o03vv2cla

October 12, 2024

[2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

import warnings
warnings.filterwarnings('ignore')
```

[3]:

```
df = pd.read_csv(r"C:\Users\maazp\Downloads\dataset.csv")
df.head()
```

[3]:

```
   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
```

[ ]:

```

```

[ ]:

```

```

[4]:

```python
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
```

[5]:

```python
# It provides statistical summary of numerical columns
df.describe
```

[5]:

```
<bound method NDFrame.describe of          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    WA        98201         2019
…            …        …            …   …          …         …
112629  7SAYGDEF2N       King         Duvall    WA        98019         2022
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)
…           …         …                                     …
112629      TESLA     MODEL Y          Battery Electric Vehicle (BEV)
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)
112633      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                  38.0         2006714
…           …                   …             …
112629          0                  45.0       217955265
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)
…                              …
112629  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]>
```

[6]:
```python
# It displays the dimensions of the data
df.shape
```

[6]:
```
(112634, 17)
```

1 EDA - Exploratary Data Analysis

[7]:
```python
df.isna().sum()
```

[7]:
```
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
```

[8]:

```
df.duplicated().sum()
```

[8]:

```
0
```

[9]:

```
df['Model'] = df['Model'].fillna(df['Model'].mode()[0])
df['Legislative District'] = df['Legislative District'].fillna(df['Legislative␣
 ↪District'].mean())
df['Vehicle Location'] = df['Vehicle Location'].fillna(df['Vehicle Location'].
 ↪mode()[0])
df['Electric Utility'] = df['Electric Utility'].fillna(df['Electric Utility'].
 ↪mode()[0])
```

[10]:

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

[10]:

```
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
```

[11]:

```
df.to_csv('Electric_Vechile_data')
```

2 Task 1:

This is an open ended problem. Apply Exploratory Data Analysis (Univariate and Bivariate) on the dataset available above.

[12]:

```
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                                   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)
memory usage: 14.6+ MB
```

[13]:

```
df.columns
```

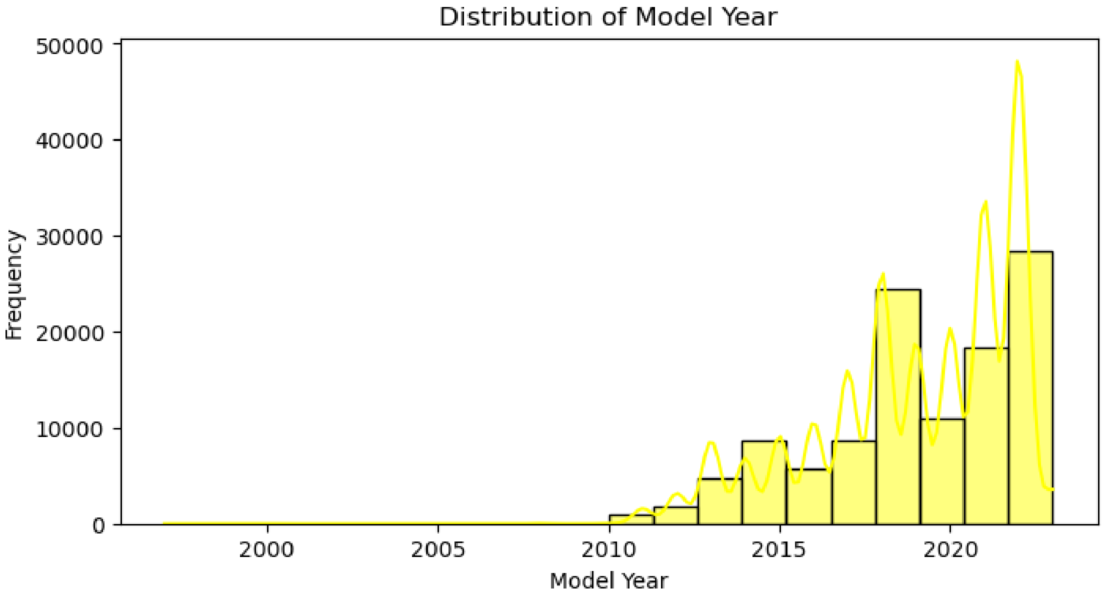
[13]:

```
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')
```

3 Univariate Analysis

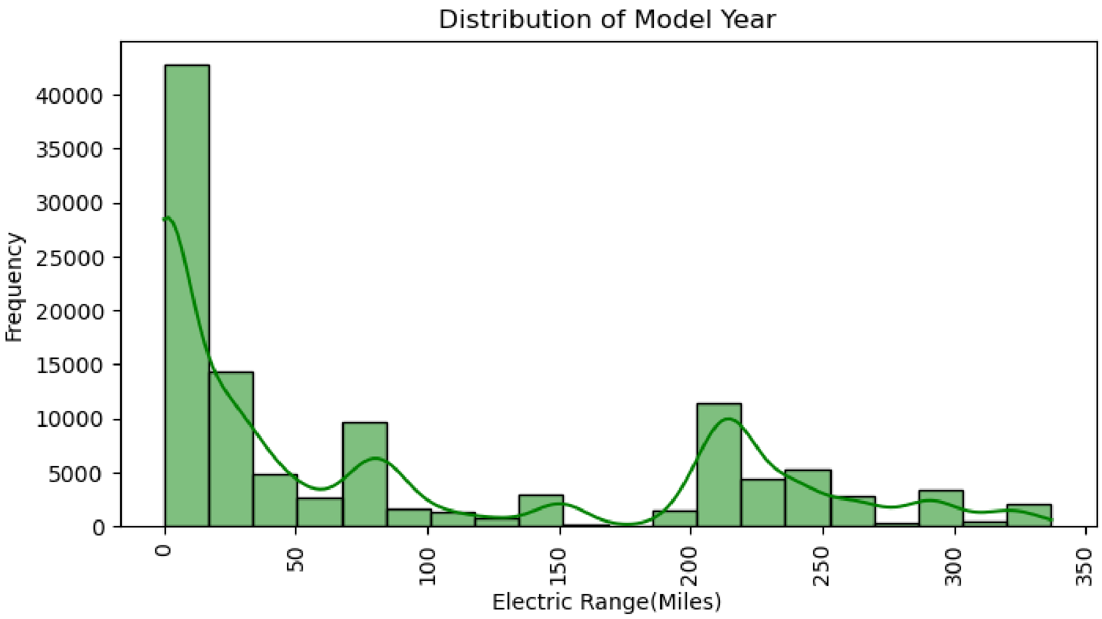
[14]:

```
#Distribution of Model Year
plt.figure(figsize=(8,4))
sns.histplot(df['Model Year'], bins=20, kde=True, color='yellow')
plt.title('Distribution of Model Year')
plt.xlabel('Model Year')
plt.ylabel('Frequency')
plt.show()
```



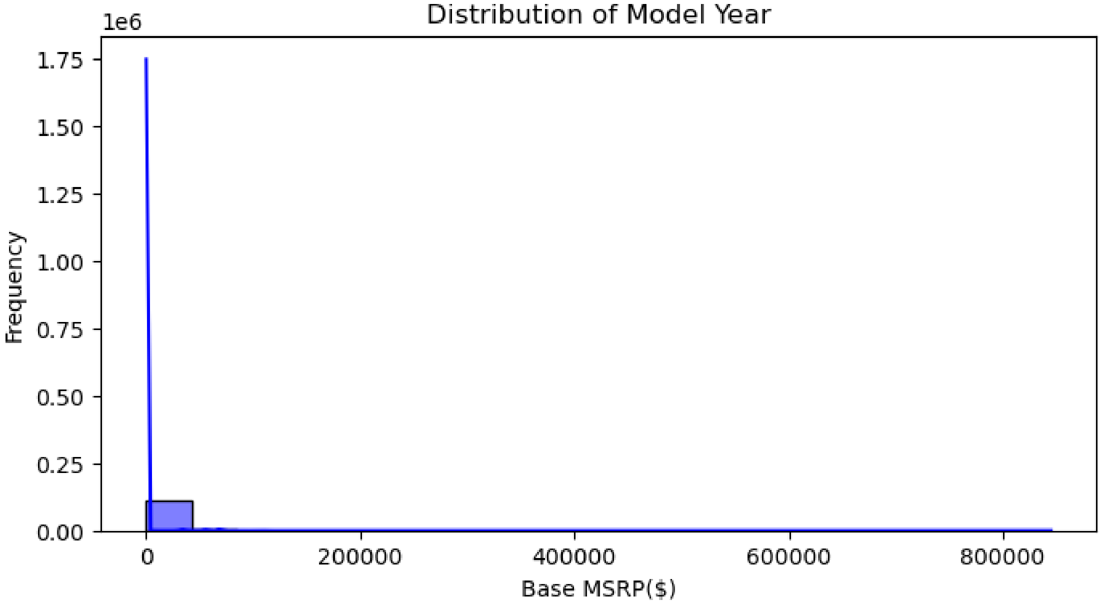
[15]:

```
#Distribution of Electric Range
plt.figure(figsize=(8,4))
sns.histplot(df['Electric Range'], bins=20, kde=True, color='green')
plt.title('Distribution of Model Year')
plt.xticks(rotation=90)
plt.xlabel('Electric Range(Miles)')
plt.ylabel('Frequency')
plt.show()
```



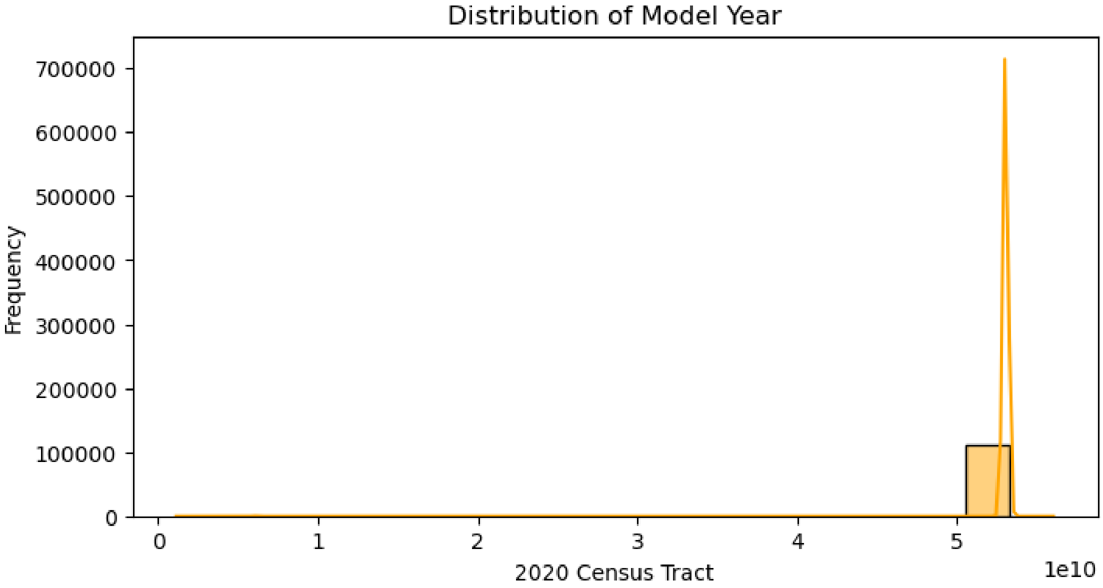
[16]:

```
#Distribution of Base MSRP
plt.figure(figsize=(8,4))
sns.histplot(df['Base MSRP'], bins=20, kde=True, color='blue')
plt.title('Distribution of Model Year')
plt.xlabel('Base MSRP($)')
plt.ylabel('Frequency')
plt.show()
```



[17]:

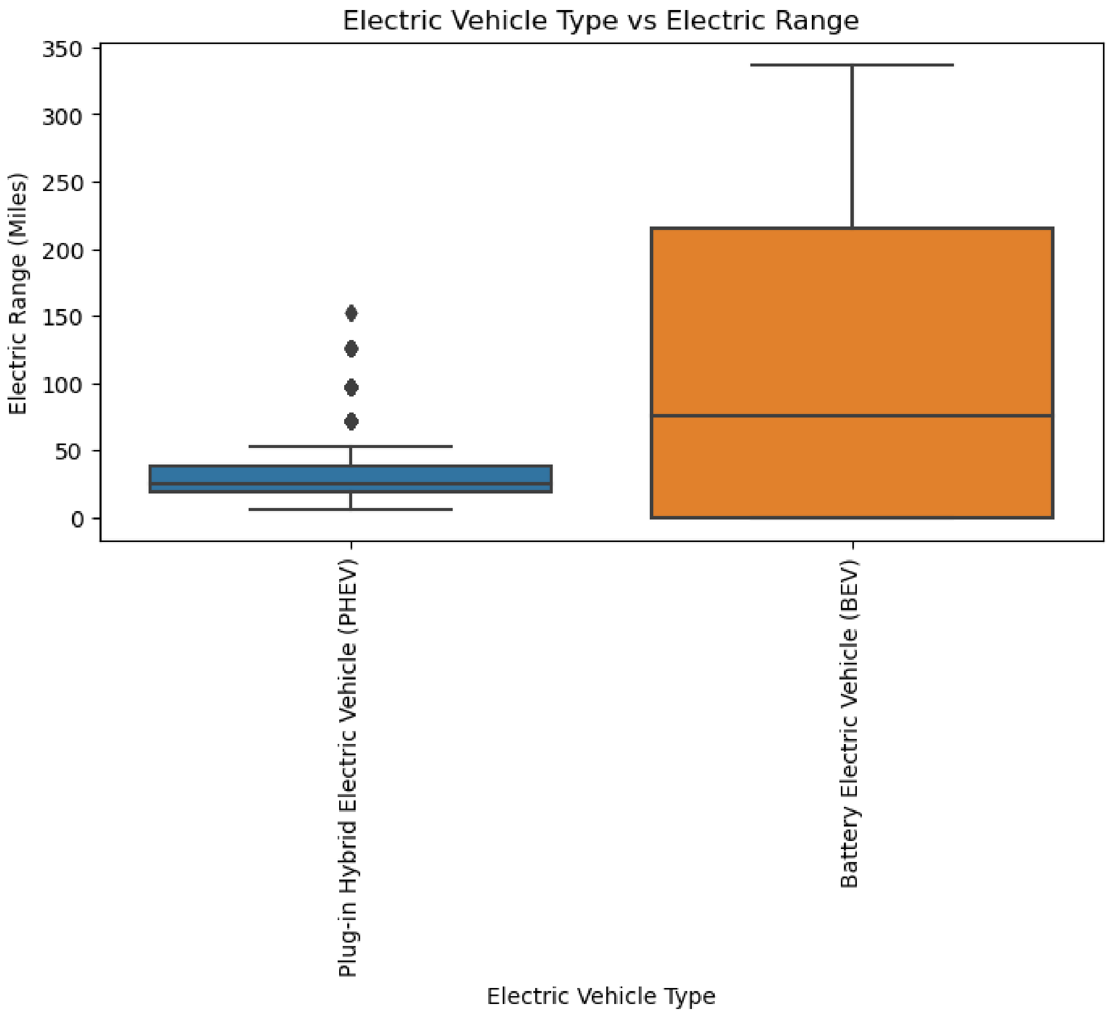
```
#Distribution of 2020 Census Tract
plt.figure(figsize=(8,4))
sns.histplot(df['2020 Census Tract'], bins=20, kde=True, color='orange')
plt.title('Distribution of Model Year')
plt.xlabel('2020 Census Tract')
plt.ylabel('Frequency')
plt.show()
```



4 Bivariate Analysis

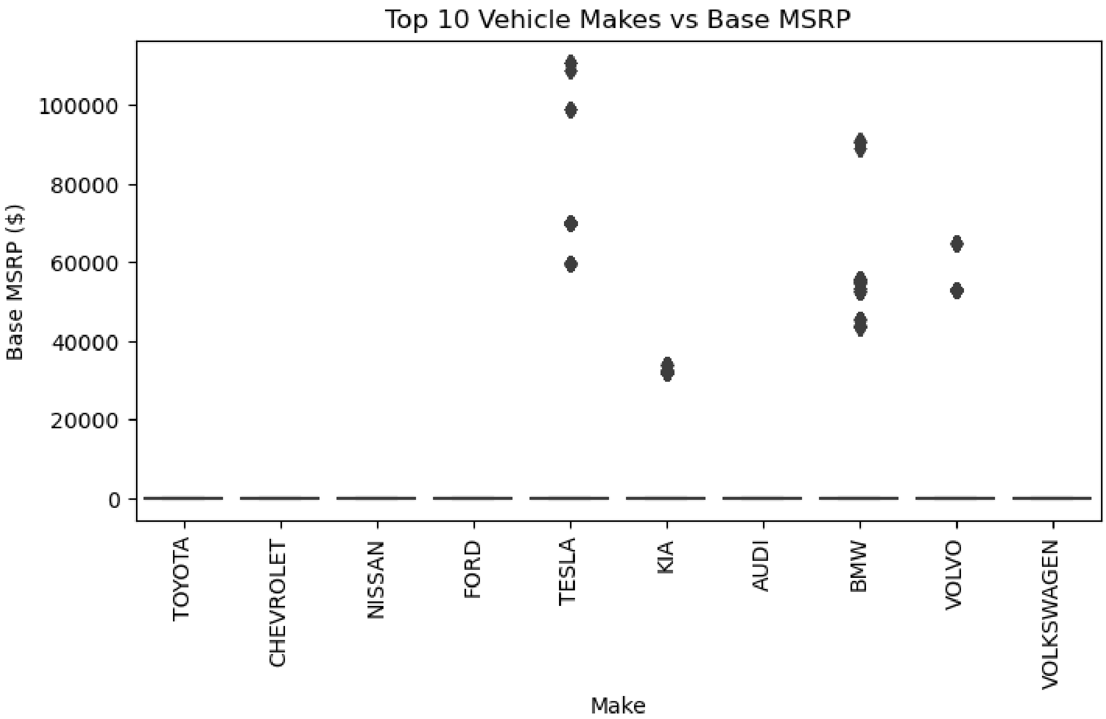
[18]:

```
# Electric Vehicle Type vs Electric Range
plt.figure(figsize=(8,4))
sns.boxplot(x='Electric Vehicle Type', y='Electric Range', data=df)
plt.title('Electric Vehicle Type vs Electric Range')
plt.xticks(rotation=90)
plt.ylabel('Electric Range (Miles)')
plt.show()
```



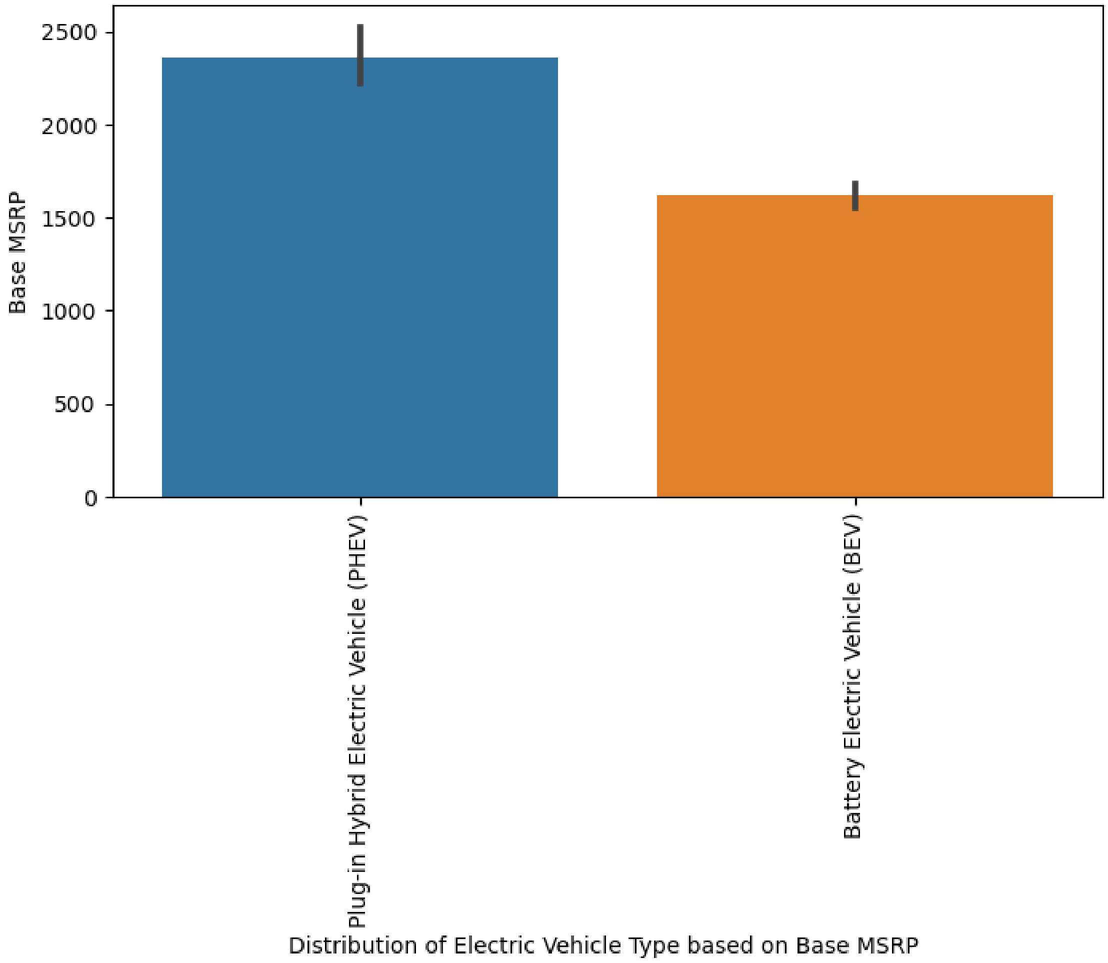
[19]:

```
#Make vs Base MSRP
plt.figure(figsize=(8,4))
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=90)
plt.ylabel('Base MSRP ($)')
plt.show()
```



[20]:
```
# Electric Vehicle Type vs Base MSRP
plt.figure(figsize=(8,4))
sns.barplot(x='Electric Vehicle Type', y='Base MSRP',data=df)
plt.xticks(rotation = 90)
plt.xlabel('Distribution of Electric Vehicle Type based on Base MSRP ')
plt.ylabel('Base MSRP')
plt.show
```

[20]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

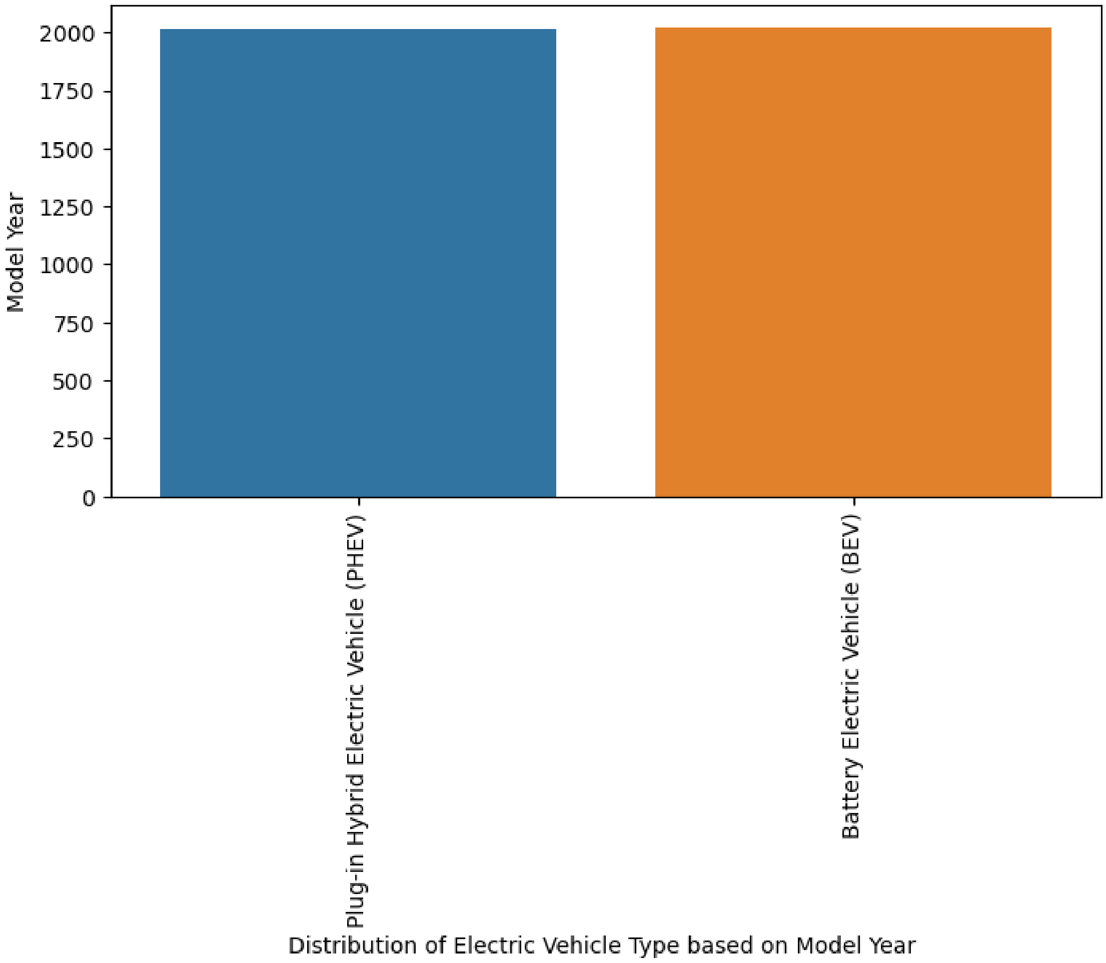


[21]:

```
# Electric Vehicle Type vs Model Year
plt.figure(figsize=(8,4))
sns.barplot(x='Electric Vehicle Type', y='Model Year',data=df)
plt.xticks(rotation = 90)
plt.xlabel('Distribution of Electric Vehicle Type based on Model Year ')
plt.ylabel('Model Year')
plt.show
```

[21]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



5 Task 2:

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

[22]:
```
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()
```

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.

[23]:

```
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

[24]:

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

[25]:

```
!pip install bar-chart-race
```

```
Collecting bar-chart-race
  Obtaining dependency information for bar-chart-race from https://files.pythonh
osted.org/packages/09/01/f6d1a1a0978b39560843c54be7349804d7d2faef0a869acd7c8a6fc
920b0/bar_chart_race-0.1.0-py3-none-any.whl.metadata
  Downloading bar_chart_race-0.1.0-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: pandas>=0.24 in
c:\users\maazp\anaconda3\lib\site-packages (from bar-chart-race) (2.0.3)
Requirement already satisfied: matplotlib>=3.1 in
c:\users\maazp\anaconda3\lib\site-packages (from bar-chart-race) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (1.0.5)
Requirement already satisfied: cycler>=0.10 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\maazp\anaconda3\lib\site-
packages (from matplotlib>=3.1->bar-chart-race) (1.24.3)
Requirement already satisfied: packaging>=20.0 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\maazp\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\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\maazp\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-
race) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\maazp\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\maazp\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
(2023.3)
Requirement already satisfied: six>=1.5 in c:\users\maazp\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 -:--:--
   ----- ---------------------------------- 20.5/156.8 kB ? eta -:--:--
   ------------------------------- -------- 122.9/156.8 kB 1.8 MB/s eta 0:00:01
   ---------------------------------------- 156.8/156.8 kB 1.6 MB/s eta 0:00:00
```

```
Installing collected packages: bar-chart-race
Successfully installed bar-chart-race-0.1.0
```

[26]:

```python
import bar_chart_race as bcr
import warnings

# 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()
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: