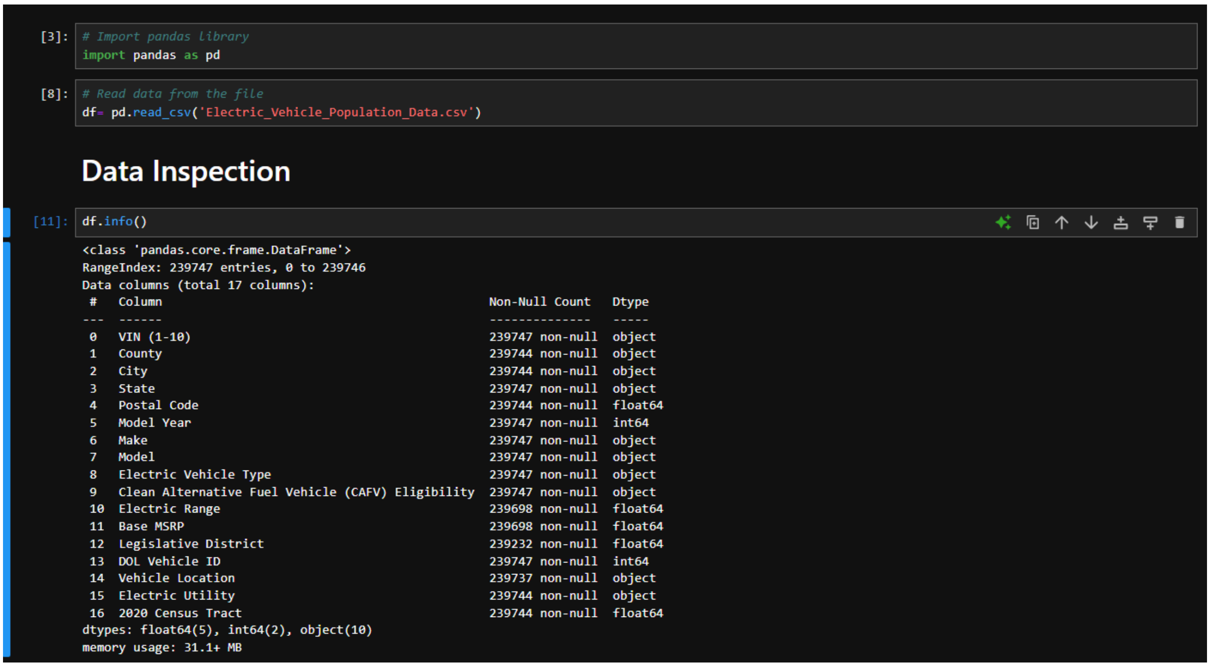
Electric Vehicle Population Data

- Import Library and Data Inspection



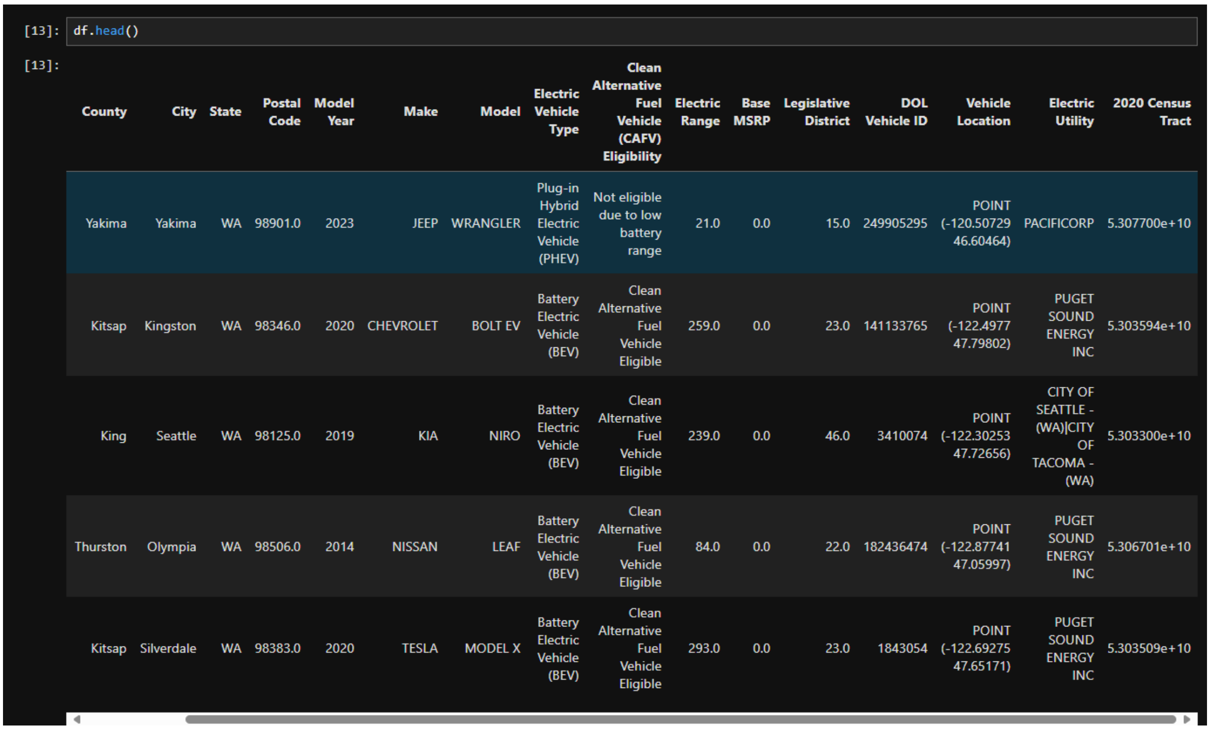
```
[3]: # Import pandas library
     import pandas as pd
```

```
[8]: # Read data from the file
     df= pd.read_csv('Electric_Vehicle_Population_Data.csv')
```

Data Inspection

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

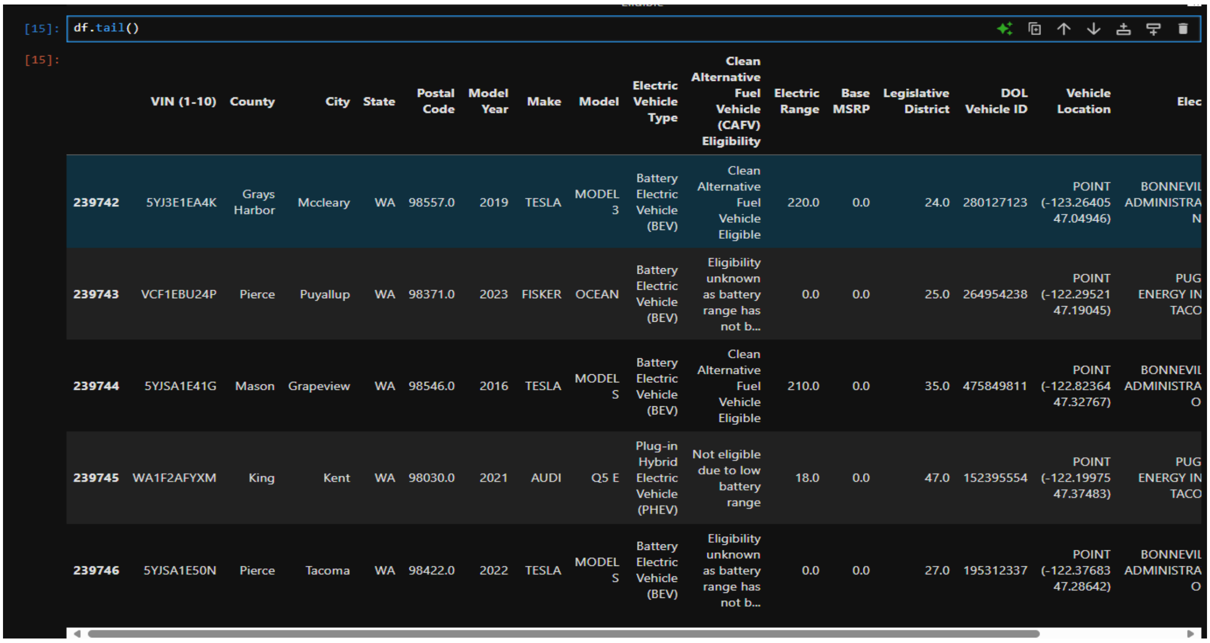
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239747 entries, 0 to 239746
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         239747 non-null  object
 1   County                                             239744 non-null  object
 2   City                                               239744 non-null  object
 3   State                                              239747 non-null  object
 4   Postal Code                                        239744 non-null  float64
 5   Model Year                                         239747 non-null  int64
 6   Make                                               239747 non-null  object
 7   Model                                              239747 non-null  object
 8   Electric Vehicle Type                              239747 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  239747 non-null  object
 10  Electric Range                                     239698 non-null  float64
 11  Base MSRP                                          239698 non-null  float64
 12  Legislative District                               239232 non-null  float64
 13  DOL Vehicle ID                                     239747 non-null  int64
 14  Vehicle Location                                   239737 non-null  object
 15  Electric Utility                                   239744 non-null  object
 16  2020 Census Tract                                  239744 non-null  float64
dtypes: float64(5), int64(2), object(10)
memory usage: 31.1+ MB
```



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

[13]:

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
Yakima	Yakima	WA	98901.0	2023	JEEP	WRANGLER	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	21.0	0.0	15.0	249905295	POINT (-120.50729 46.60464)	PACIFICORP	5.307700e+10
Kitsap	Kingston	WA	98346.0	2020	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	259.0	0.0	23.0	141133765	POINT (-122.4977 47.79802)	PUGET SOUND ENERGY INC	5.303594e+10
King	Seattle	WA	98125.0	2019	KIA	NIRO	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	239.0	0.0	46.0	3410074	POINT (-122.30253 47.72656)	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	5.303300e+10
Thurston	Olympia	WA	98506.0	2014	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	84.0	0.0	22.0	182436474	POINT (-122.87741 47.05997)	PUGET SOUND ENERGY INC	5.306701e+10
Kitsap	Silverdale	WA	98383.0	2020	TESLA	MODEL X	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	293.0	0.0	23.0	1843054	POINT (-122.69275 47.65171)	PUGET SOUND ENERGY INC	5.303509e+10

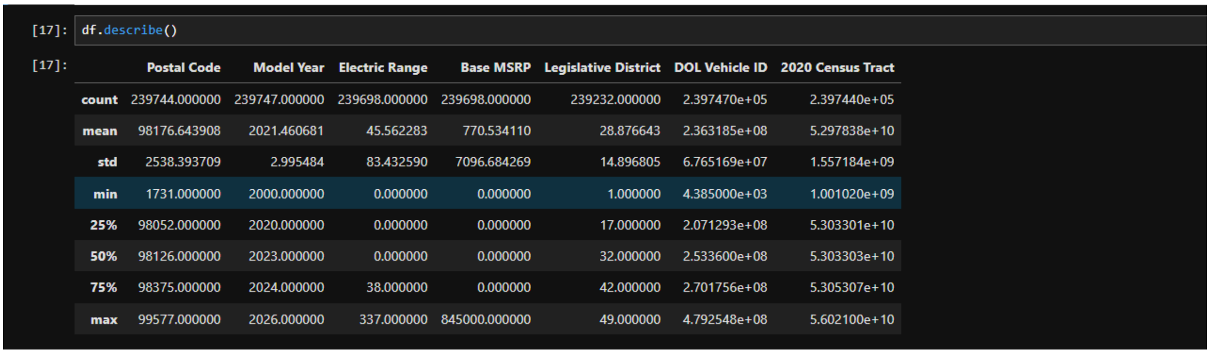


```
[15]: df.tail()
```

[15]:

	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	Elec
239742	5YJ3E1EA4K	Grays Harbor	Mccleary	WA	98557.0	2019	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	220.0	0.0	24.0	280127123	POINT (-123.26405 47.04946)	BONNEVIL ADMINISTRA N
239743	VCF1EBU24P	Pierce	Puyallup	WA	98371.0	2023	FISKER	OCEAN	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0.0	0.0	25.0	264954238	POINT (-122.29521 47.19045)	PUG ENERGY IN TACO
239744	5YJSA1E41G	Mason	Grapeview	WA	98546.0	2016	TESLA	MODEL S	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	210.0	0.0	35.0	475849811	POINT (-122.82364 47.32767)	BONNEVIL ADMINISTRA O
239745	WA1F2AFYXM	King	Kent	WA	98030.0	2021	AUDI	Q5 E	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	18.0	0.0	47.0	152395554	POINT (-122.19975 47.37483)	PUG ENERGY IN TACO
239746	5YJSA1E50N	Pierce	Tacoma	WA	98422.0	2022	TESLA	MODEL S	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0.0	0.0	27.0	195312337	POINT (-122.37683 47.28642)	BONNEVIL ADMINISTRA O

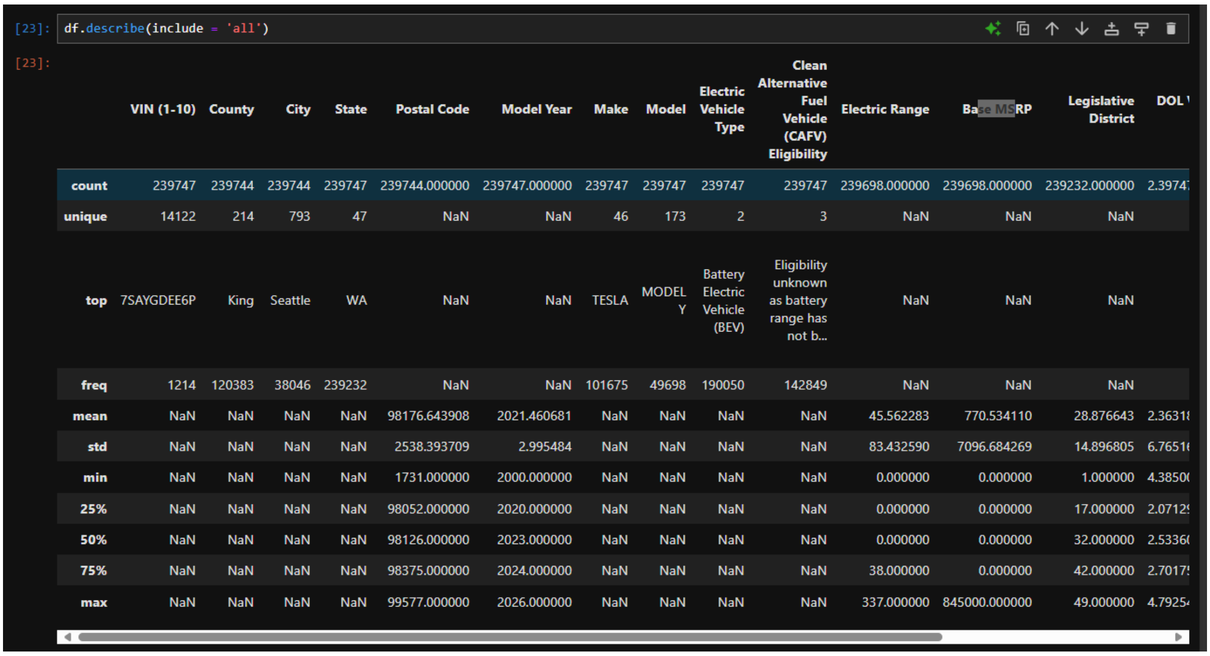
- Statistical Summary



```
[17]: df.describe()
```

[17]:

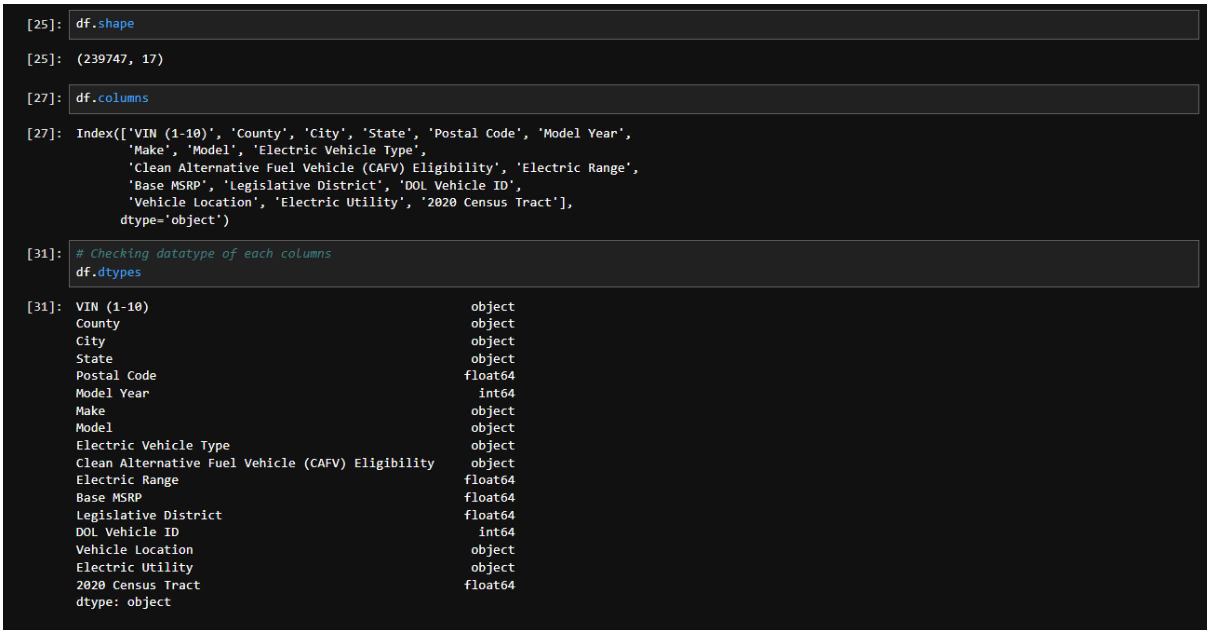
	Postal Code	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
count	239744.000000	239747.000000	239698.000000	239698.000000	239232.000000	2.397470e+05	2.397440e+05
mean	98176.643908	2021.460681	45.562283	770.534110	28.876643	2.363185e+08	5.297838e+10
std	2538.393709	2.995484	83.432590	7096.684269	14.896805	6.765169e+07	1.557184e+09
min	1731.000000	2000.000000	0.000000	0.000000	1.000000	4.385000e+03	1.001020e+09
25%	98052.000000	2020.000000	0.000000	0.000000	17.000000	2.071293e+08	5.303301e+10
50%	98126.000000	2023.000000	0.000000	0.000000	32.000000	2.533600e+08	5.303303e+10
75%	98375.000000	2024.000000	38.000000	0.000000	42.000000	2.701756e+08	5.305307e+10
max	99577.000000	2026.000000	337.000000	845000.000000	49.000000	4.792548e+08	5.602100e+10



```
[23]: df.describe(include = 'all')
```

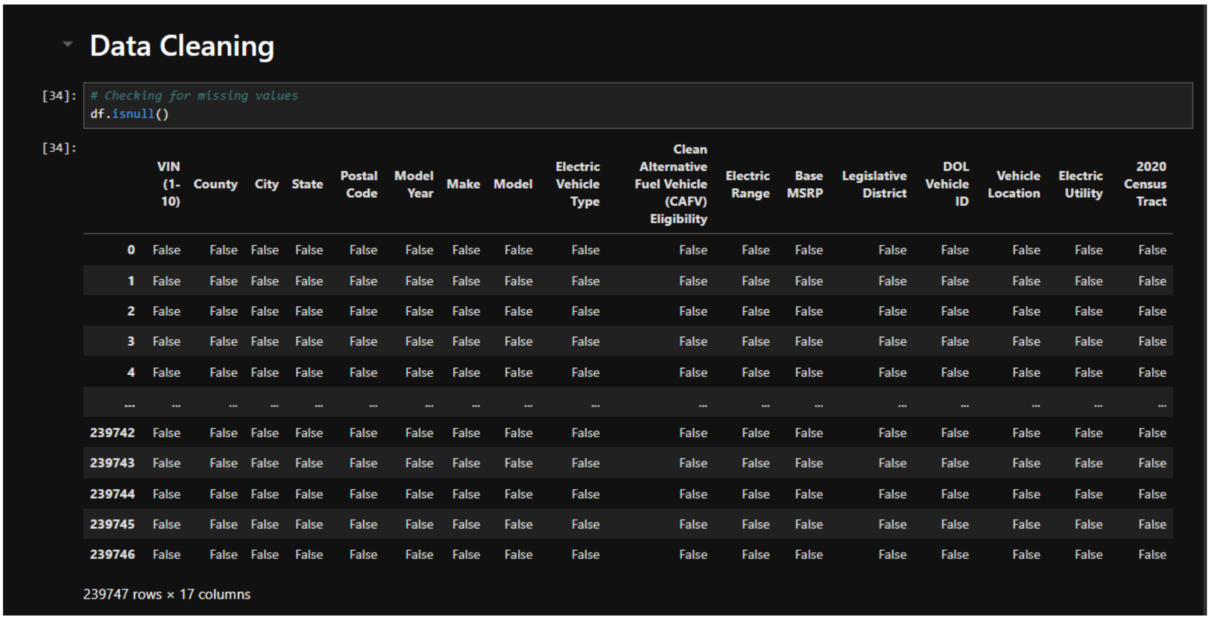
[23]:

	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 '
count	239747	239744	239744	239747	239744.000000	239747.000000	239747	239747	239747	239747	239698.000000	239698.000000	239232.000000	2.39747
unique	14122	214	793	47	NaN	NaN	46	173	2	3	NaN	NaN	NaN	
top	7SAYGDEE6P	King	Seattle	WA	NaN	NaN	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	NaN	NaN	NaN	
freq	1214	120383	38046	239232	NaN	NaN	101675	49698	190050	142849	NaN	NaN	NaN	
mean	NaN	NaN	NaN	NaN	98176.643908	2021.460681	NaN	NaN	NaN	NaN	45.562283	770.534110	28.876643	2.36318
std	NaN	NaN	NaN	NaN	2538.393709	2.995484	NaN	NaN	NaN	NaN	83.432590	7096.684269	14.896805	6.76516
min	NaN	NaN	NaN	NaN	1731.000000	2000.000000	NaN	NaN	NaN	NaN	0.000000	0.000000	1.000000	4.38500
25%	NaN	NaN	NaN	NaN	98052.000000	2020.000000	NaN	NaN	NaN	NaN	0.000000	0.000000	17.000000	2.07129
50%	NaN	NaN	NaN	NaN	98126.000000	2023.000000	NaN	NaN	NaN	NaN	0.000000	0.000000	32.000000	2.53360
75%	NaN	NaN	NaN	NaN	98375.000000	2024.000000	NaN	NaN	NaN	NaN	38.000000	0.000000	42.000000	2.70175
max	NaN	NaN	NaN	NaN	99577.000000	2026.000000	NaN	NaN	NaN	NaN	337.000000	845000.000000	49.000000	4.7925



```
[25]: df.shape

[25]: (239747, 17)

[27]: df.columns

[27]: 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')

[31]: # Checking datatype of each columns
      df.dtypes

[31]: VIN (1-10)                                            object
      County                                                object
      City                                                  object
      State                                                 object
      Postal Code                                          float64
      Model Year                                             int64
      Make                                                  object
      Model                                                 object
      Electric Vehicle Type                                 object
      Clean Alternative Fuel Vehicle (CAFV) Eligibility     object
      Electric Range                                       float64
      Base MSRP                                            float64
      Legislative District                                 float64
      DOL Vehicle ID                                         int64
      Vehicle Location                                      object
      Electric Utility                                      object
      2020 Census Tract                                    float64
      dtype: object
```

- Data Cleaning: Checking for null values and removing duplicates



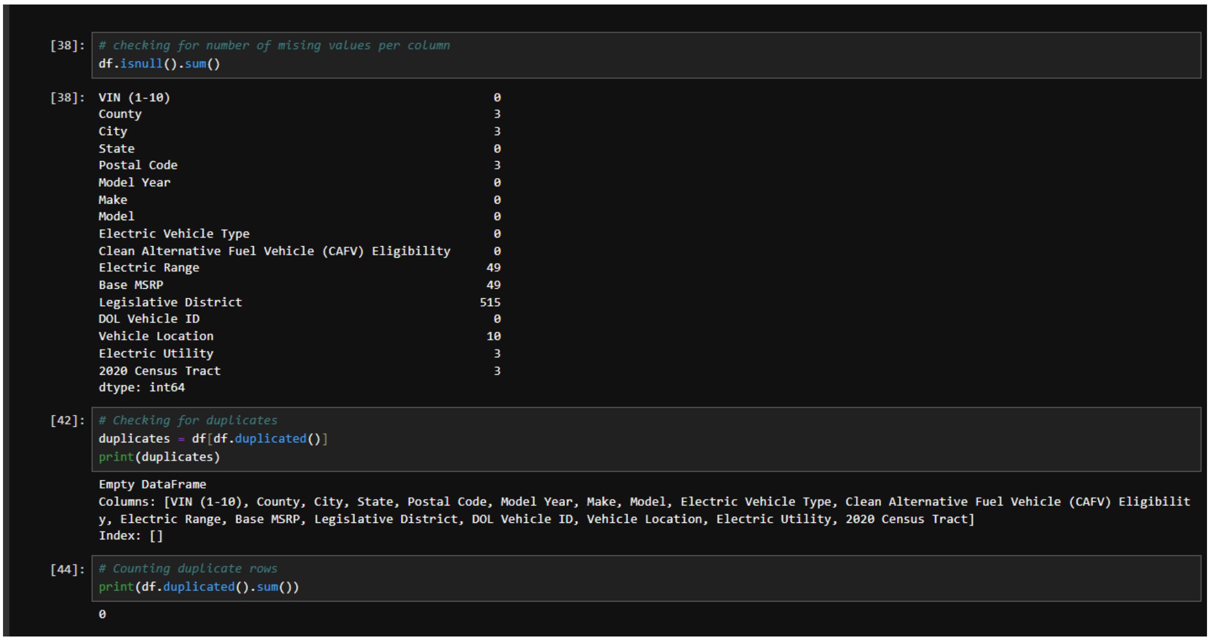
Data Cleaning

```
[34]: # Checking for missing values
      df.isnull()
```

[34]:

	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
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
239742	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
239743	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
239744	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
239745	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
239746	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

239747 rows × 17 columns



```
[38]: # checking for number of mising values per column
      df.isnull().sum()
```

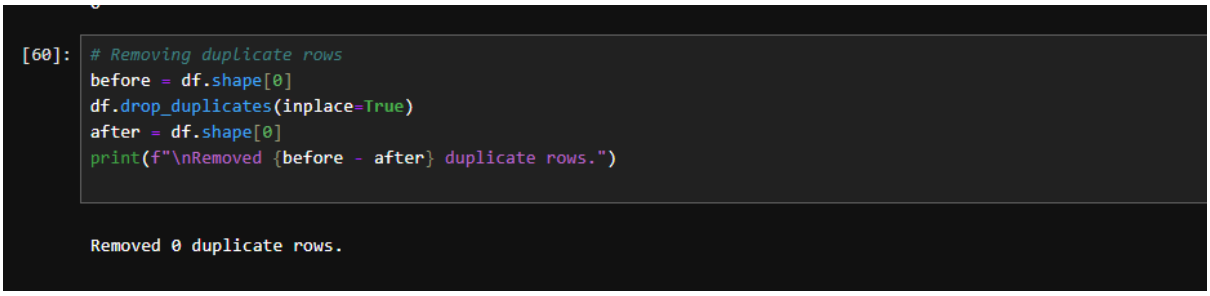
```
[38]: VIN (1-10)                                            0
      County                                                3
      City                                                  3
      State                                                 0
      Postal Code                                           3
      Model Year                                            0
      Make                                                  0
      Model                                                 0
      Electric Vehicle Type                                 0
      Clean Alternative Fuel Vehicle (CAFV) Eligibility     0
      Electric Range                                       49
      Base MSRP                                            49
      Legislative District                                515
      DOL Vehicle ID                                        0
      Vehicle Location                                     10
      Electric Utility                                      3
      2020 Census Tract                                     3
      dtype: int64
```

```
[42]: # Checking for duplicates
      duplicates = df[df.duplicated()]
      print(duplicates)
```

```
Empty DataFrame
Columns: [VIN (1-10), County, City, State, Postal Code, Model Year, Make, Model, Electric Vehicle Type, Clean Alternative Fuel Vehicle (CAFV) Eligibilit
y, Electric Range, Base MSRP, Legislative District, DOL Vehicle ID, Vehicle Location, Electric Utility, 2020 Census Tract]
Index: []
```

```
[44]: # Counting duplicate rows
      print(df.duplicated().sum())
```

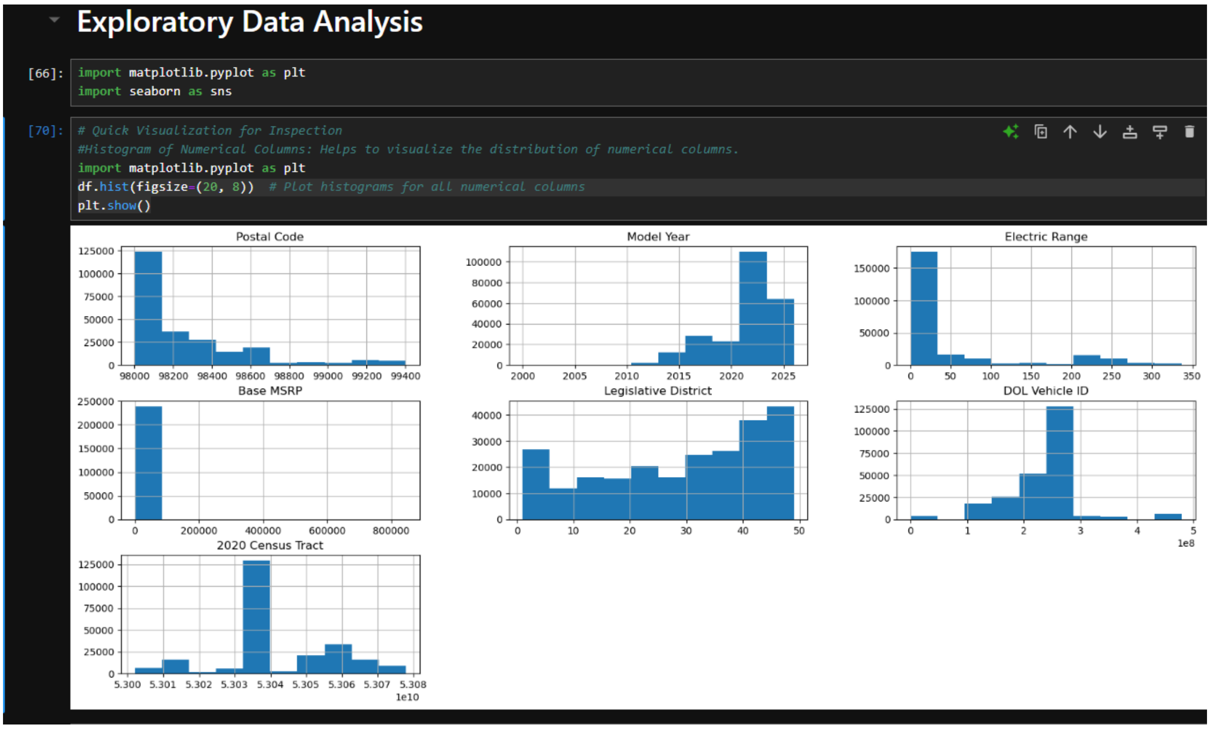
```
0
```



```
[60]: # Removing duplicate rows
      before = df.shape[0]
      df.drop_duplicates(inplace=True)
      after = df.shape[0]
      print(f"\nRemoved {before - after} duplicate rows.")
```

```
Removed 0 duplicate rows.
```

- Exploratory Data Analysis:
    - Quick Visualization for Inspection

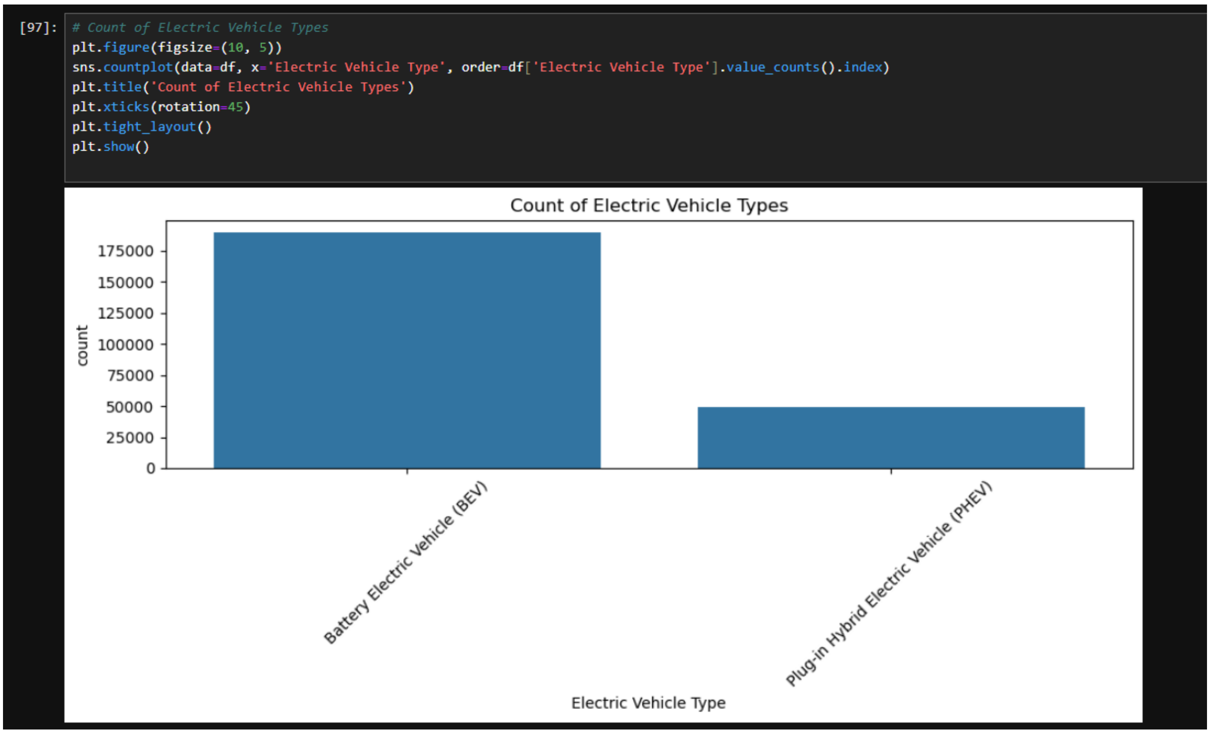


Exploratory Data Analysis

[66]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[70]:
```
# Quick Visualization for Inspection
#Histogram of Numerical Columns: Helps to visualize the distribution of numerical columns.
import matplotlib.pyplot as plt
df.hist(figsize=(20, 8))  # Plot histograms for all numerical columns
plt.show()
```

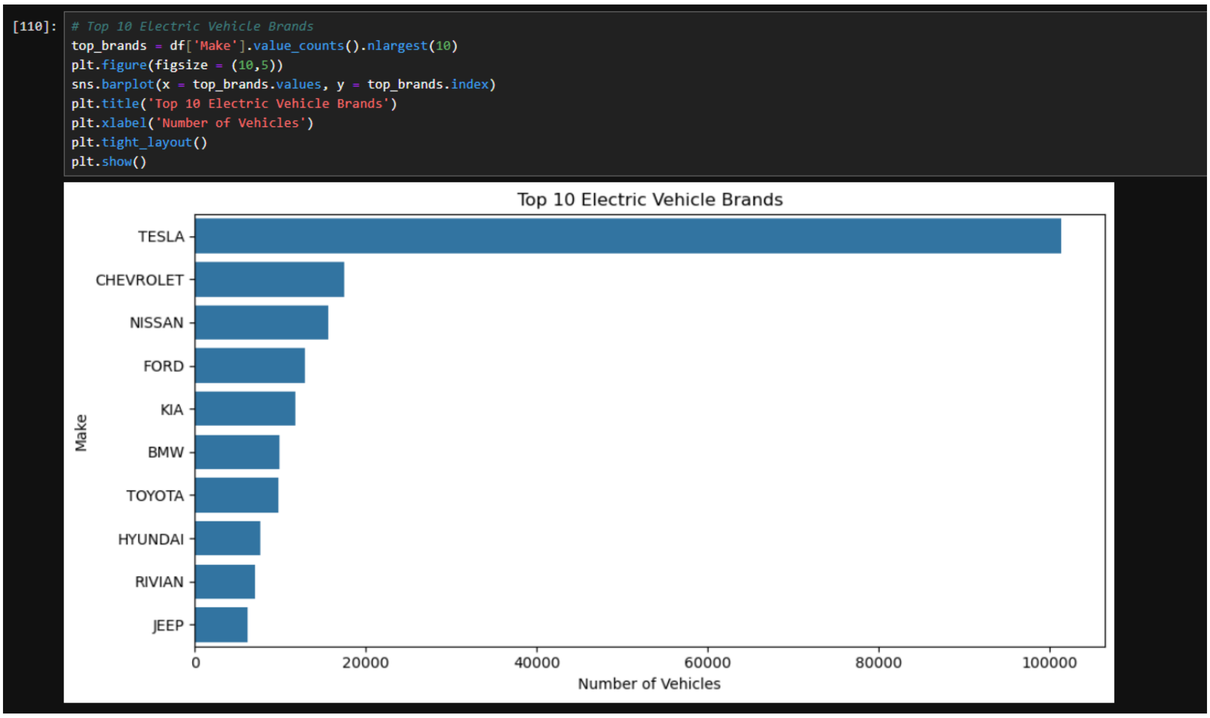
- Counting Electric Vehicle Types



[97]:
```
# Count of Electric Vehicle Types
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Electric Vehicle Type', order=df['Electric Vehicle Type'].value_counts().index)
plt.title('Count of Electric Vehicle Types')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

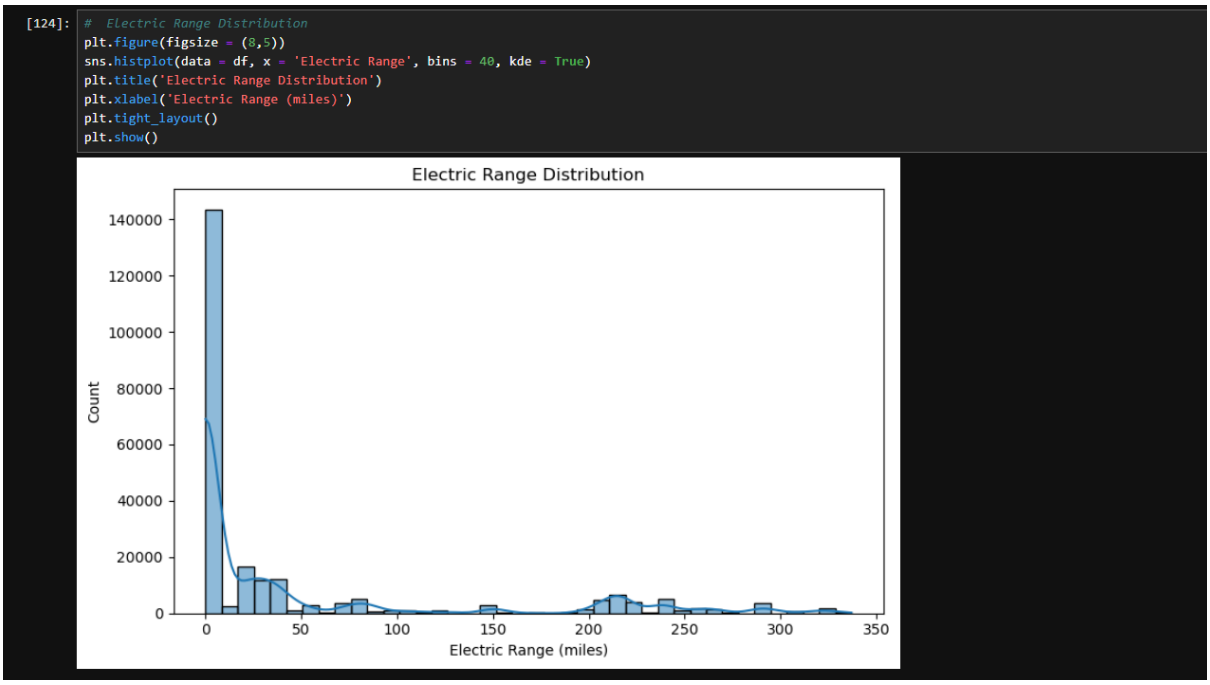
Count of Electric Vehicle Types
count
175000
150000
125000
100000
75000
50000
25000
0
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)
Electric Vehicle Type

- Top 10 Electric Vehicle Brands



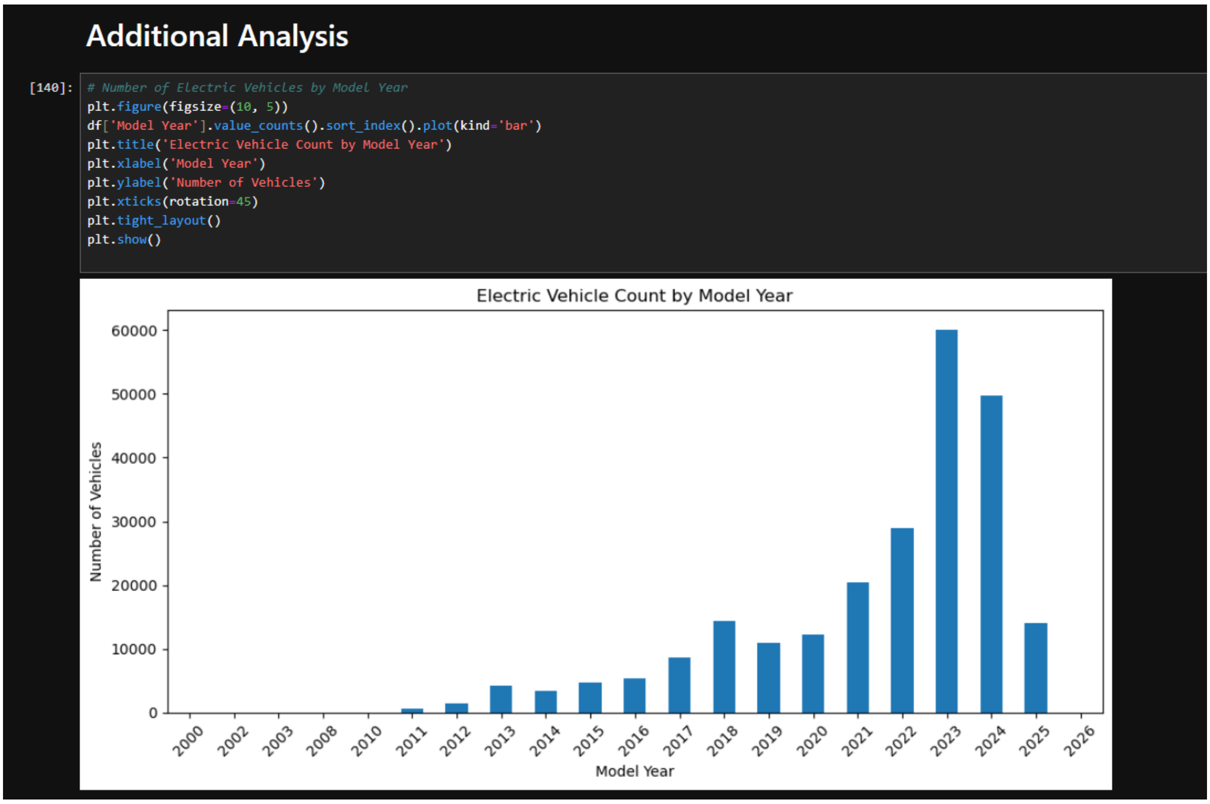
```
[110]: # Top 10 Electric Vehicle Brands
top_brands = df['Make'].value_counts().nlargest(10)
plt.figure(figsize = (10,5))
sns.barplot(x = top_brands.values, y = top_brands.index)
plt.title('Top 10 Electric Vehicle Brands')
plt.xlabel('Number of Vehicles')
plt.tight_layout()
plt.show()
```

- Electric Range Distribution



```
[124]: #  Electric Range Distribution
plt.figure(figsize = (8,5))
sns.histplot(data = df, x = 'Electric Range', bins = 40, kde = True)
plt.title('Electric Range Distribution')
plt.xlabel('Electric Range (miles)')
plt.tight_layout()
plt.show()
```

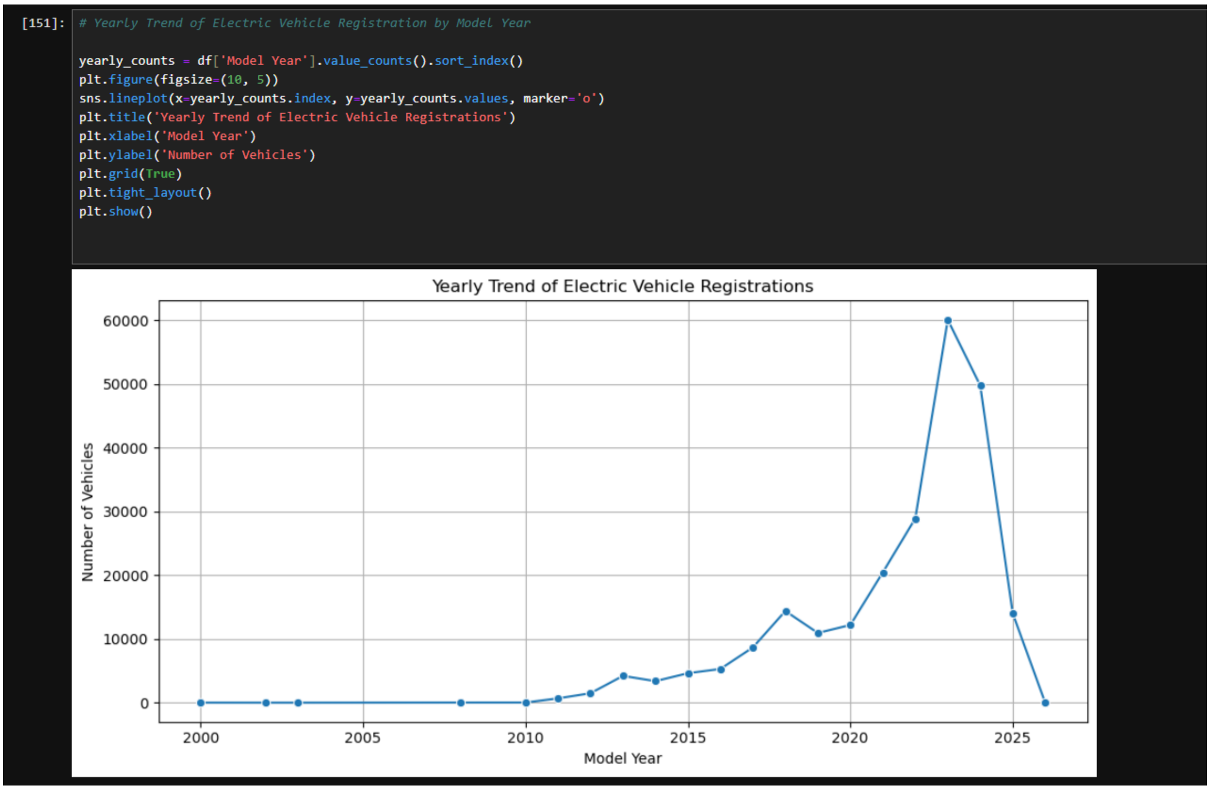
- Additional Analysis:
    - Number of Electric Vehicles by Model Year



Additional Analysis

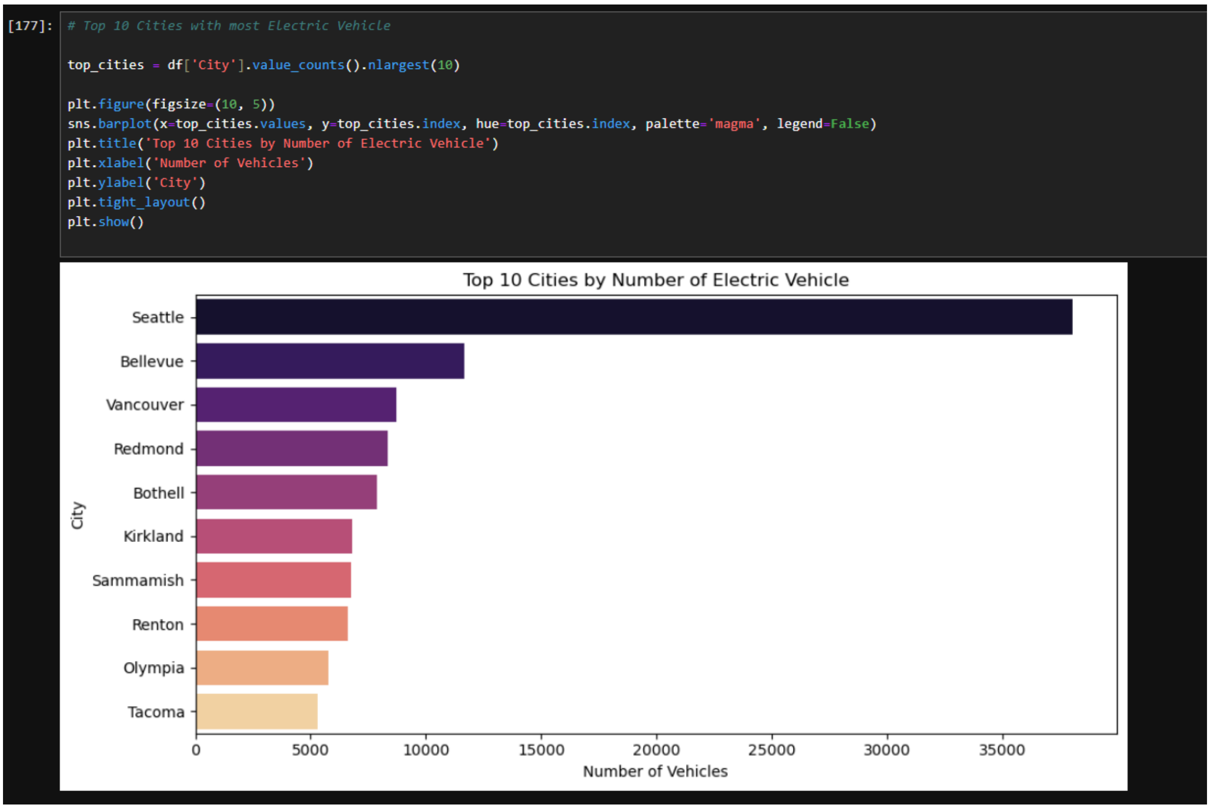
[140]:
```
# Number of Electric Vehicles by Model Year
plt.figure(figsize=(10, 5))
df['Model Year'].value_counts().sort_index().plot(kind='bar')
plt.title('Electric Vehicle Count by Model Year')
plt.xlabel('Model Year')
plt.ylabel('Number of Vehicles')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

- Yearly Trend of Electric Vehicle Registration by model year



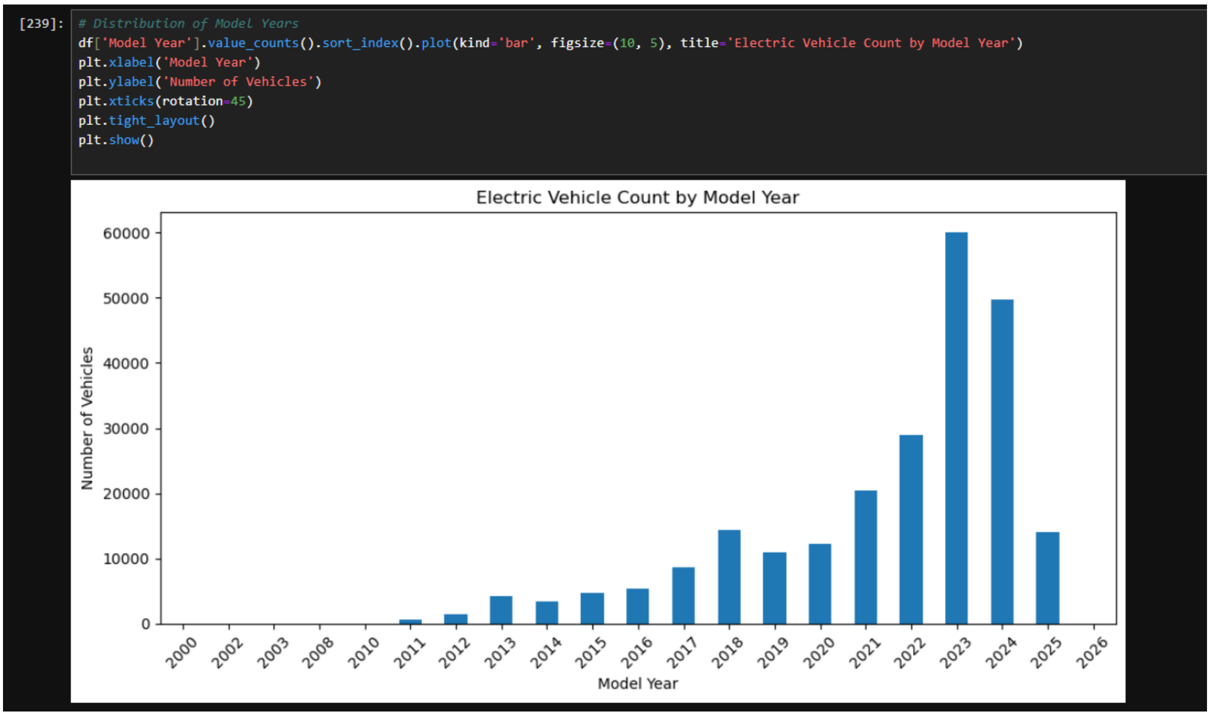
[151]:
```
# Yearly Trend of Electric Vehicle Registration by Model Year

yearly_counts = df['Model Year'].value_counts().sort_index()
plt.figure(figsize=(10, 5))
sns.lineplot(x=yearly_counts.index, y=yearly_counts.values, marker='o')
plt.title('Yearly Trend of Electric Vehicle Registrations')
plt.xlabel('Model Year')
plt.ylabel('Number of Vehicles')
plt.grid(True)
plt.tight_layout()
plt.show()
```

- Top 10 Cities with most Electric Vehicle



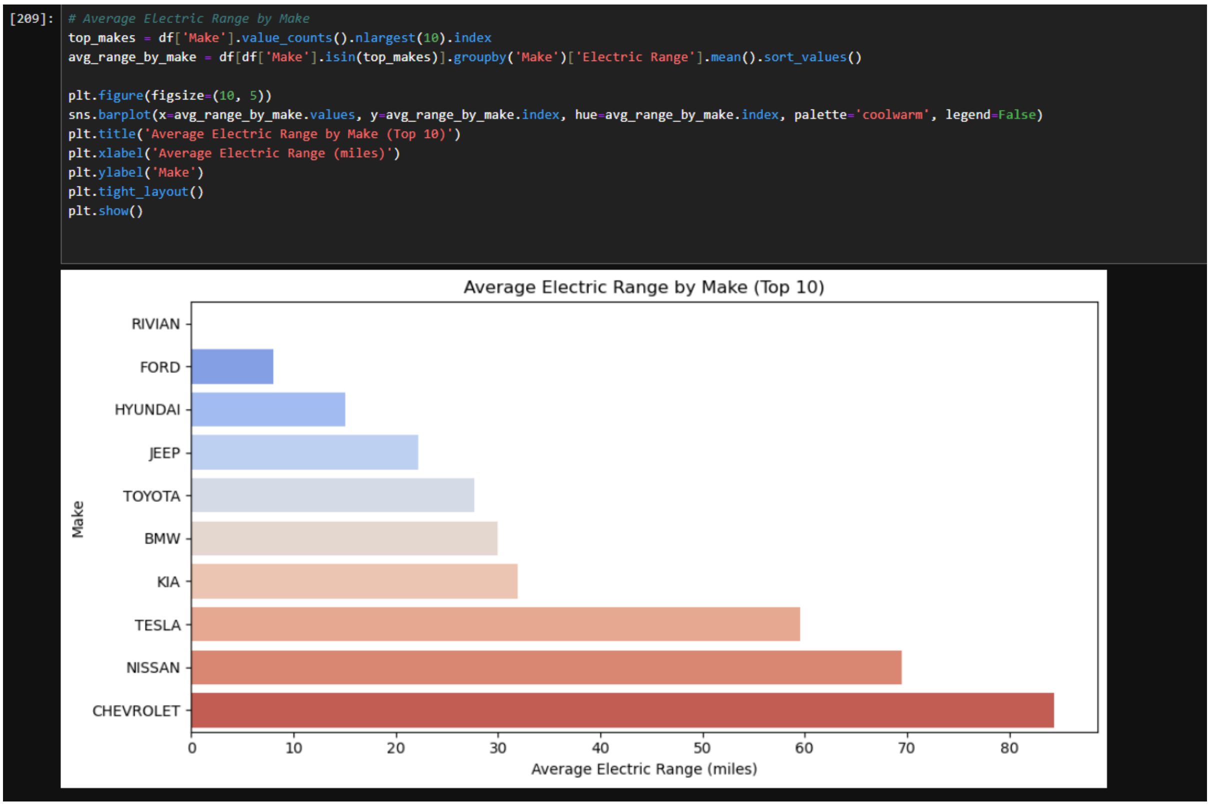
```
[177]: # Top 10 Cities with most Electric Vehicle

top_cities = df['City'].value_counts().nlargest(10)

plt.figure(figsize=(10, 5))
sns.barplot(x=top_cities.values, y=top_cities.index, hue=top_cities.index, palette='magma', legend=False)
plt.title('Top 10 Cities by Number of Electric Vehicle')
plt.xlabel('Number of Vehicles')
plt.ylabel('City')
plt.tight_layout()
plt.show()
```

- Distribution of Model Years



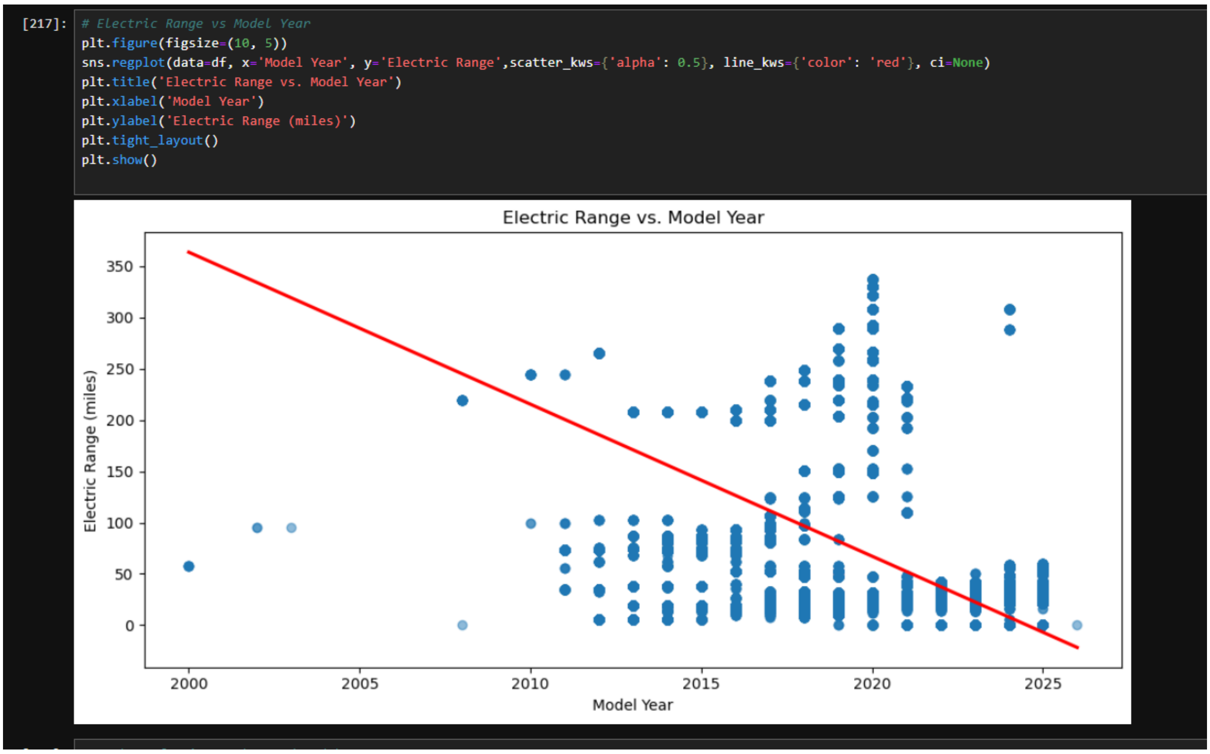
```
[239]: # Distribution of Model Years
df['Model Year'].value_counts().sort_index().plot(kind='bar', figsize=(10, 5), title='Electric Vehicle Count by Model Year')
plt.xlabel('Model Year')
plt.ylabel('Number of Vehicles')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

- Average Electric Range by Make



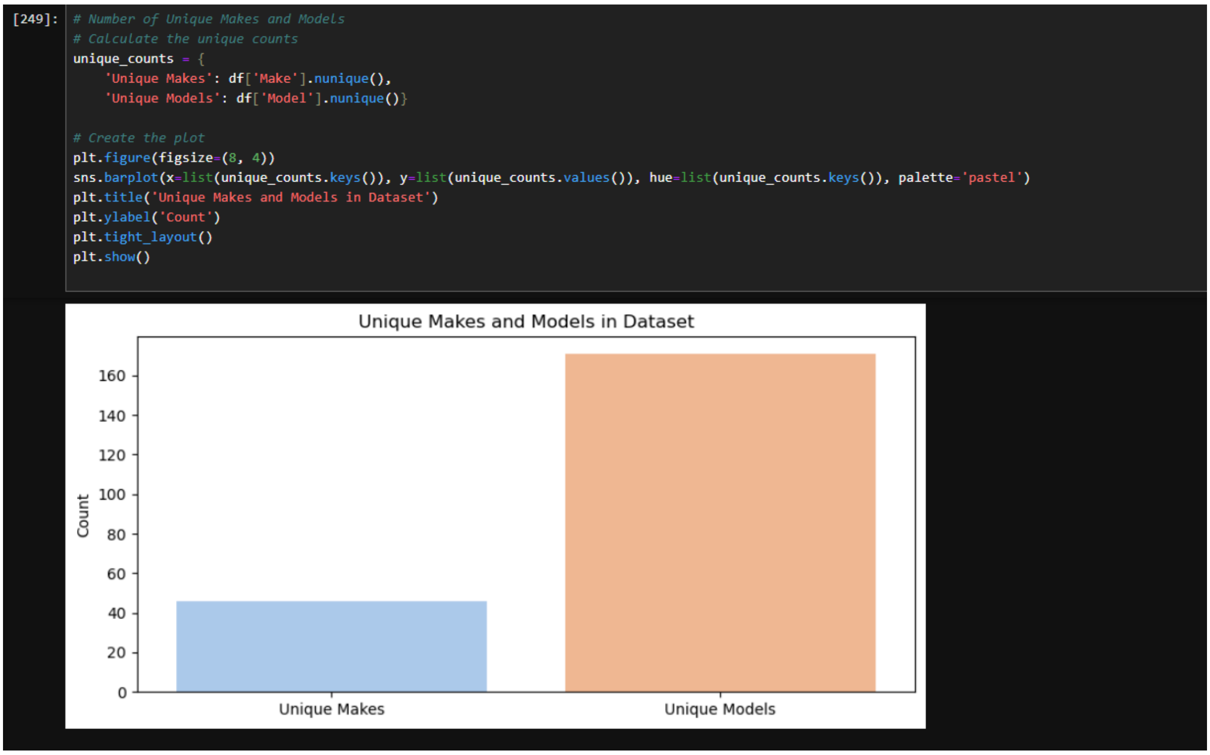
```
[209]: # Average Electric Range by Make
top_makes = df['Make'].value_counts().nlargest(10).index
avg_range_by_make = df[df['Make'].isin(top_makes)].groupby('Make')['Electric Range'].mean().sort_values()

plt.figure(figsize=(10, 5))
sns.barplot(x=avg_range_by_make.values, y=avg_range_by_make.index, hue=avg_range_by_make.index, palette='coolwarm', legend=False)
plt.title('Average Electric Range by Make (Top 10)')
plt.xlabel('Average Electric Range (miles)')
plt.ylabel('Make')
plt.tight_layout()
plt.show()
```

- Electric Range vs Model Year



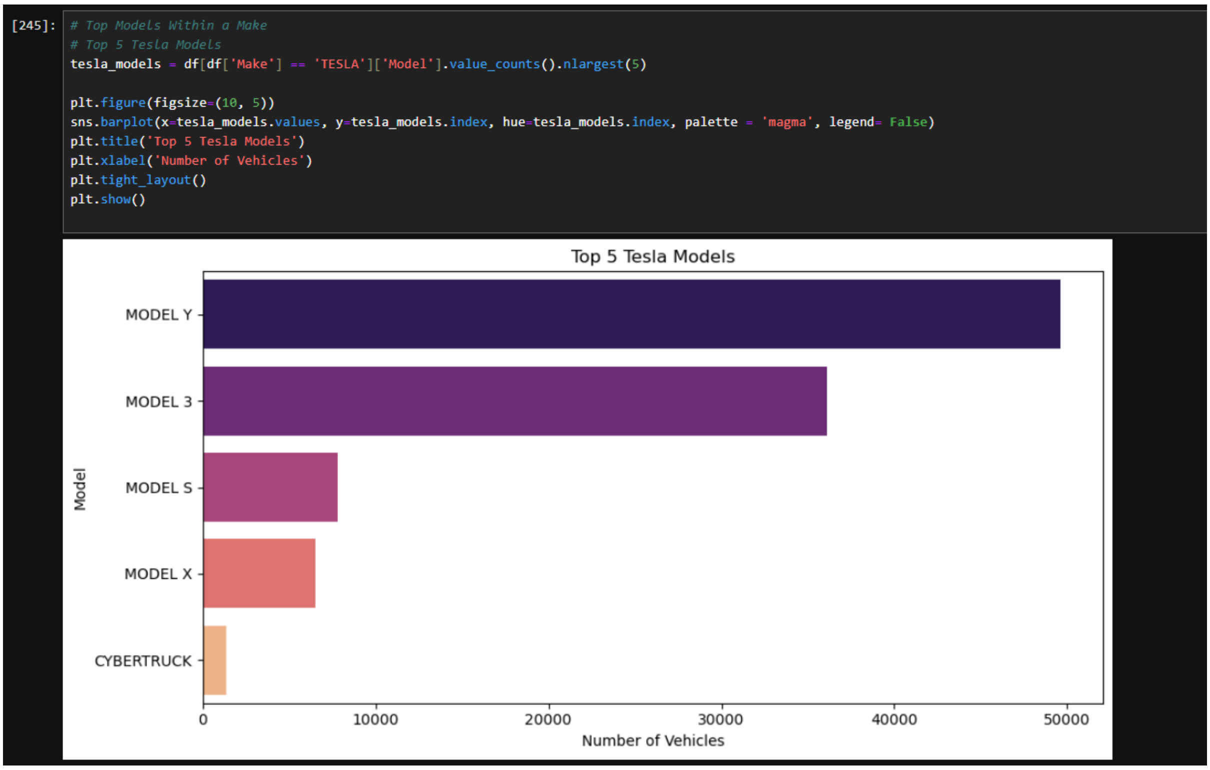
```
[217]: # Electric Range vs Model Year
plt.figure(figsize=(10, 5))
sns.regplot(data=df, x='Model Year', y='Electric Range',scatter_kws={'alpha': 0.5}, line_kws={'color': 'red'}, ci=None)
plt.title('Electric Range vs. Model Year')
plt.xlabel('Model Year')
plt.ylabel('Electric Range (miles)')
plt.tight_layout()
plt.show()
```

- Number of Unique Makes and Models



```
[249]: # Number of Unique Makes and Models
       # Calculate the unique counts
       unique_counts = {
           'Unique Makes': df['Make'].nunique(),
           'Unique Models': df['Model'].nunique()}

       # Create the plot
       plt.figure(figsize=(8, 4))
       sns.barplot(x=list(unique_counts.keys()), y=list(unique_counts.values()), hue=list(unique_counts.keys()), palette='pastel')
       plt.title('Unique Makes and Models in Dataset')
       plt.ylabel('Count')
       plt.tight_layout()
       plt.show()
```

- Top Models within a Make: Top 5 Tesla Models



```
[245]: # Top Models Within a Make
       # Top 5 Tesla Models
       tesla_models = df[df['Make'] == 'TESLA']['Model'].value_counts().nlargest(5)

       plt.figure(figsize=(10, 5))
       sns.barplot(x=tesla_models.values, y=tesla_models.index, hue=tesla_models.index, palette = 'magma', legend= False)
       plt.title('Top 5 Tesla Models')
       plt.xlabel('Number of Vehicles')
       plt.tight_layout()
       plt.show()
```