ELECTRIC VEHICLE ANALYSIS

CODEBASICS RESUME PROJECT CHALLENGE 12

Presented By Rana Basak

Project Overview

Overview: The objective of this project is to analyze electric vehicle sales data across Indian states to uncover trends, identify discrepancies, and provide insights for promoting electric mobility in the country.

Datasets Overview: This analysis uses datasets covering electric vehicle sales trends in India from April 2021 to April 2024, sourced from Codebasics with data from Vahan Sewa:

1.Electric Vehicle Sales by State: Monthly sales data categorized by vehicle type (2-Wheeler or 4-Wheeler) and state, including the date, number of EVs sold, and total vehicle sales.

2.Electric Vehicle Sales by Makers: Sales data by manufacturer, categorized by vehicle type, showing date, vehicle category, manufacturer, and number of EVs sold.

3.Date Dimension: Provides time-series analysis details, including specific dates, fiscal years, and quarters.

Data correction & standardization

Data Entry Error Correction: In the case of Kerala for the fiscal year 2024, a discrepancy was found where for the month of January the total vehicles sold were recorded as 164, while electric vehicles sold were 734.This inconsistency suggested a data entry error. To correct this, the total vehicle count was replaced with the median of total vehicle sales values for Kerala for the fiscal year 2024 in the 4-wheeler category. The median(13,932) was chosen over the mean due to the presence of the extreme outlier (164), which would have skewed the mean value.

IMPORT LIBRARIES

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

LOAD DATA

In [2]:

```
df1= pd.read_csv('electric_vehicle_sales_by_makers.csv')
df2 = pd.read_csv('electric_vehicle_sales_by_state.csv')
df3 = pd.read_csv('dim_date.csv')
```

In [67]:

```
df1.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 [68]:

```
df2.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   datetime64[ns]
 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: datetime64[ns](1), int64(2), object(2)
memory usage: 95.6+ KB
```

In [69]:

```
df3.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: 996.0+ bytes
```

PRIMARY QUESTIONS

List the top 3 and bottom 3 makers for the fiscal years 2023 and 2024 in terms of the number of 2-wheelers sold.

In [77]:

```
# Convert 'date' column to datetime format
df1['date'] = pd.to_datetime(df1['date'], format='%Y-%m-%d')

# Filter for 2-Wheelers sold in 2023
Filter = df1[(df1['date'].dt.year == 2023) & (df1['vehicle_category'] == '2-Wheelers')]

# Group by 'maker' and sum 'electric_vehicles_sold'
```

```
maker = Filter.groupby('maker').agg({'electric_vehicles_sold': 'sum'})

# Identify top and bottom 3 makers by electric vehicles sold
top_makers2023 = makers['electric_vehicles_sold'].nlargest(3)
bottom_makers2023 = makers['electric_vehicles_sold'].nsmallest(3)
```

In [80]:

```
top_makers2023
```

Out[80]:

```
maker
OLA ELECTRIC    271925
TVS             165030
ATHER           103812
Name: electric_vehicles_sold, dtype: int64
```

In [81]:

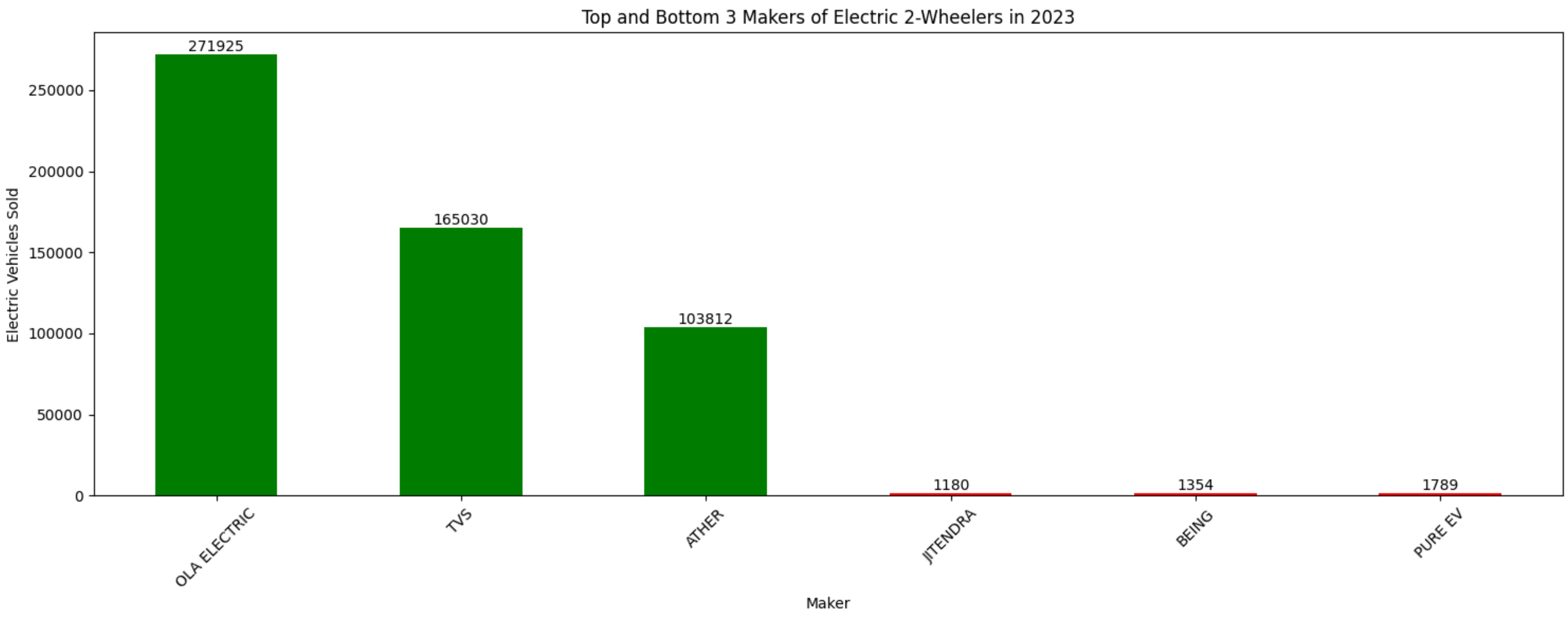
```
bottom_makers2023
```

Out[81]:

```
maker
JITENDRA    1180
BEING       1354
PURE EV     1789
Name: electric_vehicles_sold, dtype: int64
```

In [83]:

```
# Combine top and bottom makers for plotting
combined_makers = pd.concat([top_makers2023, bottom_makers2023])

# Create a bar plot
bars = combined_makers.plot(kind='bar', color=['green' if x in top_makers2023.index else 'red' for x in combined_makers.index]

# Annotate bars with values
for bar in bars.patches:
    height = bar.get_height()
    bars.annotate(f'{height}', (bar.get_x() + bar.get_width() / 2, height), ha='center', va='bottom', fontsize=10)

# Set plot titles and labels
plt.title('Top and Bottom 3 Makers of Electric 2-Wheelers in 2023')
plt.xlabel('Maker')
plt.ylabel('Electric Vehicles Sold')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [84]:

```
# Filter for 2-Wheelers sold in 2024
Filter2024= df1[(df1['date'].dt.year == 2024) & (df1['vehicle_category'] == '2-Wheelers')]

# Group by 'maker' and sum 'electric_vehicles_sold'
makers24 = Filter2024.groupby('maker').agg({'electric_vehicles_sold': 'sum'})

# Identify top and bottom 3 makers by electric vehicles sold
top_makers2024 = makers24['electric_vehicles_sold'].nlargest(3)
bottom_makers2024 = makers24['electric_vehicles_sold'].nsmallest(3)
```

In [85]:

```
top_makers2024
```

Out[85]:

```
maker
OLA ELECTRIC    110872
TVS              55742
BAJAJ            40072
Name: electric_vehicles_sold, dtype: int64
```

In [86]:

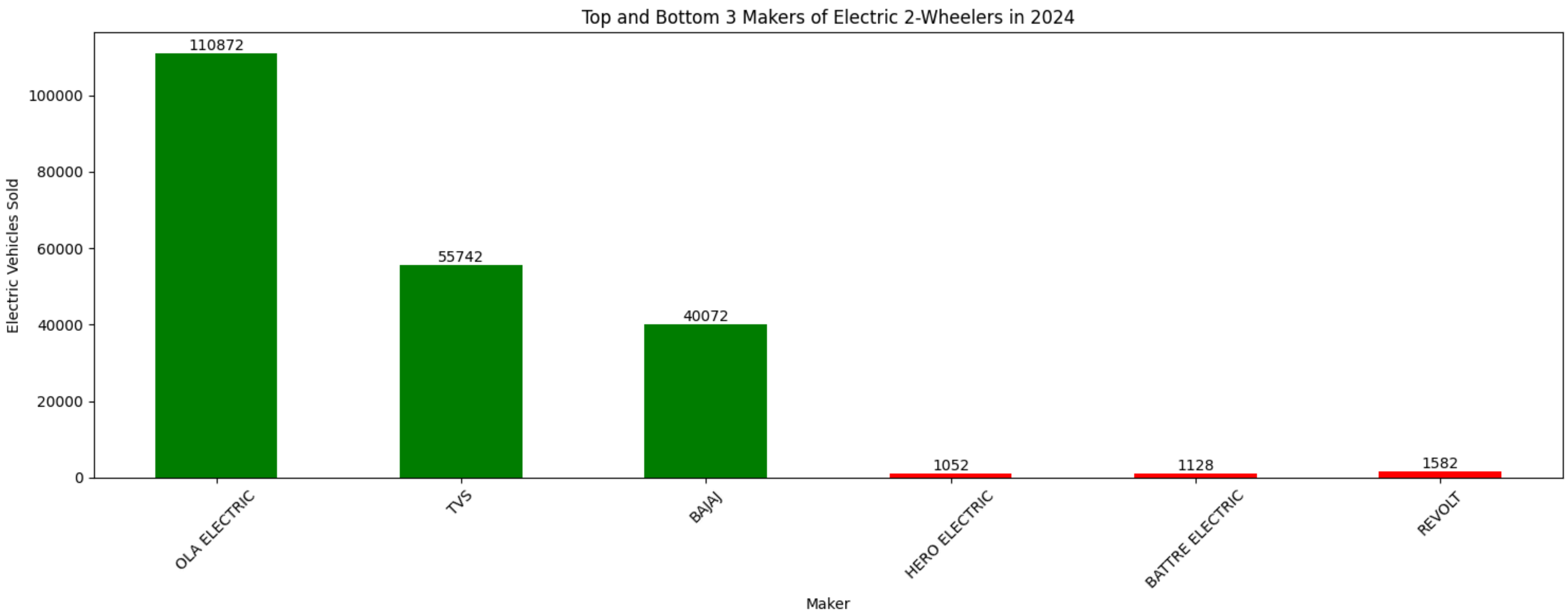
```
bottom_makers2024
```

Out[86]:

```
maker
HERO ELECTRIC      1052
BATTRE ELECTRIC    1128
REVOLT             1582
Name: electric_vehicles_sold, dtype: int64
```

In [89]:

```
# Combine top and bottom makers for plotting
combined_makers24 = pd.concat([top_makers2024, bottom_makers2024])

# Create a bar plot
bars = combined_makers24.plot(kind='bar', color=['green' if x in top_makers2024.index
                                                 else 'red' for x in combined_makers24.index],figsize = (15,6))

# Annotate bars with values
for bar in bars.patches:
    height = bar.get_height()
    bars.annotate(f'{height}', (bar.get_x() + bar.get_width() / 2, height), ha='center', va='bottom', fontsize=10)

# Set plot titles and labels
plt.title('Top and Bottom 3 Makers of Electric 2-Wheelers in 2024')
plt.xlabel('Maker')
plt.ylabel('Electric Vehicles Sold')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Identify the top 5 states with the highest penetration rate in 2-wheeler and 4-wheeler EV sales in FY 2024.

In [92]:

```
# Convert 'date' column to datetime format
df2['date'] = pd.to_datetime(df2['date'], format='%Y-%m-%d')

# Filter for 2-Wheelers sold in 2024
fd2 = df2[(df2['date'].dt.year == 2024) & (df2['vehicle_category'] == '2-Wheelers')]

# Group by 'state' and aggregate sales
grouped = (fd2.groupby('state')
            .agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'})
            .reset_index())

# Calculate percentage of electric vehicles sold
grouped['percentage'] = 100 * grouped['electric_vehicles_sold'] / grouped['total_vehicles_sold']
```

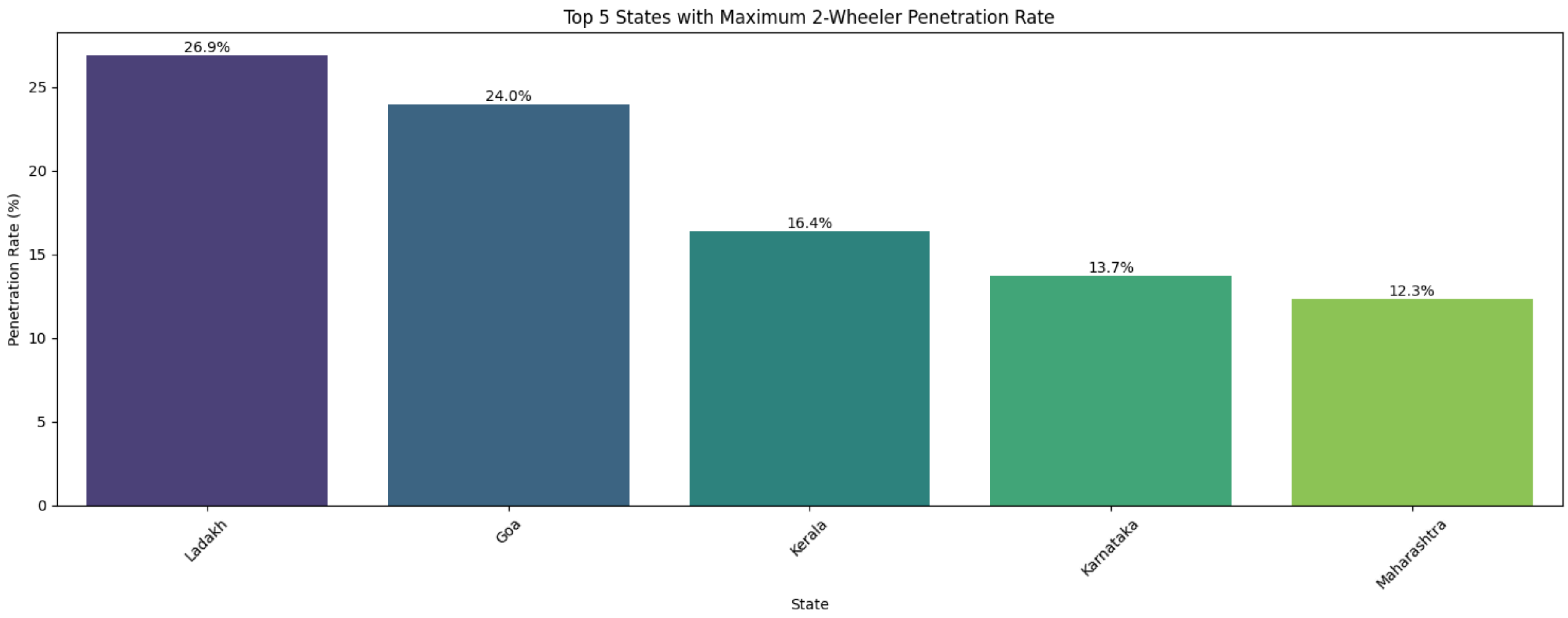
```python
# Get the top states by percentage
top_states = grouped.sort_values('percentage', ascending=False).head()
top_states
```

Out[92]:

	state	electric_vehicles_sold	total_vehicles_sold	percentage
18	Ladakh	14	52	26.923077
10	Goa	3567	14856	24.010501
17	Kerala	20239	123774	16.351576
16	Karnataka	47762	348572	13.702191
20	Maharashtra	58228	473042	12.309266

In [100…

```python
# Plotting the penetration rate of electric 2-wheelers by state
plt.figure(figsize = (15,6))
ax = sns.barplot(x='state', y='percentage', data=top_states ,hue = 'state' , palette = 'viridis')

# Set plot titles and labels
plt.title('Top 5 States with Maximum 2-Wheeler Penetration Rate')
plt.xlabel('State')
plt.ylabel('Penetration Rate (%)')
plt.xticks(rotation=45)

# Adding data labels on top of bars
for bar in ax.patches:
    ax.annotate(f'{bar.get_height():.1f}%',
                (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                ha='center', va='bottom', fontsize=10)
plt.tight_layout()
plt.show()
```



In [186…

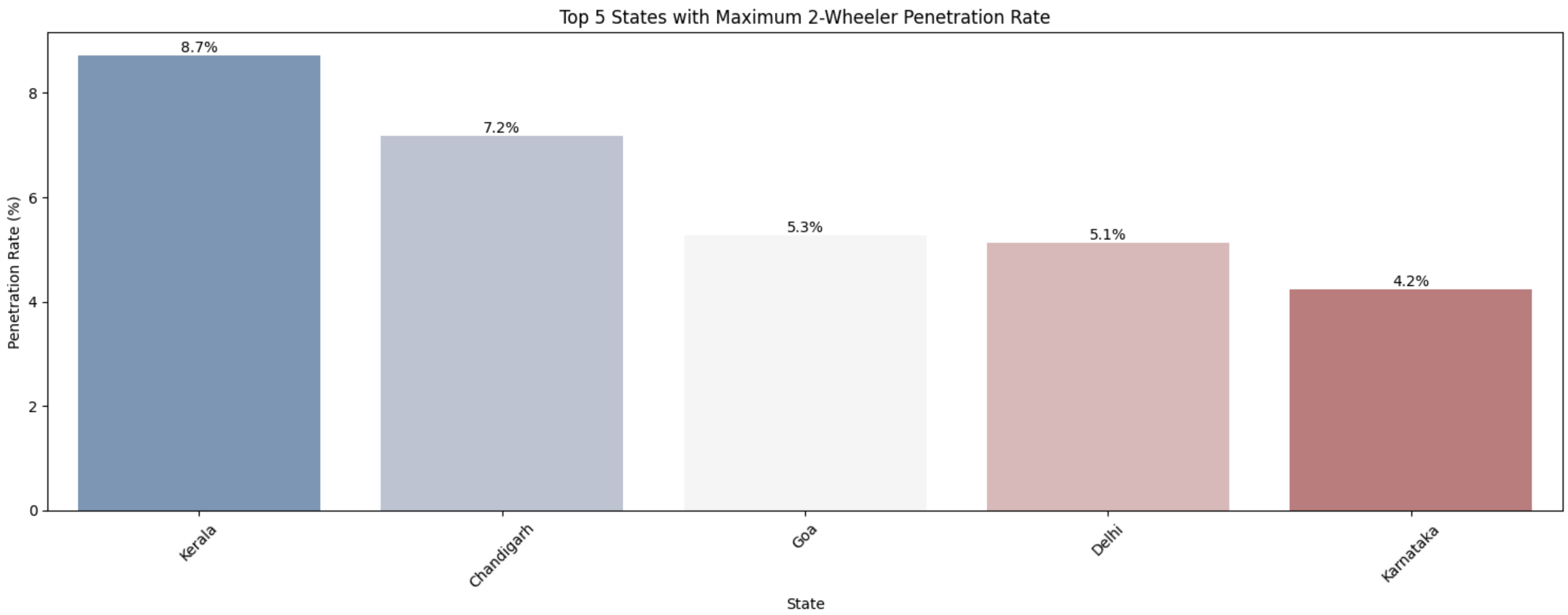
```python
# Filter for 2-Wheelers sold in 2024
fd4 = df2[(df2['date'].dt.year == 2024) & (df2['vehicle_category'] == '4-Wheelers')]

# Group by 'state' and aggregate sales
grouped4 = (fd4.groupby('state')
            .agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'})
            .reset_index())

# Calculate percentage of electric vehicles sold
grouped4['percentage'] = 100 * grouped4['electric_vehicles_sold'] / grouped4['total_vehicles_sold']

# Get the top states by percentage
top_states4 = grouped4.sort_values('percentage', ascending=False).head()
top_states4
```

Out[186…

	state	electric_vehicles_sold	total_vehicles_sold	percentage
17	Kerala	2294	26306	8.720444
6	Chandigarh	304	4239	7.171503
10	Goa	308	5833	5.280302
9	Delhi	2838	55323	5.129874
16	Karnataka	3485	82333	4.232811

In [109…

```
# Plotting the penetration rate of electric 2-wheelers by state
plt.figure(figsize = (15,6))
ay = sns.barplot( x='state', y='percentage', data = top_states4 , hue = 'state' ,palette ='vlag')

# Set plot titles and labels
plt.title('Top 5 States with Maximum 2-Wheeler Penetration Rate')
plt.xlabel('State')
plt.ylabel('Penetration Rate (%)')
plt.xticks(rotation=45)

# Add data labels on top of bars
for bar in ay.patches:
    ay.annotate(f'{bar.get_height():.1f}%',
                (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                ha='center', va='bottom', fontsize=10)

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



List the states with negative penetration (decline) in EV sales from 2022 to 2024?

In [110…

```
# Filter data for 2022 and calculate percentages
data2022 = df2[df2['date'].dt.year == 2022]
group2022 = data2022.groupby('state').agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'}).reset_index()
group2022['percentage2022'] = 100 * group2022['electric_vehicles_sold'] / group2022['total_vehicles_sold']
sort2022 = group2022.sort_values('percentage2022', ascending=False).tail()

# Filter data for 2024 and calculate percentages
data2024 = df2[df2['date'].dt.year == 2024]
group2024 = data2024.groupby('state').agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'}).reset_index()
group2024['percentage2024'] = 100 * group2024['electric_vehicles_sold'] / group2024['total_vehicles_sold']
sort2024 = group2024.sort_values('percentage2024', ascending=False).tail()

# Merge data for 2022 and 2024 and calculate the difference
```

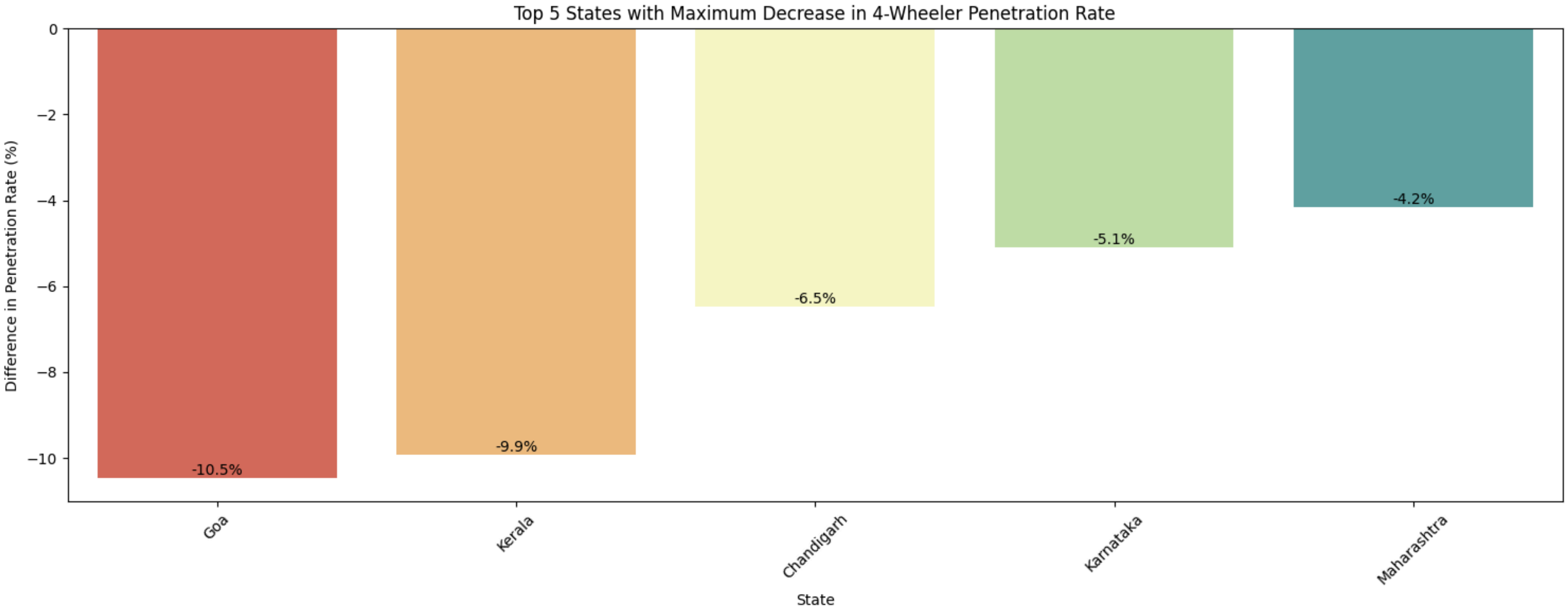
```python
merged_df = pd.merge(group2022, group2024, on='state')
merged_df['difference'] = merged_df['percentage2022'] - merged_df['percentage2024']

# Get states with the smallest decrease in penetration rate
result = merged_df[['state', 'difference']].sort_values('difference').head()
result
```

Out[110…

	state	difference
9	Goa	-10.472151
16	Kerala	-9.929776
5	Chandigarh	-6.483224
15	Karnataka	-5.099794
19	Maharashtra	-4.154834

In [115…

```python
# Plotting the difference in penetration rates
plt.figure(figsize=(15, 6))
ax = sns.barplot( x='state', y='difference',data = result,hue = 'state', palette ='Spectral')

# Set plot titles and labels
plt.title('Top 5 States with Maximum Decrease in 4-Wheeler Penetration Rate')
plt.xlabel('State')
plt.ylabel('Difference in Penetration Rate (%)')
plt.xticks(rotation=45)

# Add data labels on top of bars
for bar in ax.patches:
    ax.annotate(f'{bar.get_height():.1f}%',
                (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                ha='center', va='bottom', fontsize=10)

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



What are the quarterly trends based on sales volume for the top 5 EV makers (4-wheelers) from 2022 to 2024?

In [131…

```
# Convert 'date' column to datetime format
df3['date'] = pd.to_datetime(df3['date'],format='%Y-%m-%d')

# Merge df1 and df3 on 'date'
merge = pd.merge(df1, df3, on='date')

# Extract year from the 'date' column
merge['year'] = merge['date'].dt.year

# Filter for 4-wheelers sold from 2022 onwards
filter = merge[(merge['year'] >= 2022) & (merge['vehicle_category'] == '4-Wheelers')]

# Group by 'maker' and 'quarter', summing electric vehicles sold
```

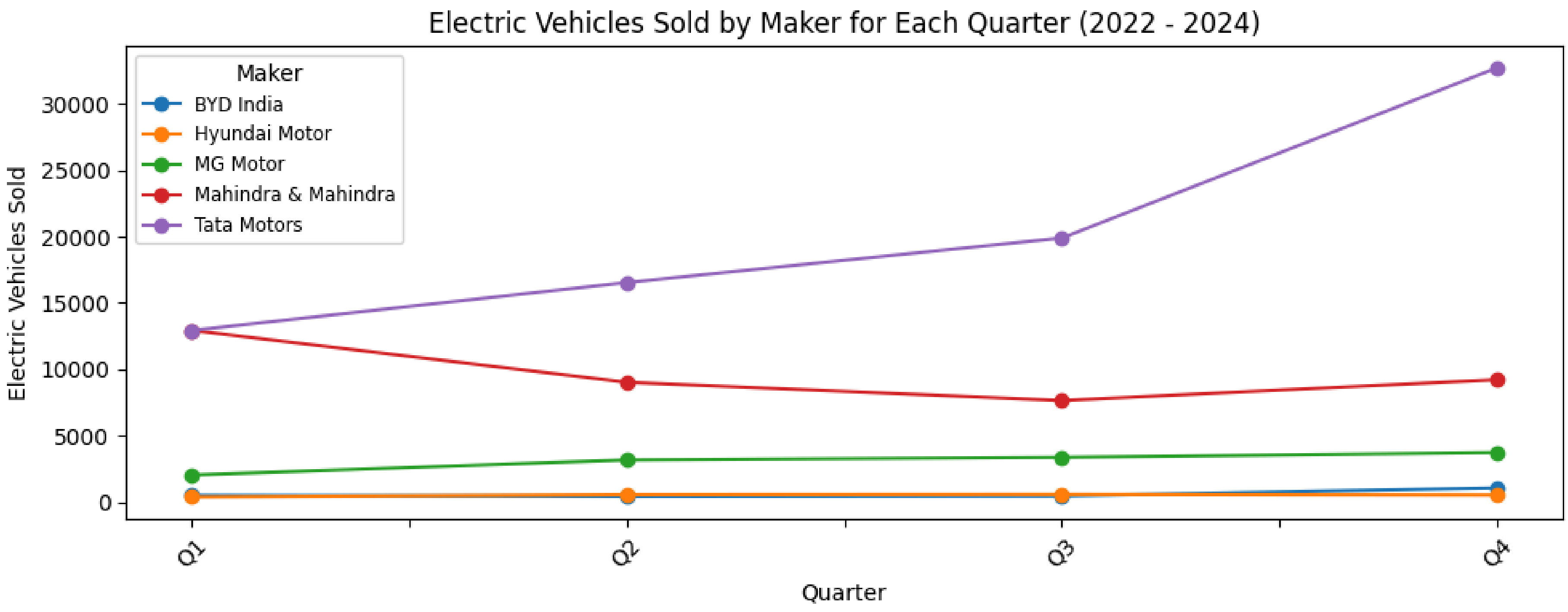
```python
group2 = filter.groupby(['maker', 'quarter']).agg({'electric_vehicles_sold': 'sum'})

# Unstack the data to separate quarters and calculate total sales
stack = group2.unstack(level='quarter')
stack['total'] = stack.sum(axis=1)

# Sort by total sales and get the top 5 makers
top5 = stack.sort_values('total', ascending=False).head()
top5
```

Out[131…

	electric_vehicles_sold				total
quarter	**Q1**	**Q2**	**Q3**	**Q4**	
maker					
Tata Motors	12922	16529	19887	32723	82061
Mahindra & Mahindra	12931	9019	7642	9212	38804
MG Motor	2024	3159	3355	3721	12259
BYD India	487	423	453	1055	2418
Hyundai Motor	367	545	561	519	1992

In [130…

```python
# Pivot the data for plotting
plot_data = top5.pivot_table(columns='maker', values='electric_vehicles_sold', aggfunc='sum')

# Plotting
plot_data.plot(kind='line', marker='o', figsize=(10, 4))

# Set plot titles and labels
plt.title('Electric Vehicles Sold by Maker for Each Quarter (2022 - 2024)')
plt.xlabel('Quarter')
plt.ylabel('Electric Vehicles Sold')
plt.xticks(rotation=45)
plt.legend(title='Maker', fontsize='small', title_fontsize='medium')
plt.tight_layout()
plt.show()
```



How do the EV sales and penetration rates in Delhi compare to Karnataka for 2024?

```
In [146…  df2.head()
```

Out[146…

	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

In [187…

```python
# Filter sales data for Delhi in 2024
delhi_sale = df2[(df2['state'] == 'Delhi') & (df2['date'].dt.year == 2024)]

# Filter sales data for Karnataka in 2024
karnataka_sale = df2[(df2['state'] == 'Karnataka') & (df2['date'].dt.year == 2024)]

karnataka_sale
```

Out[187…

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
1365	2024-01-01	Karnataka	4-Wheelers	1080	30770
1393	2024-02-01	Karnataka	4-Wheelers	923	24812
1422	2024-03-01	Karnataka	4-Wheelers	1482	26751
2356	2024-01-01	Karnataka	2-Wheelers	12415	115920
2387	2024-02-01	Karnataka	2-Wheelers	12605	108852
2417	2024-03-01	Karnataka	2-Wheelers	22742	123800

In [140…

```python
# Group sales data by electric vehicle category for Karnataka
evk = karnataka_sale.groupby('vehicle_category').agg({
    'electric_vehicles_sold': 'sum',
    'total_vehicles_sold': 'sum'
}).reset_index()

# Calculate the percentage of electric vehicles sold
evk['percentage'] = 100 * evk['electric_vehicles_sold'] / evk['total_vehicles_sold']

# Rename columns for clarity (if needed)
evk.rename(columns={
    'electric_vehicles_sold': 'Karnataka_EV_sold',
    'total_vehicles_sold': 'Total_Karnataka_EV_Sold',
    'percentage': 'Karnataka_Penetratio_rate'
}, inplace=True)

# Display the results
evk
```

Out[140…

	vehicle_category	Karnataka_EV_sold	Total_Karnataka_EV_Sold	Karnataka_Penetratio_rate
0	2-Wheelers	47762	348572	13.702191
1	4-Wheelers	3485	82333	4.232811

In [154…

```
delhi_sale
```

Out[154…

	date	state	vehicle_category	electric_vehicles_sold	total_vehicles_sold
1370	2024-01-01	Delhi	4-Wheelers	1345	21520
1400	2024-02-01	Delhi	4-Wheelers	596	17898
1425	2024-03-01	Delhi	4-Wheelers	897	15905
2361	2024-01-01	Delhi	2-Wheelers	3073	38413
2394	2024-02-01	Delhi	2-Wheelers	2364	30504
2420	2024-03-01	Delhi	2-Wheelers	5002	31599

In [137…

```
# Group sales data by vehicle category for Delhi
evd = delhi_sale.groupby('vehicle_category').agg({
    'electric_vehicles_sold': 'sum',
    'total_vehicles_sold': 'sum'
}).reset_index()

# Calculate the percentage of electric vehicles sold
evd['percentage'] = 100 * evd['electric_vehicles_sold'] / evd['total_vehicles_sold']

# Rename columns for clarity
evd.rename(columns={
    'electric_vehicles_sold': 'Delhi_EV_sold',
    'total_vehicles_sold': 'Total_delhi_EV_Sold',
    'percentage': 'Delhi_Penetratio_rate'
}, inplace=True)
```

```
# Display the results
evd
```

Out[137…

	vehicle_category	Delhi_EV_sold	Total_delhi_EV_Sold	Delhi_Penetratio_rate
0	2-Wheelers	10439	100516	10.385411
1	4-Wheelers	2838	55323	5.129874

In [141…

```
merge = pd.merge(evk,evd,on='vehicle_category')
merge
```

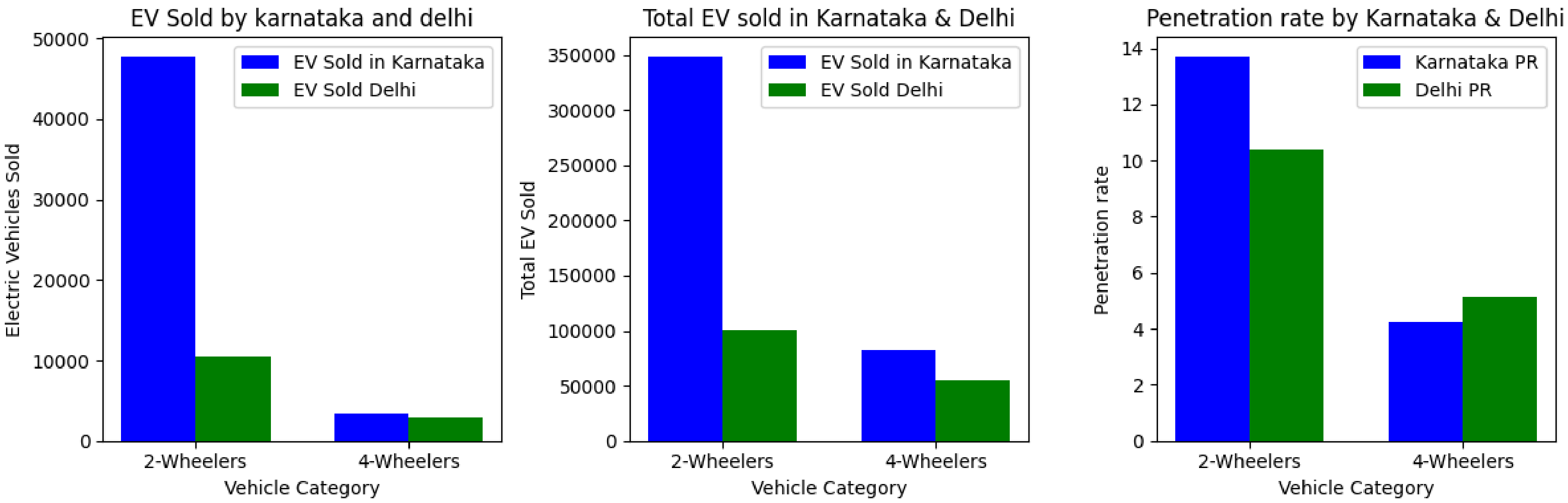
Out[141…

	vehicle_category	Karnataka_EV_sold	Total_Karnataka_EV_Sold	Karnataka_Penetratio_rate	Delhi_EV_sold	Total_delhi_EV_Sold	Delhi_Penet
0	2-Wheelers	47762	348572	13.702191	10439	100516	
1	4-Wheelers	3485	82333	4.232811	2838	55323	

In [142…

```
fig, ax = plt.subplots(1,3,figsize=(12, 4))

# Set the bar width and the positions of the bars
bar_width = 0.35
r1 = np.arange(len(merge['vehicle_category']))  # Positions for the first set of bars
r2 = [x + bar_width for x in r1]                # Positions for the second set of bars

# Create the bar plots
ax[0].bar(r1, merge['Karnataka_EV_sold'], width=bar_width, label='EV Sold in Karnataka', color='b')
ax[0].bar(r2, merge['Delhi_EV_sold'], width=bar_width, label='EV Sold Delhi', color='g')

ax[0].set_xlabel('Vehicle Category')
ax[0].set_ylabel('Electric Vehicles Sold')
ax[0].set_title('EV Sold by karnataka and delhi')

# Customize x-ticks
ax[0].set_xticks([r + bar_width / 2 for r in range(len(merge['vehicle_category']))])
ax[0].set_xticklabels(merge['vehicle_category'])

# Show legend
```

```
ax[0].legend()


# Create the bar plots
ax[1].bar(r1, merge['Total_Karnataka_EV_Sold'], width=bar_width, label='EV Sold in Karnataka', color='b')
ax[1].bar(r2, merge['Total_delhi_EV_Sold'], width=bar_width, label='EV Sold Delhi', color='g')

# Add labels and title
ax[1].set_xlabel('Vehicle Category')
ax[1].set_ylabel('Total EV Sold')
ax[1].set_title('Total EV sold in Karnataka & Delhi')

# Customize x-ticks
ax[1].set_xticks([r + bar_width / 2 for r in range(len(merge['vehicle_category']))])
ax[1].set_xticklabels(merge['vehicle_category'])

# Show legend
ax[1].legend()

# Create the bar plots
ax[2].bar(r1, merge['Karnataka_Penetratio_rate'], width=bar_width, label='Karnataka PR', color='b')
ax[2].bar(r2, merge['Delhi_Penetratio_rate'], width=bar_width, label='Delhi PR', color='g')

ax[2].set_xlabel('Vehicle Category')
ax[2].set_ylabel('Penetration rate')
ax[2].set_title('Penetration rate by Karnataka & Delhi')

# Customize x-ticks
ax[2].set_xticks([r + bar_width / 2 for r in range(len(merge['vehicle_category']))])
ax[2].set_xticklabels(merge['vehicle_category'])

# Show legend
ax[2].legend()

# Show grid and layout
#ax.grid(axis='y')  # Optional: Add gridlines to the y-axis
plt.tight_layout()
plt.show()
```



List down the compounded annual growth rate (CAGR) in 4-wheeler units for the top 5 makers from 2022 to 2024.

In [7]:

```
df1.head()
```

Out[7]:

	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

In [146…

```
# Filter for 4-wheelers sold in 2022
car2022 = df1[(df1['date'].dt.year == 2022) & (df1['vehicle_category'] == '4-Wheelers')]
group2022 = car2022.groupby('maker').agg({'electric_vehicles_sold': 'sum'})
```

```python
sort2022 = group2022.sort_values('electric_vehicles_sold', ascending=False).head()

# Filter for 4-wheelers sold in 2024
car2024 = df1[df1['vehicle_category'] == '4-Wheelers']
group2024 = car2024.groupby('maker').agg({'electric_vehicles_sold': 'sum'})
sort2024 = group2024.sort_values('electric_vehicles_sold', ascending=False).head()

# Merge the results from 2022 and 2024
merged = pd.merge(sort2022, sort2024, on='maker')

# Calculate the Compound Annual Growth Rate (CAGR)
merged['CAGR'] = (((merged['electric_vehicles_sold_y'] / merged['electric_vehicles_sold_x']) ** (1/2)) - 1)

# Rename columns for clarity and reset index
dm = merged.rename(columns={
    'electric_vehicles_sold_x': '2022 EV Sold',
    'electric_vehicles_sold_y': 'Total EV Sold'
}).reset_index()
dm
```

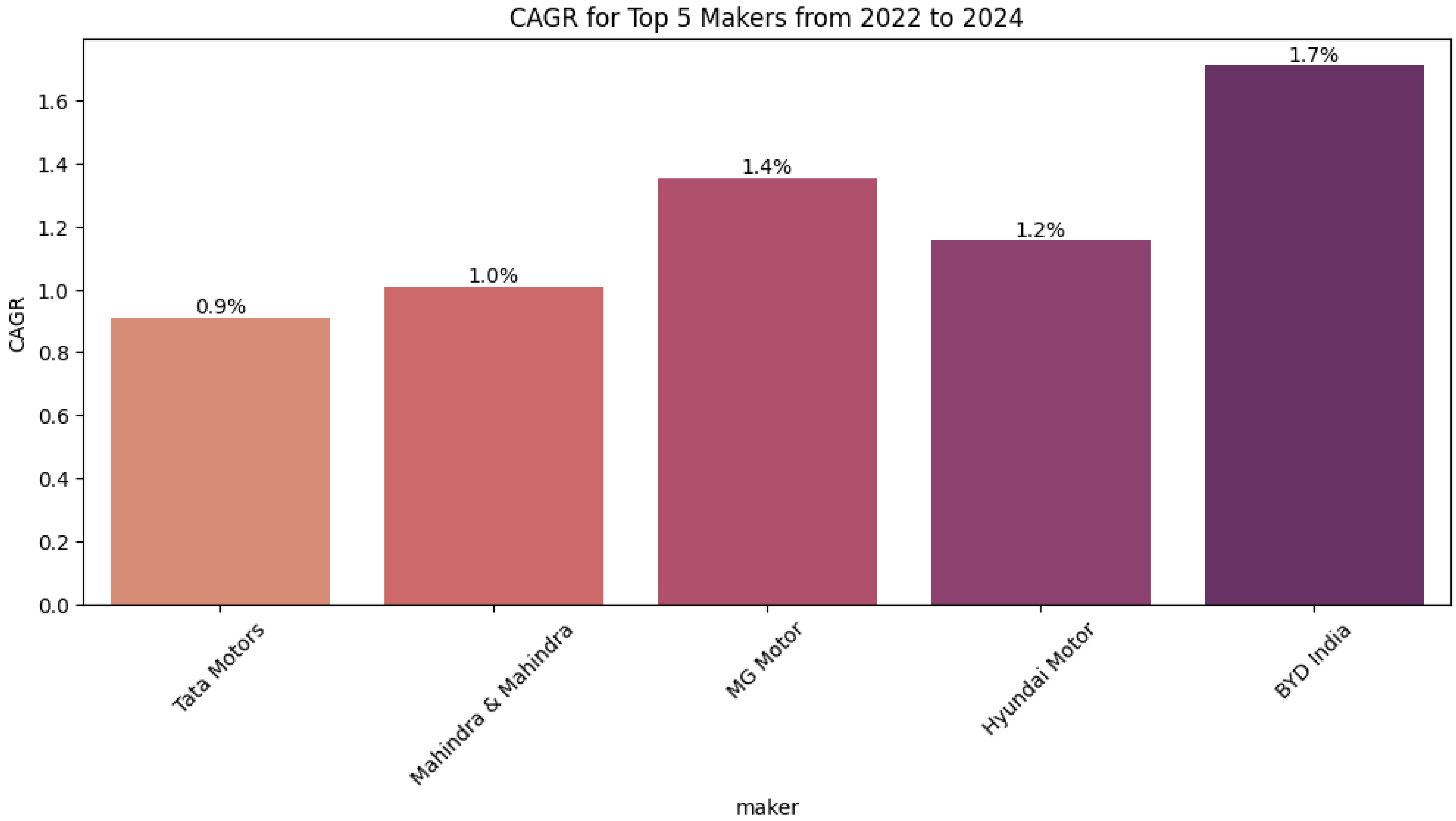
Out[146…

	maker	2022 EV Sold	Total EV Sold	CAGR
0	Tata Motors	24352	88935	0.911037
1	Mahindra & Mahindra	10215	41193	1.008133
2	MG Motor	2484	13753	1.353005
3	Hyundai Motor	447	2076	1.155063
4	BYD India	329	2419	1.711565

In [154…

```python
# Create a bar plot for CAGR by maker
plt.figure(figsize = (12,5))
ax = sns.barplot(x='maker', y='CAGR', data=dm, hue='maker', palette='flare')

# Set plot title and labels
plt.title('CAGR for Top 5 Makers from 2022 to 2024')
plt.xticks(rotation=45)

# Add data labels on top of bars
```

```
for bar in ax.patches:
    ax.annotate(f'{bar.get_height():.1f}%',
                (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                ha='center', va='bottom', fontsize=10)

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



List down the top 10 states that had the highest compounded annual growth rate (CAGR) from 2022 to 2024 in total vehicles sold.

In [188…

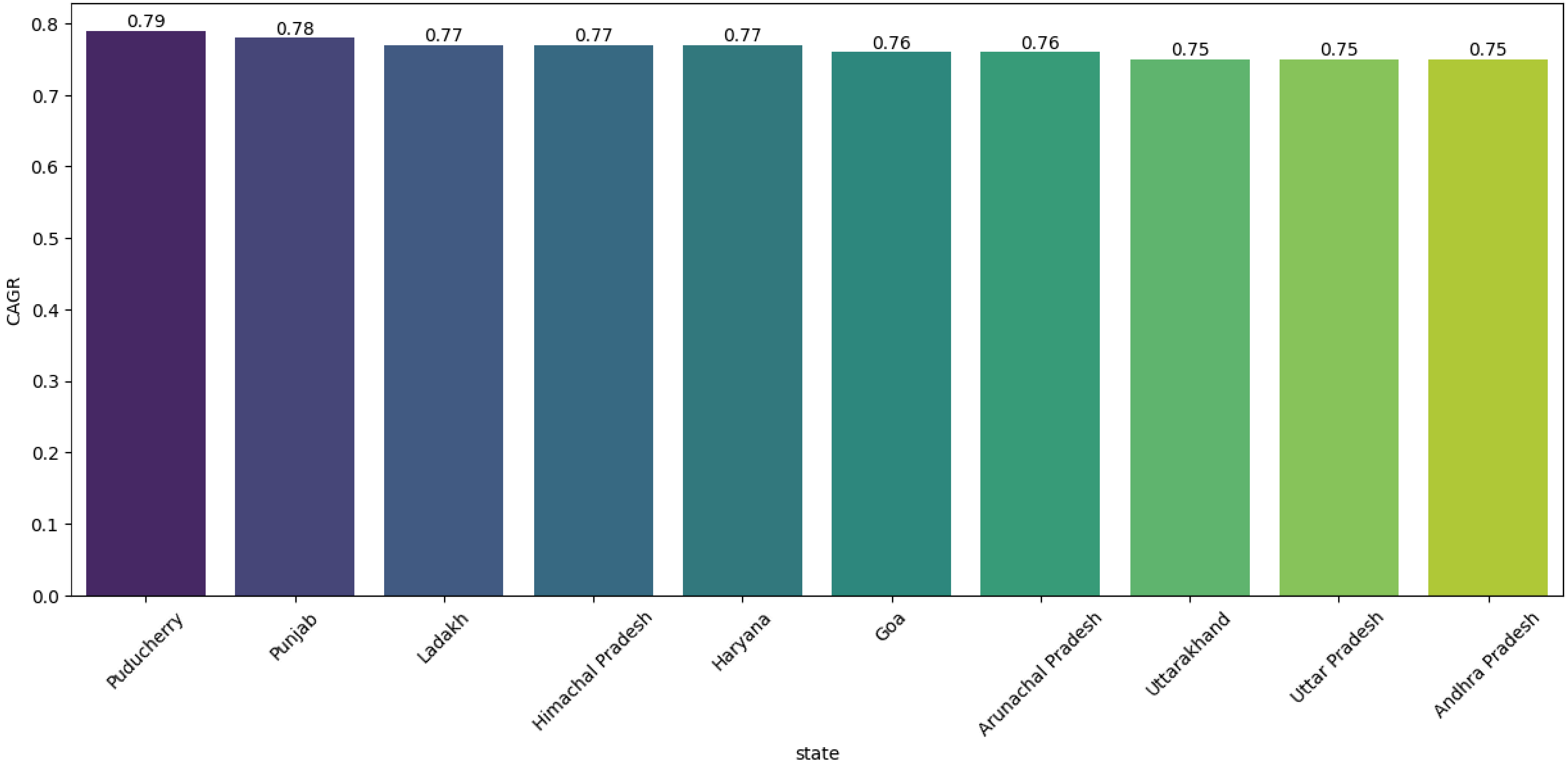
```python
# Filter for 4-wheelers sold in 2022 and group by state
state = df2[(df2['date'].dt.year == 2022) & (df2['vehicle_category'] == '4-Wheelers')]
groupstate = state.groupby('state').agg({'total_vehicles_sold': 'sum'})
sortstate = groupstate.sort_values('total_vehicles_sold', ascending=False)

# Filter for total 4-wheelers sold from 2021 onwards and group by state
totalstate = df2[(df2['date'].dt.year >= 2021) & (df2['vehicle_category'] == '4-Wheelers')]
totalgroupstate = totalstate.groupby('state').agg({'total_vehicles_sold': 'sum'})
totalsortstate = totalgroupstate.sort_values('total_vehicles_sold', ascending=False)

# Merge the results for 2022 and total sales
mergedstate = pd.merge(sortstate, totalsortstate, on='state')

# Rename columns and calculate CAGR
rename = mergedstate.rename(columns={
    'total_vehicles_sold_x': 'total_vehicles_sold_2022',
    'total_vehicles_sold_y': 'total_vehicles_sold'
})
rename['CAGR'] = (((rename['total_vehicles_sold'] / rename['total_vehicles_sold_2022']) ** (1/2)) - 1)
rename['CAGR'] = rename['CAGR'].round(2)

# Get top 10 states by CAGR
d = rename.sort_values('CAGR', ascending=False).head(10)
d
```

Out[188…

state	total_vehicles_sold_2022	total_vehicles_sold	CAGR
Puducherry	8956	28546	0.79
Punjab	120278	378958	0.78
Ladakh	2375	7418	0.77
Himachal Pradesh	43010	134672	0.77
Haryana	216160	674214	0.77
Goa	19278	59645	0.76
Arunachal Pradesh	10247	31892	0.76
Uttarakhand	48838	150265	0.75
Uttar Pradesh	358576	1098634	0.75
Andhra Pradesh	86208	265029	0.75

In [159…

```
# Set the figure size
plt.figure(figsize=(15, 6))

# Create a bar plot for CAGR by state
bar_plot = sns.barplot(x='state', y='CAGR', data=d, hue = 'state', palette='viridis')

# Add data labels on top of the bars
for index, value in enumerate(d['CAGR']):
    bar_plot.text(index, value, str(value), ha='center', va='bottom')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

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



What are the peak and low season months for EV sales based on the data from 2022 to 2024?

In [160…

```
df1.head()
```

Out[160…

	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

In [190…

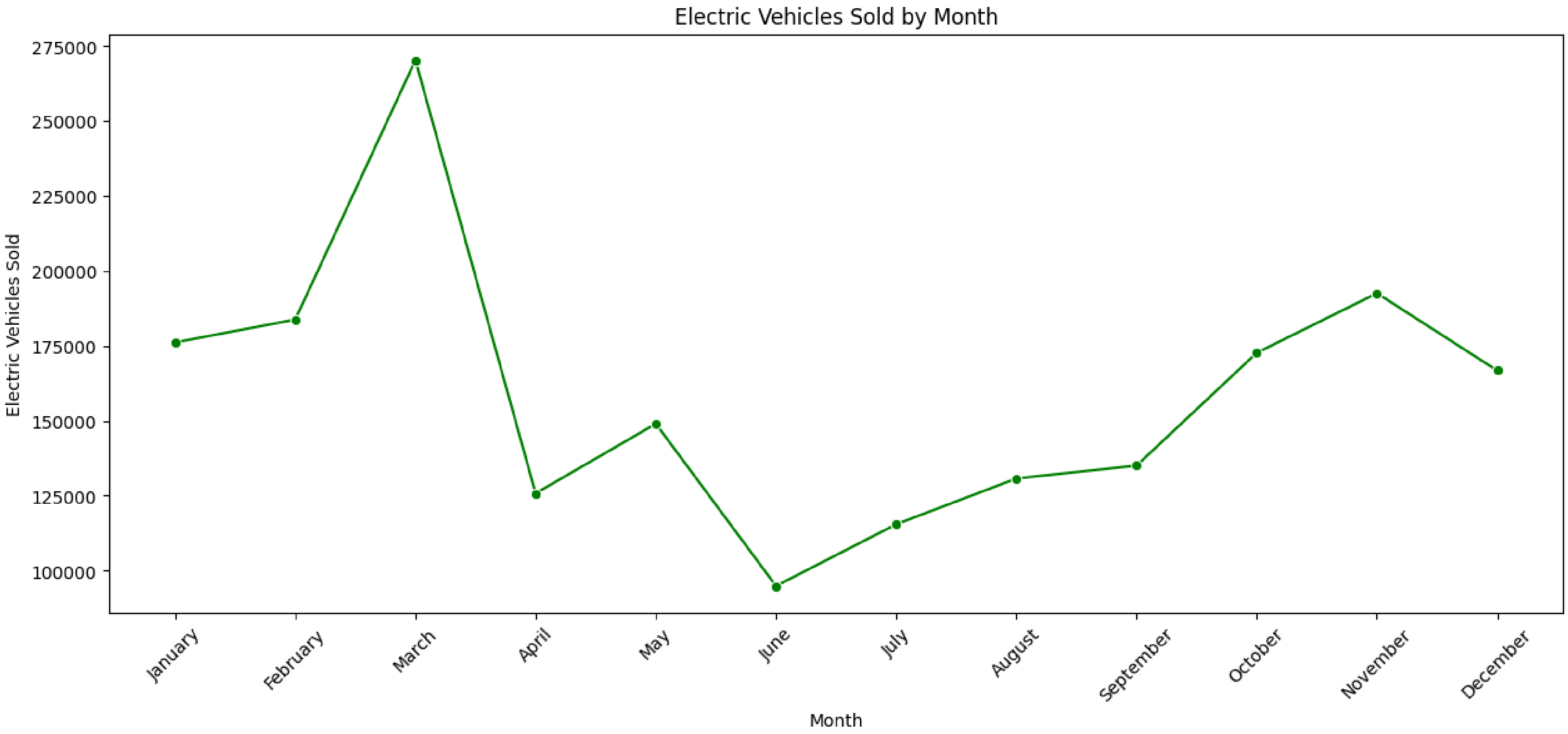
```
# Extract month name and month number
df1['month'] = df1['date'].dt.month_name()
df1['month_num'] = df1['date'].dt.month

# Filter for 2-wheelers and group by month
filter = df1[df1['vehicle_category'] == '2-Wheelers']
group1 = filter.groupby(['month', 'month_num']).agg({'electric_vehicles_sold': 'sum'})

# Sort the results by month number
sortq1 = group1.sort_values('month_num')
sortq1
```

Out[190…

		electric_vehicles_sold
month	**month_num**	
January	**1**	176143
February	**2**	183756
March	**3**	270309
April	**4**	125778
May	**5**	149013
June	**6**	94891
July	**7**	115472
August	**8**	130747
September	**9**	135060
October	**10**	172686
November	**11**	192552
December	**12**	166761

In [173…

```
# Set the figure size
plt.figure(figsize=(15, 6))

# Create a line plot for electric vehicles sold by month
sns.lineplot(x='month', y='electric_vehicles_sold', data=sortq1, color='g',marker = 'o')

# Set plot title and labels
plt.title('Electric Vehicles Sold by Month')
plt.xlabel('Month')
plt.ylabel('Electric Vehicles Sold')

# Display the plot
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```



In [176…

```
# Filter for 4-wheelers
dq2 = df1[df1['vehicle_category'] == '4-Wheelers']

# Group by month and month number, aggregating electric vehicles sold
group2 = dq2.groupby(['month', 'month_num']).agg({'electric_vehicles_sold': 'sum'})

# Sort the results by month number
sortq2 = group2.sort_values('month_num')
sortq2
```

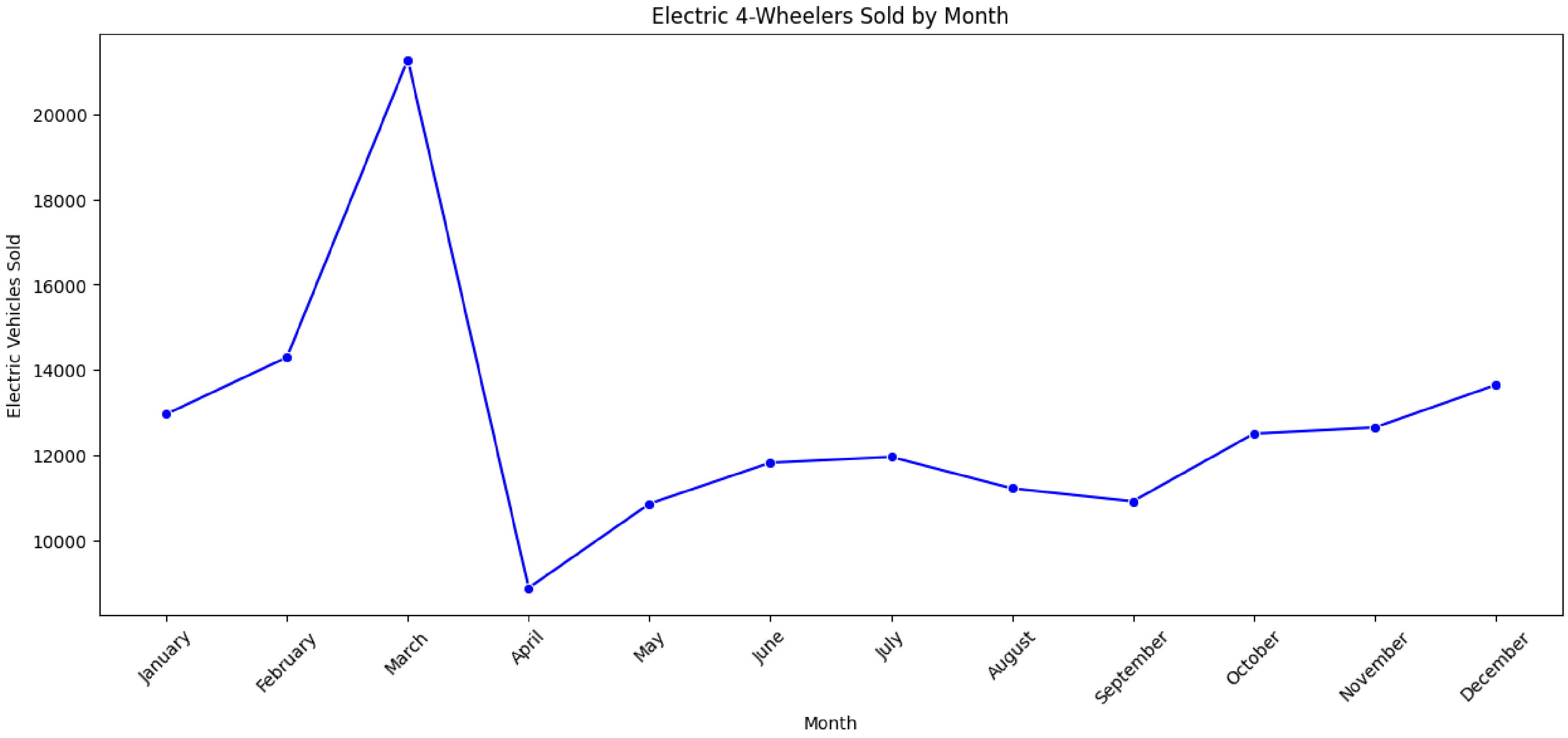
Out[176…

		electric_vehicles_sold
month	**month_num**	
January	**1**	12956
February	**2**	14293
March	**3**	21278
April	**4**	8879
May	**5**	10856
June	**6**	11818
July	**7**	11954
August	**8**	11214
September	**9**	10912
October	**10**	12499
November	**11**	12644
December	**12**	13640

In [178…

```python
# Set the figure size
plt.figure(figsize=(15, 6))

# Create a line plot for electric vehicles sold by month for 4-wheelers
sns.lineplot(x='month', y='electric_vehicles_sold', data=sortq2, color='b',marker = 'o')

# Set plot title and labels
plt.title('Electric 4-Wheelers Sold by Month')
plt.xlabel('Month')
plt.ylabel('Electric Vehicles Sold')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)
```

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



What is the projected number of EV sales (including 2-wheelers and 4wheelers) for the top 10 states by penetration rate in 2030, based on the compounded annual growth rate (CAGR) from previous years?

```
In [5]: df2.head()
```

Out[5]:

	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

In [184…

```python
# Filter data for 2023 and group by state
d23 = df2[df2['date'].dt.year == 2023]
grp_23 = d23.groupby('state').agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'})

# Filter data for 2024 and group by state
df_24 = df2[df2['date'].dt.year == 2024]
grp_24 = df_24.groupby('state').agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'})

# Filter data from 2021 onwards and group by state
df_all = df2[df2['date'].dt.year >= 2021]
grp_all = df_all.groupby('state').agg({'electric_vehicles_sold': 'sum', 'total_vehicles_sold': 'sum'})

# Merge grouped dataframes
merge_grp = pd.merge(pd.merge(grp_23, grp_all, on='state'), grp_24, on='state')

# Calculate CAGR for electric vehicles and total vehicles sold
merge_grp['CAGR'] = ((merge_grp['electric_vehicles_sold_y'] / merge_grp['electric_vehicles_sold_x']) - 1).round(2)
merge_grp['CAGR_V'] = ((merge_grp['total_vehicles_sold_y'] / merge_grp['total_vehicles_sold_x']) - 1).round(2)

# Predict future sales based on CAGR
merge_grp['predicted_EV'] = (merge_grp['electric_vehicles_sold'] * (merge_grp['CAGR'] + 1) ** 6).round()
merge_grp['predicted_v'] = (merge_grp['total_vehicles_sold'] * (merge_grp['CAGR_V'] + 1) ** 6).round()

# Calculate penetration rate
merge_grp['penetration_rate'] = (100 * merge_grp['predicted_EV'] / merge_grp['predicted_v']).round(2)

# Get the top 10 states by penetration rate
grp = merge_grp.sort_values(['penetration_rate'], ascending=False).head(10)
```

```python
# Select relevant columns for sales metrics
sales = grp[['CAGR', 'predicted_EV', 'penetration_rate']]
sales
```

Out[184…

state	CAGR	predicted_EV	penetration_rate
Ladakh	2.78	40839.0	9.16
Goa	1.12	351793.0	5.77
Haryana	2.05	3719127.0	4.28
Delhi	1.35	2236182.0	4.05
Karnataka	1.15	5061733.0	3.47
Rajasthan	1.35	3203111.0	3.15
Jammu and Kashmir	2.03	619852.0	3.05
Maharashtra	1.18	6635735.0	2.81
Andhra Pradesh	1.58	3727319.0	2.53
Tamil Nadu	1.37	5590468.0	2.34

SECONDARY QUESTIONS

WHAT ARE THE PRIMARY REASONS FOR CUSTOMERS CHOOSING 4-WHEELER EVS IN 2023 AND 2024 (COST SAVINGS, ENVIRONMENTAL CONCERNS, GOVERNMENT INCENTIVES)?

Cost Savings:

• Lower Operating Costs: EVs are appealing due to their lower operating costs compared to traditional internal combustion engine (ICE) vehicles. The savings come from cheaper electricity versus fuel costs and reduced maintenance requirements, as EVs have fewer moving parts.

• Government Incentives: The Indian government’s initiatives, particularly the FAME II subsidy, provide up to ₹1.5 lakhs in incentives, significantly lowering the initial purchase cost of EVs. Additionally, states like Maharashtra, Delhi, and Gujarat offer extra benefits such as tax exemptions and reduced registration fees, making EVs more affordable.

Environmental Concerns:

• Reduction in Carbon Emissions: With increasing awareness of climate change, consumers are more conscious of their environmental impact. EVsproduce zero tailpipe emissions, which helps in reducing air pollution, especially in urban areas where air quality is a major concern.

• Government Support for Clean Energy: The Indian government’s commitment to reducing greenhouse gas emissions aligns with global efforts to combat climate change, encouraging consumers to switch to cleaner energy alternatives like EVs.

Technological Advancements and Infrastructure:

• Improved Range and Charging Infrastructure: The range of entry-level EVs has improved, now offering around 200-300 km per charge, sufficient for most daily commutes. While the number of public charging stations in India is growing, with around 1,500 stations as of August 2024, the majority are concentrated in urban areas. The expansion of charging infrastructure is reducing range anxiety, making EVs a more practical choice.

• Innovation and Competition: The EV market is becoming increasingly competitive, with established players like Tata Motors, MG Motors, and Mahindra leading innovation. This competition drives advancements in battery technology and vehicle performance, making EVs more attractive.

Corporate Social Responsibility (CSR):

• Corporate Adoption of EVs: Many companies in India are integrating EVs into their CSR strategies, focusing on sustainability. This trend not only supports environmental goals but also influences consumer preferences, as more people choose EVs for both personal and environmental reasons.

State-Specific Benefits:

• Regional Incentives: States like Maharashtra, Delhi, and Gujarat offer additional financial benefits, such as registration fee waivers and road tax exemptions.These state-specific incentives make EVs even more financially viable for consumers in these regions.

HOW DO GOVERNMENT INCENTIVES AND SUBSIDIES IMPACT THE ADOPTION RATES OF 2 WHEELERS AND 4-WHEELERS? WHICH STATES IN INDIA PROVIDED THE MOST SUBSIDIES?

Government Incentives and Impact on EV Adoption:

Key Schemes:

NEMMP (National Electric Mobility Mission Plan): Promotes EV adoption through financial incentives and infrastructure support.

FAME India Scheme: Provides subsidies for electric vehicles to reduce their cost and encourage adoption.

EMPS (Electric Mobility Promotion Scheme): Offers additional subsidies and supports EV infrastructure, including charging stations.

Recent Updates:

Extended Subsidies: Financial support for electric two-wheelers and three-wheelers has been extended until September 2023.

GST Reductions: GST on electric vehicles reduced from 12% to 5%, and on EV charging stations from 18% to 5%.

Top Subsidy State:

Maharashtra: Provides the highest subsidies for electric vehicles, reflecting strong state-level support for EV adoption

WHO SHOULD BE THE BRAND AMBASSADOR IF ATLIQ MOTORS LAUNCHES THEIR EV/HYBRID VEHICLES IN INDIA AND WHY?

Mahendra Singh Dhoni

- Wide Appeal and Popularity: Dhoni, known as "Captain Cool," is one of India's most beloved sports figures. His leadership, calm demeanor, and cricketing success make him a highly recognizable and trusted figure, which can greatly enhance brand recall and trust for AtliQ Motors.

- Positive Image: Dhoni’s disciplined and responsible persona aligns with the values of sustainability and innovation central to the EV/Hybrid vehicle market.

- Influence on Youth: With a significant following among the younger demographic, Dhoni can effectively drive awareness and adoption of EV/Hybrid vehicles among those who are increasingly interested in sustainable solutions.

- High Credibility: Dhoni’s credibility and respect lend authority to AtliQ Motors’ EV/Hybrid vehicles, reinforcing the brand’s commitment to quality and sustainability.

- Alignment with Green Initiatives: His involvement in charitable and environmental causes mirrors the eco-friendly nature of EV/Hybrid vehicles, enhancing the brand’s image as a promoter of positive change.

INSIGHTS

WHICH STATE OF INDIA IS IDEAL TO START THE MANUFACTURING UNIT? (BASED ON SUBSIDIES PROVIDED, EASE OF DOING BUSINESS, STABILITY IN GOVERNANCE, ETC.)

Gujarat:

- Significant Subsidies: Offers substantial subsidies for two-wheelers, three-wheelers, and four-wheelers.

- Business-Friendly Environment: Known for its ease of doing business and supportive industrial policies.

- Strong Infrastructure: Well-developed infrastructure and commitment to expanding EV infrastructure

Maharashtra:

• Substantial Incentives: Provides significant incentives for various EV categories and waives registration fees and road taxes.

• Robust Industrial Base: Has a strong industrial infrastructure and a high level of EV adoption and conversion targets.

• Government Support: Active government initiatives and subsidies support EV growth and manufacturing

RECOMMENDATIONS

YOUR TOP 3 RECOMMENDATIONS FOR ATLIQ MOTORS:

1. Choose Gujarat for manufacturing: Establish the manufacturing unit in Gujarat due to its significant subsidies, business-friendly environment, and well-developed infrastructure. Additionally, target high-growth states like Maharashtra, Karnataka, and Gujarat, which have strong EV adoption rates and government support.

2. Expand charging infrastructure: Collaborate with local entities to expand public charging stations, starting in highrevenue cities. Focus on building a comprehensive EV ecosystem, including a network of charging stations and integrated services, to enhance the overall customer experience.

3. Invest in affordable two-wheelers: Concentrate on producing affordable two-wheeler EVs with enhanced features, as this segment is highly popular. Consider acquiring small EV manufacturing companies to reduce infrastructure costs and facilitate market entry, while leveraging available government incentives and support

THANK YOU

Presenter By Rana Basak