However, if the goal is to minimize costs while still associating with a prominent figure, they can consider hiring Neeraj Chopra, a celebrated personality admired by Indians across the nation.

In [145…
```
#5.Which state of India is ideal to start the manufacturing unit? (Based on subsidi
```

Maharashtra, Karnataka,Tamil Nadu, Gujarat have better EV penetration and sales than other states (based on datasets) Information collected from Internet: Based on subsidies: Gujarat Based on Ease of doing business:Gujarat, Tamilnadu Based on political

stability: Gujarat Based on Lithium reserve: Jharkhand,Rajasthan, Tamilnadu Based on OEM presence: Maharastra, Haryana, Gujarat Connecting port: Gujarat(Mundra), Maharastra(JNPT) Skilled Manpower: Maharastra, Tamilnadu,Haryana,Gujarat EV customer base: Maharastra Infrastructure and market for growth: Gujarat so, I will recommend Gujarat

In [146…
```
#6.Your top 3 recommendations for AtliQ Motors.
```

1>Create manufacturing unit in Gujarat nearby Mundra port 2>Procure Lithium from Nagaur district of Rajasthan 3>Transport Electric vehicles to Navi Mumbai JNPT port from Mundra port and use Maharastra's connectivity and largest customer base. Also use Customer demand of Rajasthan,Karnataka,Tamilnadu. Try to maximise the impact of Gujarat's infrastructure and market. 4> Create a research unit for alternative fuels like hydrogen or Aluminium. 5>> Initially, Atliq Motors can introduce and attract Indian customers by launching its premium EV products, which are best sellers in North America. However, it is crucial to consider the purchasing power of Indian consumers and the pricing strategies of competitors to ensure market fit.

In [ ]:

In [ ]:

In [ ]: