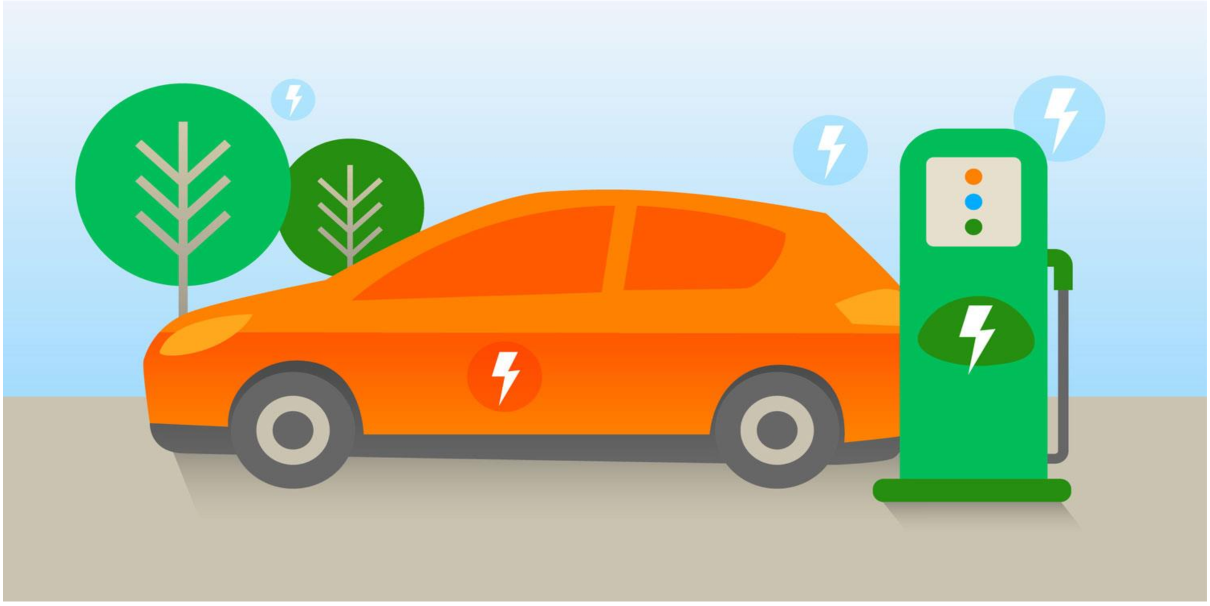
Electric Vehicle Market

In 1888, the German Andreas Flocken designed the Flocken Elektrowagen, regarded by some as the first "real" electric car.

Since then, a lot of improvement has been made in the field of automobiles even more so in electric vehicles.

The electric vehicle market is currently a lucrative goal for companies and start-ups in India, several obstacles still remain to be addressed in order for EVs to be ready for mass adoption. Battery packs are assured for several years, whilst the service networks are fitted with technicians who are specially qualified to welcome and manage an EV. Wall charging units are often built at the chosen customers' location by several producers without any extra expense.



NAME : PRABHRAJ SINGH

Problem Statement (EV Market)

You are a team working under an Electric Vehicle Startup. The Startup is still deciding in which vehicle/customer space it will be develop its EVs.

You 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

Introduction

The electric vehicle (EV) market in India has been growing rapidly due to increasing environmental awareness, government incentives, and advancements in technology. This report provides an analysis of various aspects of the EV market, including vehicle specifications, price distributions, powertrain types, body styles, range, battery pack capacity, fast charging capabilities, and vehicle acceleration.

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
- Low maintenance cost
- Zero Tailpipe Emissions
- Tax and financial benefits
- Creates very little noise
- No exhaust, spark plugs

Data Collection and Analysis

- https://www.kaggle.com/datasets/
- https://data.gov.in/
- https://www.data.gov/
- https://data.worldbank.org/
- https://datasetsearch.research.google.com/

Market Segmentation

Target Market:

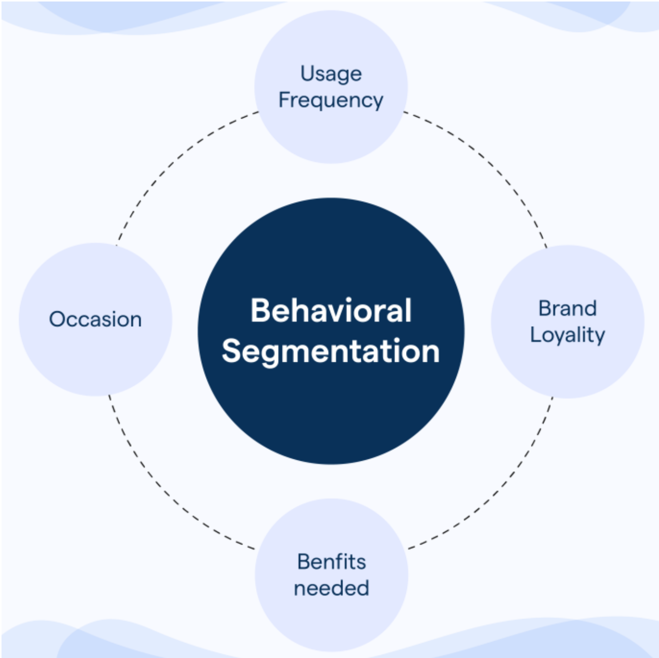
The target market of Electric Vehicle Market Segmentation can be categorized into

- Geographic,
- SocioDemographic,
- Behavioral, and
- Psychographic Segmentation

Behavioral Segmentation:

searches directly for similarities in behavior or reported behavior.

Example: prior experience with the product, amount spent on the purchase

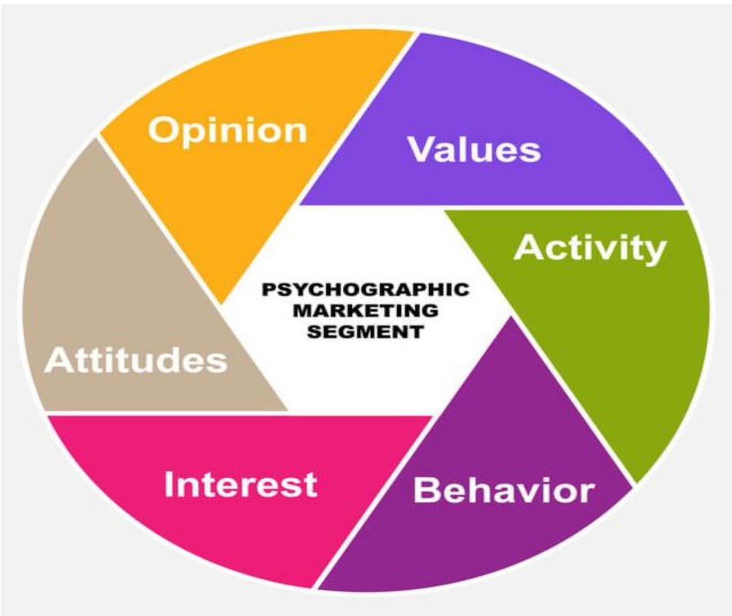


Advantage: uses the behavior of interest is used as the basis of segment extraction.

Disadvantage: not always readily available.

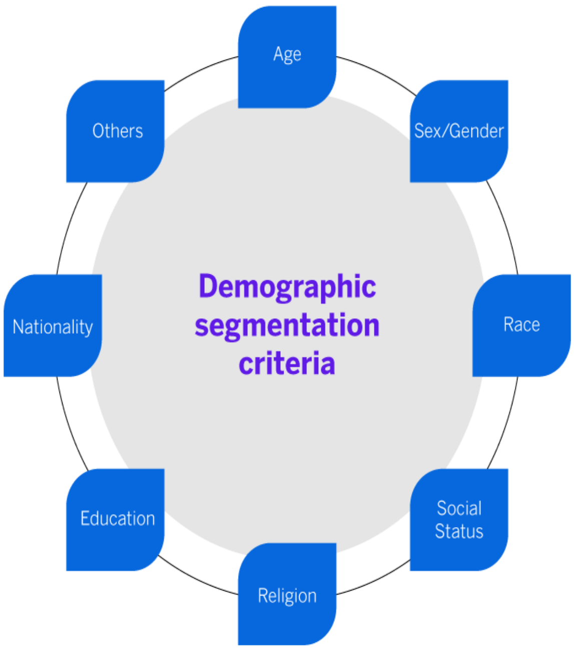
Psychographic Segmentation:

grouped based on beliefs, interests, preferences, aspirations, or benefits sought when purchasing a product. Suitable for lifestyle segmentation. Involves many segmentation variables.



Advantage: generally, more reflective of the underlying reasons for differences in consumer behavior.
Disadvantage: increased complexity of determining segment memberships for consumers.

Socio-Demographic Segmentation:
includes age, gender, income and education. Useful in industries.



Advantage: Segment membership can easily be determined for every customer.
Disadvantage: if this criterion is not the cause for customers product preferences, then it does not provide sufficient market insight for optimal segmentation decisions.

EV Market

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

In [2]:
```python
df=pd.read_csv("/kaggle/input/ev-india-market/ElectricCarData_Clean_Me.csv")
```

In [3]:
```python
df
```

Out[3]:

	Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Battery_Pack Kwh	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType	BodyStyle	Segment	Seats	PriceEuro	INR
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	460	70.0	161	940	Yes	AWD	Type 2 CCS	Sedan	D	5	55480	4540988.068
1	Volkswagen	ID.3 Pure	10.0	160	270	45.0	167	250	Yes	RWD	Type 2 CCS	Hatchback	C	5	30000	2455473.000
2	Polestar	2	4.7	210	400	75.0	181	620	Yes	AWD	Type 2 CCS	Liftback	D	5	56440	4619563.204
3	BMW	iX3	6.8	180	360	74.0	206	560	Yes	RWD	Type 2 CCS	SUV	D	5	68040	5569012.764
4	Honda	e	9.5	145	170	28.5	168	190	Yes	RWD	Type 2 CCS	Hatchback	B	4	32997	2700774.753
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
97	Nissan	Ariya 63kWh	7.5	160	330	63.0	191	440	Yes	FWD	Type 2 CCS	Hatchback	C	5	45000	3683209.500
98	Audi	e-tron S Sportback 55 quattro	4.5	210	335	86.5	258	540	Yes	AWD	Type 2 CCS	SUV	E	5	96050	7861606.055
99	Nissan	Ariya e-4ORCE 63kWh	5.9	200	325	63.0	194	440	Yes	AWD	Type 2 CCS	Hatchback	C	5	50000	4092455.000
100	Nissan	Ariya e-4ORCE 87kWh Performance	5.1	200	375	87.0	232	450	Yes	AWD	Type 2 CCS	Hatchback	C	5	65000	5320191.500
101	Byton	M-Byte 95 kWh 2WD	7.5	190	400	95.0	238	480	Yes	AWD	Type 2 CCS	SUV	E	5	62000	5074644.200

102 rows × 16 columns

In [4]:
```python
# 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
```

In [5]:
```python
df.columns
```

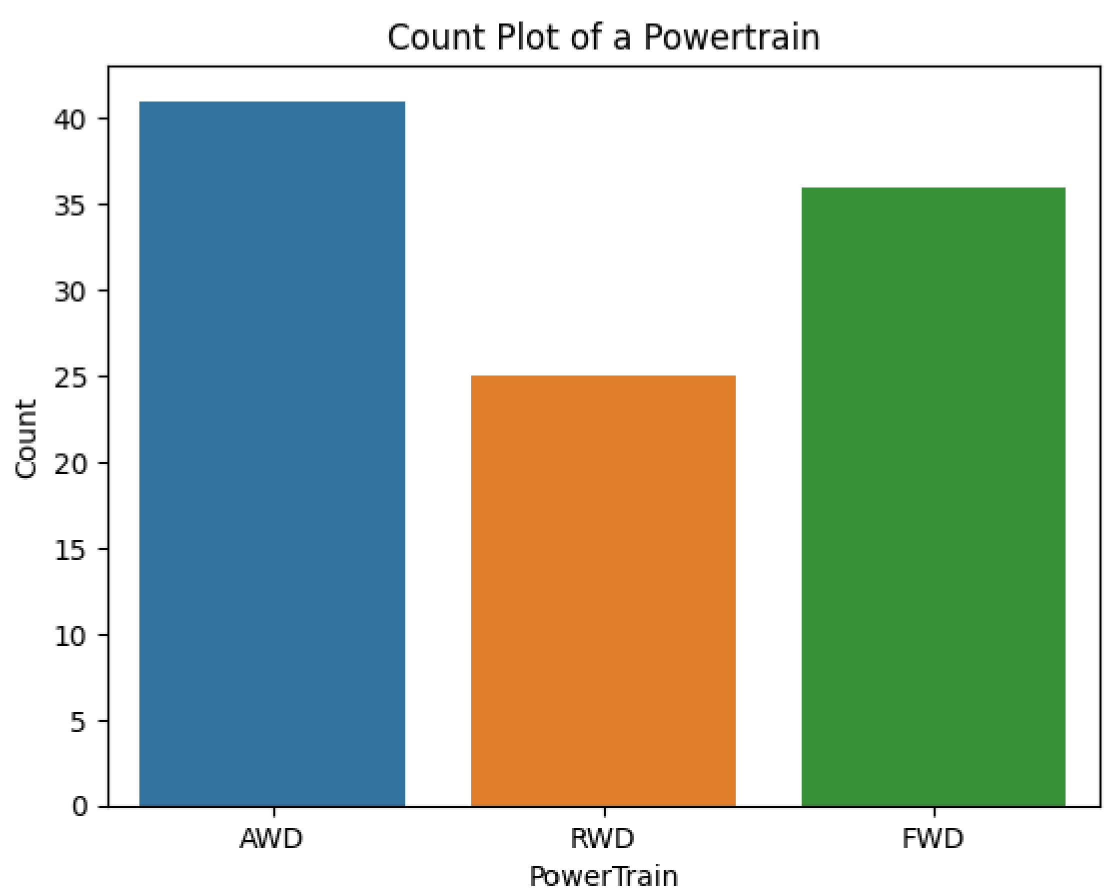
Out[5]:
```
Index(['Brand', 'Model', 'AccelSec', 'TopSpeed_KmH', 'Range_Km',
       'Battery_Pack Kwh', 'Efficiency_WhKm', 'FastCharge_KmH', 'RapidCharge',
       'PowerTrain', 'PlugType', 'BodyStyle', 'Segment', 'Seats', 'PriceEuro',
       'INR'],
      dtype='object')
```

Distributing vehicle price above and below INR 4000000

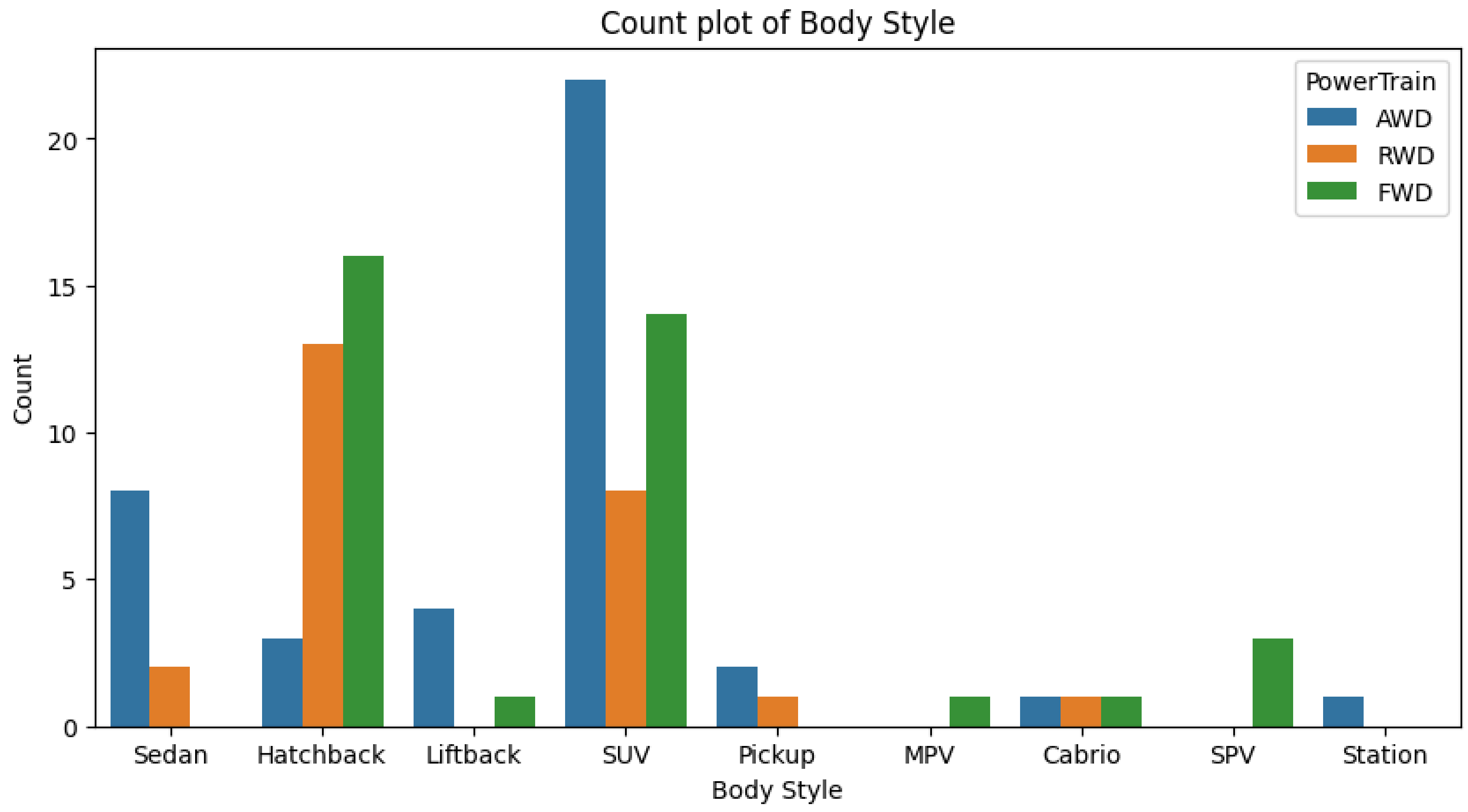
In [6]:
```python
df['CarName'] = df['Brand'] + '-' + df['Model']
df_1= df.loc[df['INR'] <=4000000]
df_2 = df.loc[df['INR'] >4000000]
t1 = ['Less than INR 4000000']
t2 = ['More thanINR 4000000']
```

Count plot for PowerTrain

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

train(df)
```

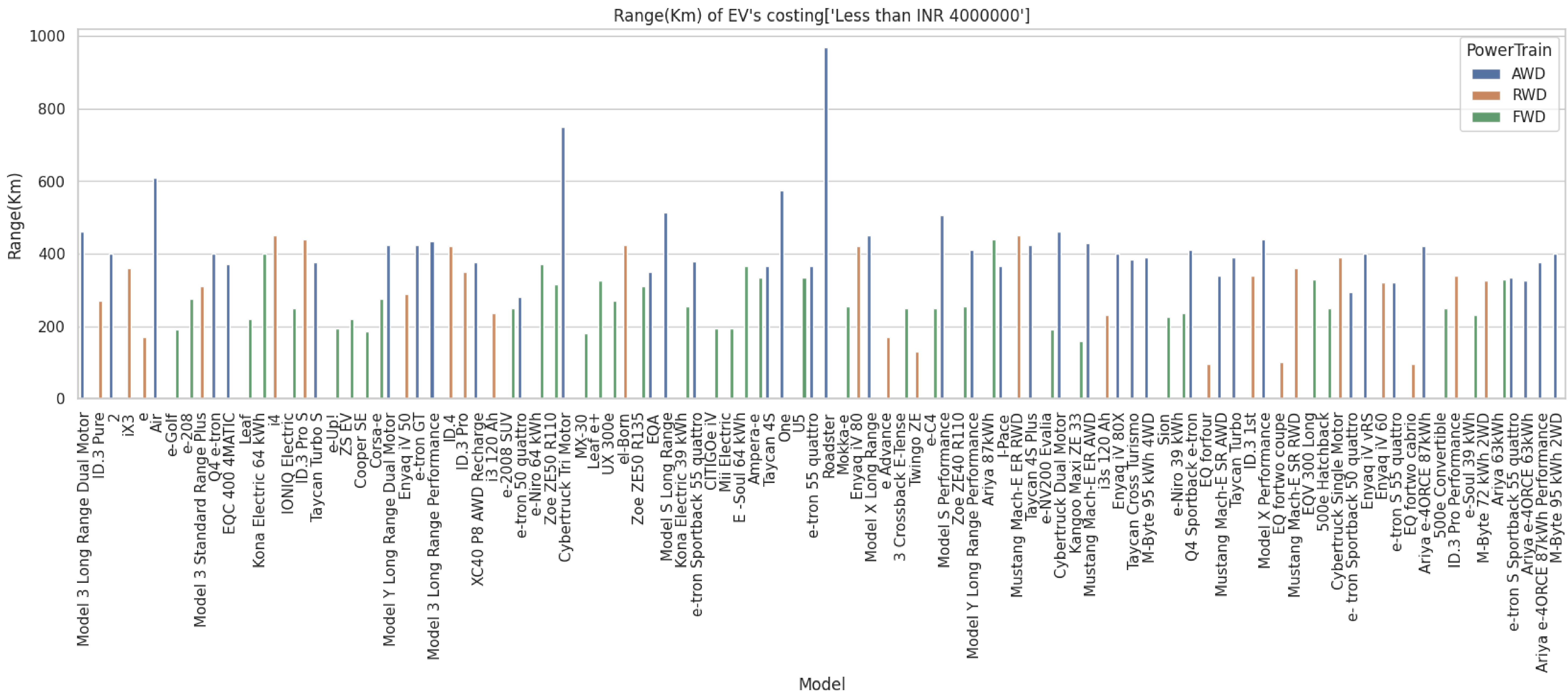


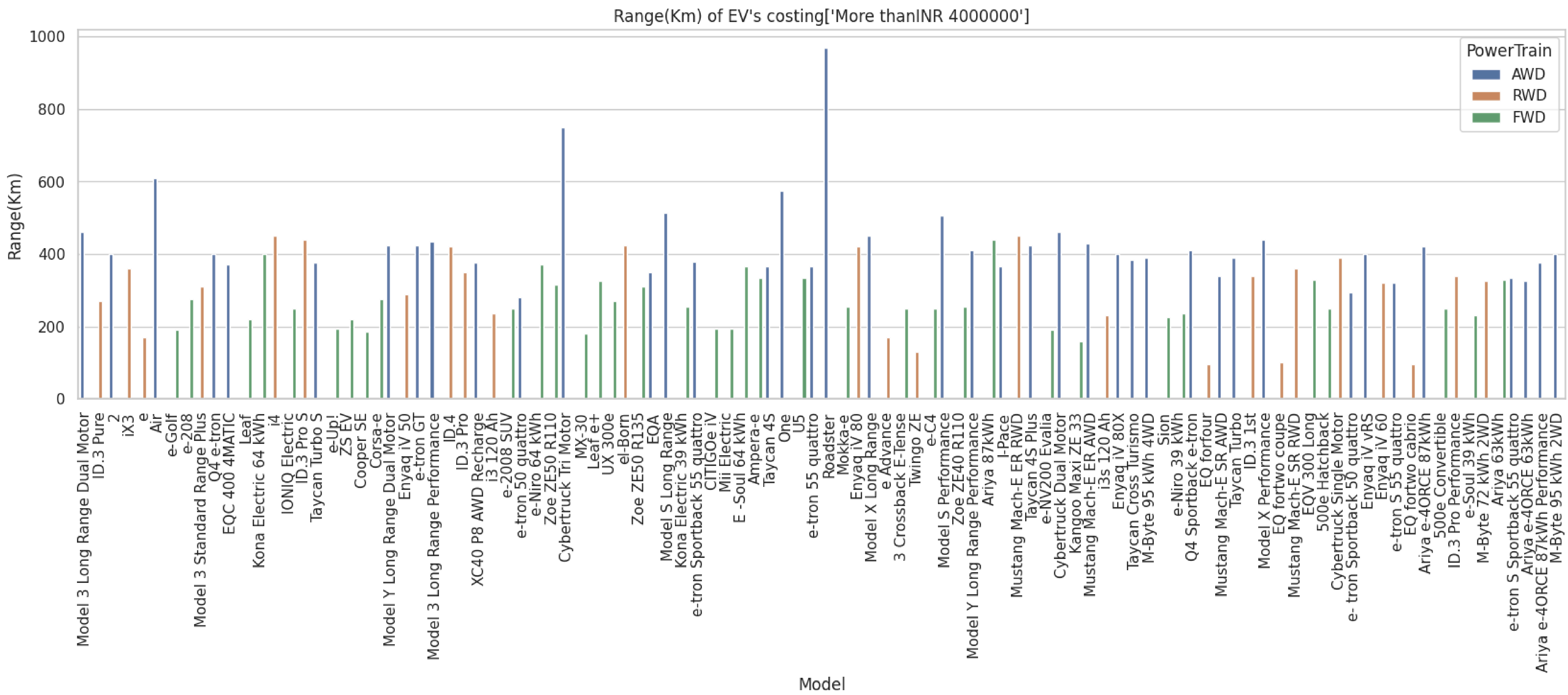
In [8]:
```python
def bodystyle(dataframe):
    plt.figure(figsize=(10,5))
    sbn.countplot(x='BodyStyle', data=dataframe, hue='PowerTrain')
    plt.title('Count plot of Body Style')
    plt.xlabel('Body Style')
    plt.ylabel('Count')
    plt.show()

bodystyle(df)
```



Range of Vehicles

In [9]:
```python
def range(dataframe, price):
    plt.figure(figsize=(20,5))
    sbn.set_theme(style="whitegrid")
    sbn.barplot(x = 'Model', y = '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)
```



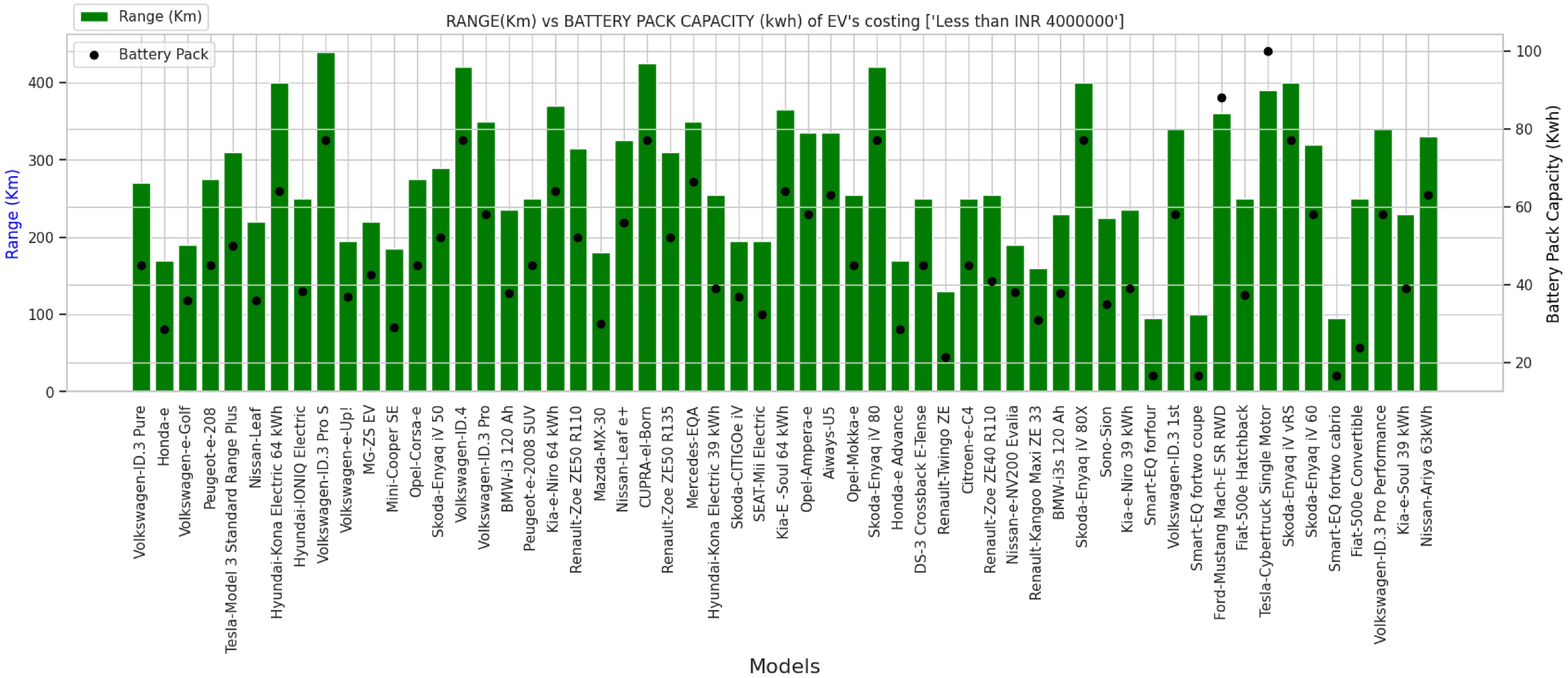


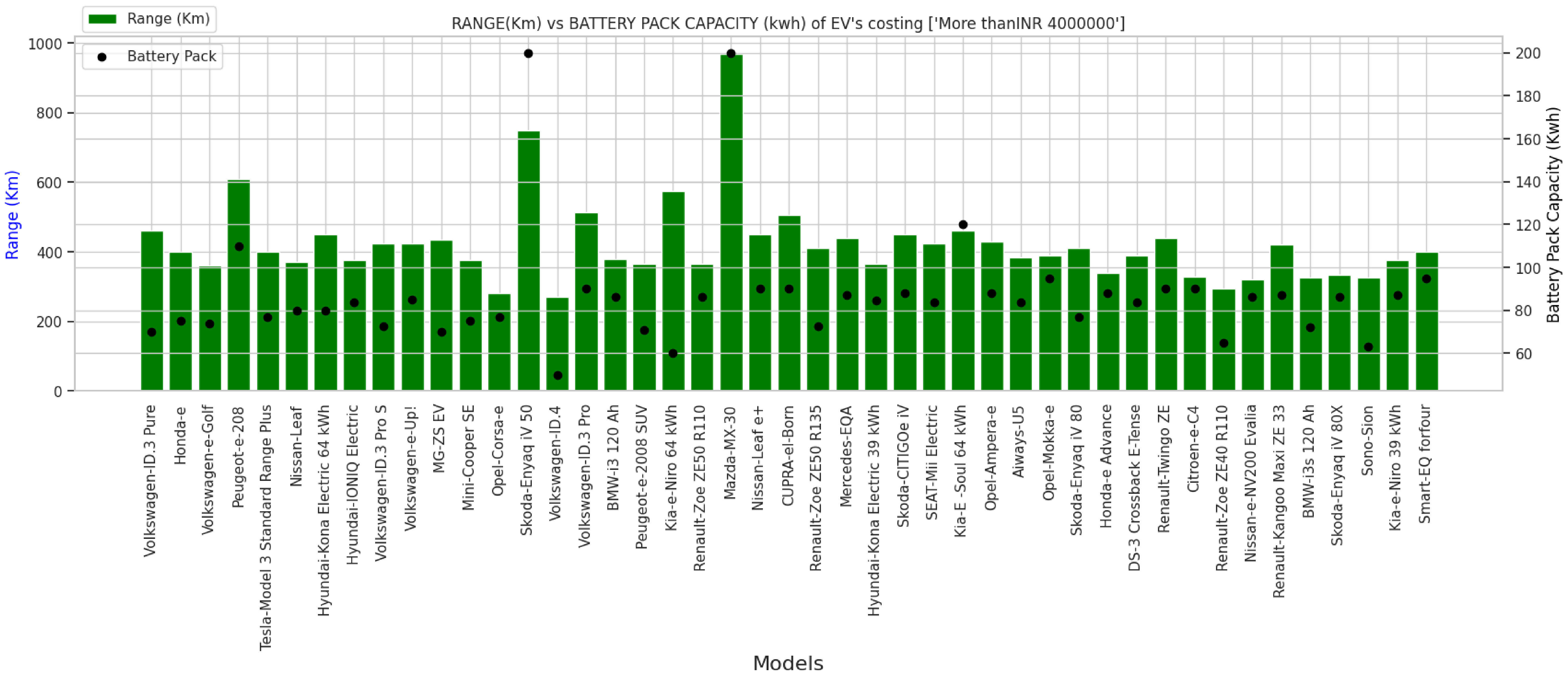
Range - Battery Pack

In [10]:
```python
#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')
```

```python
    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)
```

```
/tmp/ipykernel_33/251817763.py:14: UserWarning: FixedFormatter should only be used together with FixedLocator
  a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
```



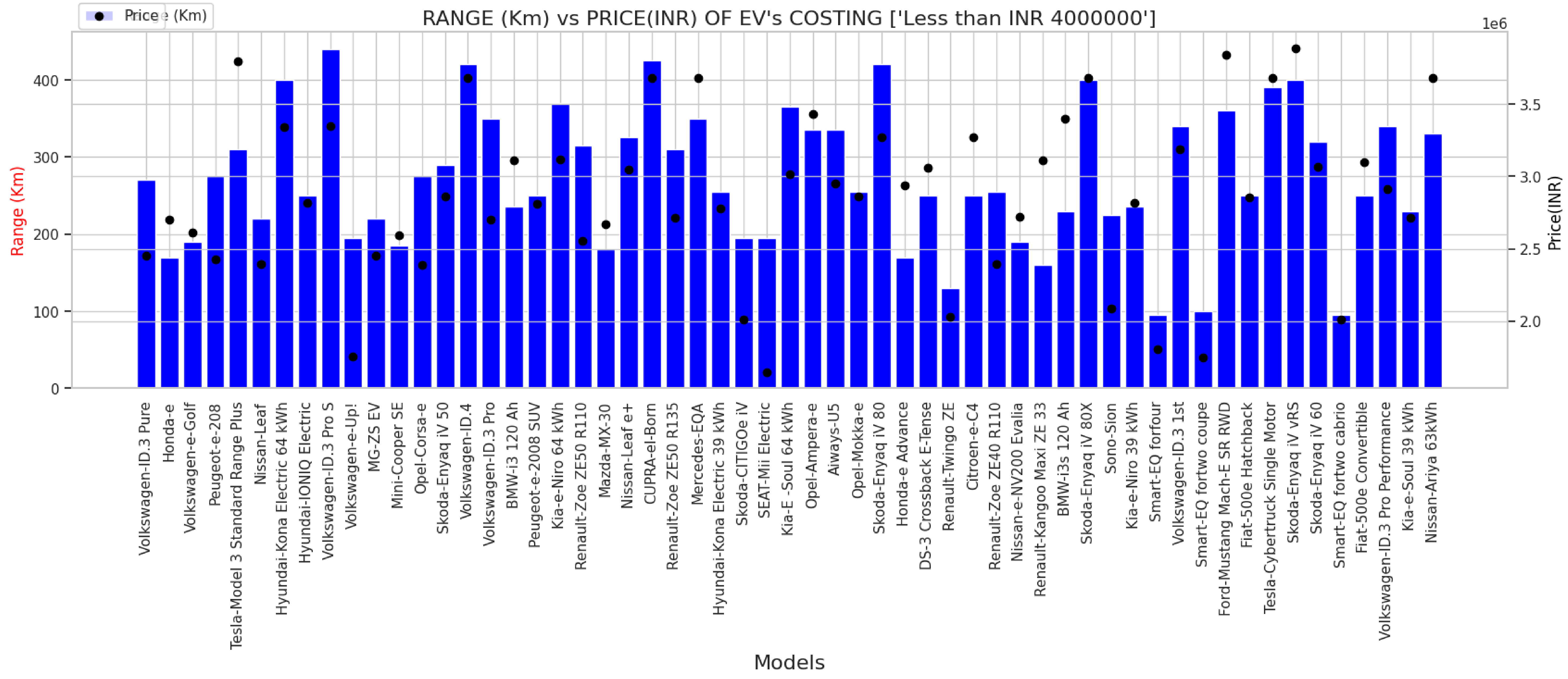


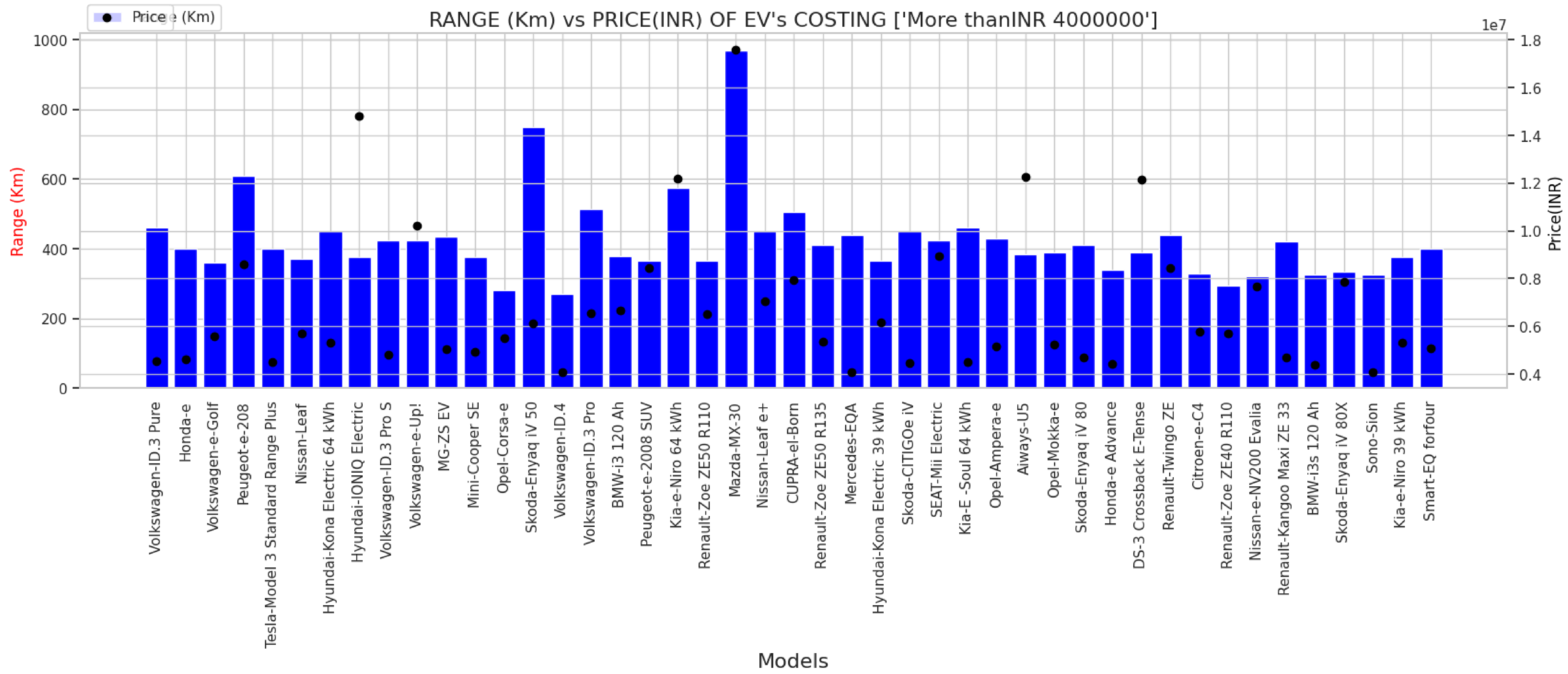
Range - Vehicle Price

In [11]:

```python
#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)
```

```
/tmp/ipykernel_33/4025405043.py:14: UserWarning: FixedFormatter should only be used together with FixedLocator
  a1.set_xticklabels(df_1['CarName'], rotation = 'vertical')
```

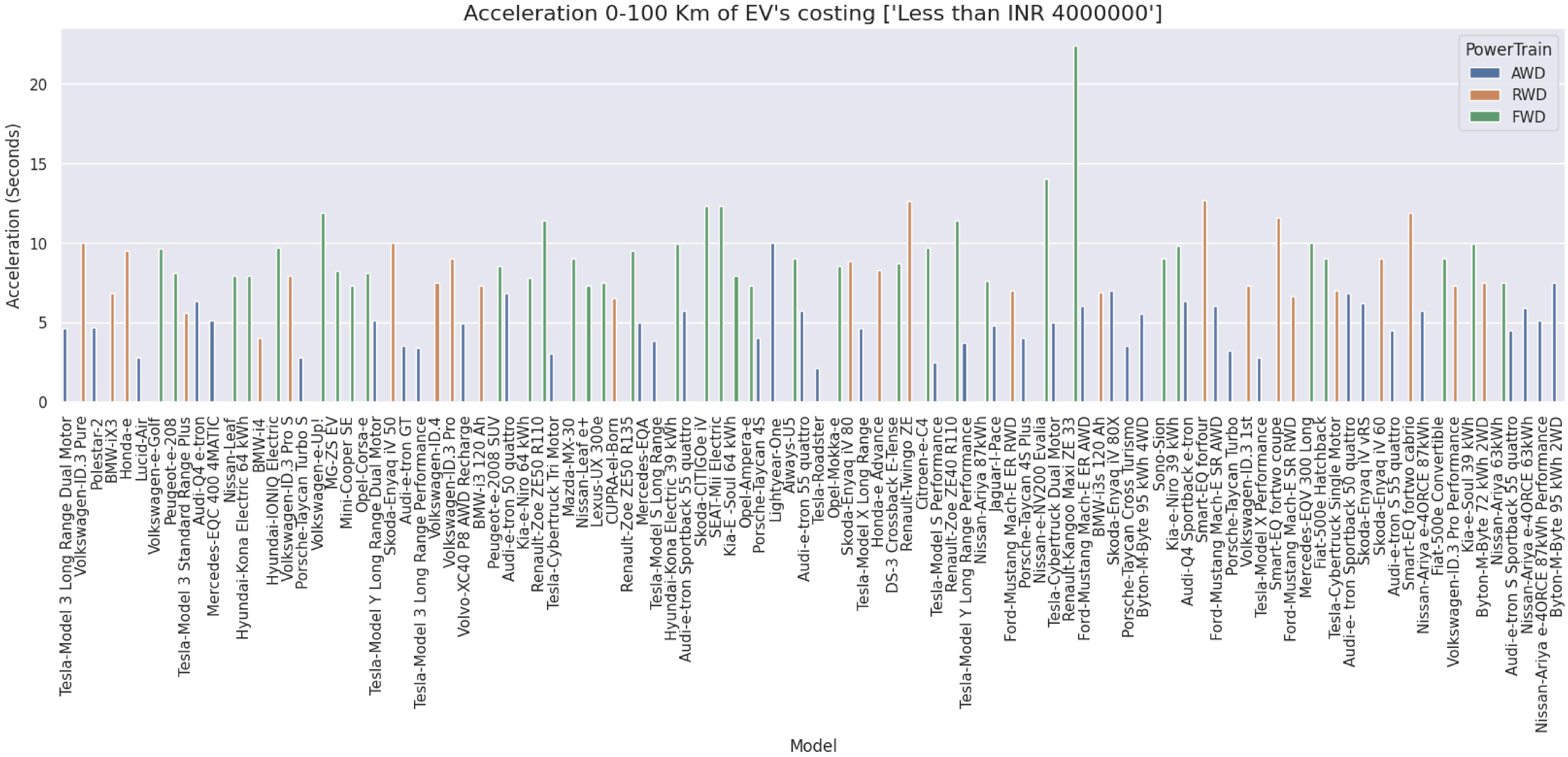


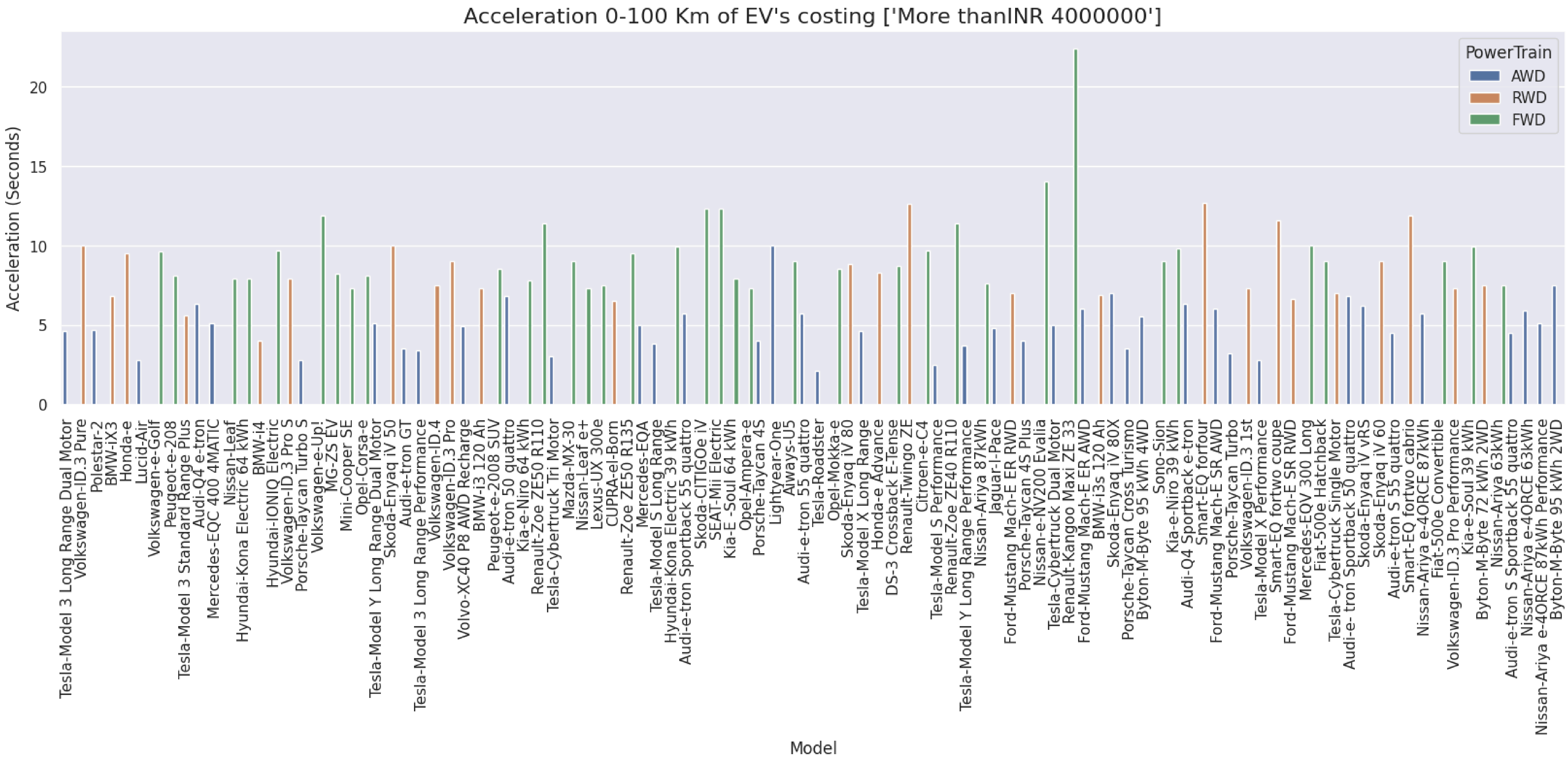


Acceleration(0-100km/hr)

In [12]:

```
#Accelaration(0-100km/hr)
def acc(dataframe, text):
    plt.figure(figsize=(20,5))
    sbn.set_theme(style="darkgrid")
    sbn.barplot(x = 'CarName', y = '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)
```

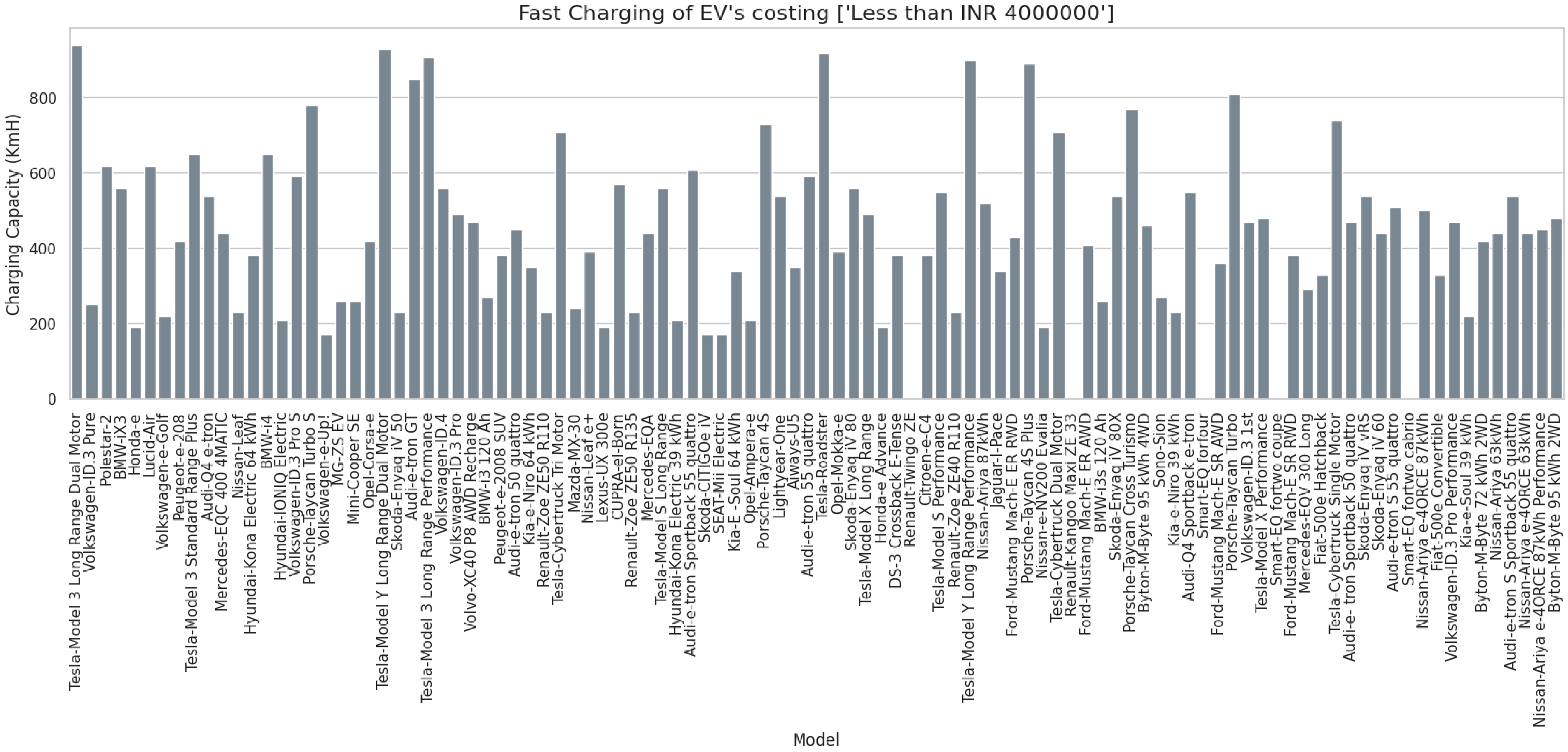


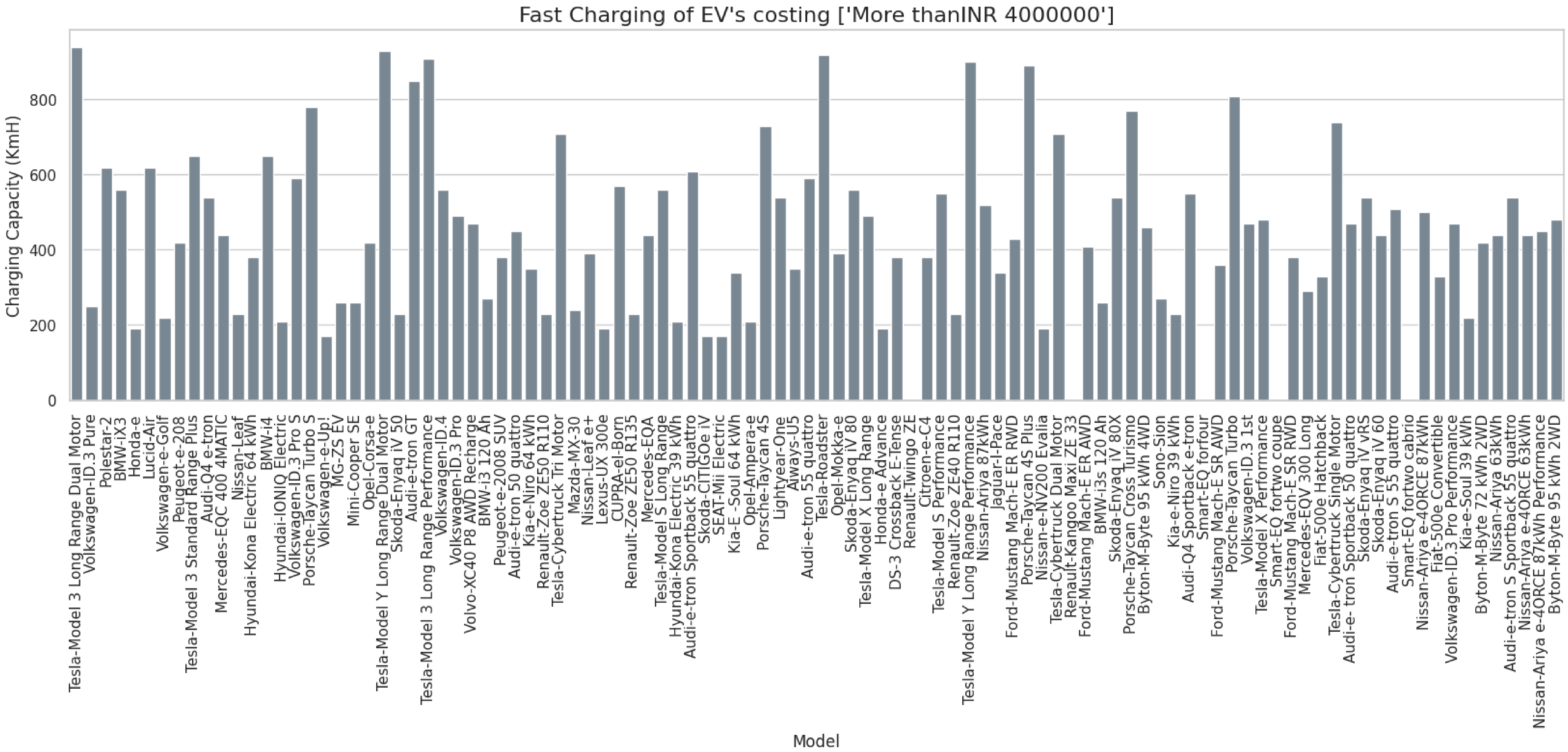


Fast Charging Vehicles

In [13]:

```python
# Fast Charging data
def fastcharge(dataframe, price):
    plt.figure(figsize=(20, 5))
    sbn.set_theme(style="whitegrid")
    sbn.barplot(x='CarName', y='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)
```





Basic Analysis

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)
top_range_1[['CarName', 'Range_Km', 'Battery_Pack Kwh', 'INR', 'RapidCharge']]
```

Out[14]:

	CarName	Range_Km	Battery_Pack Kwh	INR	RapidCharge
15	Volkswagen-ID.3 Pro S	440	77.0	3350574.758	Yes
37	CUPRA-el-Born	425	77.0	3683209.500	Yes
53	Skoda-Enyaq iV 80	420	77.0	3273964.000	Yes
25	Volkswagen-ID.4	420	77.0	3683209.500	Yes
88	Skoda-Enyaq iV vRS	400	77.0	3887832.250	Yes
12	Hyundai-Kona Electric 64 kWh	400	64.0	3339034.035	Yes
71	Skoda-Enyaq iV 80X	400	77.0	3683209.500	Yes
86	Tesla-Cybertruck Single Motor	390	100.0	3683209.500	Yes
31	Kia-e-Niro 64 kWh	370	64.0	3118859.956	Yes
45	Kia-E -Soul 64 kWh	365	64.0	3015075.297	Yes
83	Ford-Mustang Mach-E SR RWD	360	88.0	3838722.790	Yes
39	Mercedes-EQA	350	66.5	3683209.500	Yes
26	Volkswagen-ID.3 Pro	350	58.0	2701020.300	Yes
94	Volkswagen-ID.3 Pro Performance	340	58.0	2911781.733	Yes
80	Volkswagen-ID.3 1st	340	58.0	3191050.862	Yes
49	Aiways-U5	335	63.0	2951232.999	Yes
46	Opel-Ampera-e	335	58.0	3429968.385	Yes
97	Nissan-Ariya 63kWh	330	63.0	3683209.500	Yes
35	Nissan-Leaf e+	325	56.0	3047814.937	Yes
89	Skoda-Enyaq iV 60	320	58.0	3069341.250	Yes
32	Renault-Zoe ZE50 R110	315	52.0	2552382.334	Yes

	CarName	Range_Km	Battery_Pack Kwh	INR	RapidCharge
38	Renault-Zoe ZE50 R135	310	52.0	2711906.230	Yes
8	Tesla-Model 3 Standard Range Plus	310	50.0	3796161.258	Yes
22	Skoda-Enyaq iV 50	290	52.0	2864718.500	Yes
20	Opel-Corsa-e	275	45.0	2385573.869	Yes
7	Peugeot-e-208	275	45.0	2429444.986	Yes
1	Volkswagen-ID.3 Pure	270	45.0	2455473.000	Yes
52	Opel-Mokka-e	255	45.0	2864718.500	Yes
60	Renault-Zoe ZE40 R110	255	41.0	2392776.589	Yes
41	Hyundai-Kona Electric 39 kWh	255	39.0	2780495.776	Yes
29	Peugeot-e-2008 SUV	250	45.0	2812416.925	Yes
14	Hyundai-IONIQ Electric	250	38.3	2820438.137	Yes
93	Fiat-500e Convertible	250	23.8	3102080.890	Yes
56	DS-3 Crossback E-Tense	250	45.0	3062957.020	Yes
85	Fiat-500e Hatchback	250	37.3	2856533.590	Yes
58	Citroen-e-C4	250	45.0	3273964.000	Yes
28	BMW-i3 120 Ah	235	37.9	3111657.235	Yes
75	Kia-e-Niro 39 kWh	235	39.0	2815609.040	Yes
70	BMW-i3s 120 Ah	230	37.9	3398865.727	Yes
95	Kia-e-Soul 39 kWh	230	39.0	2711906.230	Yes
74	Sono-Sion	225	35.0	2087152.050	Yes
11	Nissan-Leaf	220	36.0	2392776.589	Yes
18	MG-ZS EV	220	42.5	2455473.000	Yes
17	Volkswagen-e-Up!	195	36.8	1753289.571	Yes
43	Skoda-CITIGOe iV	195	36.8	2008085.819	Yes
44	SEAT-Mii Electric	195	32.3	1647540.534	Yes
6	Volkswagen-e-Golf	190	35.8	2610986.290	Yes
66	Nissan-e-NV200 Evalia	190	38.0	2721155.179	Yes
19	Mini-Cooper SE	185	28.9	2593061.337	Yes
34	Mazda-MX-30	180	30.0	2672045.719	Yes
55	Honda-e Advance	170	28.5	2940101.521	Yes
4	Honda-e	170	28.5	2700774.753	Yes
68	Renault-Kangoo Maxi ZE 33	160	31.0	3110265.800	No
57	Renault-Twingo ZE	130	21.3	2029039.189	No
82	Smart-EQ fortwo coupe	100	16.7	1750506.702	No
91	Smart-EQ fortwo cabrio	95	16.7	2010623.142	No
77	Smart-EQ forfour	95	16.7	1803135.673	No

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')
acceleration_1[['CarName', 'AccelSec', 'Range_Km', 'PowerTrain', 'Battery_Pack Kwh', 'INR']]
```

Out[15]:

	CarName	AccelSec	Range_Km	PowerTrain	Battery_Pack Kwh	INR
39	Mercedes-EQA	5.0	350	AWD	66.5	3683209.500
8	Tesla-Model 3 Standard Range Plus	5.6	310	RWD	50.0	3796161.258
88	Skoda-Enyaq iV vRS	6.2	400	AWD	77.0	3887832.250
37	CUPRA-el-Born	6.5	425	RWD	77.0	3683209.500
83	Ford-Mustang Mach-E SR RWD	6.6	360	RWD	88.0	3838722.790
70	BMW-i3s 120 Ah	6.9	230	RWD	37.9	3398865.727
86	Tesla-Cybertruck Single Motor	7.0	390	RWD	100.0	3683209.500
71	Skoda-Enyaq iV 80X	7.0	400	AWD	77.0	3683209.500
35	Nissan-Leaf e+	7.3	325	FWD	56.0	3047814.937
19	Mini-Cooper SE	7.3	185	FWD	28.9	2593061.337
28	BMW-i3 120 Ah	7.3	235	RWD	37.9	3111657.235
80	Volkswagen-ID.3 1st	7.3	340	RWD	58.0	3191050.862
94	Volkswagen-ID.3 Pro Performance	7.3	340	RWD	58.0	2911781.733
46	Opel-Ampera-e	7.3	335	FWD	58.0	3429968.385
25	Volkswagen-ID.4	7.5	420	RWD	77.0	3683209.500
97	Nissan-Ariya 63kWh	7.5	330	FWD	63.0	3683209.500
31	Kia-e-Niro 64 kWh	7.8	370	FWD	64.0	3118859.956
45	Kia-E -Soul 64 kWh	7.9	365	FWD	64.0	3015075.297
15	Volkswagen-ID.3 Pro S	7.9	440	RWD	77.0	3350574.758
12	Hyundai-Kona Electric 64 kWh	7.9	400	FWD	64.0	3339034.035
11	Nissan-Leaf	7.9	220	FWD	36.0	2392776.589
20	Opel-Corsa-e	8.1	275	FWD	45.0	2385573.869
7	Peugeot-e-208	8.1	275	FWD	45.0	2429444.986
18	MG-ZS EV	8.2	220	FWD	42.5	2455473.000
55	Honda-e Advance	8.3	170	RWD	28.5	2940101.521
29	Peugeot-e-2008 SUV	8.5	250	FWD	45.0	2812416.925
52	Opel-Mokka-e	8.5	255	FWD	45.0	2864718.500
56	DS-3 Crossback E-Tense	8.7	250	FWD	45.0	3062957.020
53	Skoda-Enyaq iV 80	8.8	420	RWD	77.0	3273964.000
26	Volkswagen-ID.3 Pro	9.0	350	RWD	58.0	2701020.300
85	Fiat-500e Hatchback	9.0	250	FWD	37.3	2856533.590
49	Aiways-U5	9.0	335	FWD	63.0	2951232.999
89	Skoda-Enyaq iV 60	9.0	320	RWD	58.0	3069341.250
93	Fiat-500e Convertible	9.0	250	FWD	23.8	3102080.890
34	Mazda-MX-30	9.0	180	FWD	30.0	2672045.719
74	Sono-Sion	9.0	225	FWD	35.0	2087152.050
38	Renault-Zoe ZE50 R135	9.5	310	FWD	52.0	2711906.230

	CarName	AccelSec	Range_Km	PowerTrain	Battery_Pack Kwh	INR
4	Honda-e	9.5	170	RWD	28.5	2700774.753
6	Volkswagen-e-Golf	9.6	190	FWD	35.8	2610986.290
58	Citroen-e-C4	9.7	250	FWD	45.0	3273964.000
14	Hyundai-IONIQ Electric	9.7	250	FWD	38.3	2820438.137
75	Kia-e-Niro 39 kWh	9.8	235	FWD	39.0	2815609.040
95	Kia-e-Soul 39 kWh	9.9	230	FWD	39.0	2711906.230
41	Hyundai-Kona Electric 39 kWh	9.9	255	FWD	39.0	2780495.776
1	Volkswagen-ID.3 Pure	10.0	270	RWD	45.0	2455473.000
22	Skoda-Enyaq iV 50	10.0	290	RWD	52.0	2864718.500
32	Renault-Zoe ZE50 R110	11.4	315	FWD	52.0	2552382.334
60	Renault-Zoe ZE40 R110	11.4	255	FWD	41.0	2392776.589
82	Smart-EQ fortwo coupe	11.6	100	RWD	16.7	1750506.702
17	Volkswagen-e-Up!	11.9	195	FWD	36.8	1753289.571
91	Smart-EQ fortwo cabrio	11.9	95	RWD	16.7	2010623.142
44	SEAT-Mii Electric	12.3	195	FWD	32.3	1647540.534
43	Skoda-CITIGOe iV	12.3	195	FWD	36.8	2008085.819
57	Renault-Twingo ZE	12.6	130	RWD	21.3	2029039.189
77	Smart-EQ forfour	12.7	95	RWD	16.7	1803135.673
66	Nissan-e-NV200 Evalia	14.0	190	FWD	38.0	2721155.179
68	Renault-Kangoo Maxi ZE 33	22.4	160	FWD	31.0	3110265.800

Vehicles with Maximum Efficiency

In [16]:

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

Out[16]:

	CarName	Efficiency_WhKm	Range_Km	PowerTrain	Battery_Pack Kwh	INR
48	Lightyear-One	104	575	AWD	60.0	1.219552e+07
14	Hyundai-IONIQ Electric	153	250	FWD	38.3	2.820438e+06
8	Tesla-Model 3 Standard Range Plus	153	310	RWD	50.0	3.796161e+06
41	Hyundai-Kona Electric 39 kWh	154	255	FWD	39.0	2.780496e+06
74	Sono-Sion	156	225	FWD	35.0	2.087152e+06
...	...	...	...	...	...	...
98	Audi-e-tron S Sportback 55 quattro	258	335	AWD	86.5	7.861606e+06
67	Tesla-Cybertruck Dual Motor	261	460	AWD	120.0	4.501700e+06
33	Tesla-Cybertruck Tri Motor	267	750	AWD	200.0	6.138682e+06
90	Audi-e-tron S 55 quattro	270	320	AWD	86.5	7.677446e+06
84	Mercedes-EQV 300 Long	273	330	FWD	90.0	5.781084e+06

102 rows × 6 columns

Budget wise EV Car Analysis

In [17]:

```
df1=pd.read_csv("/kaggle/input/ev-india-market/EVIndia.csv")
PriceRange = (df1['PriceRange'].astype(str))
df1.head(20)
```

Out[17]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	939950	5 Seater	350 L	XM	Dark XZ Plus LUX
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	Automatic	Electric	1306500	5 Seater	316 L	XE	XZ Plus Dual Tone
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	Automatic	Electric	1306500	5 Seater	350 L	XZ Plus 3.3 kW	XZ Plus Lux 7.2 kW
3	MG ZS EV	Compact SUV	419 Km/Full Charge	Automatic	Electric	2393500	5 Seater	448 L	Excite	Exclusive
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	Automatic	Electric	2388500	5 Seater	na	Premium Dual Tone	HSE
5	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	10900000	5 Seater	656 L	S	Sportback 55
6	Audi E-Tron GT	Premium Coupe	388 Km/Full Charge	Automatic	Electric	18000000	5 Seater	405 L	Quattro	na
7	BYD E6	Subcompact MPV	415 Km/Full Charge	Automatic	Electric	2915000	5 Seater	580 L	STD	na
8	Mercedes-Benz EQC	Compact SUV	471 Km/Full Charge	Automatic	Electric	10000000	5 Seater	na	na	na
9	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	11600000	5 Seater	na	na	na
10	Porsche Taycan	Premium Sports Sedan	na	Automatic	Electric	15000000	4 Seater	na	na	na
11	Audi E-Tron	Compact SUV	400 Km/Full Charge	Automatic	Electric	11000000	5 Seater	660 L	na	na

In [18]:

```
df1.isnull().sum()
```

Out[18]:

```
Car             0
Style           0
Range           0
Transmission    0
VehicleType     0
PriceRange      0
Capacity        0
BootSpace       0
BaseModel       0
TopModel        0
dtype: int64
```

In [19]:

```
df1['PriceRange']
```

Out[19]:

```
0       939950
1      1306500
2      1306500
3      2393500
4      2388500
5     10900000
6     18000000
7      2915000
8     10000000
9     11600000
10    15000000
11    11000000
Name: PriceRange, dtype: int64
```

In [20]:

```
# this will replace "Boston Celtics" with "Omega Warrior"
df1 = df1.replace(to_replace="₹",value=" ")
df1
```

Out[20]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	939950	5 Seater	350 L	XM	Dark XZ Plus LUX
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	Automatic	Electric	1306500	5 Seater	316 L	XE	XZ Plus Dual Tone
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	Automatic	Electric	1306500	5 Seater	350 L	XZ Plus 3.3 kW	XZ Plus Lux 7.2 kW
3	MG ZS EV	Compact SUV	419 Km/Full Charge	Automatic	Electric	2393500	5 Seater	448 L	Excite	Exclusive
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	Automatic	Electric	2388500	5 Seater	na	Premium Dual Tone	HSE
5	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	10900000	5 Seater	656 L	S	Sportback 55
6	Audi E-Tron GT	Premium Coupe	388 Km/Full Charge	Automatic	Electric	18000000	5 Seater	405 L	Quattro	na
7	BYD E6	Subcompact MPV	415 Km/Full Charge	Automatic	Electric	2915000	5 Seater	580 L	STD	na
8	Mercedes-Benz EQC	Compact SUV	471 Km/Full Charge	Automatic	Electric	10000000	5 Seater	na	na	na
9	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	11600000	5 Seater	na	na	na
10	Porsche Taycan	Premium Sports Sedan	na	Automatic	Electric	15000000	4 Seater	na	na	na
11	Audi E-Tron	Compact SUV	400 Km/Full Charge	Automatic	Electric	11000000	5 Seater	660 L	na	na

In [21]:

```
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 [22]:

```
mid_range_cars
```

Out[22]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	939950	5 Seater	350 L	XM	Dark XZ Plus LUX
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	Automatic	Electric	1306500	5 Seater	316 L	XE	XZ Plus Dual Tone
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	Automatic	Electric	1306500	5 Seater	350 L	XZ Plus 3.3 kW	XZ Plus Lux 7.2 kW
3	MG ZS EV	Compact SUV	419 Km/Full Charge	Automatic	Electric	2393500	5 Seater	448 L	Excite	Exclusive
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	Automatic	Electric	2388500	5 Seater	na	Premium Dual Tone	HSE
7	BYD E6	Subcompact MPV	415 Km/Full Charge	Automatic	Electric	2915000	5 Seater	580 L	STD	na

In [23]:

```
high_range_cars
```

Out[23]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Capacity	BootSpace	BaseModel	TopModel
5	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	10900000	5 Seater	656 L	S	Sportback 55
6	Audi E-Tron GT	Premium Coupe	388 Km/Full Charge	Automatic	Electric	18000000	5 Seater	405 L	Quattro	na
8	Mercedes-Benz EQC	Compact SUV	471 Km/Full Charge	Automatic	Electric	10000000	5 Seater	na	na	na
9	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	11600000	5 Seater	na	na	na
10	Porsche Taycan	Premium Sports Sedan	na	Automatic	Electric	15000000	4 Seater	na	na	na
11	Audi E-Tron	Compact SUV	400 Km/Full Charge	Automatic	Electric	11000000	5 Seater	660 L	na	na

mid-range vehicles with max range

In [26]:

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

Out[26]:

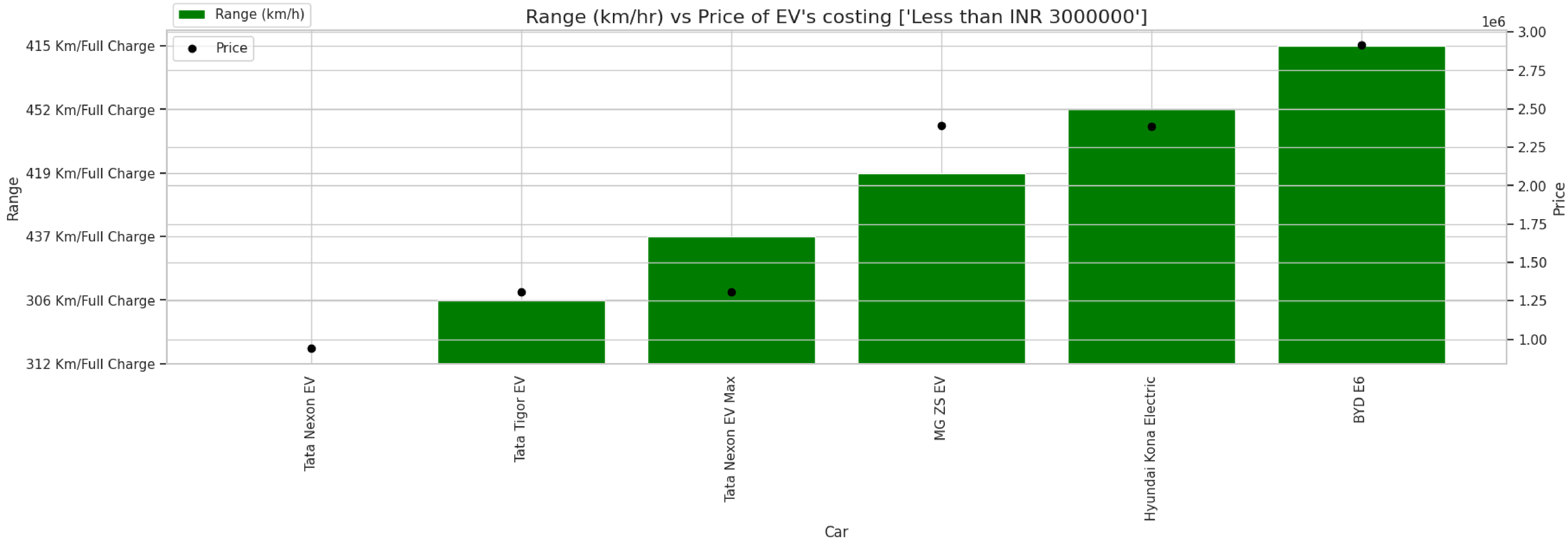
	Car	Style	Range	PriceRange	BootSpace
1	Tata Tigor EV	Subcompact Sedan	306 Km/Full Charge	1306500	316 L
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	939950	350 L
7	BYD E6	Subcompact MPV	415 Km/Full Charge	2915000	580 L
3	MG ZS EV	Compact SUV	419 Km/Full Charge	2393500	448 L
2	Tata Nexon EV Max	Compact SUV	437 Km/Full Charge	1306500	350 L
4	Hyundai Kona Electric	Compact SUV	452 Km/Full Charge	2388500	na

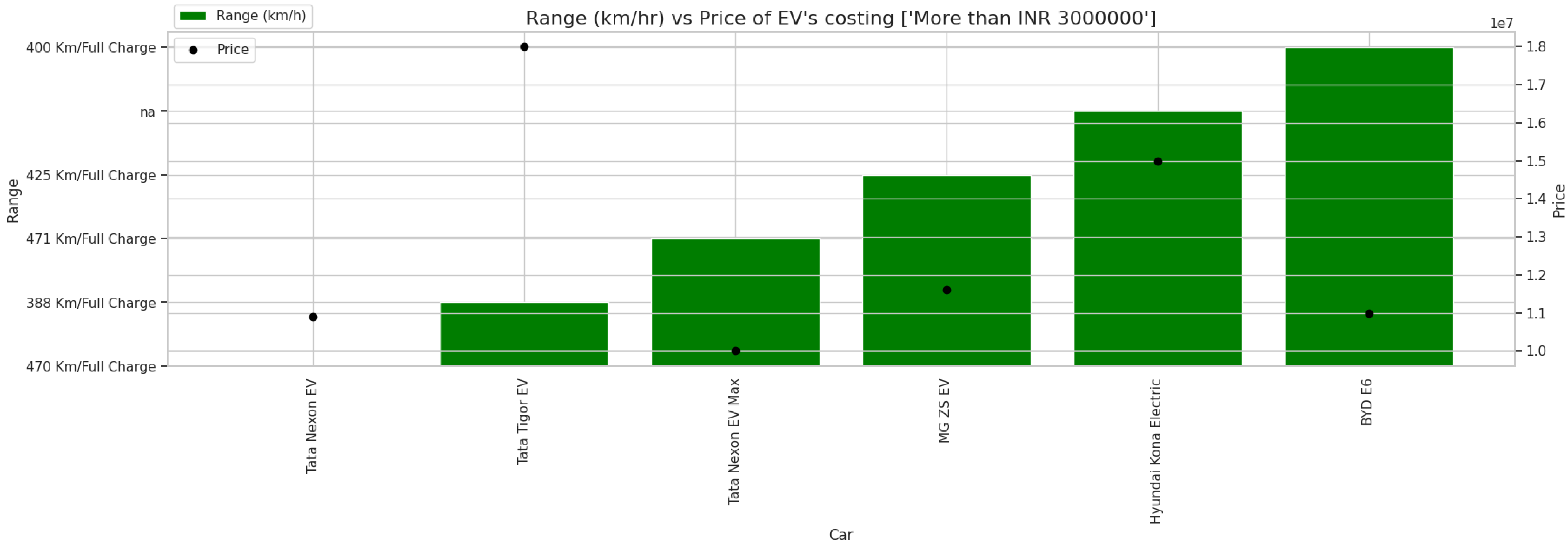
Visualizing Price - Range

In [27]:

```
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)
```

```
/tmp/ipykernel_33/4044561395.py:13: UserWarning: FixedFormatter should only be used together with FixedLocator
  a_1.set_xticklabels(mid_range_cars['Car'], rotation = 'vertical')
```





In [68]:

```
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")
```

In [69]:

```
data=pd.read_csv("/kaggle/input/ev-india-market/pollution data.csv")
data
```

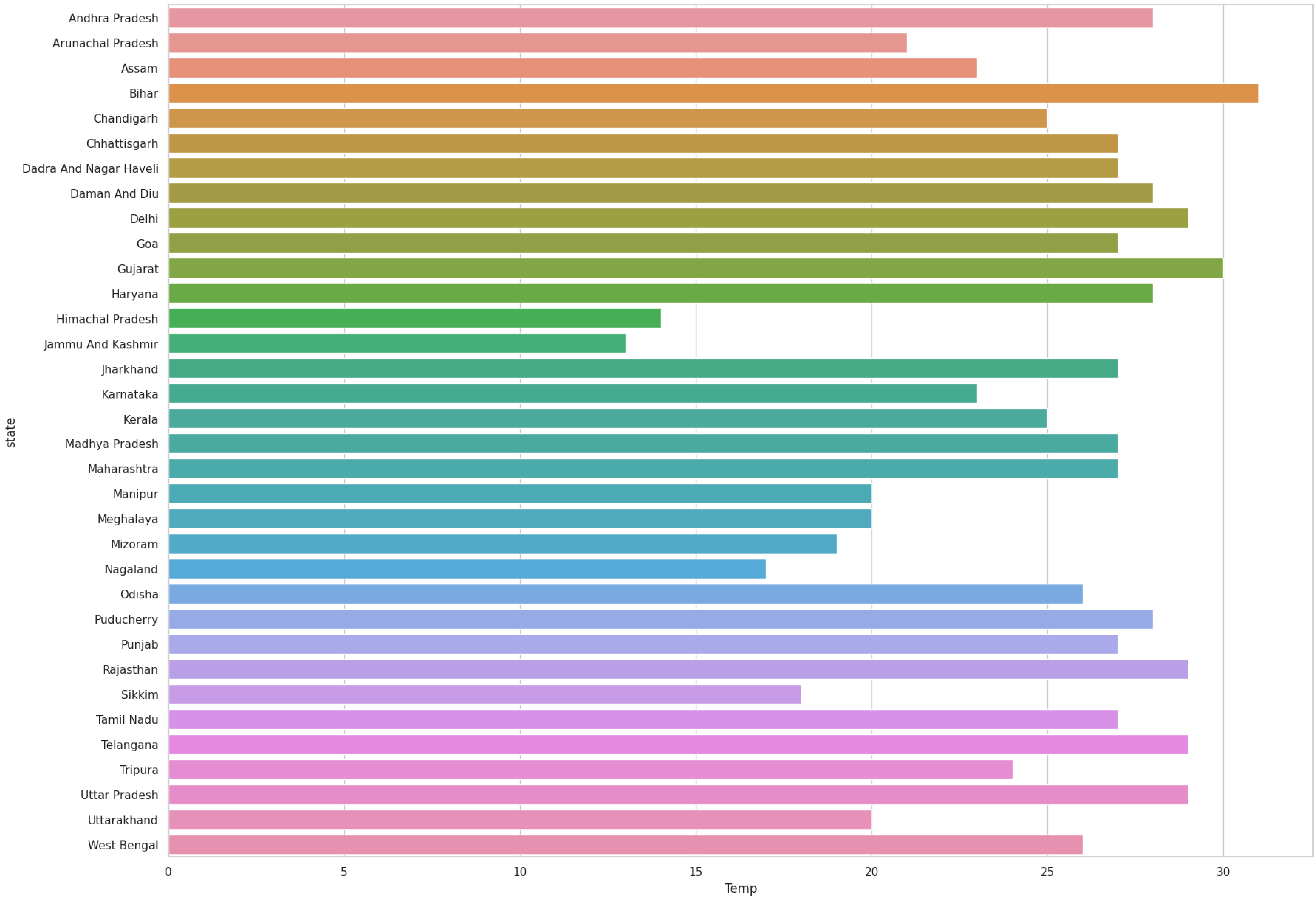
Out[69]:

	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
20	Meghalaya	MODERATE	55	16	23	20	98
21	Mizoram	GOOD	14	2	5	19	100
22	Nagaland	GOOD	25	6	10	17	97
23	Odisha	MODERATE	79	25	42	26	84
24	Puducherry	MODERATE	54	15	31	28	80
25	Punjab	MODERATE	73	22	51	27	57
26	Rajasthan	POOR	107	38	67	29	61
27	Sikkim	MODERATE	70	21	54	18	96
28	Tamil Nadu	MODERATE	61	17	36	27	77
29	Telangana	MODERATE	69	21	31	29	62
30	Tripura	GOOD	14	2	5	24	96
31	Uttar Pradesh	MODERATE	96	34	85	29	61
32	Uttarakhand	POOR	108	38	59	20	67
33	West Bengal	MODERATE	68	20	45	26	84

In [75]:

```
plt.figure(figsize=(20,15))
sns.barplot(x = "Temp",y = "state",data=data)
```

Out[75]: <Axes: xlabel='Temp', ylabel='state'>



In [77]:

```
dataframe = pd.read_csv("/kaggle/input/states/states_data_car.csv")
dataframe
```

Out[77]:

	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
20	Rajasthan	Jaipur	0.0	0.00	108.07	93.35
21	Sikkim	Gangtok	0.0	0.00	102.85	98.90
22	Tamil Nadu	Chennai	0.0	1.00	103.62	95.24
23	Telangana	Hyderabad	0.0	1.00	111.97	99.97
24	Tripura	Agartala	5000.0	0.00	98.58	87.52
25	Uttar Pradesh	Lucknow	0.0	0.75	96.38	89.55
26	Uttarakhand	Dehradun	0.0	0.00	95.62	90.55
27	West Bengal	kolkata	10000.0	1.00	106.79	93.47
28	Andaman and Nicobar Islands	Port Blair	0.0	0.00	84.10	79.74
29	Chandigarh	Chandigarh	5000.0	1.00	96.20	84.26
30	Dadra and Nagar Haveli and Daman and Diu	Daman	5000.0	0.00	94.43	89.98
31	Delhi	Delhi	10000.0	1.00	96.72	89.62
32	Jammu and Kashmir	Srinagar	0.0	0.00	100.94	86.09
33	Ladakh	Leh	0.0	0.00	107.01	91.39
34	Lakshadweep	Kavaratti	0.0	0.00	107.71	96.47
35	Puducherry	Puducherry	5000.0	1.00	95.88	86.11

In [78]:

```
dataframe.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 [79]:

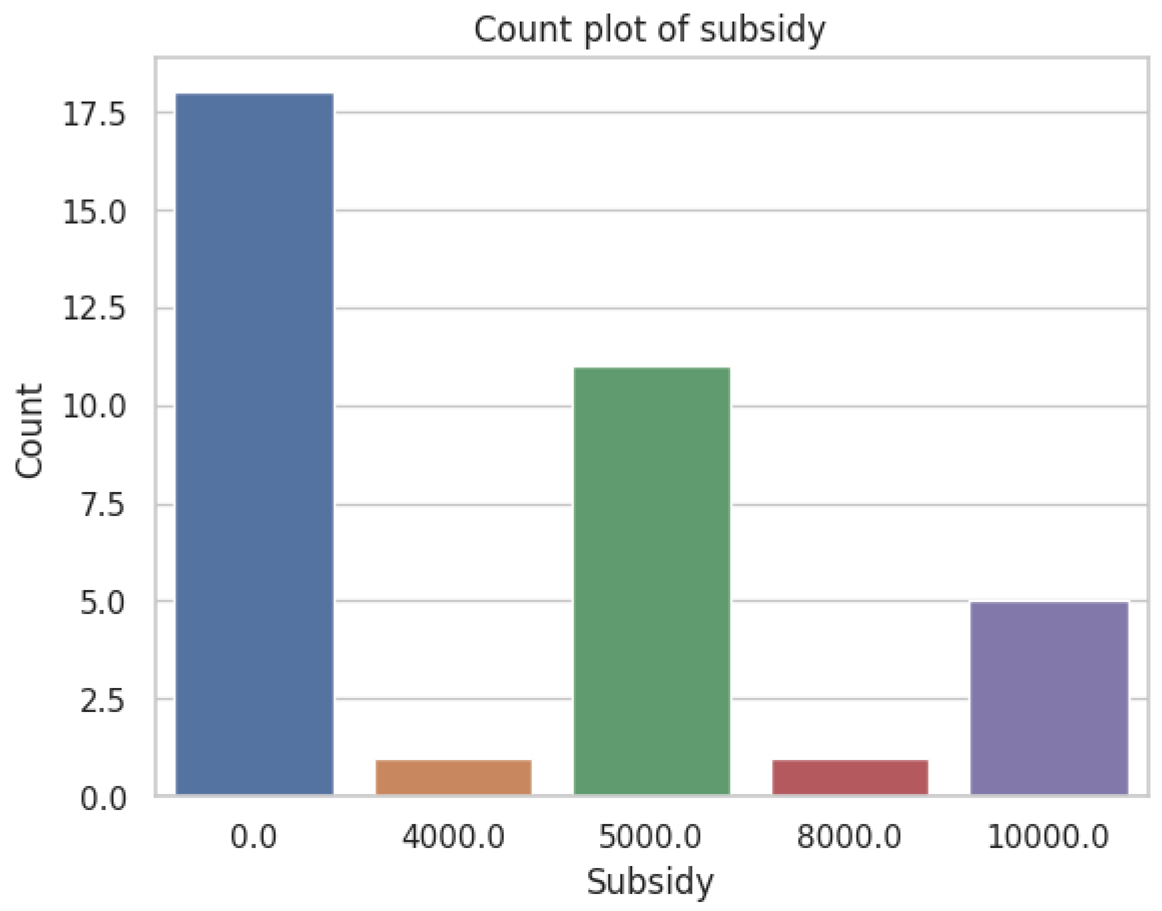
```
dataframe.isnull().sum()
```

Out[79]:

```
state       0
capital     0
subsidy     0
road tax    0
petrol      0
diesel      0
dtype: int64
```

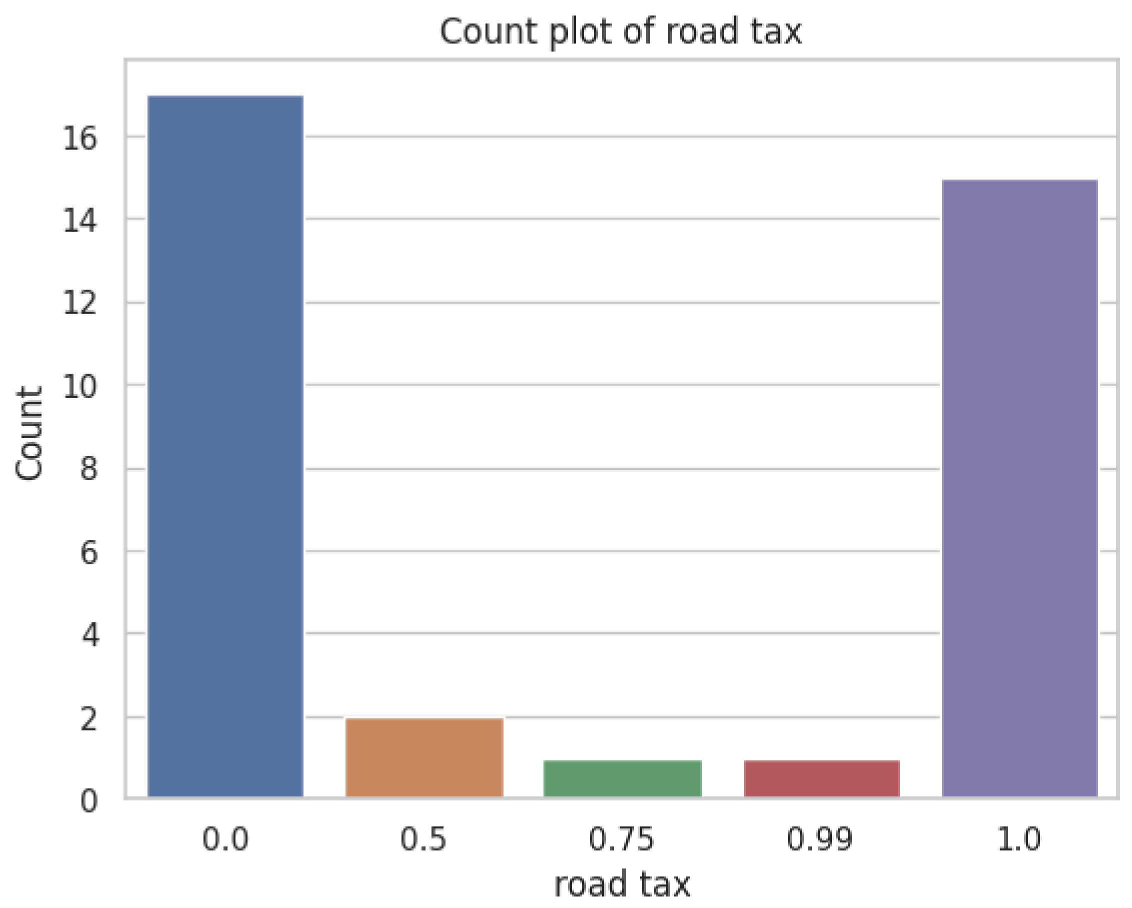
In [80]:

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



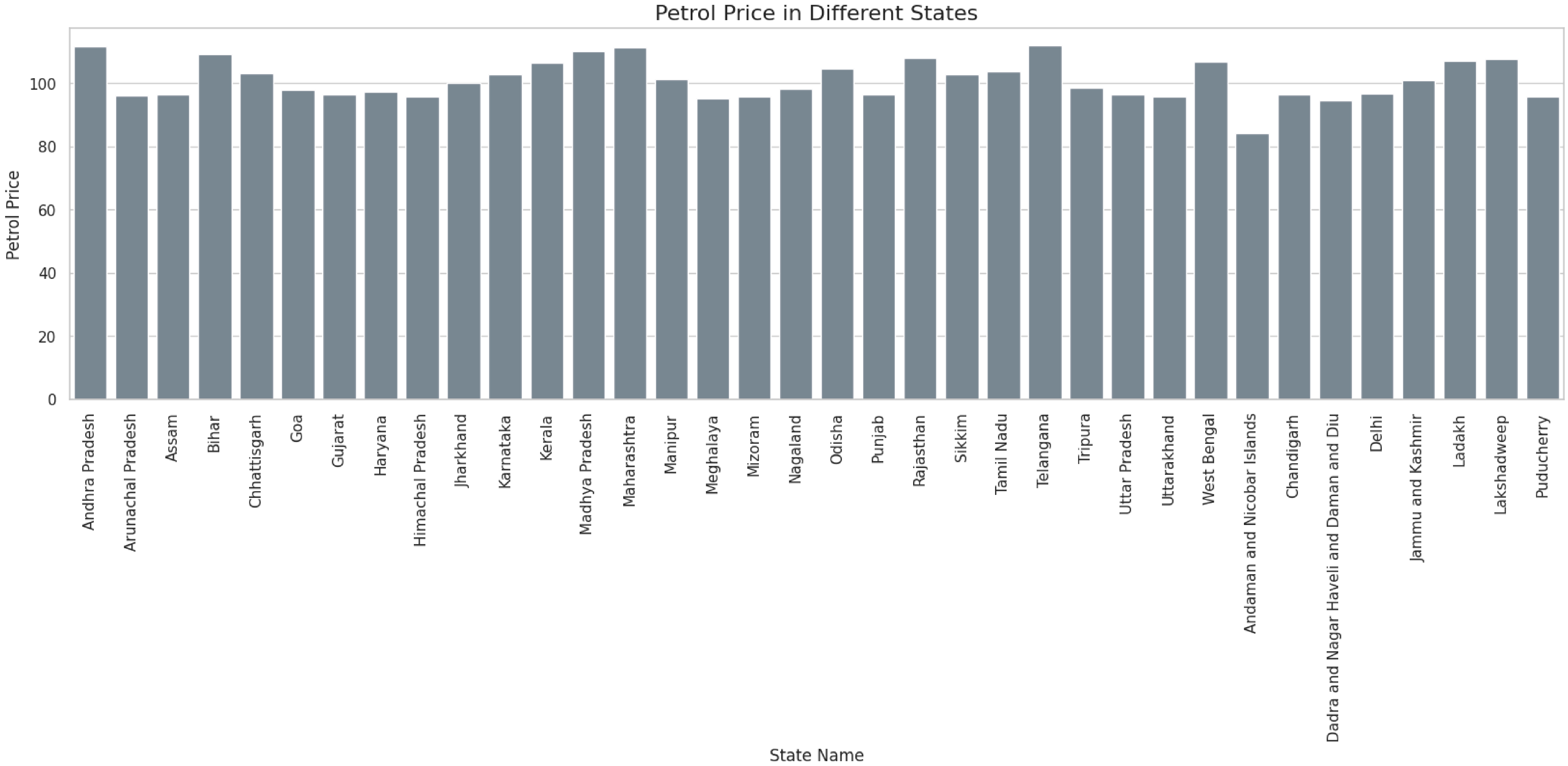
In [81]:

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



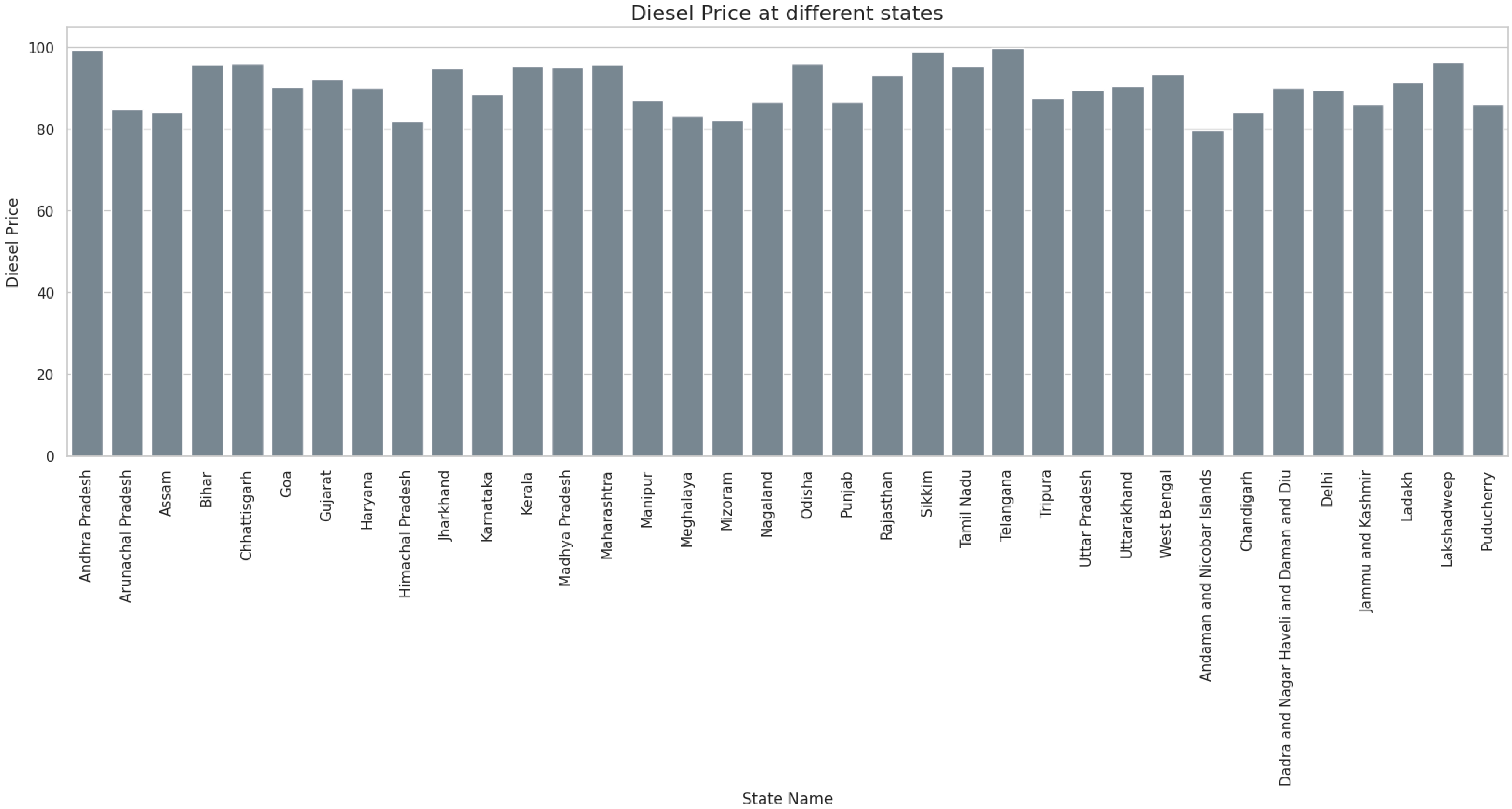
In [83]:

```
plt.figure(figsize=(20, 5))
sns.set_theme(style="whitegrid")
sns.barplot(x = 'state',y = 'petrol', data=dataframe, 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()
```



In [84]:

```
plt.figure(figsize=(20, 6))
sns.set_theme(style="whitegrid")
sns.barplot(x = 'state', y = 'diesel', data=dataframe, 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()
```



In [ ]:

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.

GitHub Link : https://github.com/prabhrajsingh/FeynnLabs/blob/main/ev-market-in-india.ipynb