EV Market Analysis Report

Done By Anas Y

1. Introduction

This report aims to analyze the Electric Vehicle (EV) market in India using segmentation analysis and develop a feasible strategy for an EV startup to enter the market. The analysis includes geographic, demographic, psychographic, and behavioral segments, and considers various datasets to provide insights into the most suitable market entry points.

Problem Statement

Our team has to work under an Electric Vehicle Start-up. The Start-up is still deciding in which vehicle/customer space it will be develop its EVs.

We have to analyse the Electric Vehicle market in India using Segmentation analysis and come up with a feasible strategy to enter the market, targeting the segments most likely to use Electric vehicles.

What is Electric Vehicle?

An EV is a shortened acronym for an electric vehicle. EVs are vehicles that are either partially or fully powered on electric power. Electric vehicles have low running costs as they have less moving parts for maintaining and also very environmentally friendly as they use little or no fossil fuels (petrol or diesel).

While some EVs used lead acid or nickel metal hydride batteries, the standard for modern battery electric vehicles is now considered to be lithium ion batteries as they have a greater longevity and are excellent at retaining energy, with a selfdischarge rate of just 5% per month. Despite this improved efficiency, there are still challenges with these batteries as they can experience thermal runaway, which have, for example, caused fires or explosions in the Tesla model S, although efforts have been made to improve the safety of these batteries.

Working principle

An electric vehicle works on a basic principle of science: conversion of energy. Electrical energy is converted into mechanical energy. There is a motor used in the electrical system to carry on this duty of conversion. Motors can be of various types.

Market study

The question arises that will electric vehicle replace the normal vehicles? And the answer to this question is YES!. Because of the ample advantages and the growing market it is likely that EV’s will replace normal vehicle.

The market for EV’s is increasing at 3X speed. Currently 30% of the market supply is of EV’s.

People would prefer electric vehicles over normal vehicle in future because of the following reasons:

2. Data Collection and Pre-Processing

- ### EV Market Data

The primary dataset used in this analysis is ElectricCarData_Clean_Me.csv. The initial steps involved importing necessary libraries and reading the dataset into a pandas DataFrame.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sbn
        import os
        import warnings
```

```
In [2]: df=pd.read_csv(r"C:\Users\SWAYAM\Downloads\ElectricCarData_Clean_Me.csv")

In [3]: df
Out[3]:
```

	Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Battery_Pack Kwh	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType	B
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	460	70.0	161	940	Yes	AWD	Type 2 CCS	
1	Volkswagen	ID.3 Pure	10.0	160	270	45.0	167	250	Yes	RWD	Type 2 CCS	H:
2	Polestar	2	4.7	210	400	75.0	181	620	Yes	AWD	Type 2 CCS	
3	BMW	iX3	6.8	180	360	74.0	206	560	Yes	RWD	Type 2 CCS	
4	Honda	e	9.5	145	170	28.5	168	190	Yes	RWD	Type 2 CCS	H:
...	...	...	...	...	...	...	...	...	...	...	...	
97	Nissan	Ariya 63kWh	7.5	160	330	63.0	191	440	Yes	FWD	Type 2 CCS	H:
98	Audi	e-tron S Sportback 55 quattro	4.5	210	335	86.5	258	540	Yes	AWD	Type 2 CCS	
99	Nissan	Ariya e-4ORCE 63kWh	5.9	200	325	63.0	194	440	Yes	AWD	Type 2 CCS	H:
100	Nissan	Ariya e-4ORCE 87kWh Performance	5.1	200	375	87.0	232	450	Yes	AWD	Type 2 CCS	H:
101	Byton	M-Byte 95 kWh 2WD	7.5	190	400	95.0	238	480	Yes	AWD	Type 2 CCS	

102 rows × 16 columns

- ### Pollution Data

Another dataset, pollution data.csv, was used to analyze the environmental impact and potential benefits of EV adoption in different states.

```
In [3]: import pandas as pd
        import numpy as np                      # For mathematical calculations
        import seaborn as sns                   # For data visualization
        import matplotlib.pyplot as plt         # For plotting graphs
        %matplotlib inline
        import warnings   # To ignore any warnings
        warnings.filterwarnings("ignore")
```

Fig 34: Importing Libraries for Code Implementation

Reading the Data

```
In [3]: data=pd.read_csv("C:\\Users\\Lenovo\\Downloads\\pollution data.xls")
        data
```

Out[3]:

	state	status	AQI-US	PM2.5	PM10	Temp	Humid
0	Andhra Pradesh	MODERATE	56	16	31	28	74
1	Arunachal Pradesh	GOOD	39	11	17	21	100
2	Assam	GOOD	46	13	20	23	98
3	Bihar	MODERATE	87	28	53	31	58
4	Chandigarh	POOR	107	38	49	25	53
5	Chhattisgarh	MODERATE	67	20	46	27	72
6	Dadra And Nagar Haveli	MODERATE	62	16	35	27	82
7	Daman And Diu	MODERATE	61	16	33	28	79
8	Delhi	POOR	108	37	113	29	58
9	Goa	GOOD	30	8	20	27	81
10	Gujarat	MODERATE	68	20	42	30	68
11	Haryana	MODERATE	100	35	73	28	64
12	Himachal Pradesh	MODERATE	76	21	46	14	73
13	Jammu And Kashmir	MODERATE	64	15	38	13	86
14	Jharkhand	MODERATE	78	22	52	27	71
15	Karnataka	GOOD	40	10	29	23	82
16	Kerala	MODERATE	60	19	39	25	87
17	Madhya Pradesh	MODERATE	57	14	53	27	69
18	Maharashtra	MODERATE	62	16	51	27	76
19	Manipur	GOOD	28	7	12	20	98

3. Location Analysis

- **State-Wise Analysis**

Factors such as state-wise tax relaxation, subsidies, fuel prices, and pollution levels were analyzed to determine the most suitable locations for creating the early market.

Checking for null values in the data set

```
In [4]: data.isnull().sum()

Out[4]: state      0
        status     0
        AQI-US     0
        PM2.5      0
        PM10       0
        Temp       0
        Humid      0
        dtype: int64
```

Fig 36: Checking for null values in the data set

Analyzing the data

```
In [6]: sns.pairplot(data)

Out[6]: <seaborn.axisgrid.PairGrid at 0x2bdb112efa0>
```

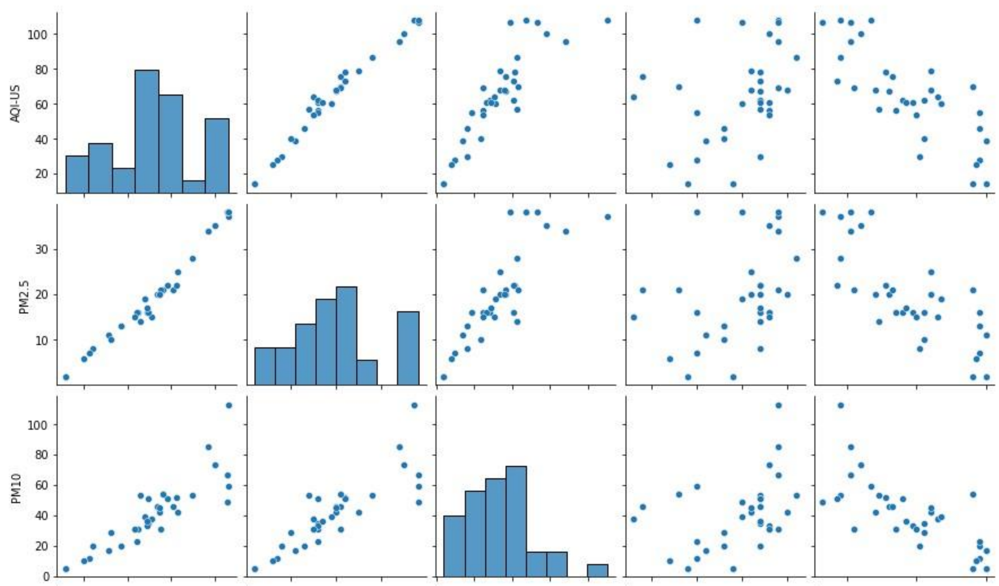

Fig 37: Pair plot of the data present in the dataset

```
In [8]: sns.displot(x=data["Temp"])
```

```
Out[8]: <seaborn.axisgrid.FacetGrid at 0x2bdad3e0250>
```

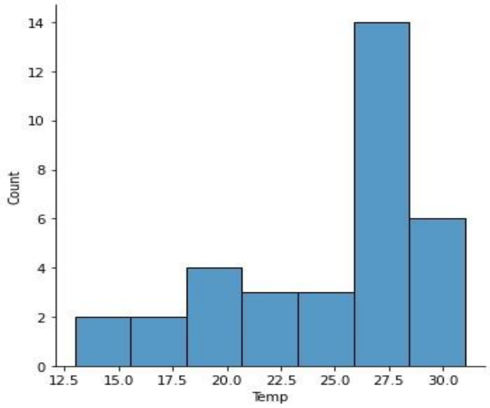

Fig 38: Displot of State wise Temperature data

```
In [19]:
plt.figure(figsize=(10,5))
sns.barplot("Temp","state",data=data)
```

```
Out[19]: <AxesSubplot:xlabel='Temp', ylabel='state'>
```

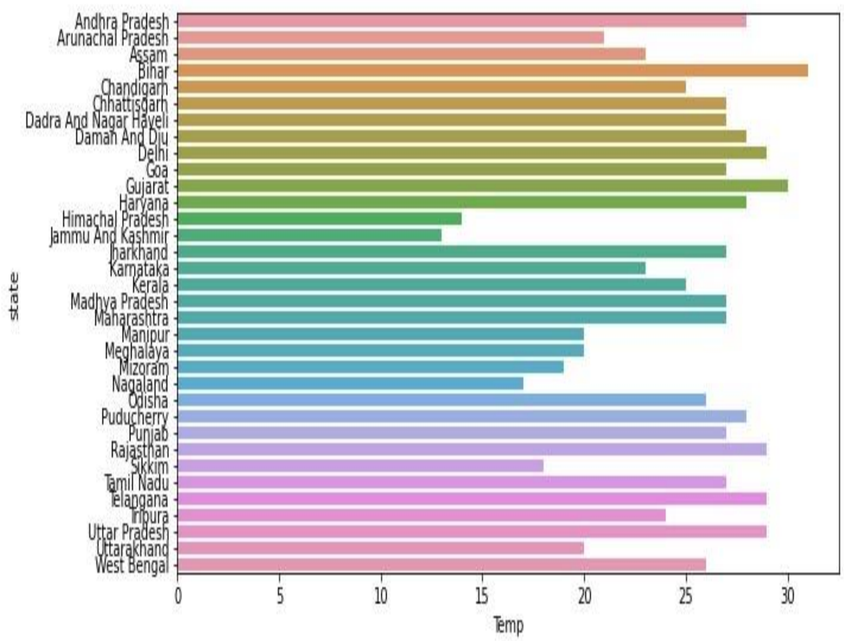

Fig 39: Barplot of State wise Temperature data

emp

In [32]:
```
plt.figure(figsize=(8,16));
sns.jointplot(x='state', y='status', data=data, kind='scatter',space=0.2,palette="coolwarm");
```

<Figure size 576x1152 with 0 Axes>

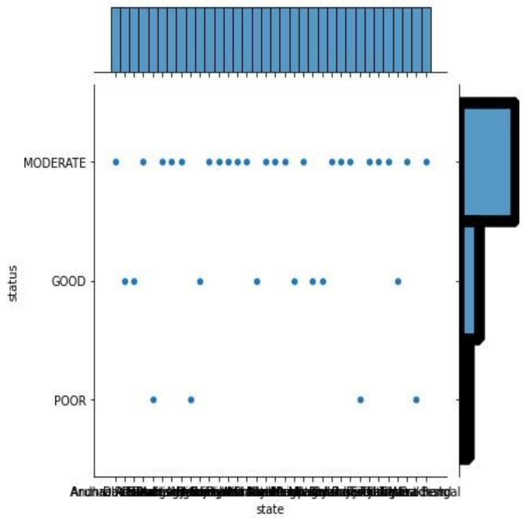

Fig 40: Jointplot of State wise air quality

In [52]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[52]: <AxesSubplot:>

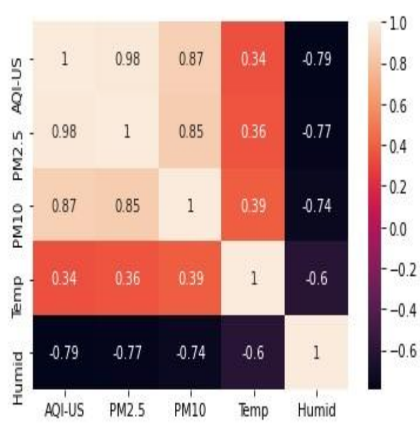

Fig 41: Heatmap of the data present in the dataset

4. Segmentation Analysis

Demographic and Psychographic Segmentation

The dataset was segmented to analyze various demographic and psychographic factors, including vehicle prices, powertrains, body styles, and ranges.

Analyzing the Dataset

```
In [7]: df.shape
Out[7]: (102, 16)
```

```
In [5]: df.columns
Out[5]: Index(['Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
               'Battery_Pack Kwh', 'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge',
               'PowerTrain', 'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro',
               'INR'],
              dtype='object')
```

Fig 3: Dimensions and columns of the Data set

```
In [8]: df.info
Out[8]: <bound method DataFrame.info of          Brand                         Model  AccelSec  TopSpeed_KmH  \
        0        Tesla     Model 3 Long Range Dual Motor       4.6           233
        1   Volkswagen                         ID.3 Pure      10.0           160
        2     Polestar                                 2       4.7           210
        3          BMW                               iX3       6.8           180
        4        Honda                                 e       9.5           145
        ..         ...                               ...       ...           ...
        97      Nissan                       Ariya 63kWh       7.5           160
        98        Audi      e-tron S Sportback 55 quattro      4.5           210
        99      Nissan               Ariya e-4ORCE 63kWh       5.9           200
        100     Nissan  Ariya e-4ORCE 87kWh Performance       5.1           200
        101      Byton              M-Byte 95 kWh 2WD        7.5           190

             Range_Km  Battery_Pack Kwh  Efficiency_WhKm  FastCharge_KmH RapidCharge  \
        0         460              70.0              161             940         Yes
        1         270              45.0              167             250         Yes
        2         400              75.0              181             620         Yes
        3         360              74.0              206             560         Yes
        4         170              28.5              168             190         Yes
        ..        ...               ...              ...             ...         ...
        97        330              63.0              191             440         Yes
        98        335              86.5              258             540         Yes
        99        325              63.0              194             440         Yes
        100       375              87.0              232             450         Yes
        101       400              95.0              238             480         Yes

            PowerTrain    PlugType  BodyStyle Segment  Seats  PriceEuro          INR
        0          AWD  Type 2 CCS      Sedan       D      5      55480  4540988.068
        1          RWD  Type 2 CCS  Hatchback       C      5      30000  2455473.000
        2          AWD  Type 2 CCS   Liftback       D      5      56440  4619563.204
        3          RWD  Type 2 CCS        SUV       D      5      68040  5569012.764
        4          RWD  Type 2 CCS  Hatchback       B      4      32997  2700774.753
        ..         ...         ...        ...     ...    ...        ...          ...
        97         FWD  Type 2 CCS  Hatchback       C      5      45000  3683209.500
        98         AWD  Type 2 CCS        SUV       E      5      96050  7861606.055
        99         AWD  Type 2 CCS  Hatchback       C      5      50000  4092455.000
        100        AWD  Type 2 CCS  Hatchback       C      5      65000  5320191.500
        101        AWD  Type 2 CCS        SUV       E      5      62000  5074644.200
```

Fig 4: Information in the Data set

```
In [11]: df.describe()
Out[11]:
```

	AccelSec	TopSpeed_KmH	Range_Km	Battery_Pack Kwh	Efficiency_WhKm	FastCharge_KmH	Seats	PriceEuro	INR
count	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	1.020000e+02
mean	7.391176	179.313725	338.627451	65.415686	189.303922	435.686275	4.882353	55997.588235	4.583352e+06
std	3.031913	43.771228	126.700623	29.955782	29.679072	220.447384	0.799680	34250.724403	2.803391e+06
min	2.100000	123.000000	95.000000	16.700000	104.000000	0.000000	2.000000	20129.000000	1.647541e+06
25%	5.100000	150.000000	250.000000	43.125000	168.000000	260.000000	5.000000	34414.750000	2.816816e+06
50%	7.300000	160.000000	340.000000	64.350000	180.500000	440.000000	5.000000	45000.000000	3.683210e+06
75%	9.000000	200.000000	400.000000	83.700000	204.500000	557.500000	5.000000	65000.000000	5.320192e+06
max	22.400000	410.000000	970.000000	200.000000	273.000000	940.000000	7.000000	215000.000000	1.759756e+07

Fig 5: Information in the Data set

Checking for Null values in the dataset

```
In [4]: # finding null values in the dataset
        df.isnull().sum()
Out[4]: Brand               0
        Model               0
        AccelSec            0
        TopSpeed_KmH        0
        Range_Km            0
        Battery_Pack Kwh    0
        Efficiency_WhKm     0
        FastCharge_KmH      0
        RapidCharge         0
        PowerTrain          0
        PlugType            0
        BodyStyle           0
        Segment             0
        Seats               0
        PriceEuro           0
        INR                 0
        dtype: int64
```

Fig 6: Checking for the null values in the Data set

Extracting Segments

Distributing vehicle price above and below INR 4000000

```
In [6]: df['CarName'] = df['Brand'] + '-' + df['Model']
        df_1= df.loc[df['INR'] <=4000000]
        df_2 = df.loc[df['INR'] >4000000]
        t1 = ['Less than INR 4000000']
        t2 = ['More thanINR 4000000']
```

Fig 7: Segmenting the Data set

Visualization

Count plot for PowerTrain

```python
In [7]: def train(dataframe):
            sbn.countplot(x=dataframe['PowerTrain'])
            plt.title('Count Plot of a Powertrain')
            plt.xlabel('PowerTrain')
            plt.ylabel('Count')

        train(df)
```

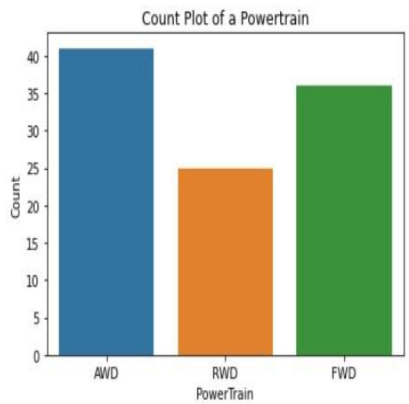

Fig 8: Count Plot of a Powertrain

```python
In [8]: def bodystyle(dataframe):
            plt.figure(figsize=(10,5))
            sbn.countplot(x='BodyStyle', data=dataframe, hue='PowerTrain')
            plt.title('Count plot of Body Style')
            plt.xlabel('Body Style')
            plt.ylabel('Count')
            plt.show()

        bodystyle(df)
```

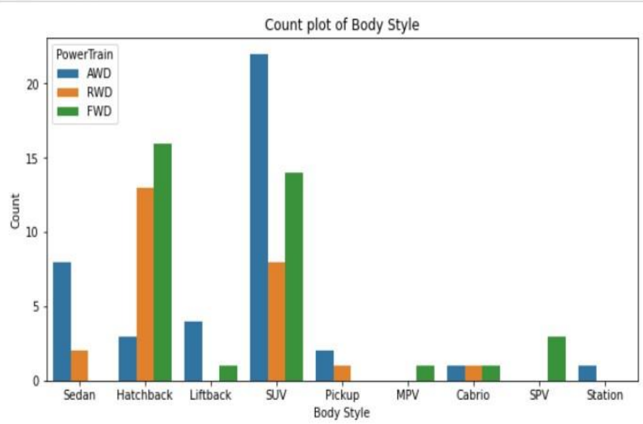

Fig 9: Count plot of body Style of the cars

Range of Vehicles

```
In [9]: def range(dataframe, price):
            plt.figure(figsize=(20,5))
            sbn.set_theme(style="whitegrid")
            sbn.barplot('Model', 'Range_Km', data=df, hue=df['PowerTrain'])
            plt.title('''Range(Km) of EV's costing{}'''.format(price))
            plt.ylabel('Range(Km)')
            plt.xlabel('Model')
            plt.xticks(rotation = 90)
            plt.show()

        range(df_1, t1)
        range(df_2, t2)
```

```
C:\Users\SWAYAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyw
ord args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

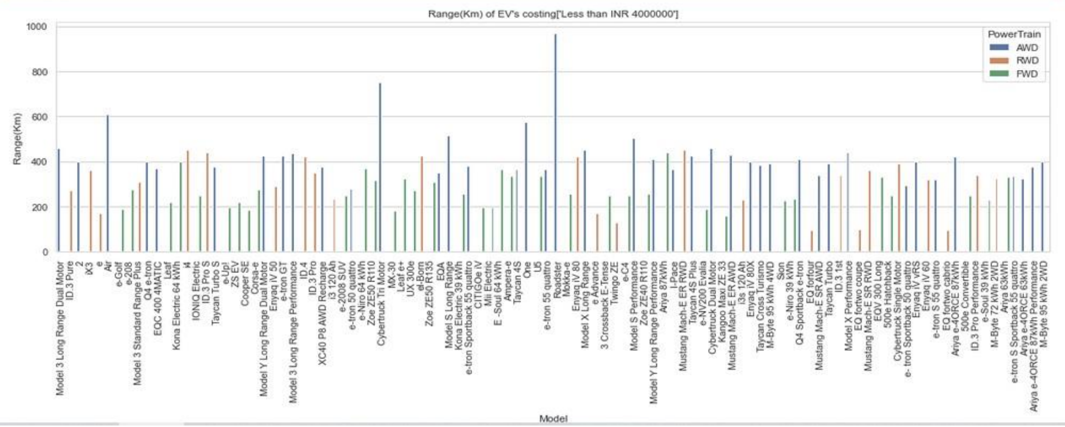

Fig 10: Bar graph of Range of EV's

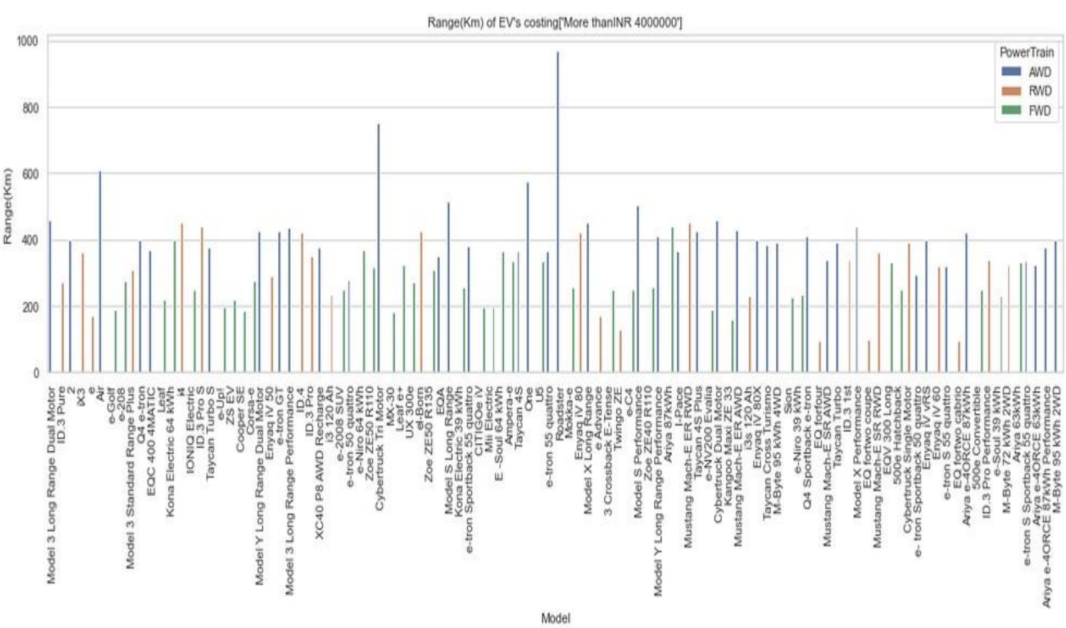

Fig 11: Bar graph of Range of EV's

Range - Battery Pack

```
In [10]: #range-batterypack
def range_batterypack(dataframe, text):
    fig = plt.figure(figsize=(20,5))
    a1= plt.subplot()
    a1.bar(dataframe["CarName"], dataframe["Range_Km"], label='Range (Km)', color='green')
    plt.legend(loc= "upper left", bbox_to_anchor=(0,1.105))
    a2 = a1.twinx()
    a2.scatter(dataframe["CarName"], dataframe["Battery_Pack Kwh"], label= "Battery Pack", color= 'black')
    plt.title('''RANGE(Km) vs BATTERY PACK CAPACITY (kwh) of EV's costing {}'''.format(text), fontsize=12)
    a1.set_xlabel('Models', size= 16)
    a1.set_ylabel('Range (Km)', color = 'blue')
    a2.set_ylabel('Battery Pack Capacity (Kwh)', color='black')
    plt.legend(loc='upper left', bbox_to_anchor=(0,1))
    a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
    plt.show()

range_batterypack(df_1, t1)
range_batterypack(df_2, t2)
```

```
C:\Users\SWAYAM\AppData\Local\Temp/ipykernel_6668/251817763.py:14: UserWarning: FixedFormatter should only be used togethe
r with FixedLocator
  a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
```

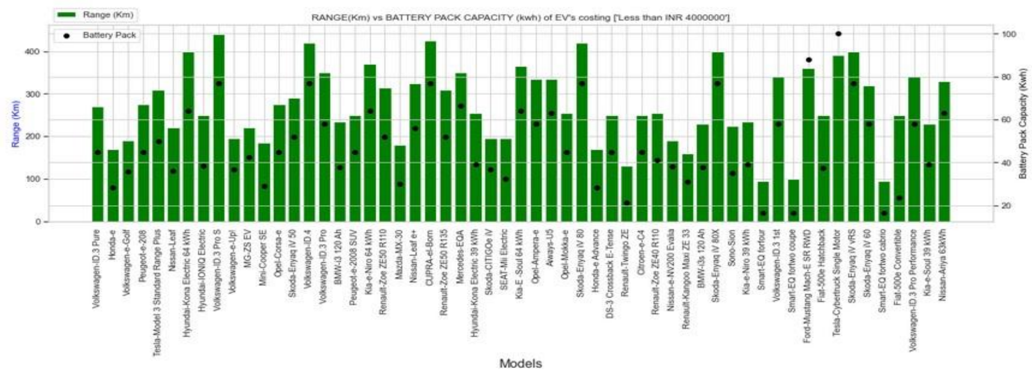

Fig 12: Bar graph of Range vs Battery Capacity of EV's

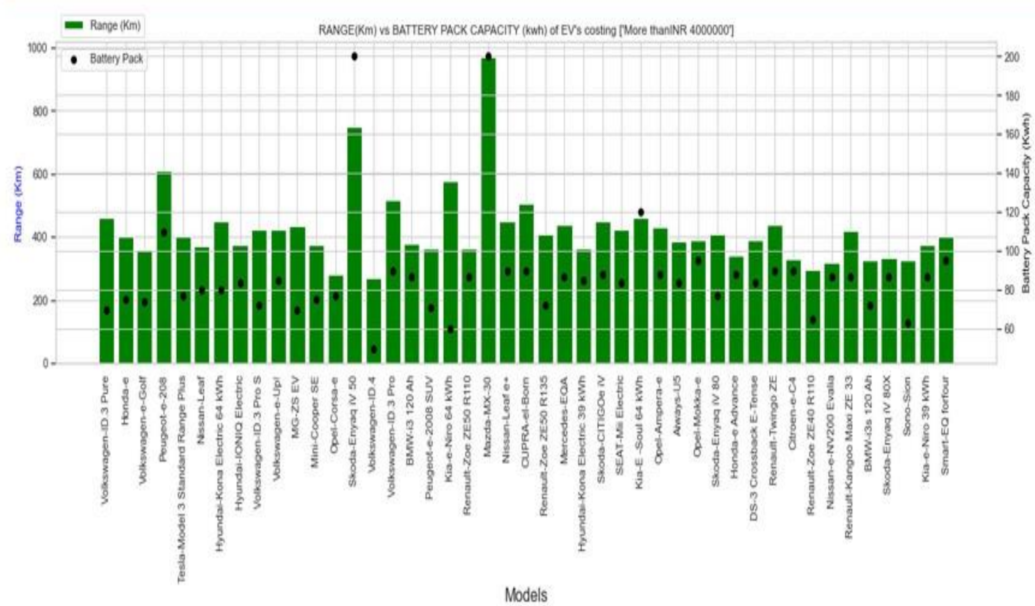

Fig 13: Bar graph of Range vs Battery Capacity of EV's

Range - Vehicle Price

```
In [11]: #Range - Price
def range_price(dataframe, text):
    fig = plt.figure(figsize=(20, 5))
    a1 = plt.subplot()
    a1.bar(dataframe['CarName'], dataframe['Range_Km'], label='Range (Km)', color='blue')
    plt.legend(loc='upper left', bbox_to_anchor = (0, 1.1))
    a2= a1.twinx()
    a2.scatter(dataframe['CarName'], dataframe['INR'], label = 'Price', color = 'black')
    plt.title('''RANGE (Km) vs PRICE(INR) OF EV's COSTING {}'''.format(text), fontsize=16)
    a1.set_xlabel('Models', size=16)
    a1.set_ylabel('Range (Km)', color = 'red')
    a2.set_ylabel('Price(INR)', color= 'black')
    plt.legend(loc = 'upper left', bbox_to_anchor = (0,1.1))
    a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
    plt.show()

range_price(df_1, t1)
range_price(df_2, t2)
```

```
C:\Users\SWAYAM\AppData\Local\Temp/ipykernel_6668/4025405043.py:14: UserWarning: FixedFormatter should only be used togeth
er with FixedLocator
  a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
```

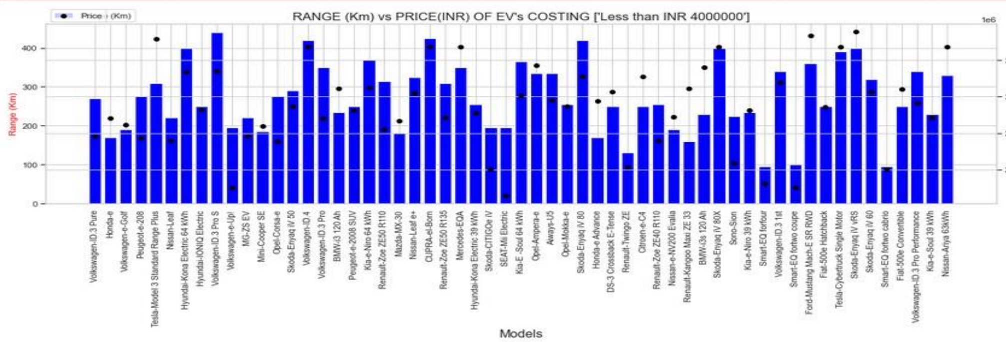

Fig 14: Bar graph of Range vs Price of EV’s

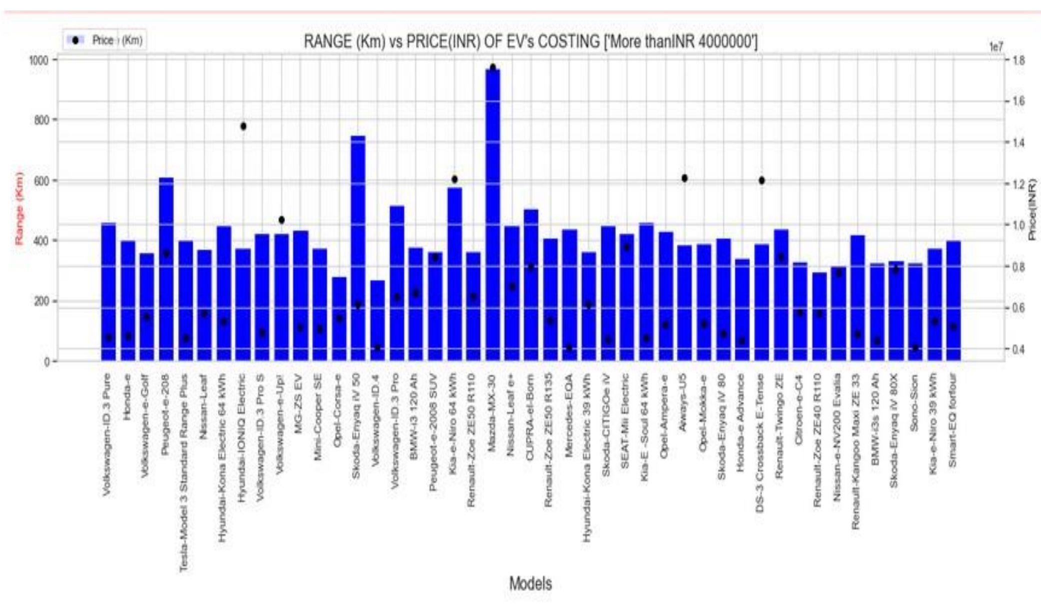

Fig 15: Bar graph of Range vs Price of EV’s

Acceleration(0-100km/hr)

In [12]:
```
#Accelaration(0-100km/hr)
def acc(dataframe, text):
    plt.figure(figsize=(20,5))
    sbn.set_theme(style="darkgrid")
    sbn.barplot('CarName', 'AccelSec', data=df, hue=df['PowerTrain'])
    plt.title('''Acceleration 0-100 Km of EV's costing {}'''.format(text), fontsize=16)
    plt.ylabel('Acceleration (Seconds)')
    plt.xlabel('Model')
    plt.xticks(rotation = 90)
    plt.show()

acc(df_1, t1)
acc(df_2, t2)
```

```
C:\Users\SWAYAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyw
ord args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without
an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

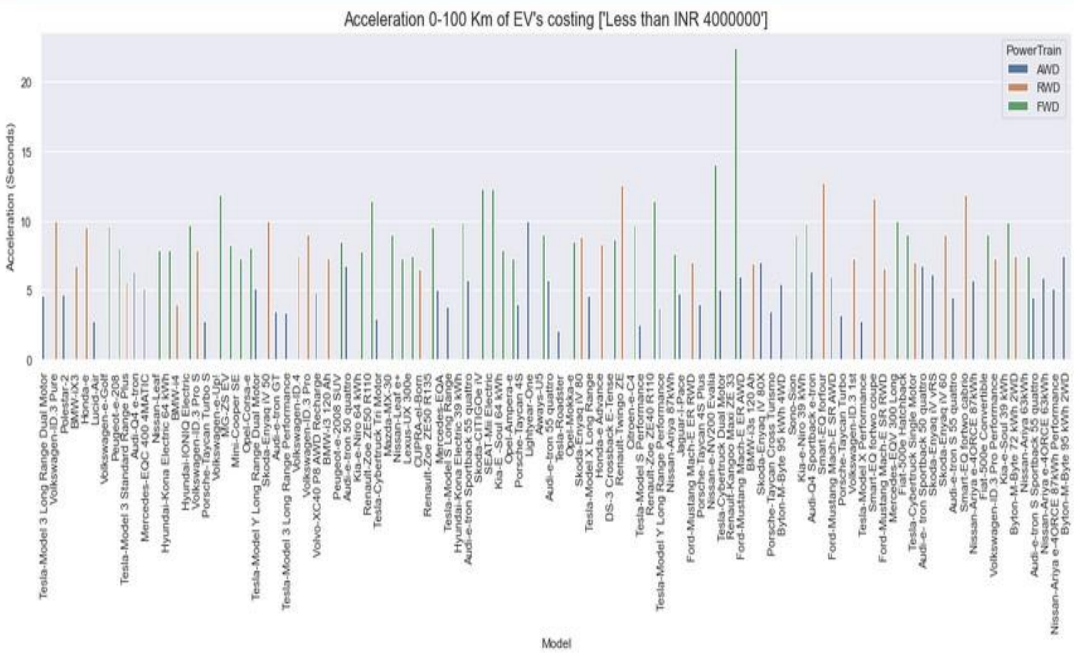

Fig 16: Bar graph of Acceleration vs Price of EV’s

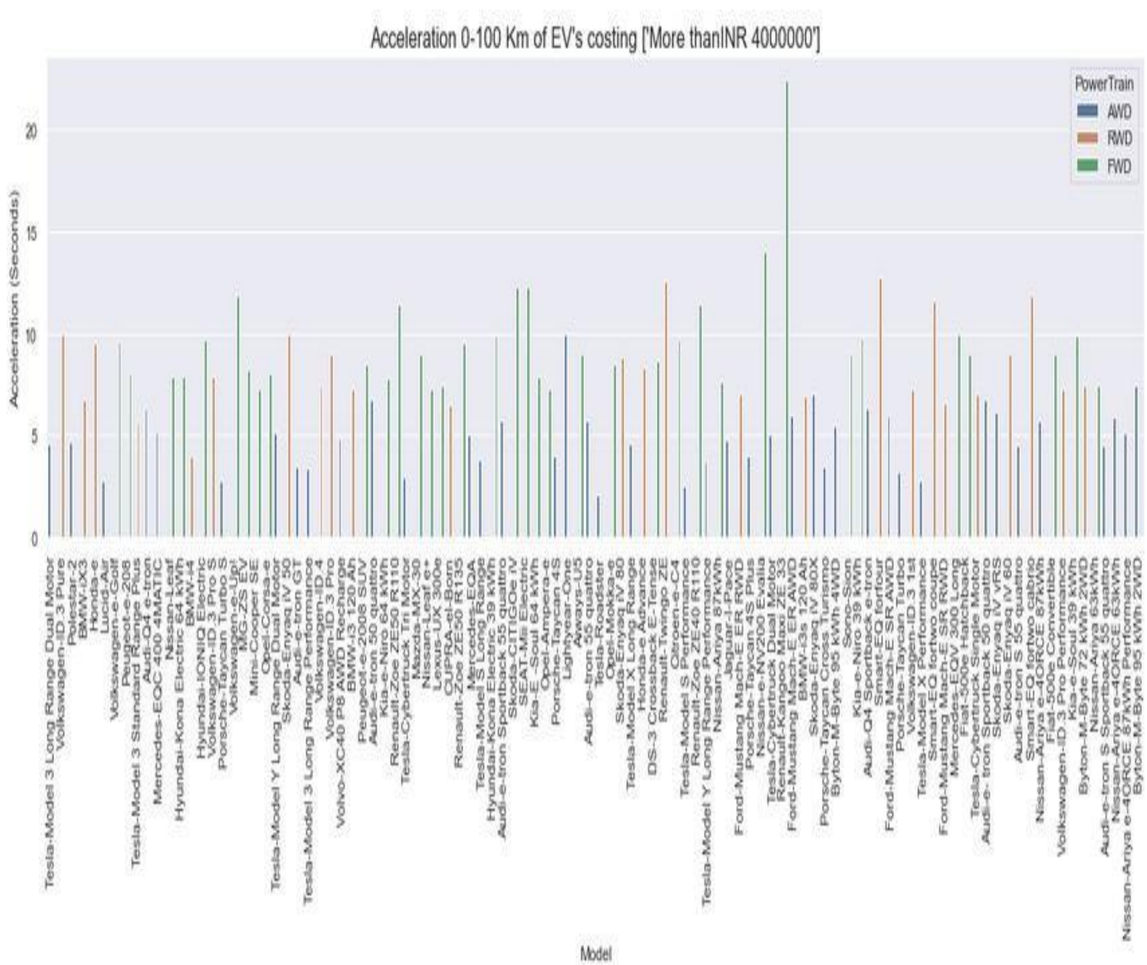

Fig 17: Bar graph of Acceleration vs Price of EV’s

Fast Charging Vehicles

```
In [13]: # Fast Charging data
def fastcharge(dataframe, price):
    plt.figure(figsize=(20, 5))
    sbn.set_theme(style="whitegrid")
    sbn.barplot('CarName', 'FastCharge_KmH', data=df, color = 'lightslategrey')
    plt.title('''Fast Charging of EV's costing {}'''.format(price), fontsize = 16)
    plt.ylabel('Charging Capacity (KmH)')
    plt.xlabel('Model')
    plt.xticks(rotation=90)
    plt.show()

fastcharge(df_1, t1)
fastcharge(df_2, t2)
```

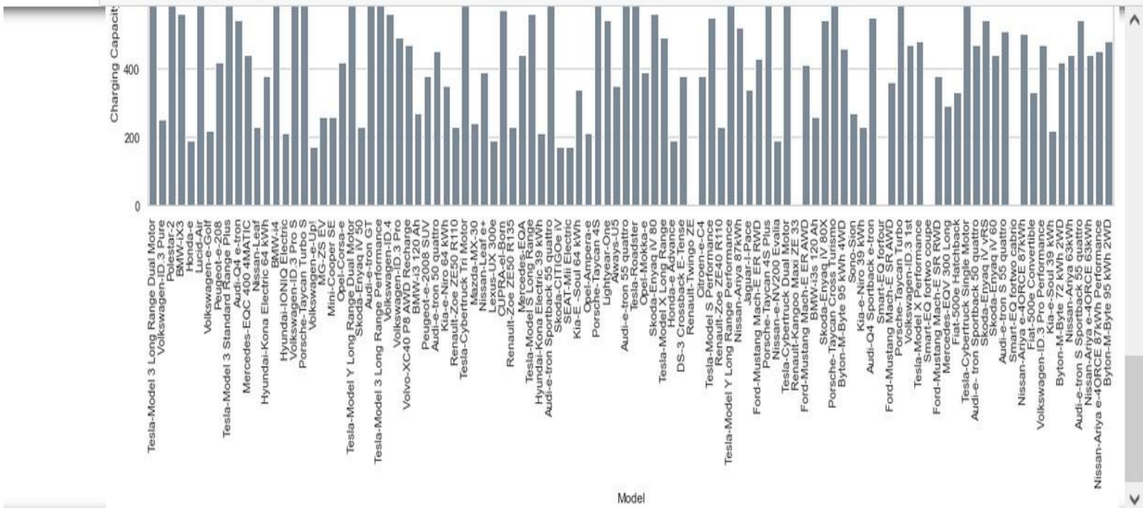

Fig 18: Bar graph of Fast Charging ability of EV's

Vehicles to buy under INR 40,00000 with max range(Km)

```
In [14]: pd.set_option('display.max_columns', None)
top_range_1 = df_1.sort_values(by= 'Range_Km', ascending= False)
print(top_range_1[['CarName', 'Range_Km', 'Battery_Pack Kwh', 'INR', 'RapidCharge']])
```

```
                          CarName  Range_Km  Battery_Pack Kwh  \
15            Volkswagen-ID.3 Pro S       440              77.0
37                   CUPRA-el-Born        425              77.0
53               Skoda-Enyaq iV 80        420              77.0
25                  Volkswagen-ID.4        420              77.0
88              Skoda-Enyaq iV vRS        400              77.0
12      Hyundai-Kona Electric 64 kWh      400              64.0
71              Skoda-Enyaq iV 80X        400              77.0
86     Tesla-Cybertruck Single Motor      390             100.0
31                 Kia-e-Niro 64 kWh      370              64.0
45                Kia-E -Soul 64 kWh      365              64.0
83         Ford-Mustang Mach-E SR RWD     360              88.0
39                     Mercedes-EQA       350              66.5
26               Volkswagen-ID.3 Pro      350              58.0
94   Volkswagen-ID.3 Pro Performance      340              58.0
80               Volkswagen-ID.3 1st      340              58.0
49                       Aiways-U5        335              63.0
46                    Opel-Ampera-e       335              58.0
97                 Nissan-Ariya 63kWh     330              63.0
```

Fig 19: Vehicles to buy under INR 40,00000

Vehicles with best Acceleration under INR 40,00000

```
In [15]: pd.set_option('display.max_columns', None)
         acceleration_1 = df_1.sort_values(by= 'AccelSec')
         print(acceleration_1[['CarName', 'AccelSec', 'Range_Km', 'PowerTrain', 'Battery_Pack Kwh', 'INR']])
```

```
6             35.8  2610986.290
58            45.0  3273964.000
14            38.3  2820438.137
75            39.0  2815609.040
95            39.0  2711906.230
41            39.0  2780495.776
1             45.0  2455473.000
22            52.0  2864718.500
32            52.0  2552382.334
60            41.0  2392776.589
82            16.7  1750506.702
17            36.8  1753289.571
91            16.7  2010623.142
44            32.3  1647540.534
43            36.8  2008085.819
57            21.3  2029039.189
77            16.7  1803135.673
66            38.0  2721155.179
68            31.0  3110265.800
```

Fig 20: Vehicles with best Acceleration under INR 40,00000

Vehicles with Maximum Efficiency

```
In [16]: pd.set_option('display.max_columns', None)
         efficiency = df.sort_values(by = 'Efficiency_WhKm')
         print(efficiency[['CarName', 'Efficiency_WhKm', 'Range_Km', 'PowerTrain', 'Battery_Pack Kwh', 'INR']])
```

```
                          CarName  Efficiency_WhKm  Range_Km PowerTrain  \
48                  Lightyear-One              104       575        AWD
14          Hyundai-IONIQ Electric             153       250        FWD
8    Tesla-Model 3 Standard Range Plus         153       310        RWD
41       Hyundai-Kona Electric 39 kWh          154       255        FWD
74                      Sono-Sion              156       225        FWD
..                            ...              ...       ...        ...
98   Audi-e-tron S Sportback 55 quattro        258       335        AWD
67       Tesla-Cybertruck Dual Motor           261       460        AWD
33        Tesla-Cybertruck Tri Motor           267       750        AWD
90          Audi-e-tron S 55 quattro           270       320        AWD
84            Mercedes-EQV 300 Long            273       330        FWD

    Battery_Pack Kwh           INR
48              60.0  1.219552e+07
14              38.3  2.820438e+06
8               50.0  3.796161e+06
41              39.0  2.780496e+06
74              35.0  2.087152e+06
..               ...           ...
98              86.5  7.861606e+06
67             120.0  4.501700e+06
33             200.0  6.138682e+06
90              86.5  7.677446e+06
84              90.0  5.781084e+06

[102 rows x 6 columns]
```

Fig 21: Vehicles with Maximum Efficiency

5. Strategic Pricing

Analysis of the range and price of EVs provided insights into setting a strategic pricing range for products. This included comparing mid-range and high-range vehicles and visualizing their price and range.

Reading the Data

Budget wise EV Car Analysis

In [57]:
```
df1=pd.read_csv(r"C:\Users\SWAYAM\Downloads\EVIndia (1).csv")
PriceRange = (df1['PriceRange'].astype(str))
df1
```

Out[57]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel	Unnamed: 10
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	939950.0	5 Seater	350 L	XM	Dark XZ Plus LUX	NaN
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	Automatic	Electric	1306500.0	5 Seater	316 L	XE	XZ Plus Dual Tone	NaN
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	Automatic	Electric	1306500.0	5 Seater	350 L	XZ Plus 3.3 kW	XZ Plus Lux 7.2 kW	NaN
3	MG ZS EV	Compact SUV	419 Km/Full Charge	Automatic	Electric	2393500.0	5 Seater	448 L	Excite	Exclusive	NaN
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	Automatic	Electric	2388500.0	5 Seater	na	Premium Dual Tone	HSE	NaN
5	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	10900000.0	5 Seater	656 L	S	Sportback 55	NaN
6	Audi E-Tron GT	Premium Coupe	388 Km/Full Charge	Automatic	Electric	18000000.0	5 Seater	405 L	Quattro	na	NaN
7	BYD E6	Subcompact MPV	415 Km/Full Charge	Automatic	Electric	2915000.0	5 Seater	580 L	STD	na	NaN
8	Mercedes-Benz EQC	Compact SUV	471 Km/Full Charge	Automatic	Electric	10000000.0	5 Seater	na	na	na	NaN
9	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	11600000.0	5 Seater	na	na	na	NaN

Fig 22: EV cars data in India

Analysing the data

In [58]:
```
mid_range_cars= df1.loc[df1['PriceRange'] <=3000000]
high_range_cars= df1.loc[df1['PriceRange'] >3000000]
s1 = ['Less than INR 3000000']
s2 = ['More than INR 3000000']
```

In [68]:
```
mid_range_cars
```

Out[68]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel	Unnamed: 10
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	939950.0	5 Seater	350 L	XM	Dark XZ Plus LUX	NaN
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	Automatic	Electric	1306500.0	5 Seater	316 L	XE	XZ Plus Dual Tone	NaN
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	Automatic	Electric	1306500.0	5 Seater	350 L	XZ Plus 3.3 kW	XZ Plus Lux 7.2 kW	NaN
3	MG ZS EV	Compact SUV	419 Km/Full Charge	Automatic	Electric	2393500.0	5 Seater	448 L	Excite	Exclusive	NaN
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	Automatic	Electric	2388500.0	5 Seater	na	Premium Dual Tone	HSE	NaN
7	BYD E6	Subcompact MPV	415 Km/Full Charge	Automatic	Electric	2915000.0	5 Seater	580 L	STD	na	NaN

In [69]:
```
high_range_cars
```

Out[69]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel	Unnamed: 10
5	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	10900000.0	5 Seater	656 L	S	Sportback 55	NaN
6	Audi E-Tron GT	Premium Coupe	388 Km/Full Charge	Automatic	Electric	18000000.0	5 Seater	405 L	Quattro	na	NaN
8	Mercedes-Benz EQC	Compact SUV	471 Km/Full Charge	Automatic	Electric	10000000.0	5 Seater	na	na	na	NaN
9	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	11600000.0	5 Seater	na	na	na	NaN
10	Porsche Taycan	Premium Sports Sedan	na	Automatic	Electric	15000000.0	4 Seater	na	na	na	NaN

Fig 23: Creating segments of high range and low-mid range cars

mid-range vehicles with max range

```
In [59]: pd.set_option('display.max_columns', None)
max_range = mid_range_cars.sort_values(by= 'Range')
print(max_range[['Car', 'Style', 'Range', 'PriceRange', 'BootSpace']])
```

```
                    Car               Style                Range  PriceRange  \
1         Tata Tigor EV  Subcompact Sedan  306 Km/Full Charge   1306500.0
0         Tata Nexon EV       Compact SUV  312 Km/Full Charge    939950.0
7                BYD E6    Subcompact MPV  415 Km/Full Charge   2915000.0
3              MG ZS EV       Compact SUV  419 Km/Full Charge   2393500.0
2     Tata Nexon EV Max       Compact SUV  437 Km/Full Charge   1306500.0
4  Hyundai Kona Electric       Compact SUV  452 Km/Full Charge   2388500.0

  BootSpace
1     316 L
0     350 L
7     580 L
3     448 L
2     350 L
4        na
```

Fig 24: Mid-range vehicles(mid-range price) with max range(Km/Full)

Visualizing Price - Range

```
In [67]: def pricerange(dataframe, text):
    plt.figure(figsize=(20,5))
    a_1 = plt.subplot()
    a_1.bar(dataframe['Car'], dataframe['Range'], label='Range (km/h)', color='green')
    plt.legend(loc = 'upper left', bbox_to_anchor = (0,1.1))
    a_2 = a_1.twinx()
    a_2.scatter(dataframe['Car'], dataframe['PriceRange'], label = 'Price', color='black')
    plt.title('''Range (km/hr) vs Price of EV's costing {}'''.format(text), fontsize = 16)
    a_1.set_xlabel('Car')
    a_1.set_ylabel('Range')
    a_2.set_ylabel('Price')
    plt.legend(loc= 'upper left', bbox_to_anchor = (0,1))
    a_1.set_xticklabels(mid_range_cars['Car'], rotation = 'vertical')
    plt.show()

pricerange(mid_range_cars,s1)
pricerange(high_range_cars,s2)
```

```
C:\Users\SWAYAM\AppData\Local\Temp/ipykernel_6668/4044561395.py:13: UserWarning: FixedFormatter should only be used togeth
er with FixedLocator
  a_1.set_xticklabels(mid_range_cars['Car'], rotation = 'vertical')
```

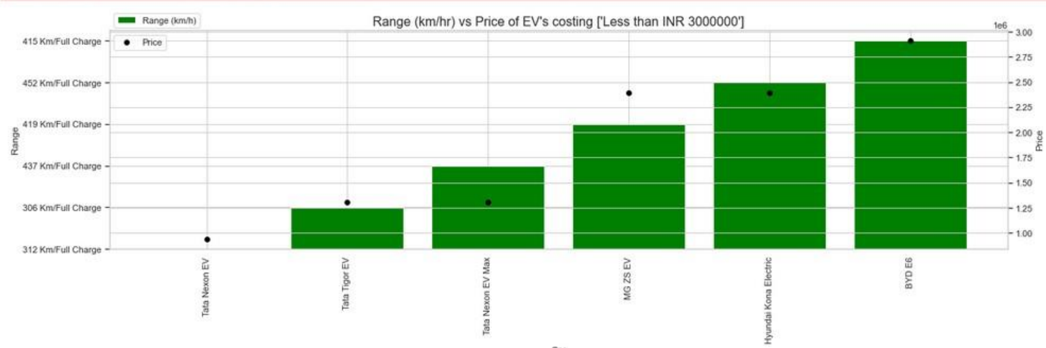

Fig 25: Barplot of Range vs Price of Mid-range cars

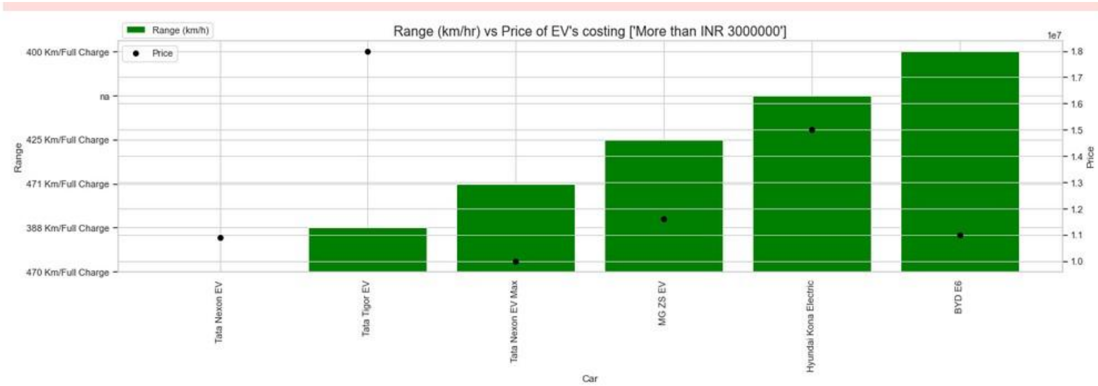

Fig 26: Barplot of Range vs Price of High-range cars

Factors Affecting an EV Startup in India

When starting an electric vehicle (EV) company in India, various factors can significantly impact the business. To analyze these factors, we have segmented the data state-wise. Our report considers the following key factors:

1. **Percentage of Tax Exemption** given by the respective State/UT
2. **Subsidy Amount (in INR)** provided by the respective State/UT
3. **Fuel Prices (Petrol and Diesel)** in the respective State/UT
4. **Pollution/Air Quality** of the respective State/UT

An EV company can strategically decide the location of their showrooms based on these factors:

- **Maximum Tax Exemption and Subsidy**: States or UTs offering high tax exemptions and subsidies are attractive as they provide financial benefits, aiding in business growth.
- **High Fuel Prices**: Regions with high petrol and diesel prices may have consumers looking for cost-effective alternatives, making EVs an appealing option.
- **Poor Air Quality**: Areas with poor air quality may have residents more inclined to switch to EVs to reduce pollution, benefiting both the company and the environment.

To identify the most advantageous regions for an EV startup, we prepared datasets based on the aforementioned factors. While the information is not 100% accurate, maximum care has been taken to ensure its reliability and error-free status.

6. Challenges and Opportunities

The analysis highlighted several challenges and opportunities for an EV startup in India, including:

1. Tax exemptions and subsidies provided by various states.

2. The correlation between fuel prices and the adoption of EVs.

3. The environmental benefits of reducing pollution through increased EV usage.

7. Conclusion

This report provides a detailed analysis of the EV market in India, considering various segments and data points. It offers insights into the most suitable locations, target demographics, and strategic pricing for entering the market. The analysis supports the development of a feasible strategy for an EV startup to thrive in the Indian market.