

data-analysis-on-electric-vehicle

October 6, 2024

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px # for choropleth map and plotly visualization
from matplotlib.animation import FuncAnimation
import plotly.graph_objects as go # for racing bar plot .....
```

load dataset

```
[ ]: df=pd.read_csv('/content/dataset (2).csv')
```

```
[ ]: df.head(6)
```

```
[ ]: 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  3FA6POSU1K      Snohomish  Everett   WA        98201      2019    FORD
5  5YJ3E1EB5J      Snohomish  Bothell   WA        98021      2018    TESLA
```

```
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)
5  MODEL 3      Battery Electric Vehicle (BEV)
```

```
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
5      Clean Alternative Fuel Vehicle Eligible     215
```

```
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
5	0	1.0	475635324

	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
5	POINT (-122.18384 47.8031)	PUGET SOUND ENERGY INC	53061051916

```
[ ]: df.tail(6)
```

```
[ ]:
      VIN (1-10)  County  City State  Postal Code  Model Year  \
112628 WA1E2AFY7L   King   Seattle   WA      98177      2020
112629 7SAYGDEF2N   King   Duvall    WA      98019      2022
112630 1N4BZ1CP7K San Juan Friday Harbor WA      98250      2019
112631 1FMCUOKZ4N   King   Vashon    WA      98070      2022
112632 KND3CD3LD4J  King   Covington  WA      98042      2018
112633 YV4BR0CL8N   King   Covington  WA      98042      2022
```

	Make	Model	Electric Vehicle Type	\
112628	AUDI	Q5 E	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	\
112628	Not eligible due to low battery range	20	
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	\
112628	0	32.0	9880104	
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 \
112628 POINT (-122.36498 47.72238)
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
112628 CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)      53033000500
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

```

```
[ ]: print(df.shape)
```

```
(112634, 17)
```

```
[ ]: print(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')

```

```
[ ]: 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

```

```
[ ]: df.describe() # summary statistics
```

```

[ ]:
      Postal Code    Model Year  Electric Range    Base MSRP  \
count  112634.000000  112634.000000  112634.000000  112634.000000
mean    98156.226850    2019.003365     87.812987    1793.439681
std     2648.733064      2.892364    102.334216   10783.753486
min     1730.000000    1997.000000      0.000000      0.000000
25%    98052.000000    2017.000000      0.000000      0.000000
50%    98119.000000    2020.000000     32.000000      0.000000
75%    98370.000000    2022.000000    208.000000      0.000000
max    99701.000000    2023.000000   337.000000  845000.000000

      Legislative District  DOL Vehicle ID  2020 Census Tract
count          112348.000000    1.126340e+05    1.126340e+05
mean           29.805604    1.994567e+08    5.296650e+10
std           14.700545    9.398427e+07    1.699104e+09
min            1.000000    4.777000e+03    1.101001e+09
25%           18.000000    1.484142e+08    5.303301e+10
50%           34.000000    1.923896e+08    5.303303e+10
75%           43.000000    2.191899e+08    5.305307e+10
max           49.000000    4.792548e+08    5.603300e+10

```

Data Cleaning & Preprocessing

```
[ ]: df.isnull().sum() # missing values
```

```

[ ]: 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	

```
[ ]: # Check for missing values
      #print(df.isnull().sum())

      # Drop rows or handle missing values accordingly
      df.dropna(subset=['VIN (1-10)', 'Make', 'Electric Vehicle Type'], inplace=True)

      # Convert necessary columns to appropriate data types
      df['Model Year'] = df['Model Year'].astype(int)
      df['Electric Range'] = df['Electric Range'].astype(float)
```

Exploratory Data Analysis (EDA)

Task 1: Univariate & Bivariate

Objective: Perform EDA to get insights from the dataset. Focus on distribution, relationships, and patterns in the data.

Instructions:

Perform univariate analysis (like histograms) for single variables.

Perform bivariate analysis (correlation, scatter plots, etc.).

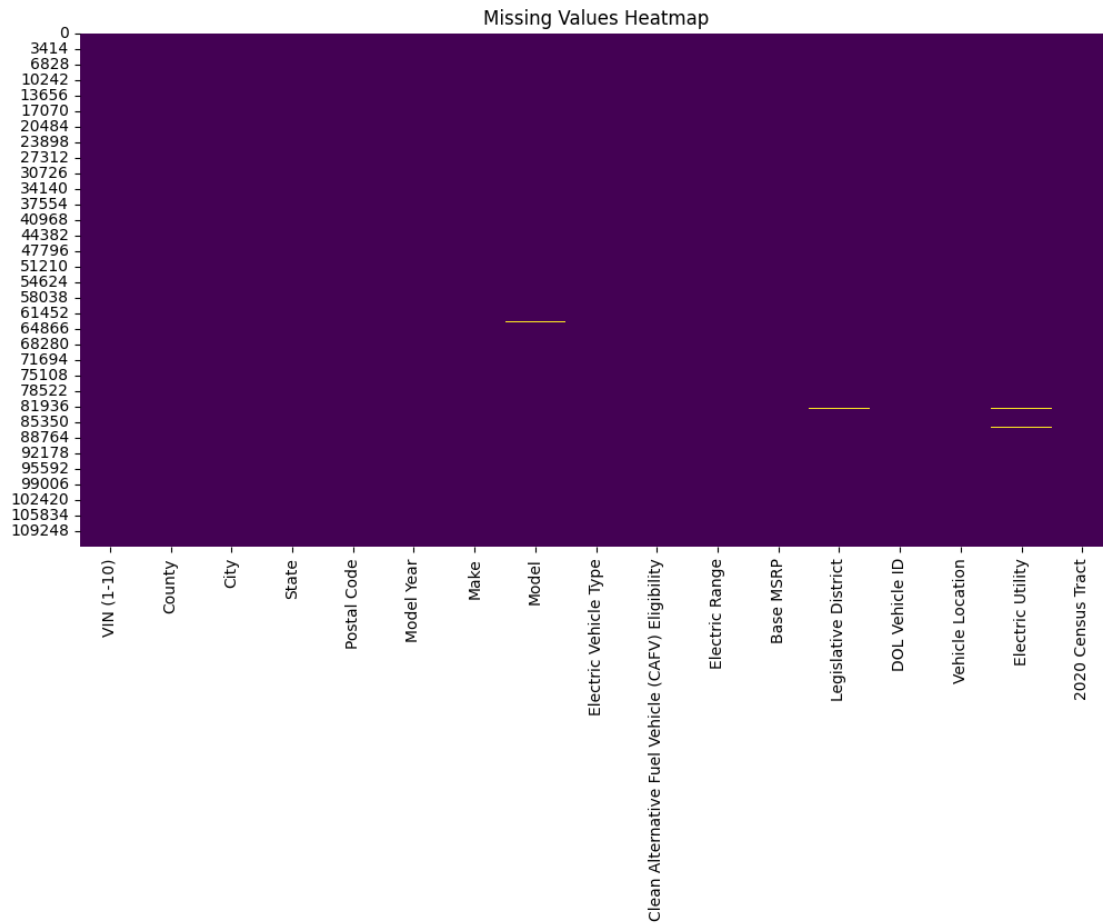
```
[ ]: # Display unique values for categorical columns
      print("Unique values in 'Make':", df['Make'].nunique())
      print("Unique values in 'Electric Vehicle Type':", df['Electric Vehicle Type'].
            ↪nunique())
```

Unique values in 'Make': 34

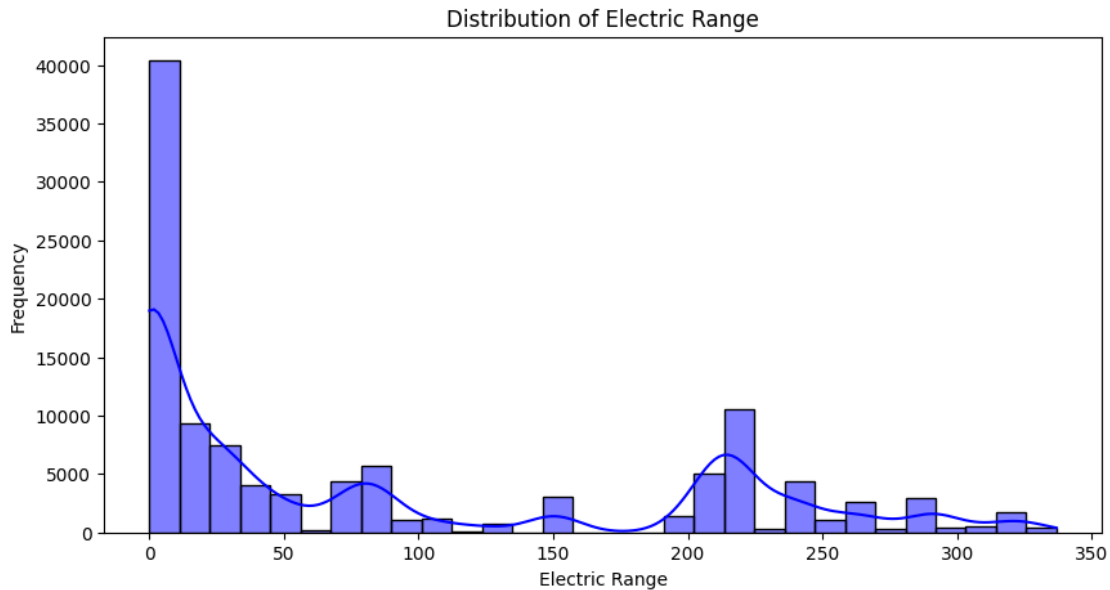
Unique values in 'Electric Vehicle Type': 2

```
[ ]: # Check for missing values
      missing_values = df.isnull().sum()

      # Visualize missing values using a heatmap
      plt.figure(figsize=(12, 6))
      sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
      plt.title('Missing Values Heatmap')
      plt.show()
```



```
[ ]: # Distribution of Electric Range
plt.figure(figsize=(10, 5))
sns.histplot(df['Electric Range'], bins=30, kde=True, color='blue')
plt.title('Distribution of Electric Range')
plt.xlabel('Electric Range')
plt.ylabel('Frequency')
plt.show()
```



```
[ ]: # Count plot for EV Make
plt.figure(figsize=(12, 6))
top_makes = df['Make'].value_counts().nlargest(10)
sns.barplot(x=top_makes.index, y=top_makes.values, palette='viridis')
plt.title('Top 10 Electric Vehicle Makes')
plt.xlabel('Make')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-35-892bb5dd2a2b>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple

to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

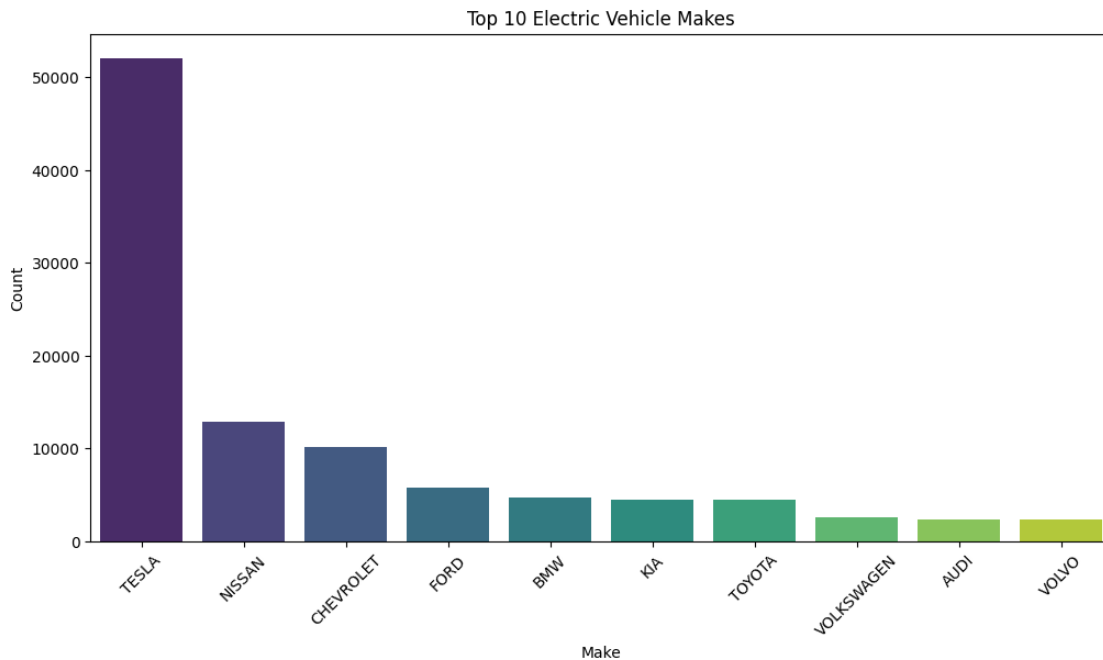
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple

to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.



```
[ ]: # Count plot for Electric Vehicle Type
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Electric Vehicle Type', palette='Set2')
plt.title('Count of Electric Vehicle Types')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-36-a3908e8a4138>:3: FutureWarning:

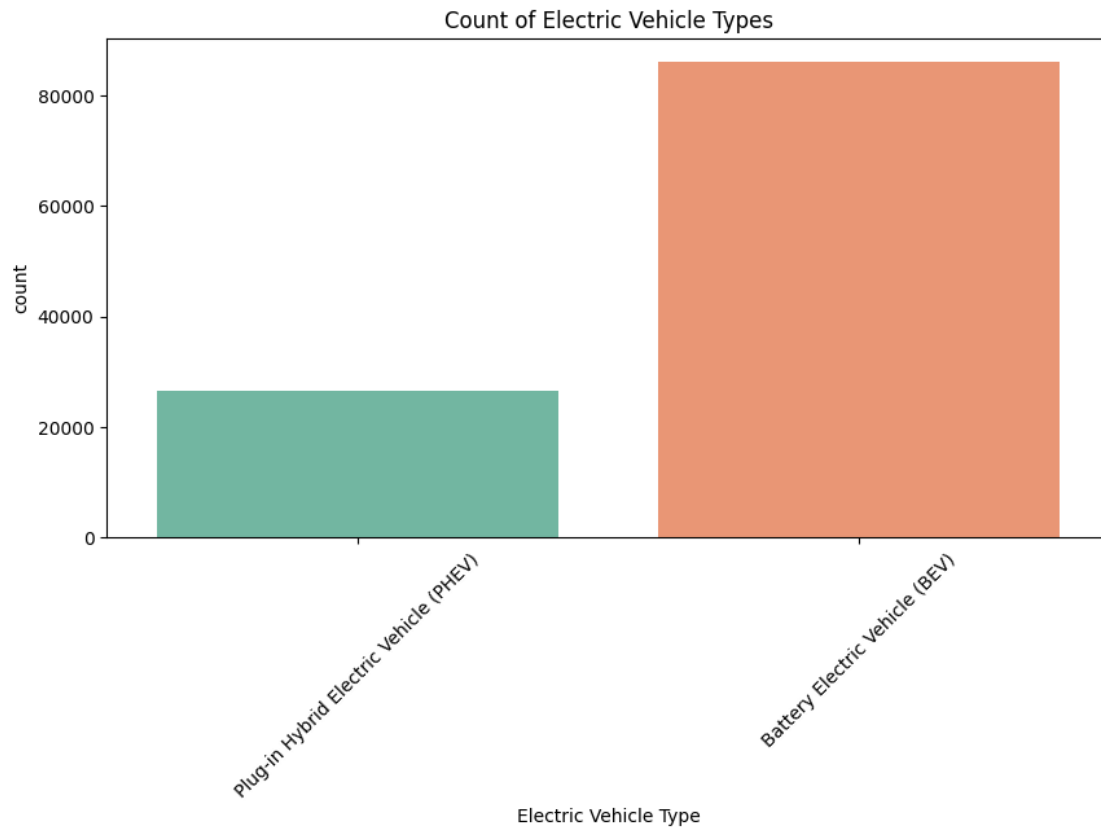
Passing ``palette`` without assigning ``hue`` is deprecated and will be removed in v0.14.0. Assign the ``x`` variable to ``hue`` and set ``legend=False`` for the same effect.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.



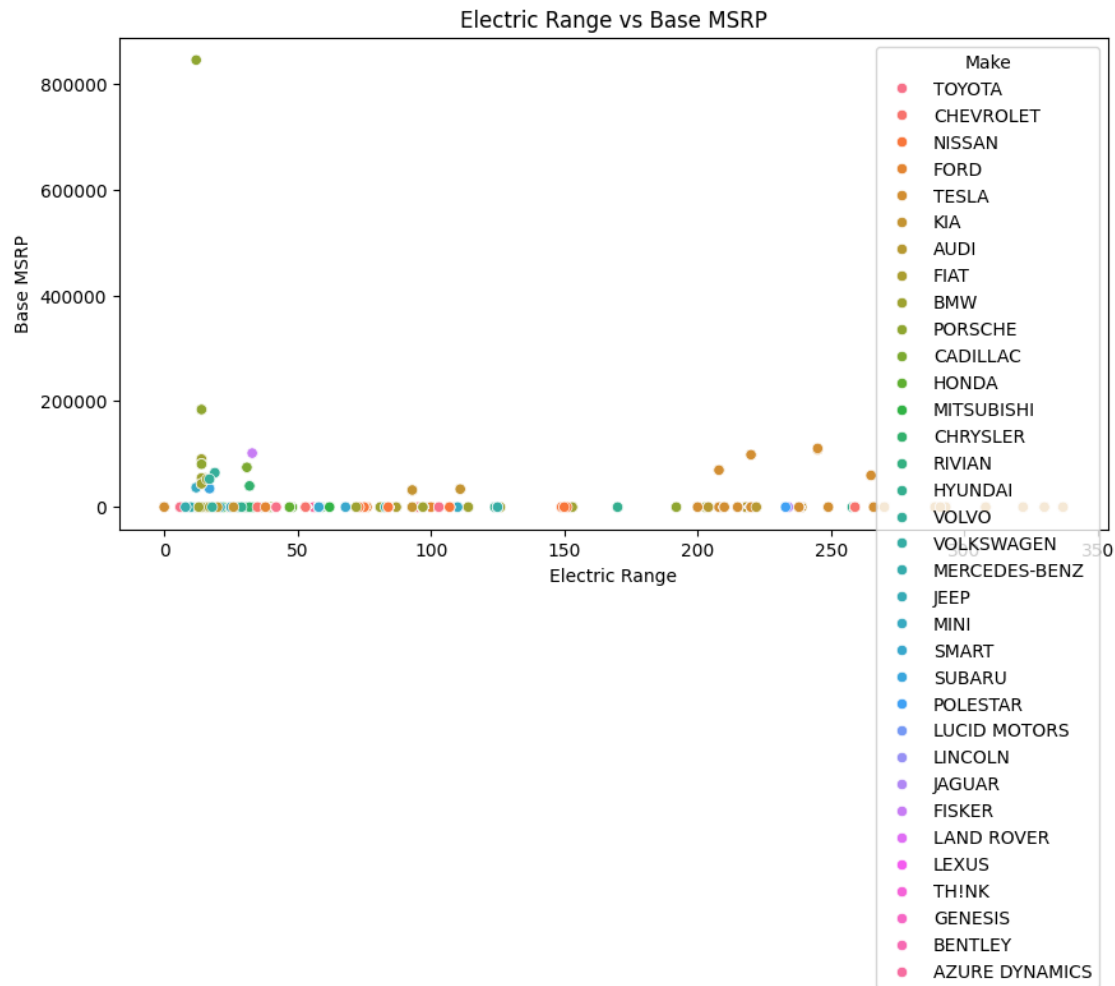
Bivariate Analysis

Objective: Investigate relationships between two features.

Instructions:

Use scatter plots, box plots, and correlation matrices.

```
[ ]: # Bivariate Analysis - Relationship between Electric Range and Base MSRP
plt.figure(figsize=(10, 5))
sns.scatterplot(x='Electric Range', y='Base MSRP', data=df, hue='Make')
plt.title('Electric Range vs Base MSRP')
plt.show()
```



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

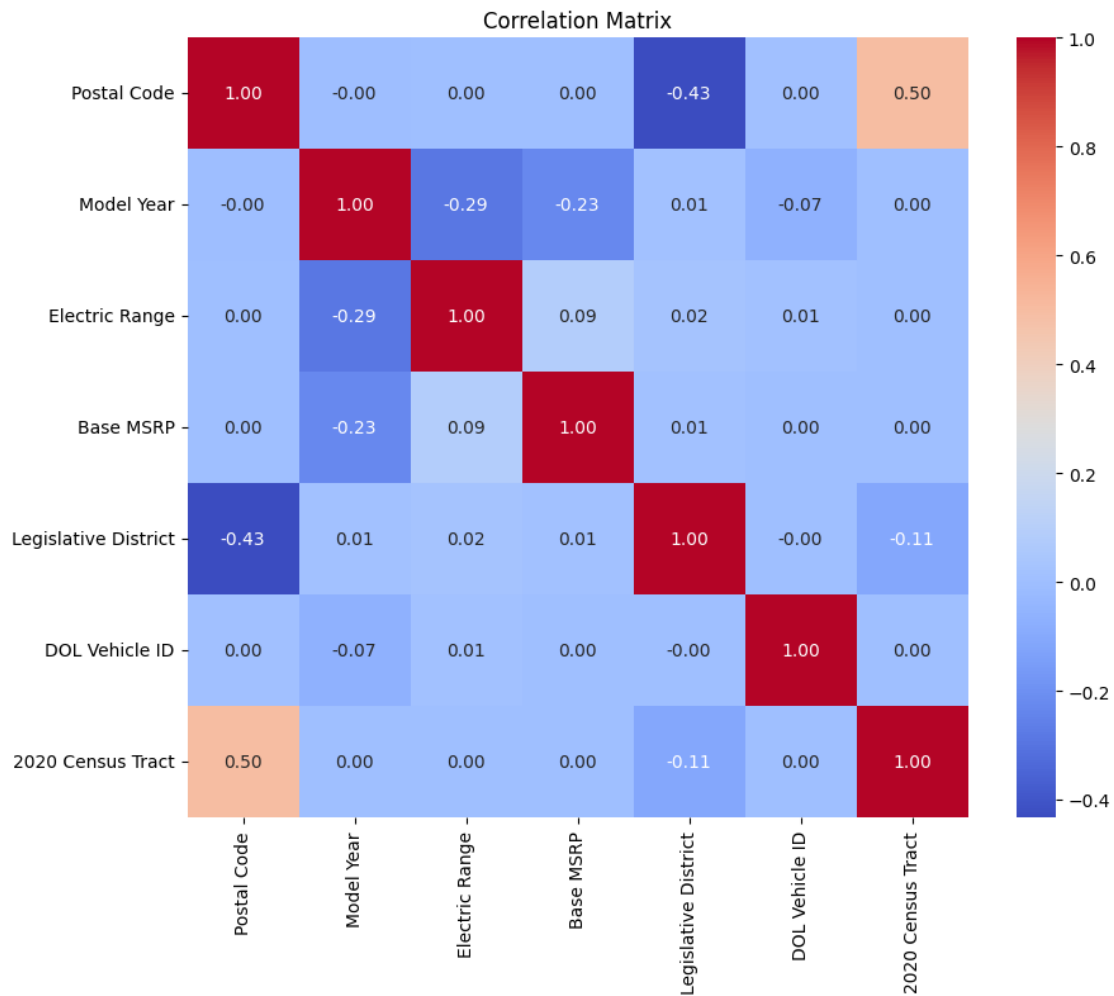
# Assuming df is already defined and preprocessed

# Step 1: Filter numeric columns
numeric_df = df.select_dtypes(include=['int64', 'float64'])

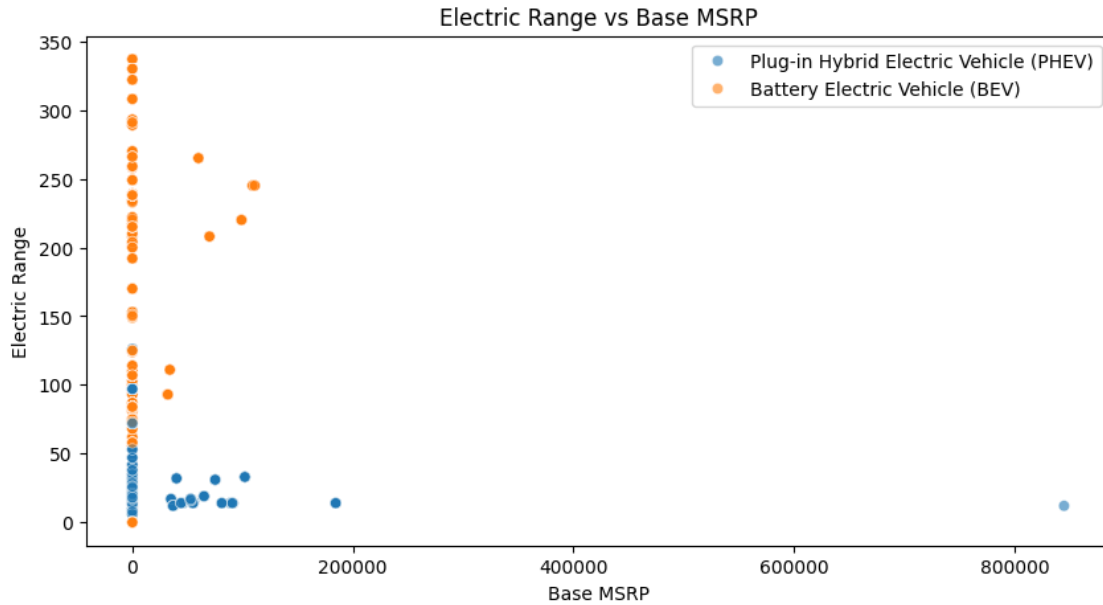
# Step 2: Calculate the correlation matrix
corr = numeric_df.corr()

# Step 3: Plot the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', square=True)
plt.title('Correlation Matrix')
```

```
plt.show()
```



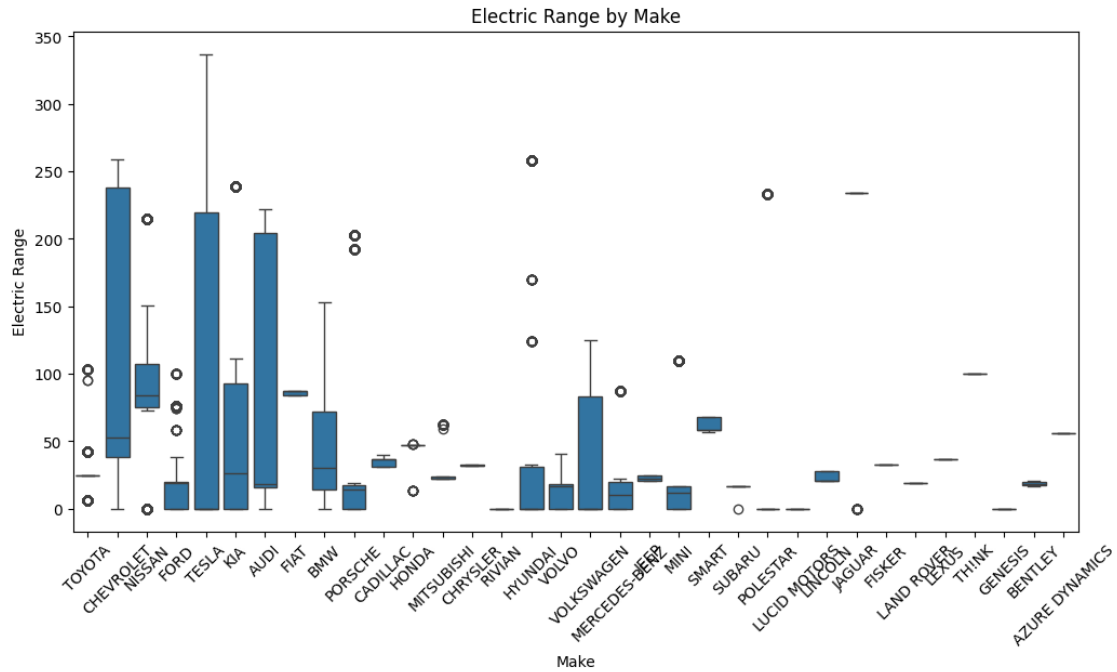
```
[ ]: # Scatter plot for Electric Range vs Base MSRP
plt.figure(figsize=(10, 5))
sns.scatterplot(data=df, x='Base MSRP', y='Electric Range', hue='Electric_Vehicle Type', alpha=0.6)
plt.title('Electric Range vs Base MSRP')
plt.xlabel('Base MSRP')
plt.ylabel('Electric Range')
plt.legend()
plt.show()
```



```
[ ]: # Box plot for Electric Range across different Makes
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='Make', y='Electric Range')
plt.title('Electric Range by Make')
plt.xticks(rotation=45)
plt.show()
```

/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning:

SeriesGroupBy.grouper is deprecated and will be removed in a future version of pandas.



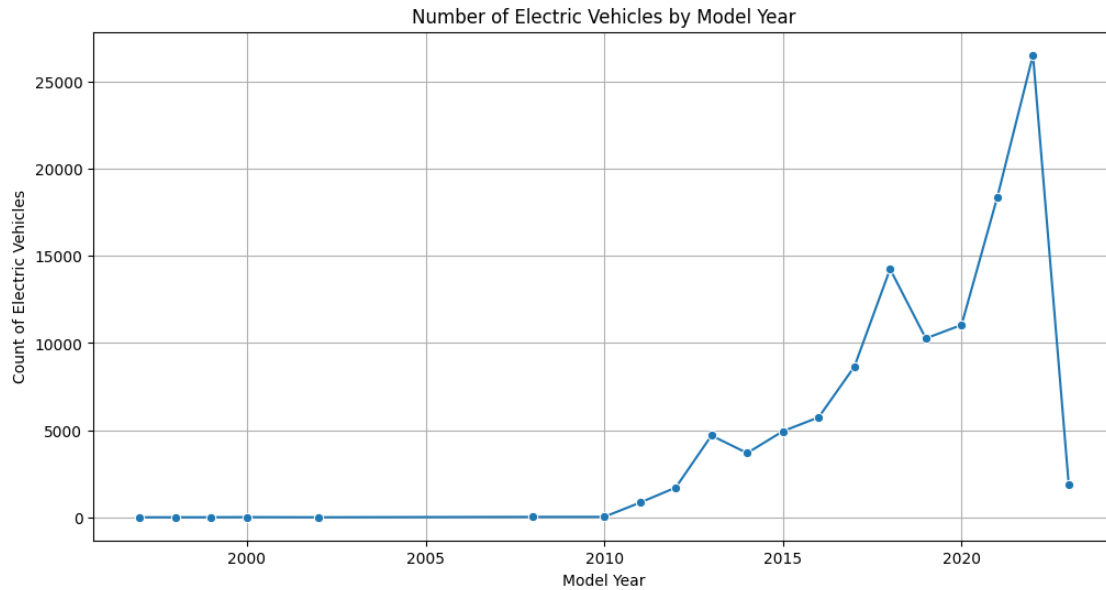
Advanced Analysis

Objective: Gain deeper insights into specific aspects of the dataset.

Instructions:

Analyze trends over time, using line plots and animations.

```
[ ]: # Time series analysis: Count of EVs by Model Year
ev_count_by_year = df['Model Year'].value_counts().sort_index()
plt.figure(figsize=(12, 6))
sns.lineplot(x=ev_count_by_year.index, y=ev_count_by_year.values, marker='o')
plt.title('Number of Electric Vehicles by Model Year')
plt.xlabel('Model Year')
plt.ylabel('Count of Electric Vehicles')
plt.grid()
plt.show()
```



```
[ ]: # Analyze Electric Utility usage
plt.figure(figsize=(12, 6))
top_utilities = df['Electric Utility'].value_counts().nlargest(10)
sns.barplot(x=top_utilities.index, y=top_utilities.values, palette='rocket')
plt.title('Top 10 Electric Utilities for EVs')
plt.xlabel('Electric Utility')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-44-0bf26ca796b0>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple

to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

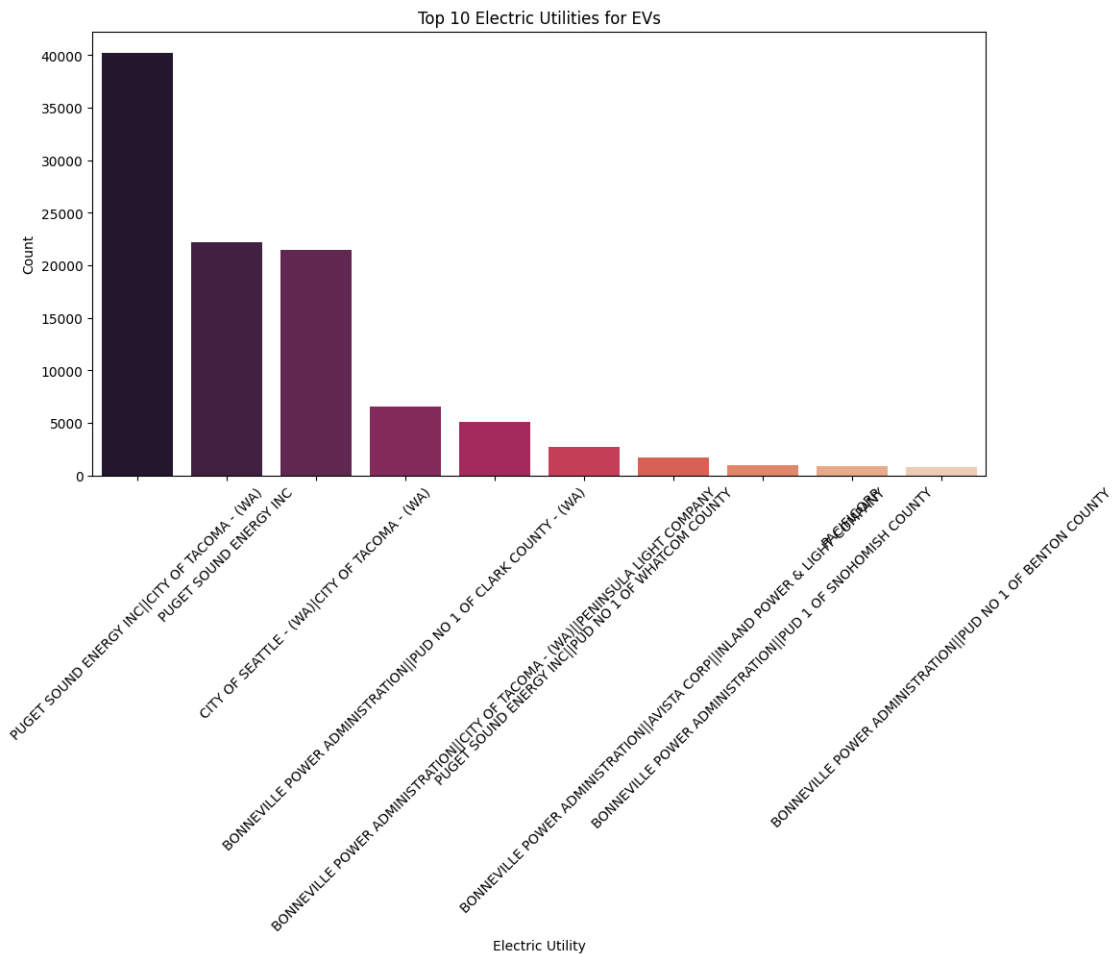
```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:
```

When grouping with a length-1 list-like, you will need to pass a length-1 tuple

to `get_group` in a future version of pandas. Pass ``(name,)`` instead of ``name`` to silence this warning.



```
[ ]: import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Creating a 3D scatter plot
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

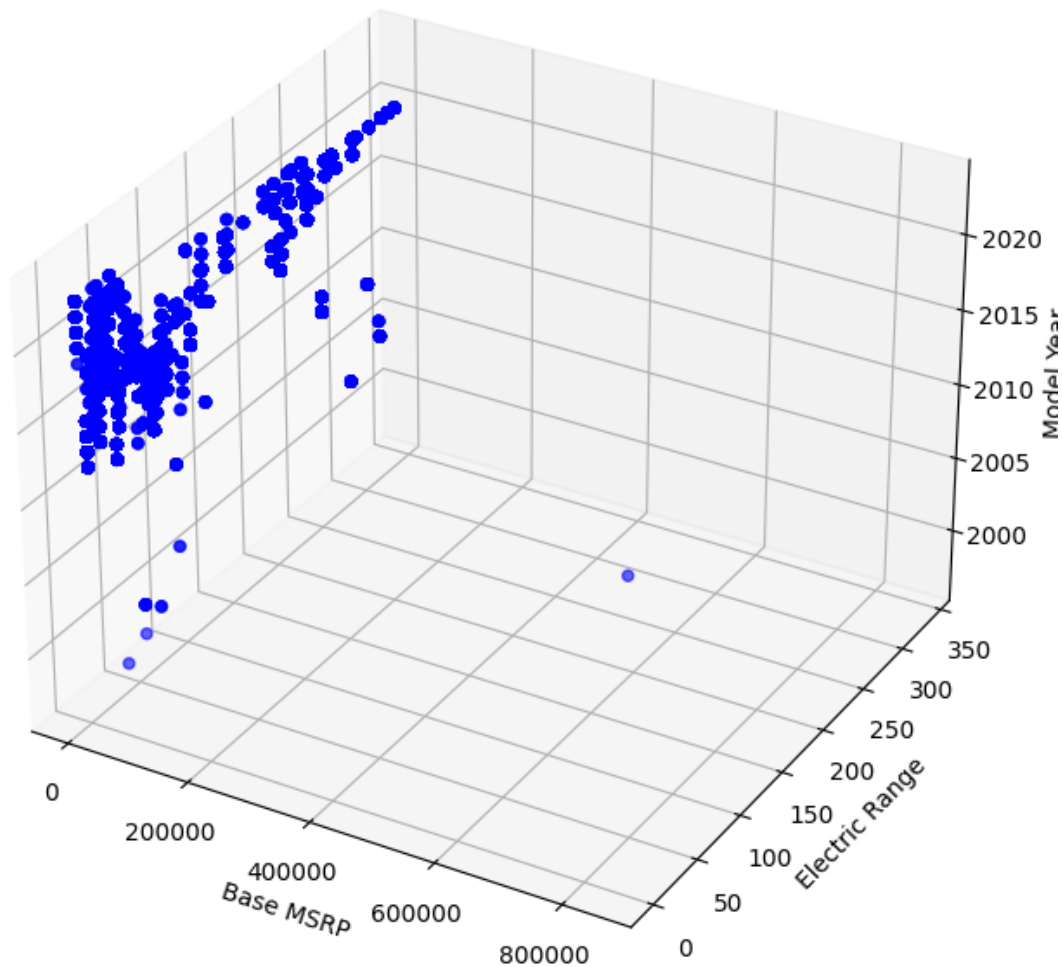
# Scatter plot points
ax.scatter(df['Base MSRP'], df['Electric Range'], df['Model Year'],
          c='b', marker='o', alpha=0.6)

# Setting labels
ax.set_xlabel('Base MSRP')
ax.set_ylabel('Electric Range')
```

```
ax.set_zlabel('Model Year')
ax.set_title('3D Scatter Plot: Electric Range vs Base MSRP vs Model Year')

plt.show()
```

3D Scatter Plot: Electric Range vs Base MSRP vs Model Year



```
[ ]: from mpl_toolkits.mplot3d import Axes3D
import numpy as np

# Create a meshgrid for Base MSRP and Model Year
x = df['Base MSRP']
y = df['Model Year']
x, y = np.meshgrid(x.unique(), y.unique())
```

```

# Calculate Z values (Electric Range) for the surface plot
z = np.zeros_like(x)

for i in range(len(x)):
    for j in range(len(y)):
        # Getting the Electric Range for each Base MSRP and Model Year
        electric_range = df[(df['Base MSRP'] == x[i, j]) & (df['Model Year'] == y[i, j])]
        if not electric_range.empty:
            z[i, j] = electric_range['Electric Range'].mean() # Taking mean
# for surface

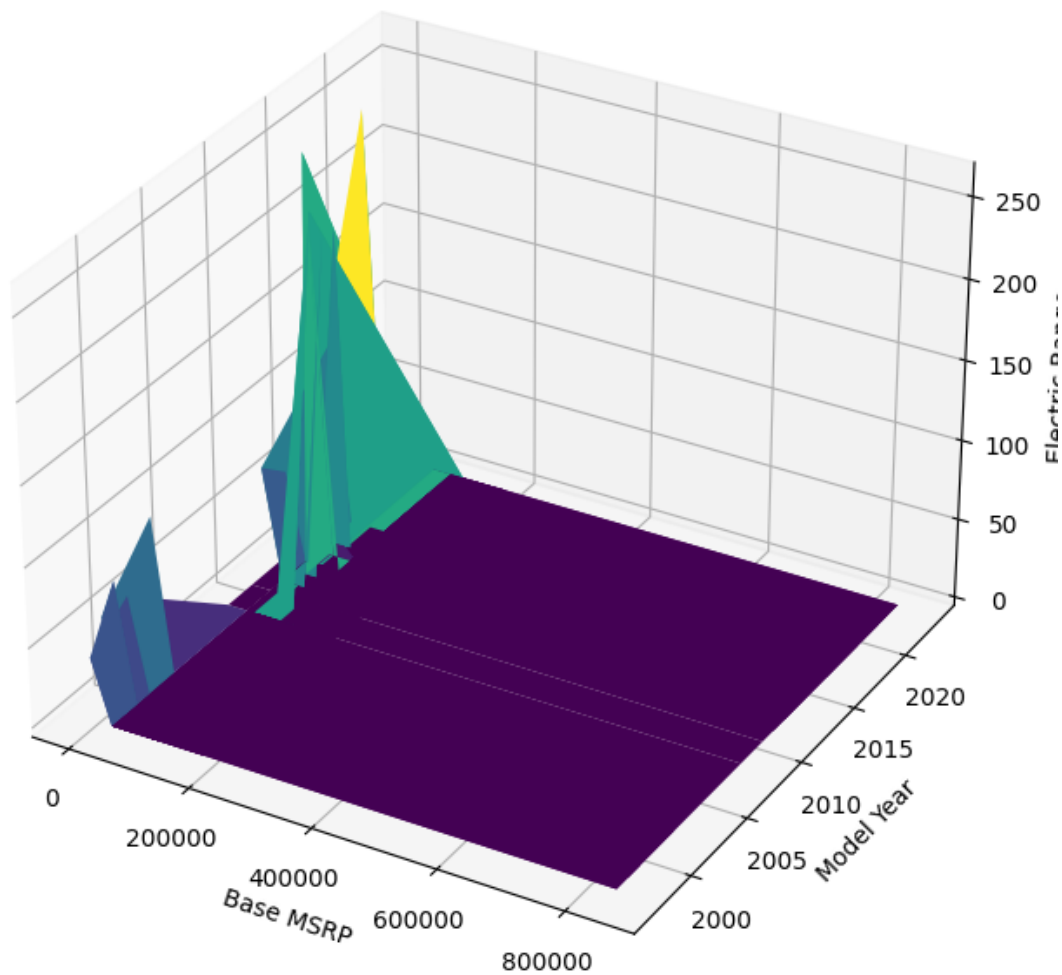
# Creating the surface plot
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z, cmap='viridis', edgecolor='none')

# Setting labels
ax.set_xlabel('Base MSRP')
ax.set_ylabel('Model Year')
ax.set_zlabel('Electric Range')
ax.set_title('3D Surface Plot: Electric Range based on Base MSRP and Model Year')

plt.show()

```

3D Surface Plot: Electric Range based on Base MSRP and Model Year



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

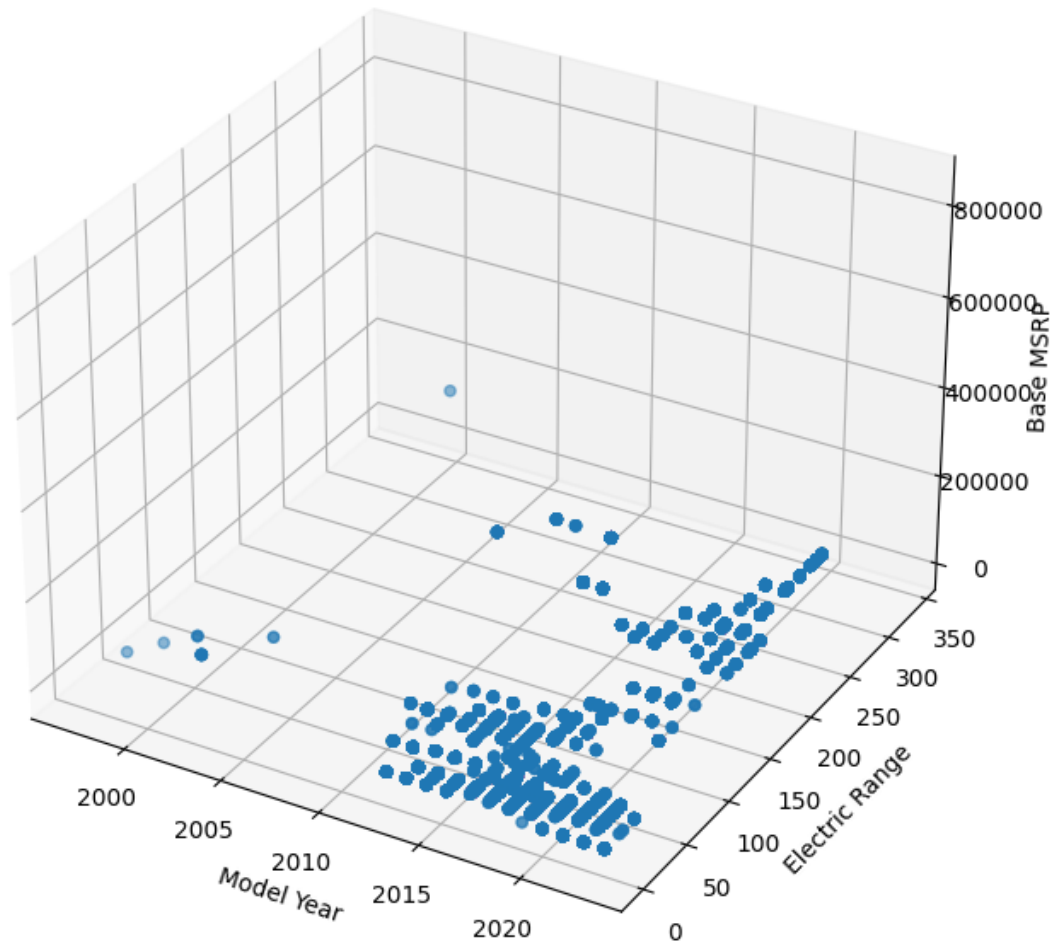
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Create a scatter plot
ax.scatter(df['Model Year'], df['Electric Range'], df['Base MSRP'], alpha=0.5)

ax.set_xlabel('Model Year')
ax.set_ylabel('Electric Range')
ax.set_zlabel('Base MSRP')
ax.set_title('3D Scatter Plot: Electric Range vs Base MSRP over Model Years')
```

```
plt.show()
```

3D Scatter Plot: Electric Range vs Base MSRP over Model Years



```
[ ]: print(grouped_data.shape)
      print(grouped_data.head())
```

(34, 20)

Model Year	1997	1998	1999	2000	2002	2008	2010	2011	2012	2013	\
Make											
AUDI	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
AZURE DYNAMICS	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	3.0	NaN	
BENTLEY	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
BMW	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
CADILLAC	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Model Year	2014	2015	2016	2017	2018	2019	2020	2021	2022	\
Make										
AUDI	NaN	NaN	214.0	187.0	174.0	392.0	224.0	544.0	585.0	
AZURE DYNAMICS	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
BENTLEY	NaN	NaN	NaN	NaN	NaN	NaN	1.0	1.0	1.0	
BMW	457.0	403.0	383.0	692.0	710.0	279.0	143.0	635.0	905.0	
CADILLAC	58.0	NaN	18.0	15.0	17.0	NaN	NaN	NaN	NaN	

Model Year	2023
Make	
AUDI	12.0
AZURE DYNAMICS	NaN
BENTLEY	NaN
BMW	73.0
CADILLAC	NaN

```
[ ]: grouped_data = grouped_data.reset_index()
```

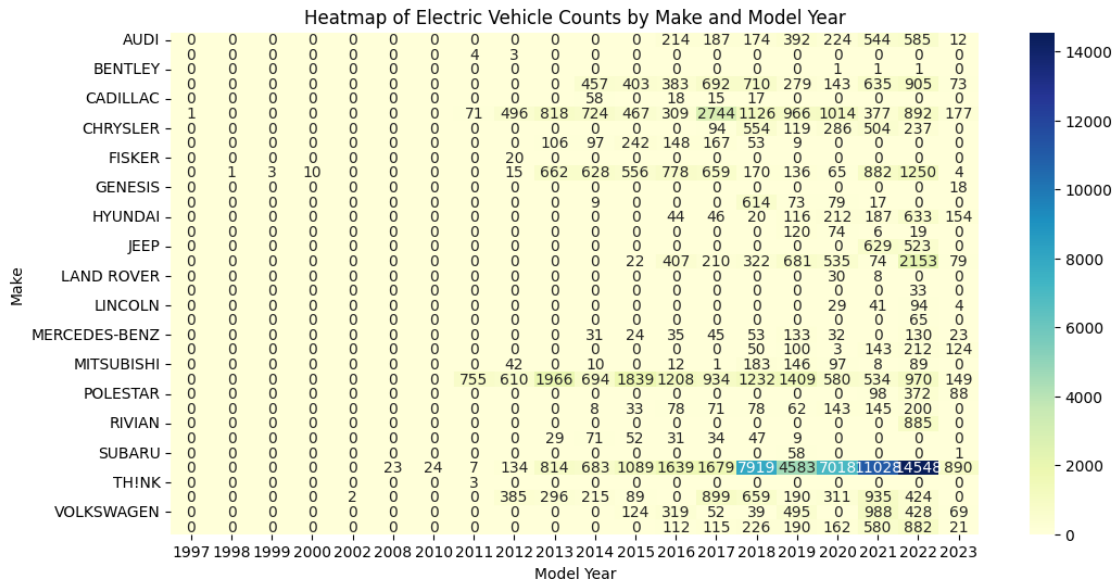
```
[ ]: grouped_data = grouped_data.fillna(0) # Replace NaNs with zeros
```

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

# Grouping data for heatmap
grouped_data = df.groupby(['Make', 'Model Year'])['Electric Range'].count().
    ↪unstack(fill_value=0)

# Reset index and fill NaNs if any
grouped_data = grouped_data.fillna(0)

# Plotting the heatmap
plt.figure(figsize=(12, 6))
sns.heatmap(grouped_data, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Heatmap of Electric Vehicle Counts by Make and Model Year')
plt.xlabel('Model Year')
plt.ylabel('Make')
plt.show()
```

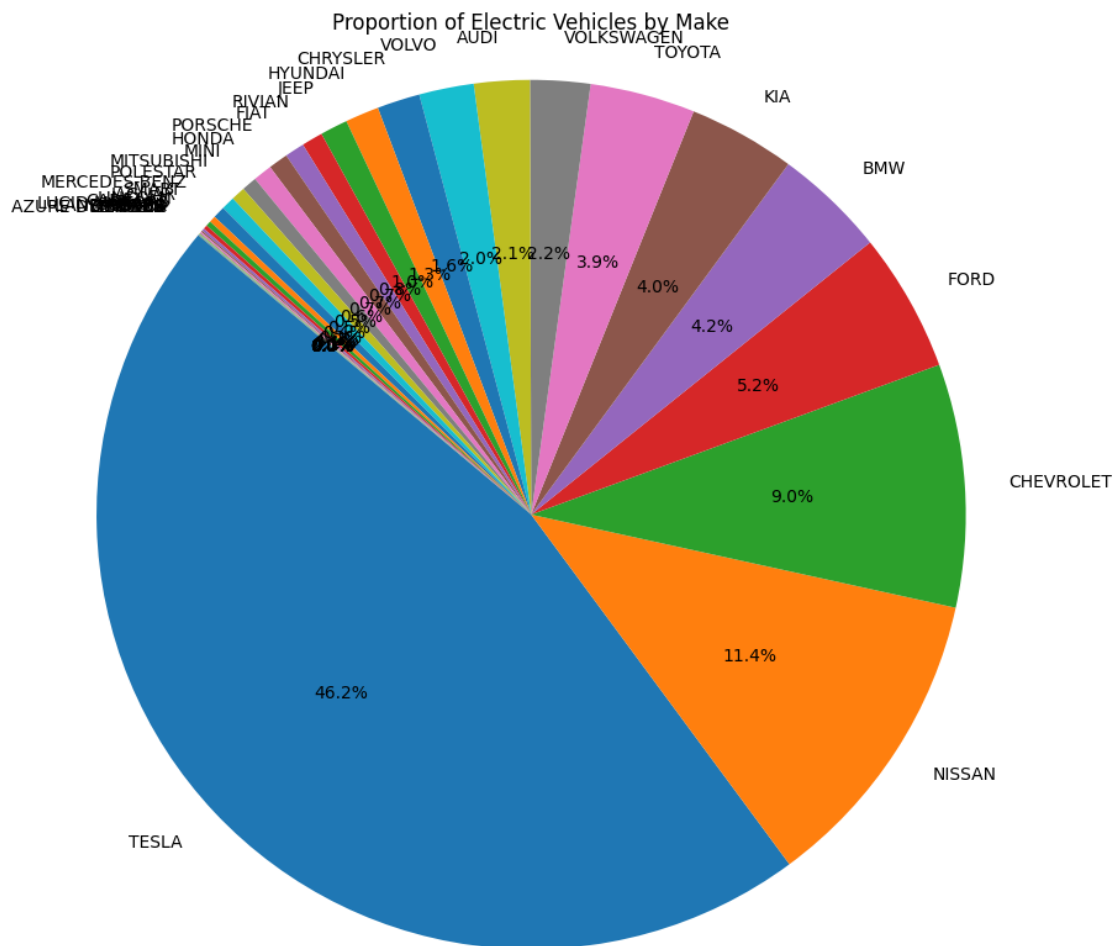


```
[ ]: g = sns.FacetGrid(df, col="Make", col_wrap=4, height=4)
g.map(sns.histplot, "Electric Range", kde=True)
g.set_titles(col_template="{col_name}")
plt.subplots_adjust(top=0.9)
g.fig.suptitle('Distribution of Electric Range by Make')
plt.show()
```

Distribution of Electric Range by Make



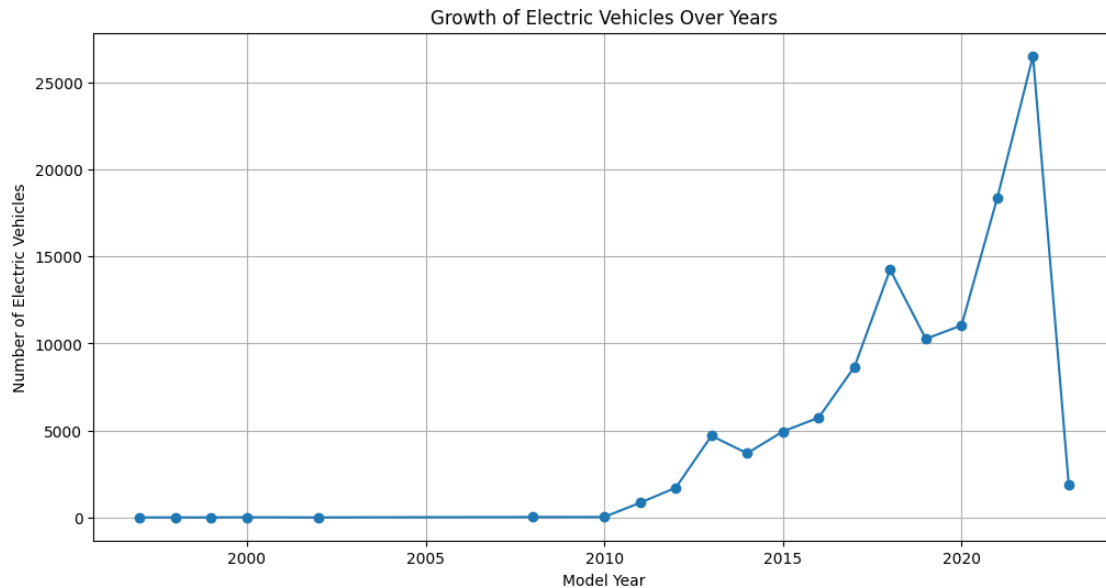

```
[ ]: make_counts = df['Make'].value_counts()
plt.figure(figsize=(10, 10))
plt.pie(make_counts, labels=make_counts.index, autopct='%1.1f%%',
        ↪startangle=140)
plt.title('Proportion of Electric Vehicles by Make')
plt.axis('equal') # Equal aspect ratio ensures that pie chart is a circle.
plt.show()
```



```
[ ]: # Group by Model Year and count the number of Electric Vehicles
yearly_counts = df.groupby('Model Year')['VIN (1-10)'].count()

plt.figure(figsize=(12, 6))
plt.plot(yearly_counts.index, yearly_counts.values, marker='o')
plt.title('Growth of Electric Vehicles Over Years')
```

```
plt.xlabel('Model Year')
plt.ylabel('Number of Electric Vehicles')
plt.grid()
plt.show()
```



```
[ ]: plt.figure(figsize=(12, 6))
sns.violinplot(data=df, x='Make', y='Electric Range', inner='quartile')
plt.xticks(rotation=45)
plt.title('Violin Plot of Electric Range by Make')
plt.show()
```

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

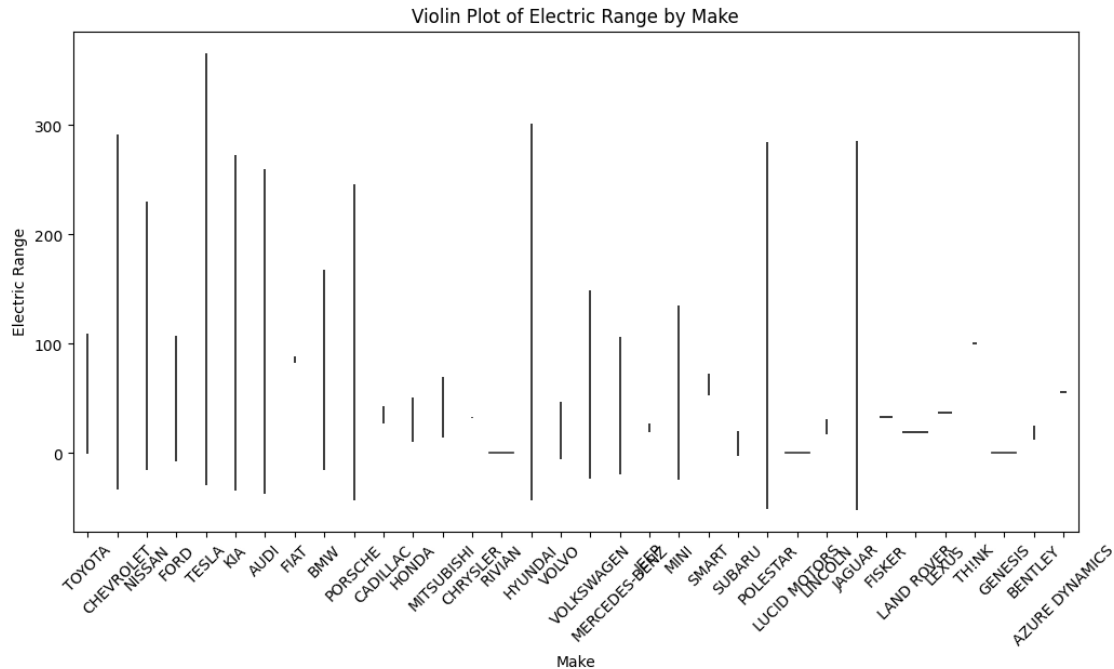
When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.

/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.



Step 5: Create a Choropleth Map

Task 2: Choropleth Visualization

Objective: Display the number of electric vehicles by location (e.g., by state) using Plotly.

Instructions:

Group the dataset by the chosen location (e.g., State) and plot a choropleth map.

```
[ ]: # Create a dataframe for EV count by state
state_ev_count = df.groupby('State').size().reset_index(name='EV Count')

# Create a Choropleth map using Plotly
fig = px.choropleth(state_ev_count,
                    locations='State',
                    locationmode='USA-states',
                    color='EV Count',
                    scope='usa',
                    color_continuous_scale='Blues',
                    title="Electric Vehicle Count by State")

fig.show()
```

Step 6: Create a Racing Bar

Task 3: Racing Bar Plot Visualization

Objective: Create a Racing Bar Plot animation that shows the count of different EV makes over the years.

Instructions:

Group the dataset by 'Make' and 'Model Year' and create a racing bar plot animation.

```
[ ]: # Grouping by 'Make' and 'Model Year'
make_year_count = df.groupby(['Model Year', 'Make']).size().
    ↪reset_index(name='EV Count')

# Racing Bar Plot using Plotly
fig = px.bar(make_year_count,
             x='Make',
             y='EV Count',
             color='Make',
             animation_frame='Model Year',
             title="Racing Bar Plot: EV Make Count over the Years")
fig.show()

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

# Grouping data by State
state_counts = df.groupby('State')['VIN (1-10)'].count().reset_index(name='EV_
    ↪Count')

# Creating the choropleth map
fig = px.choropleth(state_counts,
                    locations='State',
                    locationmode='USA-states',
                    color='EV Count',
                    color_continuous_scale='Viridis',
                    scope='usa',
                    labels={'EV Count': 'Number of Electric Vehicles'},
                    title='Number of Electric Vehicles by State')
fig.show()

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

# Grouping data by Make and Model Year
racing_data = df.groupby(['Make', 'Model Year'])['VIN (1-10)'].count().
    ↪reset_index(name='Count')

# Create a racing bar plot
fig = px.bar(racing_data,
             x='Count',
             y='Make',
             color='Make',
             animation_frame='Model Year',
```



```

        range_x=[0, racing_data['Count'].max() + 1000],
        title='Racing Bar Plot: Count of Electric Vehicles by Make and Year',
        labels={'Count': 'Number of Electric Vehicles', 'Make': 'Make'})

fig.update_layout(yaxis=dict(title='Make'),
                  xaxis_title='Count of Electric Vehicles',
                  showlegend=False)

fig.show()

```

```

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

# Grouping data by Make and Model Year
racing_data = df.groupby(['Make', 'Model Year'])['VIN (1-10)'].count().
    reset_index(name='Count')

# Highlight top N makes
top_n = 10
top_makes = racing_data.groupby('Make')['Count'].sum().nlargest(top_n).index.
    tolist()

# Create a racing bar plot with enhanced features
fig = px.bar(
    racing_data,
    x='Count',
    y='Make',
    color='Make',
    animation_frame='Model Year',
    range_x=[0, racing_data['Count'].max() + 5000],
    title='Racing Bar Plot: Count of Electric Vehicles by Make and Year',
    labels={'Count': 'Number of Electric Vehicles'},
    template='plotly_dark'
)

# Highlighting top makes
fig.for_each_trace(lambda t: t.update(marker_color='rgba(0, 128, 0, 0.8)' if t.
    name in top_makes else 'rgba(211, 211, 211, 0.5)'))

# Add text annotations to each bar
fig.update_traces(texttemplate='%{x}', textposition='outside')

# Update layout for better visibility
fig.update_layout(
    yaxis_title='Make',

```

```

        xaxis_title='Count of Electric Vehicles',
        title_x=0.5,
        title_font=dict(size=24),
        height=600,
    )

    # Add hover data to show additional metrics
    fig.update_traces(hovertemplate='Make: %{y}<br>Count: %{x}<br>Year: %{frame}')
    fig.show()

```

```

[ ]: import plotly.express as px

fig = px.scatter_3d(df, x='Base MSRP', y='Electric Range', z='Model Year',
                    color='Make',
                    title='3D Scatter Plot of Electric Vehicles',
                    labels={'Base MSRP': 'Base MSRP', 'Electric Range': 'Electric Range', 'Model Year': 'Model Year'})
fig.show()

```

```

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

# Sample DataFrame creation
# df = pd.read_csv('your_electric_vehicle_data.csv') # Load your dataset

# Grouping data by Make and Model Year
racing_data = df.groupby(['Make', 'Model Year'])['VIN (1-10)'].count().
    .reset_index(name='Count')

# Highlight top N makes
top_n = 10
top_makes = racing_data.groupby('Make')['Count'].sum().nlargest(top_n).index.
    .tolist()

# Create a racing bar plot with enhanced features
fig = px.bar(
    racing_data,
    x='Count',
    y='Make',
    color='Make',
    animation_frame='Model Year',
    range_x=[0, racing_data['Count'].max() + 5000],
    title='Racing Bar Plot: Count of Electric Vehicles by Make and Year',
    labels={'Count': 'Number of Electric Vehicles'},
    template='plotly_dark',
    height=800, # Increase height for better visibility
    width=1200 # Increase width for better visibility
)

```

```

)

# Highlighting top makes
fig.for_each_trace(lambda t: t.update(marker_color='rgba(0, 128, 0, 0.8)' if t.
    ↪name in top_makes else 'rgba(211, 211, 211, 0.5)'))

# Add text annotations to each bar
fig.update_traces(texttemplate='%{x}', textposition='outside')

# Update layout for better visibility
fig.update_layout(
    yaxis_title='Make',
    xaxis_title='Count of Electric Vehicles',
    title_x=0.5,
    title_font=dict(size=24),
    height=800, # Adjusted height
    width=1200, # Adjusted width
    font=dict(size=14), # Increase font size for better readability
    plot_bgcolor='rgba(0, 0, 0, 0.7)', # Darker background
    paper_bgcolor='rgba(0, 0, 0, 0.9)' # Darker paper background
)

# Add hover data to show additional metrics
fig.update_traces(hovertemplate='Make: %{y}<br>Count: %{x}<br>Year: %{frame}')

# Show the plot
fig.show()

```