electric-vechile-project

October 10, 2024

[ ]:
```
import pandas as pd
```

[ ]:
```
df =pd.read_csv(r"C:\Users\badveli\Downloads\Electric_
↪vechile project\dataset.csv")
df
```

[ ]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	\
0	JTMEB3FV6N	Monroe	Key West	FL	33040	2022	
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	
4	3FA6P0SU1K	Snohomish	Everett	W	98201	2019	
…	…	…	…	A …	…		
1126	7SAYGDEF	King	Duvall		9801	2022	
29	2N			W	9		
				A			
112630	1N4BZ1CP7K	San Juan	Friday Harbor	WA	98250	2019	
112631	1FMCU0KZ4N	King	Vashon	WA	98070	2022	
112632	KNDCD3LD4J	King	Covington	WA	98042	2018	
112633	YV4BR0CL8N	King	Covington	WA	98042	2022	

	Make	Model	Electric Vehicle Type	\
0	TOYOTA	RAV4 PRIME	Plug-in Hybrid Electric Vehicle (PHEV)	
1	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	
2	NISSAN	LEAF	Battery Electric Vehicle (BEV)	
3	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	
4	FORD	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	
…	…	…	…	
1126	TESL	MODEL Y	Battery Electric Vehicle (BEV)	
29	A			
112630	NISSAN	LEAF	Battery Electric Vehicle (BEV)	
112631	FORD	ESCAPE	Plug-in Hybrid Electric Vehicle (PHEV)	
112632	KIA	NIRO	Plug-in Hybrid Electric Vehicle (PHEV)	
12633	VOLVO	XC90	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
…	…	…
112629	Eligibility unknown as battery range has not b..	0
112630	Clean Alternative Fuel Vehicle Eligible	150
112631	Clean Alternative Fuel Vehicle Eligible	38
112632	Not eligible due to low battery range	26
112633	Not eligible due to low battery range	18

	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.	2006714	
…	…	0	…	
1126 29	0	…	2179552 65	
		45. 0		
112630	0	40.0	103663227	
112631	0	34.0	193878387	
112632	0	47.0	125039043	
112633	0	47.0	194673692	

	Vehicle Location	\
0	POINT (-81.80023 24.5545)	
1	POINT (-114.57245 35.16815)	
2	POINT (-120.50721 46.60448)	
3	POINT (-121.7515 48.53892)	
4	POINT (-122.20596 47.97659)	
…	…	
1126 29	POINT (-121.98609 47.74068)	
112630	POINT (-123.01648 48.53448)	
112631	POINT (-122.4573 47.44929)	
112632	POINT (-122.09124 47.33778)	
112633	POINT (-122.09124 47.33778)	

	Electric Utility	2020 Census Tract
0	NaN	12087972100
1	NaN	32003005702
2	PACIFICORP	53077001602
3	PUGET SOUND ENERGY INC	53057951101
4	PUGET SOUND ENERGY INC	53061041500
…	…	…
112629	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	53033032401
112630	BONNEVILLE POWER ADMINISTRATION||ORCAS POWER &..	53055960301
112631	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	53033027702
112632	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	53033032007
112633	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	53033032005

[112634 rows x 17 columns]

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 17 columns):
 #   Column                                             Non-Nul 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
```

[ ]:
```
df.duplicated().sum()
```

[ ]: 0

[ ]:

[ ]:
```
df['Model'].value_counts()
```

[ ]:
```
Model
MODEL3          23135
MODELY          17142
LEAF            12880
MODELS           7377
BOLTEV           4910
                ...
745LE               2
S-10 PICKUP         1
SOLTERRA            1
```

```
918                 1
FLYING SPUR         1
Name: count, Length: 114, dtype: int64
```

[ ]:
```
df['Model']  =df['Model'].fillna(df['Model'].mode()[0])
```

[ ]:
```

```

[ ]:
```
df['Legislative  District'].describe()
```

[ ]:
```
count    112348.000000
mean         29.805604
std          14.700545
min           1.000000
25%          18.000000
50%          34.000000
75%          43.000000
max          49.000000
Name: Legislative  District,  dtype: float64
```

[ ]:
```
df['Legislative  Distric t']  =df['Legislative  District' ].fillna( df['Legislativ e␣
  Distr ict'].mean())
```

[ ]:
```

```

[ ]:
```
df['Vehicle    Location'].value_counts()
```

[ ]:
```
Vehicle  Location
POINT (-122.13158 47.67858)    2916
POINT (-122.2066 47.67887)     2059
POINT (-122.1872 47.61001)     2001
POINT (-122.31765 47.70013)    1880
POINT (-122.12096 47.55584)    1852
                               ...
POINT (-124.33152 48.05431)       1
POINT (-77.41203 39.41574)        1
POINT (-123.61022 46.35588)       1
POINT (-112.04165 40.68741)       1
POINT (-116.91895 47.40077)       1
Name: count, Length: 758, dtype: int64
```

[ ]:
```
df['Vehicle  Location']  =df['Vehicle  Location'].fillna(df['Vehicle  Location'].
 ↪mode()[0])
```

[ ]:
```

```

[ ]:
```
df['Electr ic  Utility'].value_counts()
```

```
Electric Utility
PUGET SOUNDENERGY INC||CITY OF TACOMA- (WA)
40247
PUGET SOUNDENERGY INC
22172
CITY OF SEATTLE - (WA)|CITY OF TACOMA- (WA)
21447
BONNEVILLE POWERADMINISTRATION||PUD NO1 OF CLARK COUNTY - (WA)
6522
BONNEVILLE POWERADMINISTRATION||CITY OF TACOMA- (WA)||PENINSULA LIGHT COMPANY
5053
                ...
BONNEVILLE POWER ADMINISTRATION||PENINSULA LIGHT COMPANY
1
BONNEVILLE POWER ADMINISTRATION||PUD NO1 OF ASOTIN COUNTY
1
CITY OF SEATTLE - (WA)
1
BONNEVILLE POWER ADMINISTRATION||NESPELEM VALLEY ELEC COOP, INC
1
BONNEVILLE POWERADMINISTRATION||PUD NO1 OF CLALLAM COUNTY|PUD NO1 OF
JEFFERSON COUNTY        1
Name: count, Length: 73, dtype: int64
```

```
df['Electric Utility'] =df['Electric Utility'].fillna(df['Electric Utility'].
  mode()[0])
```

```
df.isna().sum()
```

```
VIN (1-10)                                          0
County                                              0
City                                                0
State                                               0
Postal Code                                         0
Model Year                                          0
Make                                                0
Model                                               0
Electric Vehicle Type                               0
Clean Alternative Fuel Vehicle (CAFV) Eligibility   0
Electric Range                                      0
Base MSRP                                           0
Legislative District                                0
DOL Vehicle ID                                      0
Vehicle Location                                    0
Electric Utility                                    0
2020 Census Tract                                   0
dtype: int64
```

[ ]:
```
df.to_csv('Electric  Vechile  data')
```

[ ]:
```
import os
os.getcwd()
```

[ ]: 'C:\\Users\\badveli\\INTERNSHIP'

1 Task 1: This is an open ended problem. Apply Exploratory Data Analysis (Univariate and Bivariate) on the dataset avail- able above.

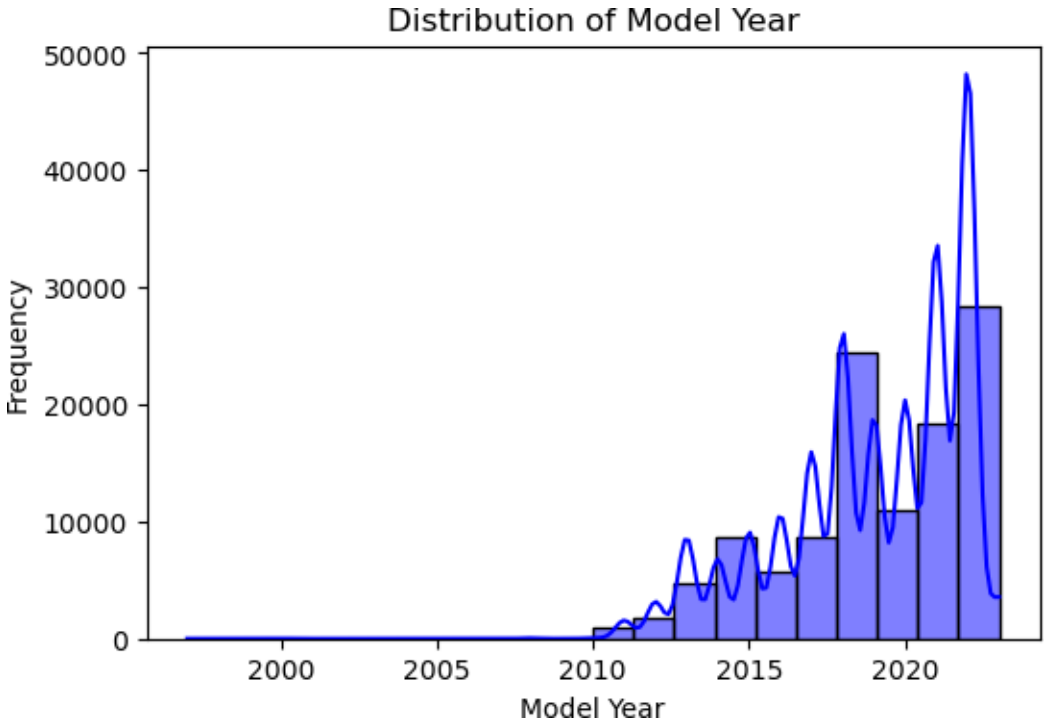
[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 17 columns):
 #   Column                                             Non-Nul 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                                   112634 non-null object
 15  Electric Utility                                   112634 non-null object
 16  2020 Census Tract                                  112634 non-null int64
dtypes: float64(1), int64(6), object(10)
memoryusage: 14.6+ MB
```

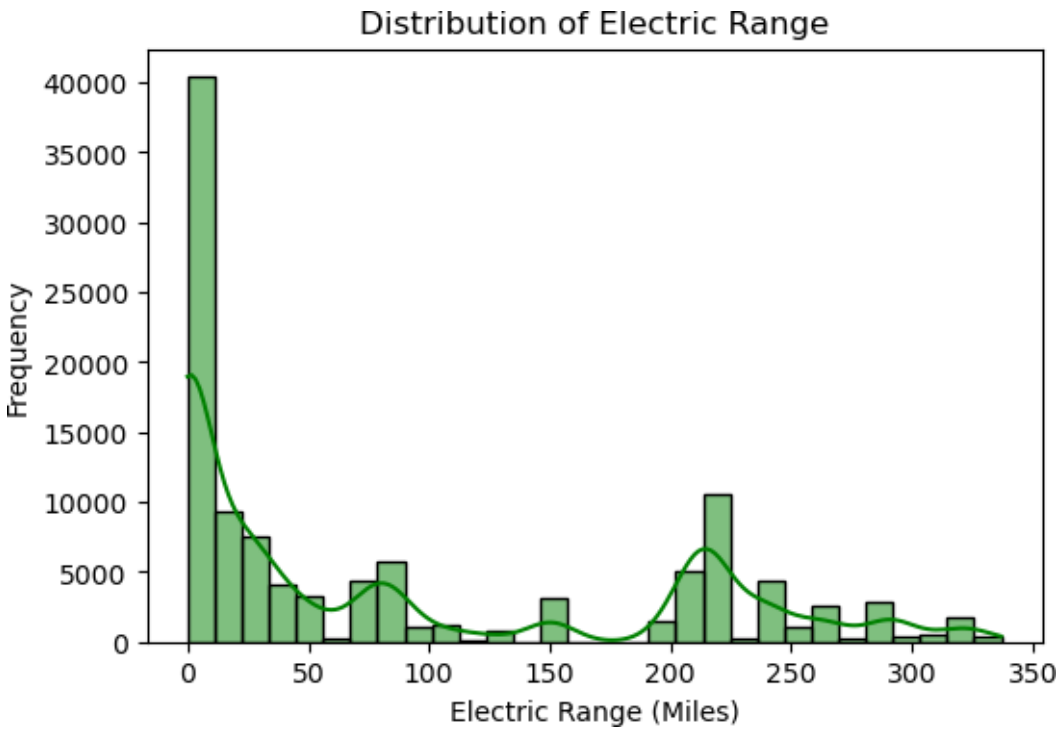
[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

2 Univariate Analysis

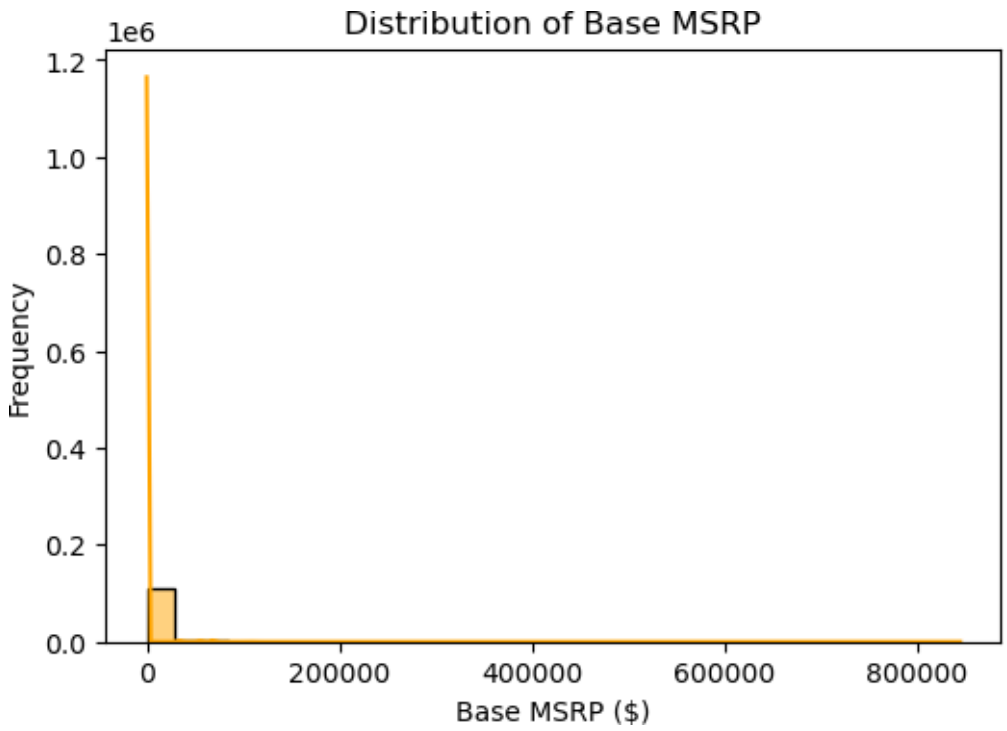
[ ]:
```
#Distribution of Model Year
plt.figure(figsize=(6,4))
sns.histplot(df['Model Year'], bins=20, kde=True, color='blue')
plt.title('Distribution of Model Year')
plt.xlabel('Model Year')
plt.ylabel('Frequency')
plt.show()
```



[ ]:
```
#Distribution of Electric Range
plt.figure(figsize=(6,4))
sns.histplot(df['Electric Range'], bins=30, kde=True, color='green')
plt.title('Distribution of Electric Range')
plt.xlabel(' Electric Range (M iles)')
plt.ylabel('Frequency')
plt.show()
```



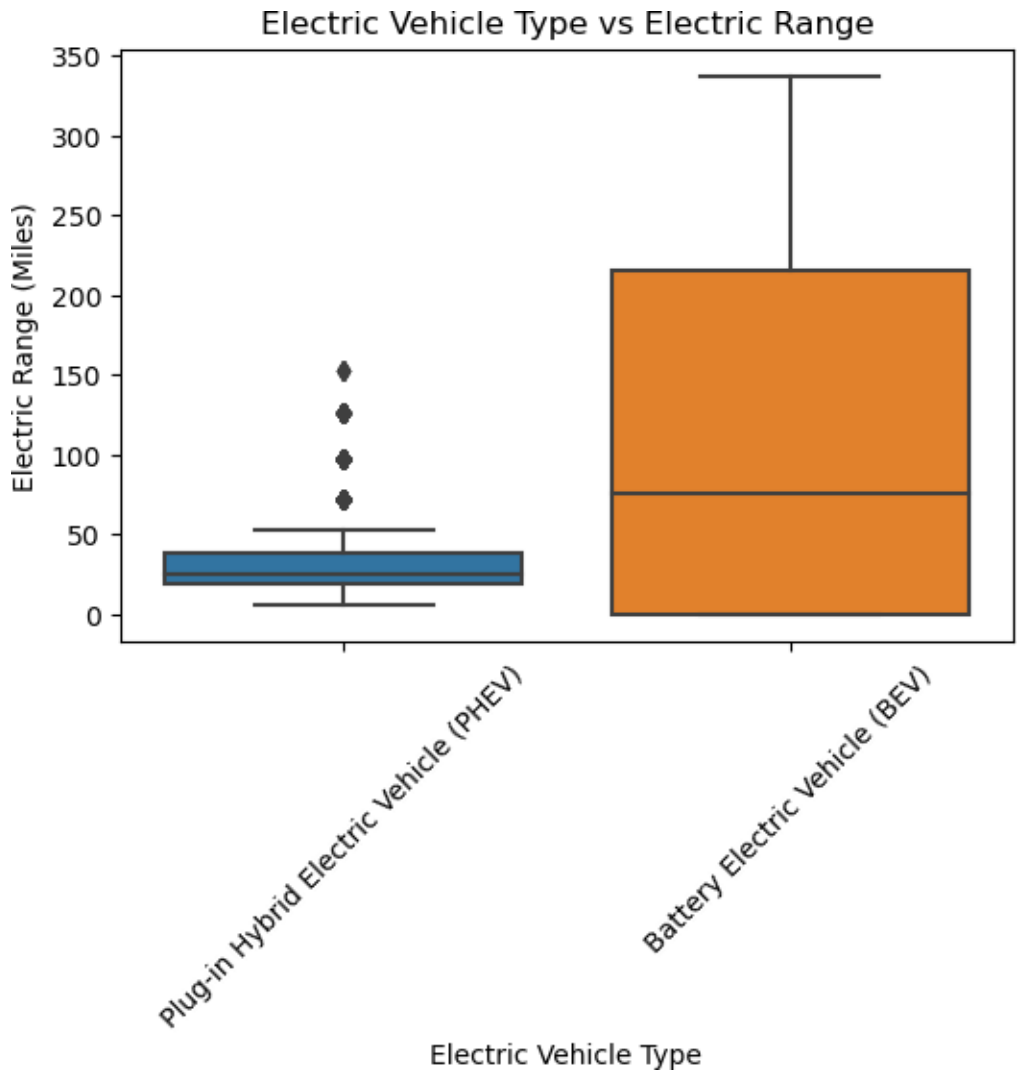
[ ]:
```
#Distribution of Base MSRP
plt.figure(figsize=(6,4))
sns.histplot(df['Base MSRP'], bins=30, kde=True, color='orange')
plt.title('Distribution of Base MSRP')
plt.xlabel('Base MSRP ( $ ) ')
plt.ylabel('Frequency')
plt.show()
```



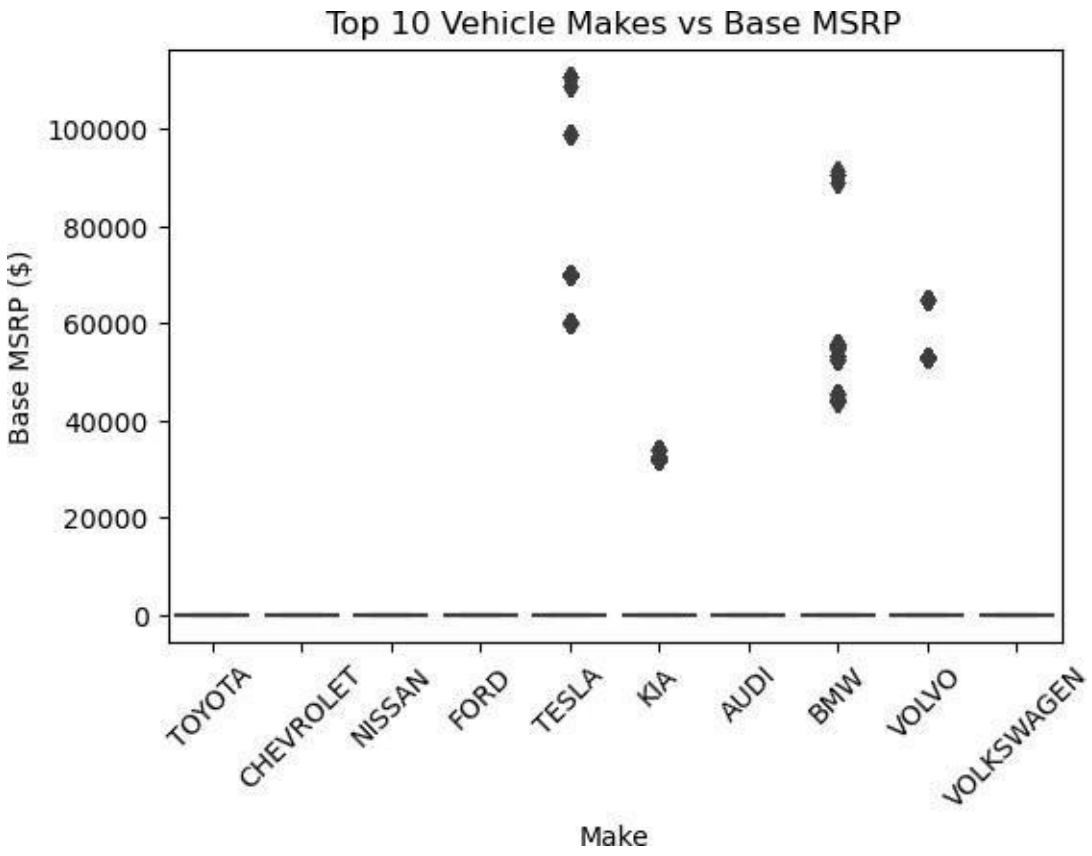
[ ]:

3 Bivariate Analysis

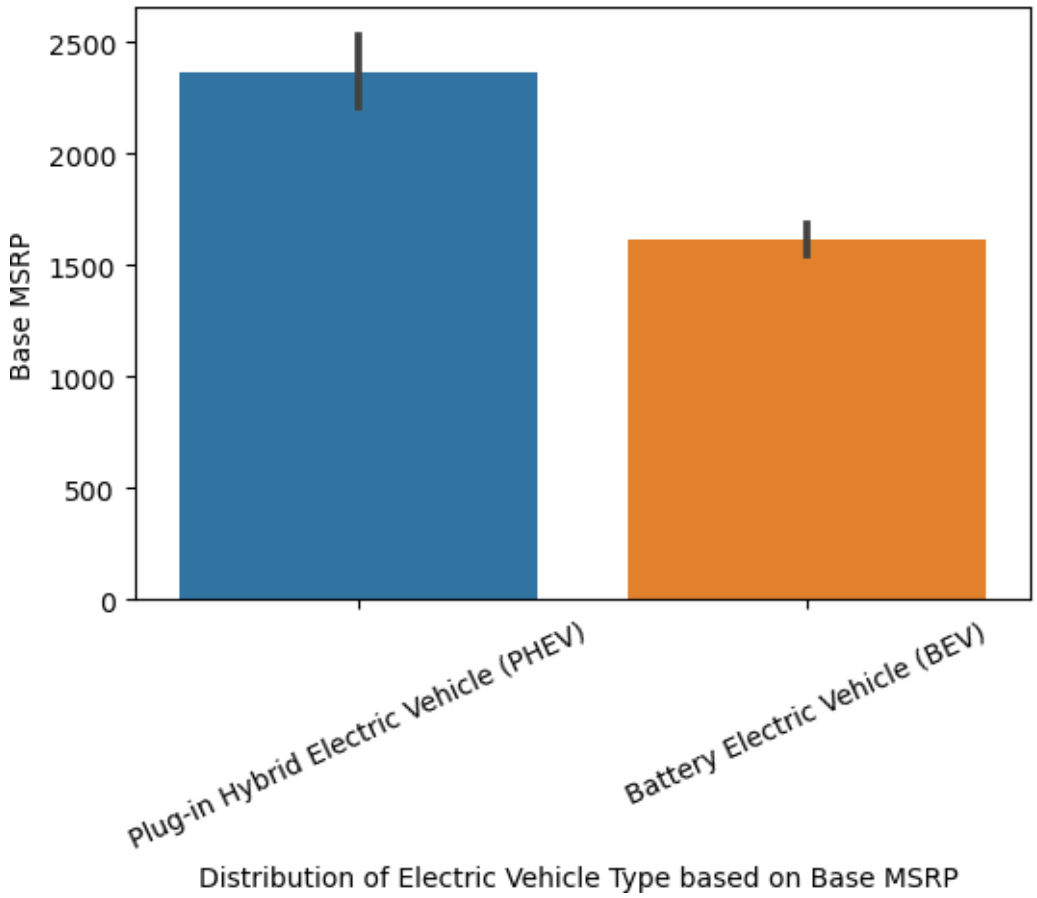
[ ]:
```
#Electric Vehicle Type vs Electric Range
plt.figure(figsize=(6,4))
sns.boxplot(x='Electric Vehicle Type', y='Electric Range', data=df)
plt.title('Electric Vehicle Type vs Electric Range')
plt.xticks(rotation=45)
plt.ylabel('Electric Range (Miles)')
plt.show()
```



[ ]:
```
#Make vs Base MSRP
plt.figure(figsize=(6,4))
top_makes =df['Make'].value_counts().nlargest(10).index
sns.boxplot(x='Make', y='Base MSRP', data=df[df['Make'].isin(top_makes)])
plt.title('Top 10 Vehicle Makes vs Base MSRP')
plt.xticks(rotation=45)
plt.ylabel('Base MSRP ($)')
plt.show()
```



[ ]:
```
plt.figure(figsize=(6,4))
sns.barplot(x='Electric Vehicle Type', y='Base MSRP',data=df)
plt.xticks(rotation =25)
plt.xlabel('Distribution of Electric Vehicle Type based on Base MSRP')
plt.ylabel('Base MSRP')
plt.show()
```



[ ]:

4 Task 2: Create a Choropleth using plotly.express to display the number of EV vehicles based on location.

[ ]:
```
!pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\srikanth\anaconda3\lib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\srikanth\anaconda3\lib\site-packages (from plotly) (8.2.2)
```

[ ]:
```
import plotly.express as px
```

[ ]:
```
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='Blues',
                    scope='usa',
                    labels={'EV Count': 'Number of EV Vehicles'},
                    title='Number of Electric Vehicles by State')

fig.show()
```

5 Scatter Plot using plotly.express

Note - Scatter Plot is a bivariate plot. Bivariate means it requires two variables / features / columns. You should make a note that both the variables should be real numerical valued.

[ ]:
```
fig = px.scatter(df,
                 x='Base MSRP',
                 y='Electric Range',
                 color='Electric Vehicle Type',
                 title='Scatter Plot of Electric Range vs Base MSRP',
                 labels={'Base MSRP': 'Base MSRP ($)', 'Electric Range':␣
↪'Electric Range (Miles)'},
                 hover_data=['Make', 'Model'])

fig.show()
```

6 Box Plot using plotly.express

Note - Box Plot can be used to create a univariate or bivariate plot. For a univariate box plot, the column type should be real numerical. For a bivariate box plot, one column should be categorical and another column should be real numerical. Below is an example of code for bivariate box plot.

[ ]:
```
fig = px.box(df,
             x='Electric Vehicle Type',
             y='Base MSRP',
             title='Box Plot of Base MSRP by Electric Vehicle Type',
             labels={'Base MSRP': 'Base MSRP ($)', 'Electric Vehicle Type': 'EV␣
↪Type'},
             hover_data=['Make', 'Model'])

fig.show()
```

7 Pie Chart Plot using plotly.express

Note - Pie Chart Plot can be used to create a bivariate plot. For a bivariate pie chart plot, one column should be categorical and another column should be real numerical. Below is an example of code for the plot. names: It should be categorical column values: It should be numeric column

[ ]:
```
vehicle_type_count =df.groupby('Electric Vehicle Type')['VIN (1-10)'].count().
↪reset_index()
vehicle_type_count.columns =['Elec tric Vehicle Type', 'Count']

#Create a pie chart
fig =px.pie(vehicle_type_count,
            names='Electric Vehicle Type',
            values='Count',
            title='Distribution of Electric Vehicles by Type',
            labels={'Electric Vehicle Type': 'EV Type', 'Count': 'Number of␣
↪Vehicles'})

#Show the pie chart
fig.show()
```

[ ]:
```

```

8 Task 3: Create a Racing Bar Plot to display the animation of EV Make and its count each year.

[ ]:
```
!pip install bar-chart-race
```

```
Requirement already satisfied: bar-chart-race in
c:\users\srikanth\anaconda3\lib\site-packages (0.1.0)
Requirement already satisfied: pandas>=0.24 in
c:\users\srikanth\anaconda3\lib\site-packages (from bar-chart-race) (2.0.3)
Requirement already satisfied: matplotlib>=3.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from bar-chart-race) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (1.0.5)
Requirement already satisfied: cycler>=0.10 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (1.4.4)
```

```
Requirement already satisfied: numpy>=1.20in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race)  (1.24.3)
Requirement already satisfied: packaging>=20.0 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\srikanth\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-
race) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\srikanth\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-
race) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\srikanth\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.1->bar-chart-race) (1.16.0)
```

[ ]:
```
import bar_chart_race as bcr
```

[ ]:
```
import plotly.express as px
import pandas as pd

# Group the data by 'Model Year' and 'Make' to get the count of vehicles each␣
↪year
make_year_data =df.groupby(['Model Year', 'Make'])['VIN (1-10)'].count().
  reset_index()
make_year_data.columns =['Model Year', 'Make', 'Count']

#Create an animated bar chart using Plotly
fig =px.bar(make_year_data,
            x='Make',
            y='Count',
            color='Make',
            animation_frame='Model Year',
            animation_group='Make',
            range_y=[0, make_year_data['Count'].max() +100],
            title='Electric Vehicle Makes Over Time',
            labels={'Count': 'Number of Vehicles', 'Make': 'EV Make'})

#Show the animated bar chart
```

```
fig.show()
```

[ ]:

[ ]:

[ ]: