<DATA SCIENCE TOOLBOX :PYTHON PROGRAMMING>

Electric_Vehicle_Population

(Project Semester January-April 2025)

(Population of Electric Vehicle)

Submitted by

(Anand kumar)

Registration No- 12309788

Programme and Section – K23ED

Course Code - 375

Under the Guidance of

(Dr. Dhiraj Kapila)

Discipline of CSE/IT

Lovely School of computer science and engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that .... (student’s name) bearing Registration no. .... has completed .... <Course Code> project titled, **“................................”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor
Designation of the Supervisor
School of …………………………………………….
Lovely Professional University
Phagwara, Punjab.

Date:

DECLARATION

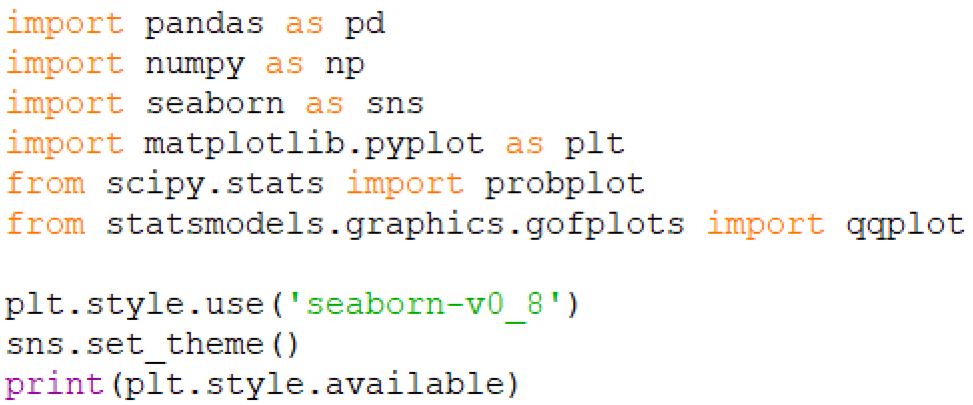
I, Anand kumar, student of Computer Science and Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 11-04-2025 Signature Anand kr.

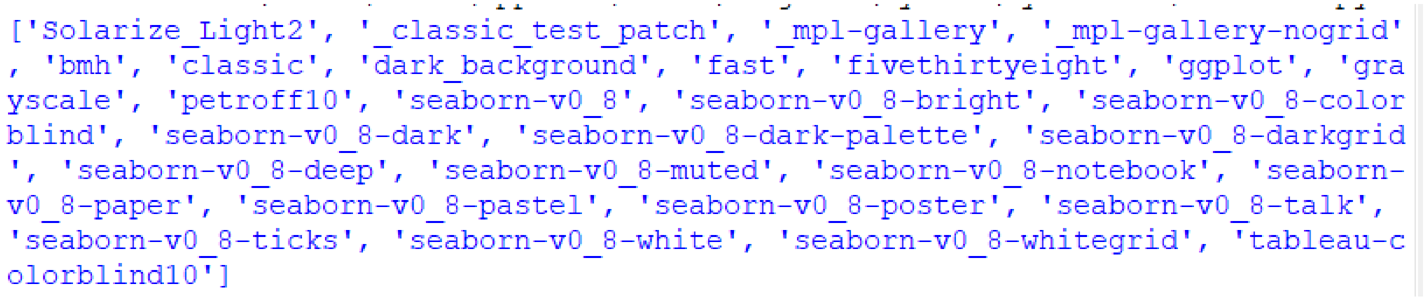
Registration No.- 12309788 Name of the student

ANAND KUMAR

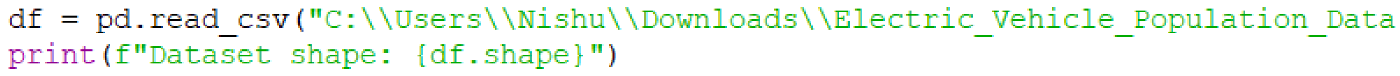
Below I attached the code and the output generated by the code



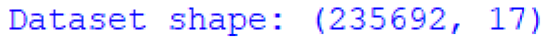
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import probplot
from statsmodels.graphics.gofplots import qqplot

plt.style.use('seaborn-v0_8')
sns.set_theme()
print(plt.style.available)
```



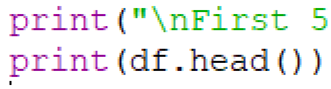
```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid'
, 'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'gra
yscale', 'petroff10', 'seaborn-v0_8', 'seaborn-v0_8-bright', 'seaborn-v0_8-color
blind', 'seaborn-v0_8-dark', 'seaborn-v0_8-dark-palette', 'seaborn-v0_8-darkgrid
', 'seaborn-v0_8-deep', 'seaborn-v0_8-muted', 'seaborn-v0_8-notebook', 'seaborn-
v0_8-paper', 'seaborn-v0_8-pastel', 'seaborn-v0_8-poster', 'seaborn-v0_8-talk',
'seaborn-v0_8-ticks', 'seaborn-v0_8-white', 'seaborn-v0_8-whitegrid', 'tableau-c
olorblind10']
```



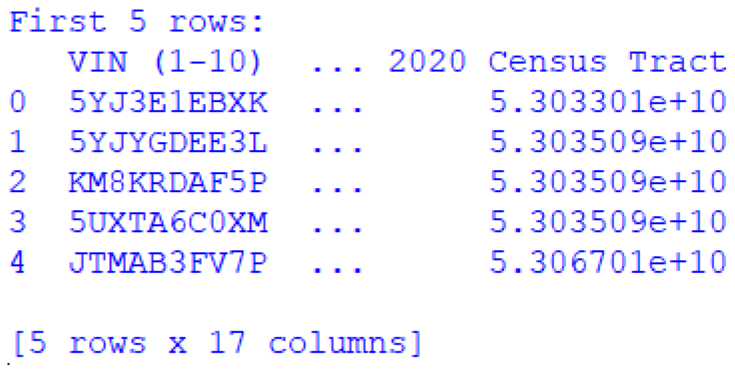
```
df = pd.read_csv("C:\\Users\\Nishu\\Downloads\\Electric_Vehicle_Population_Data
print(f"Dataset shape: {df.shape}")
```



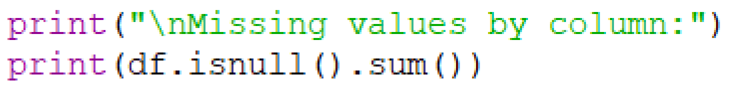
```
Dataset shape: (235692, 17)
```



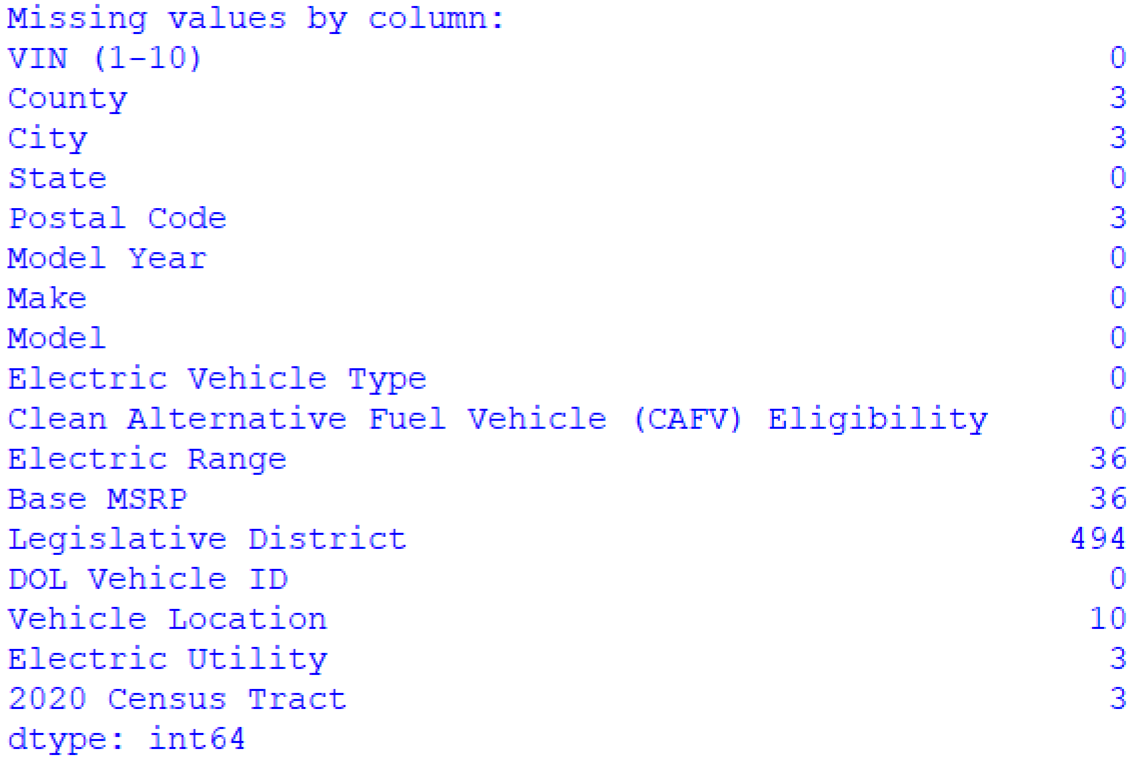
```
print("\nFirst 5 rows:")
print(df.head())
```



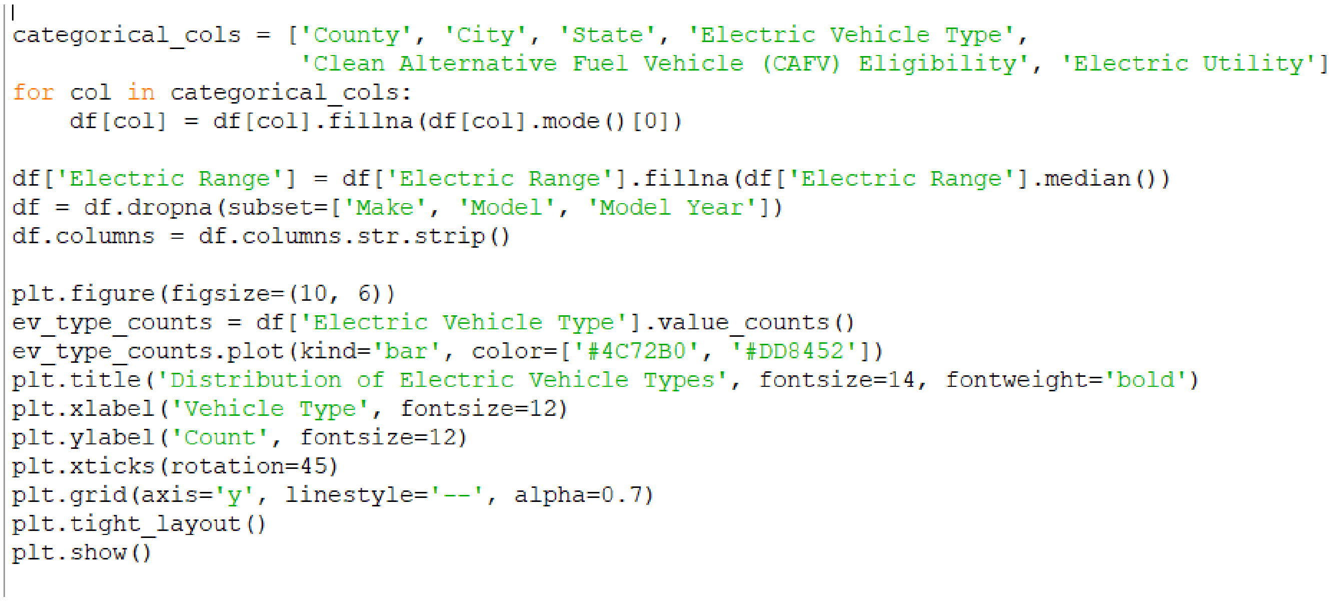
```
First 5 rows:
   VIN (1-10)  ... 2020 Census Tract
0  5YJ3E1EBXK  ...       5.303301e+10
1  5YJYGDEE3L  ...       5.303509e+10
2  KM8KRDAF5P  ...       5.303509e+10
3  5UXTA6C0XM  ...       5.303509e+10
4  JTMAB3FV7P  ...       5.306701e+10

[5 rows x 17 columns]
```



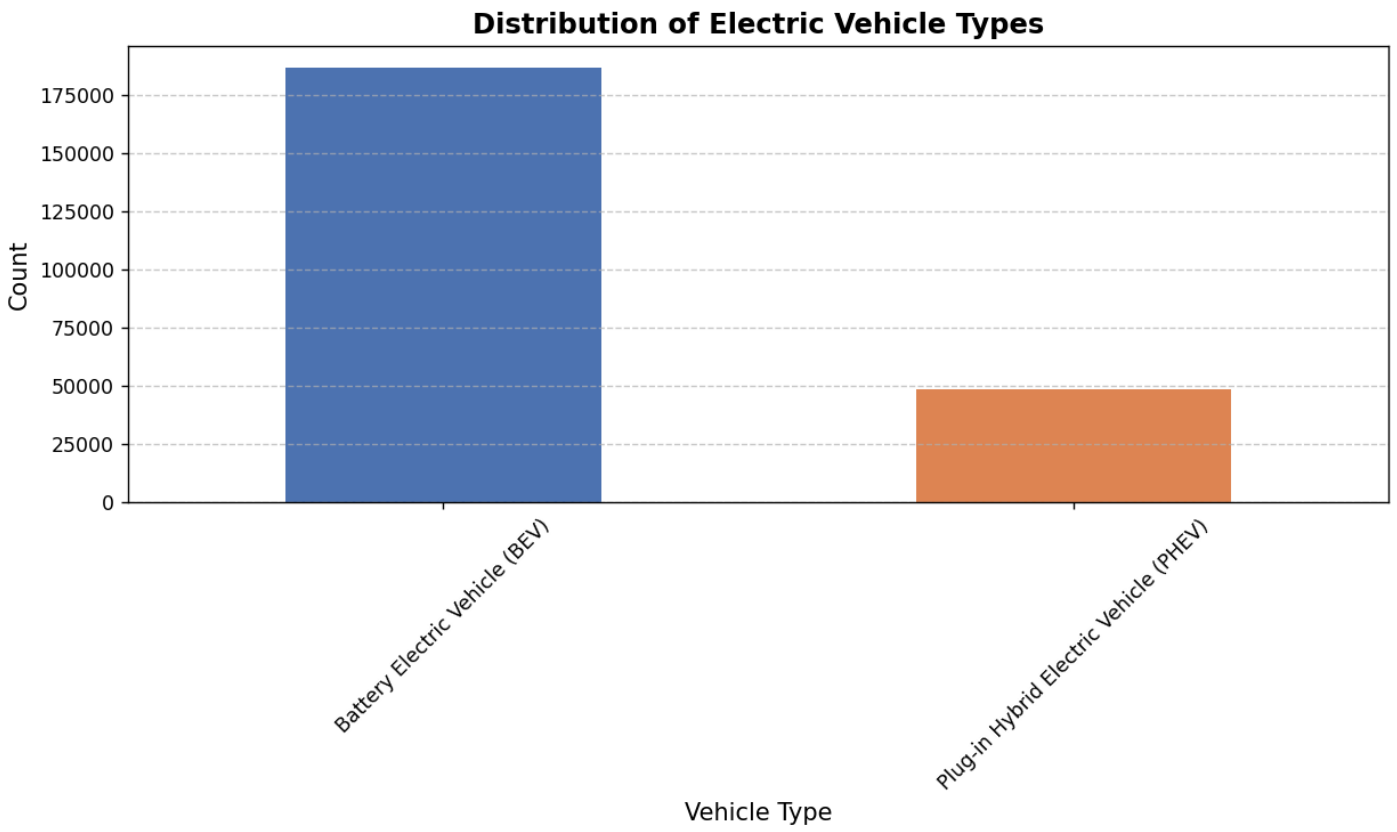
```
print("\nMissing values by column:")
print(df.isnull().sum())
```



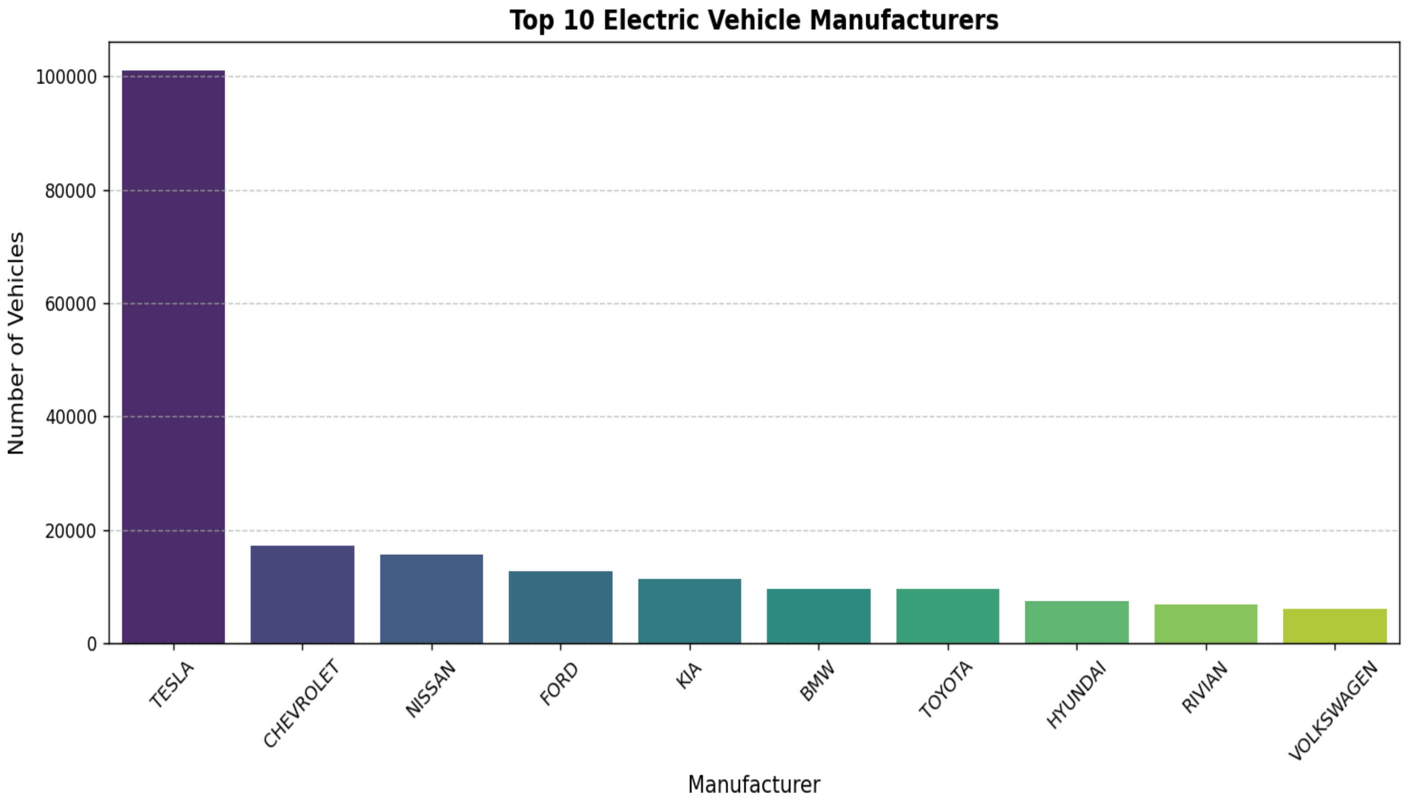
```
Missing values by column:
VIN (1-10)                                                0
County                                                    3
City                                                      3
State                                                     0
Postal Code                                               3
Model Year                                                0
Make                                                      0
Model                                                     0
Electric Vehicle Type                                     0
Clean Alternative Fuel Vehicle (CAFV) Eligibility         0
Electric Range                                           36
Base MSRP                                                36
Legislative District                                    494
DOL Vehicle ID                                            0
Vehicle Location                                         10
Electric Utility                                          3
2020 Census Tract                                         3
dtype: int64
```



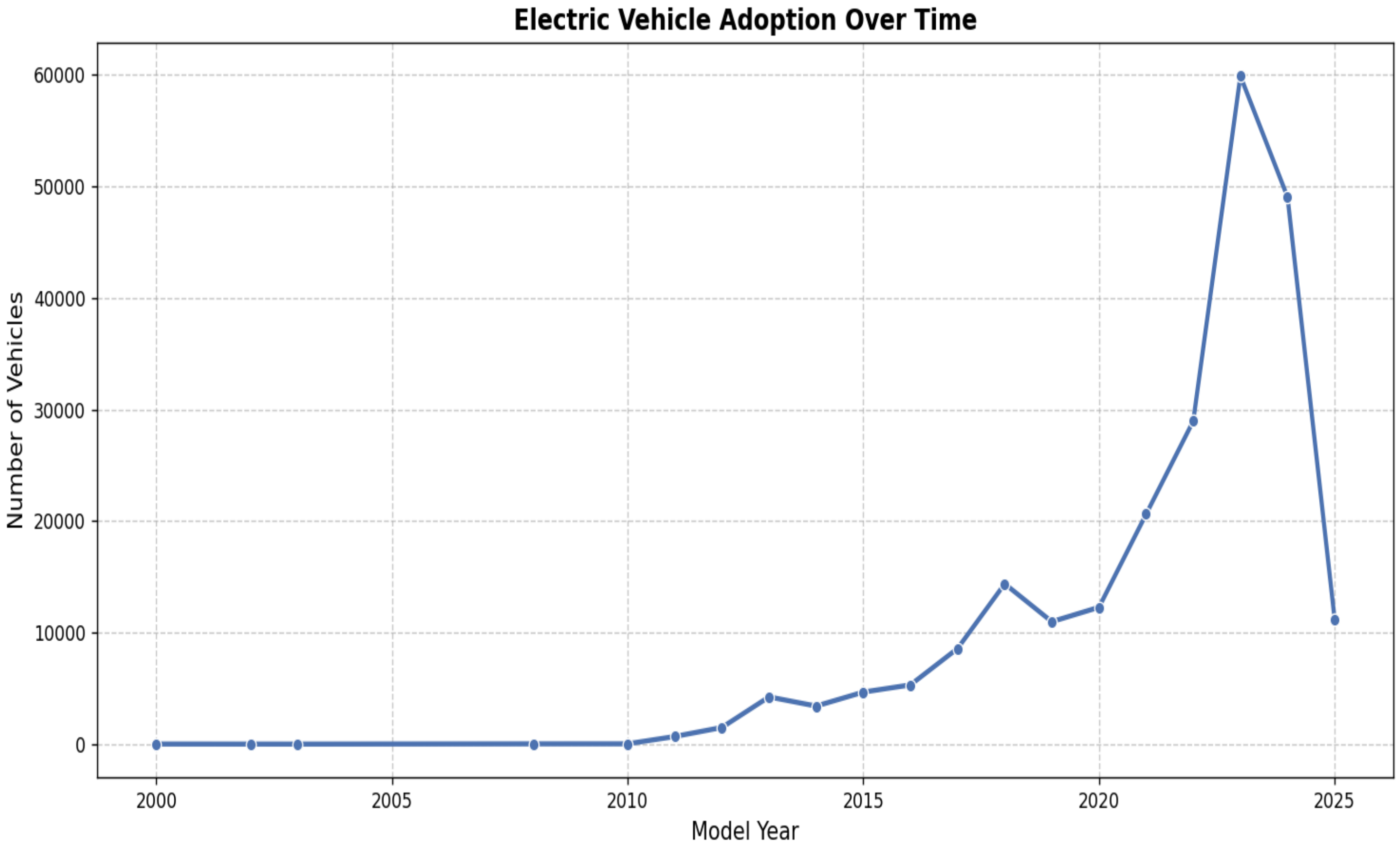
```
categorical_cols = ['County', 'City', 'State', 'Electric Vehicle Type',
                    'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Electric Utility']
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

df['Electric Range'] = df['Electric Range'].fillna(df['Electric Range'].median())
df = df.dropna(subset=['Make', 'Model', 'Model Year'])
df.columns = df.columns.str.strip()

plt.figure(figsize=(10, 6))
ev_type_counts = df['Electric Vehicle Type'].value_counts()
ev_type_counts.plot(kind='bar', color=['#4C72B0', '#DD8452'])
plt.title('Distribution of Electric Vehicle Types', fontsize=14, fontweight='bold')
plt.xlabel('Vehicle Type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

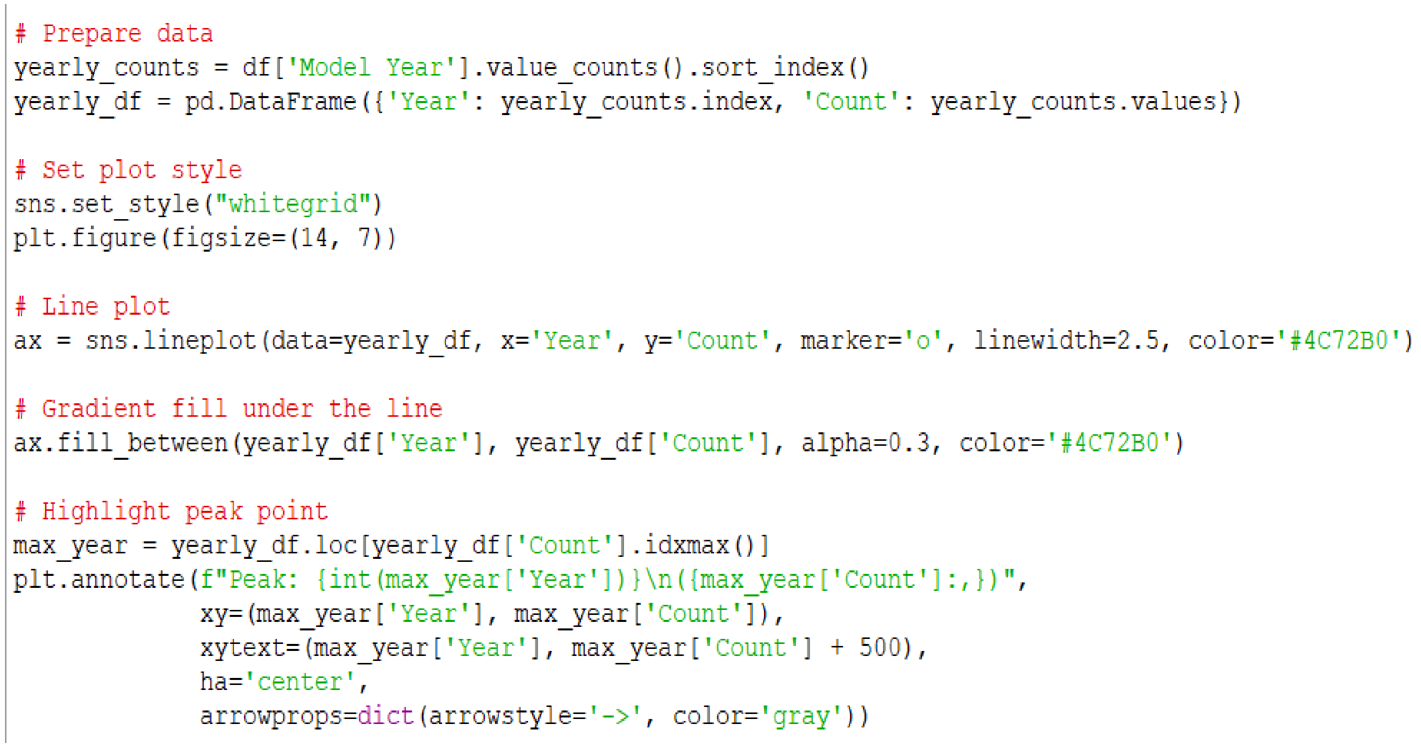


```
# Prepare data
top_makes = df['Make'].value_counts().nlargest(10)
top_makes_df = top_makes.reset_index()
top_makes_df.columns = ['Make', 'Count']

# Plot
plt.figure(figsize=(12, 6))
sns.barplot(data=top_makes_df, x='Make', y='Count', hue='Make', palette='viridis', legend=False)
plt.title('Top 10 Electric Vehicle Manufacturers', fontsize=14, fontweight='bold')
plt.xlabel('Manufacturer', fontsize=12)
plt.ylabel('Number of Vehicles', fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

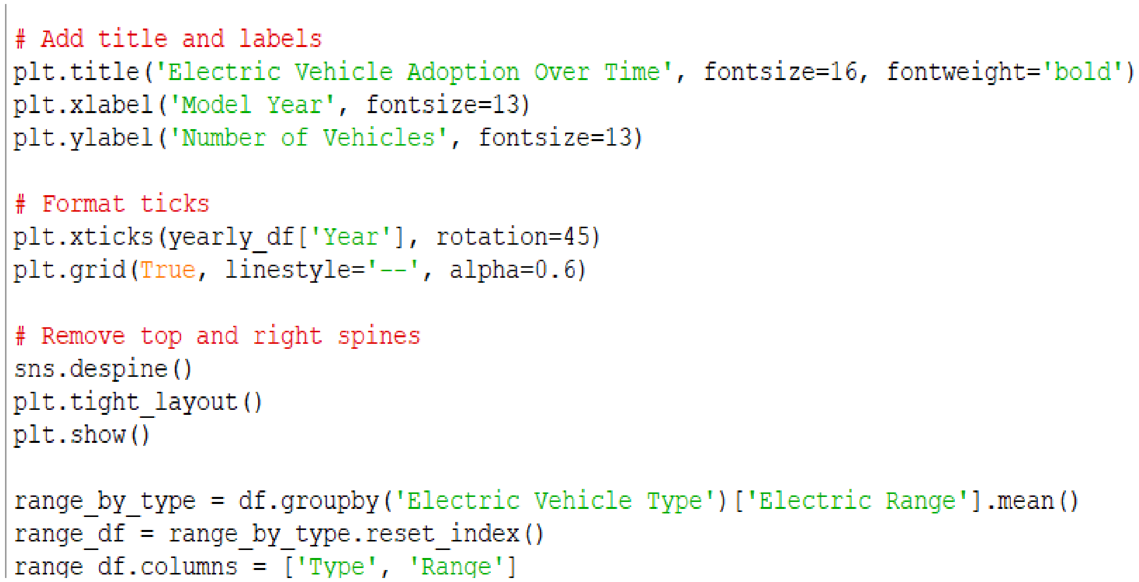


```
plt.figure(figsize=(12, 6))
yearly_counts = df['Model Year'].value_counts().sort_index()
sns.lineplot(x=yearly_counts.index, y=yearly_counts.values, marker='o',
             linewidth=2.5, color='#4C72B0')
plt.title('Electric Vehicle Adoption Over Time', fontsize=14, fontweight='bold')
plt.xlabel('Model Year', fontsize=12)
plt.ylabel('Number of Vehicles', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

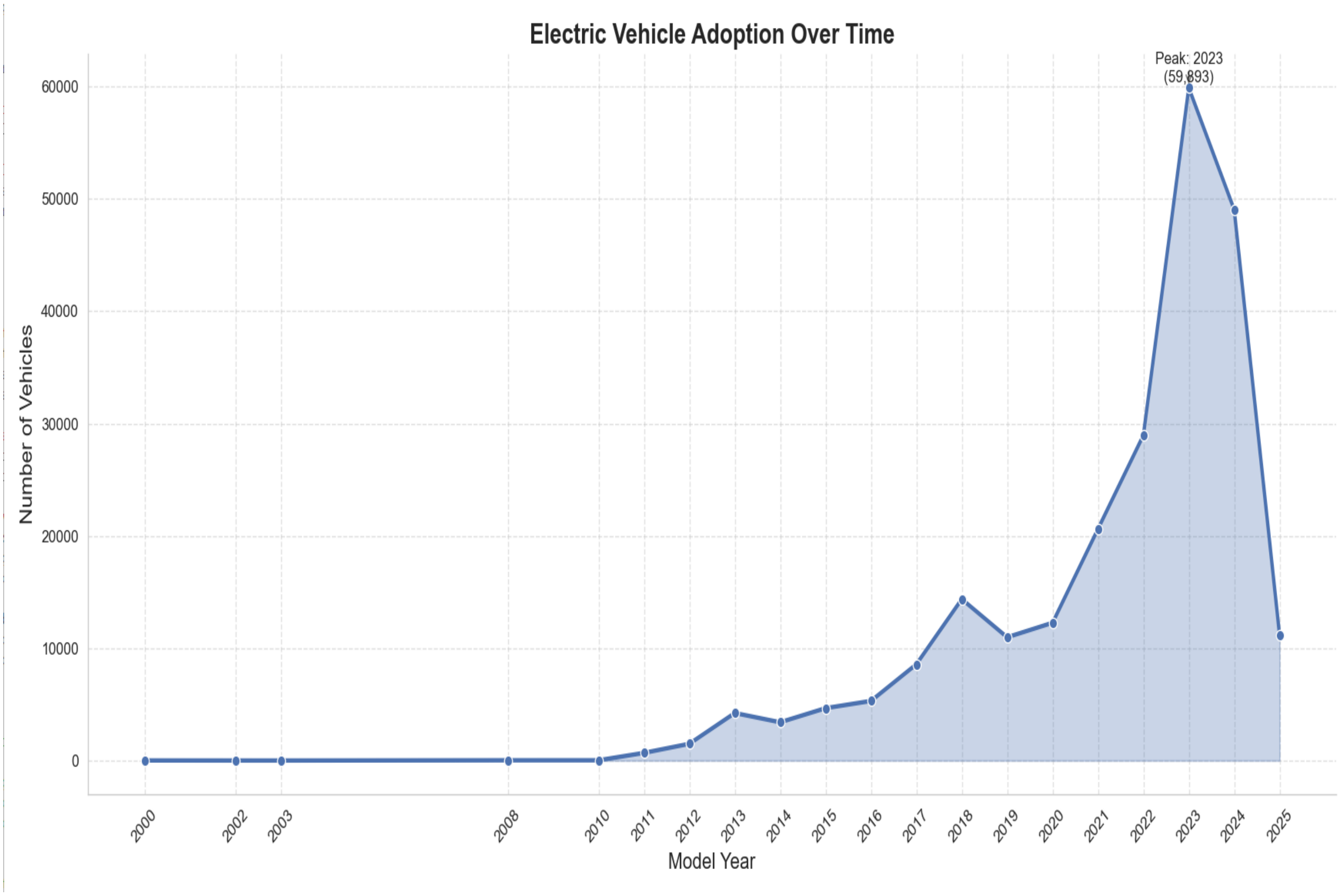




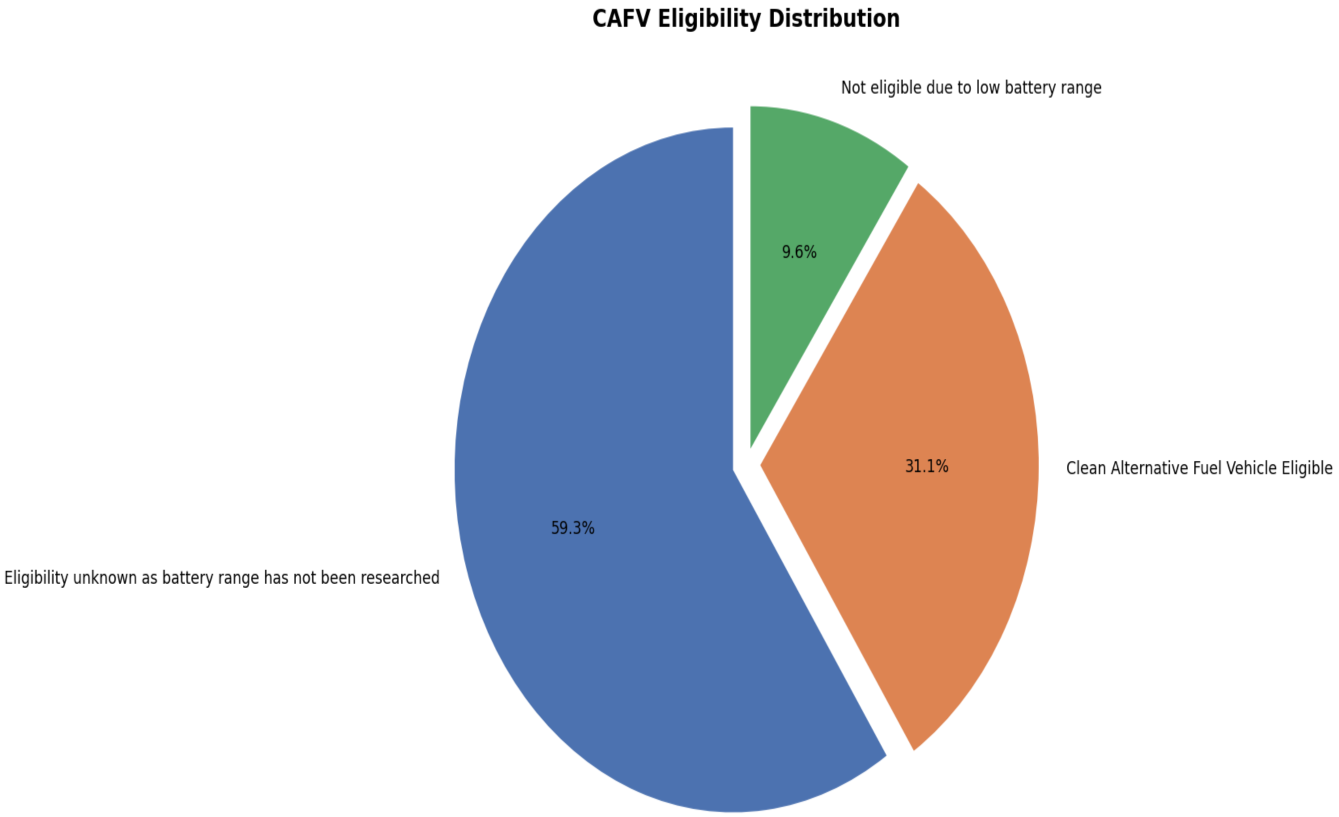
```python
# Prepare data
yearly_counts = df['Model Year'].value_counts().sort_index()
yearly_df = pd.DataFrame({'Year': yearly_counts.index, 'Count': yearly_counts.values})

# Set plot style
sns.set_style("whitegrid")
plt.figure(figsize=(14, 7))

# Line plot
ax = sns.lineplot(data=yearly_df, x='Year', y='Count', marker='o', linewidth=2.5, color='#4C72B0')

# Gradient fill under the line
ax.fill_between(yearly_df['Year'], yearly_df['Count'], alpha=0.3, color='#4C72B0')

# Highlight peak point
max_year = yearly_df.loc[yearly_df['Count'].idxmax()]
plt.annotate(f"Peak: {int(max_year['Year'])}\n({max_year['Count']:,})",
             xy=(max_year['Year'], max_year['Count']),
             xytext=(max_year['Year'], max_year['Count'] + 500),
             ha='center',
             arrowprops=dict(arrowstyle='->', color='gray'))
```

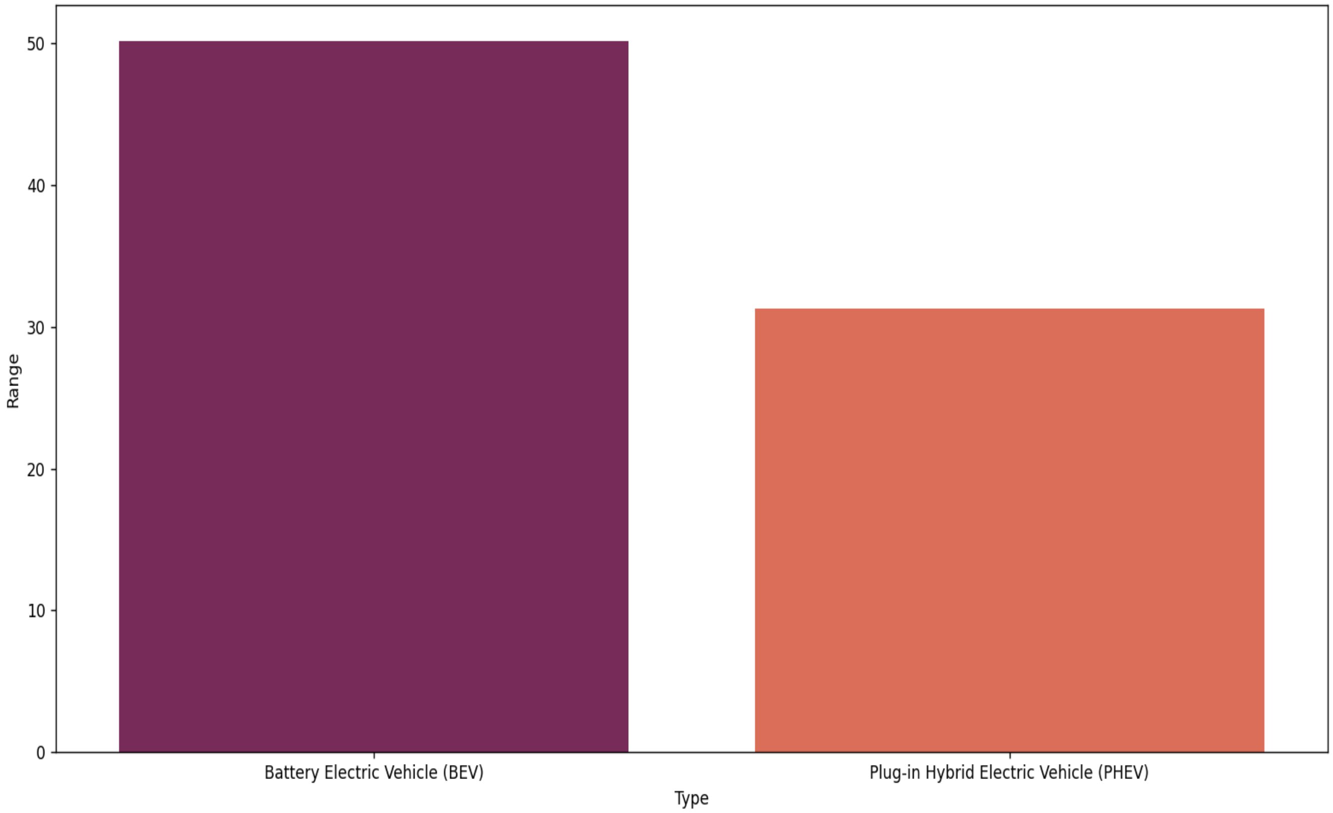


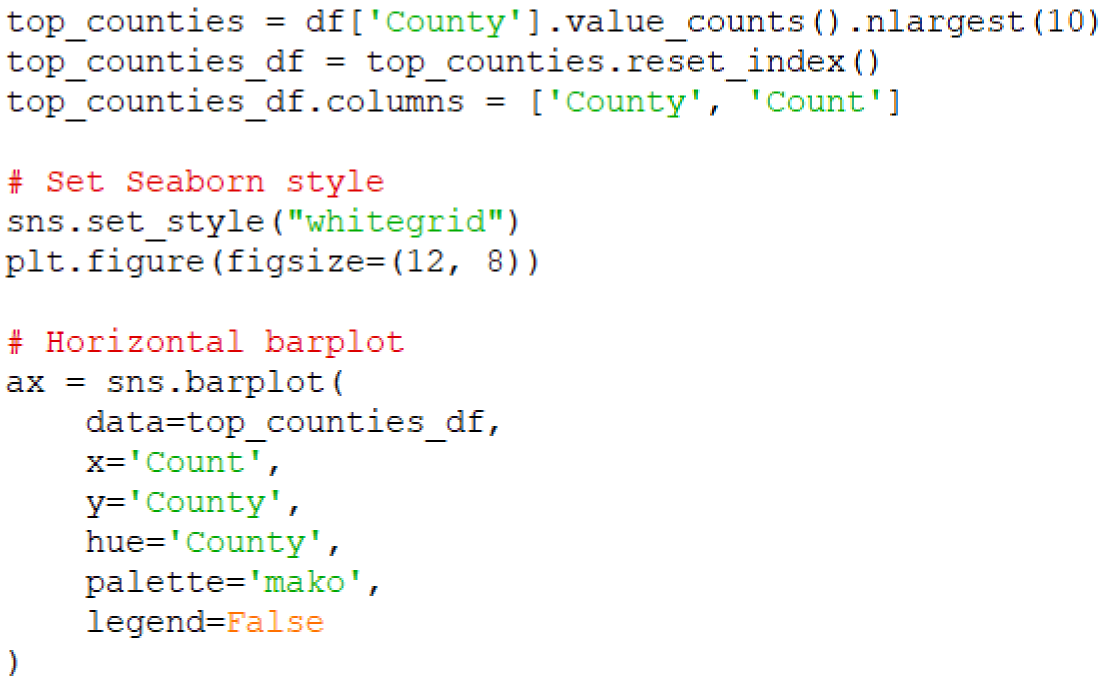
```python
# Add title and labels
plt.title('Electric Vehicle Adoption Over Time', fontsize=16, fontweight='bold')
plt.xlabel('Model Year', fontsize=13)
plt.ylabel('Number of Vehicles', fontsize=13)

# Format ticks
plt.xticks(yearly_df['Year'], rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)

# Remove top and right spines
sns.despine()
plt.tight_layout()
plt.show()

range_by_type = df.groupby('Electric Vehicle Type')['Electric Range'].mean()
range_df = range_by_type.reset_index()
range_df.columns = ['Type', 'Range']
```



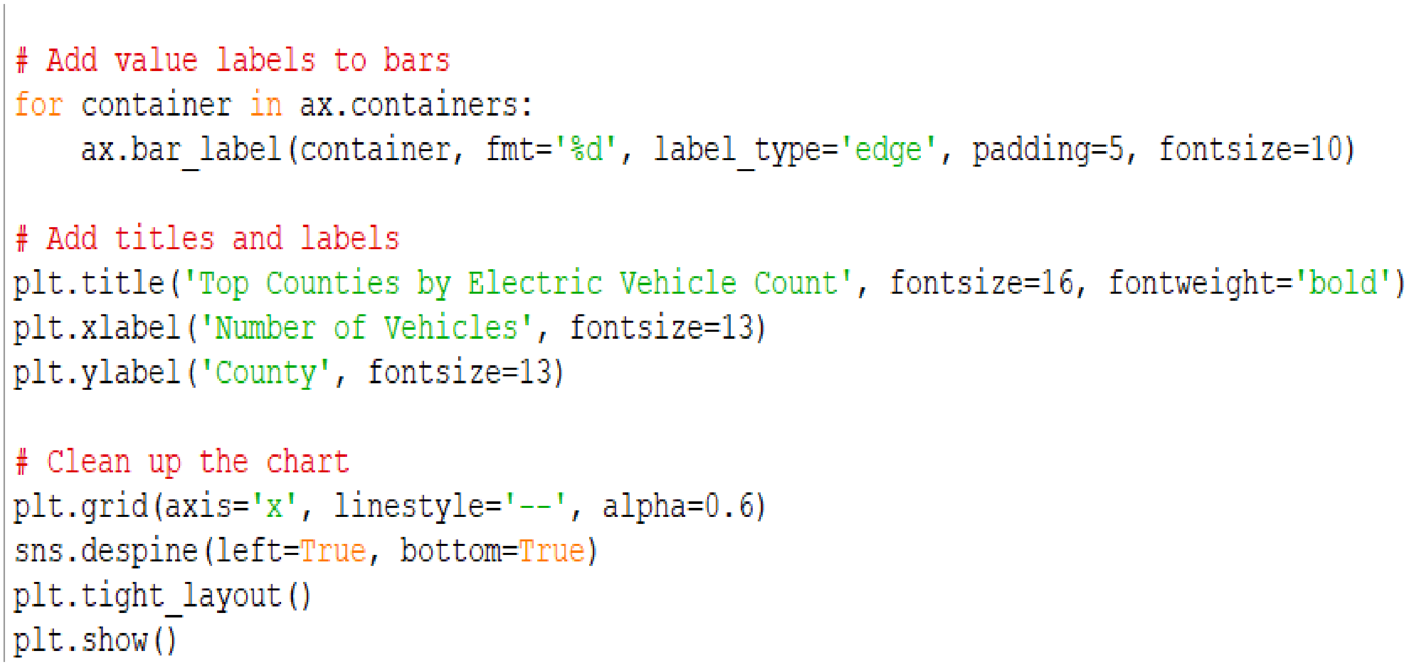
```
range_by_type = df.groupby('Electric Vehicle Type')['Electric Range'].mean()
range_df = range_by_type.reset_index()
range_df.columns = ['Type', 'Range']


# Plot
sns.barplot(data=range_df, x='Type', y='Range', hue='Type', palette='rocket', legend=False)

plt.figure(figsize=(10, 6))
cafv_counts = df['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].value_counts()
cafv_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90,
                 colors=['#4C72B0', '#DD8452', '#55A868'],
                 explode=(0.05, 0.05, 0.05))

plt.title('CAFV Eligibility Distribution', fontsize=14, fontweight='bold')
plt.ylabel('')
plt.tight_layout()
plt.show()
```


CAFV Eligibility Distribution
Not eligible due to low battery range
9.6%
31.1%
Clean Alternative Fuel Vehicle Eligible
59.3%
Eligibility unknown as battery range has not been researched

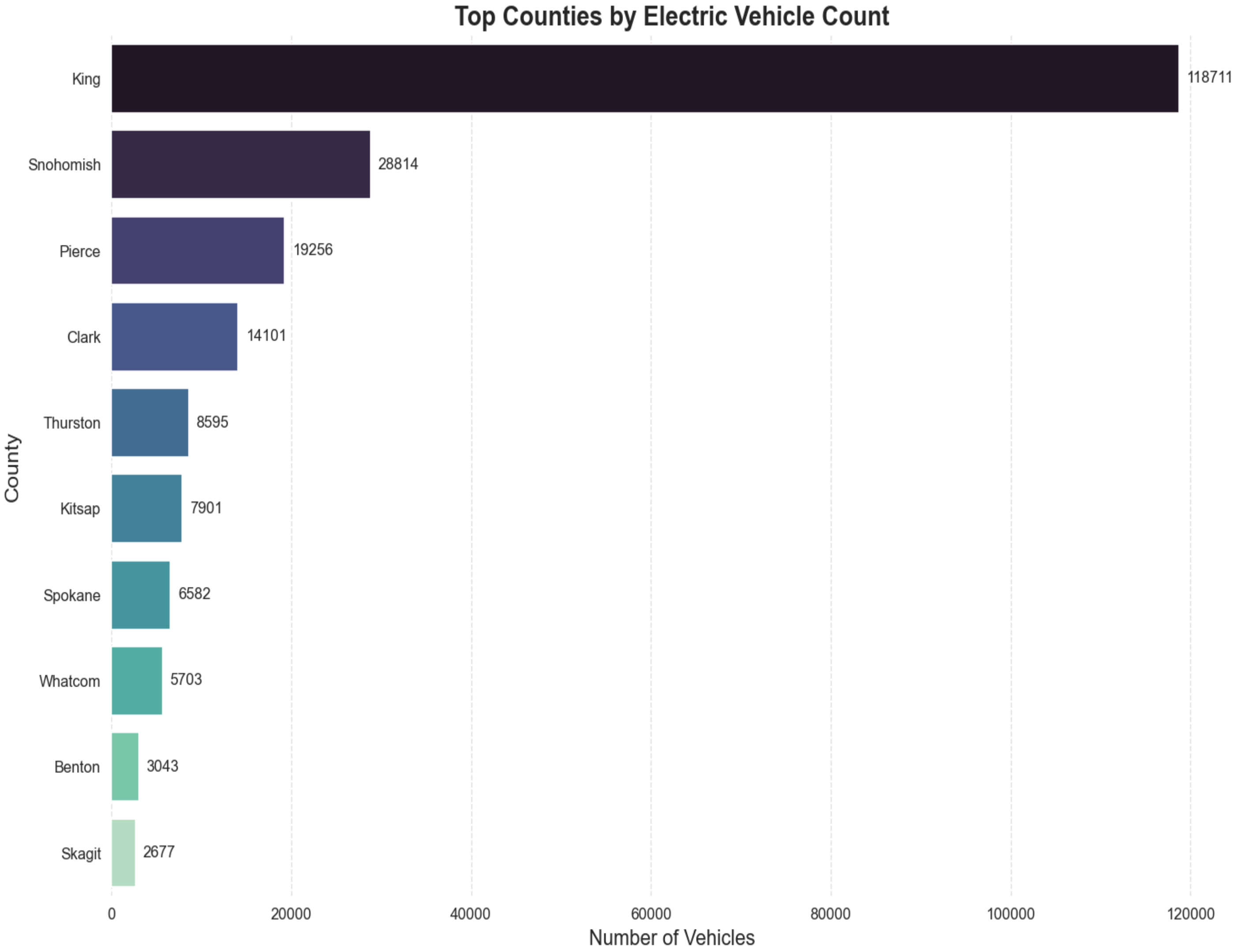

50
40
30
20
10
0
Range
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)
Type

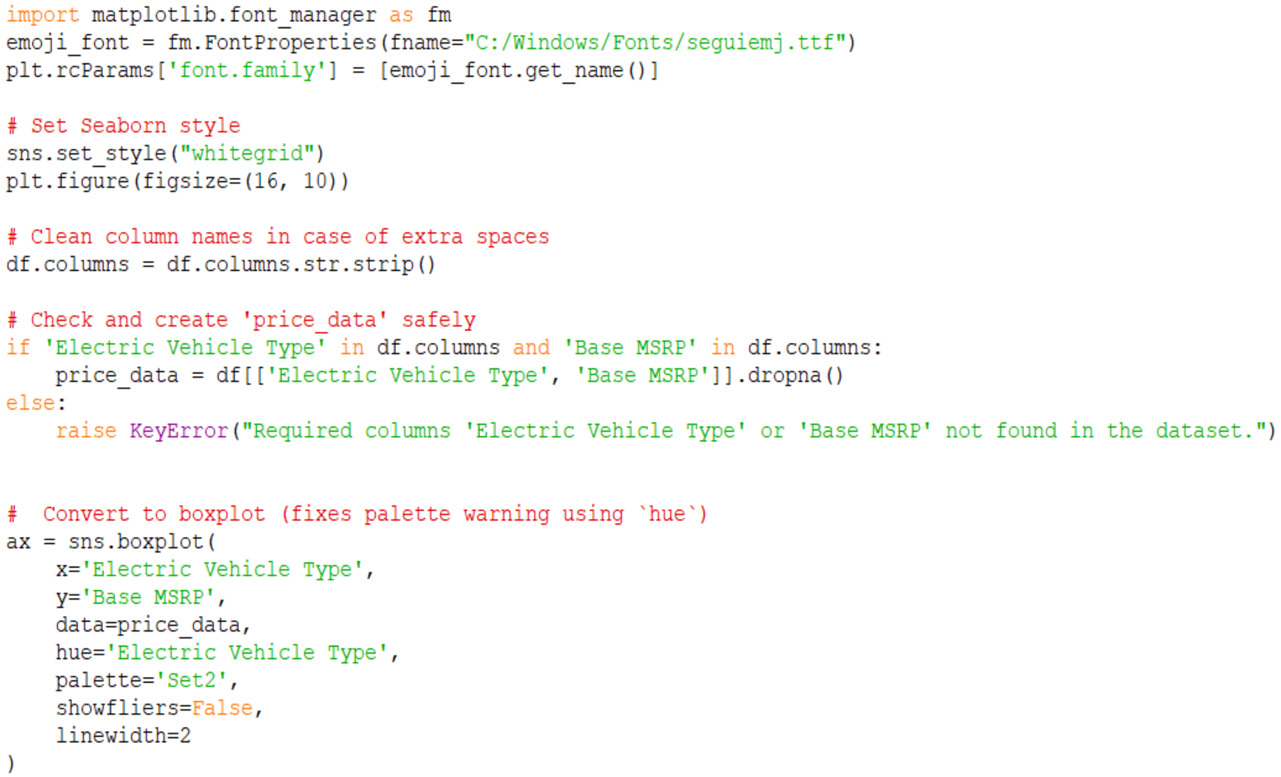


```
top_counties = df['County'].value_counts().nlargest(10)
top_counties_df = top_counties.reset_index()
top_counties_df.columns = ['County', 'Count']

# Set Seaborn style
sns.set_style("whitegrid")
plt.figure(figsize=(12, 8))

# Horizontal barplot
ax = sns.barplot(
    data=top_counties_df,
    x='Count',
    y='County',
    hue='County',
    palette='mako',
    legend=False
)
```

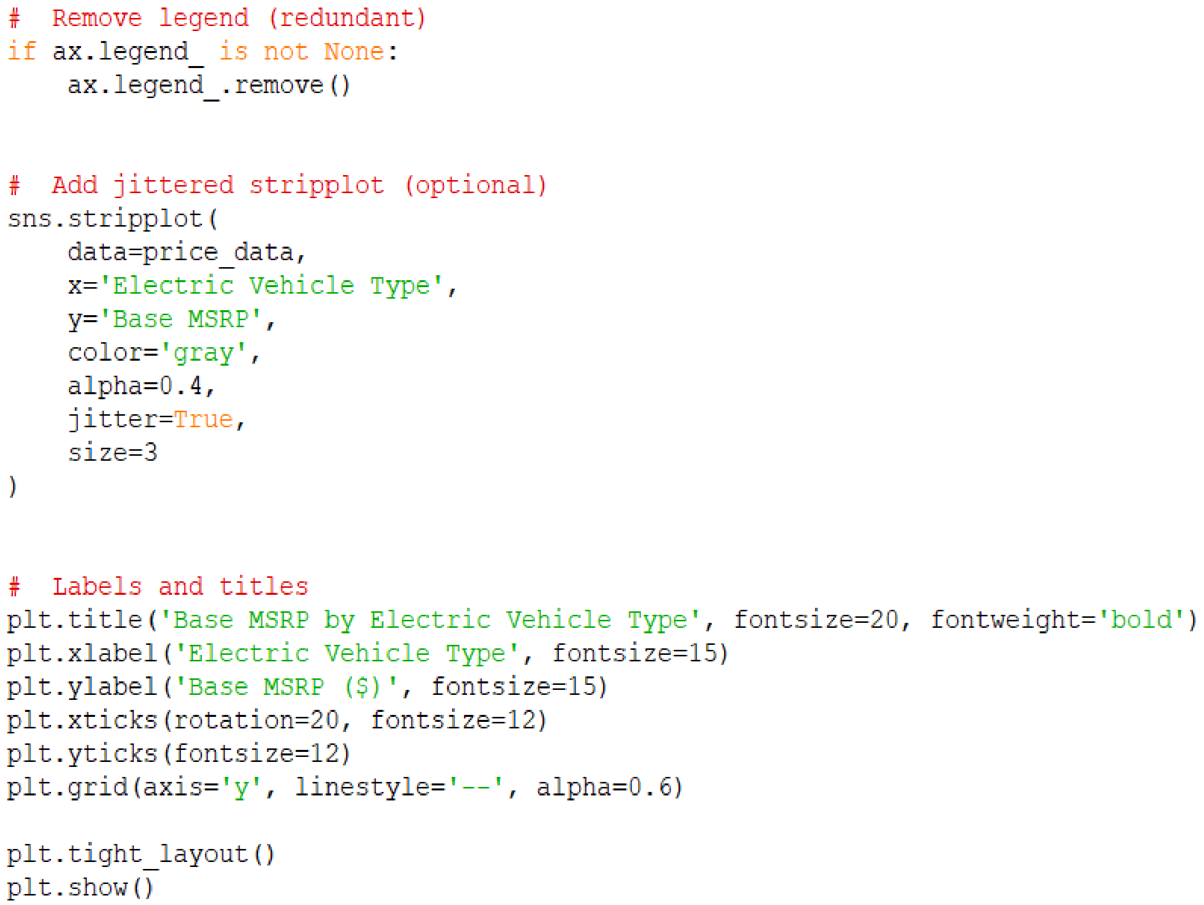


```
# Add value labels to bars
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=5, fontsize=10)

# Add titles and labels
plt.title('Top Counties by Electric Vehicle Count', fontsize=16, fontweight='bold')
plt.xlabel('Number of Vehicles', fontsize=13)
plt.ylabel('County', fontsize=13)

# Clean up the chart
plt.grid(axis='x', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)
plt.tight_layout()
plt.show()
```

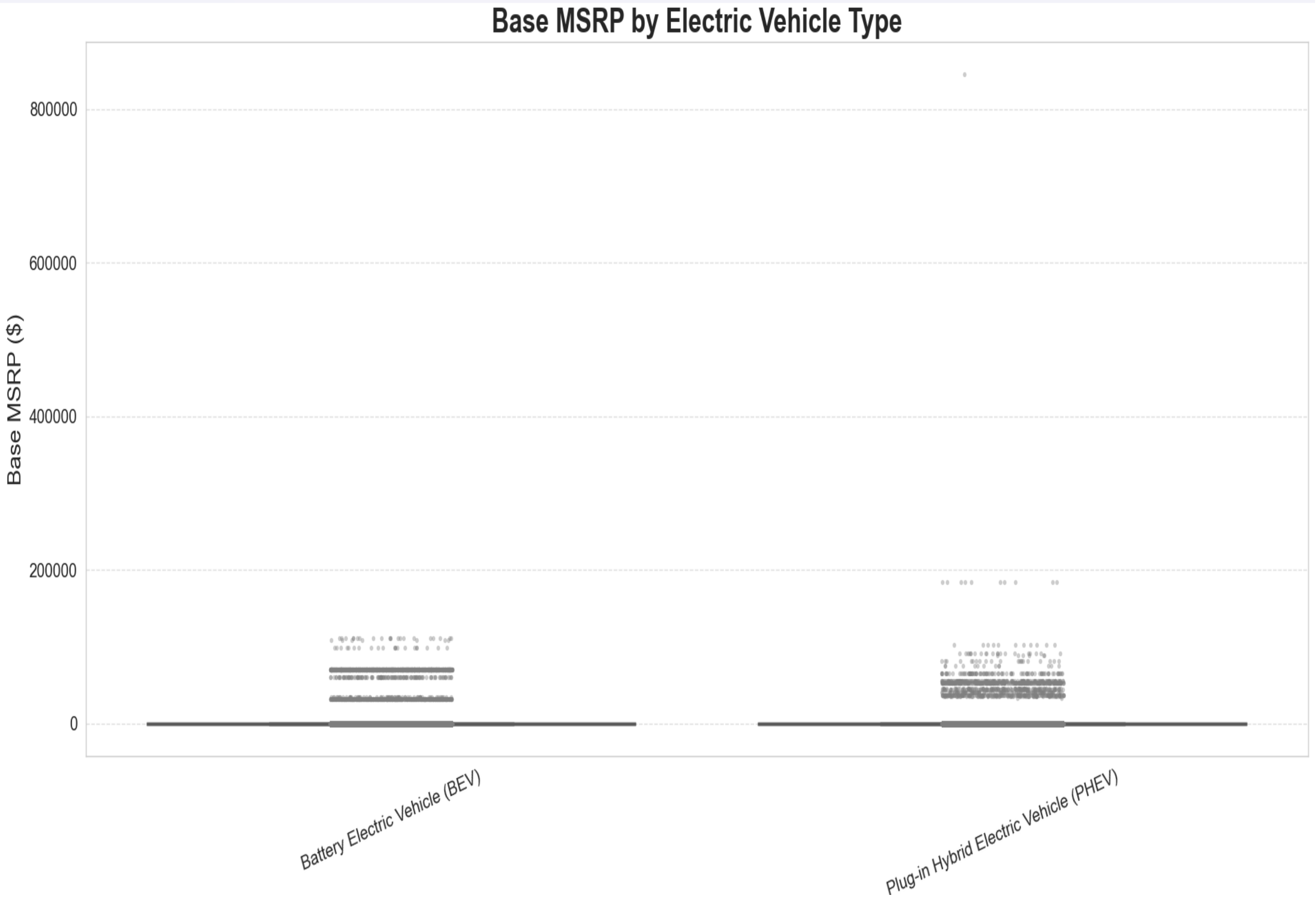

Top Counties by Electric Vehicle Count
County
Number of Vehicles
King
118711
Snohomish
28814
Pierce
19256
Clark
14101
Thurston
8595
Kitsap
7901
Spokane
6582
Whatcom
5703
Benton
3043
Skagit
2677
0
20000
40000
60000
80000
100000
120000



```
import matplotlib.font_manager as fm
emoji_font = fm.FontProperties(fname="C:/Windows/Fonts/seguiemj.ttf")
plt.rcParams['font.family'] = [emoji_font.get_name()]

# Set Seaborn style
sns.set_style("whitegrid")
plt.figure(figsize=(16, 10))

# Clean column names in case of extra spaces
df.columns = df.columns.str.strip()

# Check and create 'price_data' safely
if 'Electric Vehicle Type' in df.columns and 'Base MSRP' in df.columns:
    price_data = df[['Electric Vehicle Type', 'Base MSRP']].dropna()
else:
    raise KeyError("Required columns 'Electric Vehicle Type' or 'Base MSRP' not found in the dataset.")


#  Convert to boxplot (fixes palette warning using `hue`)
ax = sns.boxplot(
    x='Electric Vehicle Type',
    y='Base MSRP',
    data=price_data,
    hue='Electric Vehicle Type',
    palette='Set2',
    showfliers=False,
    linewidth=2
)
```



```
#  Remove legend (redundant)
if ax.legend_ is not None:
    ax.legend_.remove()


#  Add jittered stripplot (optional)
sns.stripplot(
    data=price_data,
    x='Electric Vehicle Type',
    y='Base MSRP',
    color='gray',
    alpha=0.4,
    jitter=True,
    size=3
)


#  Labels and titles
plt.title('Base MSRP by Electric Vehicle Type', fontsize=20, fontweight='bold')
plt.xlabel('Electric Vehicle Type', fontsize=15)
plt.ylabel('Base MSRP ($)', fontsize=15)
plt.xticks(rotation=20, fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.6)

plt.tight_layout()
plt.show()
```


Base MSRP by Electric Vehicle Type
Base MSRP ($)
800000
600000
400000
200000
0
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)