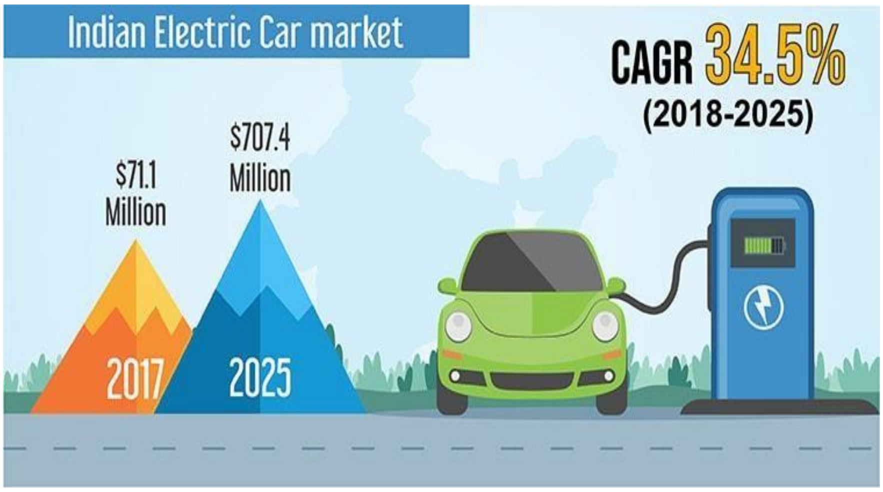
ELECTRIC VEHICLES MARKET



Team Mentor

Sanjay Basumatary

Team Members

1. Sahil Jethva(Team Leader)
2. B.Sriharsha
3. Heeta Parmar
4. Thandava Krishna

INDEX

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 self-discharge 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:

Lower running costs

The running cost of an electric vehicle is much lower than an equivalent petrol or diesel vehicle. Electric vehicles use electricity to charge their batteries instead of using fossil fuels like petrol or diesel. Electric vehicles are more efficient, and that combined with the electricity cost means that charging an electric vehicle is cheaper than filling petrol or diesel for your travel requirements. Using renewable energy sources can make the use of electric vehicles more eco-friendly. The electricity cost can be reduced further if charging is done with the help of renewable energy sources installed at home, such as solar panels.

Low maintenance cost

Electric vehicles have very low maintenance costs because they don't have as many moving parts as an internal combustion vehicle. The servicing requirements for electric vehicles are lesser than the conventional petrol or diesel vehicles. Therefore, the yearly cost of running an electric vehicle is significantly low.

Zero Tailpipe Emissions

Driving an electric vehicle can help you reduce your carbon footprint because there will be zero tailpipe emissions. You can reduce the environmental impact of charging your vehicle further by choosing renewable energy options for home electricity.

Tax and financial benefits

Registration fees and road tax on purchasing electric vehicles are lesser than petrol or diesel vehicles. There are multiple policies and incentives offered by the government depending on which state you are in.

Creates very little noise

The electric vehicles run at almost no noise hence decreasing the sound pollution and environmentally friendly.

No exhaust, spark plugs

No exhaust, hence no air, sound pollution; as it runs on electrical energy, there is no need of any spark plug.

Data Collection

1. [Kaggle]
2. [FirstPost]
3. [JmkResearch]

Segmentation Criteria

The term segmentation Criteria relates to the nature or the type of information used for market segmentation, unlike the segmentation variable which means the variable in empirical data in common sense segmentation for splitting the sample into market segments. In Segmentation we usually find the identifiable characteristics of individuals in the data sample and segmenting them into a same cluster and analyse the common interest needs to maximize the organizations profits. Segmentation Criteria is an important factor in market segmentation as well. The four main types in segmentation criteria are Geographic Segmentation, socio-demographic segmentation, psychographic segmentation and behavioural segmentation.

Geographic Segmentation

In Geographic Segmentation the key criteria to form market segments is the geographic location or the residence of the customer. There are some specific advantages of doing geographic segmentation, they are, we can segment down all the customers in that particular area, do promotions which are meaningful in that area and even run adds in news-papers, television, etc. in that local area. The only key disadvantage is that it not always the case that all the people residing in the same location will have same opinions and preferences in the products.

Socio-Demographic Segmentation

Socio-Demographic Segmentation criteria includes parameters like age, gender, education, income, etc. For ex, while buying cosmetics criteria associated is gender, while buying branded and luxury items criteria associated is income, while planning on vacation destination criteria associated is age (i.e., if people go in couple the vacation destination will be different if people going with children, then the vacation destination is different). The socio-demographic segmentation at times with better data can give us the better market segments and gives us the clear clarity on the who the customer is, this is achievable provided better data that provides sufficient insights about who the customer is and the market segments. But in many cases, socio-demographic segmentation would not be the best fit for product preferences.

Psychographic Segmentation

For making market segments using the Psychographic segmentation the criteria is the Psychological criteria for grouping people. Parameters like interest, beliefs, aspirations, preferences, benefits, etc. can be used to define psychological criteria. Psychographic segmentation is more complex by nature compared to Geographic Segmentation and Socio-Demographic segmentation because, we cannot find a single fixed parameter for insights for better segmentation, there are a lot of factors effecting the psychographic criteria and the factors are different in each person. Therefore, we must use a lot of segmentation variables. And the main advantage that psychographic segmentation has is that clustering a common set of customers based on psychographic criteria for maximizing profits. For ex, people who want to go on a vacation and has a preference for attending historic pilgrims can be clustered and can be taken together which can reduce cost for company and maximize the profit as well.

Behavioural Segmentation

In Behavioural segmentation we can directly find similarities in behaviours of customers. There can be many useful implementations possible for doing market segments. Behavioural segmentation criteria depend on the way visitors interact with the website. Some data depends on their immediate online behaviour and giving positive feedback while other data depends on their past offline behaviour or negative feedback.

Pre-Processing Data before performing Segmentation

1. **Categorical Variables**
   Two pre-processing procedures are often used for categorical variables. One is merging levels of categorical variables before further analysis, the other one is converting categorical variables to numeric ones, if it makes sense to do so. Merging levels of categorical variables is useful if the original categories are too differentiated (too many).

2. **Numerical Variables**
   In distance-based methods of segment extraction, the range of values of a segmentation variable determines its relative influence. If one of the segmentation variables is binary (with values 0 or 1 indicating whether or not a customer views on the product of fast food), and a second variable indicates the expenditure in dollars per person per day (with values ranging from zero to $1000), a one-dollar difference in spend per person per day is weighted equally as the difference in liking to dine out or not.

3. **Univariate Variables**
We take one feature and based on that we will try to classify what the output is going to be. In McDonald's dataset, we took age as feature and classified based how much they are liked. From our data all the persons who gave positive feedback '4' and above their age is around '20' and the data are fit (overlapped) one guy from age.

4. **Bivariate Variables**
Bivariate analysis is slightly more analytical than Univariate analysis. When the data set contains two variables and researchers aim to undertake comparisons between the two data set then Bivariate analysis is the right type of analysis technique.

5. **Multivariate Variables**
Multivariate analysis is a more complex form of statistical analysis technique and used when there are more than two variables in the data set. Here we can apply PCA to reduce the dimensions.

Electric Vehicles Sales trends

Fig. 1: FY2021 Quarterly Sales Trend – Registered EVs

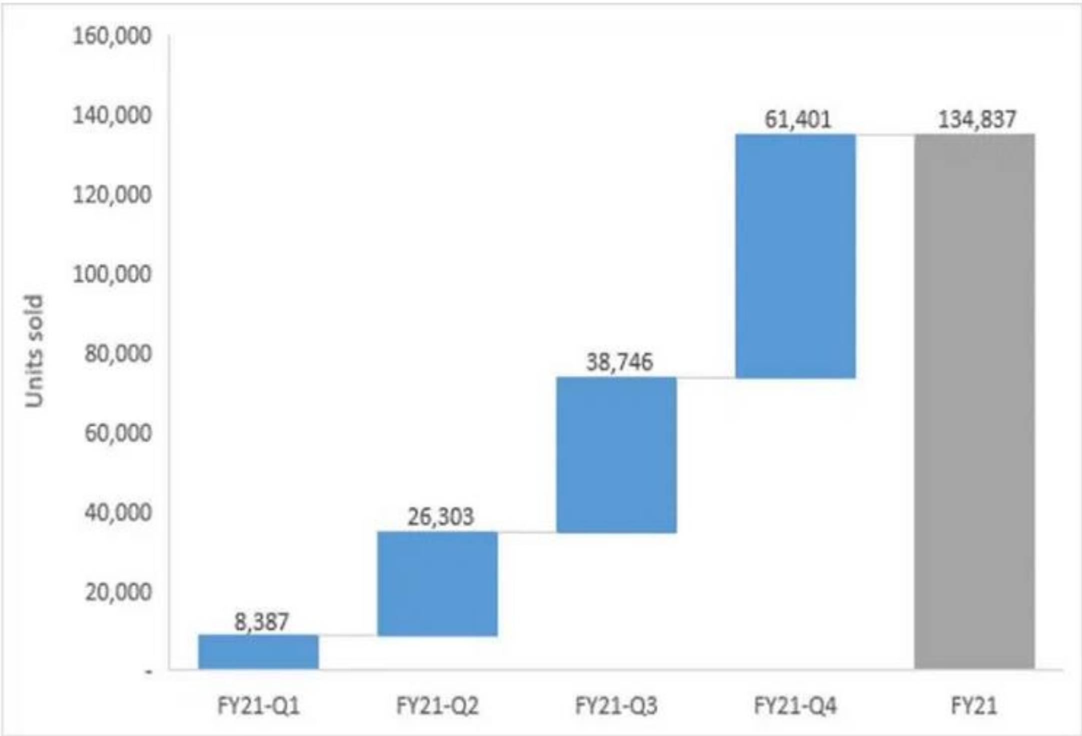
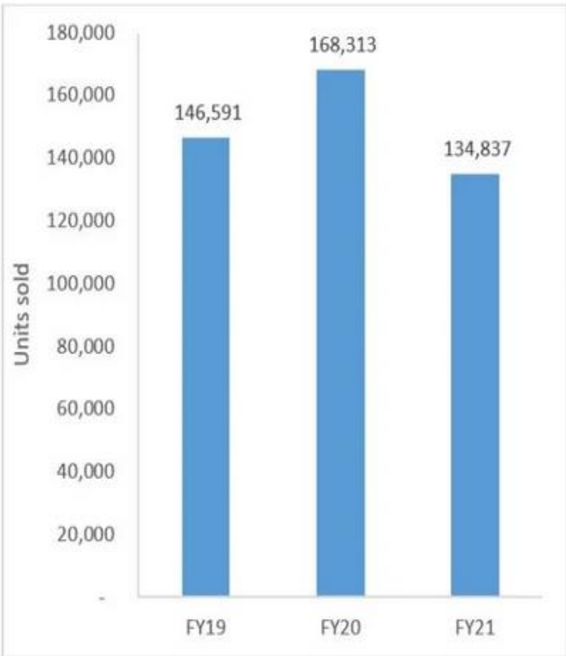
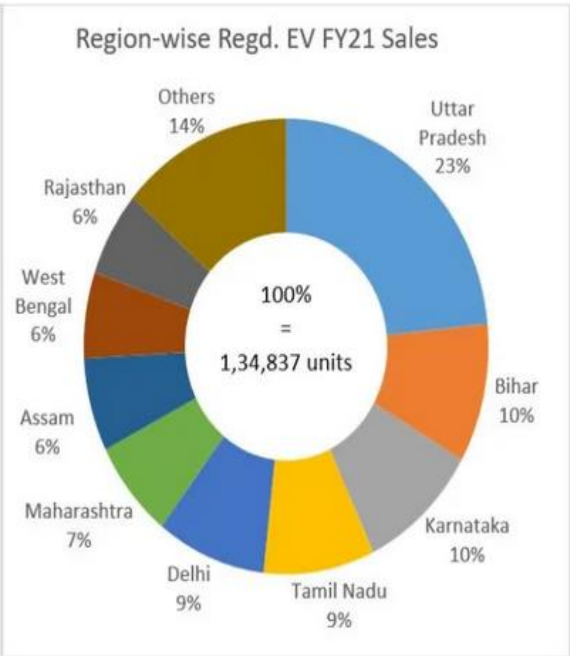


Fig. 2: FY Sales Trend – Registered EVs





Code Implementation

Importing Necessary Libraries

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

Fig 1: Importing Libraries for Code Implementation

1. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.
2. Pandas is a library written for the Python programming language for data manipulation and analysis
3. Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays.
4. Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis.
5. Warnings are provided to warn the developer of situations that aren't necessarily exceptions

Reading Data

```
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

Fig 2: Dataset used for Code Implementation

Analysing 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

```
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)
```

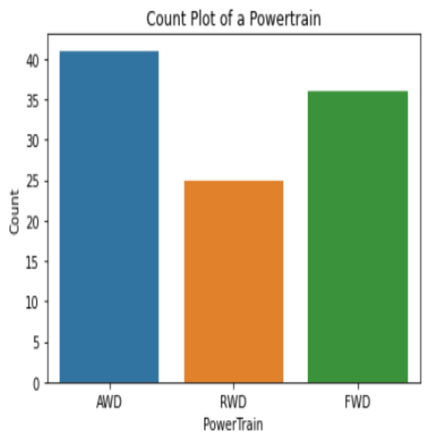


Fig 8: Count Plot of a Powertrain

```
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)
```

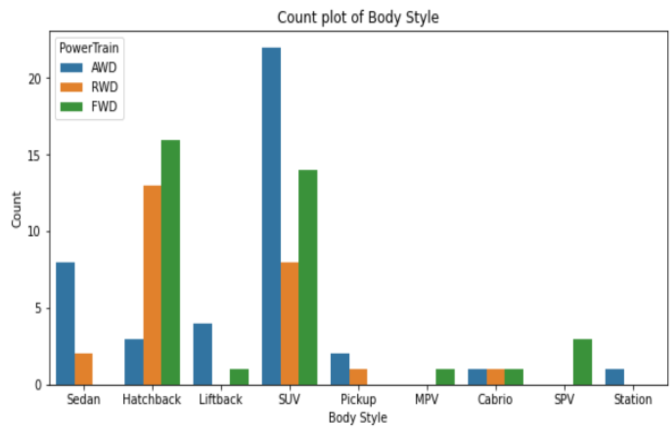


Fig 9: Count plot of body Style of the cars

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(
```

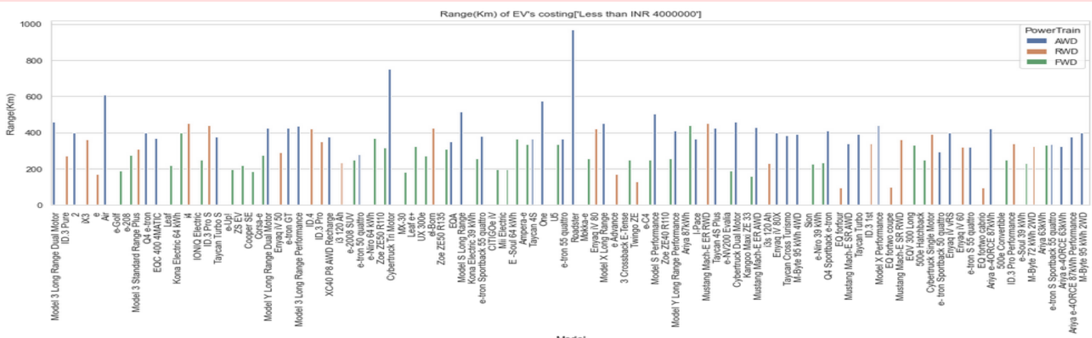


Fig 10: Bar graph of Range of EV’s

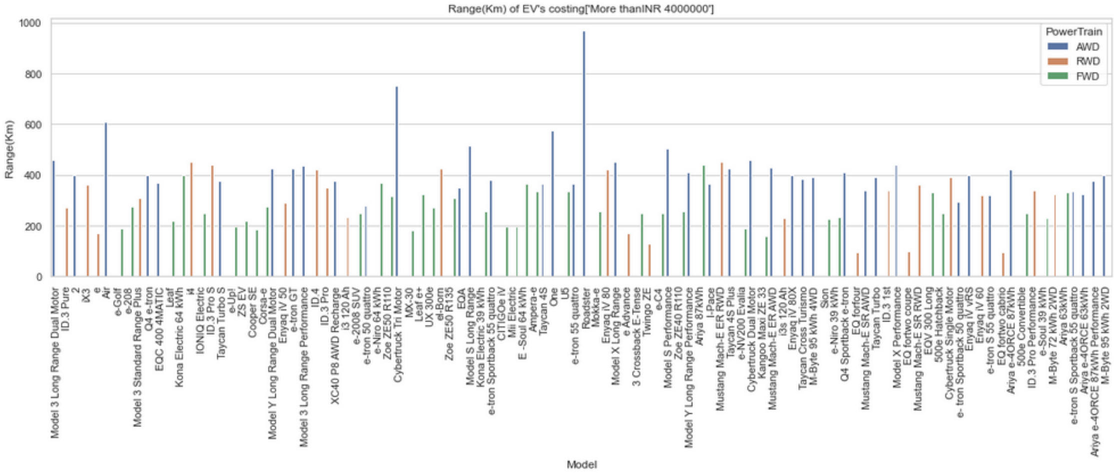


Fig 11: Bar graph of Range of EV's

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')
```

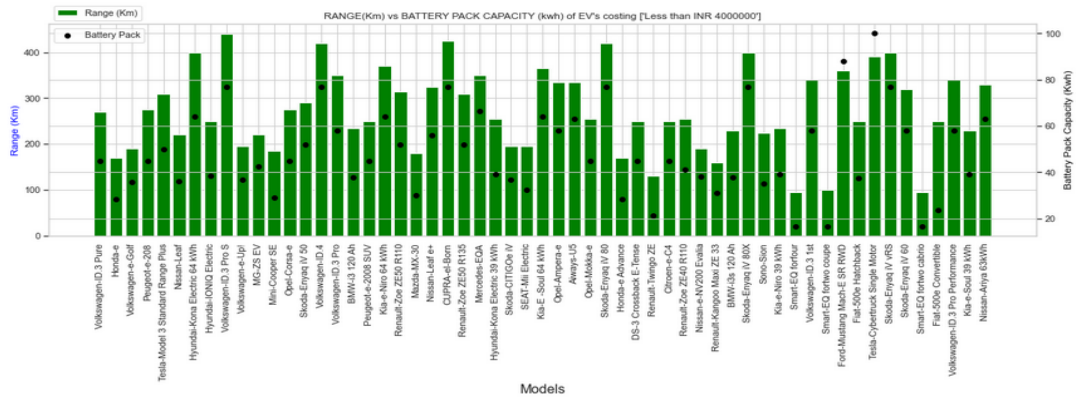


Fig 12: Bar graph of Range vs Battery Capacity of EV's

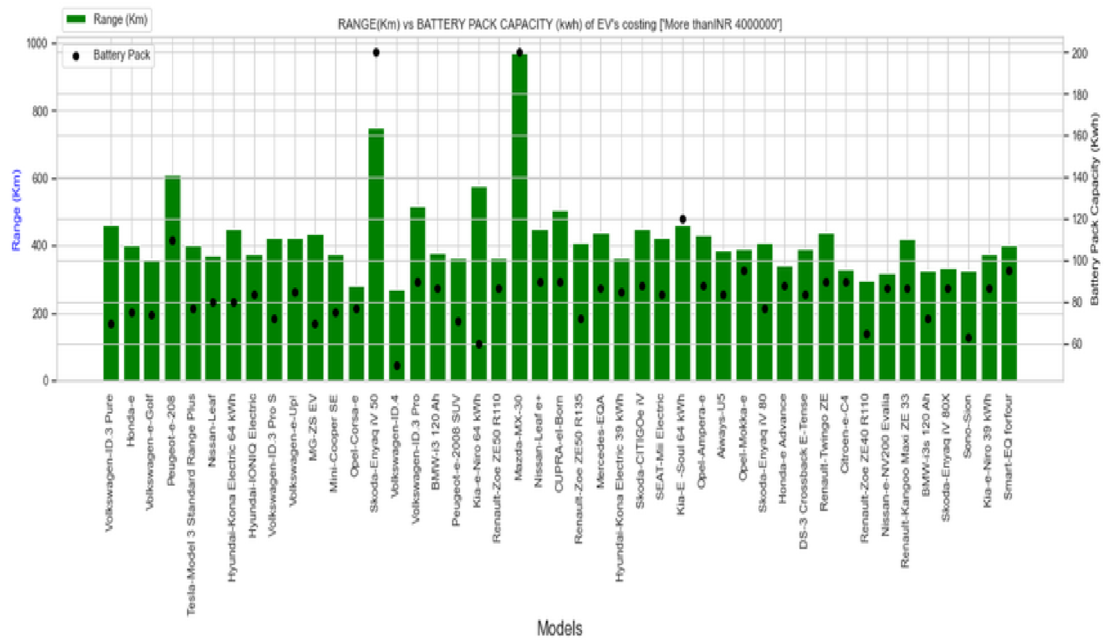


Fig 13: Bar graph of Range vs Battery Capacity of EV's

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')
```

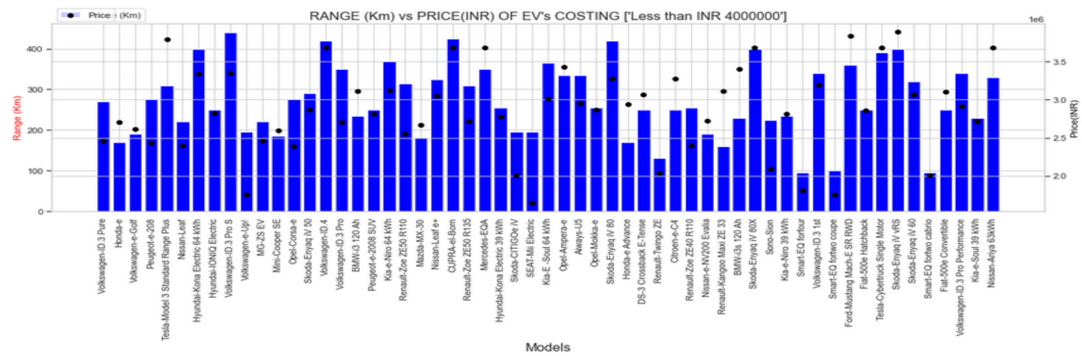


Fig 14: Bar graph of Range vs Price of EV's

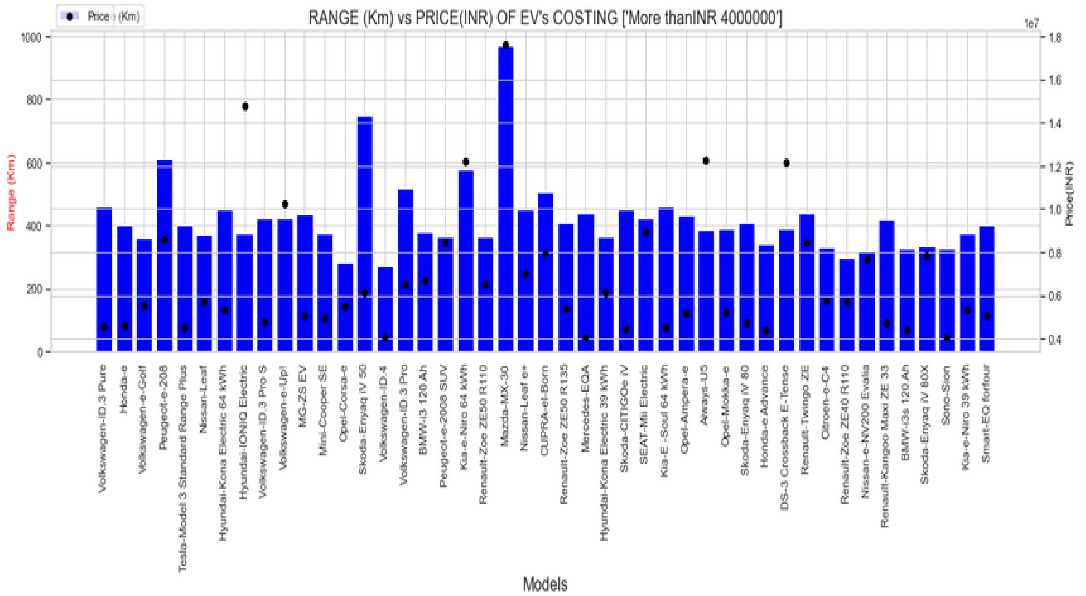


Fig 15: Bar graph of Range vs Price of EV's

Acceleration(0-100km/hr)

In [12]:
```python
#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(
```

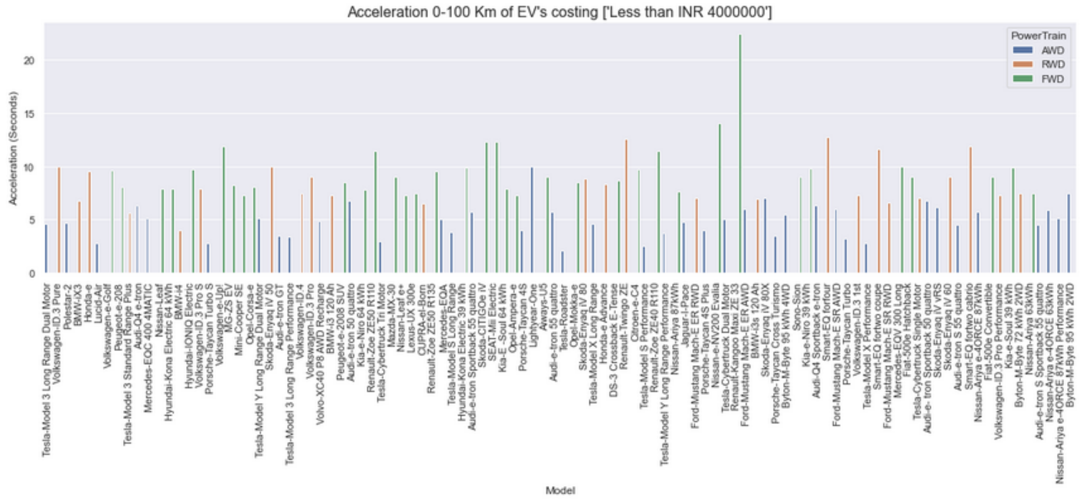


Fig 16: Bar graph of Acceleration vs Price of EV's

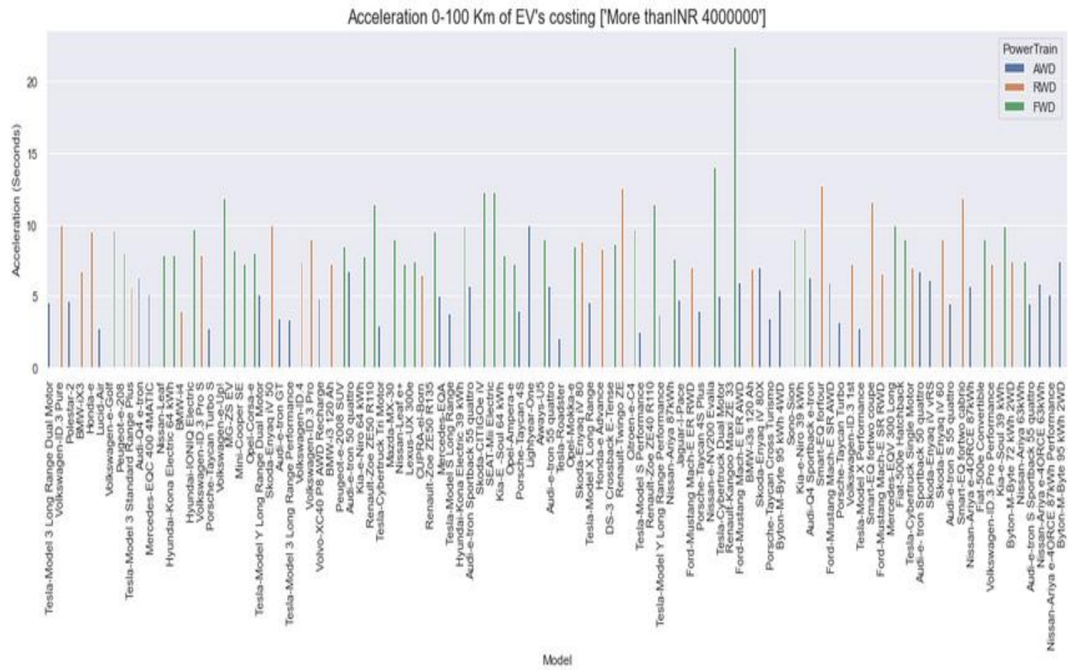


Fig 17: Bar graph of Acceleration vs Price of EV's

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)
```

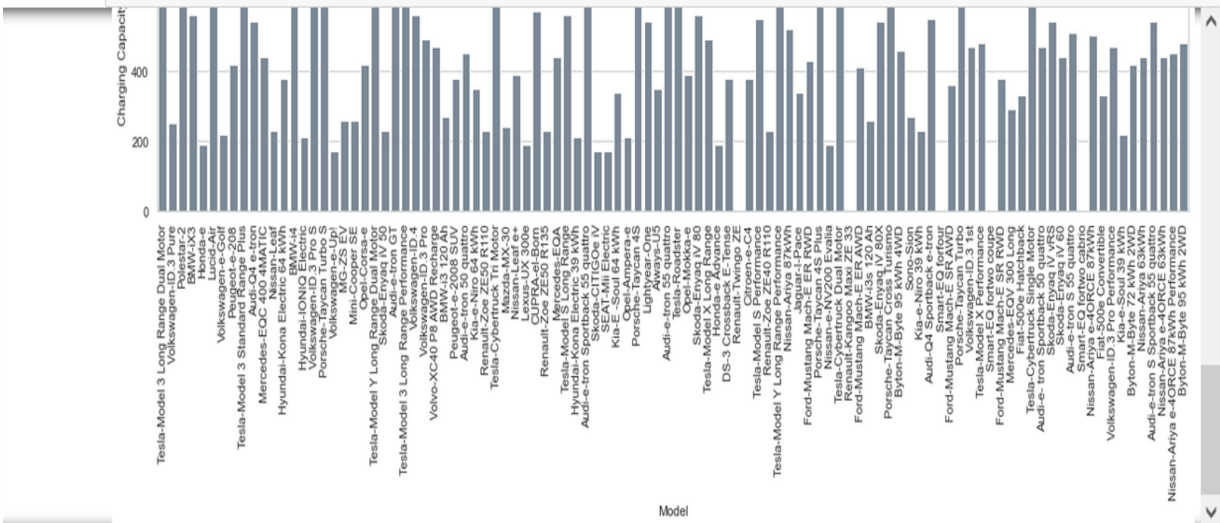


Fig 18: Bar graph of Fast Charging ability of EV's

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

Budget wise EV cars analysis

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')
```

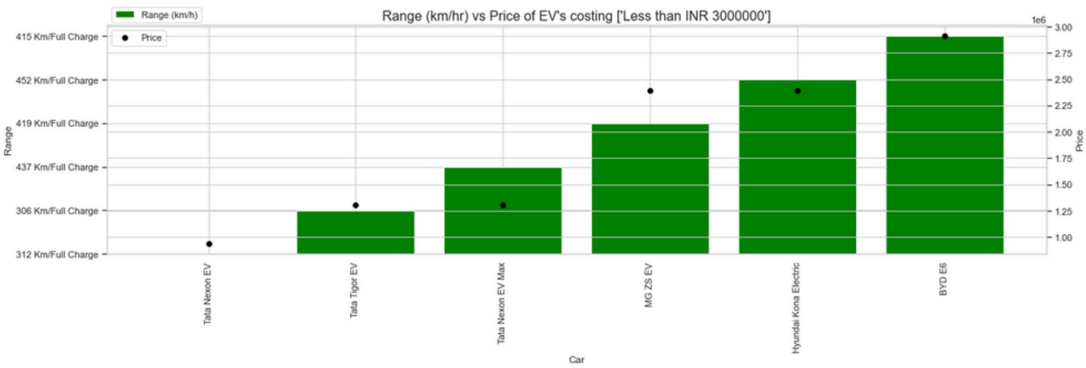


Fig 25: Barplot of Range vs Price of Mid-range cars

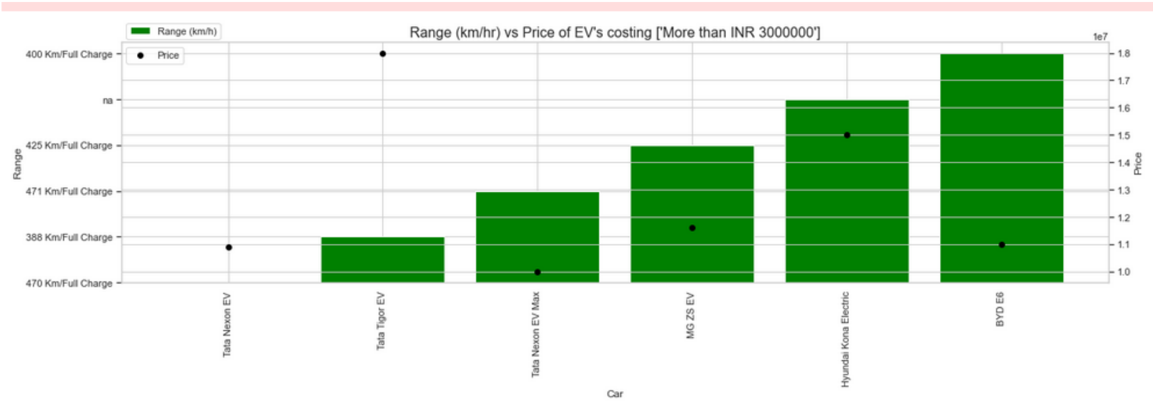


Fig 26: Barplot of Range vs Price of High-range cars

Factors Affecting an EV start up in India

For an EV start up there are some other factors which may affect its business. To analyse these factors we have divided our segments state wise. Some of the factors considered in our report are:

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

An EV company can put up their showroom in the region where the state is giving maximum Tax Exemption and Subsidy as this would be helpful in business point of view. It can also put up their showroom where the fuel prices are high as people in those states/UT's would be looking for another alternative than paying huge prices for the fuel. In environment point of view an EV company start their business in the region whose air quality is not good or poor, people over there would be also willing to decrease the pollution rate by switching their means of transport from fuel to electric, This would be helpful for both company and the environment.

Based on these factors and their dependencies some datasets are prepared manually to analyse which region would be helpful for an EV start up in India. The information present there is not 100% accurate, but maximum care has been taken for the information to be error free.

State wise tax relaxation, subsidy and fuel prices analysis

Importing Necessary libraries

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

Fig 27: Importing Libraries for Code Implementation

Reading the Data

```
In [7]: data=pd.read_csv("states_data_car.csv")
        data.head(20)
```

Out[7]:

	state	capital	subsidy	road tax	petrol	diesel
0	Andhra Pradesh	Amaravati	0.0	1.00	111.65	99.41
1	Arunachal Pradesh	Itanagar	5000.0	0.00	95.89	84.81
2	Assam	Dispur	10000.0	1.00	96.34	84.24
3	Bihar	Patna	10000.0	1.00	109.17	95.82
4	Chhattisgarh	Raipur	5000.0	0.00	102.98	95.96
5	Goa	Panaji	8000.0	1.00	97.82	90.37
6	Gujarat	Gandhinagar	10000.0	0.50	96.49	92.23
7	Haryana	Chandigarh	0.0	0.00	97.24	90.08
8	Himachal Pradesh	Shimla	5000.0	0.00	95.74	81.99
9	Jharkhand	Ranchi	5000.0	0.00	100.09	94.88
10	Karnataka	Bengaluru	0.0	1.00	102.64	88.55
11	Kerala	Thiruvananthapuram	0.0	0.50	106.45	95.34
12	Madhya Pradesh	Bhopal	0.0	0.99	110.02	95.18
13	Maharashtra	Mumbai	5000.0	1.00	111.18	95.66
14	Manipur	Imphal	5000.0	0.00	101.22	87.16
15	Meghalaya	Shillong	4000.0	1.00	95.06	83.28
16	Mizoram	Aizawl	0.0	0.00	95.72	82.17
17	Nagaland	Kohima	5000.0	0.00	98.28	86.65
18	Odisha	Bhubaneswar	0.0	1.00	104.45	95.97
19	Punjab	Chandigarh	0.0	1.00	96.26	86.63

Fig 28: Data set used for code implementation

Analysing the data

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   state     36 non-null     object
 1   capital   36 non-null     object
 2   subsidy   36 non-null     float64
 3   road tax  36 non-null     float64
 4   petrol    36 non-null     float64
 5   diesel    36 non-null     float64
dtypes: float64(4), object(2)
memory usage: 1.8+ KB
```

In [3]:
```
data.isnull().sum()
```

Out[3]:
```
state       0
capital     0
subsidy     0
road tax    0
petrol      0
diesel      0
dtype: int64
```

Fig 29: Information about the data and checking for any null values in it

Visualization

In [5]:
```
sns.countplot(x=data["subsidy"])
plt.title('Count plot of subsidy')
plt.xlabel('Subsidy')
plt.ylabel('Count')
plt.show()
```

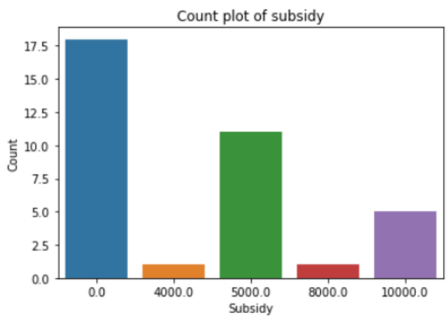


Fig 30: Count plot of subsidy

In [6]:
```
sns.countplot(x=data["road tax"])
plt.title('Count plot of road tax')
plt.xlabel('road tax')
plt.ylabel('Count')
plt.show()
```

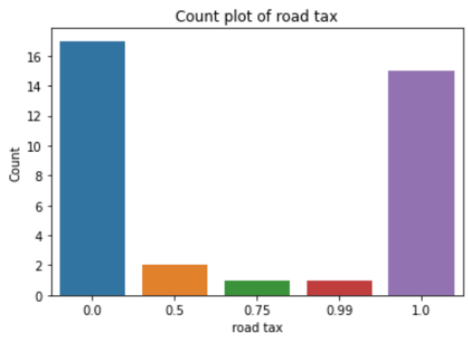


Fig 31: count plot of road tax

```
In [14]: plt.figure(figsize=(20, 5))
sns.set_theme(style="whitegrid")
sns.barplot('state', 'petrol', data=data, color = 'lightslategrey')
plt.title('Petrol Price in Different States' , fontsize = 16)
plt.ylabel('Petrol Price')
plt.xlabel('State Name')
plt.xticks(rotation=90)
plt.show()
```

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

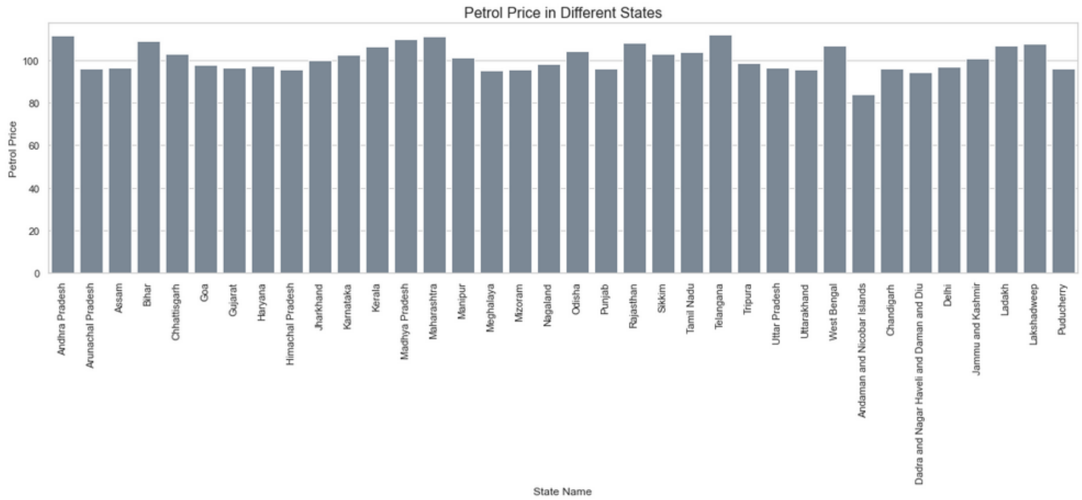


Fig 32: Bar plot of petrol prices in different states

```
In [16]: plt.figure(figsize=(20, 6))
sns.set_theme(style="whitegrid")
sns.barplot('state', 'diesel', data=data, color = 'lightslategrey')
plt.title('Diesel Price at different states' , fontsize = 16)
plt.ylabel('Diesel Price')
plt.xlabel('State Name')
plt.xticks(rotation=90)
plt.show()
```

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

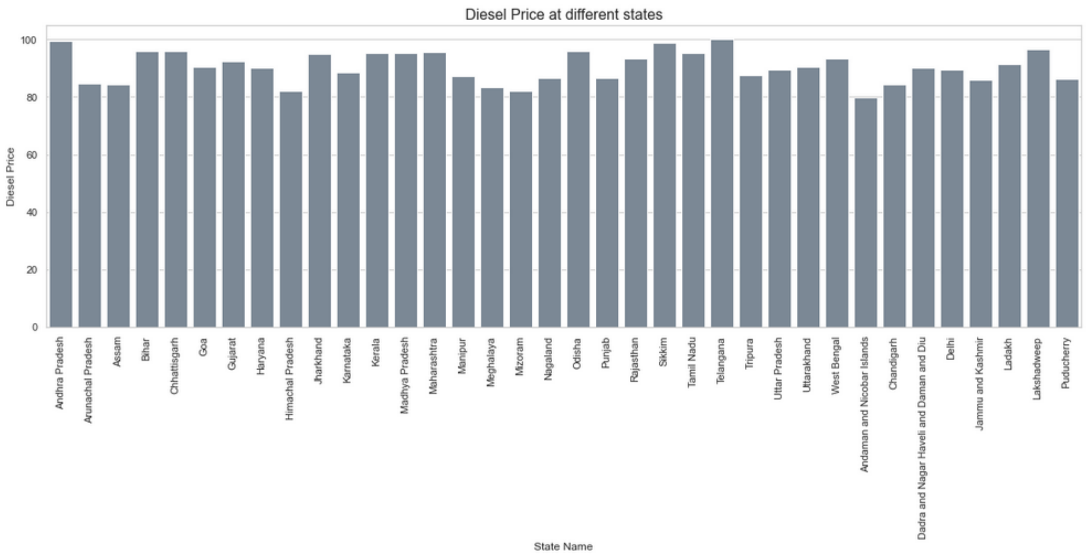


Fig 33: Bar plot of Diesel prices in different states

State wise Pollution data analysis

Importing necessary libraries

```
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

Fig 35: Data set used for code implementation

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

Analysing the data

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

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

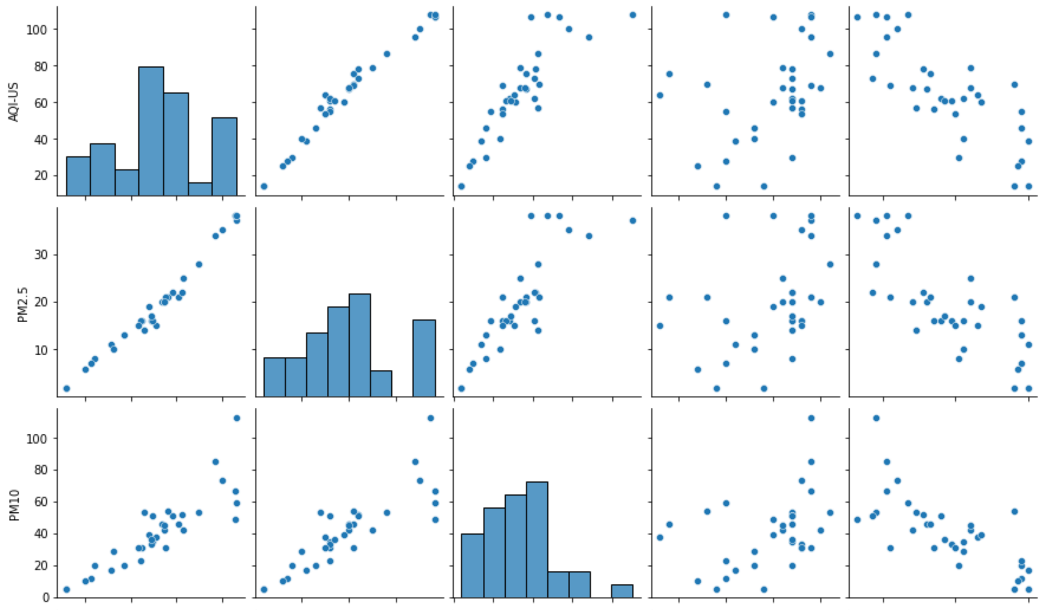


Fig 37: Pairplot of the data present in the dataset

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

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

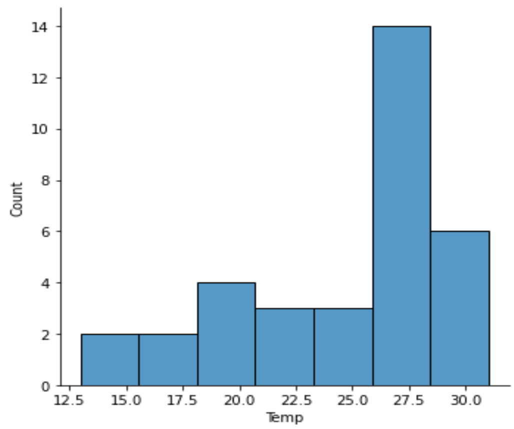


Fig 38: Displot of State wise Temperature data

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

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

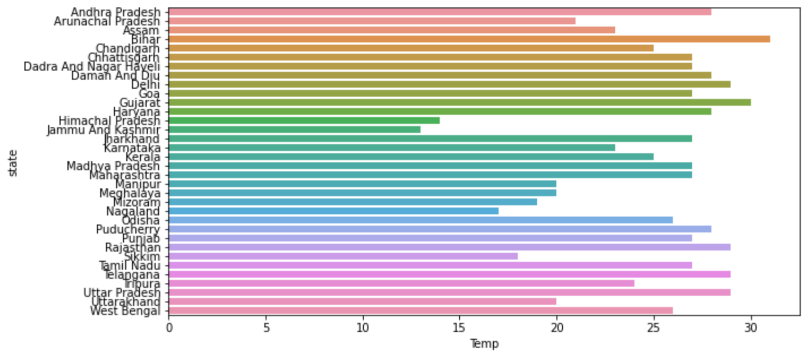


Fig 39: Barplot of State wise Temperature data

```
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>
```

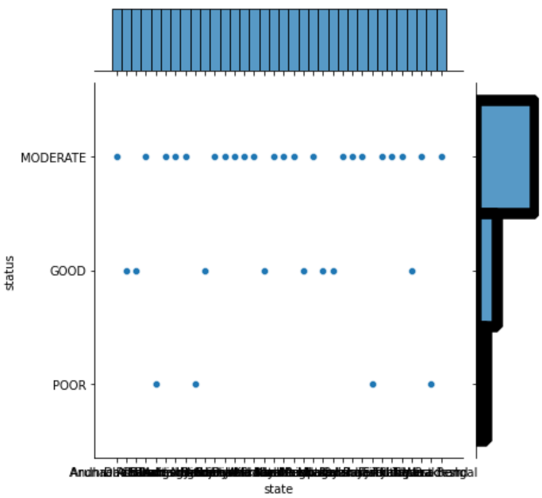


Fig 40: Jointplot of State wise air quality

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

```
Out[52]: <AxesSubplot:>
```

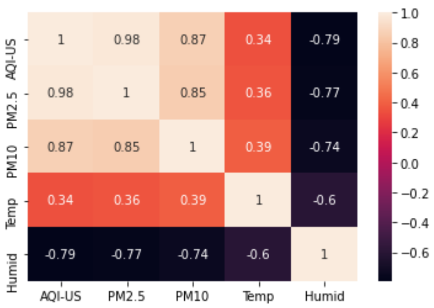


Fig 41: Heatmap of the data present in the dataset

Conclusion

Based on the above analysis and visualizations, it would be really helpful for any company which is looking to open up an EV start up in India. In this report, 4 wheeler EV's are more concentrated, the customer space has been visualized in a detailed manner to understand the trends and move accordingly.

Click here for code implementation in GitHub