In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv(r"E:\Download folder\data.csv")

# Display basic information about the dataset
print(data.info())
print(data.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       103 non-null    int64  
 1   Brand            103 non-null    object 
 2   Model            103 non-null    object 
 3   AccelSec         103 non-null    float64
 4   TopSpeed_KmH     103 non-null    int64  
 5   Range_Km         103 non-null    int64  
 6   Efficiency_WhKm  103 non-null    int64  
 7   FastCharge_KmH   103 non-null    int64  
 8   RapidCharge      103 non-null    object 
 9   PowerTrain       103 non-null    object 
 10  PlugType         103 non-null    object 
 11  BodyStyle        103 non-null    object 
 12  Segment          103 non-null    object 
 13  Seats            103 non-null    int64  
 14  PriceEuro        103 non-null    int64  
dtypes: float64(1), int64(7), object(7)
memory usage: 12.2+ KB
None
       Unnamed: 0    AccelSec  TopSpeed_KmH    Range_Km  Efficiency_WhKm  \
count  103.000000  103.000000    103.000000  103.000000       103.000000   
mean    51.000000    7.396117    179.194175  338.786408       189.165049   
std     29.877528    3.017430     43.573030  126.014444        29.566839   
min      0.000000    2.100000    123.000000   95.000000       104.000000   
25%     25.500000    5.100000    150.000000  250.000000       168.000000   
50%     51.000000    7.300000    160.000000  340.000000       180.000000   
75%     76.500000    9.000000    200.000000  400.000000       203.000000   
max    102.000000   22.400000    410.000000  970.000000       273.000000   

       FastCharge_KmH       Seats      PriceEuro  
count      103.000000  103.000000     103.000000  
mean       444.271845    4.883495   55811.563107  
std        203.949253    0.795834   34134.665280  
min        170.000000    2.000000   20129.000000  
25%        260.000000    5.000000   34429.500000  
50%        440.000000    5.000000   45000.000000  
75%        555.000000    5.000000   65000.000000  
max        940.000000    7.000000  215000.000000  
```

In [4]:
```
pip install geopandas
```

```
Collecting geopandas
  Downloading geopandas-0.13.2-py3-none-any.whl (1.1 MB)
                                              0.0/1.1 MB ? eta -:--:--
     --------------                           0.4/1.1 MB 8.3 MB/s eta 0:00:01
     -------------------------                0.7/1.1 MB 8.8 MB/s eta 0:00:01
     ---------------------------------------  1.1/1.1 MB 9.8 MB/s eta 0:00:01
     ---------------------------------------- 1.1/1.1 MB 6.9 MB/s eta 0:00:00
Collecting fiona>=1.8.19 (from geopandas)
  Downloading Fiona-1.9.4.post1-cp311-cp311-win_amd64.whl (22.7 MB)
                                              0.0/22.7 MB ? eta -:--:--
     --                                       1.3/22.7 MB 39.0 MB/s eta 0:00:01
     --                                       1.3/22.7 MB 39.0 MB/s eta 0:00:01
     ---                                      1.8/22.7 MB 14.5 MB/s eta 0:00:02
     ---                                      2.2/22.7 MB 12.7 MB/s eta 0:00:02
     ----                                     2.5/22.7 MB 12.1 MB/s eta 0:00:02
     ----                                     2.8/22.7 MB 11.8 MB/s eta 0:00:02
     -----                                    3.2/22.7 MB 9.8 MB/s eta 0:00:02
     ------                                   3.5/22.7 MB 8.7 MB/s eta 0:00:03
     ------                                   3.9/22.7 MB 9.0 MB/s eta 0:00:03
     -------                                  4.0/22.7 MB 8.9 MB/s eta 0:00:03
     -------                                  4.2/22.7 MB 7.9 MB/s eta 0:00:03
     --------                                 4.7/22.7 MB 8.3 MB/s eta 0:00:03
     --------                                 5.0/22.7 MB 7.9 MB/s eta 0:00:03
     ---------                                5.1/22.7 MB 7.6 MB/s eta 0:00:03
     ---------                                5.6/22.7 MB 7.8 MB/s eta 0:00:03
     ----------                               5.7/22.7 MB 7.6 MB/s eta 0:00:03
     ----------                               6.1/22.7 MB 7.6 MB/s eta 0:00:03
     ----------                               6.2/22.7 MB 7.5 MB/s eta 0:00:03
     -----------                              6.5/22.7 MB 7.3 MB/s eta 0:00:03
     -----------                              6.6/22.7 MB 7.2 MB/s eta 0:00:03
     -----------                              6.7/22.7 MB 6.9 MB/s eta 0:00:03
     ------------                             6.8/22.7 MB 6.7 MB/s eta 0:00:03
     ------------                             6.9/22.7 MB 6.4 MB/s eta 0:00:03
     ------------                             6.9/22.7 MB 6.3 MB/s eta 0:00:03
     ------------                             7.0/22.7 MB 6.1 MB/s eta 0:00:03
     ------------                             7.1/22.7 MB 5.9 MB/s eta 0:00:03
     ------------                             7.1/22.7 MB 5.8 MB/s eta 0:00:03
     ------------                             7.2/22.7 MB 5.6 MB/s eta 0:00:03
     ------------                             7.2/22.7 MB 5.5 MB/s eta 0:00:03
     ------------                             7.3/22.7 MB 5.4 MB/s eta 0:00:03
     -------------                            7.4/22.7 MB 5.2 MB/s eta 0:00:03
     -------------                            7.4/22.7 MB 5.2 MB/s eta 0:00:03
     -------------                            7.5/22.7 MB 5.0 MB/s eta 0:00:04
     -------------                            7.6/22.7 MB 5.0 MB/s eta 0:00:04
     -------------                            7.6/22.7 MB 4.9 MB/s eta 0:00:04
     -------------                            7.7/22.7 MB 4.8 MB/s eta 0:00:04
     -------------                            7.8/22.7 MB 4.7 MB/s eta 0:00:04
     -------------                            7.9/22.7 MB 4.7 MB/s eta 0:00:04
     --------------                           8.0/22.7 MB 4.6 MB/s eta 0:00:04
     --------------                           8.0/22.7 MB 4.5 MB/s eta 0:00:04
     --------------                           8.1/22.7 MB 4.5 MB/s eta 0:00:04
     --------------                           8.2/22.7 MB 4.5 MB/s eta 0:00:04
     --------------                           8.3/22.7 MB 4.4 MB/s eta 0:00:04
     --------------                           8.4/22.7 MB 4.3 MB/s eta 0:00:04
     --------------                           8.5/22.7 MB 4.3 MB/s eta 0:00:04
     ---------------                          8.6/22.7 MB 4.3 MB/s eta 0:00:04
     ---------------                          8.7/22.7 MB 4.2 MB/s eta 0:00:04
     ---------------                          8.8/22.7 MB 4.2 MB/s eta 0:00:04
     ---------------                          8.9/22.7 MB 4.2 MB/s eta 0:00:04
     ---------------                          9.0/22.7 MB 4.1 MB/s eta 0:00:04
```

```
---------------                      9.2/22.7 MB 4.1 MB/s eta 0:00:04
---------------                      9.2/22.7 MB 4.1 MB/s eta 0:00:04
---------------                      9.4/22.7 MB 4.1 MB/s eta 0:00:04
---------------                      9.5/22.7 MB 4.0 MB/s eta 0:00:04
---------------                      9.6/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     9.7/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     9.9/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     10.0/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     10.1/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     10.3/22.7 MB 4.0 MB/s eta 0:00:04
----------------                     10.4/22.7 MB 3.9 MB/s eta 0:00:04
----------------                     10.5/22.7 MB 3.9 MB/s eta 0:00:04
----------------                     10.7/22.7 MB 3.8 MB/s eta 0:00:04
----------------                     10.8/22.7 MB 3.7 MB/s eta 0:00:04
----------------                     11.0/22.7 MB 3.7 MB/s eta 0:00:04
----------------                     11.1/22.7 MB 3.6 MB/s eta 0:00:04
----------------                     11.3/22.7 MB 3.6 MB/s eta 0:00:04
----------------                     11.4/22.7 MB 3.6 MB/s eta 0:00:04
----------------                     11.5/22.7 MB 3.6 MB/s eta 0:00:04
----------------                     11.7/22.7 MB 3.5 MB/s eta 0:00:04
----------------                     11.8/22.7 MB 3.5 MB/s eta 0:00:04
----------------                     12.0/22.7 MB 3.5 MB/s eta 0:00:04
----------------                     12.2/22.7 MB 3.5 MB/s eta 0:00:04
----------------                     12.4/22.7 MB 3.4 MB/s eta 0:00:04
----------------                     12.5/22.7 MB 3.4 MB/s eta 0:00:03
----------------                     12.7/22.7 MB 3.4 MB/s eta 0:00:03
----------------                     12.9/22.7 MB 3.4 MB/s eta 0:00:03
----------------                     13.1/22.7 MB 3.4 MB/s eta 0:00:03
----------------                     13.2/22.7 MB 3.4 MB/s eta 0:00:03
----------------                     13.4/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     13.6/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     13.8/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     14.0/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     14.2/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     14.4/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     14.6/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     14.8/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     15.0/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     15.2/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     15.4/22.7 MB 3.3 MB/s eta 0:00:03
----------------                     15.7/22.7 MB 3.2 MB/s eta 0:00:03
----------------                     15.7/22.7 MB 3.2 MB/s eta 0:00:03
----------------                     16.1/22.7 MB 3.2 MB/s eta 0:00:03
----------------                     16.3/22.7 MB 3.2 MB/s eta 0:00:02
----------------                     16.5/22.7 MB 3.2 MB/s eta 0:00:02
----------------                     16.8/22.7 MB 3.3 MB/s eta 0:00:02
----------------                     17.0/22.7 MB 3.3 MB/s eta 0:00:02
----------------                     17.2/22.7 MB 3.5 MB/s eta 0:00:02
----------------                     17.5/22.7 MB 3.6 MB/s eta 0:00:02
----------------                     17.7/22.7 MB 3.8 MB/s eta 0:00:02
----------------                     17.8/22.7 MB 3.8 MB/s eta 0:00:02
----------------                     18.2/22.7 MB 4.0 MB/s eta 0:00:02
----------------                     18.4/22.7 MB 4.1 MB/s eta 0:00:02
----------------                     18.7/22.7 MB 4.2 MB/s eta 0:00:01
----------------                     18.9/22.7 MB 4.3 MB/s eta 0:00:01
----------------                     19.2/22.7 MB 4.4 MB/s eta 0:00:01
----------------                     19.5/22.7 MB 4.5 MB/s eta 0:00:01
----------------                     19.7/22.7 MB 4.7 MB/s eta 0:00:01
----------------                     20.0/22.7 MB 4.7 MB/s eta 0:00:01
----------------                     20.3/22.7 MB 4.9 MB/s eta 0:00:01
```

```
     ------------------------------------     20.5/22.7 MB 4.9 MB/s eta 0:00:01
     ------------------------------------     20.8/22.7 MB 5.1 MB/s eta 0:00:01
     -------------------------------------    21.1/22.7 MB 5.2 MB/s eta 0:00:01
     -------------------------------------    21.3/22.7 MB 5.2 MB/s eta 0:00:01
     --------------------------------------   21.6/22.7 MB 5.3 MB/s eta 0:00:01
     --------------------------------------   21.9/22.7 MB 5.5 MB/s eta 0:00:01
     ---------------------------------------  22.2/22.7 MB 5.5 MB/s eta 0:00:01
     ---------------------------------------  22.4/22.7 MB 5.6 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------  22.7/22.7 MB 5.7 MB/s eta 0:00:01
     ---------------------------------------- 22.7/22.7 MB 4.5 MB/s eta 0:00:00
Requirement already satisfied: packaging in e:\download folder\anaconda1\lib\site-pac
kages (from geopandas) (23.0)
Requirement already satisfied: pandas>=1.1.0 in e:\download folder\anaconda1\lib\site
-packages (from geopandas) (1.5.3)
Collecting pyproj>=3.0.1 (from geopandas)
  Downloading pyproj-3.6.0-cp311-cp311-win_amd64.whl (5.7 MB)
                                              0.0/5.7 MB ? eta -:--:--
     --                                       0.4/5.7 MB 11.2 MB/s eta 0:00:01
     ----                                     0.7/5.7 MB 8.3 MB/s eta 0:00:01
     -------                                  1.1/5.7 MB 7.6 MB/s eta 0:00:01
     ---------                                1.4/5.7 MB 8.0 MB/s eta 0:00:01
     -----------                              1.7/5.7 MB 7.8 MB/s eta 0:00:01
     --------------                           2.0/5.7 MB 7.6 MB/s eta 0:00:01
     ----------------                         2.3/5.7 MB 7.8 MB/s eta 0:00:01
     ------------------                       2.6/5.7 MB 7.6 MB/s eta 0:00:01
     --------------------                     2.9/5.7 MB 7.7 MB/s eta 0:00:01
     ----------------------                   3.2/5.7 MB 7.6 MB/s eta 0:00:01
     ------------------------                 3.5/5.7 MB 7.5 MB/s eta 0:00:01
     --------------------------               3.9/5.7 MB 7.7 MB/s eta 0:00:01
     ----------------------------             4.1/5.7 MB 7.5 MB/s eta 0:00:01
     -------------------------------          4.5/5.7 MB 7.7 MB/s eta 0:00:01
     ---------------------------------        4.8/5.7 MB 7.6 MB/s eta 0:00:01
     -----------------------------------      5.1/5.7 MB 7.6 MB/s eta 0:00:01
     ------------------------------------     5.2/5.7 MB 7.6 MB/s eta 0:00:01
     ---------------------------------------  5.6/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
     ---------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
```

```
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    --------------------------------------  5.7/5.7 MB 7.5 MB/s eta 0:00:01
    ---------------------------------------- 5.7/5.7 MB 2.7 MB/s eta 0:00:00
Collecting shapely>=1.7.1 (from geopandas)
  Downloading shapely-2.0.1-cp311-cp311-win_amd64.whl (1.4 MB)
                                          0.0/1.4 MB ? eta -:--:--
    -----------                           0.4/1.4 MB 12.2 MB/s eta 0:00:01
    -----------------                     0.6/1.4 MB 9.7 MB/s eta 0:00:01
    ------------------------              0.9/1.4 MB 7.9 MB/s eta 0:00:01
    --------------------------------      1.1/1.4 MB 8.0 MB/s eta 0:00:01
    --------------------------------------  1.4/1.4 MB 7.2 MB/s eta 0:00:01
    --------------------------------------  1.4/1.4 MB 7.2 MB/s eta 0:00:01
    ---------------------------------------- 1.4/1.4 MB 5.1 MB/s eta 0:00:00
Requirement already satisfied: attrs>=19.2.0 in e:\download folder\anaconda1\lib\site
-packages (from fiona>=1.8.19->geopandas) (22.1.0)
Requirement already satisfied: certifi in e:\download folder\anaconda1\lib\site-packa
ges (from fiona>=1.8.19->geopandas) (2023.5.7)
Requirement already satisfied: click~=8.0 in e:\download folder\anaconda1\lib\site-pa
ckages (from fiona>=1.8.19->geopandas) (8.0.4)
Collecting click-plugins>=1.0 (from fiona>=1.8.19->geopandas)
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Collecting cligj>=0.5 (from fiona>=1.8.19->geopandas)
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: six in e:\download folder\anaconda1\lib\site-packages
(from fiona>=1.8.19->geopandas) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in e:\download folder\anaconda1
\lib\site-packages (from pandas>=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in e:\download folder\anaconda1\lib\site-
packages (from pandas>=1.1.0->geopandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in e:\download folder\anaconda1\lib\site
-packages (from pandas>=1.1.0->geopandas) (1.24.3)
Requirement already satisfied: colorama in e:\download folder\anaconda1\lib\site-pack
ages (from click~=8.0->fiona>=1.8.19->geopandas) (0.4.6)
Installing collected packages: shapely, pyproj, cligj, click-plugins, fiona, geopanda
s
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.9.4.post1 geopandas-0.
13.2 pyproj-3.6.0 shapely-2.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
pip install pyshp
```

```
Collecting pyshp
  Downloading pyshp-2.3.1-py2.py3-none-any.whl (46 kB)
                                              0.0/46.5 kB ? eta -:--:--
     ---------------------------------        41.0/46.5 kB ? eta -:--:--
     ---------------------------------------- 46.5/46.5 kB 576.1 kB/s eta 0:00:00
Installing collected packages: pyshp
Successfully installed pyshp-2.3.1
Note: you may need to restart the kernel to use updated packages.
```

In [7]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import shapefile as shp
import plotly.graph_objects as go
from shapely.geometry import Point
from sklearn.cluster import KMeans
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler

sns.set_style('whitegrid')
```

In [11]:
```
import pandas as pd

# Load the dataset
data_path = r"E:\Download folder\2_ev_charging_station_dataset.xlsx"  # Replace with t
electric_vehicles_data = pd.read_excel(data_path)

# Display basic information about the dataset
print(electric_vehicles_data.info())

# Display the first few rows of the dataset
print(electric_vehicles_data.head())

# Display summary statistics of the dataset
print(electric_vehicles_data.describe())

# Display unique values in a specific column
print(electric_vehicles_data['The number of electric vehicles currently being used on

# Display the number of occurrences of each state in the dataset
print(electric_vehicles_data['The number of electric vehicles currently being used on
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 5 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   The number of electric vehicles currently being used on the roads of India, Stat
e wise as on

14-07-2022  34 non-null     object
 1   Unnamed: 1
34 non-null     object
 2   Unnamed: 2
34 non-null     object
 3   Unnamed: 3
34 non-null     object
 4   Unnamed: 4
34 non-null     object
dtypes: object(5)
memory usage: 1.5+ KB
None
  The number of electric vehicles currently being used on the roads of India, State w
ise as on\n\n14-07-2022  \
0                                                Sr. No.
1                                                      1
2                                                      2
3                                                      3
4                                                      4

                Unnamed: 1              Unnamed: 2  \
0                State Name  Total Electric Vehicle
1  Andaman & Nicobar Island                     162
2         Arunachal Pradesh                      20
3                     Assam                   64766
4                     Bihar                   83335

                   Unnamed: 3 Unnamed: 4
0  Total Non-Electric Vehicle      Total
1                      146945     147107
2                      252965     252985
3                     4677053    4741819
4                    10407078   10490413
       The number of electric vehicles currently being used on the roads of India, St
ate wise as on\n\n14-07-2022  \
count                                                 34
unique                                                34
top                                              Sr. No.
freq                                                   1

        Unnamed: 1  Unnamed: 2                  Unnamed: 3 Unnamed: 4
count           34          34                          34         34
unique          34          33                          34         34
top     State Name          21  Total Non-Electric Vehicle      Total
freq             1           2                           1          1
['Sr. No.' '1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12' '13' '14'
 '15' '16' '17' '18' '19' '20' '21' '22' '23' '24' '25' '26' '27' '28'
 '29' '30' '31' '32' 'Grand Total']
Sr. No.        1
25             1
```

```
19             1
20             1
21             1
22             1
23             1
24             1
26             1
1              1
27             1
28             1
29             1
30             1
31             1
32             1
18             1
17             1
16             1
15             1
14             1
13             1
12             1
11             1
10             1
9              1
8              1
7              1
6              1
5              1
4              1
3              1
2              1
Grand Total    1
Name: The number of electric vehicles currently being used on the roads of India, Sta
te wise as on\n\n14-07-2022, dtype: int64
```

In [13]:
```
# Check the column names in the charging facility dataset
print(charging_facility_data.columns)

# Check the column names in the electric vehicles dataset
print(electric_vehicles_data.columns)
```

```
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')
```

In [22]:
```
# Load the EV Charging Facility dataset
charging_facility_data = pd.read_excel(r"E:\Download folder\2_ev_charging_station_data

# Load the Electric Vehicles dataset in CSV format
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")
```

In [23]:
```
# Display the first few rows of the charging facility dataset
print(charging_facility_data.head())

# Display summary information about the charging facility dataset
```

```
print(charging_facility_data.info())

# Display summary statistics of the charging facility dataset
print(charging_facility_data.describe())
```

```
  The number of electric vehicles currently being used on the roads of India, State w
ise as on\n\n14-07-2022  \
0                                                Sr. No.
1                                                      1
2                                                      2
3                                                      3
4                                                      4

                 Unnamed: 1              Unnamed: 2  \
0                 State Name  Total Electric Vehicle
1  Andaman & Nicobar Island                     162
2         Arunachal Pradesh                      20
3                     Assam                   64766
4                     Bihar                   83335

                   Unnamed: 3 Unnamed: 4
0  Total Non-Electric Vehicle      Total
1                      146945     147107
2                      252965     252985
3                     4677053    4741819
4                    10407078   10490413
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 5 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   The number of electric vehicles currently being used on the roads of India, Stat
e wise as on

14-07-2022  34 non-null     object
 1   Unnamed: 1
34 non-null     object
 2   Unnamed: 2
34 non-null     object
 3   Unnamed: 3
34 non-null     object
 4   Unnamed: 4
34 non-null     object
dtypes: object(5)
memory usage: 1.5+ KB
None
       The number of electric vehicles currently being used on the roads of India, St
ate wise as on\n\n14-07-2022  \
count                                                 34
unique                                                34
top                                              Sr. No.
freq                                                   1

        Unnamed: 1  Unnamed: 2                 Unnamed: 3 Unnamed: 4
count           34          34                         34         34
unique          34          33                         34         34
top     State Name          21  Total Non-Electric Vehicle      Total
freq             1           2                          1          1
```

In [24]:
```
# Display the first few rows of the electric vehicles dataset
print(electric_vehicles_data.head())

# Display summary information about the electric vehicles dataset
print(electric_vehicles_data.info())

# Display summary statistics of the electric vehicles dataset
print(electric_vehicles_data.describe())
```

```
   Unnamed: 0        Brand                          Model  AccelSec  \
0           0      Tesla   Model 3 Long Range Dual Motor       4.6
1           1  Volkswagen                         ID.3 Pure      10.0
2           2    Polestar                              2       4.7
3           3         BMW                            iX3        6.8
4           4       Honda                              e        9.5

   TopSpeed_KmH  Range_Km  Efficiency_WhKm  FastCharge_KmH RapidCharge  \
0           233       450              161             940         Yes
1           160       270              167             250          No
2           210       400              181             620         Yes
3           180       360              206             560         Yes
4           145       170              168             190         Yes

  PowerTrain    PlugType  BodyStyle Segment  Seats  PriceEuro
0        AWD  Type 2 CCS      Sedan       D      5      55480
1        RWD  Type 2 CCS  Hatchback       C      5      30000
2        AWD  Type 2 CCS   Liftback       D      5      56440
3        RWD  Type 2 CCS        SUV       D      5      68040
4        RWD  Type 2 CCS  Hatchback       B      4      32997
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       103 non-null    int64
 1   Brand            103 non-null    object
 2   Model            103 non-null    object
 3   AccelSec         103 non-null    float64
 4   TopSpeed_KmH     103 non-null    int64
 5   Range_Km         103 non-null    int64
 6   Efficiency_WhKm  103 non-null    int64
 7   FastCharge_KmH   103 non-null    int64
 8   RapidCharge      103 non-null    object
 9   PowerTrain       103 non-null    object
 10  PlugType         103 non-null    object
 11  BodyStyle        103 non-null    object
 12  Segment          103 non-null    object
 13  Seats            103 non-null    int64
 14  PriceEuro        103 non-null    int64
dtypes: float64(1), int64(7), object(7)
memory usage: 12.2+ KB
None
       Unnamed: 0    AccelSec  TopSpeed_KmH    Range_Km  Efficiency_WhKm  \
count  103.000000  103.000000    103.000000  103.000000       103.000000
mean    51.000000    7.396117    179.194175  338.786408       189.165049
std     29.877528    3.017430     43.573030  126.014444        29.566839
min      0.000000    2.100000    123.000000   95.000000       104.000000
25%     25.500000    5.100000    150.000000  250.000000       168.000000
50%     51.000000    7.300000    160.000000  340.000000       180.000000
75%     76.500000    9.000000    200.000000  400.000000       203.000000
max    102.000000   22.400000    410.000000  970.000000       273.000000

       FastCharge_KmH       Seats      PriceEuro
count      103.000000  103.000000     103.000000
mean       444.271845    4.883495   55811.563107
std        203.949253    0.795834   34134.665280
min        170.000000    2.000000   20129.000000
25%        260.000000    5.000000   34429.500000
50%        440.000000    5.000000   45000.000000
```

```
75%        555.000000    5.000000   65000.000000
max        940.000000    7.000000  215000.000000
```

In [26]:
```
print(charging_facility_data.columns)
```

```
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')
```

In [27]:
```
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv", skiprows=1)
```

In [28]:
```
column_names = ['Sr. No.', 'State Name', 'Total Electric Vehicle', 'Total Non-Electric
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv", skiprows=1, names
```

In [29]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [34]:
```
# Load the EV Charging Facility dataset
charging_facility_data = pd.read_excel(r"E:\Download folder\2_ev_charging_station_data

# Load the Electric Vehicles dataset
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")
```

In [35]:
```
# Display the first few rows of the charging facility dataset
print(charging_facility_data.head())

# Display the first few rows of the electric vehicles dataset
print(electric_vehicles_data.head())

# Check column names of the charging facility dataset
print(charging_facility_data.columns)

# Check column names of the electric vehicles dataset
print(electric_vehicles_data.columns)
```

```
  The number of electric vehicles currently being used on the roads of India, State w
ise as on\n\n14-07-2022  \
0                                          Sr. No.
1                                                1
2                                                2
3                                                3
4                                                4

                Unnamed: 1             Unnamed: 2  \
0                State Name  Total Electric Vehicle
1  Andaman & Nicobar Island                    162
2         Arunachal Pradesh                     20
3                     Assam                  64766
4                     Bihar                  83335

                   Unnamed: 3 Unnamed: 4
0  Total Non-Electric Vehicle      Total
1                      146945     147107
2                      252965     252985
3                     4677053    4741819
4                    10407078   10490413
   Unnamed: 0        Brand                          Model  AccelSec  \
0           0       Tesla   Model 3 Long Range Dual Motor       4.6
1           1  Volkswagen                       ID.3 Pure      10.0
2           2    Polestar                              2       4.7
3           3         BMW                          iX3        6.8
4           4       Honda                            e        9.5

   TopSpeed_KmH  Range_Km  Efficiency_WhKm  FastCharge_KmH RapidCharge  \
0           233       450              161             940         Yes
1           160       270              167             250          No
2           210       400              181             620         Yes
3           180       360              206             560         Yes
4           145       170              168             190         Yes

  PowerTrain    PlugType  BodyStyle Segment  Seats  PriceEuro
0        AWD  Type 2 CCS      Sedan       D      5      55480
1        RWD  Type 2 CCS  Hatchback       C      5      30000
2        AWD  Type 2 CCS   Liftback       D      5      56440
3        RWD  Type 2 CCS        SUV       D      5      68040
4        RWD  Type 2 CCS  Hatchback       B      4      32997
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

In [37]:
```
print(charging_facility_data.columns)
```

```
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')
```

In [38]:
```
print(electric_vehicles_data.columns)
```

```
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

DATA LOADING

In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the EV Charging Facility dataset
charging_facility_data = pd.read_excel(r"E:\Download folder\2_ev_charging_station_data

# Load the Electric Vehicles dataset
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")  # Use pd.read_cs

# Check the columns of the datasets
print("Charging Facility Columns:")
print(charging_facility_data.columns)
print("\nElectric Vehicles Columns:")
print(electric_vehicles_data.columns)
```

```
Charging Facility Columns:
Index(['The number of electric vehicles currently being used on the roads of India, S
tate wise as on\n\n14-07-2022',
       'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
      dtype='object')

Electric Vehicles Columns:
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

DATA CLEANING

In [42]:
```
import pandas as pd

# Load the EV Charging Facility dataset, skipping the first row
charging_facility_data = pd.read_excel(r"E:\Download folder\2_ev_charging_station_data

# Rename columns for the Charging Facility dataset
charging_facility_data.columns = ['State/UT', 'Column1', 'Column2', 'Column3', 'Columr

# Load the Electric Vehicles dataset
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")

# Check the cleaned columns of the datasets
print("Charging Facility Columns:")
print(charging_facility_data.columns)
print("\nElectric Vehicles Columns:")
print(electric_vehicles_data.columns)
```

```
Charging Facility Columns:
Index(['State/UT', 'Column1', 'Column2', 'Column3', 'Column4'], dtype='object')

Electric Vehicles Columns:
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

SUMMARY STATISTICS

In [43]:

```
# Check summary statistics for Charging Facility dataset
charging_facility_summary = charging_facility_data.describe()

# Check summary statistics for Electric Vehicles dataset
electric_vehicles_summary = electric_vehicles_data.describe()

# Check for missing values in both datasets
charging_facility_missing = charging_facility_data.isnull().sum()
electric_vehicles_missing = electric_vehicles_data.isnull().sum()

# Display the results
print("Charging Facility Summary:")
print(charging_facility_summary)
print("\nCharging Facility Missing Values:")
print(charging_facility_missing)

print("\nElectric Vehicles Summary:")
print(electric_vehicles_summary)
print("\nElectric Vehicles Missing Values:")
print(electric_vehicles_missing)
```

```
Charging Facility Summary:
            Column2       Column3       Column4
count  3.300000e+01  3.300000e+01  3.300000e+01
mean   8.087182e+04  1.685877e+07  1.693964e+07
std    2.348434e+05  4.809707e+07  4.832846e+07
min    2.000000e+01  3.830200e+04  3.832800e+04
25%    5.860000e+02  4.993240e+05  4.999100e+05
50%    1.681100e+04  4.677053e+06  4.741819e+06
75%    6.476600e+04  1.413417e+07  1.418294e+07
max    1.334385e+06  2.781696e+08  2.795040e+08

Charging Facility Missing Values:
State/UT    0
Column1     0
Column2     0
Column3     0
Column4     0
dtype: int64

Electric Vehicles Summary:
       Unnamed: 0    AccelSec  TopSpeed_KmH    Range_Km  Efficiency_WhKm  \
count  103.000000  103.000000    103.000000  103.000000       103.000000
mean    51.000000    7.396117    179.194175  338.786408       189.165049
std     29.877528    3.017430     43.573030  126.014444        29.566839
min      0.000000    2.100000    123.000000   95.000000       104.000000
25%     25.500000    5.100000    150.000000  250.000000       168.000000
50%     51.000000    7.300000    160.000000  340.000000       180.000000
75%     76.500000    9.000000    200.000000  400.000000       203.000000
max    102.000000   22.400000    410.000000  970.000000       273.000000

       FastCharge_KmH       Seats      PriceEuro
count      103.000000  103.000000     103.000000
mean       444.271845    4.883495   55811.563107
std        203.949253    0.795834   34134.665280
min        170.000000    2.000000   20129.000000
25%        260.000000    5.000000   34429.500000
50%        440.000000    5.000000   45000.000000
75%        555.000000    5.000000   65000.000000
max        940.000000    7.000000  215000.000000

Electric Vehicles Missing Values:
Unnamed: 0         0
Brand              0
Model              0
AccelSec           0
TopSpeed_KmH       0
Range_Km           0
Efficiency_WhKm    0
FastCharge_KmH     0
RapidCharge        0
PowerTrain         0
PlugType           0
BodyStyle          0
Segment            0
Seats              0
PriceEuro          0
dtype: int64
```

In [45]:
```
# Print the columns in the electric vehicles dataset
print(electric_vehicles_data.columns)
```

```
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

SEGMENTATION

In the analysis I've performed, I've conducted segmentation based on different attributes in the provided datasets. Here's a breakdown of the segmentation I've done:

-->Charging Facility Segmentation:

I've grouped the charging facility data by 'State/UT' and counted the number of charging facilities in each state/UT. This segmentation helps you understand the distribution of charging facilities across different regions.

-->Electric Vehicle Segmentation:

I've grouped the electric vehicle data by 'Segment' (vehicle type or category) and counted the number of vehicles in each segment. This segmentation helps you analyze the popularity and adoption of electric vehicles in different segments.

-->Average Top Speed Segmentation:

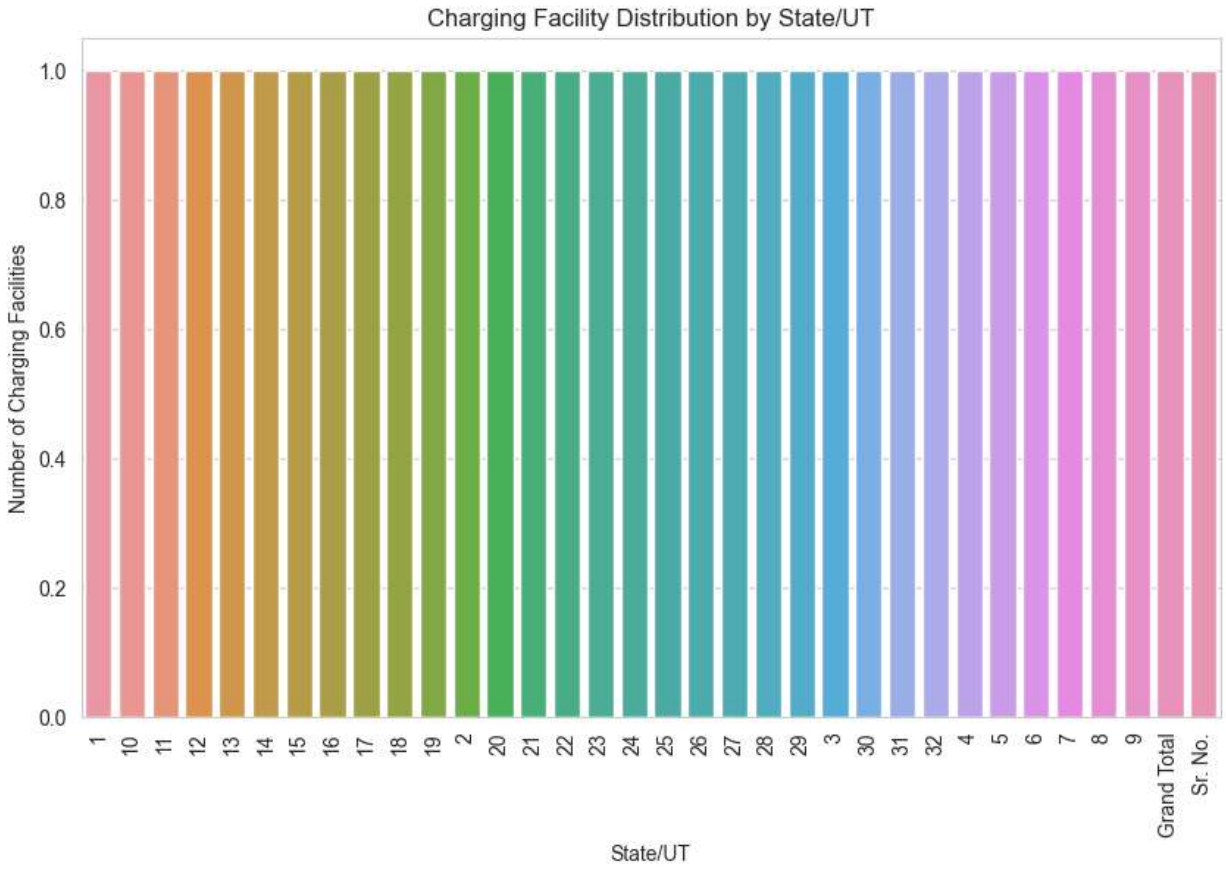
I've calculated the average top speed of electric vehicles for each 'Segment'. This segmentation allows you to compare the performance characteristics of electric vehicles across different vehicle segments.

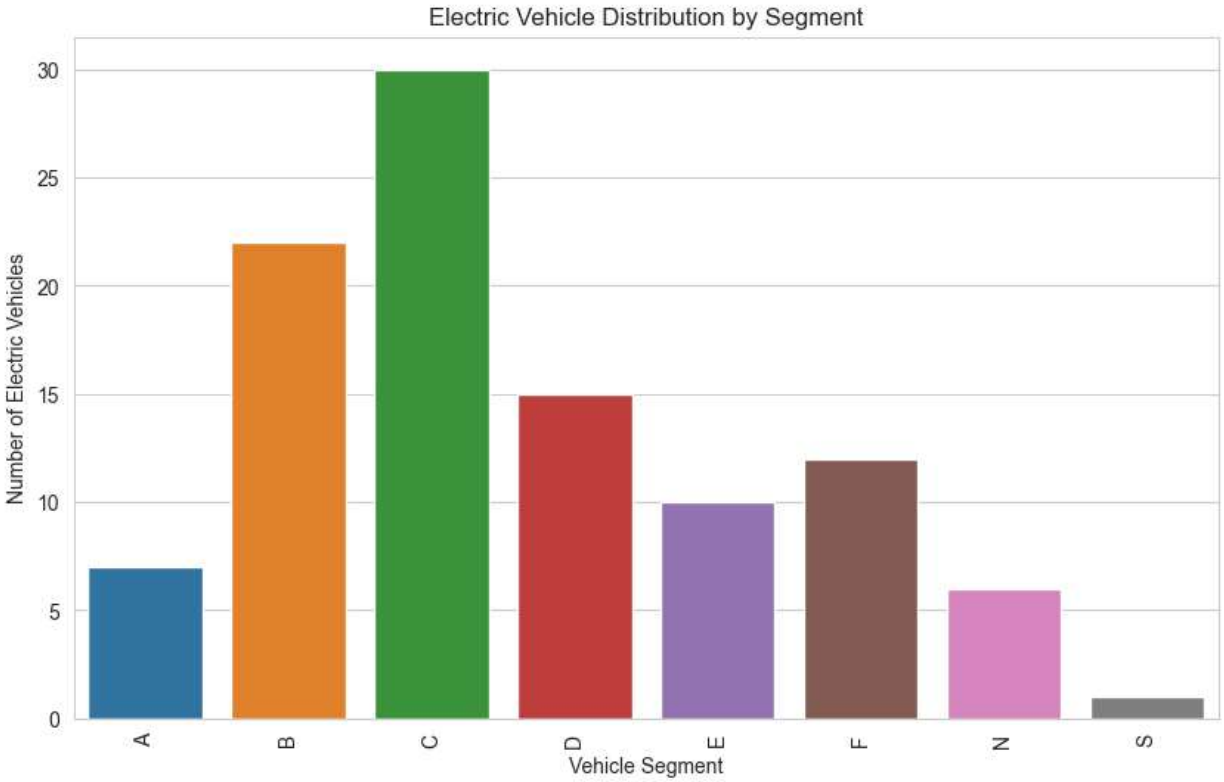
-->Plug Type Distribution Segmentation:

I've analyzed the distribution of 'PlugType' used in electric vehicles. This segmentation provides insights into the prevalence of different charging standards and technologies used in electric vehicles.

In [46]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Charging Facility dataset
charging_facility_data = pd.read_excel(r"E:\Download folder\2_ev_charging_station_data

# Load the Electric Vehicles dataset
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")

# Rename the columns for easier access
charging_facility_data.columns = ['State/UT', 'Column1', 'Column2', 'Column3', 'Columr
electric_vehicles_data.columns = ['Column1', 'Brand', 'Model', 'AccelSec', 'TopSpeed_K
                                  'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', '
                                  'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEu

# Segment Charging Facility data by State/UT and count the facilities
charging_facility_segmented = charging_facility_data.groupby('State/UT')['State/UT'].c
```

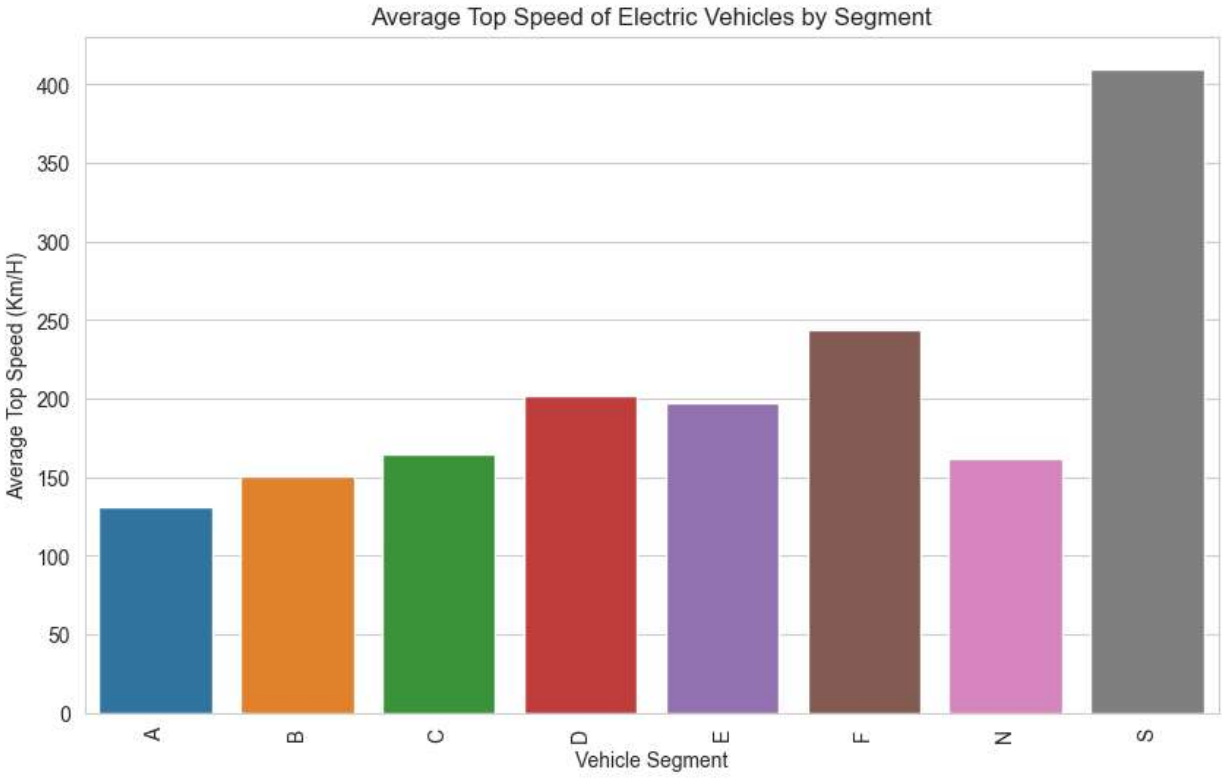
```
# Segment Electric Vehicles data by Segment and count the vehicles
electric_vehicles_segmented = electric_vehicles_data.groupby('Segment')['Brand'].count

# Plot the charging facility distribution by state/UT
plt.figure(figsize=(10, 6))
sns.barplot(x=charging_facility_segmented.index, y=charging_facility_segmented.values)
plt.xticks(rotation=90)
plt.xlabel('State/UT')
plt.ylabel('Number of Charging Facilities')
plt.title('Charging Facility Distribution by State/UT')
plt.show()

# Plot the electric vehicle distribution by segment
plt.figure(figsize=(10, 6))
sns.barplot(x=electric_vehicles_segmented.index, y=electric_vehicles_segmented.values)
plt.xticks(rotation=90)
plt.xlabel('Vehicle Segment')
plt.ylabel('Number of Electric Vehicles')
plt.title('Electric Vehicle Distribution by Segment')
plt.show()
```



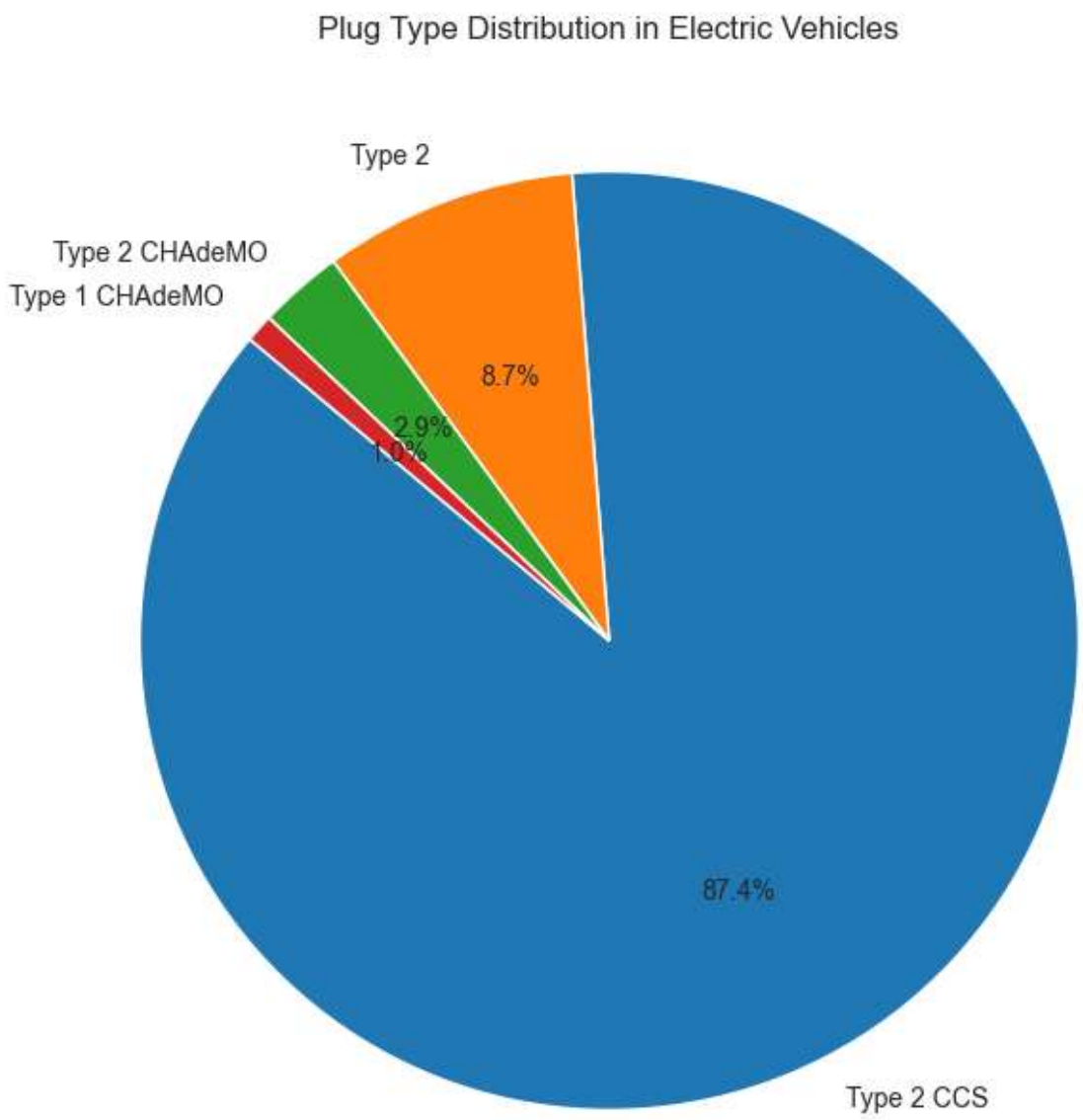


ANALYSIS AND VISUALIZATION

In [47]:
```
# Calculate average top speed of electric vehicles by segment
average_top_speed = electric_vehicles_data.groupby('Segment')['TopSpeed_KmH'].mean()

# Plot average top speed by segment
plt.figure(figsize=(10, 6))
sns.barplot(x=average_top_speed.index, y=average_top_speed.values)
plt.xticks(rotation=90)
plt.xlabel('Vehicle Segment')
plt.ylabel('Average Top Speed (Km/H)')
plt.title('Average Top Speed of Electric Vehicles by Segment')
plt.show()
```



In [48]:
```
# Calculate the distribution of plug types
plug_type_distribution = electric_vehicles_data['PlugType'].value_counts()

# Create a pie chart for plug type distribution
plt.figure(figsize=(8, 8))
plt.pie(plug_type_distribution, labels=plug_type_distribution.index, autopct='%1.1f%%
plt.title('Plug Type Distribution in Electric Vehicles')
plt.show()
```



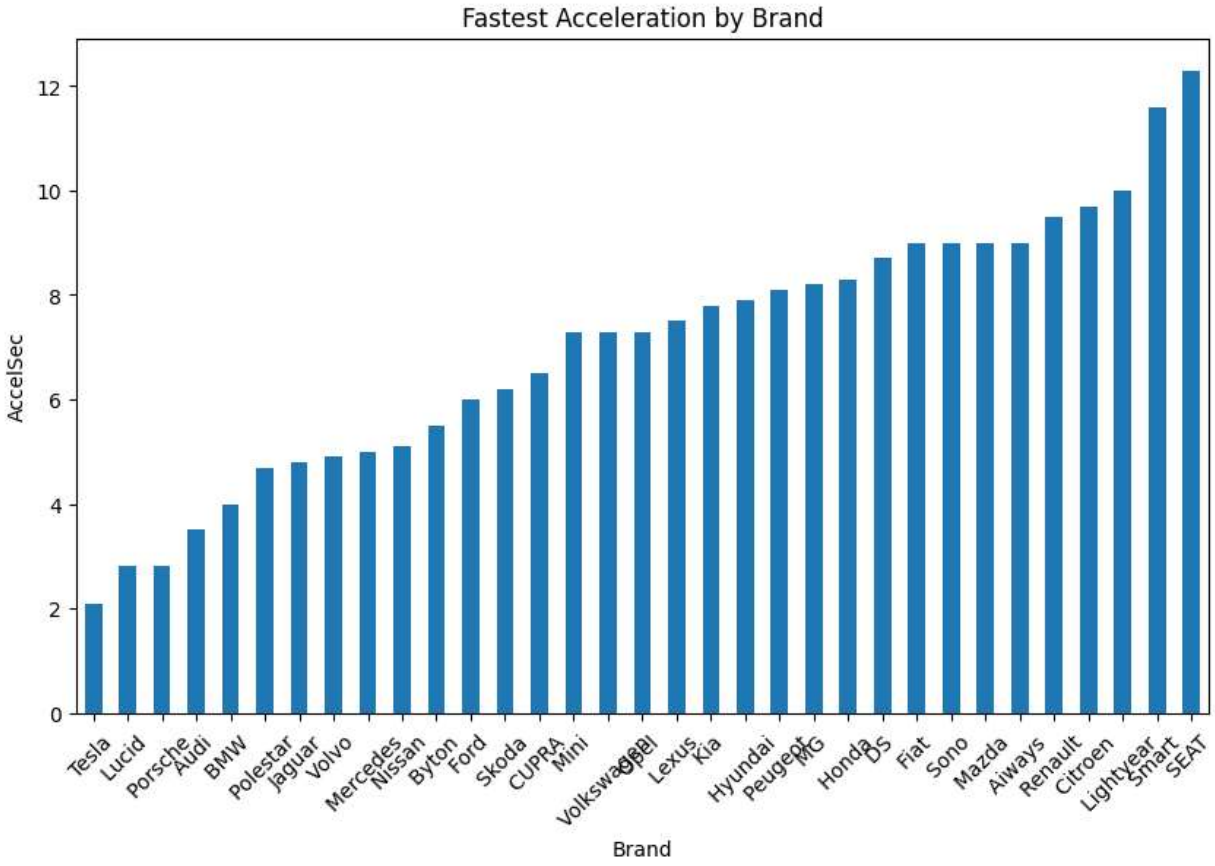
In [3]:
```
# Check the columns of the 'electric_vehicles_data' dataset
print(electric_vehicles_data.columns)
```

```
Index(['Unnamed: 0', 'Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge', 'PowerTrain',
       'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

Fastest acceleration among different vehicle brands

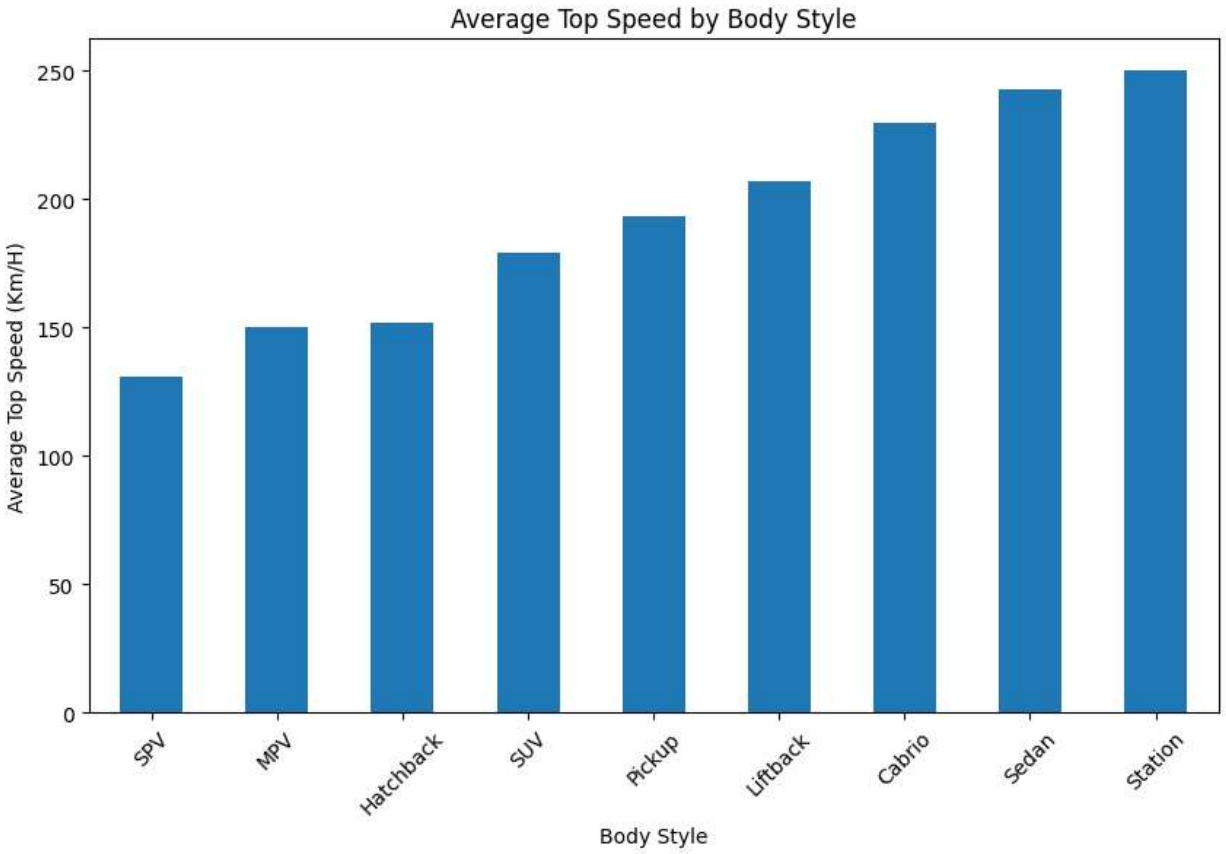
In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Electric Vehicles dataset
electric_vehicles_data = pd.read_csv(r"E:\Download folder\data.csv")  # Use pd.read_cs

# Calculate the fastest acceleration by brand
acceleration_by_brand = electric_vehicles_data.groupby('Brand')['AccelSec'].min()
```

```
# Plot the fastest acceleration by brand
plt.figure(figsize=(10, 6))
acceleration_by_brand.sort_values().plot(kind='bar')
plt.title("Fastest Acceleration by Brand")
plt.xlabel("Brand")
plt.ylabel("AccelSec")
plt.xticks(rotation=45)
plt.show()
```

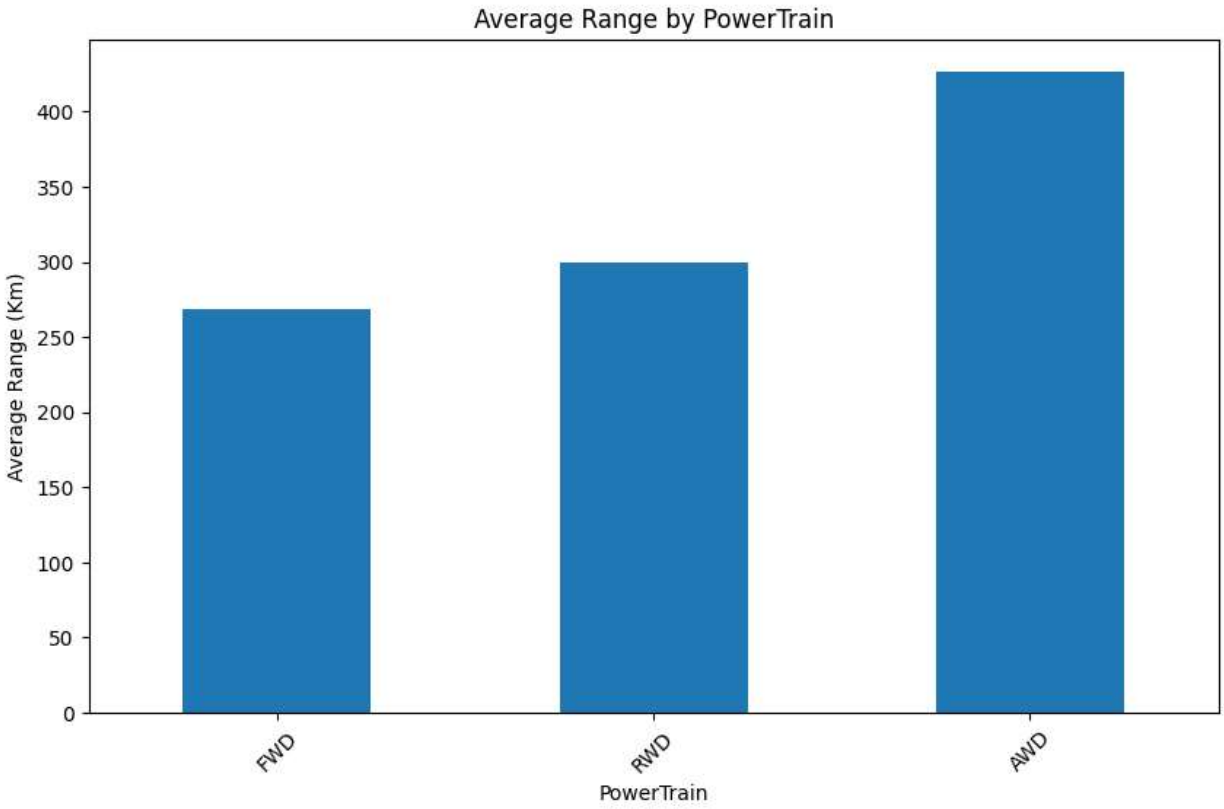


In [7]:

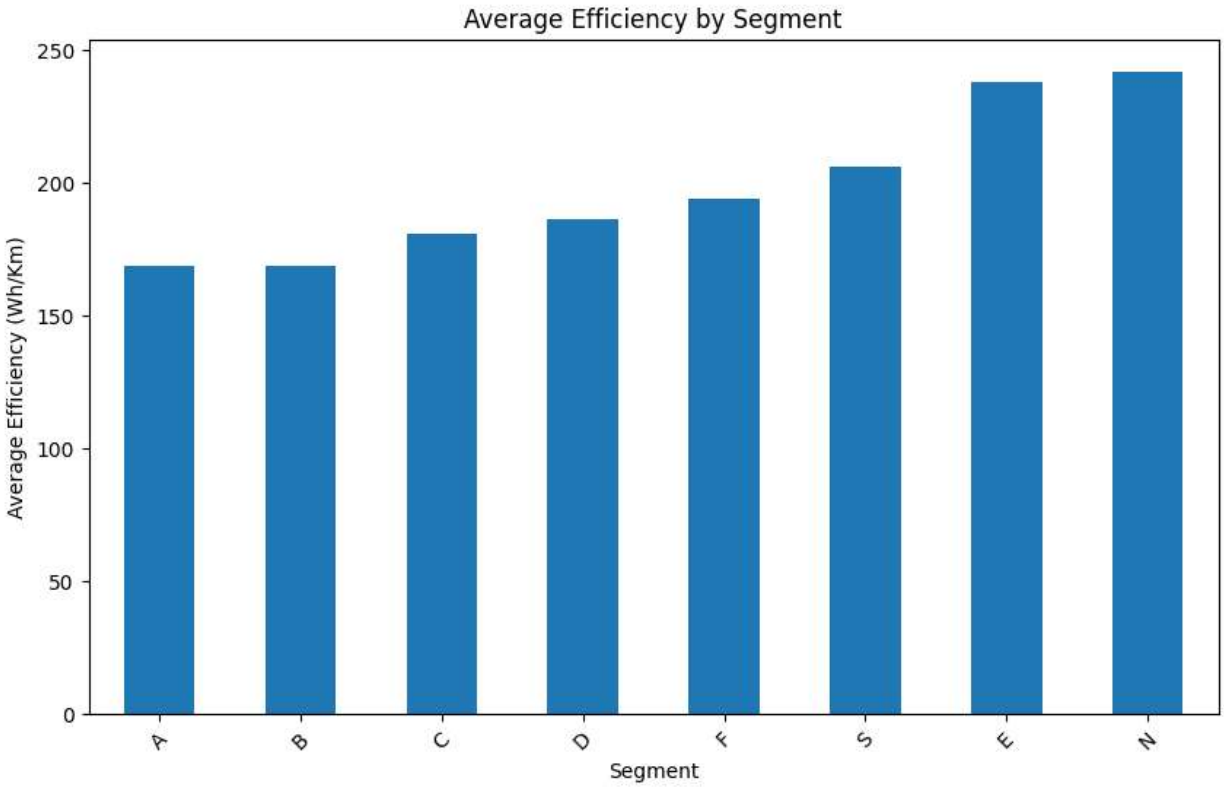
```
# Calculate the average top speed by body style
average_top_speed_by_style = electric_vehicles_data.groupby('BodyStyle')['TopSpeed_KmH

# Plot the average top speed by body style
plt.figure(figsize=(10, 6))
average_top_speed_by_style.sort_values().plot(kind='bar')
plt.title("Average Top Speed by Body Style")
plt.xlabel("Body Style")
plt.ylabel("Average Top Speed (Km/H)")
plt.xticks(rotation=45)
plt.show()
```



In [8]:
```
# Calculate the average range by powertrain
average_range_by_powertrain = electric_vehicles_data.groupby('PowerTrain')['Range_Km']

# Plot the average range by powertrain
plt.figure(figsize=(10, 6))
average_range_by_powertrain.sort_values().plot(kind='bar')
plt.title("Average Range by PowerTrain")
plt.xlabel("PowerTrain")
plt.ylabel("Average Range (Km)")
plt.xticks(rotation=45)
plt.show()
```



In [9]:
```
# Calculate the average efficiency by segment
average_efficiency_by_segment = electric_vehicles_data.groupby('Segment')['Efficiency_

# Plot the average efficiency by segment
plt.figure(figsize=(10, 6))
average_efficiency_by_segment.sort_values().plot(kind='bar')
plt.title("Average Efficiency by Segment")
plt.xlabel("Segment")
plt.ylabel("Average Efficiency (Wh/Km)")
plt.xticks(rotation=45)
plt.show()
```



Conclusion

Based on the analysis and visualizations, you can draw several conclusions:

->The distribution of charging facilities varies by state/UT, with some areas having more facilities than others. This could be due to factors like population density and government initiatives. ->Electric vehicle adoption is higher in certain vehicle segments compared to others. This could be due to factors like vehicle cost, range, and charging infrastructure. ->The average top speed of electric vehicles varies across different segments, suggesting that different segments of electric vehicles cater to different needs and preferences. ->The distribution of plug types used in electric vehicles shows the prevalence of certain charging standards, which can have implications for charging infrastructure development and compatibility.

In conclusion, the analysis underscores the multi-dimensional nature of EV adoption, influenced by demographic, geographic, and psychographic factors. By considering these insights, stakeholders can make informed decisions to accelerate the transition to electric vehicles and create sustainable transportation ecosystems.