Electric Vehicle market in India using Segmentation analysis

TEAM MEMBERS:

ROHIT PARIDA

VYSYARAJU VENUGOPALA RAJU

RIYA CHAUHAN

GithHub Link: https://github.com/Rohit-33/Feynn-Labs

November 2022

TABLE OF CONTENTS

1. Abstract……3
2. Introduction……....4
3. Problem Statement…….....5
4. Customer Segments…….....5
5. Fermi Estimations…….....6
6. Data Sources……...6
7. Environment and Tools……..6
8. Importing Libraries……6
9. Data Pre-Processing……...7
10. Exploratory Analysis…….....9
11. RO's Estimation……...15
12. Sanctioned Estimation……..17
13. K means clustering……...18
14. Conclusion……23
15. References……..…..25

ABSTRACT

With global warming on the rise, adopting ecologically sound decisions and averting climate change are critical. Electric vehicles (EVs) are one such environmentally beneficial option. The global automobile industry is undergoing a paradigm shift as it strives to migrate to alternate, less energy-intensive choices. The rise in oil import prices, growing pollution, and worldwide agreements to combat global climate change are among the key factors for India's recent initiatives to hasten the transition to e-mobility. As a result, during the Conference of the Parties 26 (COP26) Summit, India agreed to an aspirational target of having at least 30% of private autos be EVs by 2030.

By utilising an integrated research framework of "perceived benefits-attitude-intention," the primary goal of this study is to analyse and identify several sets of possible buyer groups for EVs based on psychographic, behavioural, and socioeconomic characterisation. To operationalize and validate segments from the data gathered from Press Information Bureau, India, the study used a rigorous analytical process called cluster analysis. The findings posit that the total number of EVs sold depends on two aspects: "number of EV charged sanctioned" and "number of RO's for EV"; these two features in turn rely on consumer behaviour when buying EVs.

INTRODUCTION

The India electric vehicle market was worth USD 220.1 million in 2020 and is predicted to increase at a compound annual growth rate (CAGR) of 94.4% between 2021 and 2030. The substantial incentives being granted by the Indian government on the manufacturing and purchase of electric cars to encourage their adoption are expected to fuel market expansion during the forecast period. The breakout of the COVID-19 pandemic caused a dramatic drop in both passenger and commercial vehicle sales in 2020. However, electric car sales in India were unaffected. The post-lockdown selling of pure and hybrid electric vehicles is a significant driving element in India's electric vehicle sector. The government's strict Greenhouse gas (GHG) emission rules, such as the Bharat Stage (BS) VI emission standards adopted by India's Ministry of Road Transport and Highways (MoRTH), are also anticipated to significantly contribute to the market's expansion. The growth of electrification in vehicles is anticipated to be accelerated by the rising costs of conventional fuel. The government's strict pollution regulations and Indian customers' rising environmental consciousness are also anticipated to increase demand for electric vehicles. Additionally, substantial attempts have been made by Indian manufacturers to include electrified vehicles in their product line up, including Tata Motors and Mahindra & Mahindra Ltd. This is likely to influence Indian consumers to choose electric vehicles. The expansion of the electric vehicle market in India during the anticipated term is encouraged by all of these factors.

The marketing manager uses market segmentation as a tool for decision-making when it comes to choosing a target market for a particular product and creating an effective marketing mix. Marketing's goal is to connect customers' true requirements and wants with suppliers' offerings that are best suited to meet their wants and needs. A company's marketing planning process is driven by this matching process, which is advantageous to both suppliers and customers. Simply grouping clients based on shared qualities is customer segmentation. Geography, demographics, behaviour, purchasing power, situational conditions, personality, way of life, psychographic, etc. are some of these characteristics. Customer segmentation aims to allocate resources by creating marketing programmes or measures, boosting customer profitability, enhancing customer happiness, and improving target marketing metrics.

An effective method for client segmentation is clustering. In a given dataset, clustering groups similar data points together. Each of these collections is referred to as a cluster. Although the items in each cluster are similar to one another, they are different from the items in other groupings. Unsupervised learning is a category that includes data mining techniques like clustering. This is due to the fact that it can extract patterns and data from unlabelled data. It is widely used in pattern recognition, classification, and machine learning. K means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset.

The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major applications of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.

PROBLEM STATEMENT:

Analysing the Electric Vehicle market in India using Segmentation analysis and to propose a feasible strategy to enter the market, targeting the segments most likely to use Electric vehicles.

(CUSTOMER/VEHICLE/B2B) SEGMENTS:

The following are the most typical ways that businesses divide up their clientele:

1. **Demographic information**, such as gender, age, familial and marital status, income, education, and occupation.
2. **Geographical information**, which differs depending on the scope of the company. For localized businesses, this info might pertain to specific towns or counties. For larger companies, it might mean a customer's city, state, or even country of residence.
3. **Psychographics**, such as social class, lifestyle, and personality traits.
4. **Behavioural data**, such as spending and consumption habits, product/service usage, and desired benefits.

Fermi Estimation:

We require to estimate the growth of the EV Vehicles in long race and based on that strategy and target segment require.

Data Sources:

Collected and used data source of EV Vehicles.

Environment and tools:

1. scikit-learn
2. seaborn
3. numpy
4. pandas
5. matplotlib

Importing Libraries:

Importing Electric Vehicle Data

In [200]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("EV.csv")
data.head()
```

Out[200]:

	State Name	Total Electric Vehicle	Total Non-Electric Vehicle	Total	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available
0	Andaman & Nicobar Island	162	1,46,945	1,47,107	0	10	0
1	Arunachal Pradesh	20	2,52,965	2,52,985	0	0	4
2	Assam	64,766	46,77,053	47,41,819	0	20	19
3	Bihar	83335	1,04,07,078	1,04,90,413	0	37	26
4	Chandigarh	2,812	7,46,881	7,49,693	48	70	1

DATA PRE-PROCESSING:

Data Pre-processing

In [201]:
```
data['Total Electric Vehicle'] = data['Total Electric Vehicle'].str.replace(',' , '')
data['Total Non-Electric Vehicle'] = data['Total Non-Electric Vehicle'].str.replace(',' , '')
data['Total'] = data['Total'].str.replace("," , "")
print(len(data))
data.head()
```

```
34
```

Out[201]:

	State Name	Total Electric Vehicle	Total Non-Electric Vehicle	Total	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available
0	Andaman & Nicobar Island	162	146945	147107	0	10	0
1	Arunachal Pradesh	20	252965	252985	0	0	4
2	Assam	64766	4677053	4741819	0	20	19
3	Bihar	83335	10407078	10490413	0	37	26
4	Chandigarh	2812	746881	749693	48	70	1

In [202]:
```
data['Total Electric Vehicle'] = data['Total Electric Vehicle'].astype(int)
data['Total Non-Electric Vehicle'] = data['Total Non-Electric Vehicle'].astype(int)
data['Total'] = data['Total'].astype(int)
data.dtypes
```

Out[202]:
```
State Name                                 object
Total Electric Vehicle                      int32
Total Non-Electric Vehicle                  int32
Total                                       int32
Charging Stations                           int64
No. of EV chargers sanctioned               int64
No of RO's where EV Charging Facility available    int64
dtype: object
```

In [203]:
```
State = pd.get_dummies(data['State Name'],drop_first=True)
data = pd.concat([data,State], axis = 1)

data.head()
```

Out[203]:

	State Name	Total Electric Vehicle	Total Non-Electric Vehicle	Total	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available	Andhra Pradesh	Arunachal Pradesh	Assam	...	Punjab	Rajasthan	Sikkim	Tamil Nadu	Telangana
0	Andaman & Nicobar Island	162	146945	147107	0	10	0	0	0	0	...	0	0	0	0	0
1	Arunachal Pradesh	20	252965	252985	0	0	4	0	1	0	...	0	0	0	0	0
2	Assam	64766	4677053	4741819	0	20	19	0	0	1	...	0	0	0	0	0
3	Bihar	83335	10407078	10490413	0	37	26	0	0	0	...	0	0	0	0	0
4	Chandigarh	2812	746881	749693	48	70	1	0	0	0	...	0	0	0	0	0

5 rows × 40 columns

In [204]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 40 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   State Name                                  34 non-null     object
 1   Total Electric Vehicle                      34 non-null     int32
 2   Total Non-Electric Vehicle                  34 non-null     int32
 3   Total                                       34 non-null     int32
 4   Charging Stations                           34 non-null     int64
 5   No. of EV chargers sanctioned               34 non-null     int64
 6   No of RO's where EV Charging Facility available  34 non-null     int64
 7   Andhra Pradesh                              34 non-null     uint8
 8   Arunachal Pradesh                           34 non-null     uint8
 9   Assam                                       34 non-null     uint8
 10  Bihar                                       34 non-null     uint8
 11  Chandigarh                                  34 non-null     uint8
 12  Chhattisgarh                                34 non-null     uint8
 13  Delhi                                       34 non-null     uint8
 14  Goa                                         34 non-null     uint8
 15  Gujarat                                     34 non-null     uint8
 16  Haryana                                     34 non-null     uint8
 17  Himachal Pradesh                            34 non-null     uint8
 18  Jammu and Kashmir                           34 non-null     uint8
 19  Jharkhand                                   34 non-null     uint8
 20  Karnataka                                   34 non-null     uint8
 21  Kerala                                      34 non-null     uint8
 22  Ladakh                                      34 non-null     uint8
 23  Maharashtra                                 34 non-null     uint8
 24  Manipur                                     34 non-null     uint8
 25  Meghalaya                                   34 non-null     uint8
 26  Mizoram                                     34 non-null     uint8
 27  Nagaland                                    34 non-null     uint8
 28  Odisha                                      34 non-null     uint8
 29  Puducherry                                  34 non-null     uint8
 30  Punjab                                      34 non-null     uint8
 31  Rajasthan                                   34 non-null     uint8
 32  Sikkim                                      34 non-null     uint8
 33  Tamil Nadu                                  34 non-null     uint8
 34  Telangana                                   34 non-null     uint8
 35  Tripura                                     34 non-null     uint8
 36  UT of DNH and DD                            34 non-null     uint8
 37  Uttar Pradesh                               34 non-null     uint8
 38  Uttarakhand                                 34 non-null     uint8
 39  West Bengal                                 34 non-null     uint8
dtypes: int32(3), int64(3), object(1), uint8(33)
memory usage: 2.7+ KB
```

In [205]:
```
data.describe()
```

Out[205]:

	Total Electric Vehicle	Total Non-Electric Vehicle	Total	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	...	Punjab	Rajasthan
count	34.000000	3.400000e+01	3.400000e+01	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000	...	34.000000	34.000000
mean	40066.647059	8.632410e+06	8.672476e+06	12.764706	89.882353	41.470588	0.029412	0.029412	0.029412	0.029412	...	0.029412	0.029412
std	66518.102854	1.059540e+07	1.064771e+07	23.936860	96.509621	49.115347	0.171499	0.171499	0.171499	0.171499	...	0.171499	0.171499
min	20.000000	3.830200e+04	3.832800e+04	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
25%	733.250000	5.369995e+05	5.397545e+05	0.000000	0.000000	2.250000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
50%	15807.500000	3.994547e+06	4.042434e+06	0.000000	22.500000	20.500000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
75%	47893.250000	1.371638e+07	1.375666e+07	14.250000	121.500000	69.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
max	337180.000000	4.009249e+07	4.042967e+07	94.000000	317.000000	174.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000

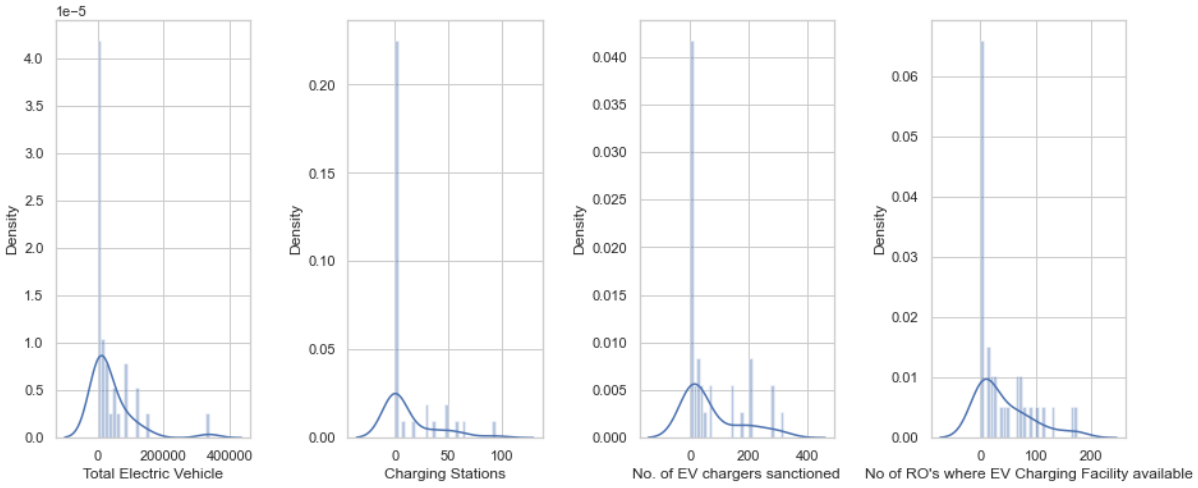
8 rows × 39 columns

In [206]:
```
# data.drop(columns =['State Name','Total Electric Vehicle' , 'Total'],axis = 1)
```

EXPLORATORY ANALYSIS:

1. Finding Gaussian Distribution of each feature:

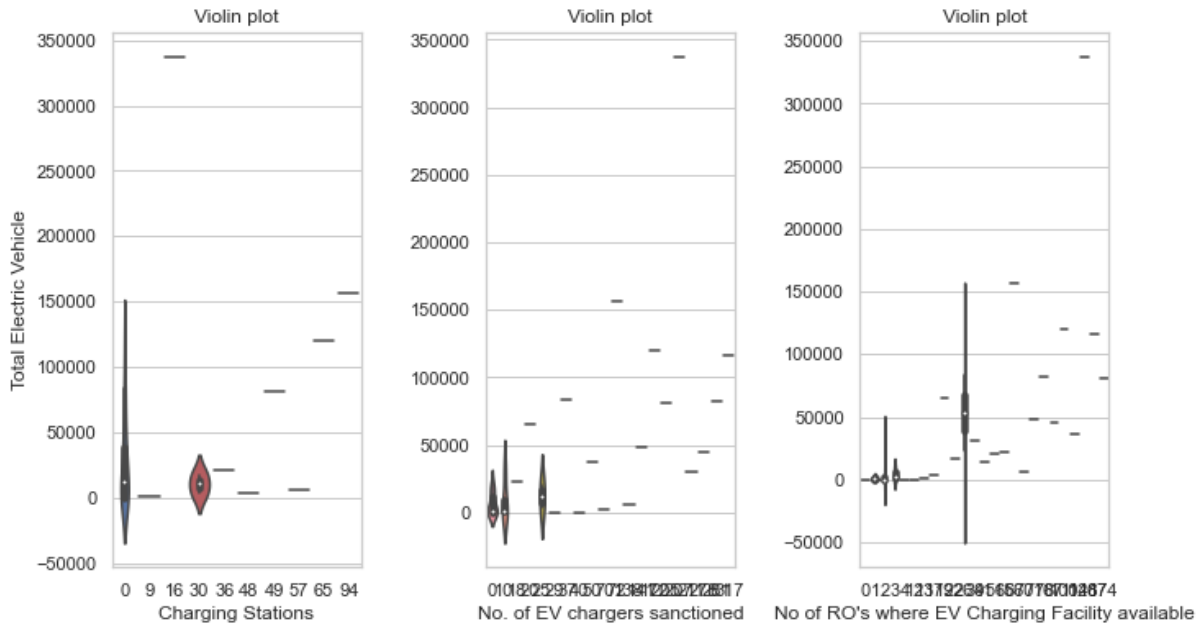
In [207]:
```
plt.figure(1,figsize=(15,6))
n=0
for x in ['Total Electric Vehicle','Charging Stations','No. of EV chargers sanctioned',
          "No of RO's where EV Charging Facility available"]:
    n += 1
    plt.subplot(1,4,n)
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.distplot(data[x] , bins=30)
plt.show()
```




```
In [208]: plt.figure(1,figsize=(15,6))
          n=0
          for cols in ['Charging Stations','No. of EV chargers sanctioned', "No of RO's where EV Charging Facility available"]:
              n += 1
              plt.subplot(1,4,n)
              sns.set(style= "whitegrid")
              plt.subplots_adjust(hspace=0.5,wspace=0.5)
              sns.violinplot(x=cols , y='Total Electric Vehicle',data= data)
              plt.ylabel("Total Electric Vehicle" if n==1 else "")
              plt.title("Violin plot")
          plt.show()
```

Result: Here we get that Total Electric Vehicle has the highest distribution.

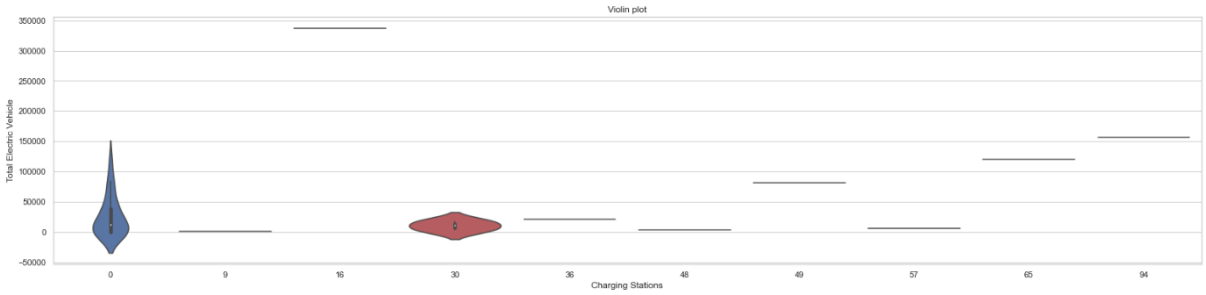
2. **<u>Here we get that most of the charging station has highest density around 16-30 with EV sold under 50000.</u>**



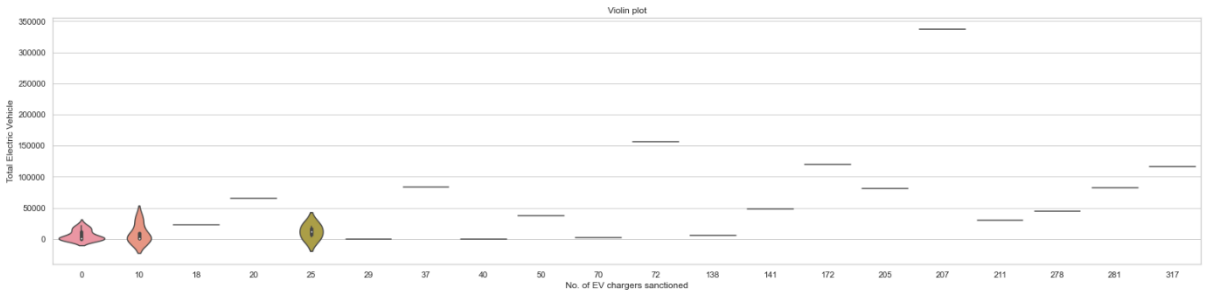
3.

In [209]:
```
plt.figure(1,figsize=(150,6))

plt.subplot(1,4,1)
sns.set(style= "whitegrid")
plt.subplots_adjust(hspace=0.5,wspace=0.5)
sns.violinplot(x="Charging Stations", y='Total Electric Vehicle',data= data)
plt.ylabel("Total Electric Vehicle" )
plt.title("Violin plot")
plt.show()
```

2-1 . More detail density distribution of each attribute with total EV sold.



In [210]:
```
plt.figure(1,figsize=(150,6))

plt.subplot(1,4,1)
sns.set(style= "whitegrid")
plt.subplots_adjust(hspace=0.5,wspace=0.5)
sns.violinplot(x="No. of EV chargers sanctioned", y='Total Electric Vehicle',data= data)
plt.ylabel("Total Electric Vehicle")
plt.title("Violin plot")
plt.show()
```

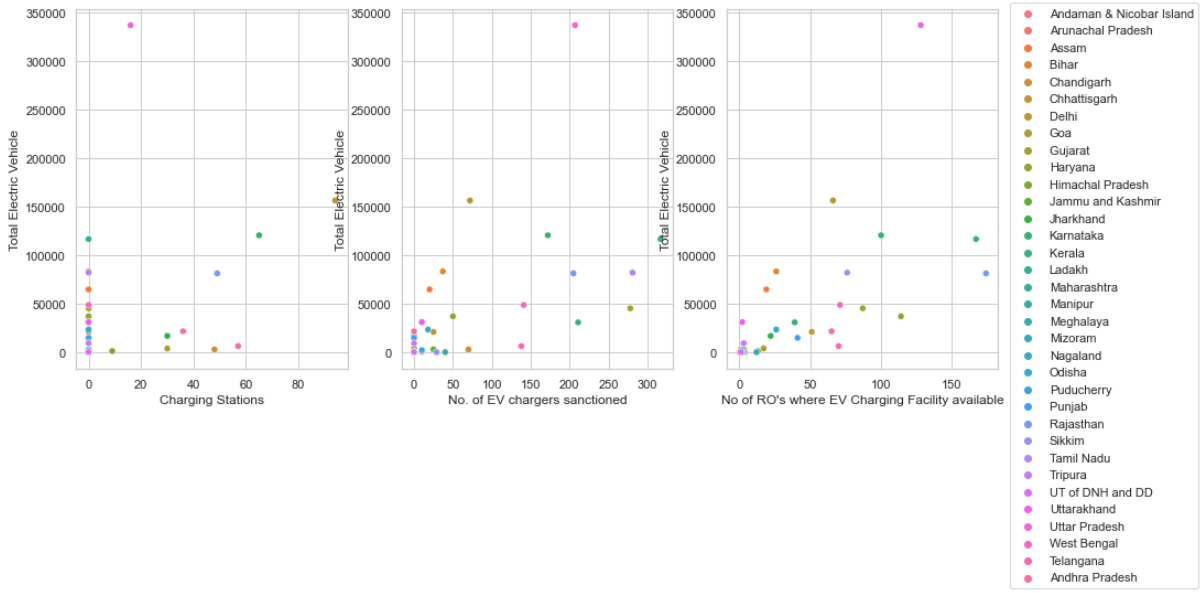


In [211]:
```
plt.figure(1,figsize=(100,50))

plt.subplot(1,4,1)
sns.set(style= "whitegrid")
plt.subplots_adjust(hspace=0.5,wspace=0.5)
sns.violinplot(x="No of RO's where EV Charging Facility available", y='Total Electric Vehicle',data= data)
plt.ylabel("Total Electric Vehicle" )
plt.title("Violin plot")
plt.show()
```


Violin plot
Total Electric Vehicle
No of RO's where EV Charging Facility available
350000
300000
250000
200000
150000
100000
50000
0
-50000
0
1
2
3
4
12
13
17
19
22
26
39
41
51
65
66
70
71
76
87
100
114
128
167
174

4. Done Scatterplot of each feature with Total EV sold to find the pattern i.e Linear or Non Linear relationship.

In [212]:
```
fig , ax = plt.subplots(nrows = 1 , ncols = 3 , figsize = (15,6))
sns.scatterplot(x='Charging Stations',y='Total Electric Vehicle',data= data , hue = "State Name",legend = False ,ax = ax[0])
sns.scatterplot(x="No. of EV chargers sanctioned", y='Total Electric Vehicle',data= data , hue = "State Name",legend = False,
                ax=ax[1])
sns.scatterplot(x="No of RO's where EV Charging Facility available", y='Total Electric Vehicle',data= data ,
                hue = "State Name",ax=ax[2])
plt.legend(loc="upper left" , bbox_to_anchor=(1,1.05) , borderaxespad=1)
```



Result:- Found that “No . of RO’s where available” is having linear relationship with Total EV sold.

5. Drop bias feature in Dataset

In [213]:
```
data.drop(columns=['State Name' , 'Total Non-Electric Vehicle','Total'] , axis =1 , inplace=True)
```

In [214]:
```
data.head()
```

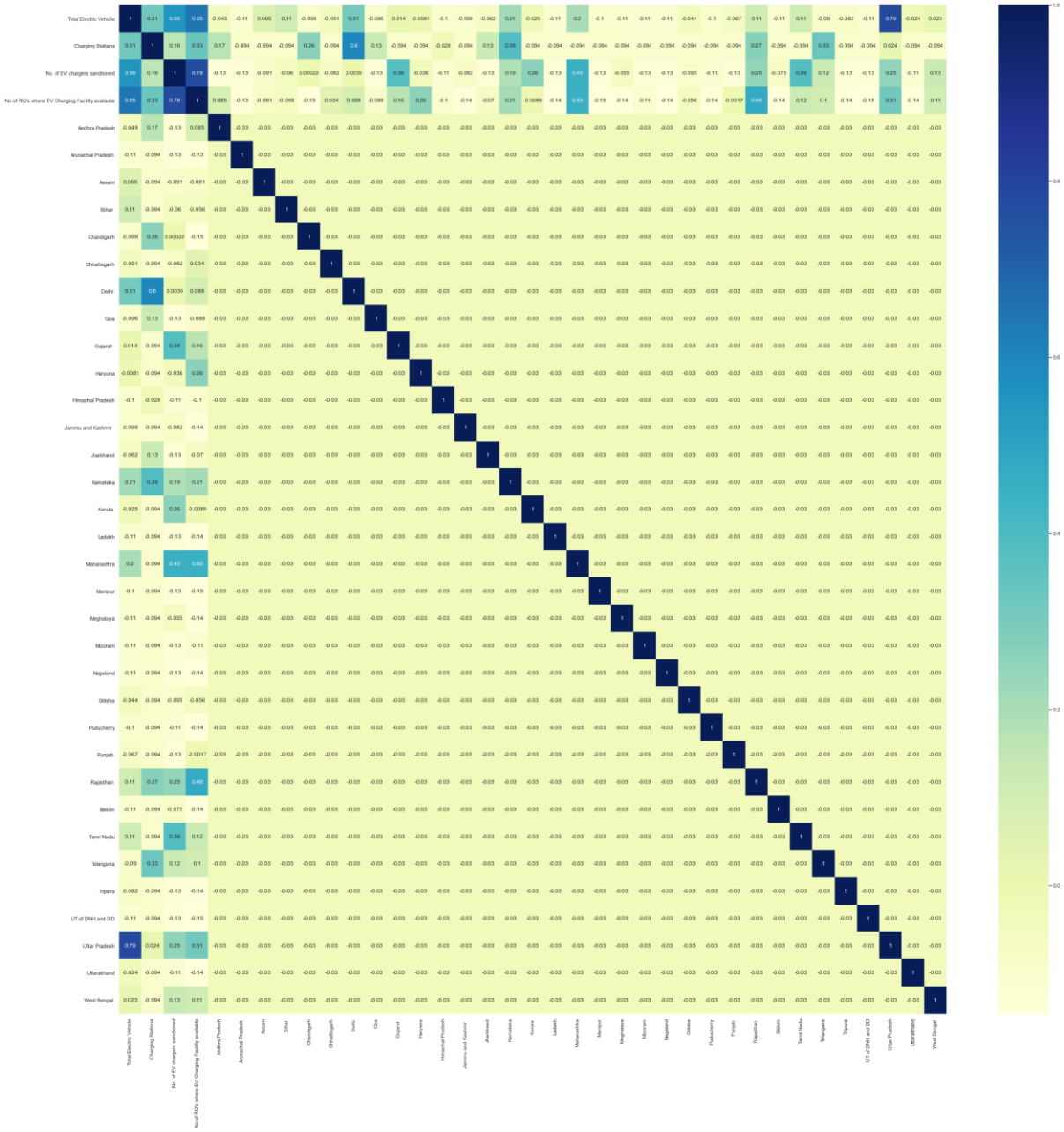
Out[214]:

	Total Electric Vehicle	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chandigarh	Chhattisgarh	...	Punjab	Rajasthan	Sikkim	Tamil Nadu	Telangana
0	162	0	10	0	0	0	0	0	0	0	...	0	0	0	0	0
1	20	0	0	4	0	1	0	0	0	0	...	0	0	0	0	0
2	64766	0	20	19	0	0	1	0	0	0	...	0	0	0	0	0
3	83335	0	37	26	0	0	0	1	0	0	...	0	0	0	0	0
4	2812	48	70	1	0	0	0	0	1	0	...	0	0	0	0	0

5 rows × 37 columns

6. Find the Correlation of each feature with each other so that high correlated feature have the most effect on the EV sold.

In [215]:
```
plt.figure(figsize=(40 , 40))
sns.heatmap(data.corr() , annot= True,cmap="YlGnBu")
```


Total Electric Vehicle
Charging Stations
No. of EV chargers sanctioned
No of RO's where EV Charging Facility available
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
UT of DNH and DD
Uttar Pradesh
Uttarakhand
West Bengal
1.0
0.8
0.6
0.4
0.2
0.0

```
In [216]: data.corr()
```

	Total Electric Vehicle	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chandigarh	Chhattisgarh	...	Punjab	Rajasthan	Sik
Total Electric Vehicle	1.000000	0.312680	0.556612	0.647558	-0.049147	-0.106378	0.065610	0.114936	-0.098961	-0.050738	...	-0.067106	0.109631	-0.106
Charging Stations	0.312680	1.000000	0.161240	0.333654	0.171517	-0.094226	-0.094226	-0.094226	0.260098	-0.094226	...	-0.094226	0.267479	-0.094
No. of EV chargers sanctioned	0.556612	0.161240	1.000000	0.775094	-0.127945	-0.127945	-0.091328	-0.060203	0.000215	-0.082173	...	-0.127945	0.247381	-0.074
No of RO's where EV Charging Facility available	0.647558	0.333654	0.775094	1.000000	0.084649	-0.134803	-0.080839	-0.055656	-0.145595	0.034283	...	-0.001693	0.476783	-0.141
Andhra Pradesh	-0.049147	0.171517	-0.127945	0.084649	1.000000	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Arunachal Pradesh	-0.106378	-0.094226	-0.127945	-0.134803	-0.030303	1.000000	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Assam	0.065610	-0.094226	-0.091328	-0.080839	-0.030303	-0.030303	1.000000	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Bihar	0.114936	-0.094226	-0.060203	-0.055656	-0.030303	-0.030303	-0.030303	1.000000	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Chandigarh	-0.098961	0.260098	0.000215	-0.145595	-0.030303	-0.030303	-0.030303	-0.030303	1.000000	-0.030303	...	-0.030303	-0.030303	-0.030
Chhattisgarh	-0.050738	-0.094226	-0.082173	0.034283	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	1.000000	...	-0.030303	-0.030303	-0.030
Delhi	0.309004	0.599657	0.003877	0.088246	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Goa	-0.096151	0.127226	-0.127945	-0.088034	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Gujarat	0.013827	-0.094226	0.381034	0.163795	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Haryana	-0.008053	-0.094226	-0.036402	0.260929	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Himachal Pradesh	-0.103310	-0.027790	-0.109636	-0.102425	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Jammu and Kashmir	-0.098619	-0.094226	-0.082173	-0.138400	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Jharkhand	-0.061775	0.127226	-0.127945	-0.070047	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Karnataka	0.213744	0.385587	0.186963	0.210563	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Kerala	-0.024682	-0.094226	0.258367	-0.008888	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Ladakh	-0.106362	-0.094226	-0.127945	-0.141998	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Maharashtra	0.203422	-0.094226	0.452438	0.451600	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Manipur	-0.104875	-0.094226	-0.127945	-0.145595	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Meghalaya	-0.106301	-0.094226	-0.054710	-0.138400	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Mizoram	-0.106375	-0.094226	-0.127945	-0.106022	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Nagaland	-0.106277	-0.094226	-0.127945	-0.141998	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Odisha	-0.044350	-0.094226	-0.094989	-0.055656	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Puducherry	-0.100723	-0.094226	-0.109636	-0.141998	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Punjab	-0.067106	-0.094226	-0.127945	-0.001693	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	1.000000	-0.030303	-0.030
Rajasthan	0.109631	0.267479	0.247381	0.476783	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	1.000000	-0.030
Sikkim	-0.106375	-0.094226	-0.074850	-0.141998	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	1.000
Tamil Nadu	0.111525	-0.094226	0.386527	0.124222	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Telangana	-0.089654	0.326533	0.124714	0.102636	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030
Tripura	-0.081828	-0.094226	-0.127945	-0.138400	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	-0.030303	...	-0.030303	-0.030303	-0.030

Result :- As we can interpret

1- No of EV charged sanctioned
2- No of RO’s

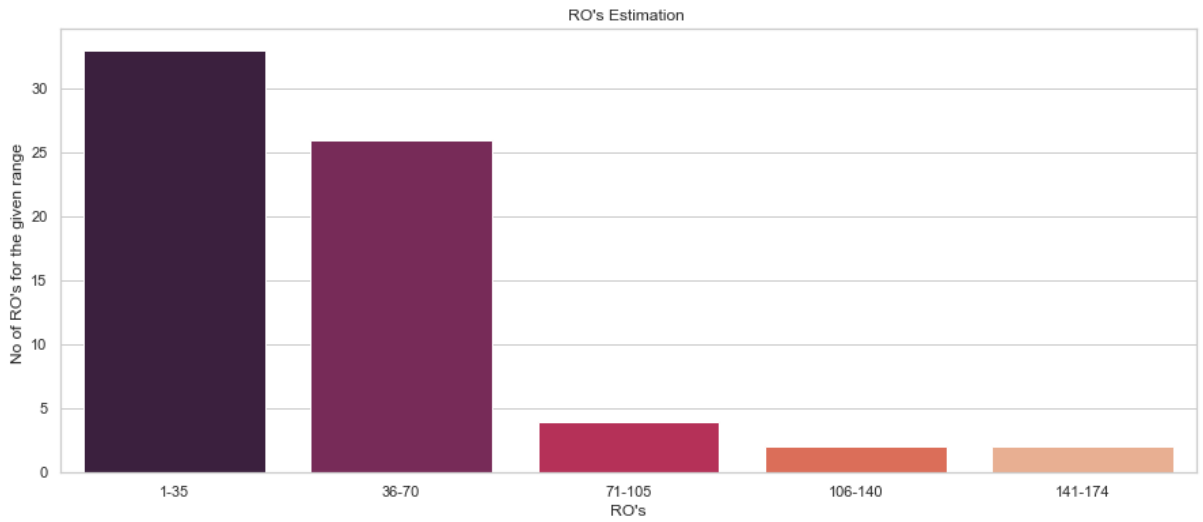
Are mostly correlated with the EV sold

7. Finding the total RO's range and sanctioned range using barplot .

6.1 RO's Estimation:

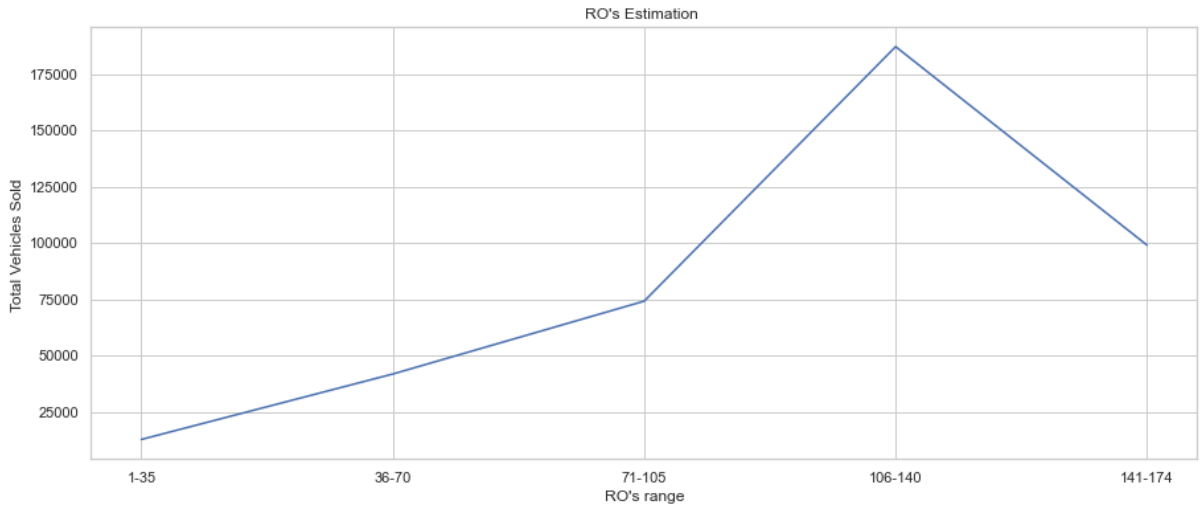
RO's Estimation

```
In [217]: ro_1_35 = data["No of RO's where EV Charging Facility available"][(data["No of RO's where EV Charging Facility available"]>=1)]
          ro_36_70 = data["No of RO's where EV Charging Facility available"][(data["No of RO's where EV Charging Facility available"]<= 70)
          ro_71_105 = data["No of RO's where EV Charging Facility available"][(data["No of RO's where EV Charging Facility available"]>=71)
                                                                              & (data["No of RO's where EV Charging Facility available"]<= 105
          ro_106_140 = data["No of RO's where EV Charging Facility available"][(data["No of RO's where EV Charging Facility available"]>=10
                                                                              & (data["No of RO's where EV Charging Facility available"]<= 140
          ro_141_174 = data["No of RO's where EV Charging Facility available"][(data["No of RO's where EV Charging Facility available"]>=14
                                                                              & (data["No of RO's where EV Charging Facility available"]<= 174

          r_x = ["1-35","36-70","71-105","106-140","141-174"]
          r_y = [len(ro_1_35.values),len(ro_36_70.values),len(ro_71_105.values),len(ro_106_140.values),len(ro_141_174.values)]


          plt.figure(figsize=(15,6))
          sns.barplot(x=r_x , y = r_y,palette="rocket")
          plt.title("RO's Estimation ")
          plt.xlabel("RO's")
          plt.ylabel("No of RO's for the given range ")
          plt.show()
```



6.1.1 Drawn lineplot to show the how the No.of RO's from particular range effect the Total Eclectric Vehicle sold.

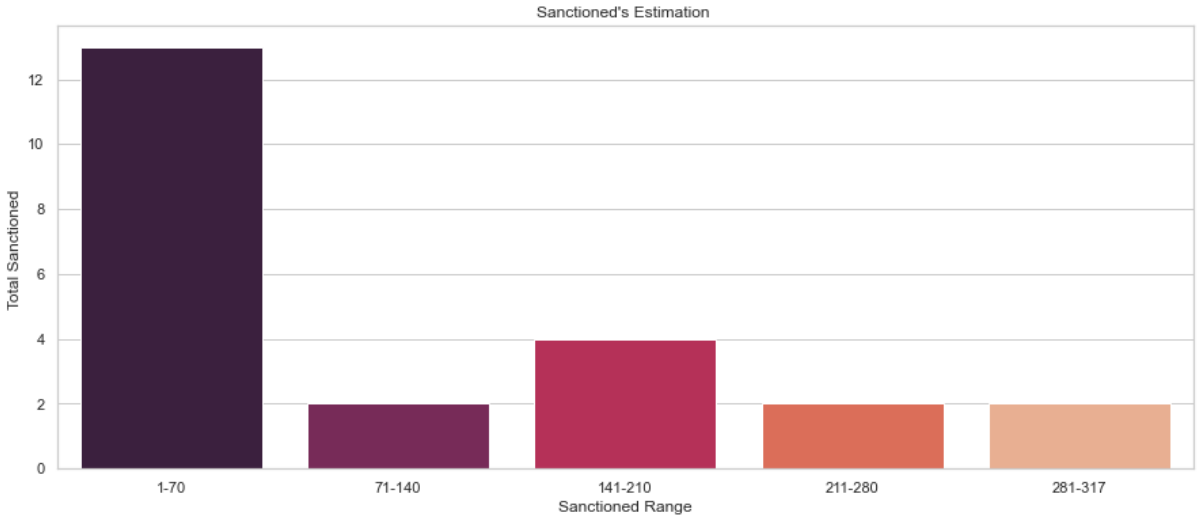
```
In [218]: ro_1_35 = data["Total Electric Vehicle"][(data["No of RO's where EV Charging Facility available"]>=1)
                                                   & (data["No of RO's where EV Charging Facility available"]<= 35)
          ro_36_70 = data["Total Electric Vehicle"][(data["No of RO's where EV Charging Facility available"]>=36)
                                                   & (data["No of RO's where EV Charging Facility available"]<= 70)
          ro_71_105 = data["Total Electric Vehicle"][(data["No of RO's where EV Charging Facility available"]>=71)
                                                   & (data["No of RO's where EV Charging Facility available"]<= 105
          ro_106_140 = data["Total Electric Vehicle"][(data["No of RO's where EV Charging Facility available"]>=106)
                                                   & (data["No of RO's where EV Charging Facility available"]<= 140
          ro_141_174 = data["Total Electric Vehicle"][(data["No of RO's where EV Charging Facility available"]>=141)
                                                   & (data["No of RO's where EV Charging Facility available"]<= 174



          ro_1_35 = ro_1_35.mean(axis = 0)
          ro_36_70 = ro_36_70.mean(axis = 0)
          ro_71_105 = ro_71_105.mean(axis = 0)
          ro_106_140 = ro_106_140.mean(axis = 0)
          ro_141_174 = ro_141_174.mean(axis = 0)
          r_x = ["1-35","36-70","71-105","106-140","141-174"]
          r_y = [ro_1_35,ro_36_70,ro_71_105,ro_106_140,ro_141_174]

          plt.figure(figsize=(15,6))
          sns.lineplot(x=r_x , y = r_y,palette="rocket")
          plt.title("RO's Estimation ")
          plt.xlabel("RO's range")
          plt.ylabel("Total Vehicles Sold ")
          plt.show()
```



Result:- As we can see it 70-40's RO's containing state are selling highest EV vehicle and form almost linear graph which will be our feature to ML algorithm.

6.2 . SANCTIONED ESTIMATION:

In [219]:
```
s_1_70 = data["No. of EV chargers sanctioned"][(data["No. of EV chargers sanctioned"]>=1)
                                               & (data["No. of EV chargers sanctioned"]<= 70)]
s_71_140 = data["No. of EV chargers sanctioned"][(data["No. of EV chargers sanctioned"]>=71)
                                               & (data["No. of EV chargers sanctioned"]<= 140)]
s_141_210 = data["No. of EV chargers sanctioned"][(data["No. of EV chargers sanctioned"]>=141)
                                               & (data["No. of EV chargers sanctioned"]<= 210)]
s_211_280 = data["No. of EV chargers sanctioned"][(data["No. of EV chargers sanctioned"]>=211)
                                               & (data["No. of EV chargers sanctioned"]<= 280)]
s_281_317 = data["No. of EV chargers sanctioned"][(data["No. of EV chargers sanctioned"]>=281)
                                               & (data["No. of EV chargers sanctioned"]<= 317)]
s_x = ["1-70","71-140","141-210","211-280","281-317"]
s_y = [len(s_1_70.values),len(s_71_140.values),len(s_141_210.values),len(s_211_280.values),len(s_281_317.values)]


plt.figure(figsize=(15,6))
sns.barplot(x=s_x , y = s_y,palette="rocket")
plt.title("Sanctioned's Estimation ")
plt.xlabel("Sanctioned Range")
plt.ylabel("Total Sanctioned ")
plt.show()
```



7 . K-Means Clustering

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters.

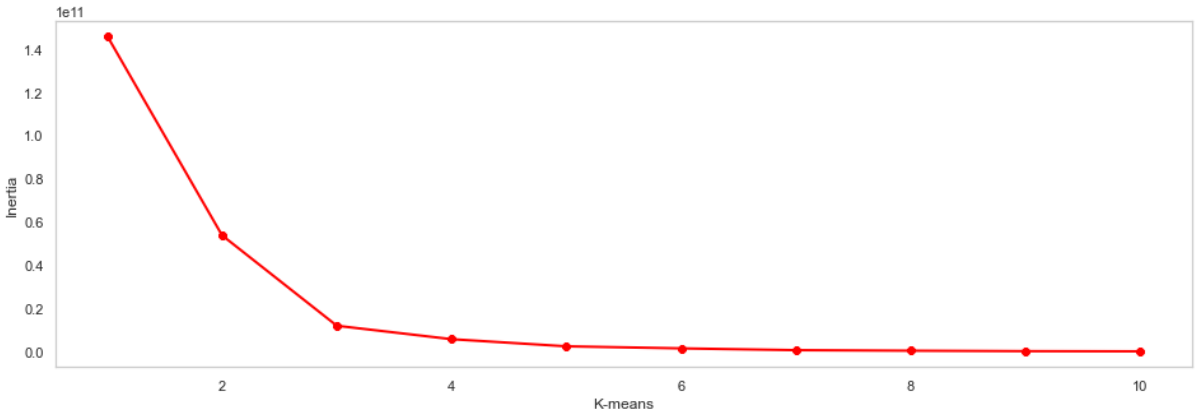
The value of k should be predetermined in this algorithm. We start by pre-processing the data and cleaning it. This essentially involves null-handling and label encoding the ordinal parameters of the data. The data is then passed into the Scikit-Learn K-Means Clustering model to obtain the elbow curve for the ideal number of clusters. Using the "elbow" or "knee of a curve" as a cutoff point is a common heuristic in mathematical optimization to choose a point where diminishing returns are no longer worth the additional cost. In clustering, this means one should choose a few clusters so that adding another cluster doesn't give much better modeling of the data. The intuition is that increasing the number of clusters will naturally improve the fit (explain more of the variation), since there are more parameters (more clusters) to use, but that at some point this is over-fitting, and the elbow reflects this.

K-Means Clustering with

x = (No. of EV chargers sanctioned, No of RO's where EV Charging Facility available)

y = (Total Electric Vehicle)

In [220]:
```
x1 = data.loc[:,["Total Electric Vehicle" , "No of RO's where EV Charging Facility available"]].values

from sklearn.cluster import KMeans
inertia = []

for i in range (1,11):
    kmeans = KMeans(n_clusters = i , init="k-means++")
    kmeans.fit(x1)
    inertia.append(kmeans.inertia_)
plt.figure(figsize=(16,5))
plt.grid()
plt.plot(range(1,11) , inertia , linewidth = 2, color = 'red' , marker='8')
plt.xlabel("K-means")
plt.ylabel("Inertia")
plt.show()
```



Using elbow method, we see that optimal cluster is at 3.

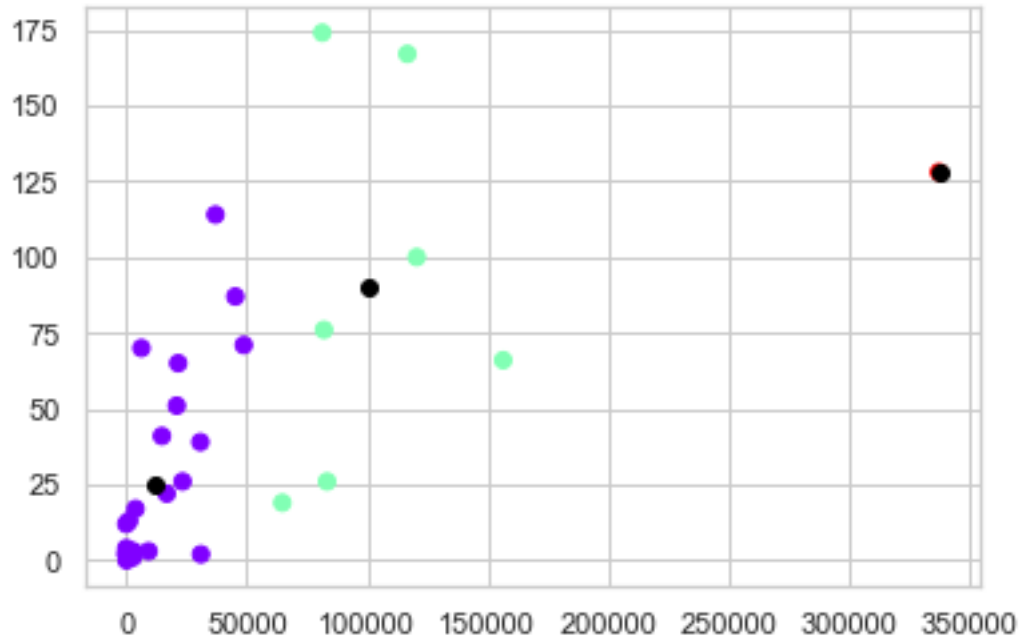
In [221]:
```
kmeans = KMeans(n_clusters = 3)
labels = kmeans.fit_predict(x1)
print(labels)
```

```
[0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 2 0 0 0]
```

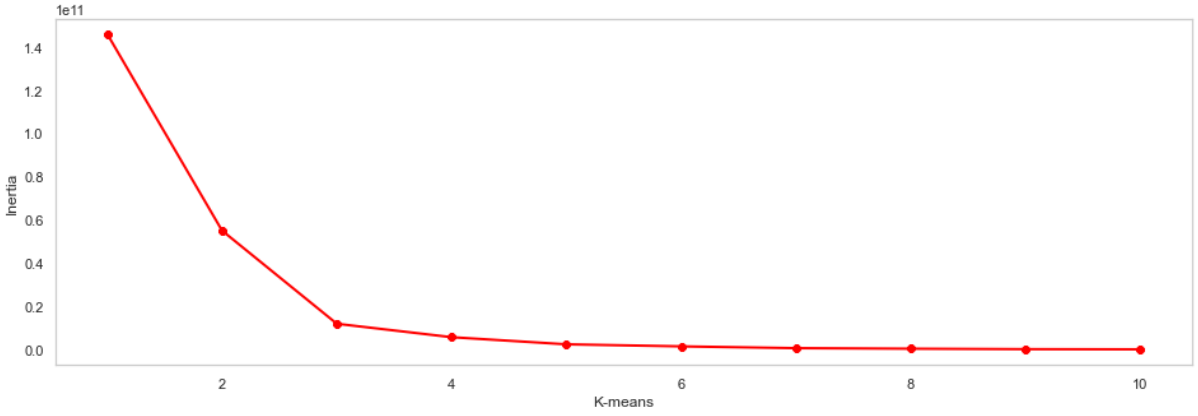
In [222]:
```
print(kmeans.cluster_centers_)
```

```
[[1.23086538e+04 2.51538462e+01]
 [1.00723000e+05 8.97142857e+01]
 [3.37180000e+05 1.28000000e+02]]
```

In [223]:
```
plt.scatter(x1[:,0] , x1[:,1] , c=labels , cmap= "rainbow")
plt.scatter(kmeans.cluster_centers_[:,0] , kmeans.cluster_centers_[:,1] , color="black")
plt.plot()
```



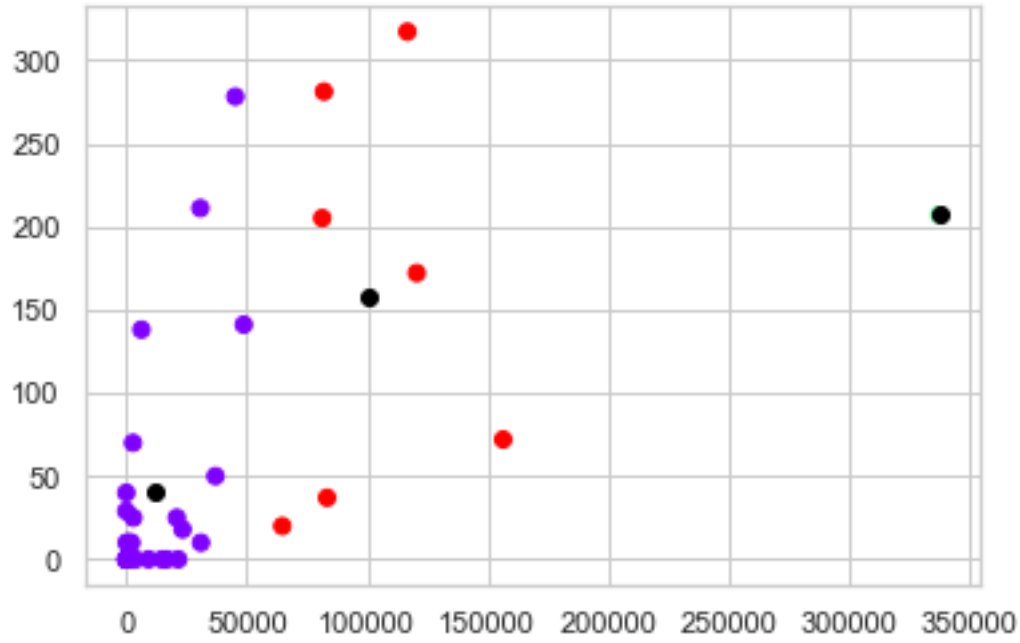
In [224]:
```
x2 = data.loc[: , ["Total Electric Vehicle" , "No. of EV chargers sanctioned"]].values

from sklearn.cluster import KMeans

inertia2 = []

for i in range (1,11):
    kmeans = KMeans(n_clusters = i , init = "k-means++")
    kmeans.fit(x2)
    inertia2.append(kmeans.inertia_)

plt.figure(figsize=(16,5))
plt.grid()
plt.plot(range(1,11) , inertia2 , color = "red" , linewidth = 2 , marker="8" )
plt.xlabel("K-means")
plt.ylabel("Inertia")
plt.show()
```



In [225]:
```
kmeans = KMeans(n_clusters = 3)
labels2 = kmeans.fit_predict(x2)
print(labels)
```

```
[0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 2 0 0 0]
```

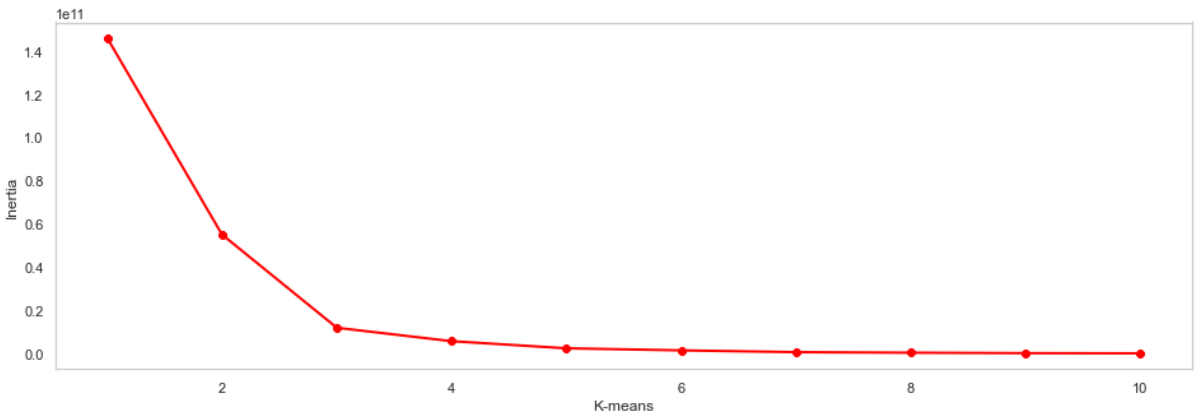
In [226]:
```
plt.scatter(x2[: ,0], x2[:,1] , c = labels2 , cmap = "rainbow")
plt.scatter(kmeans.cluster_centers_[: , 0] , kmeans.cluster_centers_[:,1] , color = "black")
plt.plot()
```



Finding Cluster for

"Total Electric Vehicle" ,"No of RO's where EV Charging Facility available", "No. of EV chargers sanctioned"

In [241]:
```
x3 = data.loc[: , ["Total Electric Vehicle" ,"No of RO's where EV Charging Facility available", "No. of EV chargers sanctioned"]]

from sklearn.cluster import KMeans

inertia3 = []

for i in range (1,11):
    kmeans = KMeans(n_clusters = i , init = "k-means++")
    kmeans.fit(x3)
    inertia3.append(kmeans.inertia_)

plt.figure(figsize=(16,5))
plt.grid()
plt.plot(range(1,11) , inertia3 , color = "red" , linewidth = 2 , marker="8" )
plt.xlabel("K-means")
plt.ylabel("Inertia")
plt.show()
```



Result :- We see that optimal cluster is 3.

In [242]:
```
kmeans = KMeans(n_clusters = 3)
labels2 = kmeans.fit_predict(x2)
print(labels)
print(kmeans.cluster_centers_)
```

```
[0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 2 0 0 0]
[[1.23086538e+04 4.09615385e+01]
 [3.37180000e+05 2.07000000e+02]
 [1.00723000e+05 1.57714286e+02]]
```

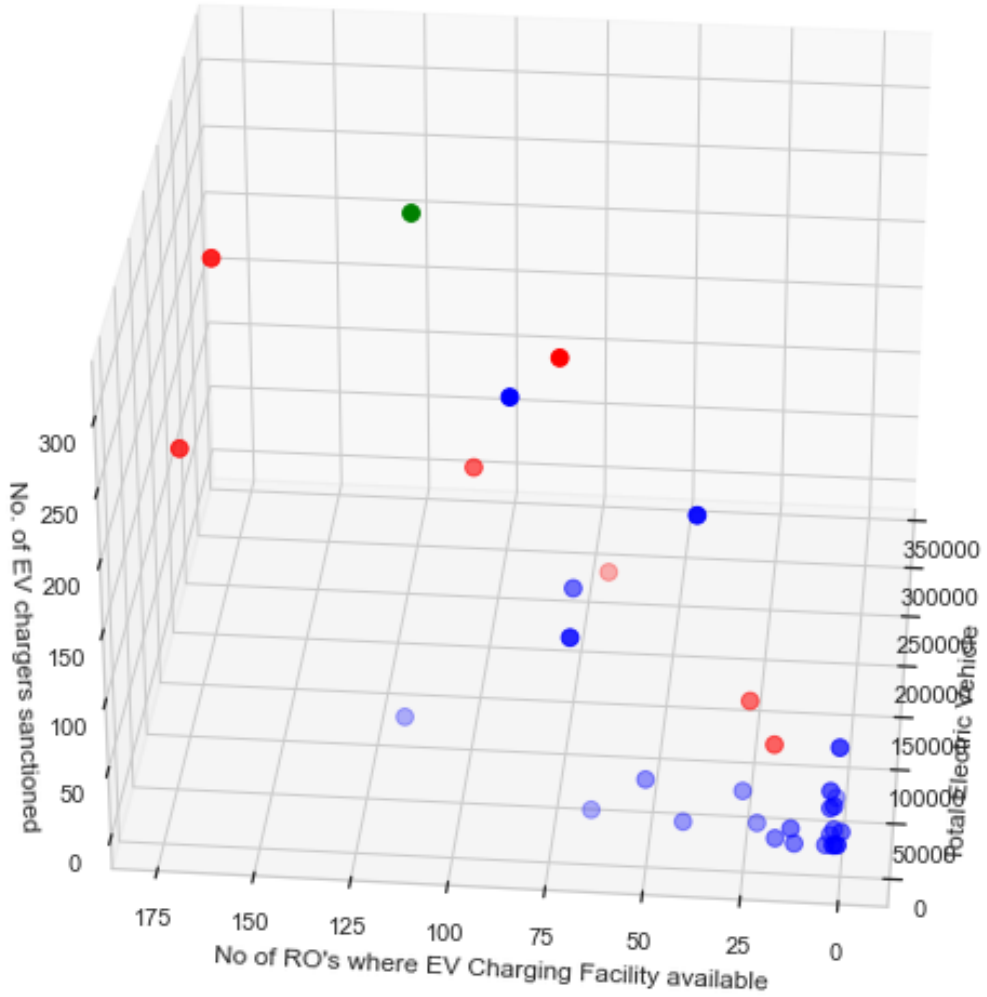
In [243]:
```
clusters = kmeans.fit_predict(x3)
data["labels"] = clusters
data.head()
```

Out[243]:

	Total Electric Vehicle	Charging Stations	No. of EV chargers sanctioned	No of RO's where EV Charging Facility available	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chandigarh	Chhattisgarh	...	Rajasthan	Sikkim	Tamil Nadu	Telangana	Tripura	D a
0	162	0	10	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	20	0	0	4	0	1	0	0	0	0	...	0	0	0	0	0	
2	64766	0	20	19	0	0	1	0	0	0	...	0	0	0	0	0	
3	83335	0	37	26	0	0	0	1	0	0	...	0	0	0	0	0	
4	2812	48	70	1	0	0	0	0	1	0	...	0	0	0	0	0	

5 rows × 38 columns

In [244]:
```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize =(15,10))
ax = fig.add_subplot(111, projection = "3d")
ax.scatter(data["Total Electric Vehicle"][data.labels == 0] , data["No of RO's where EV Charging Facility available"][data.labels
ax.scatter(data["Total Electric Vehicle"][data.labels == 1] , data["No of RO's where EV Charging Facility available"][data.labels
ax.scatter(data["Total Electric Vehicle"][data.labels == 2] , data["No of RO's where EV Charging Facility available"][data.labels
ax.view_init(30,185)

plt.xlabel("Total Electric Vehicle")
plt.ylabel("No of RO's where EV Charging Facility available")
ax.set_zlabel("No. of EV chargers sanctioned")
plt.show()
```



CONCLUSION:

As from the above inference we concluded that total EV sold is depend on the 2 features “No of EV charged sanctioned ” & “No of RO’s for EV” which in turn depend on the customer behaviour of purchasing the EV.

REFERENCES

1. Deepak Jaiswal, Arun Kumar Deshmukh, Park Thaichon, Who will adopt electric vehicles? Segmenting and exemplifying potential buyer heterogeneity and forthcoming research, Journal of Retailing and Consumer Services, Volume 67, 2022, 102969, ISSN 0969-6989, https://doi.org/10.1016/j.jretconser.2022.102969. (https://www.sciencedirect.com/science/article/pii/S0969698922000625)

2. Jeykishan Kumar K, Sudhir R Kumar, V. S. Nandakumar Standards for Electric Vehicle Charging Stations in India: A Review