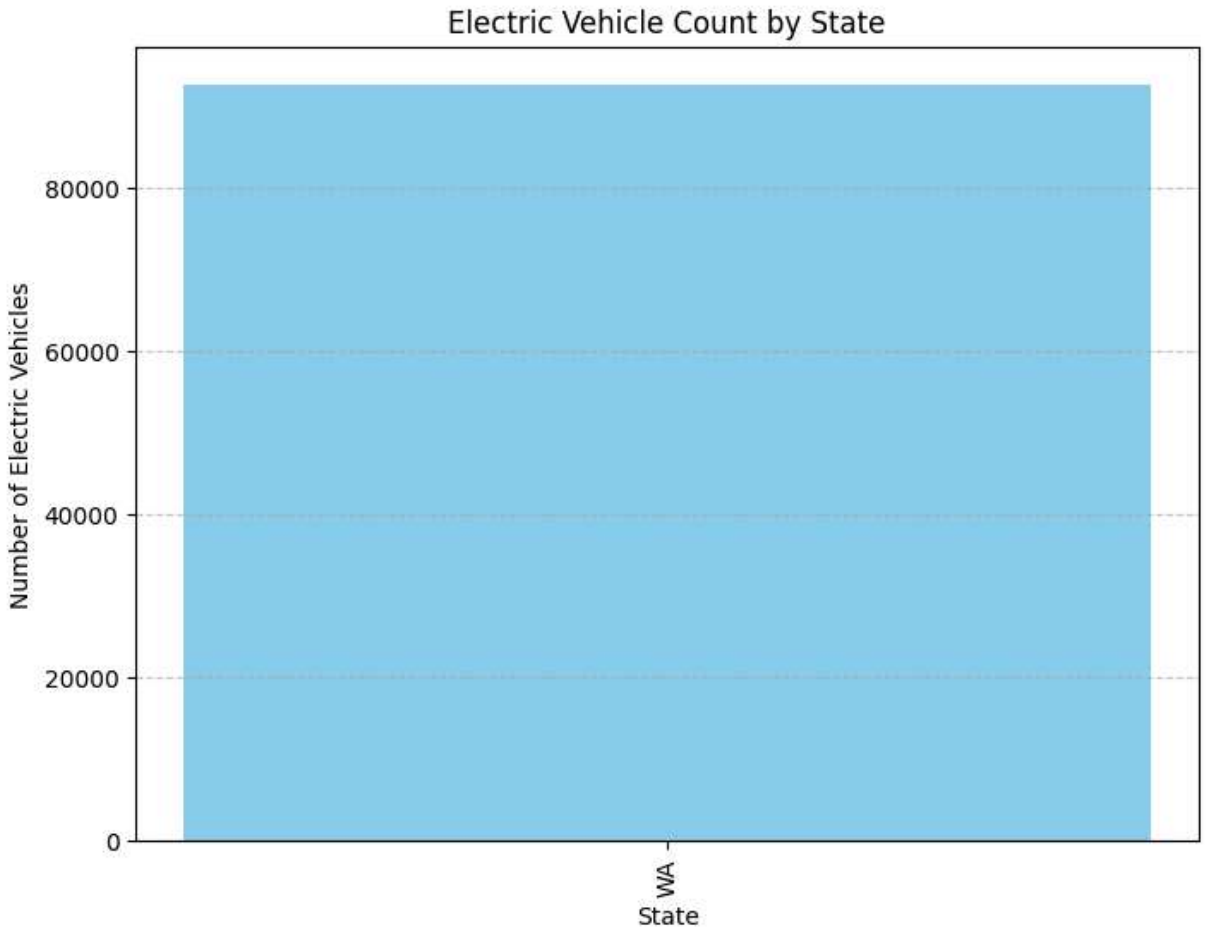
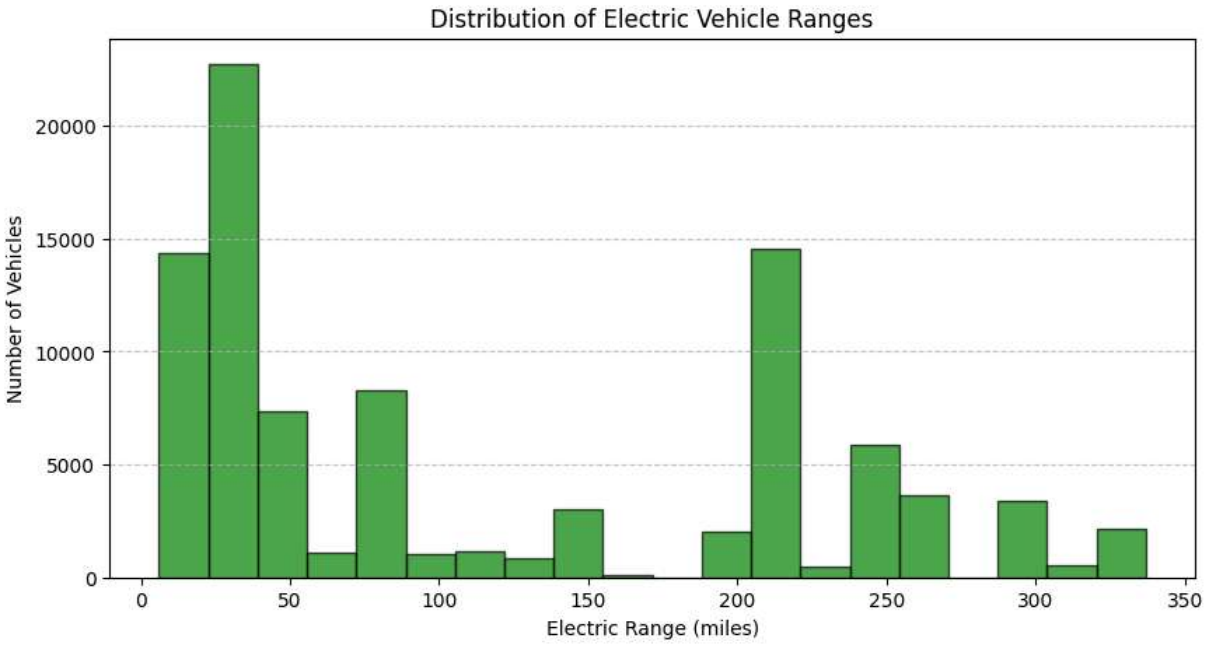
In [20]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('C://Users//satye//Downloads//EV_population.csv')
pd.set_option("display.width", 1000)
# Display first 5 rows of the dataset
print("Data:")
print(df.head())
```

```
Data:
  State  Model Year    Make    Electric Vehicle Type  Electric Range  Base MSRP  Leg
islative District CAFV Eligibility Simple
0   WA      2020         TESLA           BEV                266           0
46.0                 Eligible
1   WA      2024           BMW          PHEV                 39           0
46.0                 Eligible
2   WA      2024           BMW          PHEV                 39           0
43.0                 Eligible
3   WA      2018         TESLA           BEV                215           0
1.0                 Eligible
4   WA      2012     CHEVROLET          PHEV                 35           0
35.0                 Eligible
```

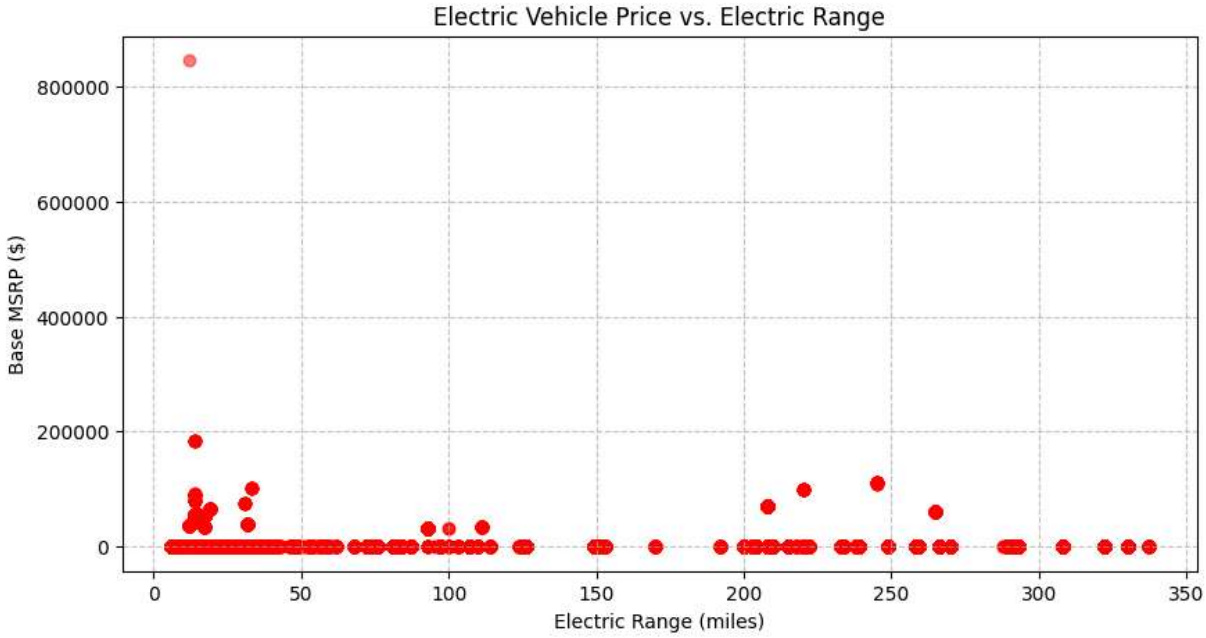
In [15]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Load CSV
df = pd.read_csv('C://Users//satye//Downloads//EV_population.csv')

# Count EVs/state
state_counts = df["State"].value_counts()

# Convert to NumPy arrays
states = np.array(state_counts.index)
ev_counts = np.array(state_counts.values)

# Plot Bar Chartz
plt.figure(figsize=(8, 6))
plt.bar(states, ev_counts, color='skyblue')
plt.xticks(rotation=90)
plt.xlabel("State")
plt.ylabel("Number of Electric Vehicles")
plt.title("Electric Vehicle Count by State")
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



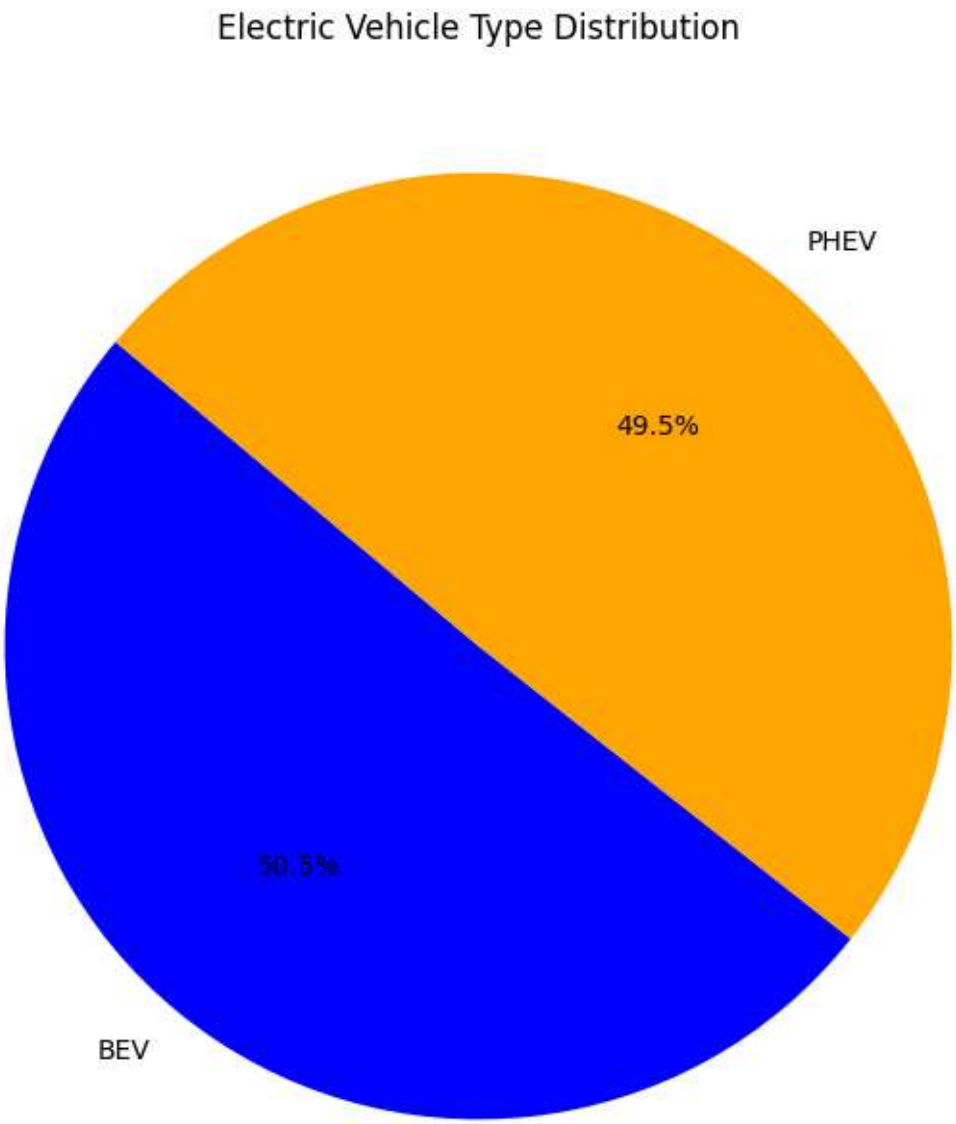
In [16]:
```
ev_range = df["Electric Range"].dropna().to_numpy()

# Plotting Histogram
plt.figure(figsize=(10, 5))
plt.hist(ev_range, bins=20, color='green', alpha=0.7, edgecolor="black")
plt.xlabel("Electric Range (miles)")
plt.ylabel("Number of Vehicles")
plt.title("Distribution of Electric Vehicle Ranges")
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



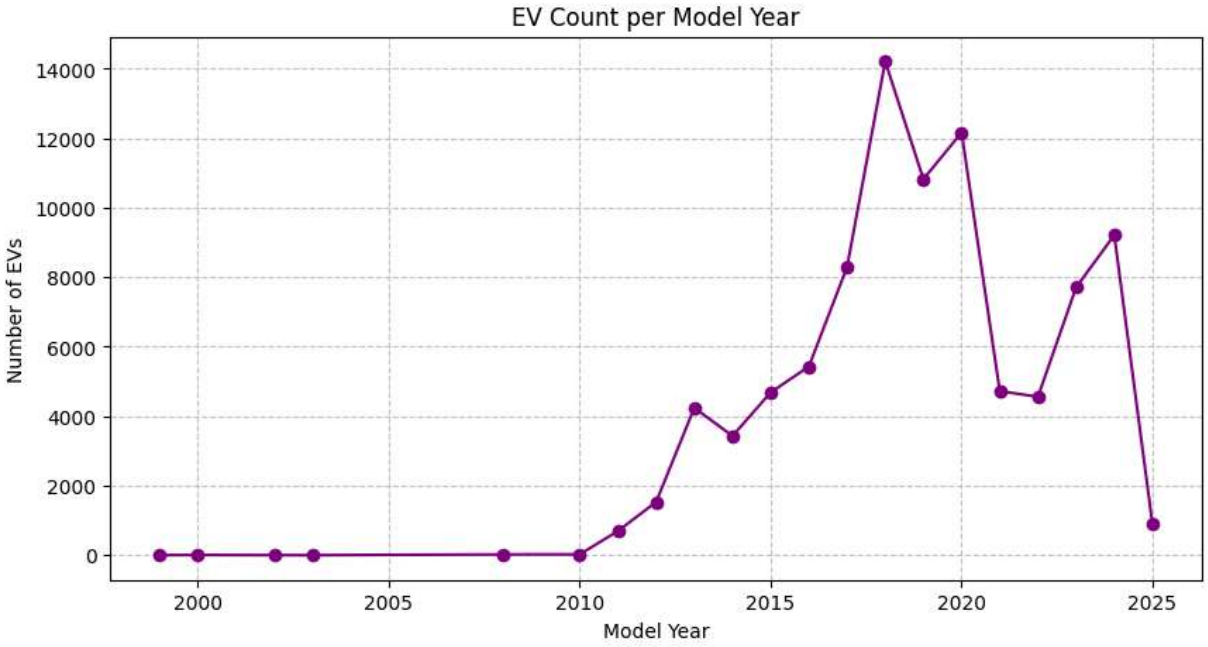
In [17]:
```
# Drop rows with missing values in 'Base MSRP' and 'Electric Range'
df_filtered = df.dropna(subset=["Base MSRP", "Electric Range"])

# Convert to NumPy arrays
msrp = df_filtered["Base MSRP"].to_numpy()
ev_range = df_filtered["Electric Range"].to_numpy()

# Scatter Plot
plt.figure(figsize=(10, 5))
plt.scatter(ev_range, msrp, color='red', alpha=0.5)
plt.xlabel("Electric Range (miles)")
plt.ylabel("Base MSRP ($)")
plt.title("Electric Vehicle Price vs. Electric Range")
plt.grid(linestyle='--', alpha=0.7)

plt.show()
```



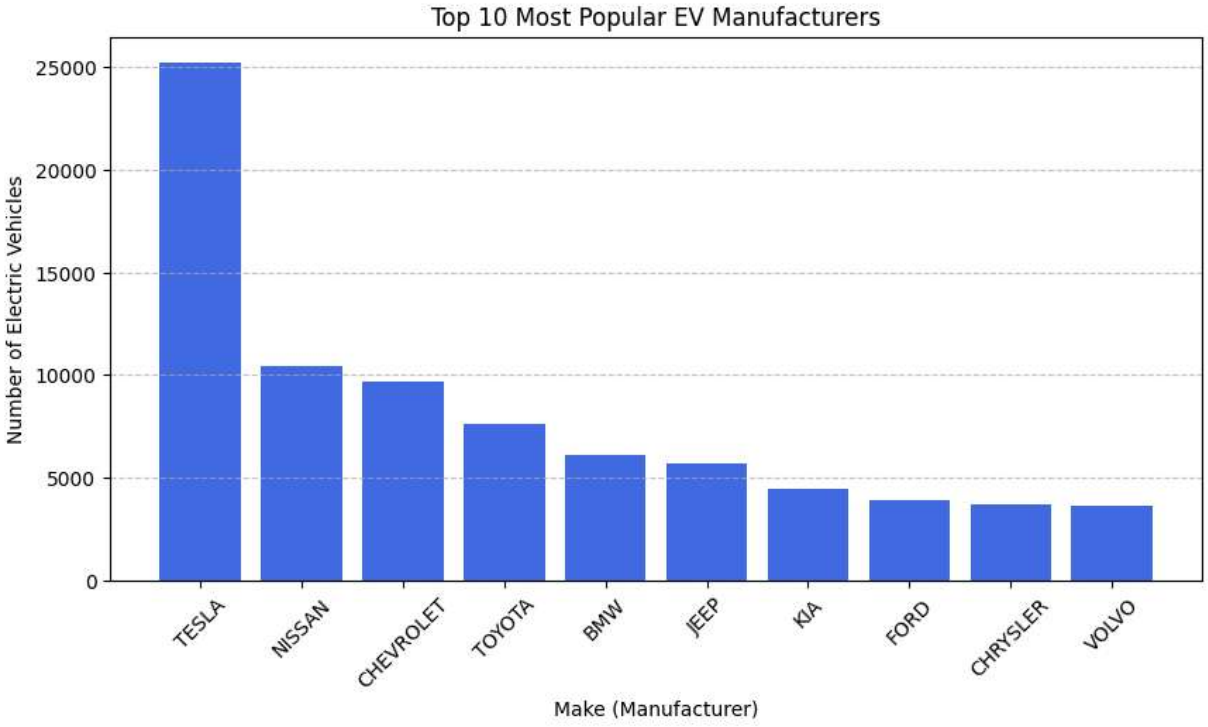
In [18]:
```
# Count EV Types
ev_type_counts = df["Electric Vehicle Type"].value_counts()

# Convert to NumPy arrays
ev_types = np.array(ev_type_counts.index)
ev_counts = np.array(ev_type_counts.values)

# Plot Pie Chart
plt.figure(figsize=(8, 8))
plt.pie(ev_counts, labels=ev_types, autopct="%1.1f%%", colors=["blue", "orange", "g
plt.title("Electric Vehicle Type Distribution")

plt.show()
```



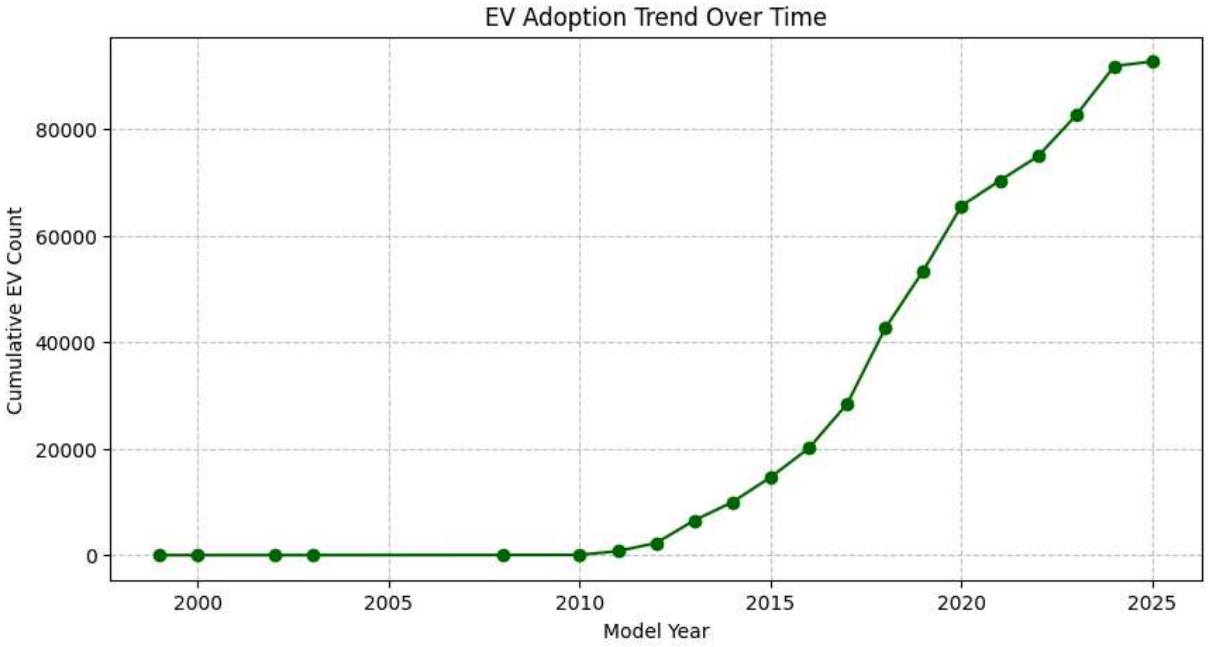
In [19]:
```
# Count EVs per Model Year
model_year_counts = df["Model Year"].value_counts().sort_index()

# Convert to NumPy arrays
years = np.array(model_year_counts.index)
ev_counts = np.array(model_year_counts.values)

# Plot Line Chart
plt.figure(figsize=(10, 5))
plt.plot(years, ev_counts, marker='o', linestyle='-', color='purple')
plt.xlabel("Model Year")
plt.ylabel("Number of EVs")
plt.title("EV Count per Model Year")
plt.grid(linestyle='--', alpha=0.7)

plt.show()
```



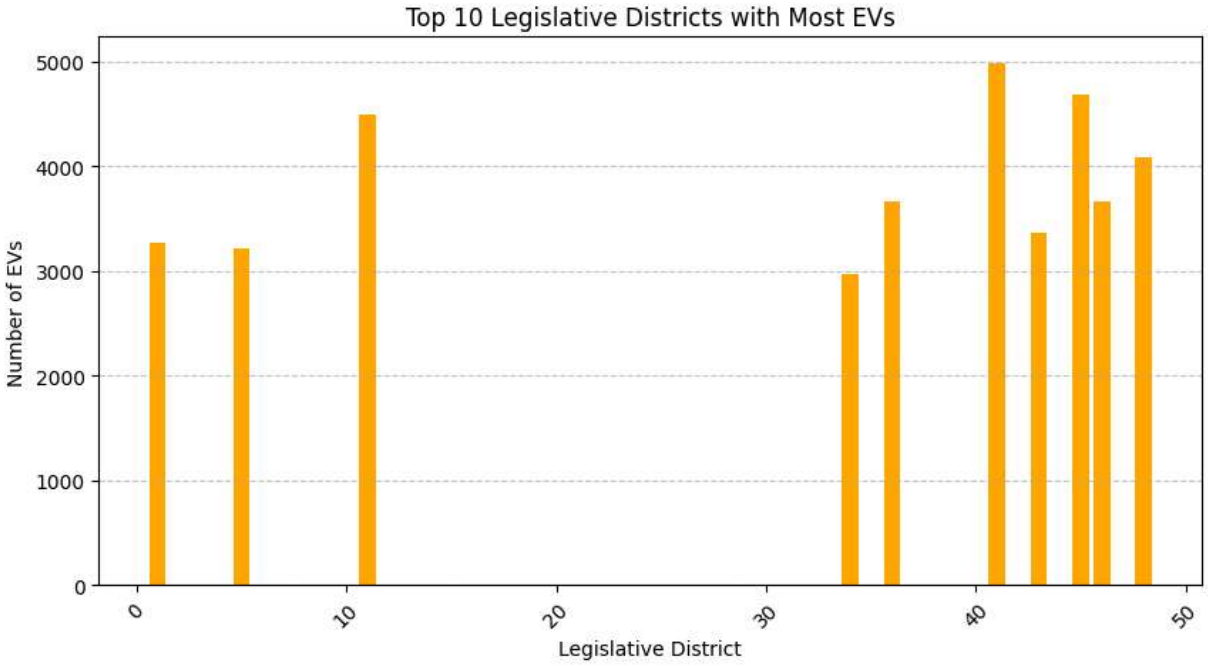
In [20]:
```
# Count EVs per Make
make_counts = df["Make"].value_counts().head(10)

# Convert to NumPy arrays
makes = np.array(make_counts.index)
ev_counts = np.array(make_counts.values)

# Plot Bar Chart
plt.figure(figsize=(10, 5))
plt.bar(makes, ev_counts, color='royalblue')
plt.xticks(rotation=45)
plt.xlabel("Make (Manufacturer)")
plt.ylabel("Number of Electric Vehicles")
plt.title("Top 10 Most Popular EV Manufacturers")
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



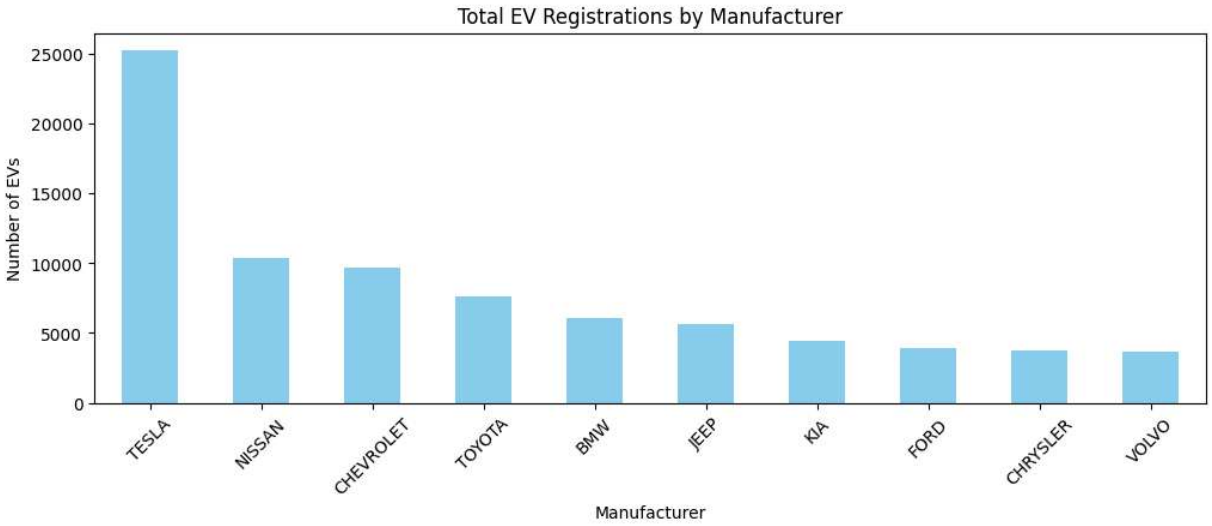
In [21]:
```
# Count EVs per Model Year and sort
model_year_counts = df["Model Year"].value_counts().sort_index()

# Convert to NumPy arrays
years = np.array(model_year_counts.index)
ev_counts = np.array(model_year_counts.values)

# Compute cumulative sum for trend
cumulative_ev_counts = np.cumsum(ev_counts)

# Plot Cumulative Line Chart
plt.figure(figsize=(10, 5))
plt.plot(years, cumulative_ev_counts, marker='o', linestyle='-', color='darkgreen')
plt.xlabel("Model Year")
plt.ylabel("Cumulative EV Count")
plt.title("EV Adoption Trend Over Time")
plt.grid(linestyle='--', alpha=0.7)

plt.show()
```



In [25]:
```
# Count EVs per Legislative District
district_counts = df["Legislative District"].value_counts().head(10)

# Convert to NumPy arrays
districts = np.array(district_counts.index)
ev_counts = np.array(district_counts.values)

# Plot Bar Chart
plt.figure(figsize=(10, 5))
plt.bar(districts, ev_counts, color='orange')
plt.xticks(rotation=45)
plt.xlabel("Legislative District")
plt.ylabel("Number of EVs")
plt.title("Top 10 Legislative Districts with Most EVs")
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

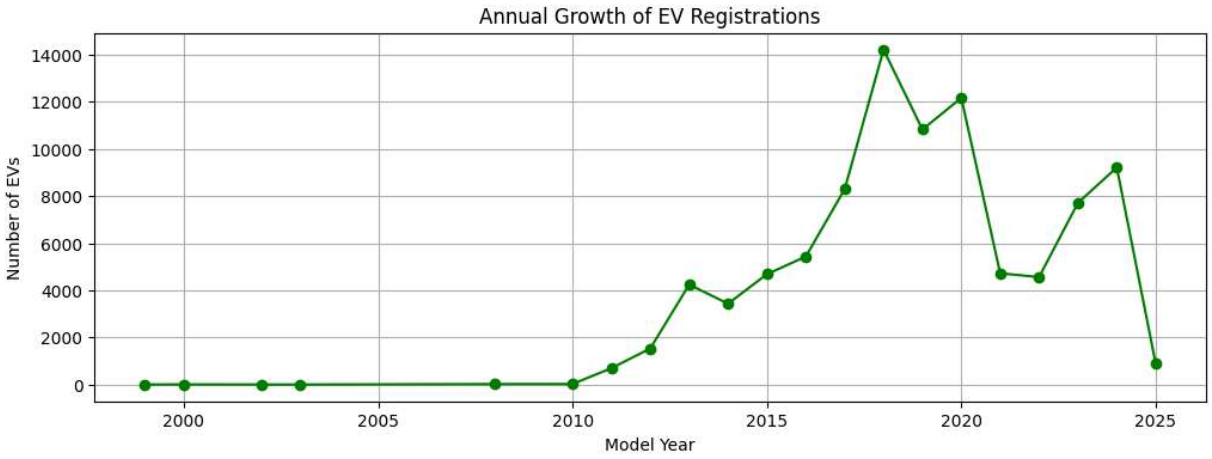


In [31]:
```
# Count the number of EVs per manufacturer
ev_count = df["Make"].value_counts().head(10)

# Plot bar chart
plt.figure(figsize=(12, 4))
ev_count.plot(kind="bar", color="skyblue")
plt.xlabel("Manufacturer")
plt.ylabel("Number of EVs")
plt.title("Total EV Registrations by Manufacturer")
plt.xticks(rotation=45)
plt.show()
```

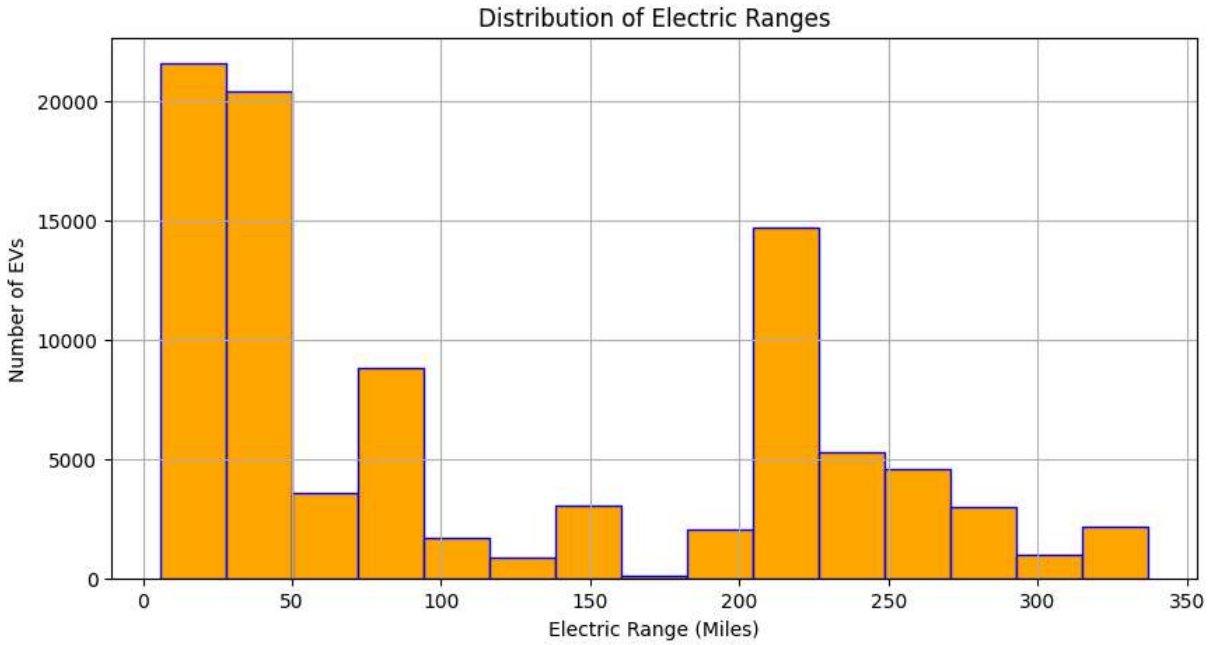


In [30]:
```
# Count EVs per model year
yearly_ev_count = df["Model Year"].value_counts().sort_index()

# Plot line chart
plt.figure(figsize=(12, 4))
plt.plot(yearly_ev_count.index, yearly_ev_count.values, marker='o', linestyle='-',
```

```
plt.xlabel("Model Year")
plt.ylabel("Number of EVs")
plt.title("Annual Growth of EV Registrations")
plt.grid(True)
plt.show()
```

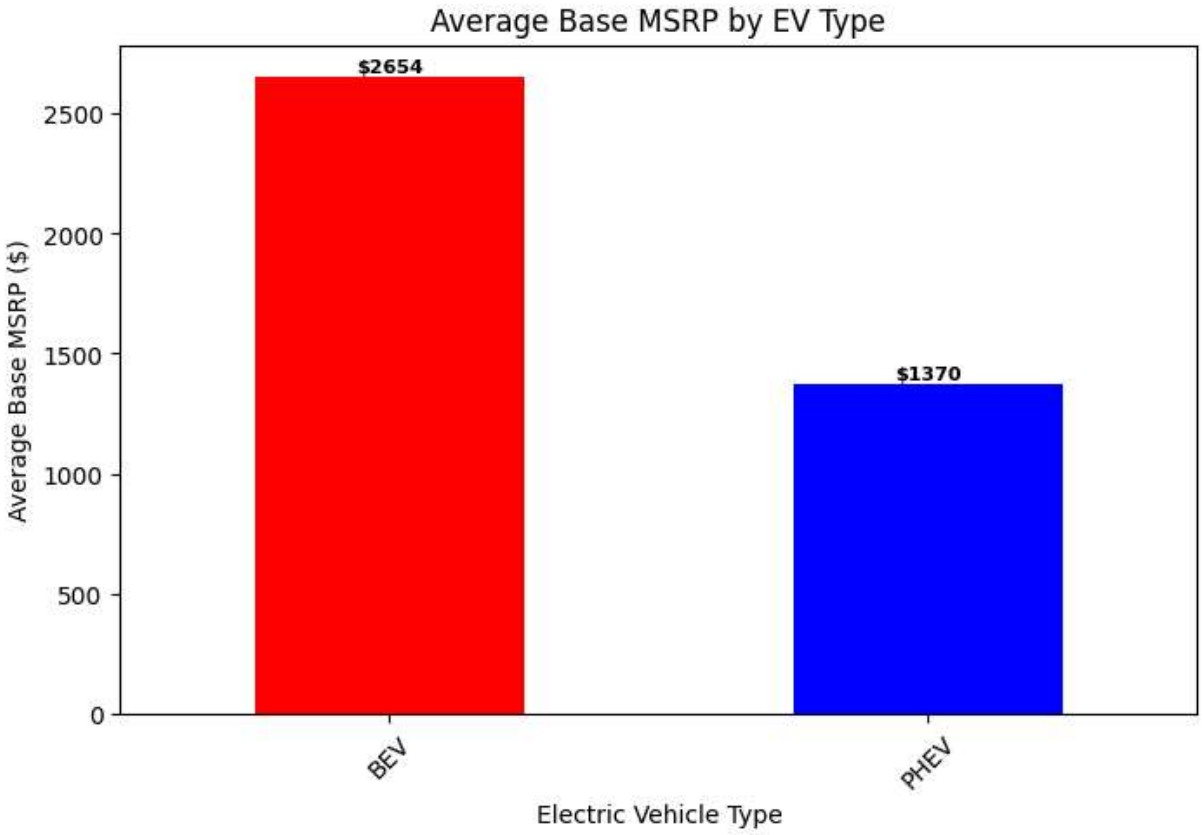


In [37]:
```
# Plot histogram of Electric Range
plt.figure(figsize=(10, 5))
plt.hist(df["Electric Range"], bins=15, color="orange", edgecolor="blue")
plt.xlabel("Electric Range (Miles)")
plt.ylabel("Number of EVs")
plt.title("Distribution of Electric Ranges")
plt.grid(True)
plt.show()
```



In [58]:
```
# Group by EV type and find average price
avg_price = df.groupby("Electric Vehicle Type")["Base MSRP"].mean()

# Plot bar chart
plt.figure(figsize=(8, 5))
ax = avg_price.plot(kind="bar", color=["red", "blue", "green"])
```

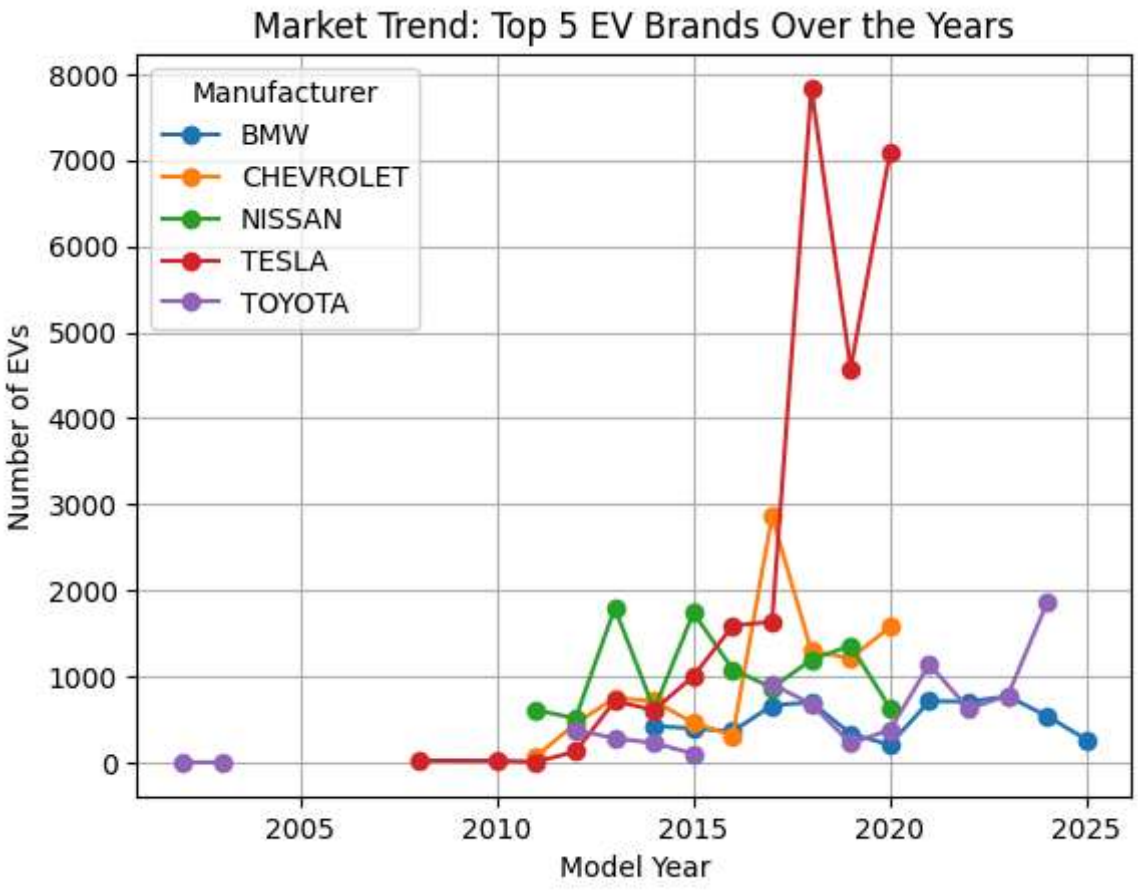
```
# Annotate bars with values

for i, value in enumerate(avg_price):
    ax.text(i, value + 20, f'${value:.0f}', ha='center', fontsize=8, fontweight='bo


plt.xlabel("Electric Vehicle Type")
plt.ylabel("Average Base MSRP ($)")
plt.title("Average Base MSRP by EV Type")
plt.xticks(rotation=45)
plt.show()
```



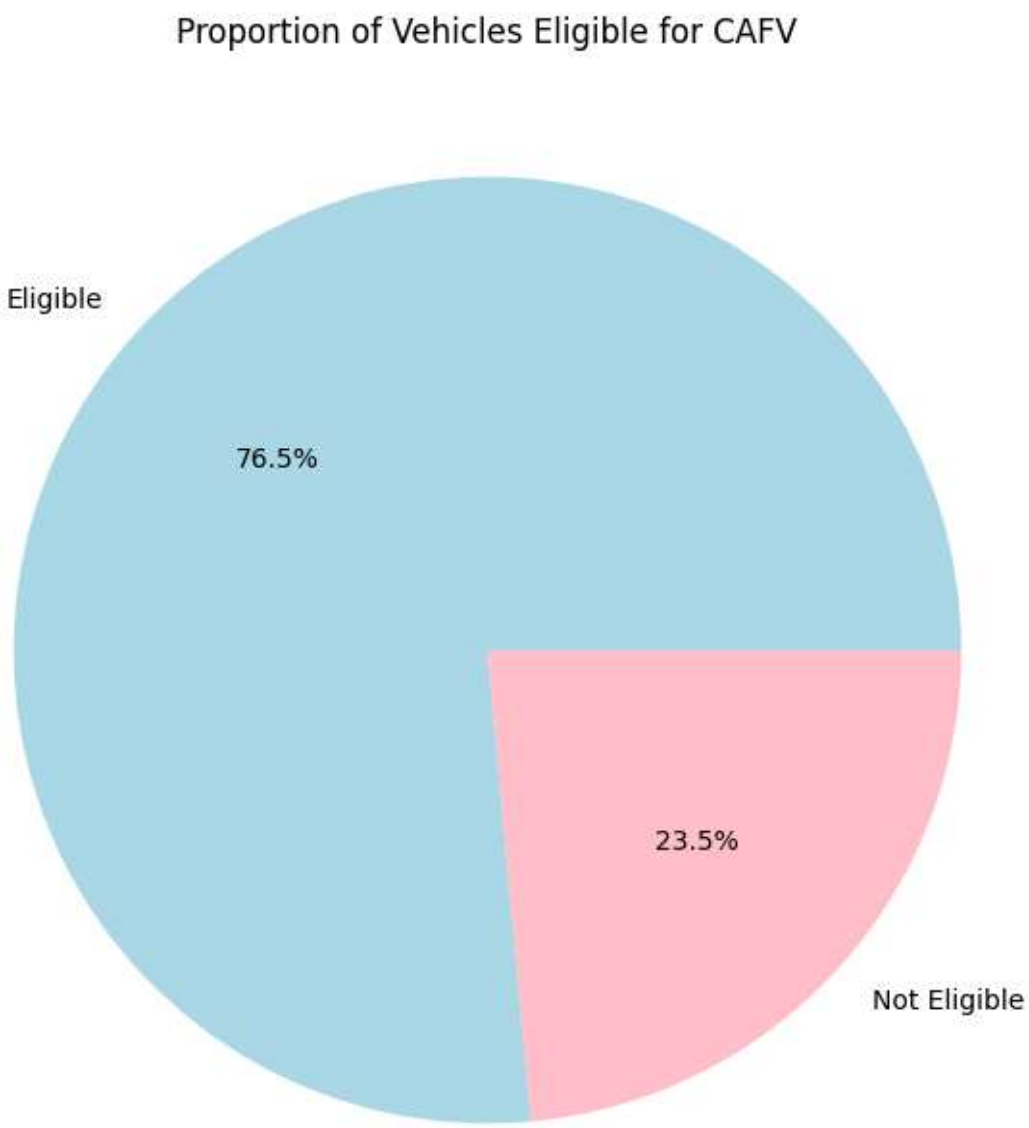
In [46]:

```
# Get top 5 brands
top_brands = df["Make"].value_counts().head(5).index

# Filter dataset for top brands
df_top_brands = df[df["Make"].isin(top_brands)]

# Group by year and brand
brand_trend = df_top_brands.groupby(["Model Year", "Make"]).size().unstack()

# Plot line chart
plt.figure(figsize=(10, 5))
brand_trend.plot(marker='o', linestyle='-')
plt.xlabel("Model Year")
plt.ylabel("Number of EVs")
plt.title("Market Trend: Top 5 EV Brands Over the Years")
plt.legend(title="Manufacturer")
plt.grid(True)
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



In [47]:
```
# Count eligibility
cafv_count = df["CAFV Eligibility Simple"].value_counts()

# Plot pie chart
plt.figure(figsize=(8, 8))
plt.pie(cafv_count, labels=cafv_count.index, autopct='%1.1f%%', colors=["lightblue"
plt.title("Proportion of Vehicles Eligible for CAFV")
plt.show()
```


Proportion of Vehicles Eligible for CAFV
Eligible
76.5%
23.5%
Not Eligible

In [ ]: