```python
#Import required libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')

#Reading the .csv file
df=pd.read_csv(r"C:\Users\DELL\Downloads\electric dataset.csv")
df.head()
```

```
   VIN (1-10)     County      City State  Postal Code  Model Year  
Make  \
0  JTMEB3FV6N     Monroe  Key West    FL        33040        2022   
TOYOTA   
1  1G1RD6E45D      Clark  Laughlin    NV        89029        2013  
CHEVROLET   
2  JN1AZ0CP8B     Yakima    Yakima    WA        98901        2011   
NISSAN   
3  1G1FW6S08H     Skagit  Concrete    WA        98237        2017  
CHEVROLET   
4  3FA6P0SU1K  Snohomish   Everett    WA        98201        2019   
FORD   

        Model                   Electric Vehicle Type  \
0  RAV4 PRIME  Plug-in Hybrid Electric Vehicle (PHEV)   
1        VOLT  Plug-in Hybrid Electric Vehicle (PHEV)   
2        LEAF          Battery Electric Vehicle (BEV)   
3     BOLT EV          Battery Electric Vehicle (BEV)   
4      FUSION  Plug-in Hybrid Electric Vehicle (PHEV)   

  Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric Range  \
0           Clean Alternative Fuel Vehicle Eligible              42   
1           Clean Alternative Fuel Vehicle Eligible              38   
2           Clean Alternative Fuel Vehicle Eligible              73   
3           Clean Alternative Fuel Vehicle Eligible             238   
4             Not eligible due to low battery range              26   

   Base MSRP  Legislative District  DOL Vehicle ID  \
0          0                   NaN       198968248   
1          0                   NaN         5204412   
2          0                  15.0       218972519   
3          0                  39.0       186750406   
4          0                  38.0         2006714   

              Vehicle Location       Electric Utility  2020 Census 
Tract  
```

```
0    POINT (-81.80023 24.5545)                     NaN
12087972100
1  POINT (-114.57245 35.16815)                     NaN
32003005702
2  POINT (-120.50721 46.60448)              PACIFICORP
53077001602
3   POINT (-121.7515 48.53892)  PUGET SOUND ENERGY INC
53057951101
4  POINT (-122.20596 47.97659)  PUGET SOUND ENERGY INC
53061041500
```

```
#shape of the data
shape=df.shape
print("The Number of rows : {}".format(shape[0]))
print("The Number of columns : {}".format(shape[1]))
```

```
The Number of rows : 112634
The Number of columns : 17
```

```
# Information about the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 17 columns):
 #   Column                                              Non-Null Count
Dtype
---  ------                                              --------------
-----
 0   VIN (1-10)                                          112634 non-
null  object
 1   County                                              112634 non-
null  object
 2   City                                                112634 non-
null  object
 3   State                                               112634 non-
null  object
 4   Postal Code                                         112634 non-
null  int64
 5   Model Year                                          112634 non-
null  int64
 6   Make                                                112634 non-
null  object
 7   Model                                               112614 non-
null  object
 8   Electric Vehicle Type                               112634 non-
null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  112634 non-
null  object
```

```
 10  Electric Range                                   112634 non-
null  int64
 11  Base MSRP                                        112634 non-
null  int64
 12  Legislative District                             112348 non-
null  float64
 13  DOL Vehicle ID                                   112634 non-
null  int64
 14  Vehicle Location                                 112610 non-
null  object
 15  Electric Utility                                 112191 non-
null  object
 16  2020 Census Tract                                112634 non-
null  int64
dtypes: float64(1), int64(6), object(10)
memory usage: 14.6+ MB
```

Exploratory data Analysis

Getting the insights from the data which includes :

- Missing values.
- Duplicated Values.
- Outliers.
- Relationships.
- Distributions.

```
# Checking the Missing values
df.isna().sum()
```

```
VIN (1-10)                                             0
County                                                 0
City                                                   0
State                                                  0
Postal Code                                            0
Model Year                                             0
Make                                                   0
Model                                                 20
Electric Vehicle Type                                  0
Clean Alternative Fuel Vehicle (CAFV) Eligibility      0
Electric Range                                         0
Base MSRP                                              0
Legislative District                                 286
DOL Vehicle ID                                         0
Vehicle Location                                      24
Electric Utility                                     443
```

```
2020 Census Tract                                      0
dtype: int64
```

Insights:

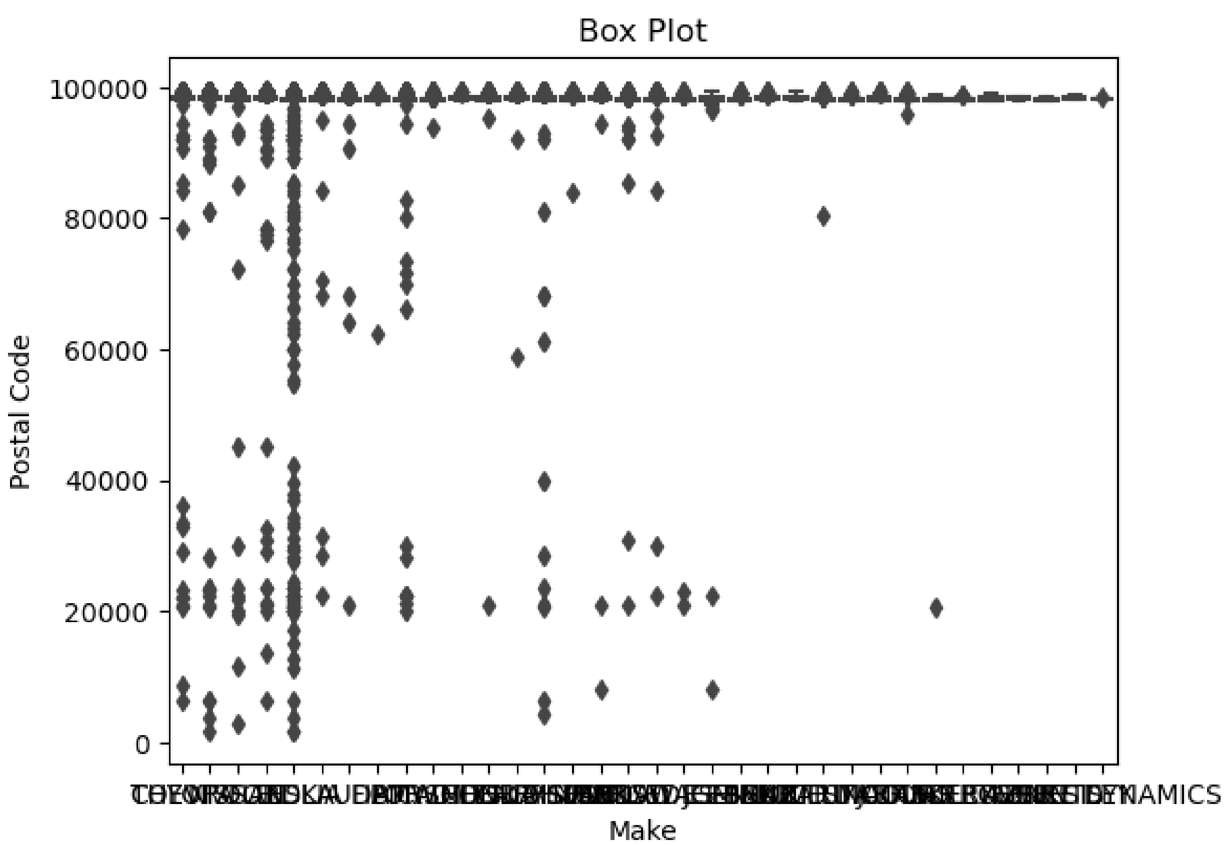
- There are 20 missing values in Model column.
- 286 missing values in Legislative District.
- 443 Missing values in Electric Utility.

```
# Checking the Duplicated values
df.duplicated().sum()

0
```

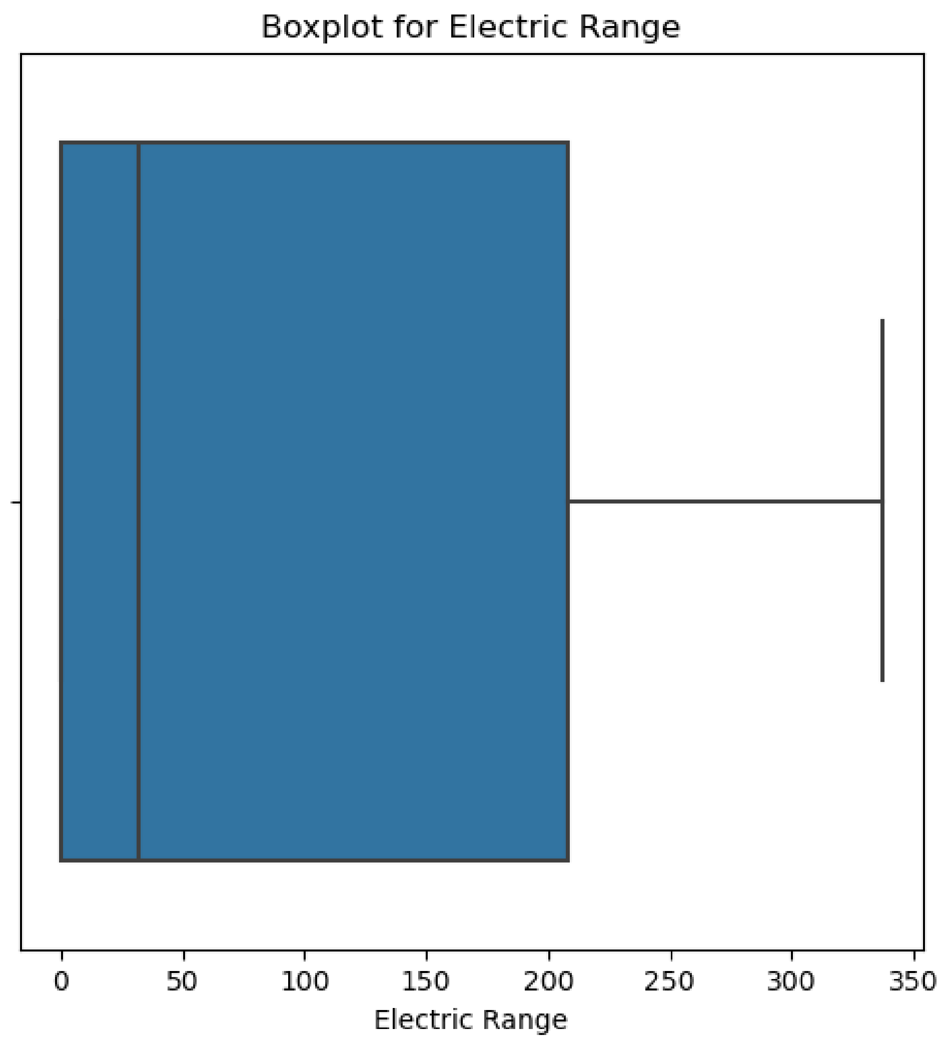
Insights:

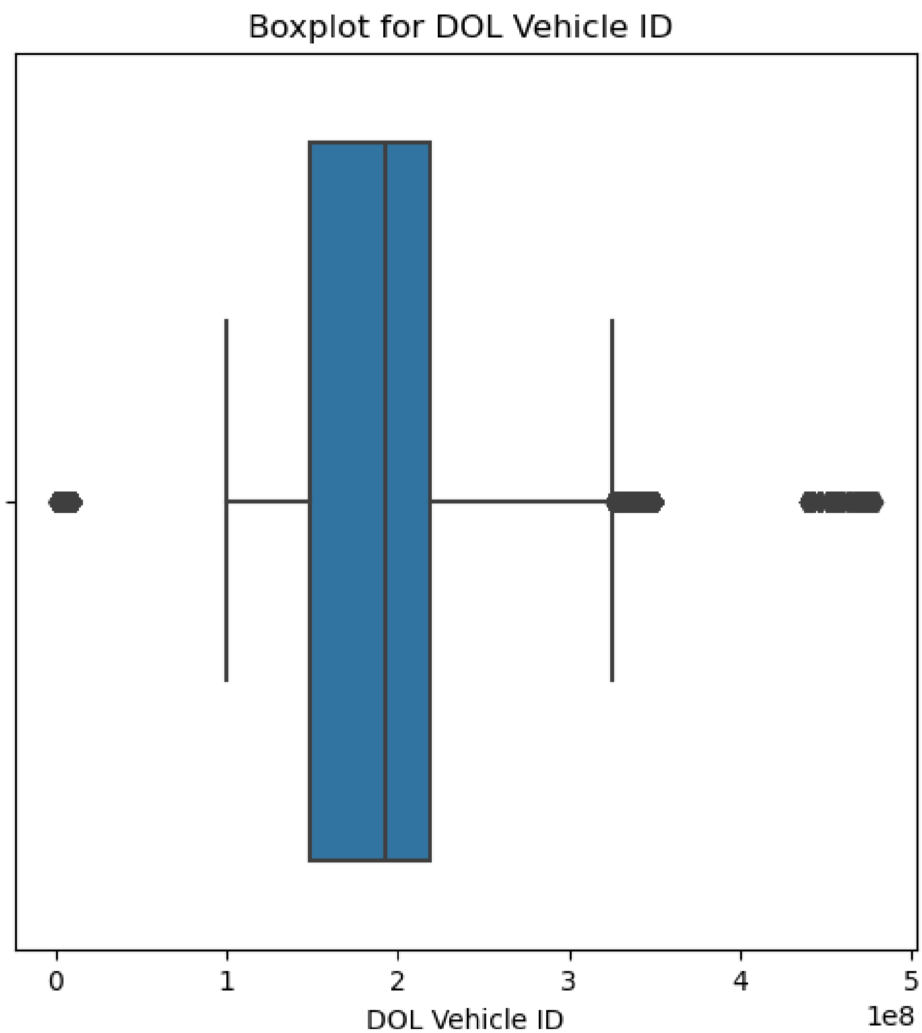
- There are no duplicated values in the data.

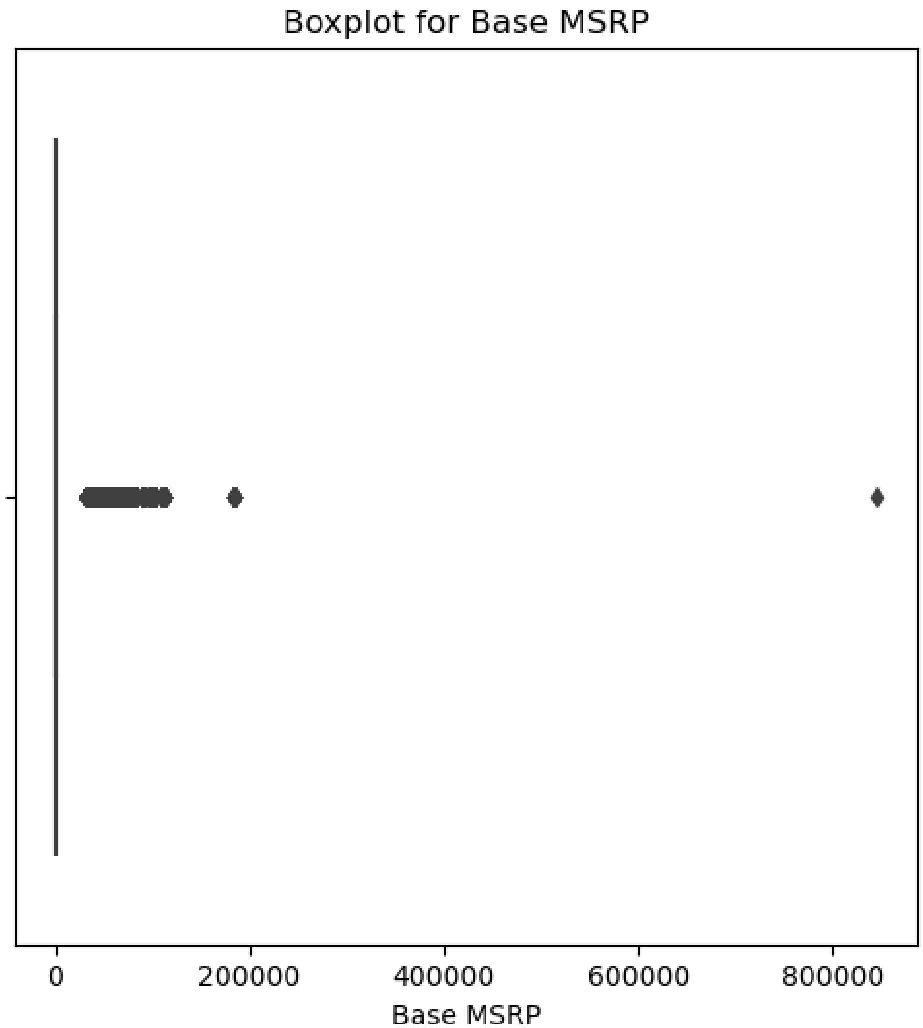
```
sns.boxplot(x='Make', y='Postal Code', data=df)
plt.title("Box Plot")
plt.show()
```



```
# Checking the outliers
plt.figure(figsize=(6,6))
sns.boxplot(x=df["Electric Range"])
plt.title("Boxplot for Electric Range")
plt.show()

plt.figure(figsize=(6,6))
sns.boxplot(x=df["DOL Vehicle ID"])
plt.title("Boxplot for DOL Vehicle ID ")
plt.show

plt.figure(figsize=(6,6))
sns.boxplot(x=df["Base MSRP"])
plt.title("Boxplot for Base MSRP")
plt.show()
```

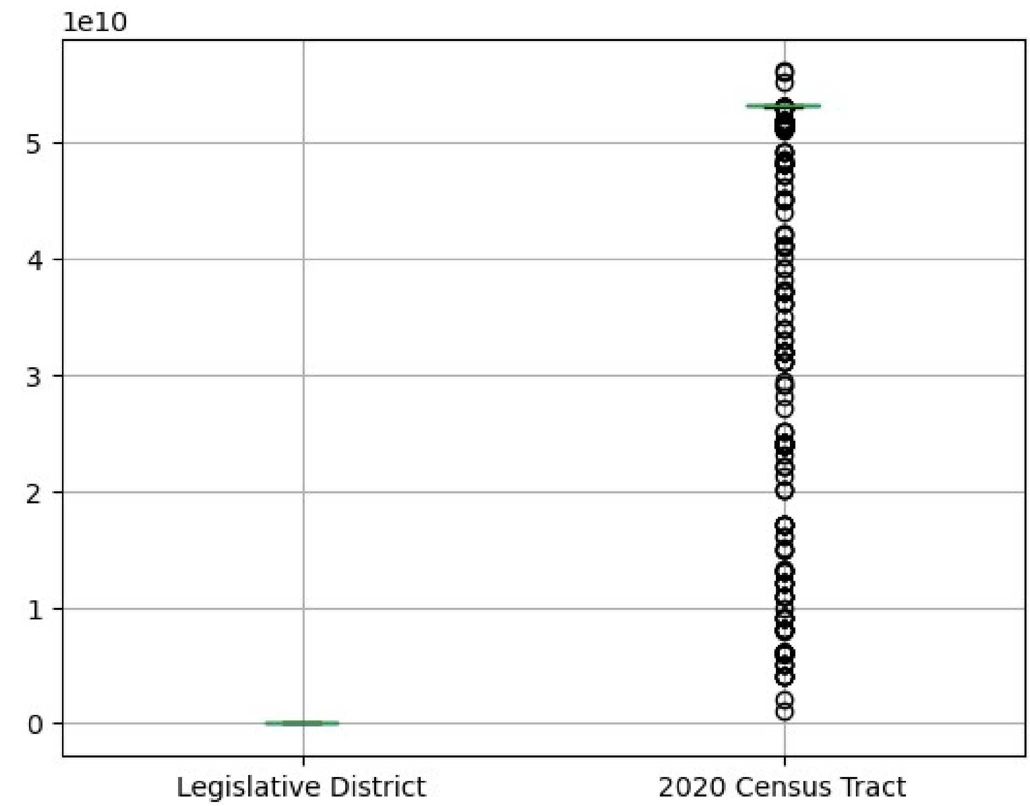



Boxplot for DOL Vehicle ID
0
1
2
3
4
5
1e8
DOL Vehicle ID



Imputing the missing values

```
Missing_columns=["Model","Legislative District","2022 Census Tract"]
SIM=SimpleImputer(strategy="most_frequent")

df[["Model"]]=SIM.fit_transform(df[["Model"]])
df["Model"].isna().sum()

0

df[["Legislative District","2020 Census Tract"]].boxplot()
plt.show()
```



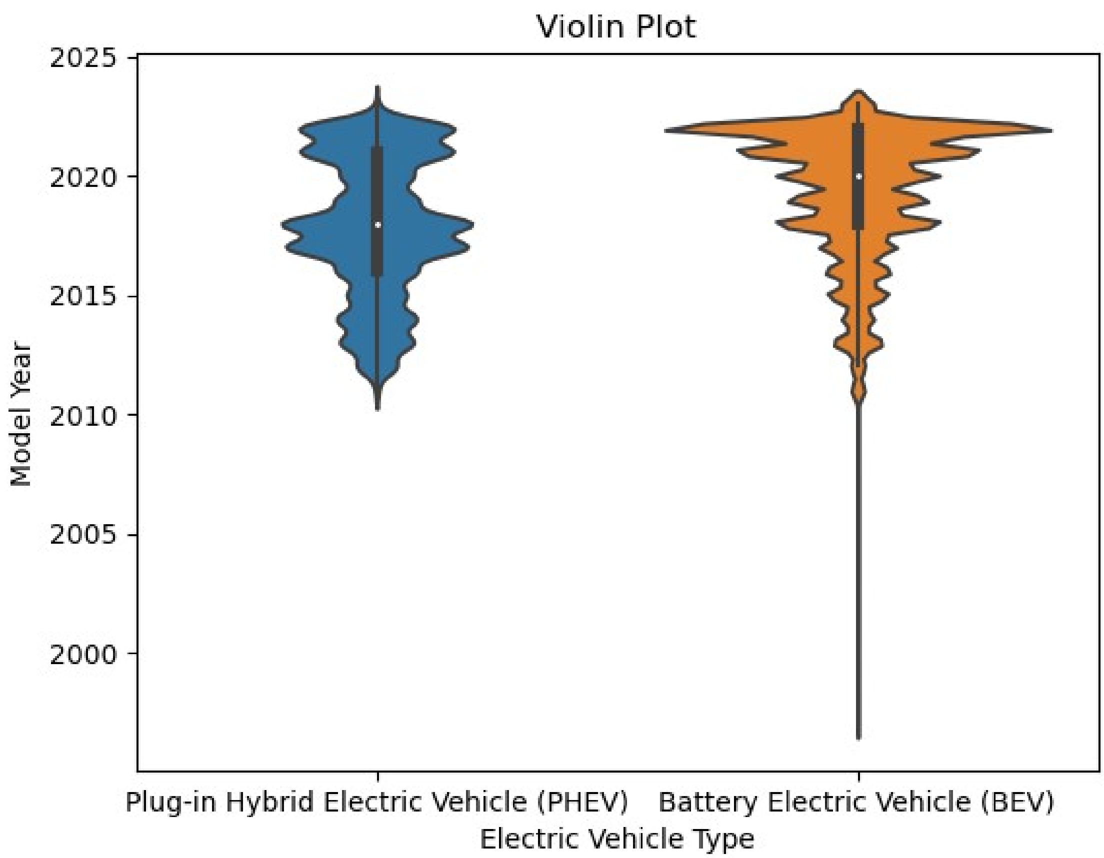

```
SIM=SimpleImputer(strategy="mean")
df[["2020 Census Tract"]]=SIM.fit_transform(df[["2020 Census Tract"]])
df["2020 Census Tract"].isna().sum()

0

SIM=SimpleImputer(strategy="median")
df[["Legislative District"]]=SIM.fit_transform(df[["Legislative
District"]])
df["Legislative District"].isna().sum()

0
```

Univariate Analysis

Analysing the data using single feature.

```
df.columns

Index(['VIN (1-10)', 'County', 'City', 'State', 'Postal Code', 'Model
Year',
       'Make', 'Model', 'Electric Vehicle Type',
       'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Electric
```

```
Range',
       'Base MSRP', 'Legislative District', 'DOL Vehicle ID',
       'Vehicle Location', 'Electric Utility', '2020 Census Tract'],
      dtype='object')

#What is the distribution of Electric Range?
sns.violinplot(x='Electric Vehicle Type', y='Model Year', data=df)
plt.title("Violin Plot")
plt.show()
```

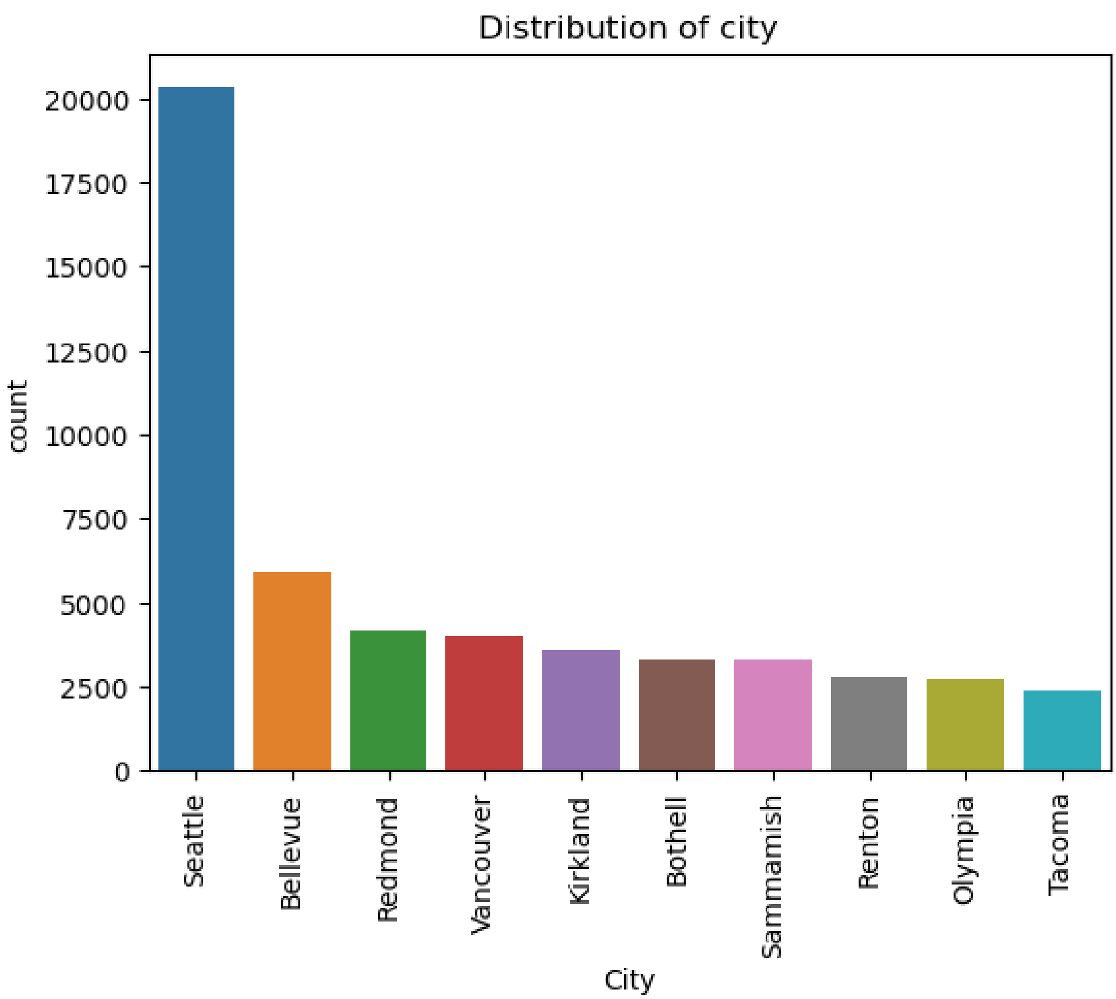


Insights:

- In between 0 to 45 the electric range density is more compared to 5 to 100.
- Above 350 the electric range is decreasing.

```
#Distribution of City?
d1=pd.DataFrame(df["City"].value_counts())
d1

              count
City
Seattle       20305
Bellevue       5921
Redmond        4201
```

```
Vancouver        4013
Kirkland         3598
...               ...
Hartline            1
Gaithersburg        1
El Paso             1
Klickitat           1
Worley              1

[629 rows x 1 columns]

sns.barplot(x=d1.index[:10], y=d1["count"][:10])  # Remove 'hue' if
not needed
plt.title("Distribution of city")
plt.xticks(rotation=90)
plt.show()
```



Insights:

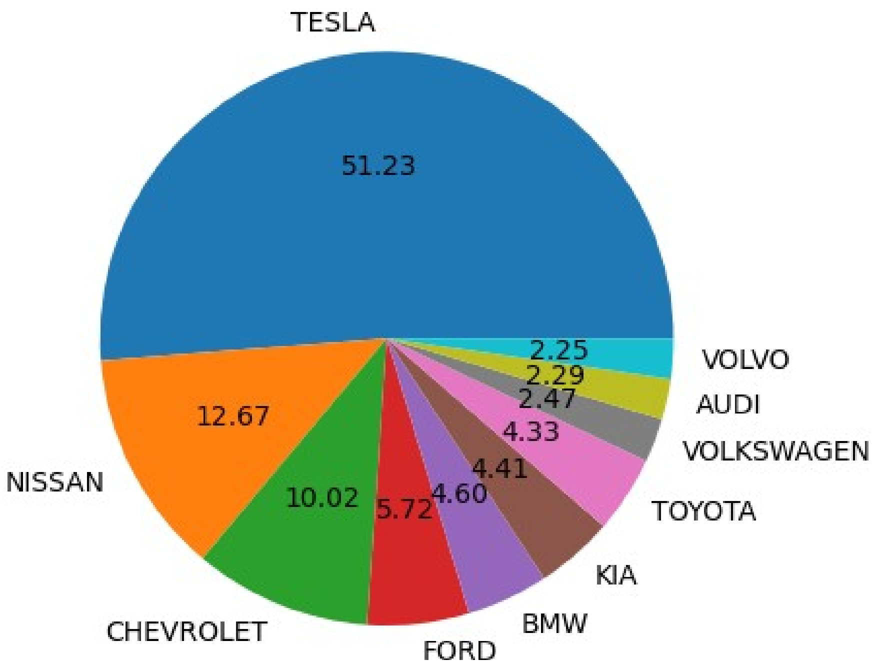
- Seattle is ranked more in distribution of cities.

- Worley is less compared to other cities.

```
#Distribution of Make?
d2=pd.DataFrame(df["Make"].value_counts())
d2
```

	count
Make	
TESLA	52078
NISSAN	12880
CHEVROLET	10182
FORD	5819
BMW	4680
KIA	4483
TOYOTA	4405
VOLKSWAGEN	2514
AUDI	2332
VOLVO	2288
CHRYSLER	1794
HYUNDAI	1412
JEEP	1152
RIVIAN	885
FIAT	822
PORSCHE	818
HONDA	792
MINI	632
MITSUBISHI	588
POLESTAR	558
MERCEDES-BENZ	506
SMART	273
JAGUAR	219
LINCOLN	168
CADILLAC	108
LUCID MOTORS	65
SUBARU	59
LAND ROVER	38
LEXUS	33
FISKER	20
GENESIS	18
AZURE DYNAMICS	7
TH!NK	3
BENTLEY	3

```
plt.pie(x=d2["count"][:10],labels=d2.index[:10],autopct="%0.2f")
plt.show()
```



Insights:

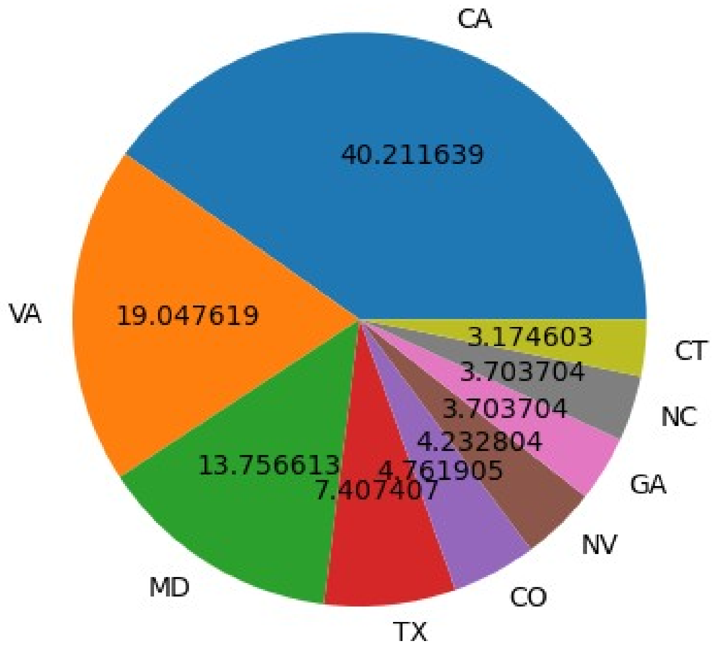
- Tesla has the highest propotion in the make compared to others.

```
#Distribution of State?
d3=pd.DataFrame(df["State"].value_counts())
d3
```

	count
State	
WA	112348
CA	76
VA	36
MD	26
TX	14
CO	9
NV	8
GA	7
NC	7
CT	6
DC	6
FL	6
AZ	6
IL	6
SC	5
OR	5
NE	5
HI	4

UT	4
AR	4
NY	4
TN	3
KS	3
MO	3
PA	3
MA	3
LA	3
NJ	3
NH	2
OH	2
WY	2
ID	2
KY	1
RI	1
ME	1
MN	1
SD	1
WI	1
NM	1
AK	1
MS	1
AL	1
DE	1
OK	1
ND	1

```
plt.pie(x=d3["count"][1:10],labels=d3.index[1:10],autopct="%1f")
plt.show()
```



Bivariate Analysis

Analysing the data using two features.

Which state has more Battery and least plug-in-hybrid electric type vehicles?

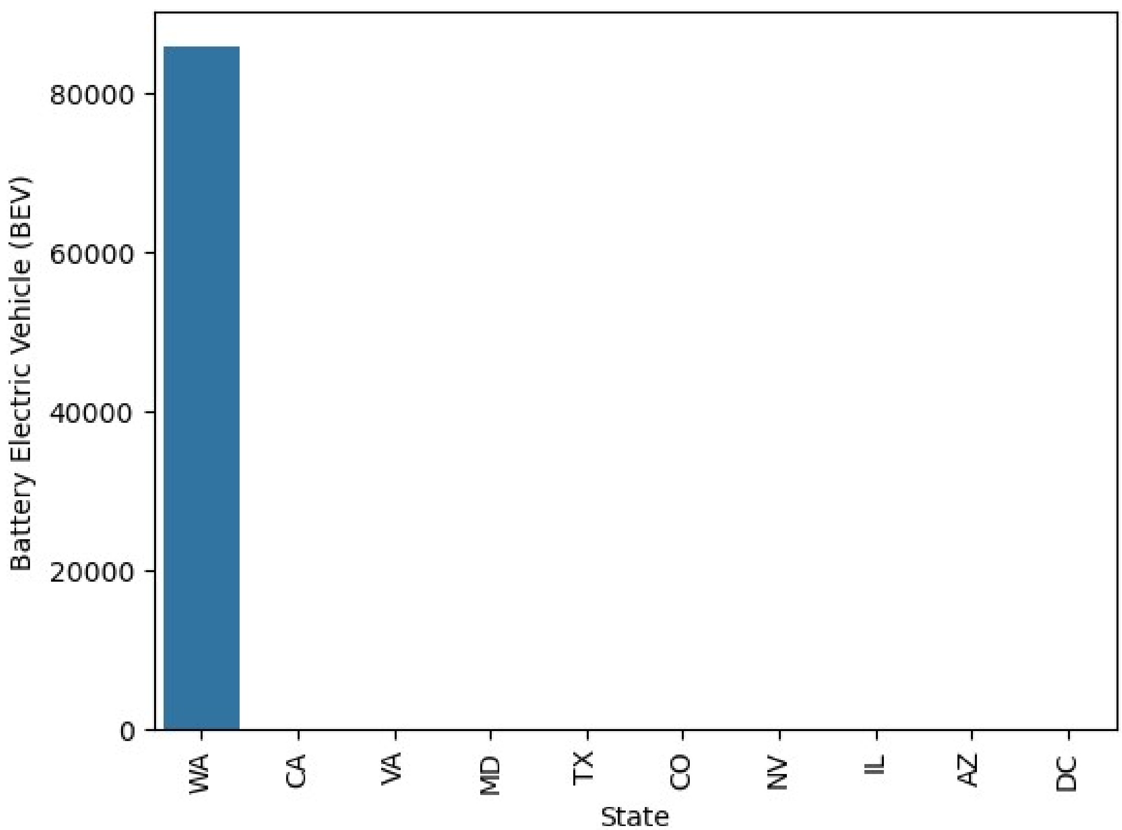
```
df.columns

Index(['VIN (1-10)', 'County', 'City', 'State', 'Postal Code', 'Model
Year',
       'Make', 'Model', 'Electric Vehicle Type',
       'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Electric
Range',
       'Base MSRP', 'Legislative District', 'DOL Vehicle ID',
       'Vehicle Location', 'Electric Utility', '2020 Census Tract'],
      dtype='object')

g1=pd.crosstab(index=df["State"],columns=df["Electric Vehicle
Type"]).sort_values(by=["Battery Electric Vehicle (BEV)","Plug-in
Hybrid Electric Vehicle (PHEV)"],ascending=False)

g1.head()
len(g1)
```

```
45
```

```
g1.index
```

```
Index(['WA', 'CA', 'VA', 'MD', 'TX', 'CO', 'NV', 'IL', 'AZ', 'DC',
'SC', 'GA',
       'NC', 'FL', 'NE', 'AR', 'NY', 'PA', 'TN', 'OR', 'HI', 'UT',
'KS', 'LA',
       'MA', 'MO', 'ID', 'OH', 'WY', 'CT', 'NH', 'DE', 'MN', 'MS',
'NM', 'RI',
       'SD', 'WI', 'NJ', 'AK', 'AL', 'KY', 'ME', 'ND', 'OK'],
      dtype='object', name='State')
```

```
sns.barplot(x=g1.index[:10],y=g1["Battery Electric Vehicle (BEV)"]
[:10])
plt.xticks(rotation=90)
plt.show()
```



Insights:

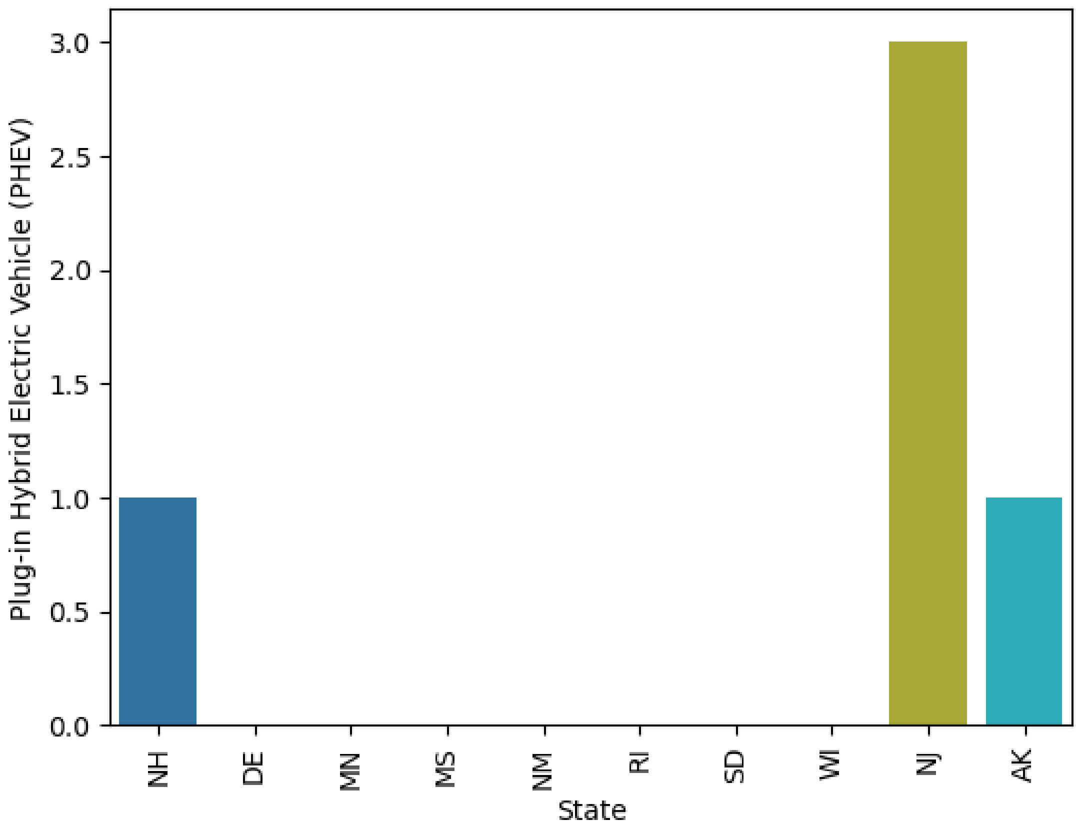
- WA has more Battery Electric vehicles compared to other states.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 17 columns):
 #   Column                                             Non-Null Count
Dtype
---  ------                                             --------------
-----
 0   VIN (1-10)                                         112634 non-
null  object
 1   County                                             112634 non-
null  object
 2   City                                               112634 non-
null  object
 3   State                                              112634 non-
null  object
 4   Postal Code                                        112634 non-
null  int64
 5   Model Year                                         112634 non-
null  int64
 6   Make                                               112634 non-
null  object
 7   Model                                              112634 non-
null  object
 8   Electric Vehicle Type                              112634 non-
null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  112634 non-
null  object
 10  Electric Range                                     112634 non-
null  int64
 11  Base MSRP                                          112634 non-
null  int64
 12  Legislative District                               112634 non-
null  float64
 13  DOL Vehicle ID                                     112634 non-
null  int64
 14  Vehicle Location                                   112610 non-
null  object
 15  Electric Utility                                   112191 non-
null  object
 16  2020 Census Tract                                  112634 non-
null  float64
dtypes: float64(2), int64(5), object(10)
memory usage: 14.6+ MB
```

```
sns.barplot(x=g1.index[30:40],y=g1["Plug-in Hybrid Electric Vehicle
(PHEV)"][30:40])
```

```
plt.xticks(rotation=90)
plt.show()
```



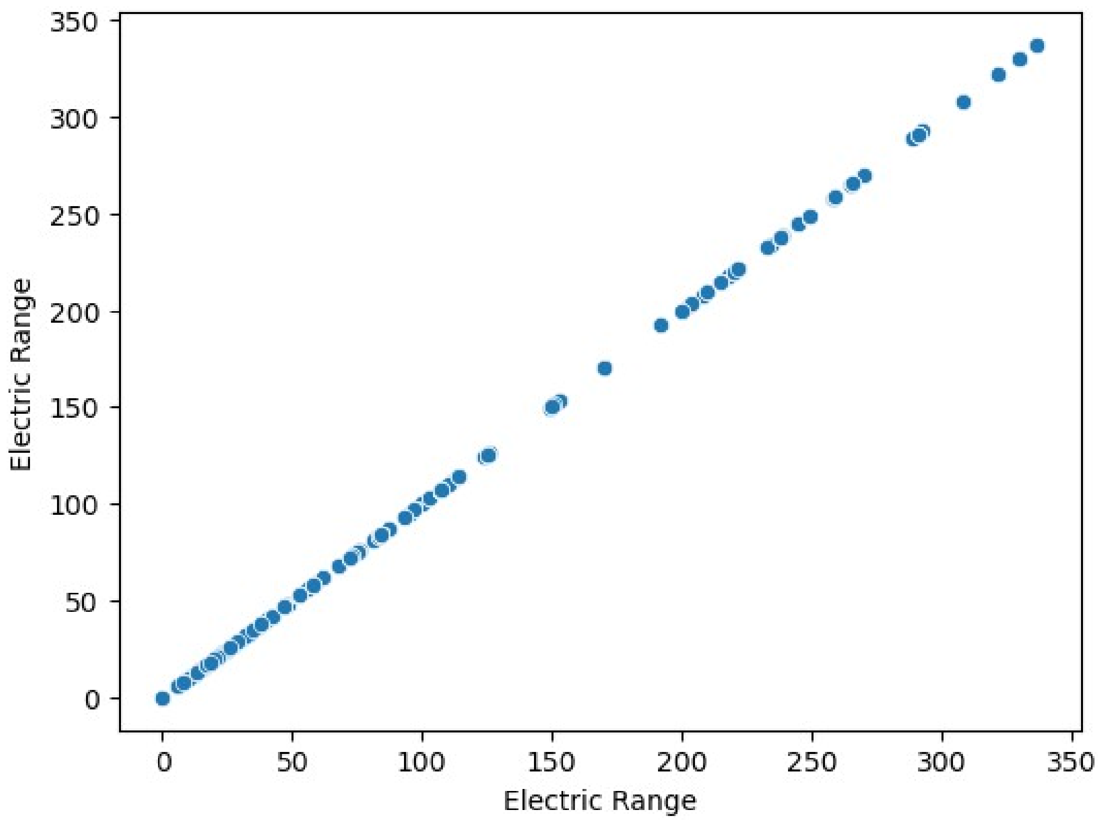
Insights:

- OK,ND has less plug-in-hybrid electric vehicles.

```
#What is the relationship between the Electric Range and Base MSRP of
electric vehicles?
df[["Electric Range","Base MSRP"]].corr()
```

	Electric Range	Base MSRP
Electric Range	1.000000	0.085025
Base MSRP	0.085025	1.000000

```
sns.scatterplot(x=df["Electric Range"],y=df["Electric Range"])
plt.show()
```



Insights:

- Since the correlation is minimal, Electric Range is not a reliable predictor of the Base MSRP

```
#Create a Choropleth using plotly.express to display the number of EV vehicles based on location.
state_data = df.groupby('State')['VIN (1-10)'].count().reset_index()
state_data.columns = ['State', 'EV Count']

fig = px.choropleth(state_data,
                    locations='State',
                    locationmode='USA-states',
                    color='EV Count',
                    color_continuous_scale='greens',
                    scope='usa',
                    labels={'EV Count': 'Number of EV Vehicles'},
                    title='Number of Electric Vehicles by State'
                  )

fig.show()
```


```
!pip install bar-chart-race

Collecting bar-chart-race
  Downloading bar_chart_race-0.1.0-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: pandas>=0.24 in e:\users\dell\
anaconda3\lib\site-packages (from bar-chart-race) (2.1.4)
Requirement already satisfied: matplotlib>=3.1 in e:\users\dell\
anaconda3\lib\site-packages (from bar-chart-race) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(1.4.4)
Requirement already satisfied: numpy<2,>=1.21 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(1.26.4)
Requirement already satisfied: packaging>=20.0 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in e:\users\dell\
anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in e:\users\dell\
anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
```

```
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in e:\users\dell\
anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
(2023.3)
Requirement already satisfied: six>=1.5 in e:\users\dell\anaconda3\
lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.1->bar-
chart-race) (1.16.0)
Downloading bar_chart_race-0.1.0-py3-none-any.whl (156 kB)
   ---------------------------------------- 0.0/156.8 kB ? eta -:--:--
   ---------------------------------------- 0.0/156.8 kB ? eta -:--:--
   -- ------------------------------------- 10.2/156.8 kB ? eta
-:--:--
   ------- -------------------------------- 30.7/156.8 kB 435.7 kB/s
eta 0:00:01
   -------------------------- ------------- 102.4/156.8 kB 737.3 kB/s
eta 0:00:01
   ---------------------------------------  153.6/156.8 kB 1.0 MB/s
eta 0:00:01
   ---------------------------------------  153.6/156.8 kB 1.0 MB/s
eta 0:00:01
   ---------------------------------------  153.6/156.8 kB 1.0 MB/s
eta 0:00:01
   ---------------------------------------- 156.8/156.8 kB 519.8 kB/s
eta 0:00:00
Installing collected packages: bar-chart-race
Successfully installed bar-chart-race-0.1.0
```

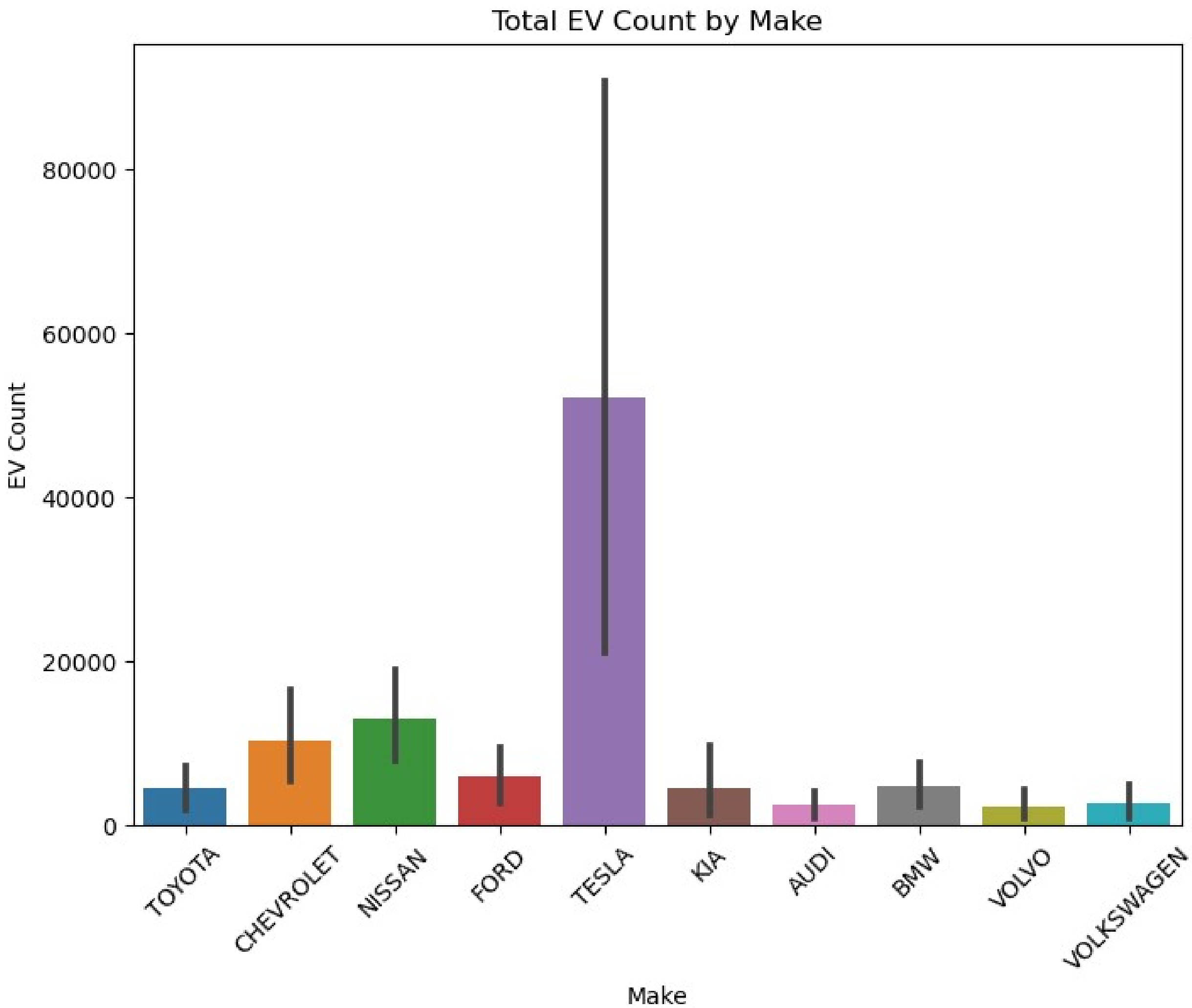
```
df.columns
```

```
Index(['VIN (1-10)', 'County', 'City', 'State', 'Postal Code', 'Model
Year',
       'Make', 'Model', 'Electric Vehicle Type',
       'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Electric
Range',
       'Base MSRP', 'Legislative District', 'DOL Vehicle ID',
       'Vehicle Location', 'Electric Utility', '2020 Census Tract'],
      dtype='object')
```

```
ev_make_by_year_full
```

```
     Model Year        Make  EV Count
0          1997      TOYOTA         0
1          1997   CHEVROLET         1
2          1997      NISSAN         0
3          1997        FORD         0
4          1997       TESLA         0
..          ...         ...       ...
195        2023         KIA        79
196        2023        AUDI        12
197        2023         BMW        73
```

```
198         2023       VOLVO        21
199         2023  VOLKSWAGEN        69

[200 rows x 3 columns]

plt.figure(figsize=(8, 6))
sns.barplot(x='Make', y='EV Count', data=ev_make_by_year_full,
estimator=sum)
plt.xticks(rotation=45)
plt.title('Total EV Count by Make')
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming df contains columns 'Year' and 'EV_count'
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Model Year', y='EV
```

```
Count',hue='Make' ,data=ev_make_by_year_full)
plt.title('EV Count Over the Years')
plt.show()
```