<u>EV POPULATION DATA</u>

NAME: SHUVAM DAS

Date: 23/03/2025

Course Details: PYTHON - PDA 2025

==

DATA →

VIN (1-10)	County	City	State	Postal Coc	Model Yea	Make	Model	Electric Ve	Clean Alte	Electric Ra	Base MSRI	Legislative	DOL Vehic	Vehicle Lc	Electric Ut	2020 Census Tract
5YJ3E1EBX	King	Seattle	WA	98178	2019	TESLA	MODEL 3	Battery El	Clean Alte	220	0	37	4.77E+08	POINT (-1:	CITY OF SE	5.3E+10
5YJYGDEE:	Kitsap	Poulsbo	WA	98370	2020	TESLA	MODEL Y	Battery El	Clean Alte	291	0	23	1.1E+08	POINT (-1:	PUGET SO	5.3E+10
KM8KRDA	Kitsap	Olalla	WA	98359	2023	HYUNDAI	IONIQ 5	Battery El	Eligibility	0	0	26	2.3E+08	POINT (-1:	PUGET SO	5.3E+10
5UXTA6C0	Kitsap	Seabeck	WA	98380	2021	BMW	X5	Plug-in Hy	Clean Alte	30	0	35	2.68E+08	POINT (-1:	PUGET SO	5.3E+10
JTMAB3FV	Thurston	Rainier	WA	98576	2023	TOYOTA	RAV4 PRIM	Plug-in Hy	Clean Alte	42	0	2	2.37E+08	POINT (-1:	PUGET SO	5.31E+10
5YJSA1DN	Thurston	Olympia	WA	98502	2012	TESLA	MODEL S	Battery El	Clean Alte	265	59900	22	1.87E+08	POINT (-1:	PUGET SO	5.31E+10
WBY1Z6C:	King	Bellevue	WA	98004	2017	BMW	I3	Battery El	Clean Alte	81	0	48	1.97E+08	POINT (-1:	PUGET SO	5.3E+10
3MW5P9J	Snohomis	Marysville	WA	98271	2022	BMW	330E	Plug-in Hy	Not eligib	22	0	39	2.05E+08	POINT (-1:	PUGET SO	5.31E+10
5YJ3E1EA6	King	Kirkland	WA	98034	2018	TESLA	MODEL 3	Battery El	Clean Alte	215	0	45	2039222	POINT (-1:	PUGET SO	5.3E+10
5YJ3E1EA2	King	Redmond	WA	98052	2018	TESLA	MODEL 3	Battery El	Clean Alte	215	0	45	4.75E+08	POINT (-1:	PUGET SO	5.3E+10
1N4AZ0CP	King	Newcastle	WA	98059	2014	NISSAN	LEAF	Battery El	Clean Alte	84	0	41	1.31E+08	POINT (-1:	PUGET SO	5.3E+10
5YJXCDE2	King	Seattle	WA	98125	2020	TESLA	MODEL X	Battery El	Clean Alte	289	0	46	2.41E+08	POINT (-1:	CITY OF SE	5.3E+10
KNDCC3LD	King	Seattle	WA	98125	2019	KIA	NIRO	Plug-in Hy	Not eligib	26	0	46	4.75E+08	POINT (-1:	CITY OF SE	5.3E+10
LPSED3KA	King	Seattle	WA	98125	2021	POLESTAR	PS2	Battery El	Clean Alte	233	0	46	1.83E+08	POINT (-1:	CITY OF SE	5.3E+10
5YJSA1H2	Thurston	Olympia	WA	98506	2015	TESLA	MODEL S	Battery El	Clean Alte	208	0	22	2.41E+08	POINT (-1:	PUGET SO	5.31E+10
WBY7Z6C5	King	Kent	WA	98031	2018	BMW	I3	Battery El	Clean Alte	114	0	11	2.22E+08	POINT (-1:	PUGET SO	5.3E+10
1N4AZ0CP	Kitsap	Olalla	WA	98359	2016	NISSAN	LEAF	Battery El	Clean Alte	84	0	26	2.26E+08	POINT (-1:	PUGET SO	5.3E+10
1N4AZ0CP	King	Issaquah	WA	98029	2015	NISSAN	LEAF	Battery El	Clean Alte	84	0	5	1.12E+08	POINT (-1:	PUGET SO	5.3E+10

SUMMARY→

The dataset provides information on Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) registered in Washington State through the Department of Licensing (DOL). It includes details such as the vehicle's VIN (first 10 characters), registration location (county, city, state, postal code), model year, make, model, and electric vehicle type (BEV or PHEV). Additional data covers Clean Alternative Fuel Vehicle (CAFV) eligibility, electric range, base MSRP (Manufacturer's Suggested Retail Price), legislative district, DOL Vehicle ID, GPS coordinates of the registered location, electric utility provider, and 2020 Census Tract for demographic analysis. This dataset is useful for understanding EV adoption, geographic distribution, and eligibility for clean energy incentives in Washington State.

OBJECTIVE→

- **Data Cleaning and Preparation**
- **Exploratory Data Analysis (EDA)**
- **Data Visualization**
- **Predictive Modeling**
- **Insight Generation and Reporting**

PYTHON CODE:

```
import numpy as np
import pandas as pd
import hashlib
df = pd.read_csv("Electric_Vehicle_Population_Data (1).csv")
df.info()
df.head()
```

DATA CLEANING

```
# DATA CLEANING
# How many missing values exist in the dataset, and in which columns?
missing_values = df.isnull().sum()
print ("\n the total missing values in dataset", missing_values.sum())
print ("\n the total missing values in each column", missing_values)
#How should missing or zero values in the Base MSRP and Electric Range columns be handled?
 # replace zero values
df["Base MSRP"] = df["Base MSRP"].replace(0, np.nan)
df["Electric Range"] = df["Electric Range"].replace(0, np.nan)
 # replace missing values by median values
df["Base MSRP"] = df["Base MSRP"].fillna(df["Base MSRP"].median())
df["Electric Range"] = df["Electric Range"].fillna(df["Electric Range"].median())
print ("\n missing and zero values in base MSRP and Electric Range columns handled", df["Base MSRP"].isnull().sum(), df["Electric Range"].isn
#Are there duplicate records in the dataset? If so, how should they be managed?
duplicate_value = df.duplicated().sum()
print ("\n duplicate records in dataset", duplicate_value)
df.drop_duplicates(inplace=True)
#How can VINs be anonymized while maintaining uniqueness?
 # unique Id in this dataset is VIN[1-10]
df["VIN (1-10)"] = df["VIN (1-10)"].apply(lambda x: hashlib.sha256(x.encode()).hexdigest())
print ("\n maintaing uniqueness ", df["VIN (1-10)"])
#How can Vehicle Location (GPS coordinates) be cleaned or converted for better readability?
df["Latitude"] = df["Vehicle Location"].str.extract(r'POINT \((-?\d+\.\d+) (-?\d+\.\d+)\)')[1]
df["Longitude"] = df["Vehicle Location"].str.extract(r'POINT \((-?\d+\.\d+) (-?\d+\.\d+)\)')[0]
   # convert into neumeric
df["Latitude"] = pd.to_numeric(df["Latitude"])
df["Longitude"] = pd.to_numeric(df["Longitude"])
print ("\n cleaned or converted for better readability", df["Latitude"], df["Longitude"])
```

DATA CLEANING output:

```
the total missing values in dataset 591

 the total missing values in each 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

 missing and zero values in base MSRP and Electric Range columns handled 0 0

 duplicate records in dataset 0

 maintaing uniqueness  0        78953a9f9d62e8cc12a944c5a3c1e08a4d3e1b55a9759e...
1        1c9d2d25cd197a4ce1ae18cfa52ba501595bc080e302b5...
2        b6c2f4f0ec2c32784fe5fa4baee0df7354efc30a099f45...
3        7d99e2eebf8784a9fb6675cfea068614bf06f41f281295...
4        e3ea04034fda2a426a9c141dc1fa4b903adf05fdf1f514...
                               ...
235687    d56843e71102a5a4fbf0e3c553189fc66135f55de8ed6d...
235688    18bdea8b8f3ba6af1df770f743e0cc49cb4d0b165eed08...
235689    55dfe19ac9780e08e476343bca6f00a135e5d2984e82c2...
235690    b473fe66e7749a8624292b8cbdceeec21229f650d206c4...
235691    084df2ada60f054781d8acd25a187f68e244c0bfa566df...
Name: VIN (1-10), Length: 235692, dtype: object

 cleaned or converted for better readability 0        47.49461
1        47.73689
2        47.42602
3        47.64509
4        46.88897
           ...
235687    47.29238
235688    48.24159
235689    47.67858
235690    48.01497
235691    47.53010
Name: Latitude, Length: 235692, dtype: float64 0        -122.23825
1       -122.64681
2       -122.54729
3       -122.81585
4       -122.68993
           ...
235687   -122.51134
235688   -122.37265
235689   -122.13158
235690   -122.06402
235691   -122.03439
Name: Longitude, Length: 235692, dtype: float64
```

DATA EXPLORATION

```
import numpy as np
import pandas as pd
import hashlib
df = pd.read_csv("Electric_Vehicle_Population_Data (1).csv")
# Data EXPLORATION
# What are the top 5 most common EV makes and models in the dataset
top_5_makes = df["Make"].value_counts().head(5)
top_5_models = df["Model"].value_counts().head(5)
print ("\n top 5 makes ", top_5_makes)
print ("\n top 5 models ", top_5_models)
#What is the distribution of EVs by county? Which county has the most registrations?
ev_distribution_by_county = df["County"].value_counts()
most_registered_county = ev_distribution_by_county.idxmax()
print ("\n distribution of EVs by county ", ev_distribution_by_county)
print ("\n county with most registrations ", most_registered_county)
# How has EV adoption changed over different model years?
ev_adoptation_over_model_year = df["Model Year"].value_counts()
print ("\n adoptation changed by year", ev_adoptation_over_model_year)
```

```
# What is the average electric range of EVs in the dataset?
average_electric_range = df["Electric Range"].mean()
print ("\n average electric range ", average_electric_range)
# What percentage of EVs are eligible for Clean Alternative Fuel Vehicle (CAFV) incentives?
cafv_incentives_percentage = (df["Clean Alternative Fuel Vehicle (CAFV) Eligibility"] == "Clean Alternative Fuel Vehicle Eligible").mean() * 100
print ("\n percentage of EVs eligible for CAFV incentives ", cafv_incentives_percentage)
# How does the electric range vary across different makes and models?
electric_range_by_make = df.groupby('Make')['Electric Range'].mean()
print ("\n electric range across different makes and models ", electric_range_by_make)
electric_range_by_model = df.groupby('Model')['Electric Range'].mean()
print ("\n electric range across different models ", electric_range_by_model)
# What is the average Base MSRP for each EV model?
average_base_msrp_by_ev_model = df.groupby('Model')['Base MSRP'].mean()
print ("\n average Base MSRP for each EV model ", average_base_msrp_by_ev_model)
# Are there any regional trends in EV adoption (e.g., urban vs. rural areas)?
def check_urbal_rural(census):
  if census >= 2000000000:
    return "urban"
  else:
    return "rural"
df["Urban/Rural"]= df["2020 Census Tract"].apply(check_urbal_rural)
ev_adoption_by_urban_rural = df.groupby('Urban/Rural')['Model'].count()
print ("\n regional trends in EV adoption ", ev_adoption_by_urban_rural)

df.head()
```

OTUTPUT

```
 top 5 makes   Make
TESLA         26670
NISSAN         4056
CHEVROLET      4018
KIA            2920
BMW            2715
```

```
Name: count, dtype: int64
```

```
 top 5 models  Model
MODEL Y    13089
MODEL 3     9497
LEAF        3653
MODEL S     1961
MODEL X     1842
Name: count, dtype: int64
```

```
 distribution of EVs by county  County
King             44464
Clark             6004
Snohomish         3442
Kitsap            2776
Thurston          1858
Cowlitz            518
Jefferson          439
Yakima             429
Pierce             260
Island             207
Spokane             98
Whatcom             98
Clallam             73
Skagit              44
Stevens             43
Benton              35
Klickitat           31
Walla Walla         20
Chelan              17
Whitman             15
Grant               13
San Juan            12
Kittitas             9
Franklin             9
Lewis                9
Douglas              8
Mason                7
Skamania             7
Grays Harbor         5
Okanogan             5
Wahkiakum            4
Pend Oreille         3
Asotin               2
Pacific              2
Lincoln              2
Adams                1
Name: count, dtype: int64
```

```
 county with most registrations  King
```

```
 adoptation changed by year Model Year
2023    16370
2024    12316
2022     7425
2021     5233
2018     3779
2020     3107
2025     2964
2019     2833
```

```
2017     2083
2016     1381
2015     1125
2013     1024
2014      792
2012      341
2011      174
2008        9
2010        9
2000        3
2002        1
Name: count, dtype: int64
```

```
 average electric range  46.413794234902305
```

```
 percentage of EVs eligible for CAFV incentives  31.407764601682825
```

```
 electric range across different makes and models  Make
ACURA                          0.000000
ALFA ROMEO                    33.000000
AUDI                          45.315834
BENTLEY                       21.000000
BMW                           29.419890
BRIGHTDROP                     0.000000
CADILLAC                       2.678571
CHEVROLET                     81.097063
CHRYSLER                      32.126917
DODGE                         32.000000
FIAT                          76.797170
FISKER                         1.783784
FORD                           8.324701
GENESIS                        0.000000
GMC                            0.000000
HONDA                         19.549889
HYUNDAI                       16.362060
JAGUAR                       190.894737
JEEP                          22.052117
KIA                           34.982877
LAMBORGHINI                    6.000000
LAND ROVER                    44.166667
LEXUS                         21.404762
LINCOLN                       23.800000
LUCID                          0.000000
MAZDA                         25.392523
MERCEDES-BENZ                 11.699571
MINI                          12.027304
MITSUBISHI                    32.674157
MULLEN AUTOMOTIVE INC.         0.000000
NISSAN                        72.956114
POLESTAR                      30.338542
PORSCHE                       57.432184
RIVIAN                         0.000000
SMART                         61.365079
SUBARU                         0.640873
TESLA                         59.467209
TH!NK                        100.000000
TOYOTA                        28.187973
VINFAST                        0.000000
VOLKSWAGEN                    22.538828
VOLVO                         17.682214
```

```
Name: Electric Range, dtype: float64
```

```
 electric range across different models  Model
330E    18.444444
500     85.689474
500E     0.000000
530E    16.080645
550E    40.000000
          ...
XC60    27.282805
XC90    24.193141
XM      31.000000
ZDX      0.000000
ZEVO     0.000000
Name: Electric Range, Length: 159, dtype: float64
```

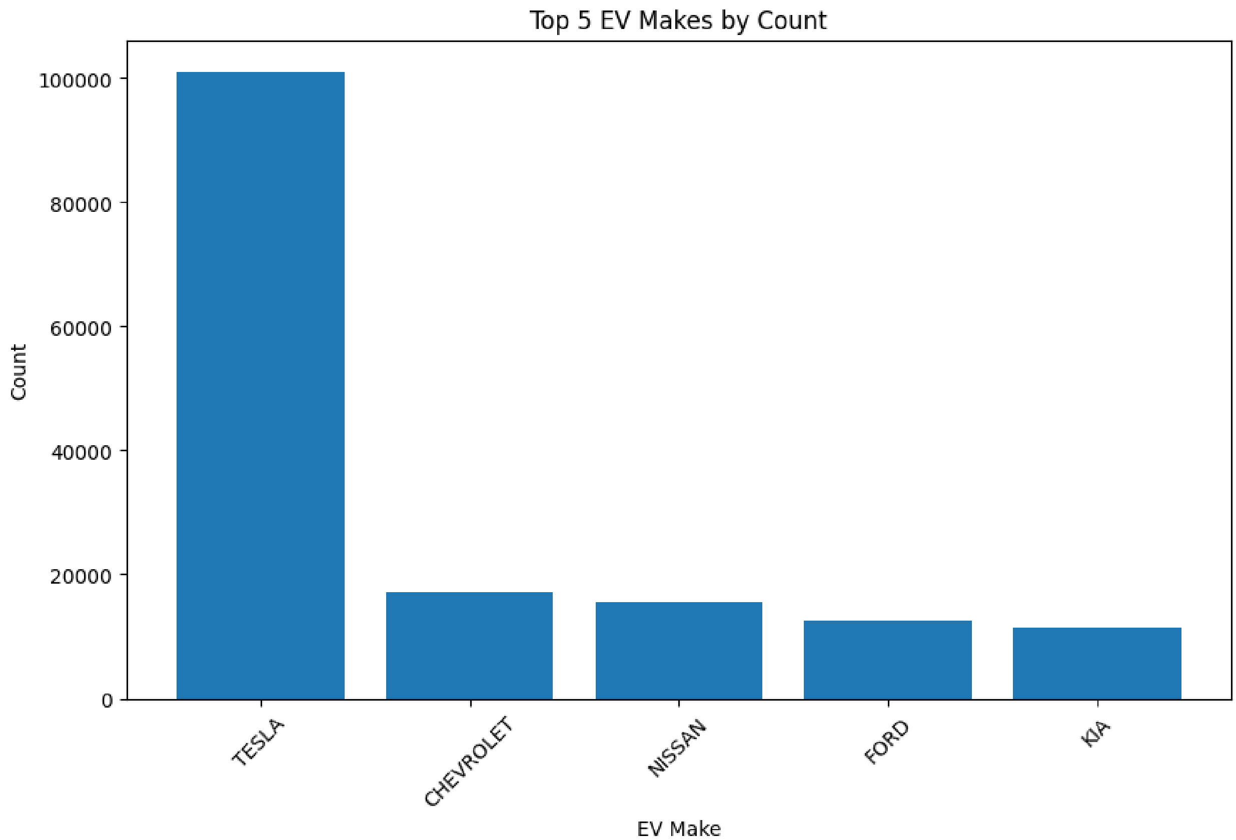
```
 average Base MSRP for each EV model  Model
330E    15257.516340
500         0.000000
500E        0.000000
530E    37279.838710
550E        0.000000
            ...
XC60     7420.361991
XC90     3399.909747
XM          0.000000
ZDX         0.000000
ZEVO        0.000000
Name: Base MSRP, Length: 159, dtype: float64
```

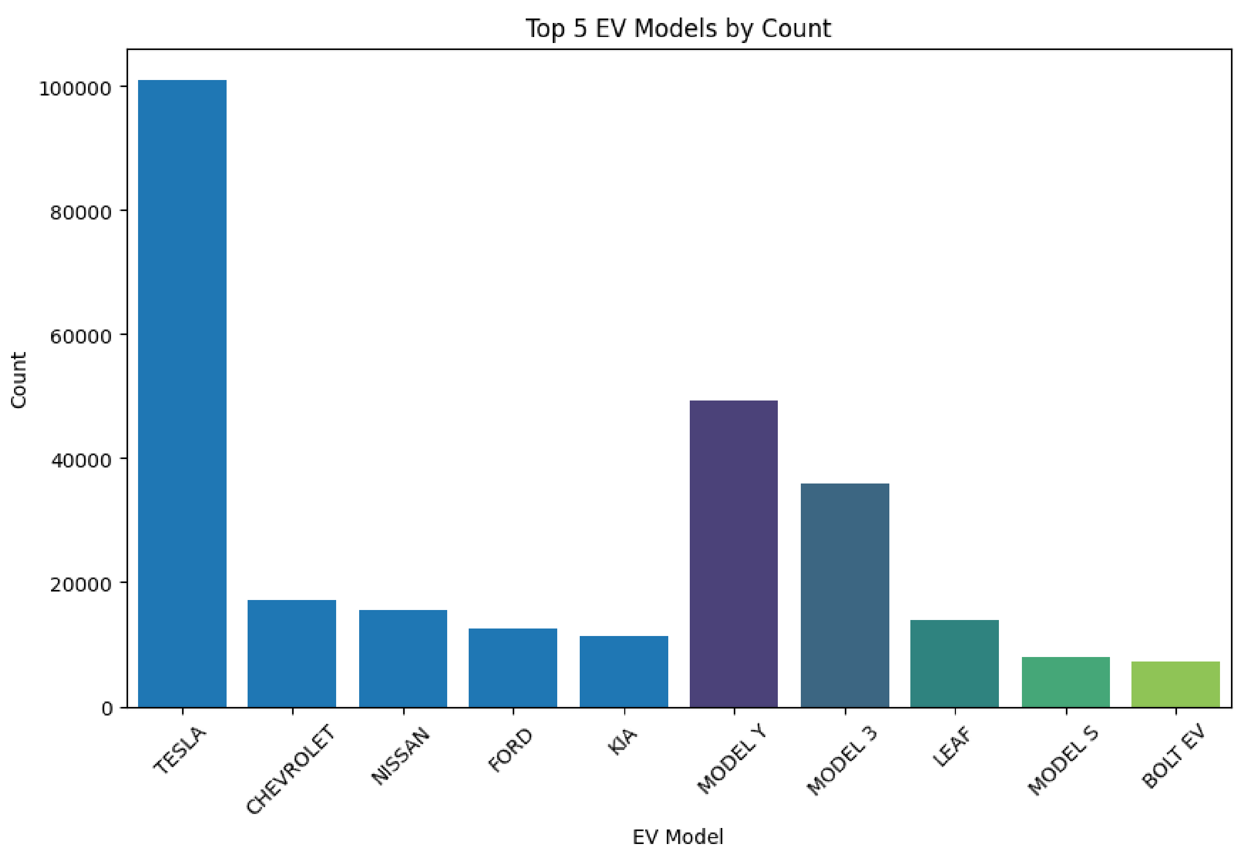
```
 regional trends in EV adoption  Urban/Rural
rural        1
urban    60968
```

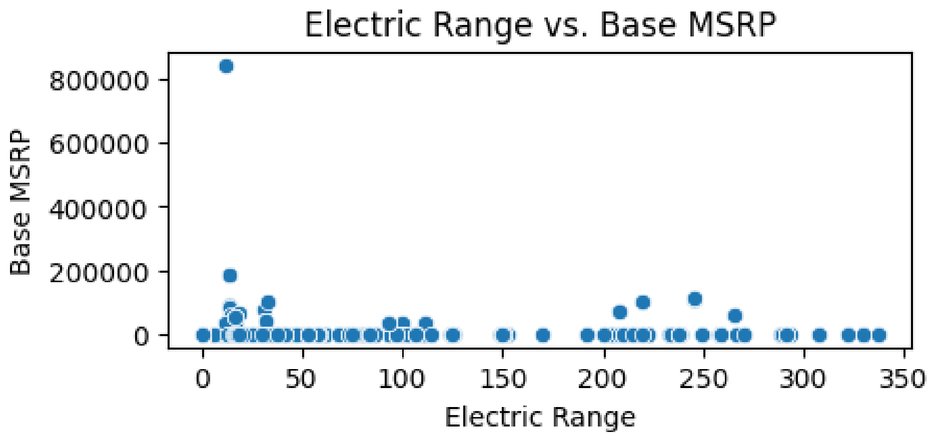
DATA VISUALIZATION

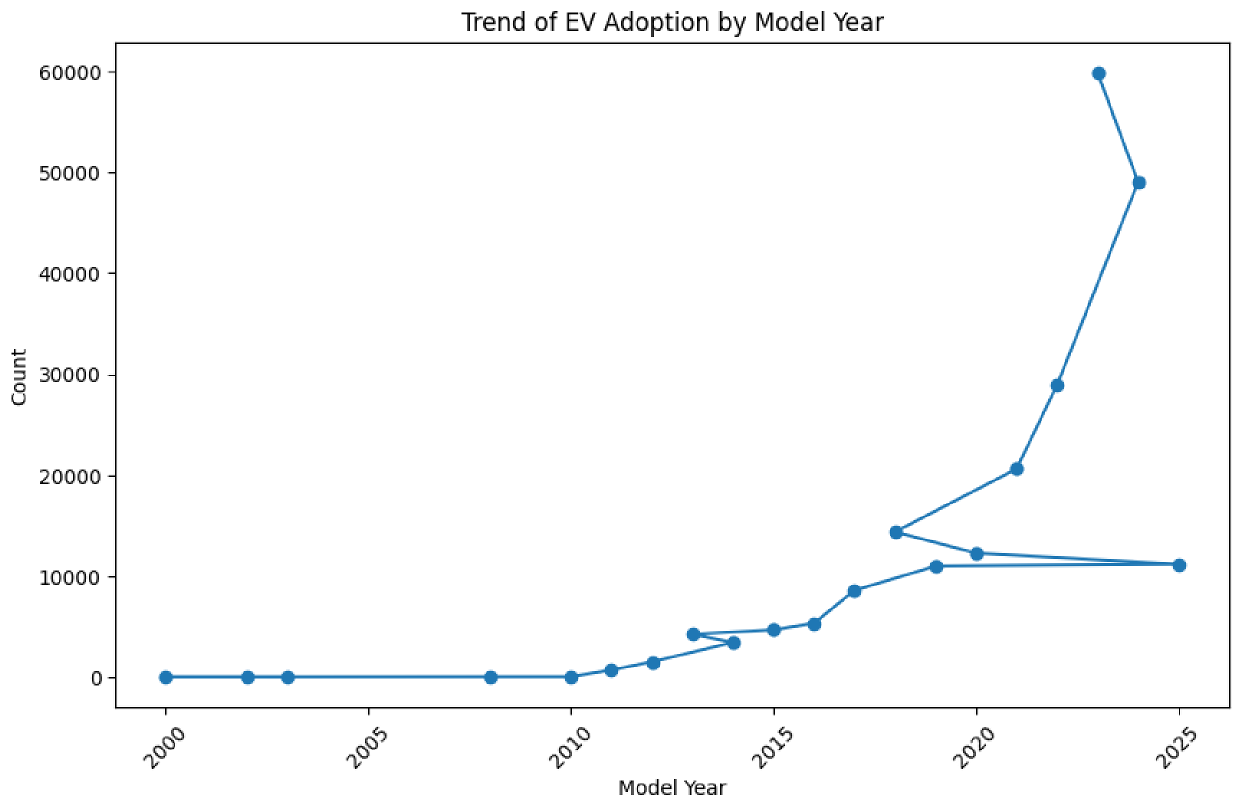
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("Electric_Vehicle_Population_Data (1).csv")

# DATA VISUALIZATION
#Create a bar chart showing the top 5 EV makes and models by count
plt.figure(figsize=(5, 2))
plt.bar(top_5_makes.index, top_5_makes.values)
plt.xlabel("EV Make")
plt.ylabel("Count")
plt.title("Top 5 EV Makes by Count")
```

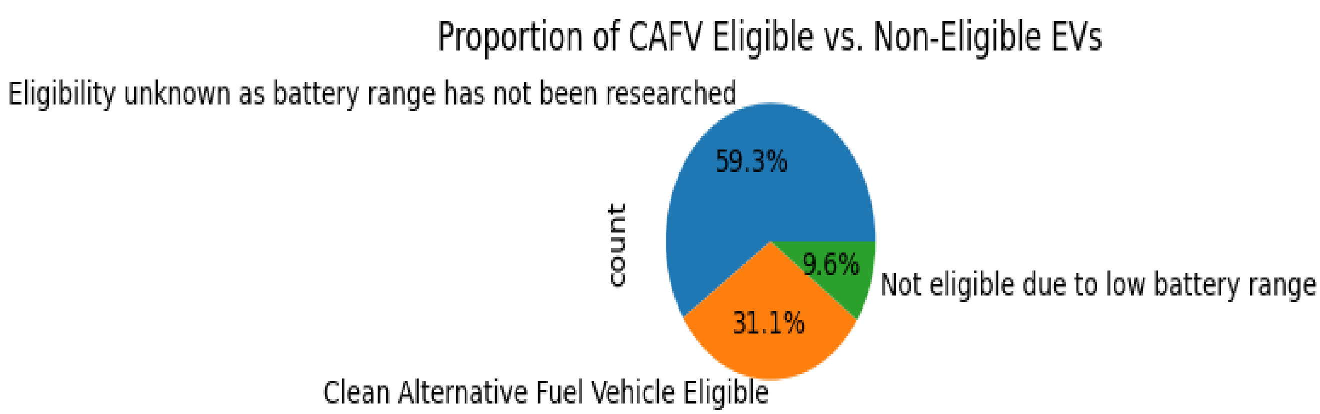
```
plt.xticks(rotation=45)
plt.show
sns.countplot(x="Model", data=df, order=df["Model"].value_counts().head(5).index,
palette="viridis")
plt.xlabel("EV Model")
plt.ylabel("Count")
plt.title("Top 5 EV Models by Count")
plt.xticks(rotation=45)
plt.show
#Use a heatmap or choropleth map to visualize EV distribution by county.

#Create a line graph showing the trend of EV adoption by model year
plt.figure(figsize=(5, 2))
ev_adoptation_over_model_year.plot(kind="line", marker="o")
plt.xlabel("Model Year")
plt.ylabel("Count")
plt.title("Trend of EV Adoption by Model Year")
plt.xticks(rotation=45)
# Generate a scatter plot comparing electric range vs. base MSRP to see pricing
trends.
plt.figure(figsize=(5, 2))
sns.scatterplot(x = "Electric Range", y= "Base MSRP", data=df)
plt.xlabel("Electric Range")
plt.ylabel("Base MSRP")
plt.title("Electric Range vs. Base MSRP")
plt.show
# Plot a pie chart showing the proportion of CAFV-eligible vs. non-eligible EVs.
plt.figure(figsize=(5, 2))
df["Clean Alternative Fuel Vehicle (CAFV)
Eligibility"].value_counts().plot(kind="pie", autopct="%1.1f%%")
plt.title("Proportion of CAFV Eligible vs. Non-Eligible EVs")
# Use a geospatial map to display EV registrations based on vehicle location.
import folium
#map = folium.Map(location=[df["Latitude"].mean(), df["Longitude"].mean()],
zoom_start=6)
```


Top 5 EV Makes by Count
100000
80000
60000
40000
20000
0
Count
TESLA
CHEVROLET
NISSAN
FORD
KIA
EV Make


Top 5 EV Models by Count
100000
80000
60000
40000
20000
0
Count
TESLA
CHEVROLET
NISSAN
FORD
KIA
MODEL Y
MODEL 3
LEAF
MODEL S
BOLT EV
EV Model


Electric Range vs. Base MSRP
Base MSRP
800000
600000
400000
200000
0
0
50
100
150
200
250
300
350
Electric Range


Trend of EV Adoption by Model Year
Count
60000
50000
40000
30000
20000
10000
0
2000
2005
2010
2015
2020
2025
Model Year


Proportion of CAFV Eligible vs. Non-Eligible EVs
Eligibility unknown as battery range has not been researched
59.3%
count
9.6%
Not eligible due to low battery range
31.1%
Clean Alternative Fuel Vehicle Eligible

Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
df = pd.read_csv("Electric_Vehicle_Population_Data (1).csv")
# Linear Regression

# Linear Regression Model
features = ['Model Year', 'Base MSRP']
df = df.dropna(subset=['Electric Range', 'Base MSRP'])
X = df[features]
y = df['Electric Range']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
print(f'R² score: {r2:.2f}')

# Influence of Base MSRP on Electric Range
plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Base MSRP'], y=df['Electric Range'], alpha=0.5)
plt.title('Electric Range vs. Base MSRP')
plt.xlabel('Base MSRP')
plt.ylabel('Electric Range')
plt.show()

# Prediction for a new EV model
new_ev = pd.DataFrame({'Model Year': [2025], 'Base MSRP': [40000]})
predicted_range = model.predict(new_ev)
print(f'Predicted Electric Range for new EV: {predicted_range[0]:.2f} miles')




# 4.6 What steps are needed to improve the accuracy of the Linear Regression model?
# Possible steps: Feature engineering, adding more relevant features, using more
complex models, etc.
```

```
# 4.7 Can we use this model to predict the range of new EV models based on their specifications?
# Yes, the model can be used to predict the range of new EV models by providing their specifications.
```