data-analysis-on-electric-vehicle

October 11, 2024

1 Data Analysis

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2 Load the DataSet

[3]:
```
# Load the dataset
file_path = "D:\INNOMATICS FILE\dataset.csv"
data = pd.read_csv(file_path)
```

[3]:
```
data.head()
```

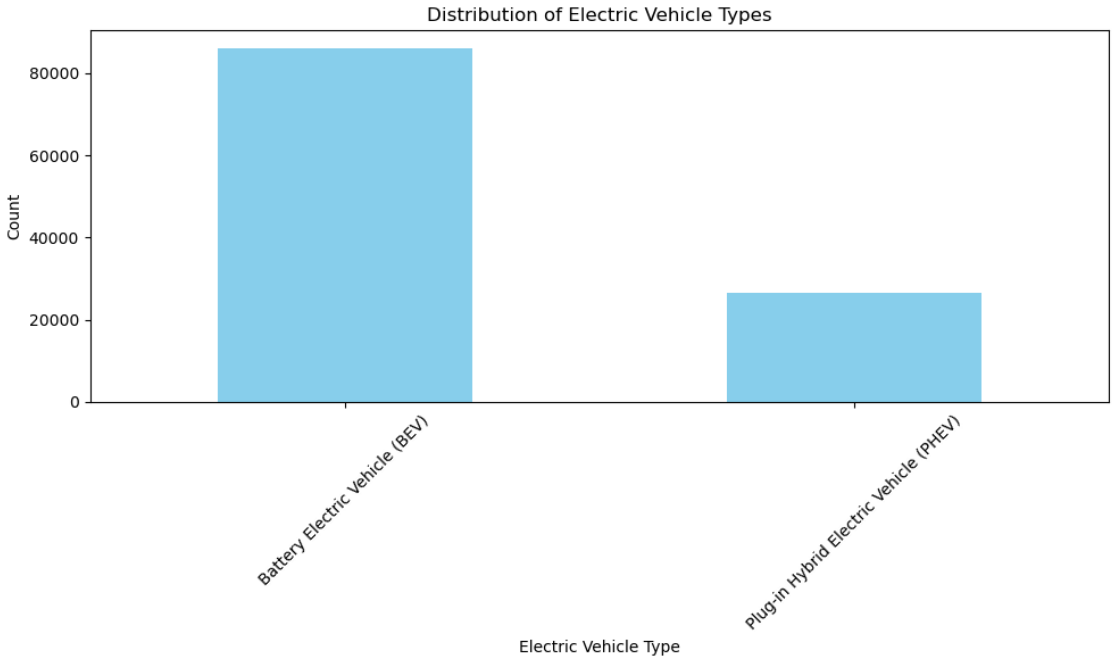
[3]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range
0	JTMEB3FV6N	Monroe	Key West	FL	33040	2022	TOYOTA	RAV4 PRIME	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	42
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238
4	3FA6P0SU1K	Snohomish	Everett	WA	98201	2019	FORD	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	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
```

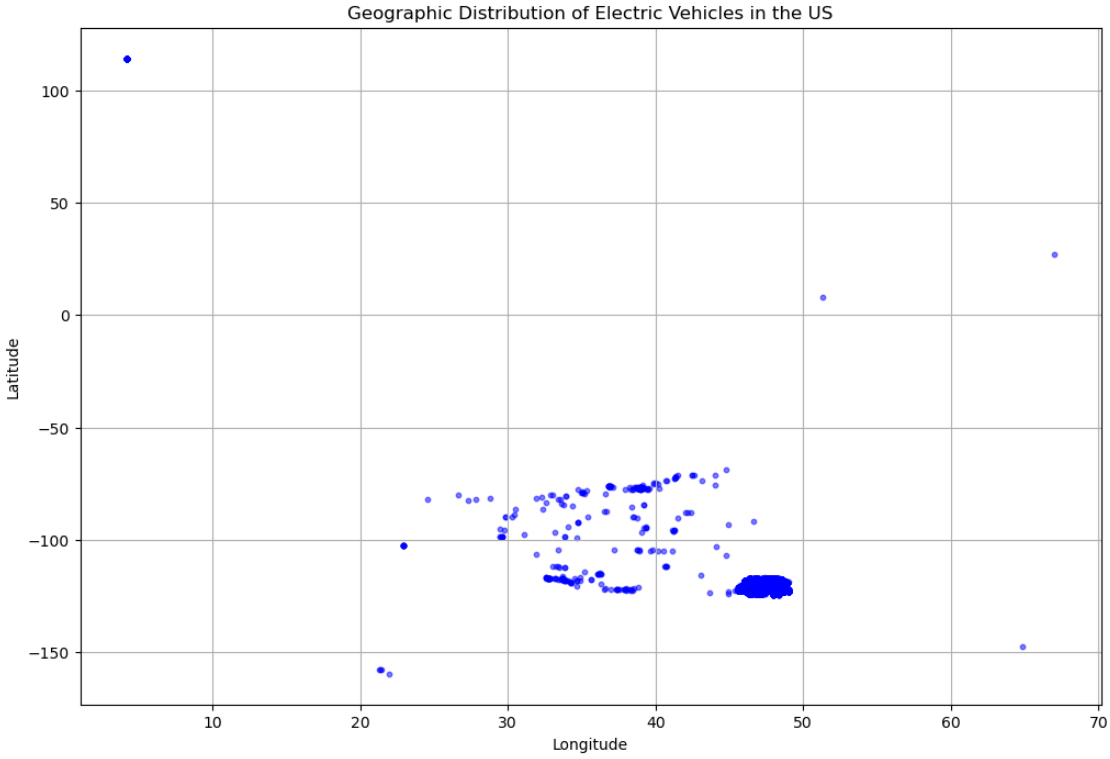
3 Distribution Of Electric Vehicle Types

[4]:
```
data[['Latitude', 'Longitude']] = data['Vehicle Location'].str.extract(r'\((.*)␣
 ↪(.*)\)').astype(float)

# Dropping rows with missing values in important columns like Electric Vehicle␣
 ↪Type, Make, and Electric Range
cleaned_data = data.dropna(subset=['Electric Vehicle Type', 'Make', 'Electric␣
 ↪Range'])

# Handle missing values for Legislative District and Electric Utility
cleaned_data['Legislative District'].fillna('Unknown', inplace=True)
cleaned_data['Electric Utility'].fillna('Unknown', inplace=True)

# Prepare visualization: Distribution of Electric Vehicle Types
vehicle_type_distribution = cleaned_data['Electric Vehicle Type'].value_counts()

# Visualizing the distribution
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
vehicle_type_distribution.plot(kind='bar', color='skyblue')
plt.title('Distribution of Electric Vehicle Types')
plt.xlabel('Electric Vehicle Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()

# Display the plot
plt.show()
```



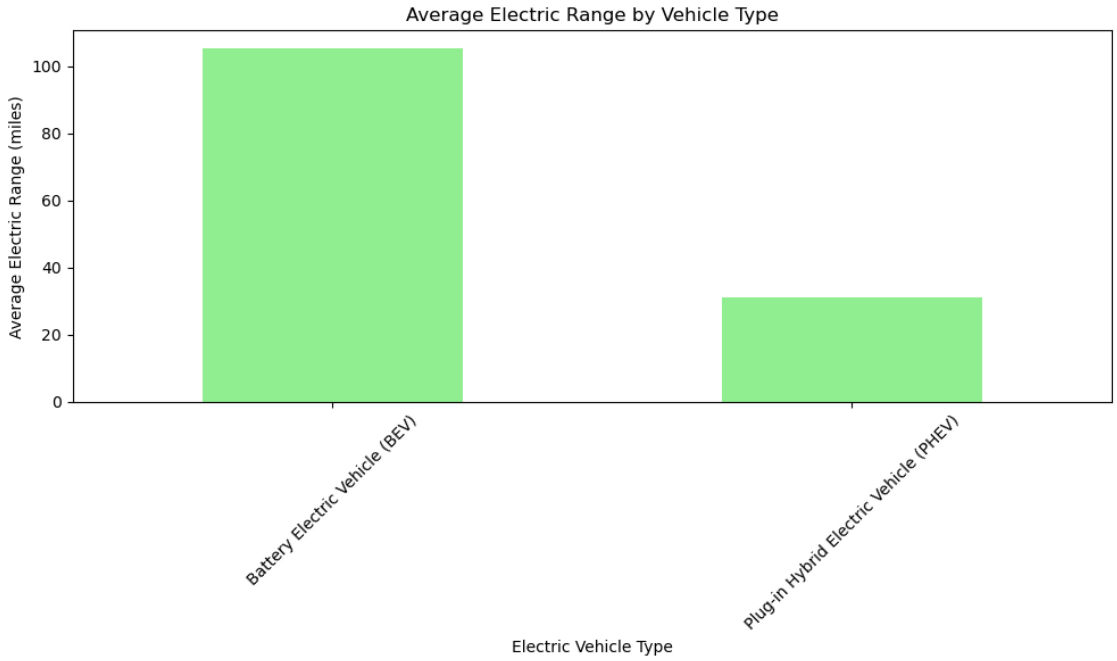
4 Distribution Of Electric Vehicles In The US

[4]:
```
plt.figure(figsize=(12, 8))

# Plot the points using a scatter plot
plt.scatter(cleaned_data['Longitude'], cleaned_data['Latitude'], c='blue',␣
 ↪s=10, alpha=0.5)

plt.title('Geographic Distribution of Electric Vehicles in the US')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.grid(True)

# Display the scatter plot
plt.show()
```



5 Average Electric Range By Vehicle Type

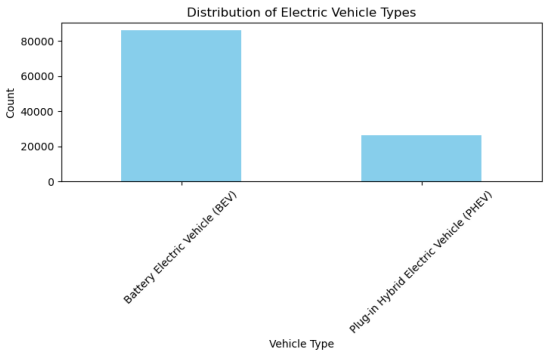
[5]:
```
# Prepare the data for visualization: Electric Range by Vehicle Type
vehicle_type_range = cleaned_data.groupby('Electric Vehicle Type')['Electric␣
 ↪Range'].mean()

# Plot the average electric range by vehicle type
plt.figure(figsize=(10, 6))
vehicle_type_range.plot(kind='bar', color='lightgreen')
plt.title('Average Electric Range by Vehicle Type')
plt.xlabel('Electric Vehicle Type')
plt.ylabel('Average Electric Range (miles)')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```

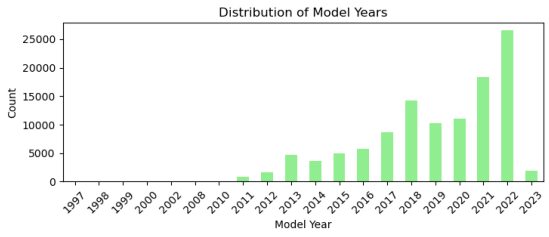


6 Univariate Analysis:

[6]:
```
#  Distribution of Electric Vehicle Types
vehicle_type_distribution = cleaned_data['Electric Vehicle Type'].value_counts()

# Univariate Analysis: Distribution of Model Years
model_year_distribution = cleaned_data['Model Year'].value_counts().sort_index()

# Univariate Analysis: Distribution of Electric Range
electric_range_distribution = cleaned_data['Electric Range'].describe()

# Bivariate Analysis: Electric Range by Vehicle Make
make_vs_range = cleaned_data.groupby('Make')['Electric Range'].mean().
 ↪sort_values(ascending=False)

# Bivariate Analysis: Electric Range vs Model Year
range_vs_year = cleaned_data.groupby('Model Year')['Electric Range'].mean().
 ↪sort_index()

# Plotting the results
fig, axs = plt.subplots(3, 2, figsize=(15, 12))

# Electric Vehicle Type Distribution
vehicle_type_distribution.plot(kind='bar', ax=axs[0, 0], color='skyblue')
```

```
axs[0, 0].set_title('Distribution of Electric Vehicle Types')
axs[0, 0].set_xlabel('Vehicle Type')
axs[0, 0].set_ylabel('Count')
axs[0, 0].tick_params(axis='x', rotation=45)

# Model Year Distribution
model_year_distribution.plot(kind='bar', ax=axs[0, 1], color='lightgreen')
axs[0, 1].set_title('Distribution of Model Years')
axs[0, 1].set_xlabel('Model Year')
axs[0, 1].set_ylabel('Count')
axs[0, 1].tick_params(axis='x', rotation=45)

# Electric Range Distribution (Descriptive Stats shown as text)
axs[1, 0].axis('off')  # Turn off this axis, just showing text
axs[1, 0].text(0.5, 0.5, f'Descriptive Stats of Electric Range:
 ↪\n\n{electric_range_distribution}', ha='center', va='center', fontsize=12)

# Electric Range by Vehicle Make
make_vs_range.plot(kind='bar', ax=axs[1, 1], color='coral')
axs[1, 1].set_title('Average Electric Range by Vehicle Make')
axs[1, 1].set_xlabel('Vehicle Make')
axs[1, 1].set_ylabel('Average Electric Range')
axs[1, 1].tick_params(axis='x', rotation=45)

# Electric Range vs Model Year
range_vs_year.plot(kind='line', marker='o', ax=axs[2, 0])
axs[2, 0].set_title('Average Electric Range Over Model Years')
axs[2, 0].set_xlabel('Model Year')
axs[2, 0].set_ylabel('Average Electric Range (miles)')

# Adjust layout
plt.tight_layout()
plt.show()
```

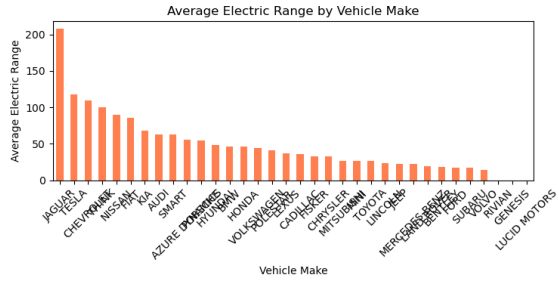


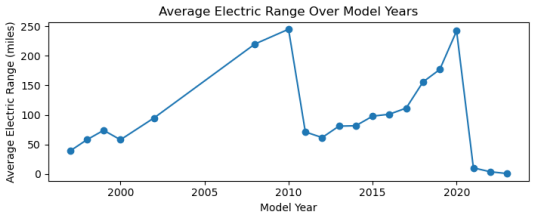


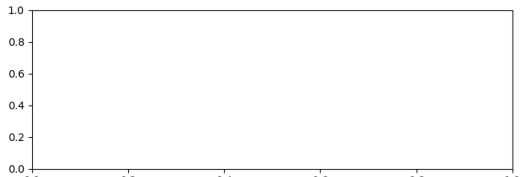
Descriptive Stats of Electric Range:

count	112634.000000
mean	87.812987
std	102.334216
min	0.000000
25%	0.000000
50%	32.000000
75%	208.000000
max	337.000000

Name: Electric Range, dtype: float64







7 Task 2: Choropleth map of Electric Vehicles by State

[7]:
```
import plotly.express as px

# Task 2: Choropleth map of Electric Vehicles by State

# Prepare data for the choropleth map (Count of vehicles per state)
state_vehicle_counts = cleaned_data['State'].value_counts().reset_index()
state_vehicle_counts.columns = ['State', 'Vehicle Count']

# Plot the choropleth map using Plotly Express
fig = px.choropleth(state_vehicle_counts,
                    locations='State',
                    locationmode="USA-states",
                    color='Vehicle Count',
                    scope="usa",
                    color_continuous_scale="Blues",
                    title="Number of Electric Vehicles by State")
```

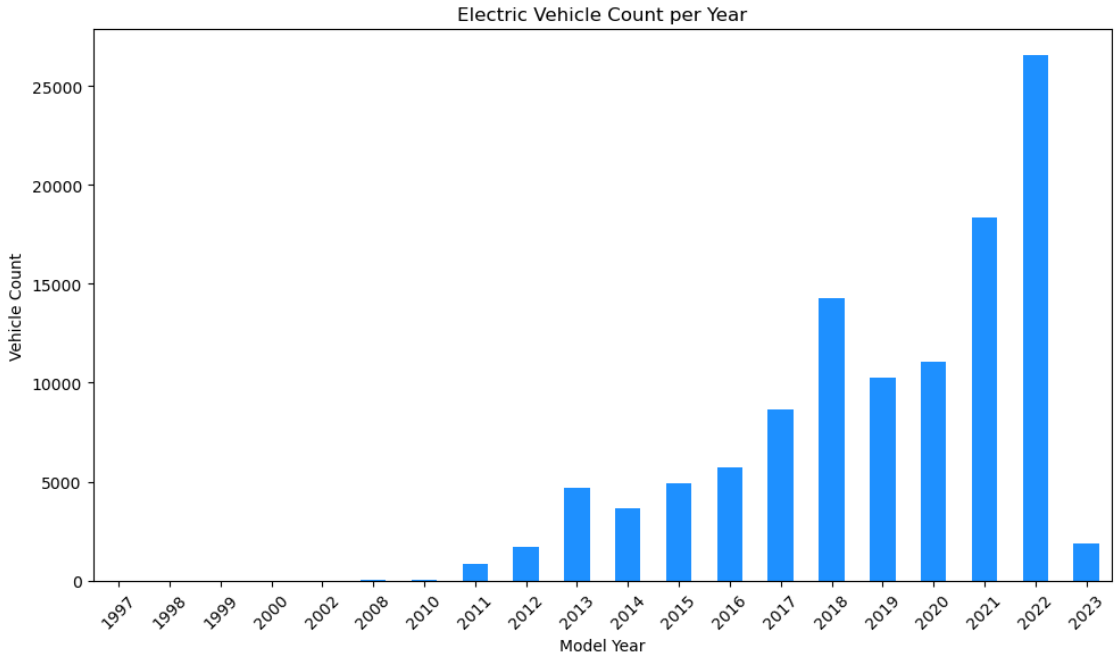
```
# Display the map
fig.show()
```

8 Task 3: Racing Bar Plot to display the animation of EV Make and its count over time

[8]:
```
# Count the number of vehicles by Make and Model Year
make_year_data = cleaned_data.groupby(['Model Year', 'Make']).size().
 ↪reset_index(name='Vehicle Count')

# Use Plotly Express to create the racing bar plot
fig = px.bar(make_year_data,
             x='Vehicle Count',
             y='Make',
             color='Make',
             animation_frame='Model Year',
             orientation='h',
             title='EV Make and Count Over Time (Racing Bar Plot)',
             range_x=[0, make_year_data['Vehicle Count'].max() + 50])

# Display the plot
fig.show()
```

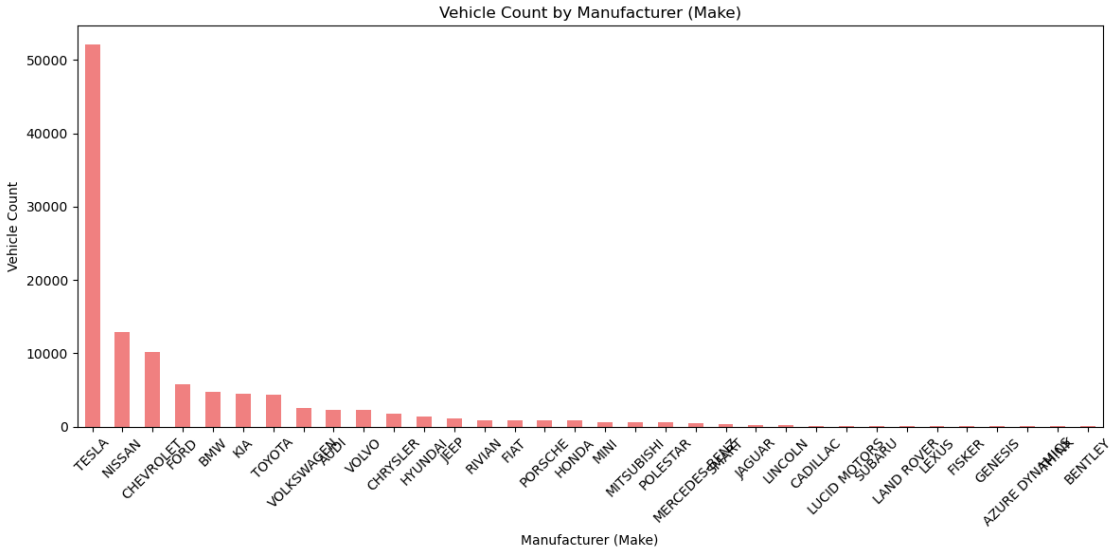
9 Show the electric vehicle count per year

[9]:
```
# Group the data by 'Model Year' and count the number of vehicles per year
vehicle_count_per_year = cleaned_data.groupby('Model Year').size()

# Plotting the electric vehicle count per year
plt.figure(figsize=(10, 6))
vehicle_count_per_year.plot(kind='bar', color='dodgerblue')
plt.title('Electric Vehicle Count per Year')
plt.xlabel('Model Year')
plt.ylabel('Vehicle Count')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



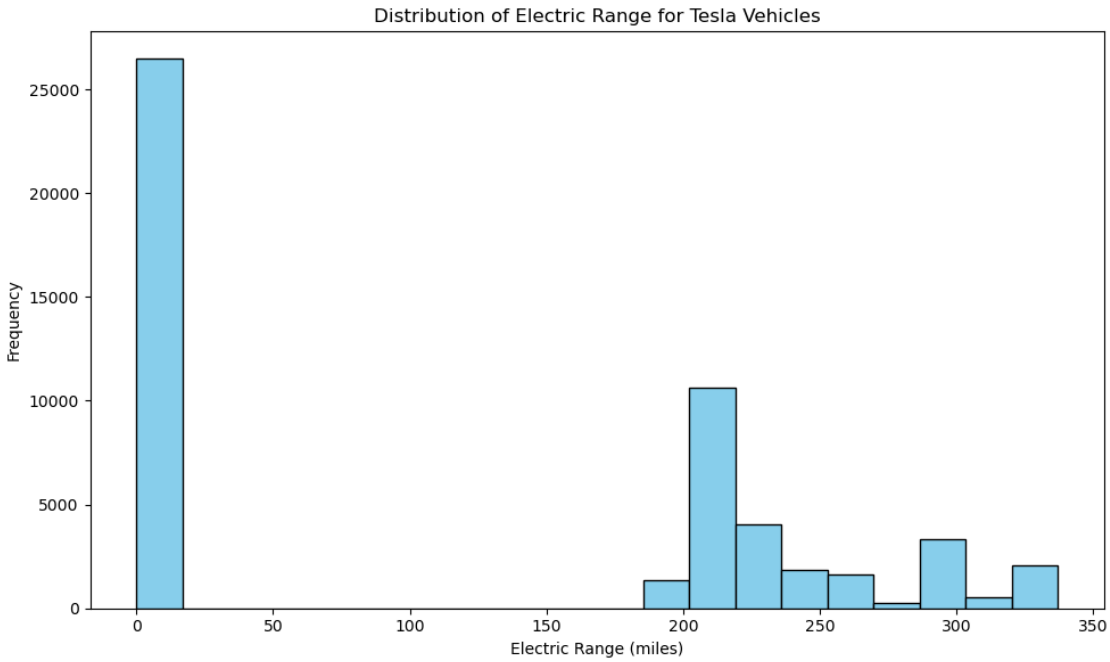
10 Show vehicle count based on manufacturers

[10]:
```
# Data cleaning: extracting Latitude and Longitude, and handling missing values
data[['Latitude', 'Longitude']] = data['Vehicle Location'].str.extract(r'\((.*)␣
 ↪(.*)\)').astype(float)
cleaned_data = data.dropna(subset=['Electric Vehicle Type', 'Make', 'Electric␣
 ↪Range'])
cleaned_data['Legislative District'].fillna('Unknown', inplace=True)
cleaned_data['Electric Utility'].fillna('Unknown', inplace=True)

# Task: Show the vehicle counts based on manufacturers (Make)
vehicle_count_per_make = cleaned_data['Make'].value_counts()

# Plotting the vehicle count per manufacturer
plt.figure(figsize=(12, 6))
vehicle_count_per_make.plot(kind='bar', color='lightcoral')
plt.title('Vehicle Count by Manufacturer (Make)')
plt.xlabel('Manufacturer (Make)')
plt.ylabel('Vehicle Count')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



11 Electric Range For Tesla Vehicles

[11]:
```
# Filter the data for Tesla vehicles only
tesla_data = cleaned_data[cleaned_data['Make'] == 'TESLA']

# Plot the distribution of electric ranges for Tesla vehicles
plt.figure(figsize=(10, 6))
plt.hist(tesla_data['Electric Range'], bins=20, color='skyblue',␣
 ↪edgecolor='black')
plt.title('Distribution of Electric Range for Tesla Vehicles')
plt.xlabel('Electric Range (miles)')
plt.ylabel('Frequency')
plt.tight_layout()

# Show the plot
plt.show()
```



12 Find the Tesla model with the highest electric range

[12]:
```
# Identify the Tesla vehicle with the highest electric range
tesla_max_range = tesla_data[tesla_data['Electric Range'] ==␣
 ↪tesla_data['Electric Range'].max()]

# Display the relevant columns for the vehicle with the highest electric range
tesla_max_range[['Make', 'Model', 'Electric Range', 'Model Year']]
```

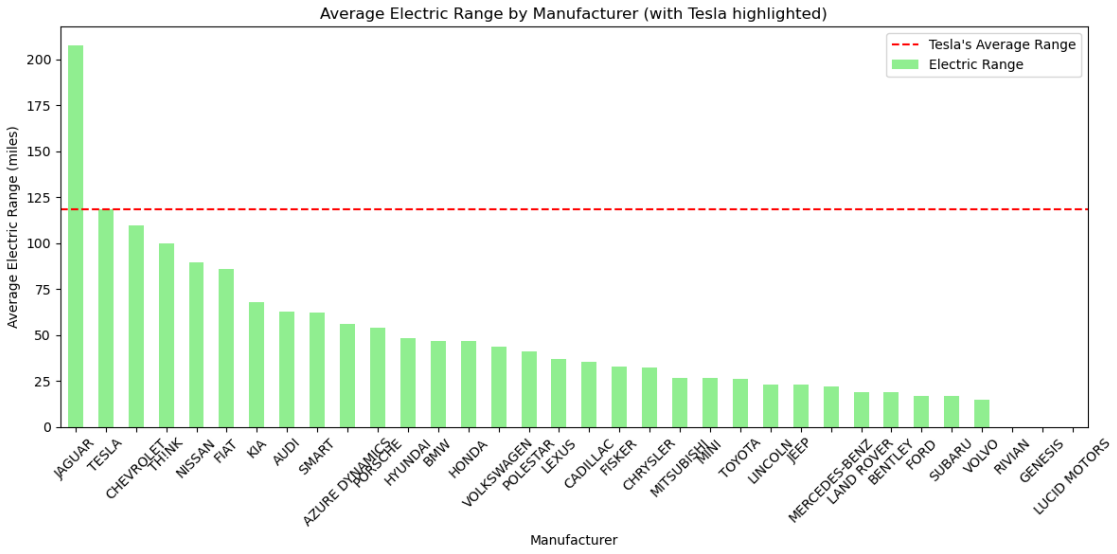
[12]:

	Make	Model	Electric Range	Model Year
3211	TESLA	MODEL S	337	2020
3792	TESLA	MODEL S	337	2020
8612	TESLA	MODEL S	337	2020
11828	TESLA	MODEL S	337	2020
12130	TESLA	MODEL S	337	2020
…	…	…	…	…
98960	TESLA	MODEL S	337	2020
99670	TESLA	MODEL S	337	2020
100161	TESLA	MODEL S	337	2020
102293	TESLA	MODEL S	337	2020
103718	TESLA	MODEL S	337	2020

[68 rows x 4 columns]

13 Compare Tesla's electric range with other manufacturers

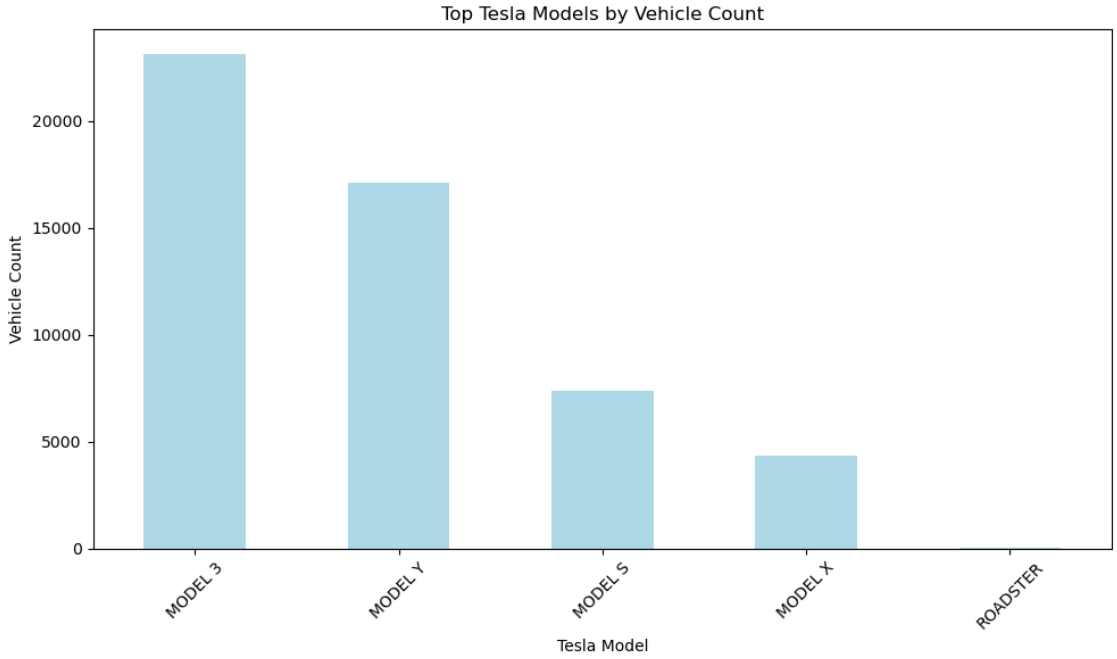
[13]:
```
# Group the data by Make and calculate the average electric range for each manufacturer
average_range_per_make = cleaned_data.groupby('Make')['Electric Range'].mean().sort_values(ascending=False)

# Plot the comparison of Tesla's range with other manufacturers
plt.figure(figsize=(12, 6))
average_range_per_make.plot(kind='bar', color='lightgreen')
plt.axhline(y=tesla_data['Electric Range'].mean(), color='red', linestyle='--', label="Tesla's Average Range")
plt.title('Average Electric Range by Manufacturer (with Tesla highlighted)')
plt.xlabel('Manufacturer')
plt.ylabel('Average Electric Range (miles)')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()

# Show the plot
plt.show()
```



14 Identify the top Tesla models by vehicle count

[14]:
```
# Filter data for Tesla vehicles and group by model to count the number of each model
top_tesla_models = tesla_data['Model'].value_counts()
```

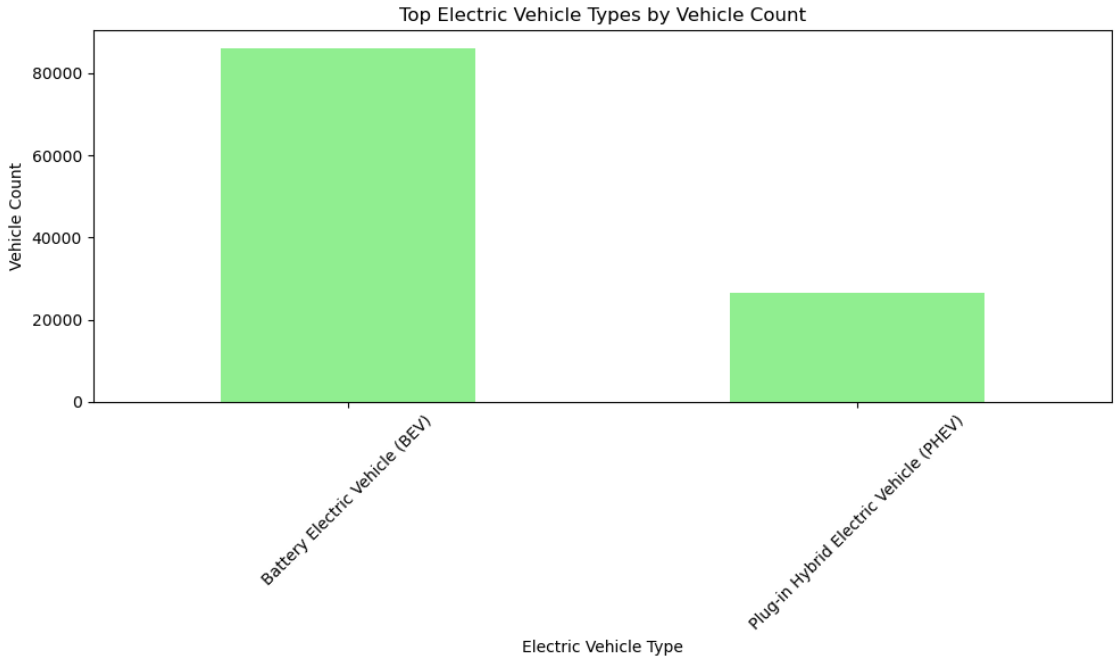
```
# Plot the top Tesla models by count
plt.figure(figsize=(10, 6))
top_tesla_models.plot(kind='bar', color='lightblue')
plt.title('Top Tesla Models by Vehicle Count')
plt.xlabel('Tesla Model')
plt.ylabel('Vehicle Count')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



15 the top electric vehicle types by count

[15]:
```
# Group the data by 'Electric Vehicle Type' and count the number of vehicles␣
↪for each type
top_ev_types = cleaned_data['Electric Vehicle Type'].value_counts()

# Plot the top electric vehicle types by count
plt.figure(figsize=(10, 6))
top_ev_types.plot(kind='bar', color='lightgreen')
plt.title('Top Electric Vehicle Types by Vehicle Count')
plt.xlabel('Electric Vehicle Type')
```

```
plt.ylabel('Vehicle Count')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



16 the electric vehicle model with the lowest range

[16]:
```
# Find the vehicle(s) with the lowest electric range in the dataset
lowest_range_vehicle = cleaned_data[cleaned_data['Electric Range'] ==␣
↪cleaned_data['Electric Range'].min()]

# Display the relevant columns for the vehicle(s) with the lowest electric range
lowest_range_vehicle[['Make', 'Model', 'Electric Range', 'Model Year']]
```

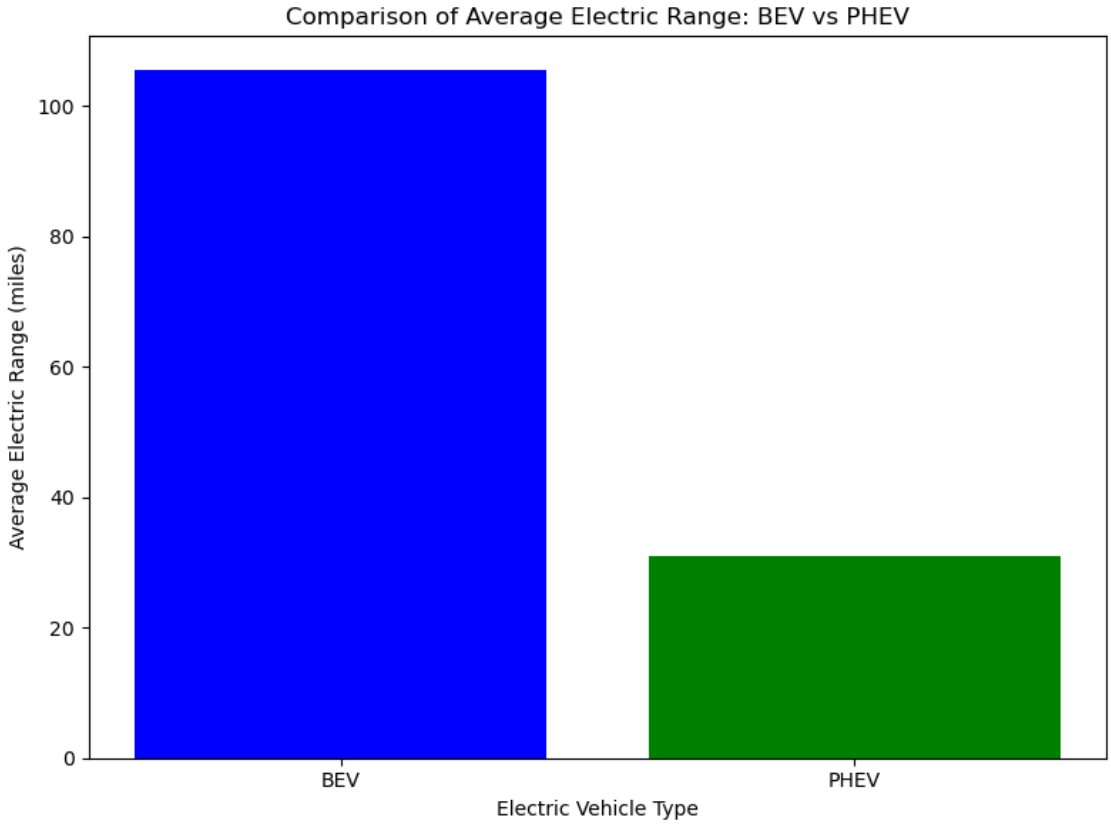
[16]:

	Make	Model	Electric Range	Model Year
22	NISSAN	LEAF	0	2021
40	FORD	F-150	0	2022
62	NISSAN	LEAF	0	2022
63	CHEVROLET	BOLT EV	0	2022
65	TESLA	MODEL Y	0	2022
…	…	…	…	…
112622	VOLVO	NaN	0	2023

```
112623      TESLA          MODEL Y                0        2022
112625       FORD  MUSTANG MACH-E                0        2022
112626      TESLA          MODEL 3                0        2022
112629      TESLA          MODEL Y                0        2022

[39236 rows x 4 columns]
```

17 BEV and PHEV vehicles

[17]:
```
# Filter the data for BEV and PHEV vehicles
bev_data = cleaned_data[cleaned_data['Electric Vehicle Type'] == 'Battery␣
 ↪Electric Vehicle (BEV)']
phev_data = cleaned_data[cleaned_data['Electric Vehicle Type'] == 'Plug-in␣
 ↪Hybrid Electric Vehicle (PHEV)']

# Calculate the average electric range for both types
bev_avg_range = bev_data['Electric Range'].mean()
phev_avg_range = phev_data['Electric Range'].mean()

# Plot the comparison of BEV and PHEV ranges
plt.figure(figsize=(8, 6))
plt.bar(['BEV', 'PHEV'], [bev_avg_range, phev_avg_range], color=['blue',␣
 ↪'green'])
plt.title('Comparison of Average Electric Range: BEV vs PHEV')
plt.xlabel('Electric Vehicle Type')
plt.ylabel('Average Electric Range (miles)')
plt.tight_layout()

# Show the plot
plt.show()
```



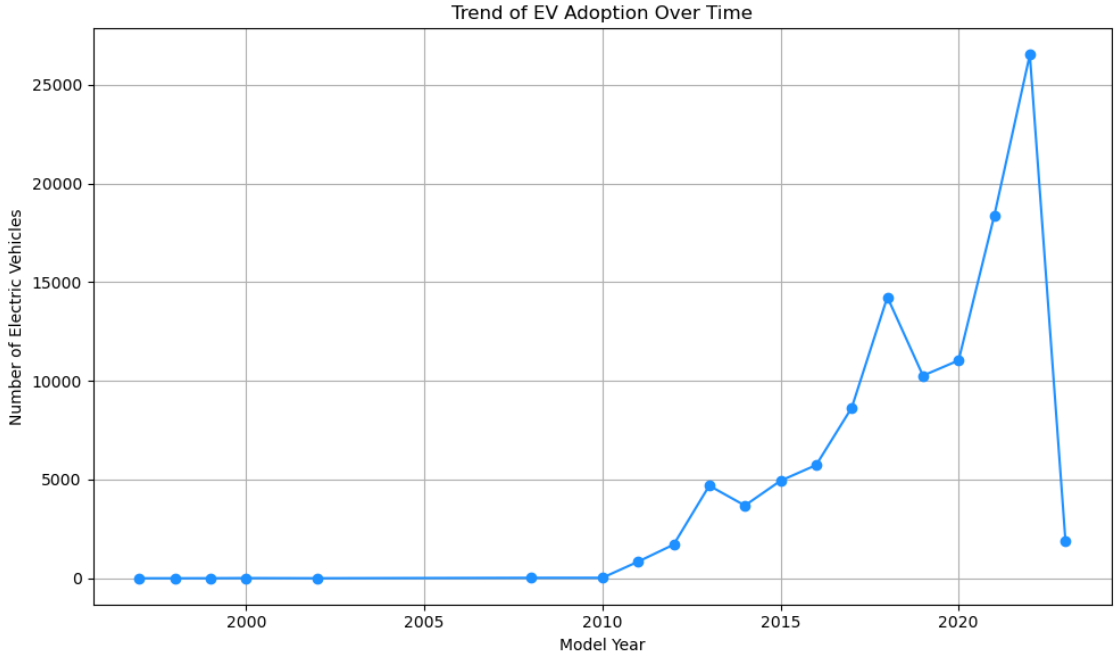
18 the manufacturer with the best average PHEV range

[18]:
```
# Group the PHEV data by 'Make' and calculate the average electric range for each manufacturer
phev_avg_range_per_make = phev_data.groupby('Make')['Electric Range'].mean().sort_values(ascending=False)

# Display the top manufacturer(s) with the best PHEV range
best_phev_range_manufacturer = phev_avg_range_per_make.head(1)

# Show the result
best_phev_range_manufacturer
```

[18]:
```
Make
HONDA    46.618687
Name: Electric Range, dtype: float64
```

19 'Model Year' and count the number of vehicles for each year

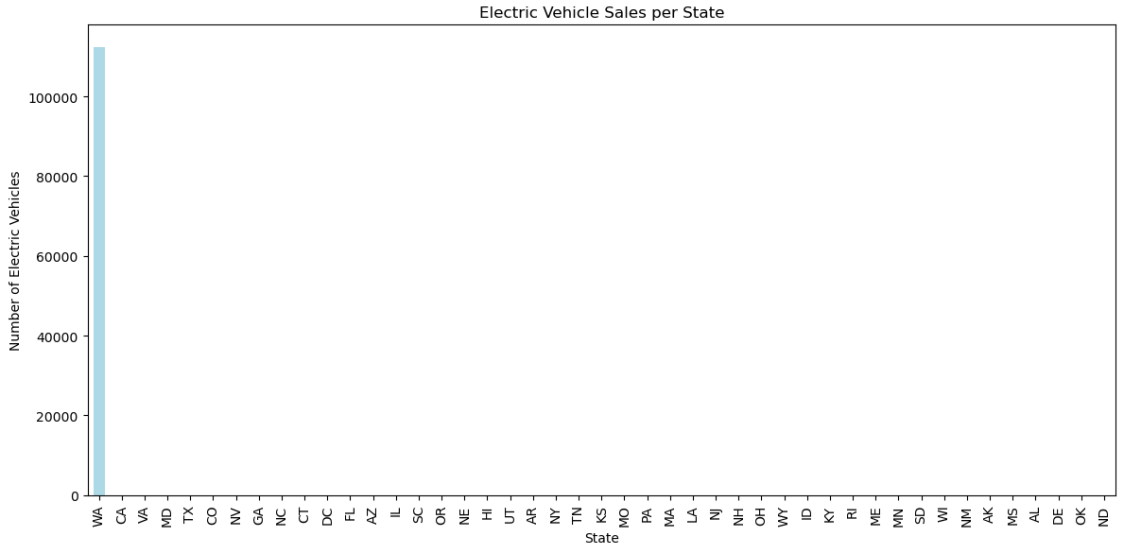
[19]:
```
import matplotlib.pyplot as plt
ev_adoption_trend = cleaned_data.groupby('Model Year').size()

# Plot the EV adoption trend over time
plt.figure(figsize=(10, 6))
ev_adoption_trend.plot(kind='line', marker='o', color='dodgerblue')
plt.title('Trend of EV Adoption Over Time')
plt.xlabel('Model Year')
plt.ylabel('Number of Electric Vehicles')
plt.grid(True)
plt.tight_layout()

# Show the plot
plt.show()
```



20 'State' and count the number of vehicles for each state

[21]:
```
import matplotlib.pyplot as plt

ev_sales_per_state = cleaned_data['State'].value_counts()

# Plotting the electric vehicle sales per state
plt.figure(figsize=(12, 6))
```

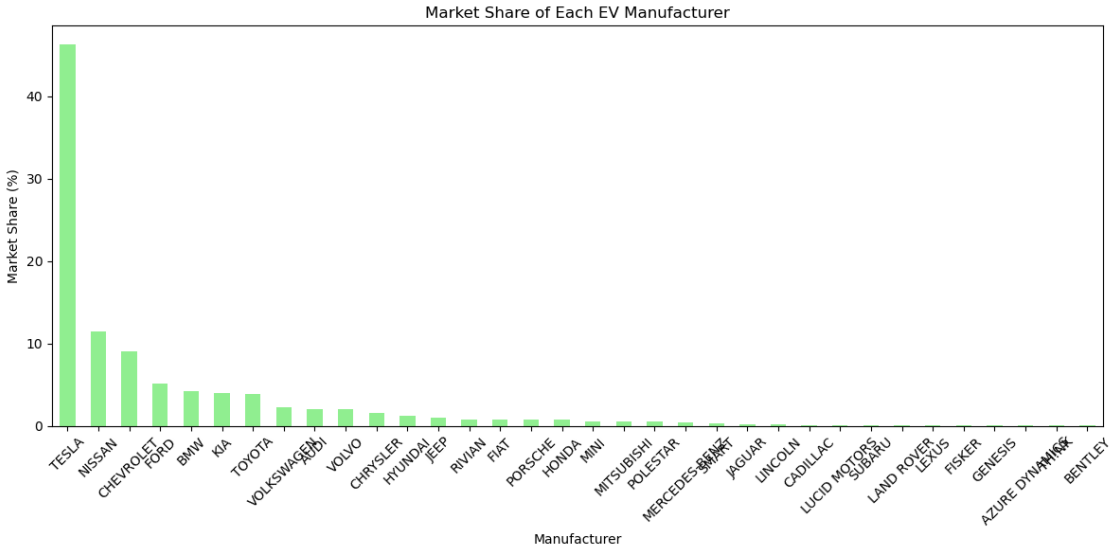
```
ev_sales_per_state.plot(kind='bar', color='lightblue')
plt.title('Electric Vehicle Sales per State')
plt.xlabel('State')
plt.ylabel('Number of Electric Vehicles')
plt.xticks(rotation=90)
plt.tight_layout()

# Show the plot
plt.show()
```



21 calculate the total count for each manufacturer

[22]:
```
import matplotlib.pyplot as plt

# Group the data by 'Make' and calculate the total count for each manufacturer
ev_manufacturer_market_share = cleaned_data['Make'].
 ↪value_counts(normalize=True) * 100

# Plotting the market share of each EV manufacturer
plt.figure(figsize=(12, 6))
ev_manufacturer_market_share.plot(kind='bar', color='lightgreen')
plt.title('Market Share of Each EV Manufacturer')
plt.xlabel('Manufacturer')
plt.ylabel('Market Share (%)')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
```

```
plt.show()
```


Market Share of Each EV Manufacturer
Market Share (%)
0
10
20
30
40
TESLA
NISSAN
CHEVROLET
FORD
BMW
KIA
TOYOTA
VOLKSWAGEN
AUDI
VOLVO
CHRYSLER
HYUNDAI
JEEP
RIVIAN
FIAT
PORSCHE
HONDA
MINI
MITSUBISHI
POLESTAR
MERCEDES-BENZ
SMART
JAGUAR
LINCOLN
CADILLAC
LUCID MOTORS
SUBARU
LAND ROVER
LEXUS
FISKER
GENESIS
AZURE DYNAMICS
HINO
BENTLEY
Manufacturer

[ ]: