Electric Vehicles Market Size Analysis

Market size analysis for electric vehicles involves a multi-step process that includes defining the market scope, collecting and preparing data, analytical modelling, and communicating findings through visualization and reporting.

Below is the process you can follow for the task of electric vehicles market size analysis:

1. Define whether the analysis is global, regional, or focused on specific countries.
2. Gather information from industry associations, market research firms (e.g., BloombergNEF, IEA), and government publications relevant to the EV market.
3. Use historical data to identify trends in EV sales, production, and market.
4. Analyze the market size and growth rates for different EV segments.
5. Based on the market size analysis, provide strategic recommendations for businesses looking to enter or expand in the EV market.

Importing Files

In [1]:
```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  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  3FA6P0SU1K  Snohomish   Everett    WA        98201        2019       FORD

       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)

   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

   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

              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
```

Cleaning the dataset

In [2]:
```python
ev_data.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
```

In [3]:
```python
ev_data.isnull().sum()
```

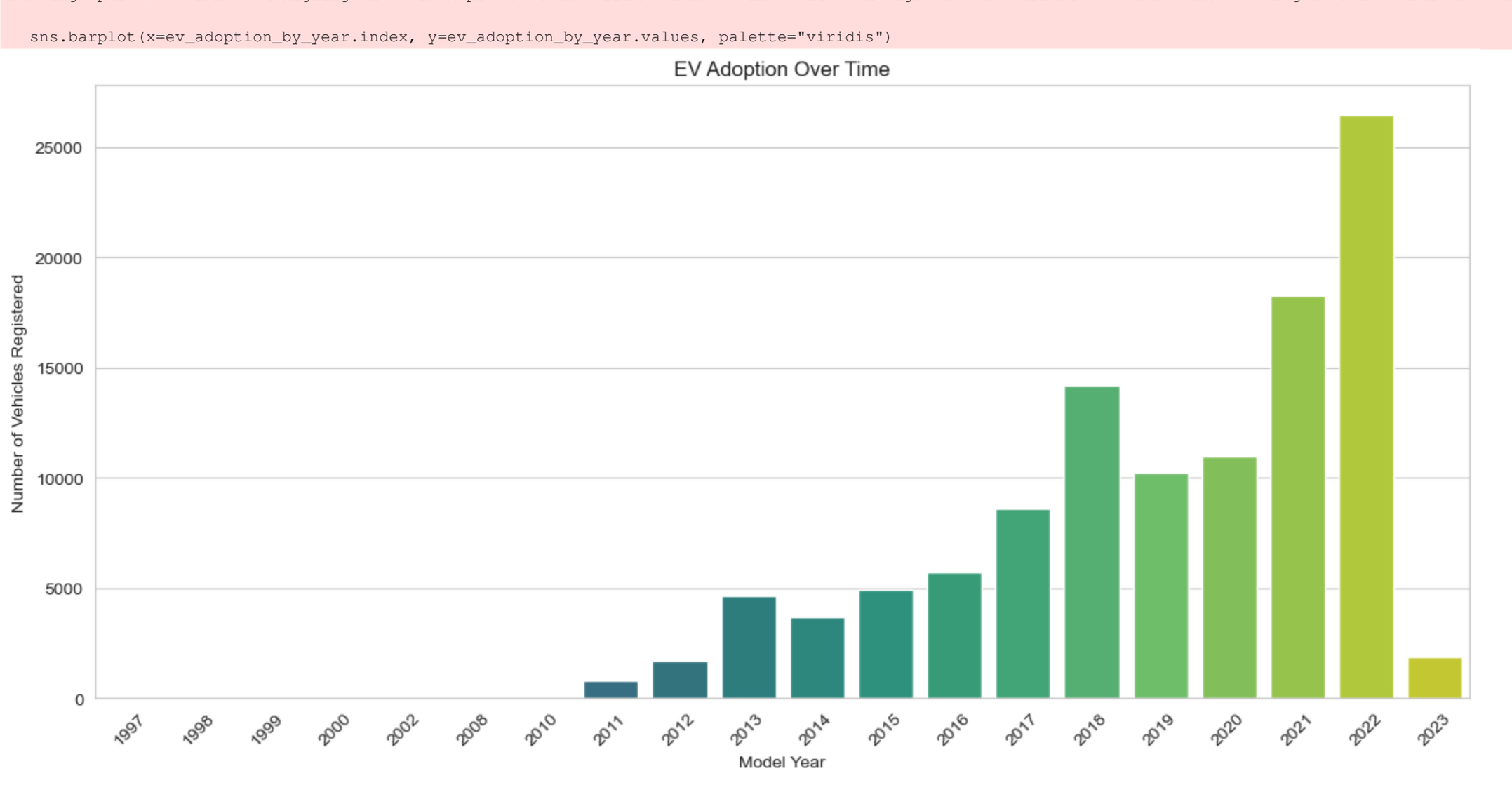
Out[3]:
```
VIN (1-10)                                             0
County                                                 0
City                                                   0
State                                                  0
Postal Code                                            0
Model 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
```

In [4]:
```python
ev_data = ev_data.dropna()
```

EV Adoption/Growth

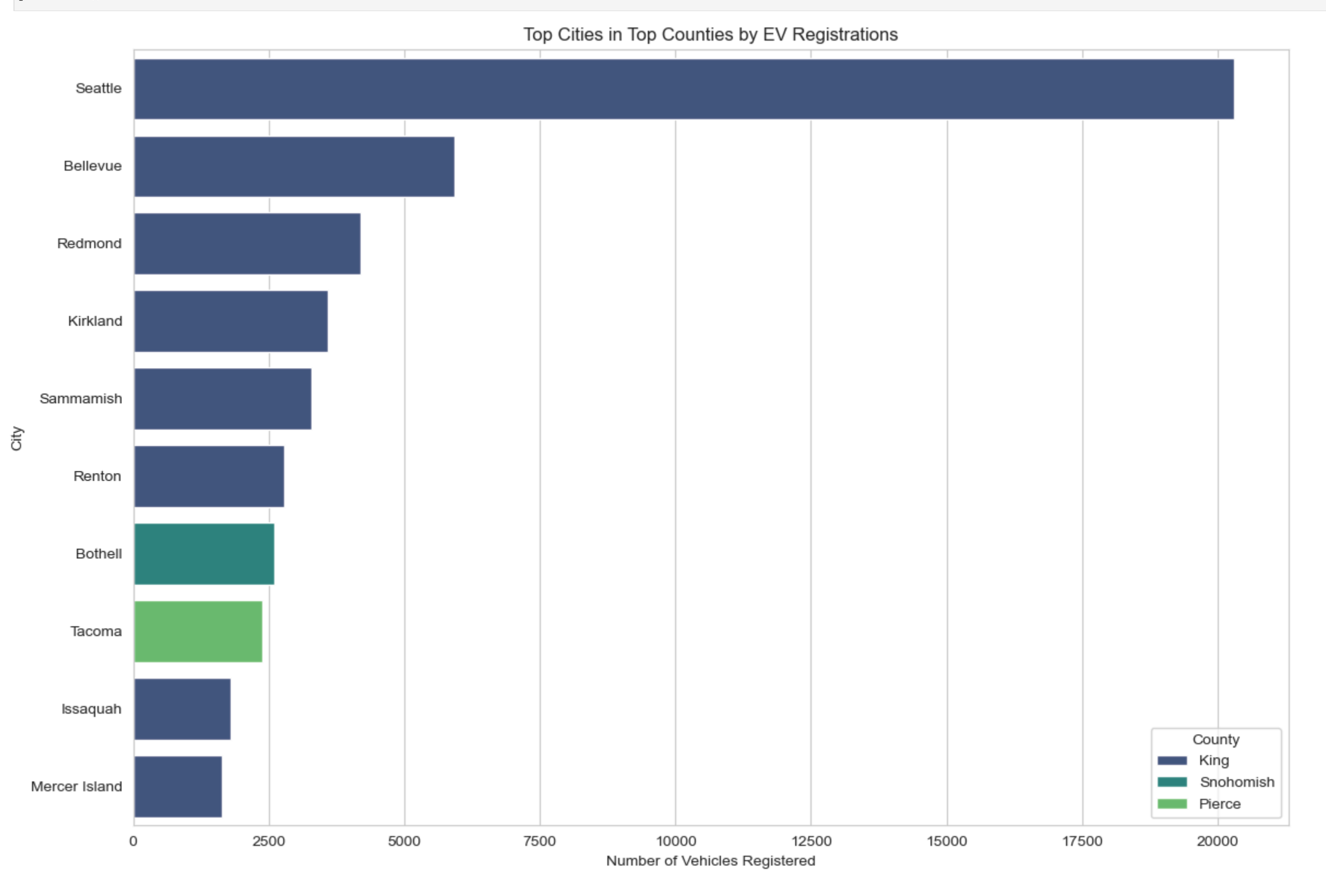
In [5]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("whitegrid")

# 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")
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()
```

```
C:\Users\Nalini Mudgil\AppData\Local\Temp\ipykernel_13108\3186072121.py:8: 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.

  sns.barplot(x=ev_adoption_by_year.index, y=ev_adoption_by_year.values, palette="viridis")
```



Registration of EVs in top 3 countries

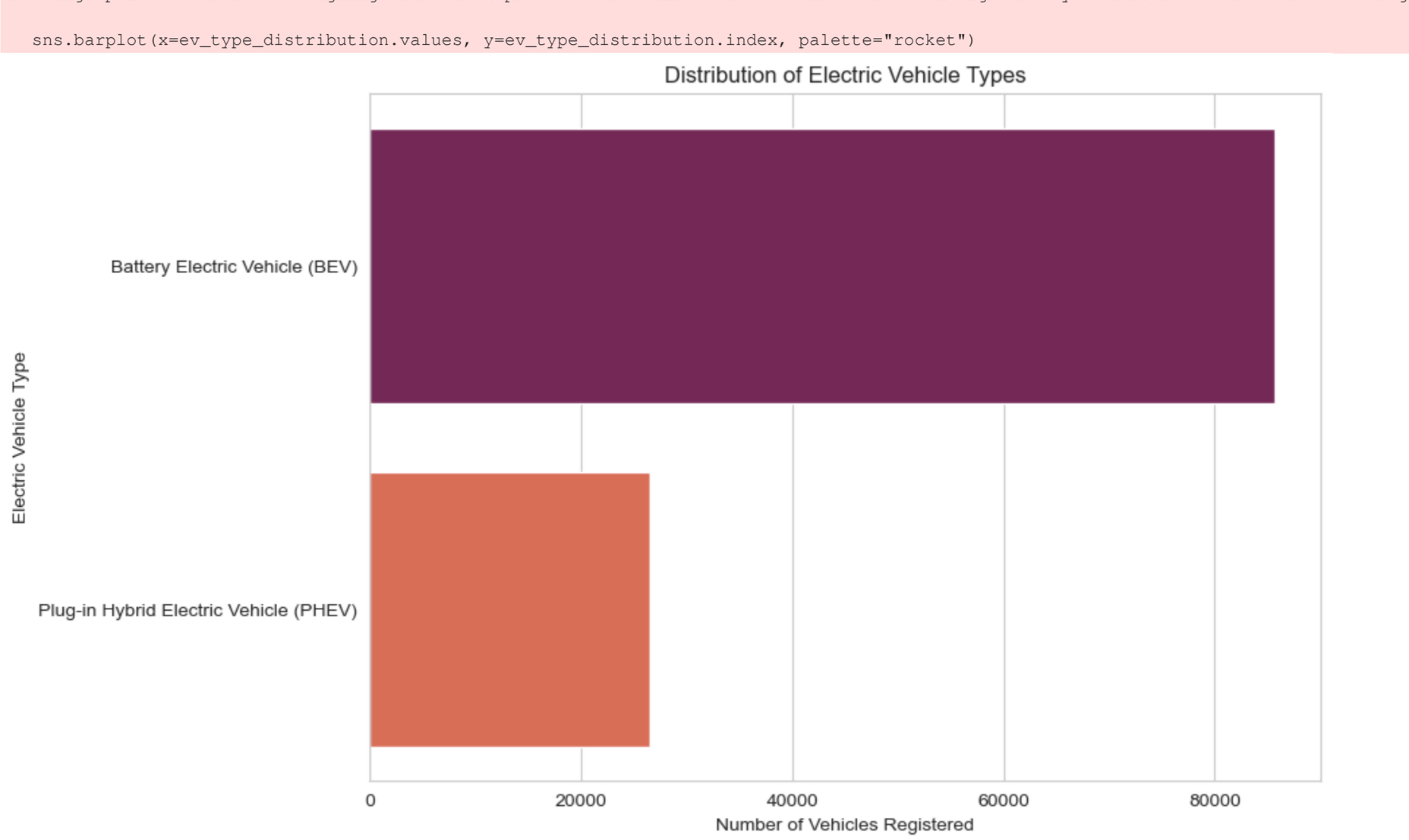
In [6]:
```python
# geographical distribution at county level
ev_county_distribution = ev_data['County'].value_counts()
top_counties = ev_county_distribution.head(3).index

# 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="viridis")
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()
```



Categories of EVs Registered

In [7]:
```python
# 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")
plt.title('Distribution of Electric Vehicle Types')
plt.xlabel('Number of Vehicles Registered')
plt.ylabel('Electric Vehicle Type')
plt.tight_layout()
plt.show()
```

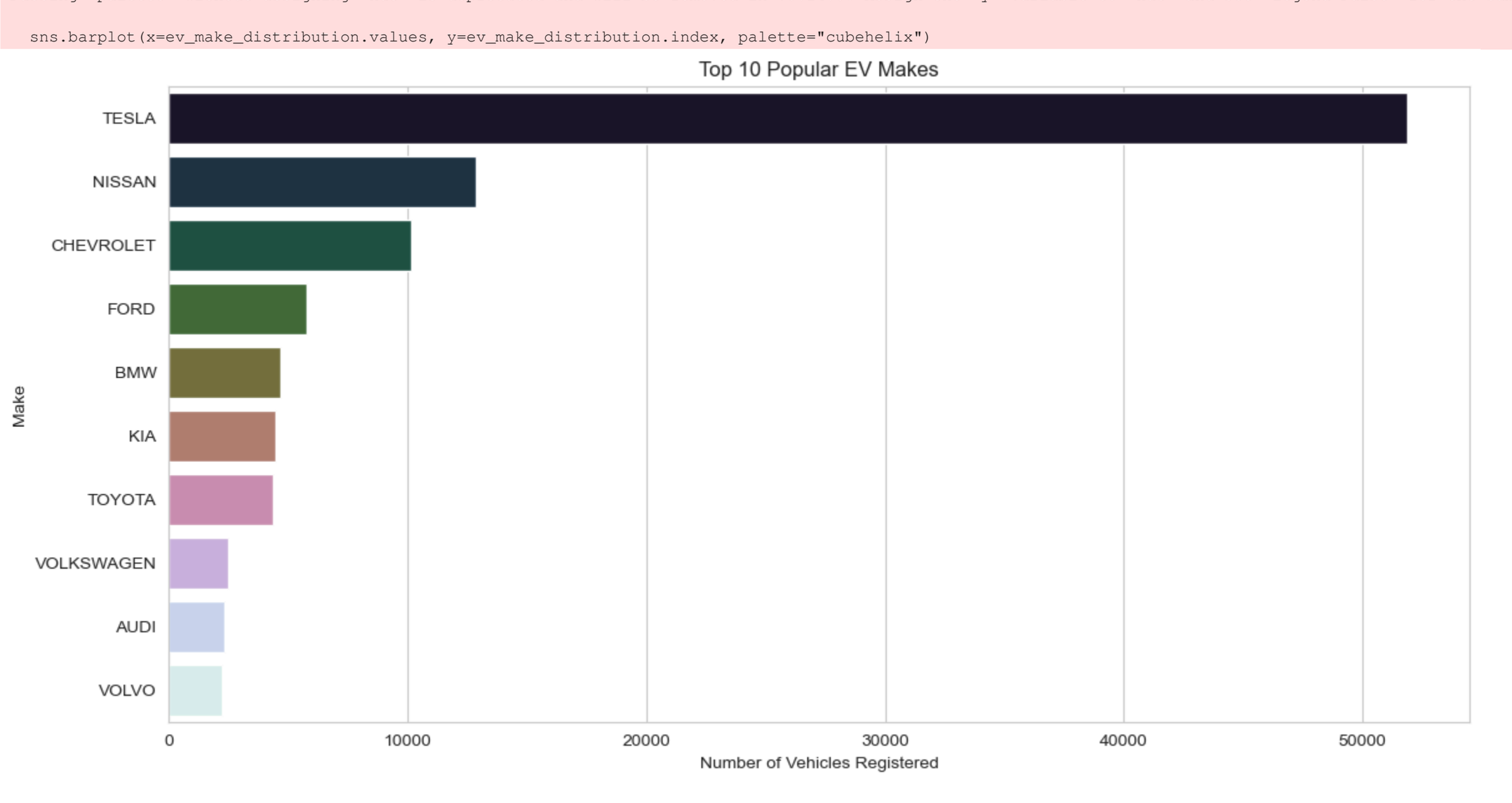
```
C:\Users\Nalini Mudgil\AppData\Local\Temp\ipykernel_13108\2107807950.py:5: FutureWarning:

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

  sns.barplot(x=ev_type_distribution.values, y=ev_type_distribution.index, palette="rocket")
```



Most Popular Manufacturers

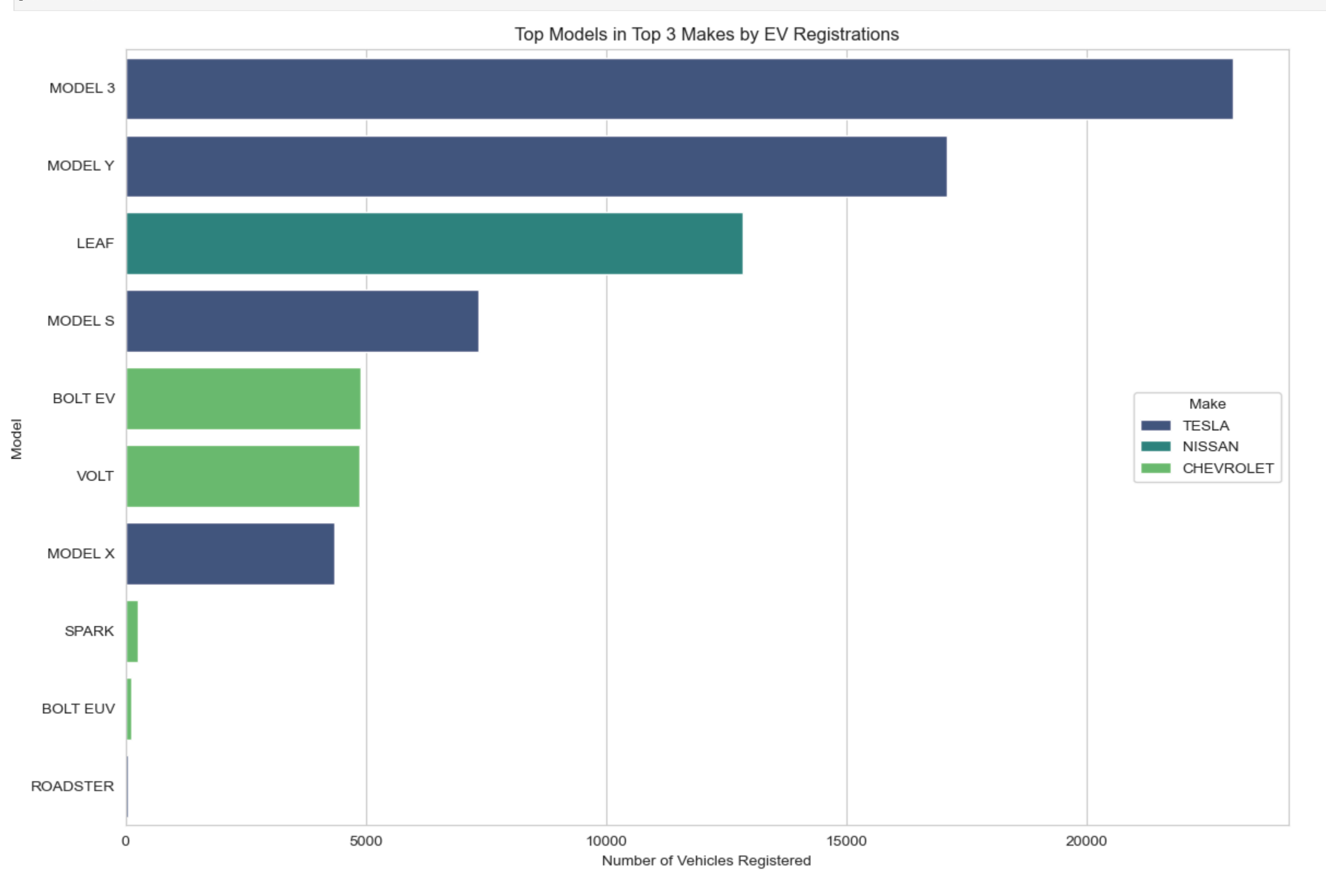
In [8]:
```python
# analyzing the popularity of EV manufacturers
ev_make_distribution = ev_data['Make'].value_counts().head(10)  # Limiting to top 10 for clarity

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

```
C:\Users\Nalini Mudgil\AppData\Local\Temp\ipykernel_13108\4123994712.py:5: FutureWarning:

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

  sns.barplot(x=ev_make_distribution.values, y=ev_make_distribution.index, palette="cubehelix")
```



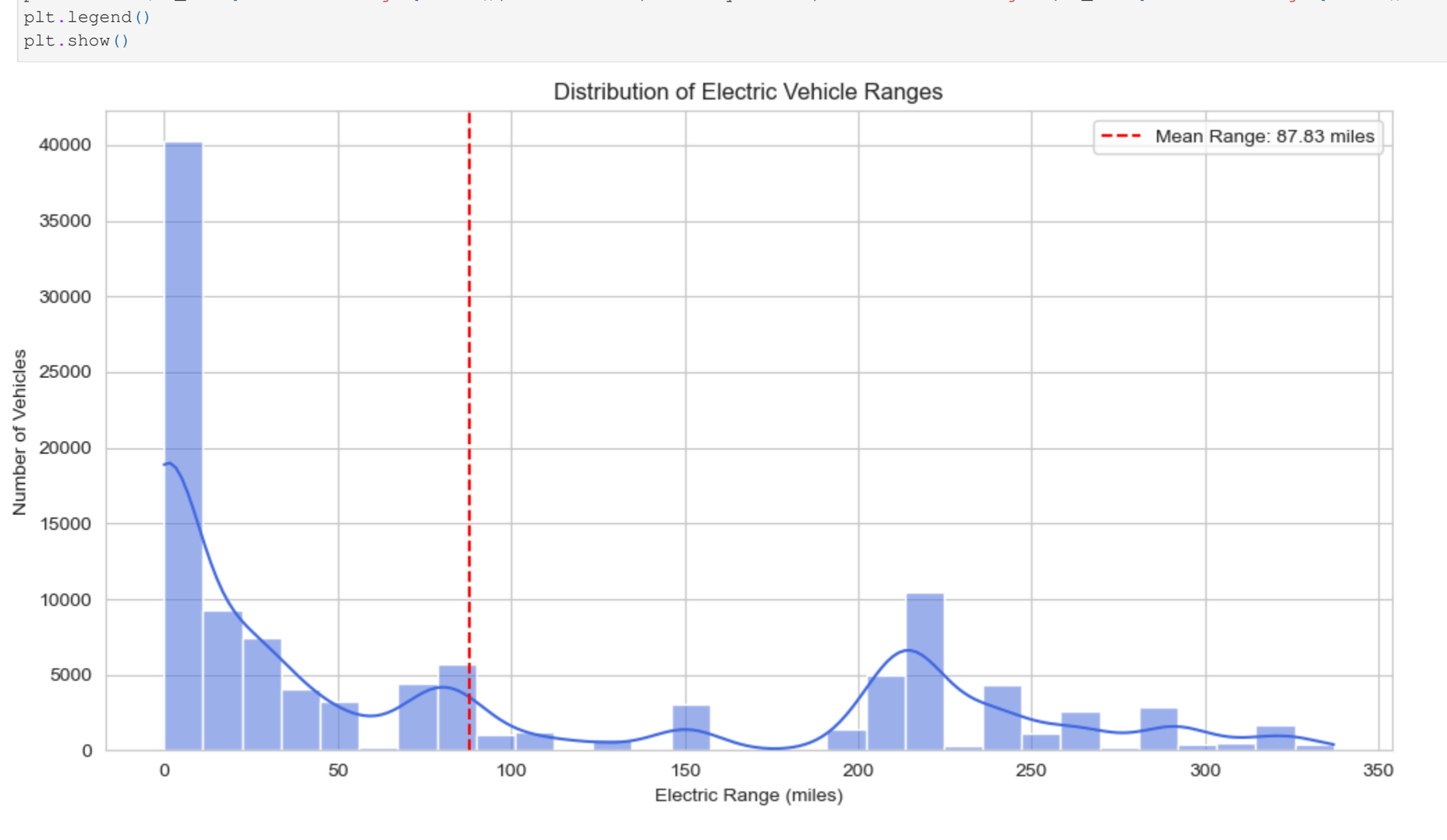
Consumer Preference based on models and manufacturers

In [9]:
```python
# 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()
```



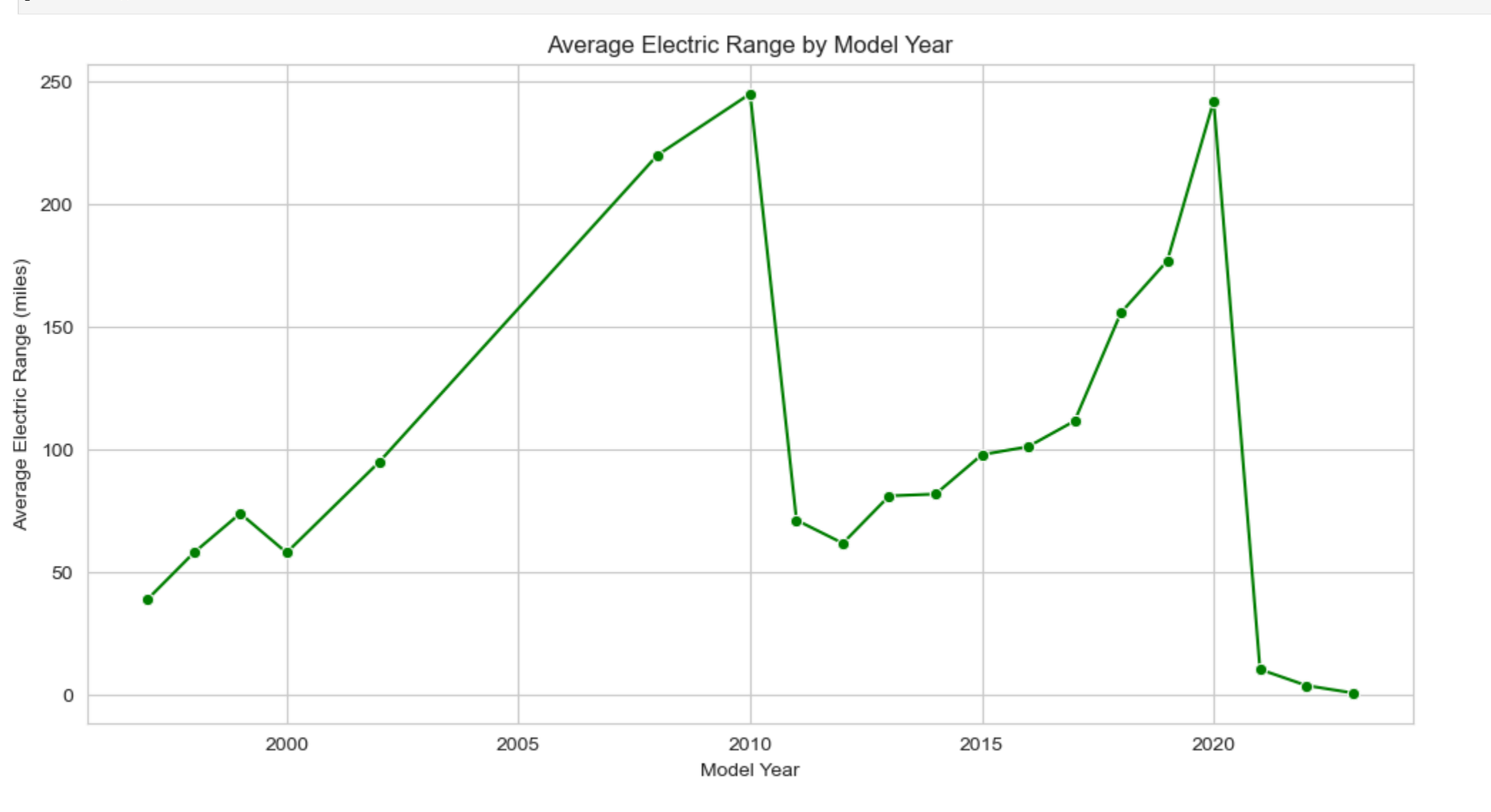
Electric Range Of Vehicles

In [10]:
```python
# 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()
```



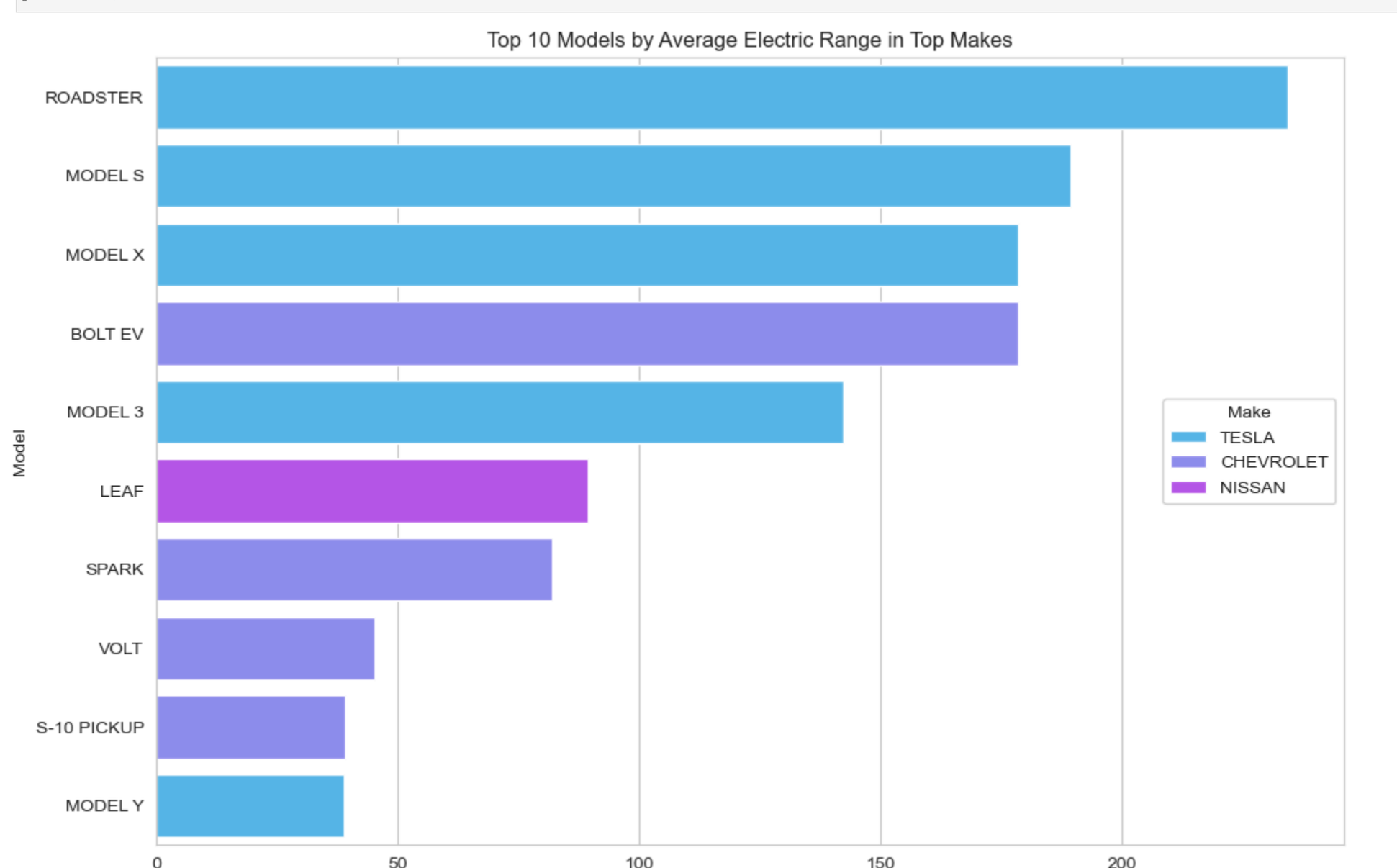
Electric Range based on the year of the car model

In [11]:
```python
# 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()
```



Variation in electric range based on the manufacturer of the model

In [12]:
```python
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

In [19]:
```python
# calculate the number of EVs registered each year
ev_registration_counts = ev_data['Model Year'].value_counts().sort_index()
ev_registration_counts
```

Out[19]:
```
Model Year
1997        1
1998        1
1999        3
2000       10
2002        2
2008       23
2010       24
2011      835
2012     1695
2013     4669
2014     3665
2015     4918
2016     5709
2017     8598
2018    14190
2019    10216
2020    10998
2021    18277
2022    26455
2023     1863
Name: count, dtype: int64
```

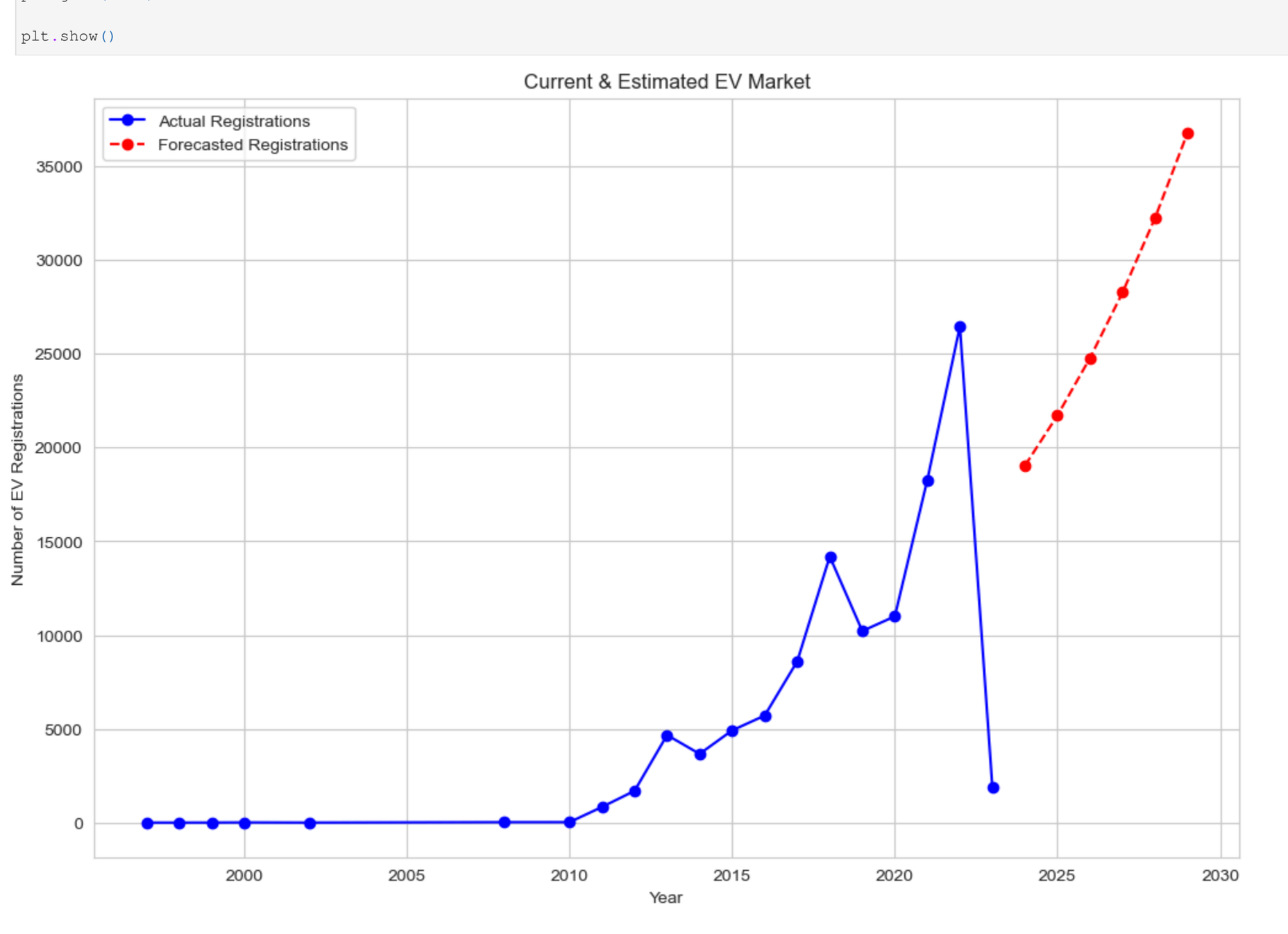
Annual Growth Rate

In [14]:
```python
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))

print(forecasted_evs)
```

```
{2024: 19034.481802860133, 2025: 21716.031459965205, 2026: 24775.353868542807, 2027: 28265.669095346875, 2028: 32247.69477145872, 2029: 36790.702338065574}
```

Plot for the Estimated Market Size

In [15]:
```python
# 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()
```



Summary

So, market size analysis is a crucial aspect of market research that determines the potential sales volume within a given market. It helps businesses understand the magnitude of demand, assess market saturation levels, and identify growth opportunities. From our market size analysis of electric vehicles, we found a promising future for the EV industry, indicating a significant shift in consumer preferences and a potential increase in related investment and business opportunities.