```python
import pandas as pd

ev_data = pd.read_csv('Electric_Vehicle_Population_Data.csv')
print(ev_data.head())
```

```
   VIN (1-10)     County       City State  Postal Code  Model Year
Make  \
0  5YJYGDEE1L       King    Seattle    WA      98122.0        2020
TESLA
1  7SAYGDEE9P  Snohomish    Bothell    WA      98021.0        2023
TESLA
2  5YJSA1E4XK       King    Seattle    WA      98109.0        2019
TESLA
3  5YJSA1E27G       King   Issaquah    WA      98027.0        2016
TESLA
4  5YJYGDEE5M     Kitsap  Suquamish    WA      98392.0        2021
TESLA

     Model           Electric Vehicle Type  \
0  MODEL Y  Battery Electric Vehicle (BEV)
1  MODEL Y  Battery Electric Vehicle (BEV)
2  MODEL S  Battery Electric Vehicle (BEV)
3  MODEL S  Battery Electric Vehicle (BEV)
4  MODEL Y  Battery Electric Vehicle (BEV)

   Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric
Range  \
0           Clean Alternative Fuel Vehicle Eligible             291

1  Eligibility unknown as battery range has not b...               0

2           Clean Alternative Fuel Vehicle Eligible             270

3           Clean Alternative Fuel Vehicle Eligible             210

4  Eligibility unknown as battery range has not b...               0


   Base MSRP  Legislative District  DOL Vehicle ID  \
0          0                  37.0       125701579
1          0                   1.0       244285107
2          0                  36.0       156773144
3          0                   5.0       165103011
4          0                  23.0       205138552

                Vehicle Location  \
0   POINT (-122.30839 47.610365)
1  POINT (-122.179458 47.802589)
2   POINT (-122.34848 47.632405)
3   POINT (-122.03646 47.534065)
```

```
4   POINT (-122.55717 47.733415)

                                Electric Utility  2020 Census Tract
0   CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)       5.303301e+10
1                         PUGET SOUND ENERGY INC       5.306105e+10
2   CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)       5.303301e+10
3  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303303e+10
4                         PUGET SOUND ENERGY INC       5.303594e+10
```

So, this data is based on the EV population in the United States. Now, let's clean the dataset before moving forward:

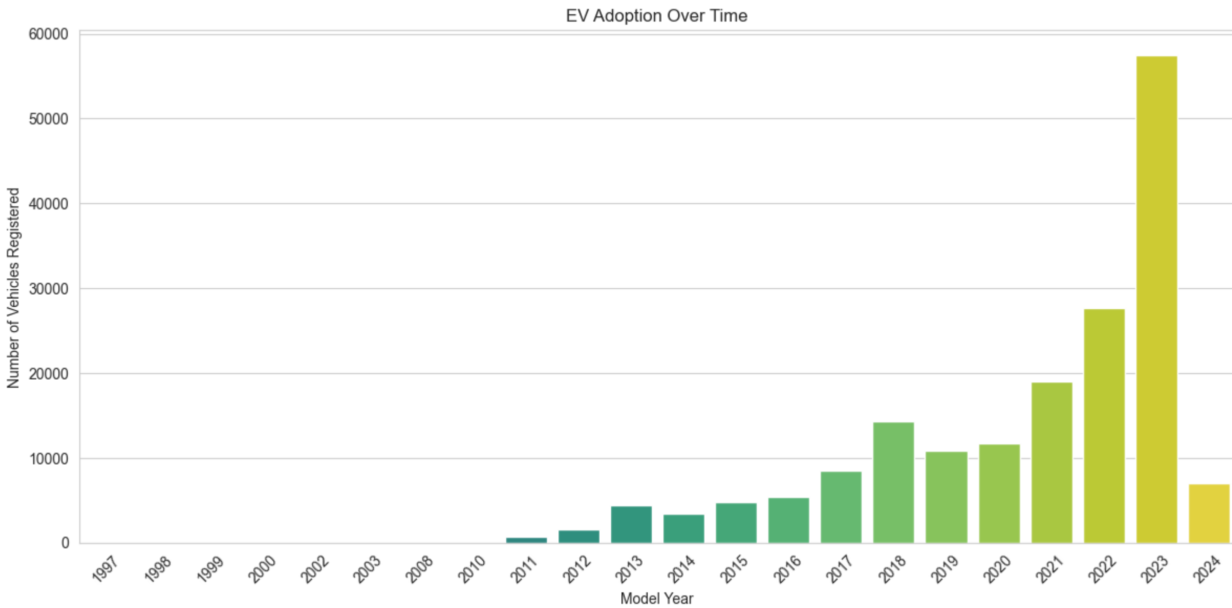
```
ev_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177866 entries, 0 to 177865
Data columns (total 17 columns):
 #   Column                                             Non-Null Count
Dtype
---  ------                                             --------------
-----
 0   VIN (1-10)                                         177866 non-
null  object
 1   County                                             177861 non-
null  object
 2   City                                               177861 non-
null  object
 3   State                                              177866 non-
null  object
 4   Postal Code                                        177861 non-
null  float64
 5   Model Year                                         177866 non-
null  int64
 6   Make                                               177866 non-
null  object
 7   Model                                              177866 non-
null  object
 8   Electric Vehicle Type                              177866 non-
null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  177866 non-
null  object
 10  Electric Range                                     177866 non-
null  int64
 11  Base MSRP                                          177866 non-
null  int64
 12  Legislative District                               177477 non-
null  float64
 13  DOL Vehicle ID                                     177866 non-
null  int64
```

```
 14  Vehicle Location                                   177857 non-
null  object
 15  Electric Utility                                   177861 non-
null  object
 16  2020 Census Tract                                  177861 non-
null  float64
dtypes: float64(3), int64(4), object(10)
memory usage: 23.1+ MB

ev_data.isnull().sum()

VIN (1-10)                                             0
County                                                 5
City                                                   5
State                                                  0
Postal Code                                            5
Model Year                                             0
Make                                                   0
Model                                                  0
Electric Vehicle Type                                  0
Clean Alternative Fuel Vehicle (CAFV) Eligibility      0
Electric Range                                         0
Base MSRP                                              0
Legislative District                                 389
DOL Vehicle ID                                         0
Vehicle Location                                       9
Electric Utility                                       5
2020 Census Tract                                      5
dtype: int64

ev_data = ev_data.dropna()
```

Let's start with analyzing the EV Adoption Over Time by visualizing the number of EVs registered by model year. It will give us an insight into how the EV population has grown over the years:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("whitegrid")

# EV Adoption Over Time
plt.figure(figsize=(12, 6))
ev_adoption_by_year = ev_data['Model
Year'].value_counts().sort_index()
sns.barplot(x=ev_adoption_by_year.index, y=ev_adoption_by_year.values,
palette="viridis", hue=ev_adoption_by_year.index, legend=False)
plt.title('EV Adoption Over Time')
plt.xlabel('Model Year')
plt.ylabel('Number of Vehicles Registered')
plt.xticks(rotation=45)
```

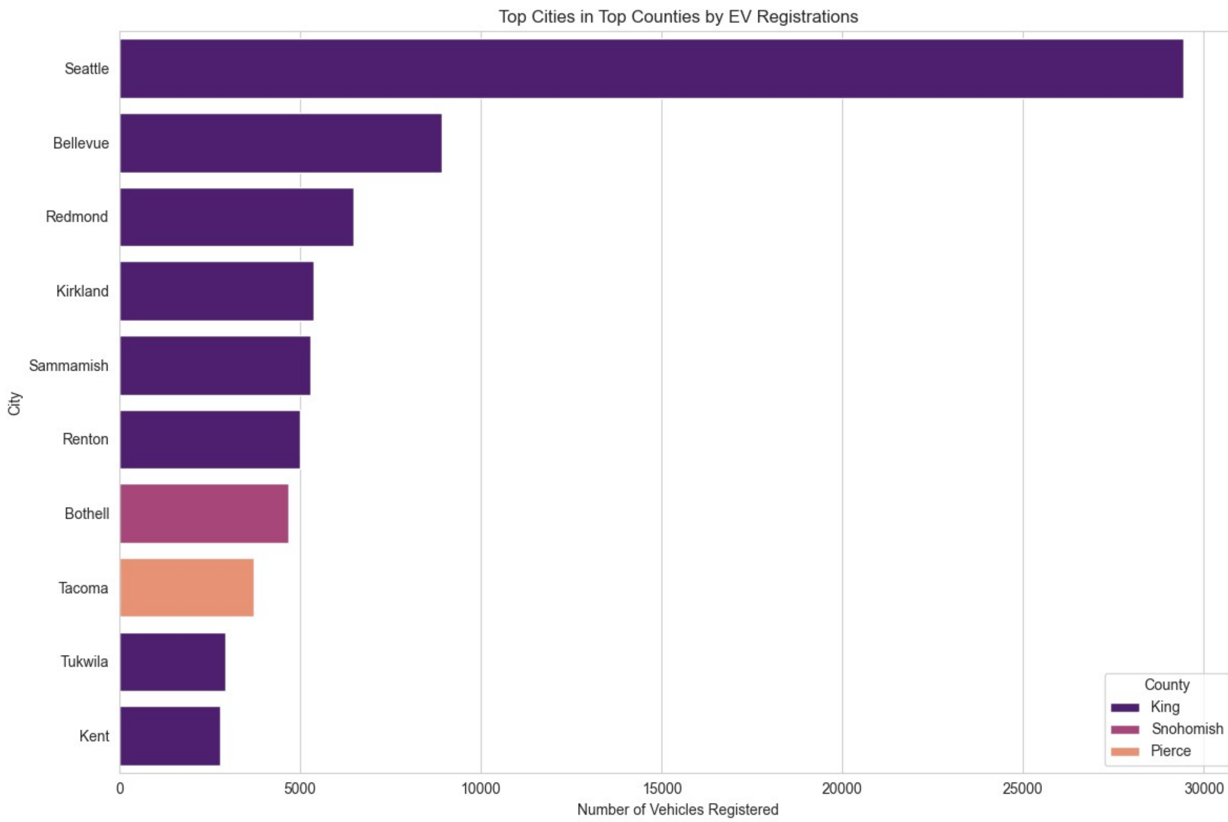
```
plt.tight_layout()
plt.show()
```



From the above bar chart, it’s clear that EV adoption has been increasing over time, especially noting a significant upward trend starting around 2016. The number of vehicles registered grows modestly up until that point and then begins to rise more rapidly from 2017 onwards. The year 2023 shows a particularly sharp increase in the number of registered EVs, with the bar for 2023 being the highest on the graph, indicating a peak in EV adoption.

```
# geographical distribution at county level
ev_county_distribution = ev_data['County'].value_counts()
top_counties = ev_county_distribution.head(3).index

# filtering the dataset for these top counties
top_counties_data = ev_data[ev_data['County'].isin(top_counties)]

# analyzing the distribution of EVs within the cities of these top
counties
ev_city_distribution_top_counties =
top_counties_data.groupby(['County',
'City']).size().sort_values(ascending=False).reset_index(name='Number
of Vehicles')

# visualize the top 10 cities across these counties
top_cities = ev_city_distribution_top_counties.head(10)

plt.figure(figsize=(12, 8))
sns.barplot(x='Number of Vehicles', y='City', hue='County',
data=top_cities, palette="magma")
plt.title('Top Cities in Top Counties by EV Registrations')
```

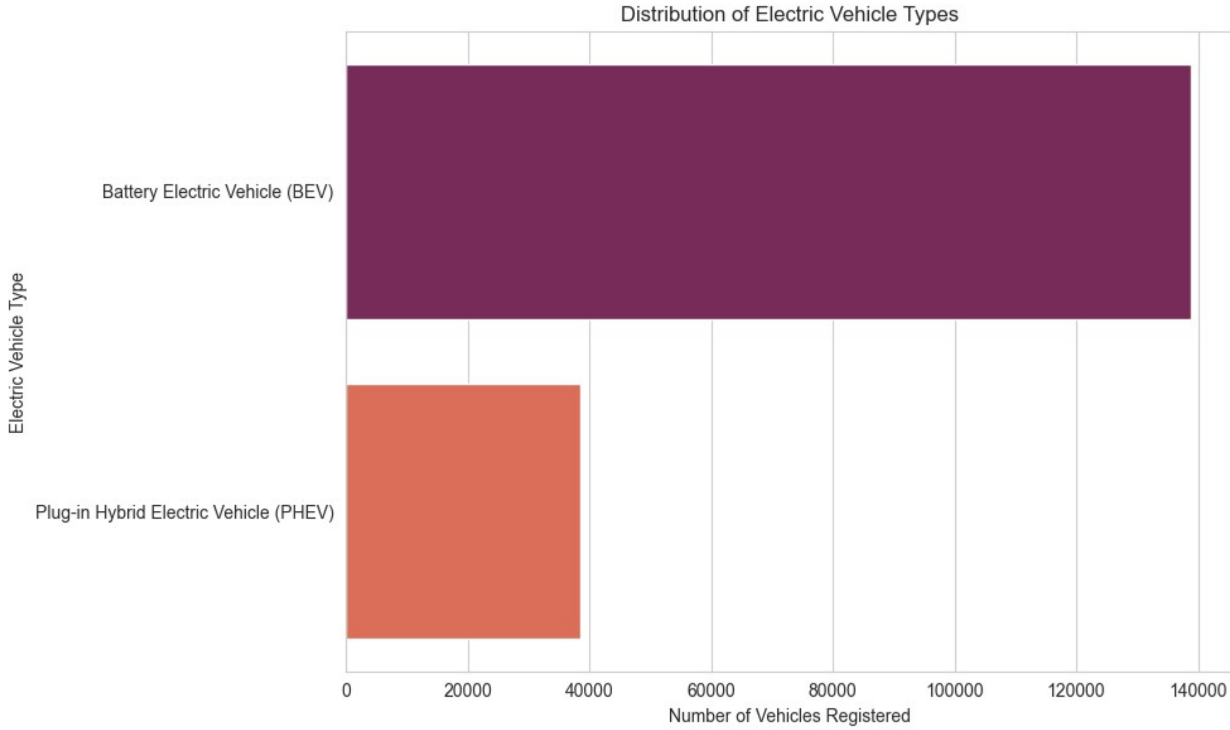
```
plt.xlabel('Number of Vehicles Registered')
plt.ylabel('City')
plt.legend(title='County')
plt.tight_layout()
plt.show()
```



Next, let’s explore the types of electric vehicles represented in this dataset. Understanding the breakdown between different EV types, such as Battery Electric Vehicles (BEV) and Plug-in Hybrid Electric Vehicles (PHEV), can provide insights into consumer preferences and the adoption patterns of purely electric vs. hybrid electric solutions. So, let’s visualize the distribution of electric vehicle types to see which categories are most popular among the registered vehicles:

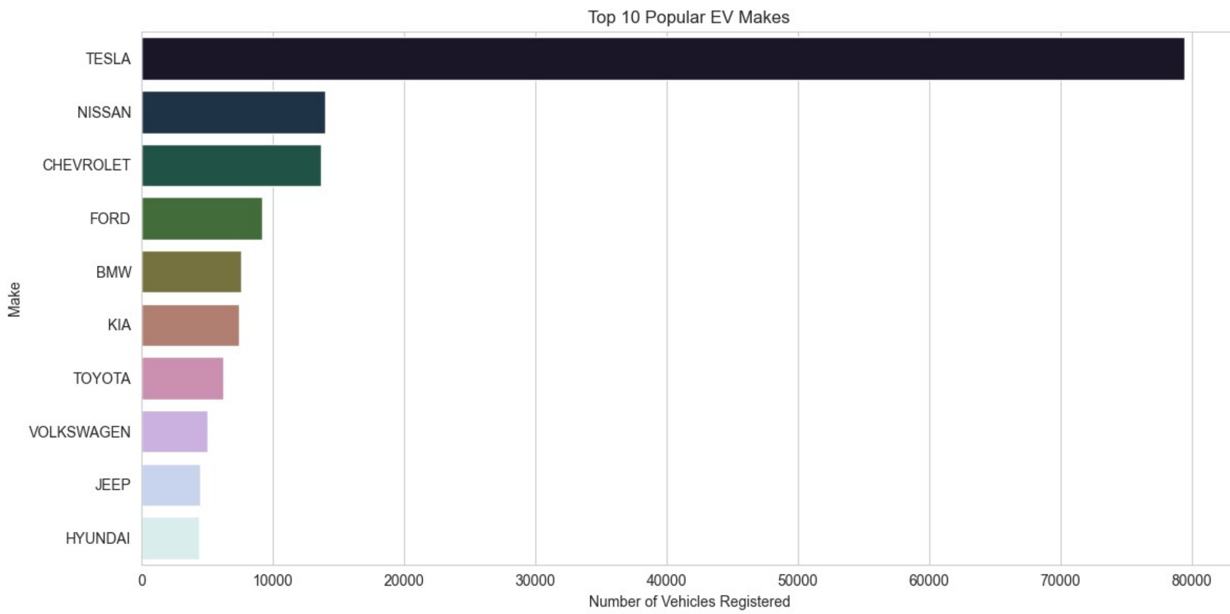
```
# analyzing the distribution of electric vehicle Types
ev_type_distribution = ev_data['Electric Vehicle Type'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=ev_type_distribution.values,
y=ev_type_distribution.index, palette="rocket",
hue=ev_type_distribution.index, legend=False)
plt.title('Distribution of Electric Vehicle Types')
plt.xlabel('Number of Vehicles Registered')
plt.ylabel('Electric Vehicle Type')
```

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



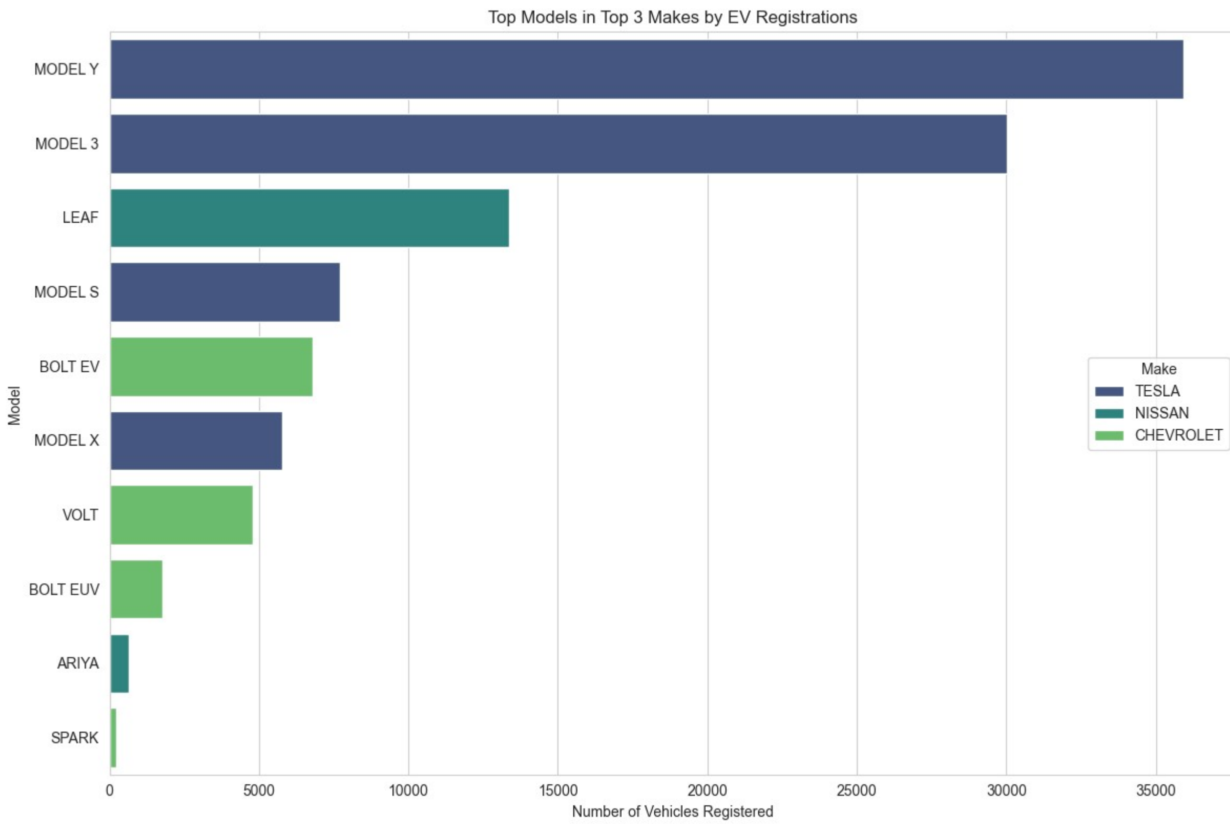
So, let's have a look at the most popular manufacturers and then drill down into the most popular models within those manufacturers:

```
# analyzing the popularity of EV manufacturers
ev_make_distribution = ev_data['Make'].value_counts().head(10)  # Limiting to top 10 for clarity

plt.figure(figsize=(12, 6))
sns.barplot(x=ev_make_distribution.values, y=ev_make_distribution.index, palette="cubehelix", hue=ev_make_distribution.index, legend=False)
plt.title('Top 10 Popular EV Makes')
plt.xlabel('Number of Vehicles Registered')
plt.ylabel('Make')
plt.tight_layout()
plt.show()
```



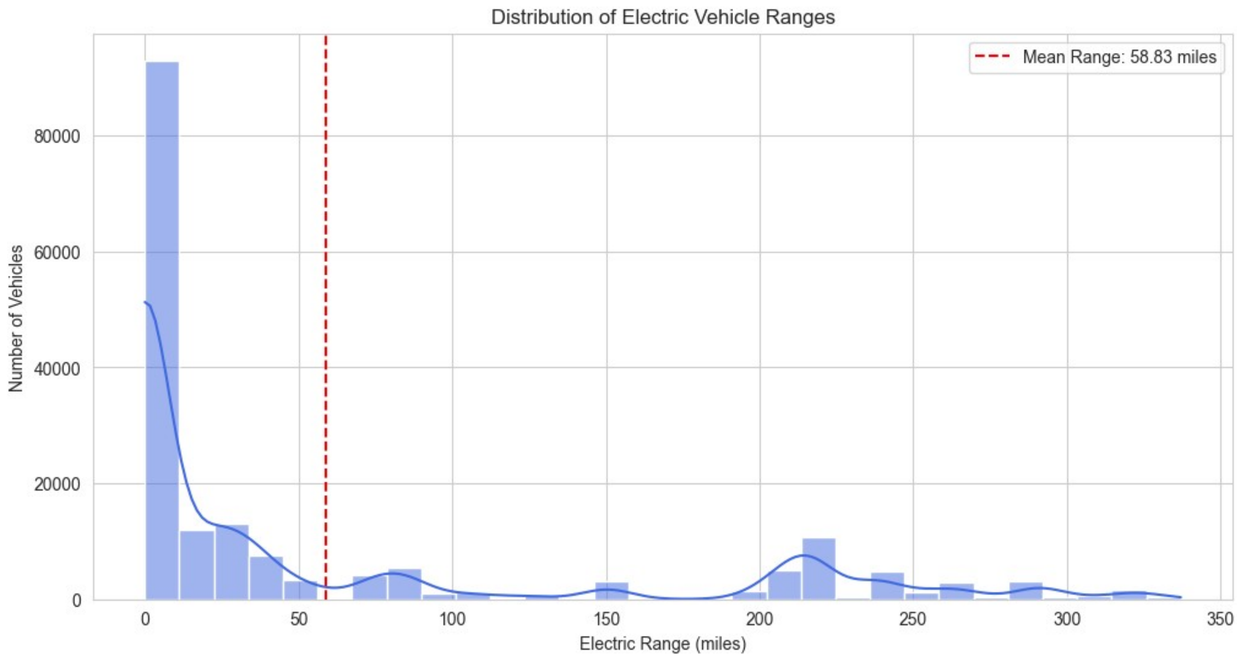
Next, let's drill down into the most popular models within these top manufacturers to get a more detailed understanding of consumer preferences at the model level:

```
# selecting the top 3 manufacturers based on the number of vehicles
registered
top_3_makes = ev_make_distribution.head(3).index

# filtering the dataset for these top manufacturers
top_makes_data = ev_data[ev_data['Make'].isin(top_3_makes)]

# analyzing the popularity of EV models within these top manufacturers
ev_model_distribution_top_makes = top_makes_data.groupby(['Make',
'Model']).size().sort_values(ascending=False).reset_index(name='Number
of Vehicles')

# visualizing the top 10 models across these manufacturers for clarity
top_models = ev_model_distribution_top_makes.head(10)

plt.figure(figsize=(12, 8))
sns.barplot(x='Number of Vehicles', y='Model', hue='Make',
data=top_models, palette="viridis")
plt.title('Top Models in Top 3 Makes by EV Registrations')
plt.xlabel('Number of Vehicles Registered')
plt.ylabel('Model')
plt.legend(title='Make', loc='center right')
plt.tight_layout()
plt.show()
```



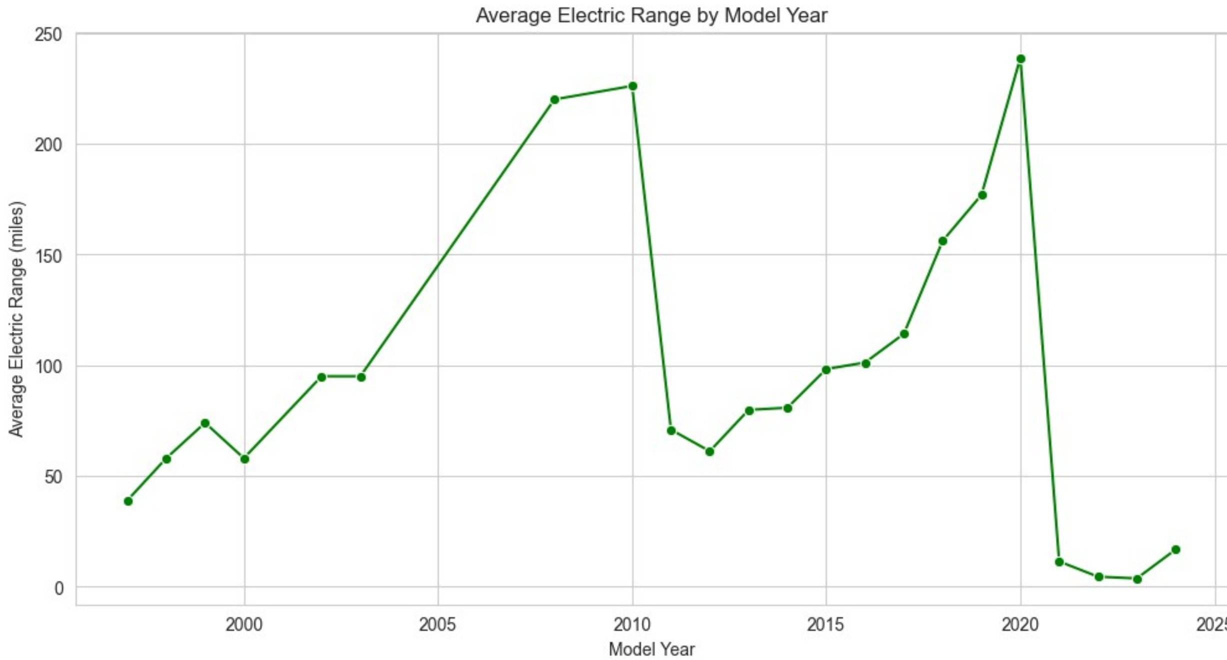
Next, we'll explore the electric range of vehicles, which is a critical factor for analyzing the market size of electric vehicles. The electric range indicates how far an EV can travel on a single charge, and advancements in battery technology have been steadily increasing these ranges over the years. So, let's look at the distribution of electric ranges in the dataset and identify any notable trends, such as improvements over time or variations between different vehicle types or manufacturers:

```
# analyzing the distribution of electric range
plt.figure(figsize=(12, 6))
sns.histplot(ev_data['Electric Range'], bins=30, kde=True,
color='royalblue')
plt.title('Distribution of Electric Vehicle Ranges')
plt.xlabel('Electric Range (miles)')
plt.ylabel('Number of Vehicles')
plt.axvline(ev_data['Electric Range'].mean(), color='red',
linestyle='--', label=f'Mean Range: {ev_data["Electric
Range"].mean():.2f} miles')
plt.legend()
plt.show()
```



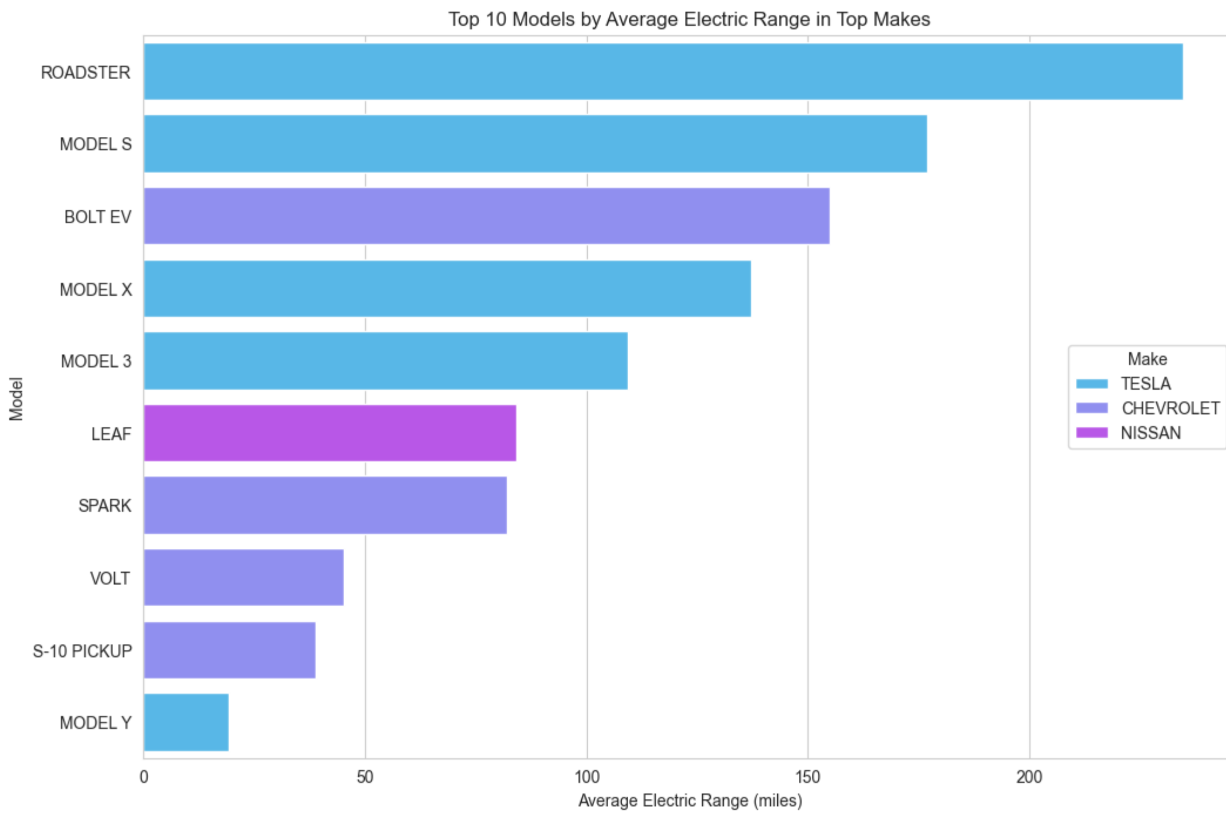
Now, let's delve into the trend of electric ranges over model years, which can provide insights into how advancements in battery technology and vehicle design have influenced the electric range capabilities of electric vehicles over time. A positive trend in this analysis would indicate continuous improvements, offering consumers EVs with longer driving ranges and potentially addressing one of the major concerns regarding the EV market (range anxiety):

```
# calculating the average electric range by model year
average_range_by_year = ev_data.groupby('Model Year')['Electric
Range'].mean().reset_index()

plt.figure(figsize=(12, 6))
sns.lineplot(x='Model Year', y='Electric Range',
data=average_range_by_year, marker='o', color='green')
plt.title('Average Electric Range by Model Year')
plt.xlabel('Model Year')
plt.ylabel('Average Electric Range (miles)')
plt.grid(True)
plt.show()
```



Next, let's explore how electric ranges vary among the top manufacturers and models. This analysis can reveal how different manufacturers are addressing the crucial aspect of electric range and highlight which models stand out for their superior range capabilities:

```
average_range_by_model = top_makes_data.groupby(['Make', 'Model'])
['Electric Range'].mean().sort_values(ascending=False).reset_index()

# the top 10 models with the highest average electric range
top_range_models = average_range_by_model.head(10)

plt.figure(figsize=(12, 8))
barplot = sns.barplot(x='Electric Range', y='Model', hue='Make',
data=top_range_models, palette="cool")
plt.title('Top 10 Models by Average Electric Range in Top Makes')
plt.xlabel('Average Electric Range (miles)')
plt.ylabel('Model')
plt.legend(title='Make', loc='center right')
plt.show()
```



#Estimated Market Size Analysis of Electric Vehicles in the United States

Now, let's move forward towards finding the estimated market size of electric vehicles in the United States. I'll first count the number of EVs registered every year:

```
# calculate the number of EVs registered each year
ev_registration_counts = ev_data['Model
Year'].value_counts().sort_index()
ev_registration_counts

Model Year
1997        1
1998        1
1999        5
2000        7
2002        2
2003        1
2008       19
2010       23
2011      775
2012     1614
2013     4399
2014     3496
2015     4826
2016     5469
```

```
2017     8534
2018    14286
2019    10913
2020    11740
2021    19063
2022    27708
2023    57519
2024     7072
Name: count, dtype: int64
```

We'll calculate the Compound Annual Growth Rate (CAGR) between a recent year with complete data (2023) and an earlier year to project the 2024 figures. Additionally, using this growth rate, we can estimate the market size for the next five years. Let's proceed with these calculations:

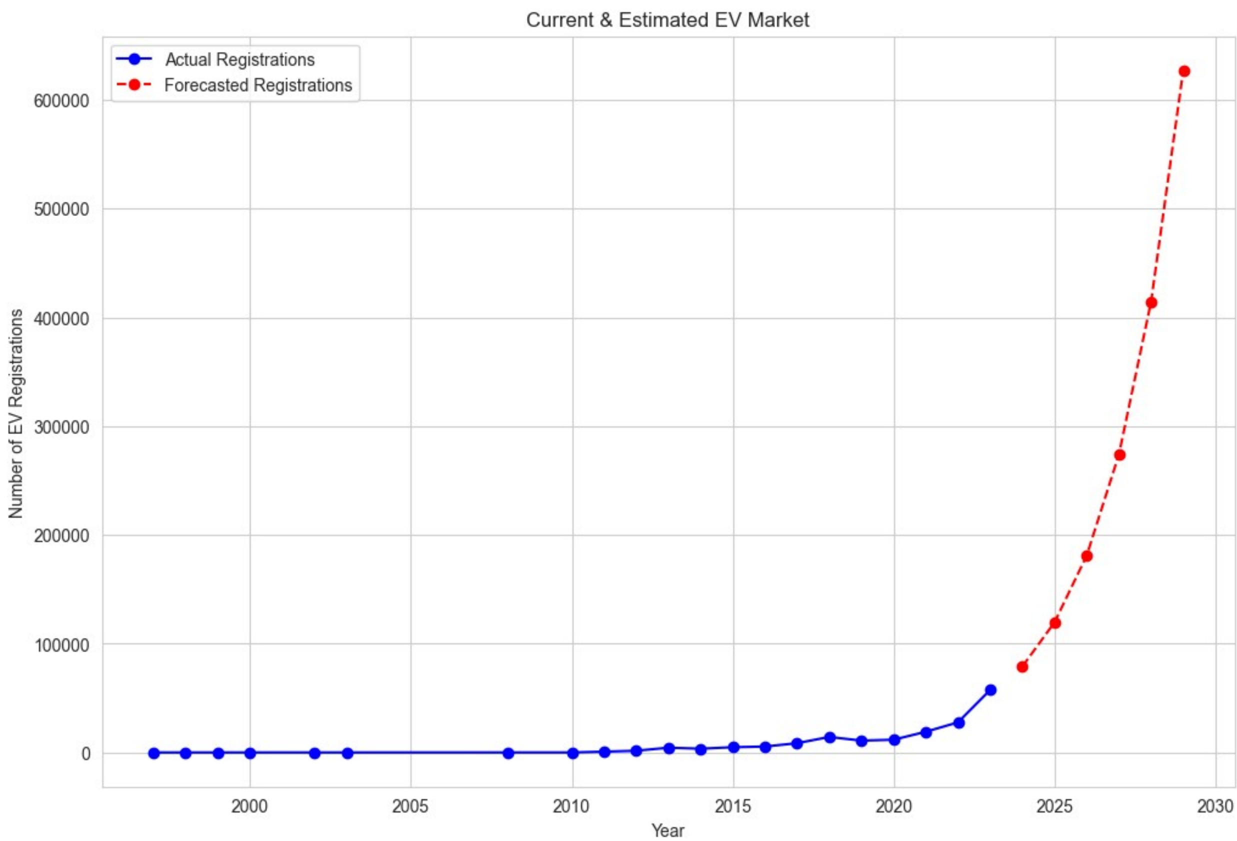
```
from scipy.optimize import curve_fit
import numpy as np

# filter the dataset to include years with complete data, assuming
2023 is the last complete year
filtered_years = ev_registration_counts[ev_registration_counts.index
<= 2023]

# define a function for exponential growth to fit the data
def exp_growth(x, a, b):
    return a * np.exp(b * x)

# prepare the data for curve fitting
x_data = filtered_years.index - filtered_years.index.min()
y_data = filtered_years.values

# fit the data to the exponential growth function
params, covariance = curve_fit(exp_growth, x_data, y_data)

# use the fitted function to forecast the number of EVs for 2024 and
the next five years
forecast_years = np.arange(2024, 2024 + 6) -
filtered_years.index.min()
forecasted_values = exp_growth(forecast_years, *params)

# create a dictionary to display the forecasted values for easier
interpretation
forecasted_evs = dict(zip(forecast_years + filtered_years.index.min(),
forecasted_values))
forecasted_evs
```

```
{np.int64(2024): np.float64(79079.20808938889),
 np.int64(2025): np.float64(119653.96274428742),
 np.int64(2026): np.float64(181047.22020265696),
 np.int64(2027): np.float64(273940.74706208805),
```

```
 np.int64(2028): np.float64(414497.01805382164),
 np.int64(2029): np.float64(627171.3128407666)}
```

Now, let's plot the estimated market size data:

```
# prepare data for plotting
years = np.arange(filtered_years.index.min(), 2029 + 1)
actual_years = filtered_years.index
forecast_years_full = np.arange(2024, 2029 + 1)

# actual and forecasted values
actual_values = filtered_years.values
forecasted_values_full = [forecasted_evs[year] for year in
forecast_years_full]

plt.figure(figsize=(12, 8))
plt.plot(actual_years, actual_values, 'bo-', label='Actual
Registrations')
plt.plot(forecast_years_full, forecasted_values_full, 'ro--',
label='Forecasted Registrations')

plt.title('Current & Estimated EV Market')
plt.xlabel('Year')
plt.ylabel('Number of EV Registrations')
plt.legend()
plt.grid(True)

plt.show()
```



Current & Estimated EV Market
Actual Registrations
Forecasted Registrations
Number of EV Registrations
600000
500000
400000
300000
200000
100000
0
2000
2005
2010
2015
2020
2025
2030
Year