In [4]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [2]:

```
# pip install plotly.express
```

In [6]:

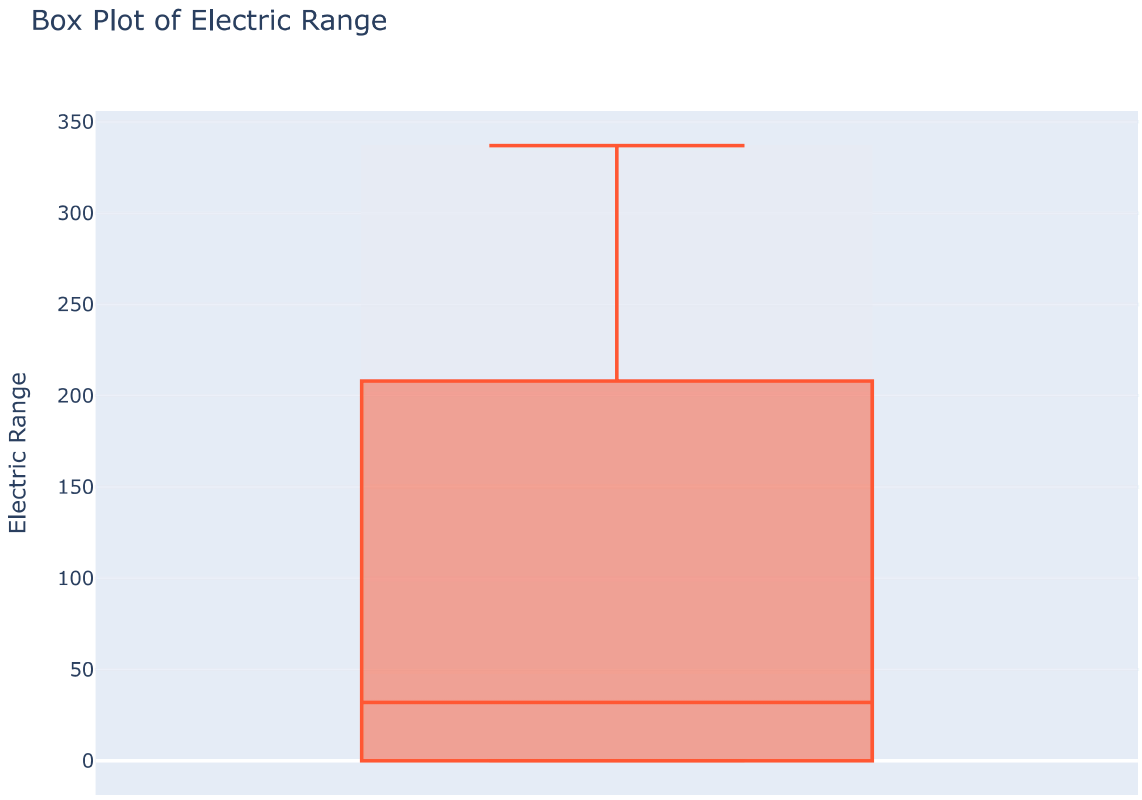
```
df = pd.read_csv("dataset.csv")
df.head()
```

Out[6]:

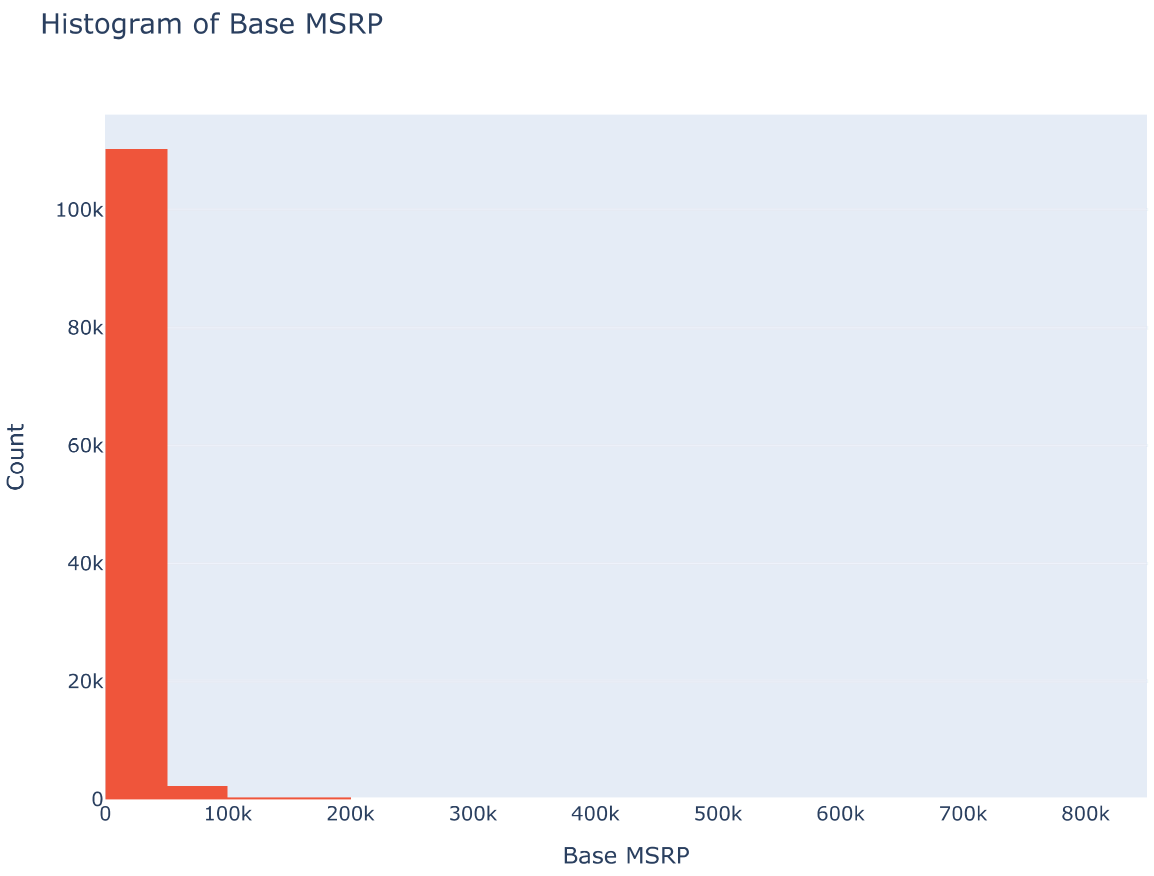
| | VIN (1-10) | County | City | State | Postal Code | Model Year | Make | Model | Electric Vehicle Type | Clean Alternative Fuel Vehicle (CAFV) Eligibility | Electric Range | Base MSRP | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | JTMEB3FV6N | Monroe | Key West | FL | 33040 | 2022 | TOYOTA | RAV4 PRIME | Plug-in Hybrid Electric Vehicle (PHEV) | Clean Alternative Fuel Vehicle Eligible | 42 | 0 | |
| **1** | 1G1RD6E45D | Clark | Laughlin | NV | 89029 | 2013 | CHEVROLET | VOLT | Plug-in Hybrid Electric Vehicle (PHEV) | Clean Alternative Fuel Vehicle Eligible | 38 | 0 | |
| **2** | JN1AZ0CP8B | Yakima | Yakima | WA | 98901 | 2011 | NISSAN | LEAF | Battery Electric Vehicle (BEV) | Clean Alternative Fuel Vehicle Eligible | 73 | 0 | |
| **3** | 1G1FW6S08H | Skagit | Concrete | WA | 98237 | 2017 | CHEVROLET | BOLT EV | Battery Electric Vehicle (BEV) | Clean Alternative Fuel Vehicle Eligible | 238 | 0 | |
| **4** | 3FA6P0SU1K | Snohomish | Everett | WA | 98201 | 2019 | FORD | FUSION | Plug-in Hybrid Electric Vehicle (PHEV) | Not eligible due to low battery range | 26 | 0 | |

# Univariate Analysis

```
In [7]: fig = px.box(df, y = 'Electric Range',
                     title = 'Box Plot of Electric Range',
                     labels = {'Electric Range':'Electric Range'},
                     color_discrete_sequence = ['#FF5733']) # Custom color (e.g., orange)

        # customize layout
        fig.update_layout(yaxis_title = 'Electric Range',width = 800, height = 600)

        # show plot
        fig.show()
```

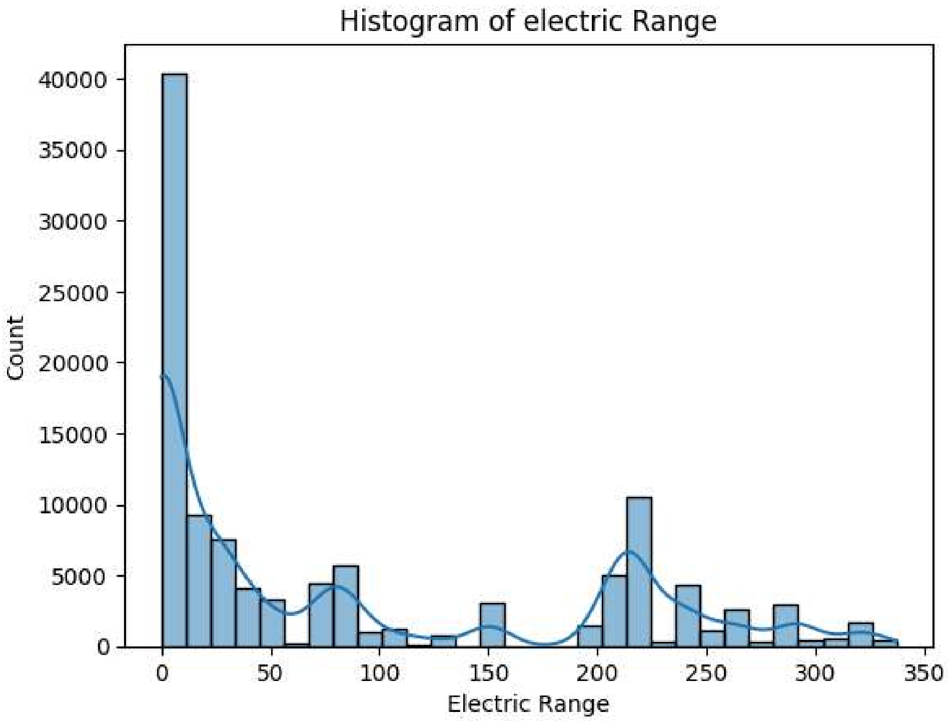


In [8]:
```
px.histogram(df,x = 'Base MSRP', title = 'Histogram of Base MSRP',nbins = 30,
color_discrete_sequence = ['#EF553B']).update_layout(
xaxis_title = 'Base MSRP',yaxis_title = 'Count',width = 800, height = 600).show()
```

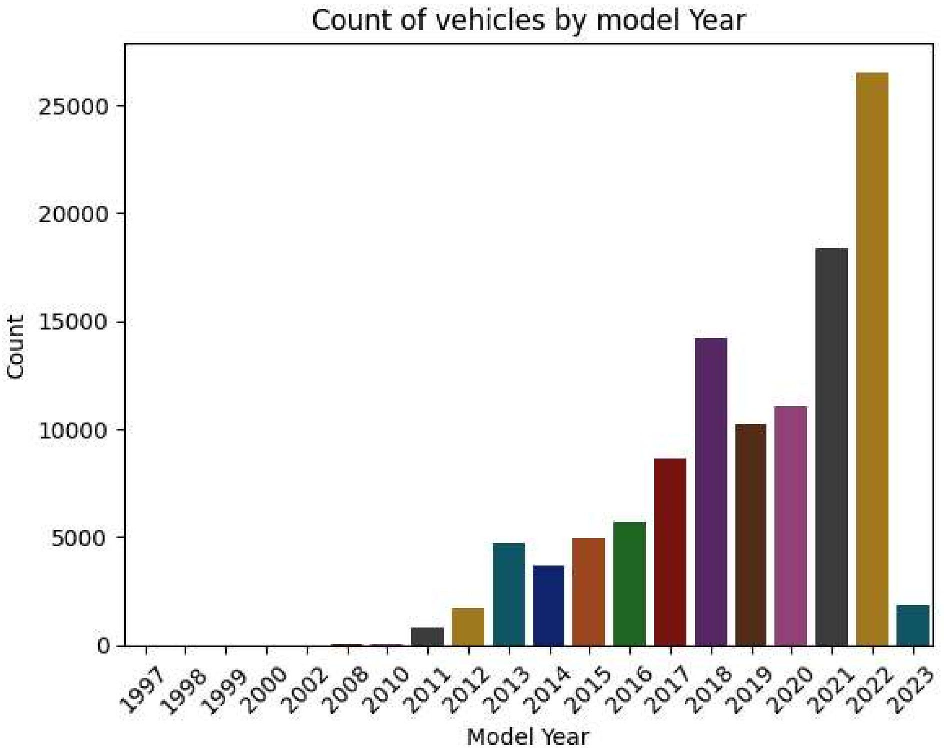


## Histograms for numerical features

In [9]:
```
# Replace 'electric Range' with the actual column name if it contains spaces.
sns.histplot(df['Electric Range'],
bins = 30, kde = True).set_title("Histogram of electric Range")
plt.show()
```

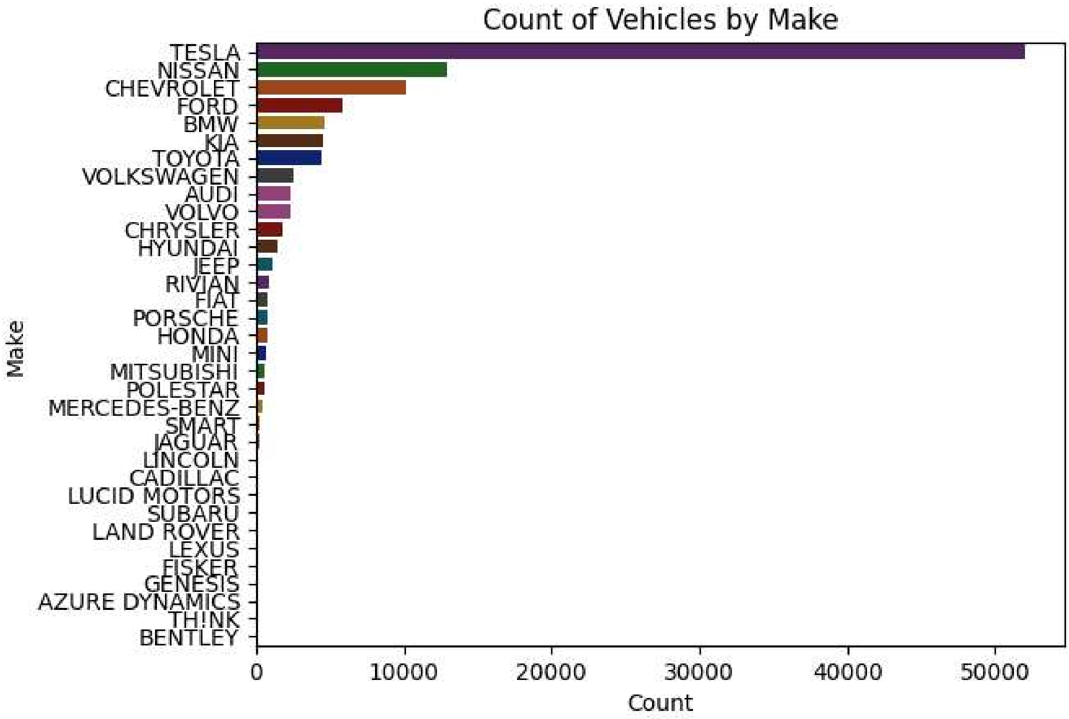


In [10]:
```
# use a darker color palette
sns.countplot(x = 'Model Year', data = df, palette = 'dark',hue = "Model Year",legend =False)
plt.title("Count of vehicles by model Year")
plt.xticks(rotation = 45)
plt.xlabel('Model Year')
plt.ylabel('Count')
plt.show()
```



In [11]:

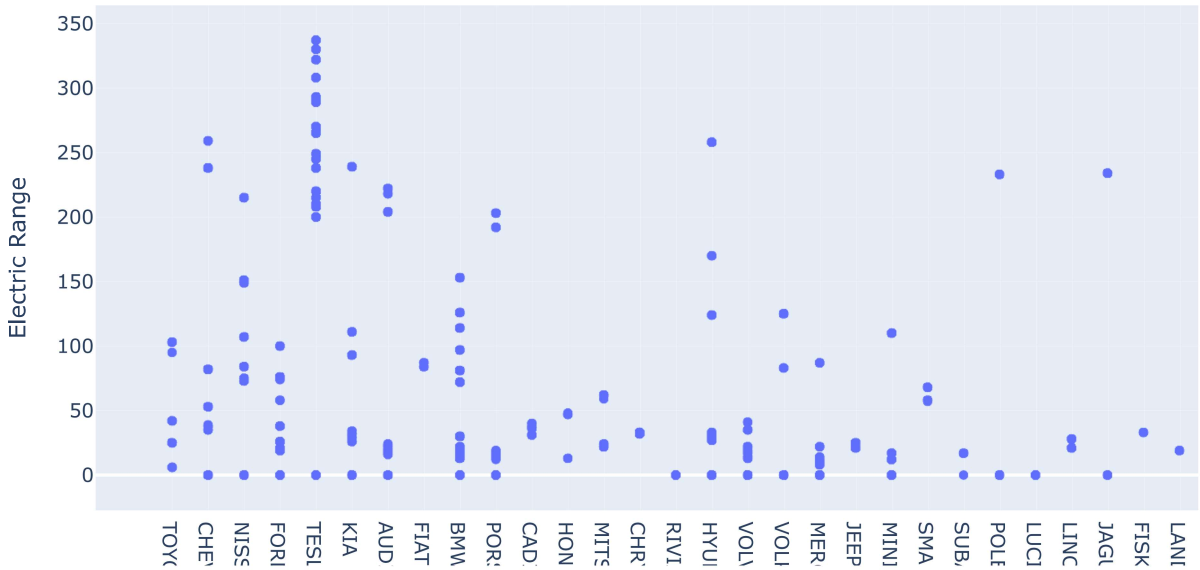
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming you have a DataFrame named 'df' with a column 'Make'

# Use seaborn's built-in dark palette
sns.countplot(y="Make", data=df, order=df['Make'].value_counts().index, hue="Make", palette="dark",

plt.title("Count of Vehicles by Make")
plt.ylabel('Make')
plt.xlabel("Count")
plt.show()
```



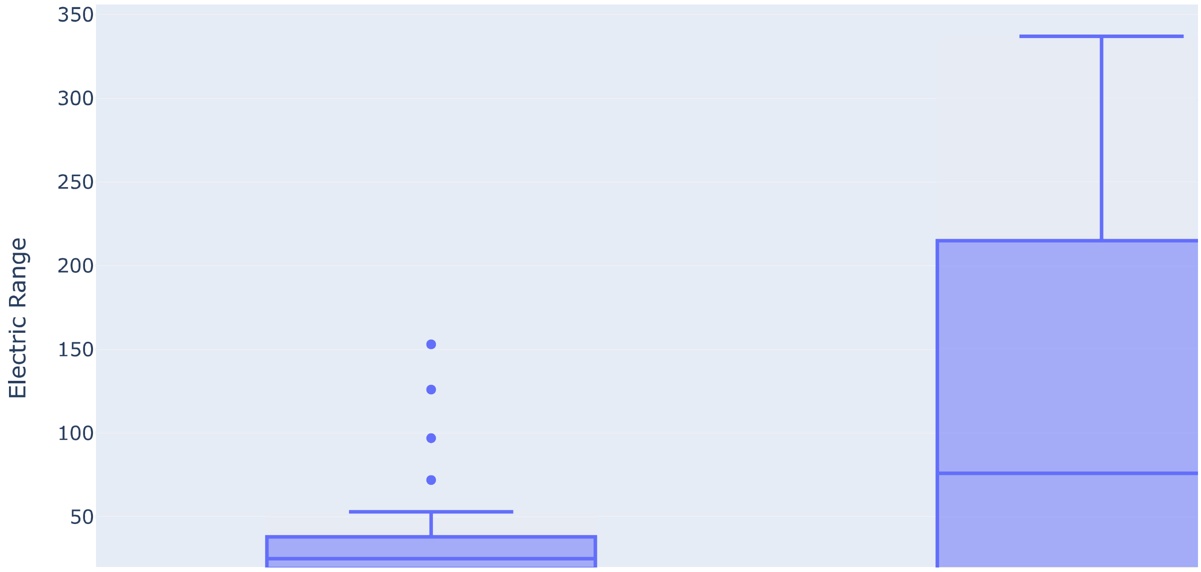
## Bivariate Analysis

Scatter plot using plotly.express Numerical vs Numerical A (Electric Range vs Make) Task - This is an open ended problem. apply exploratory data analysis (Univariate and Bivariate) on the dataset available above.

```
In [12]: px.scatter(df,x = 'Make',y = 'Electric Range')
```

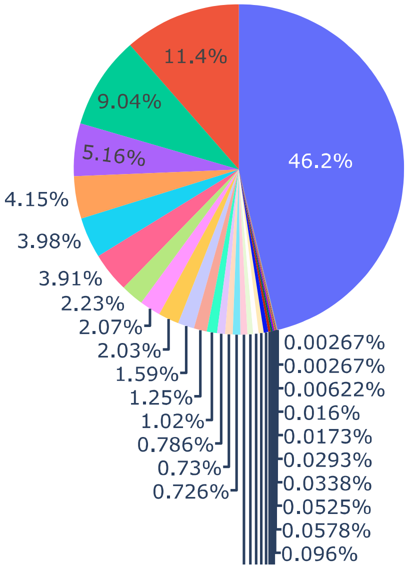


```
In [13]: # Box plot using plotly.express
         px.box(df, x = "Electric Vehicle Type",y = "Electric Range")
```



## pie chart plot using plotly.express

```
In [14]: px.pie(df,names = 'Make',values = '2020 Census Tract')
```



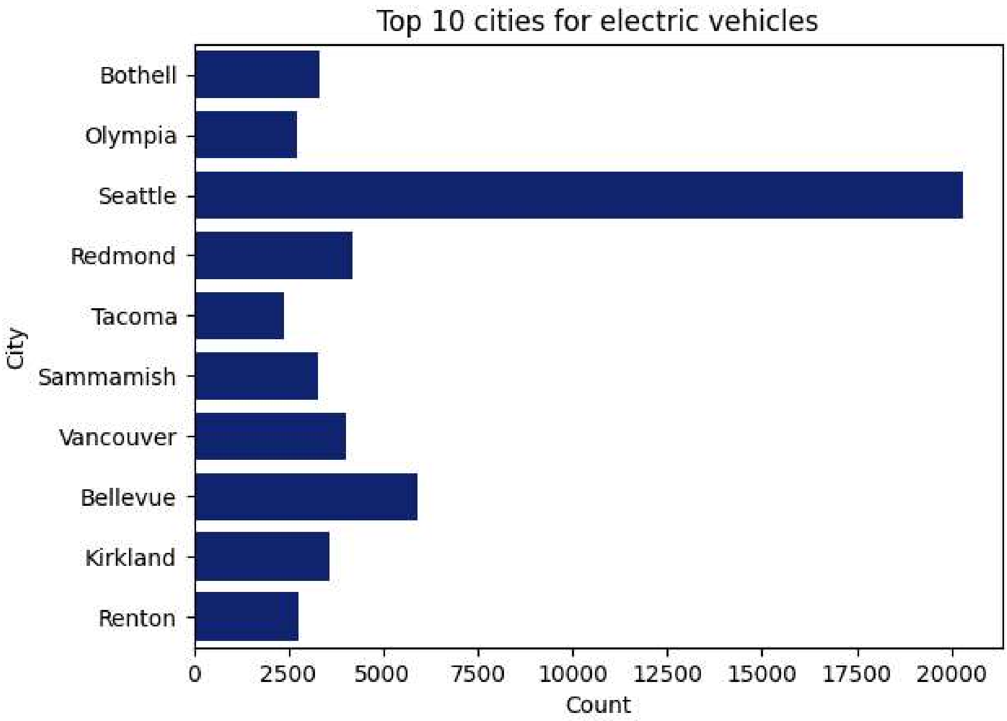
```
In [15]: df["State"].unique()

Out[15]: array(['FL', 'NV', 'WA', 'IL', 'NY', 'VA', 'OK', 'KS', 'CA', 'NE', 'MD',
                'CO', 'DC', 'TN', 'SC', 'CT', 'OR', 'TX', 'SD', 'HI', 'GA', 'MS',
                'AR', 'NC', 'MO', 'UT', 'PA', 'DE', 'OH', 'WY', 'AL', 'ID', 'AZ',
                'AK', 'LA', 'NM', 'WI', 'KY', 'NJ', 'MN', 'MA', 'ME', 'RI', 'NH',
                'ND'], dtype=object)
```

```
In [16]: grouped_df = df.groupby("State").agg({"Electric Range":"mean"})
```

```
In [17]: df.shape

Out[17]: (112634, 17)
```

In [18]:

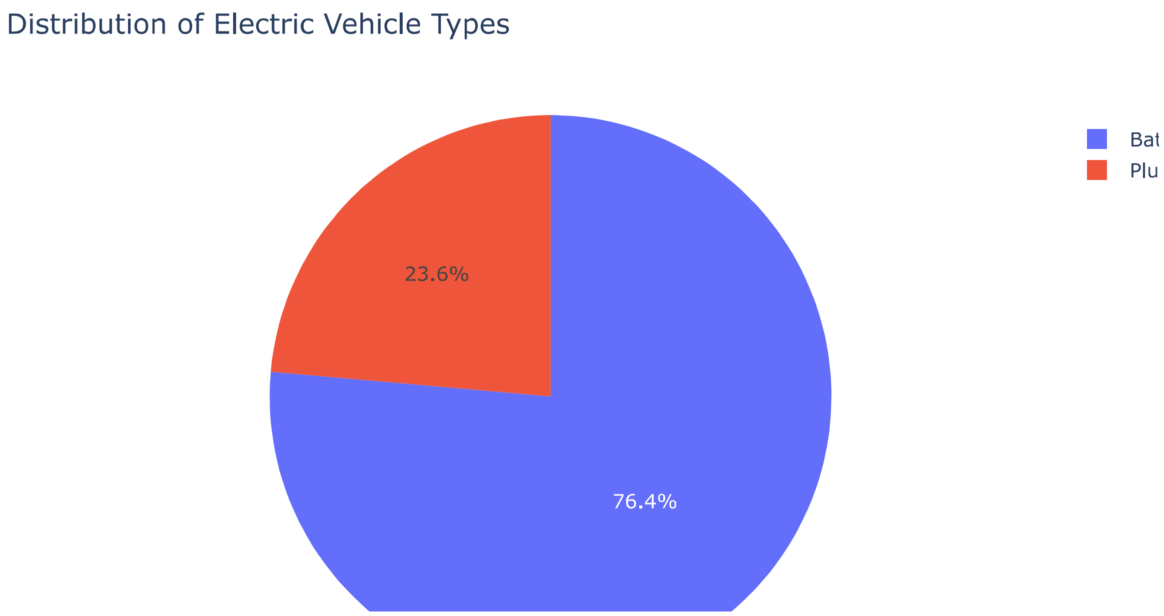
```
#use a darker color palette
sns.set_palette("dark")

top_cities = df['City'].value_counts().nlargest(10).index # top 10 cities
sns.countplot(y = 'City',data = df[df['City'].isin(top_cities)])
plt.title('Top 10 cities for electric vehicles')
plt.xlabel('Count')
plt.ylabel('City')
plt.show()
```



Type *Markdown* and LaTeX: $\alpha^2$

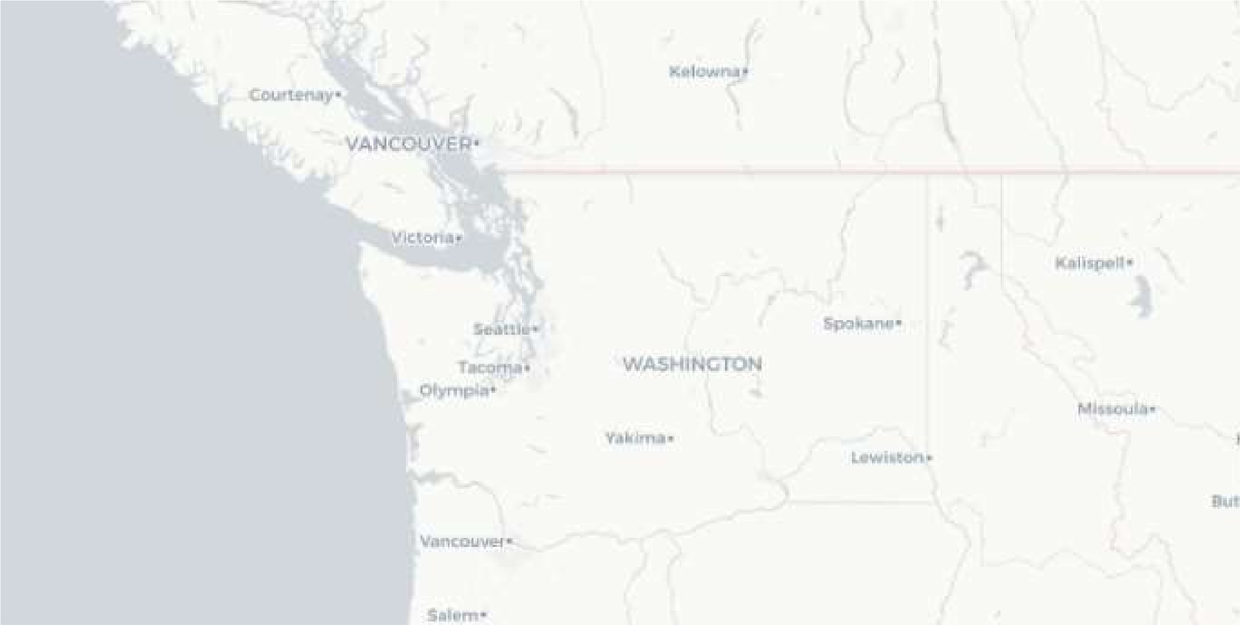
In [19]:

```python
# Pie chart for Electric Vehicle type
fig = px.pie(df, names = 'Electric Vehicle Type',title = 'Distribution of Electric Vehicle Types')
fig.show()
```



In [20]:

```
import pandas as pd, plotly.express as px

df = pd.read_csv("dataset.csv")
state_nyc = df.groupby(['Postal Code', 'Model Year']).size().reset_index(name='Number_of_Vehicles')

fig = px.choropleth_mapbox(state_nyc, geojson='https://raw.githubusercontent.com/python-visualizati
    locations='Postal Code', color='Number_of_Vehicles', featureidkey="properties.ZCTA5CE10", mapbo
    zoom=5, center={"lat": 47.7511, "lon": -120.7401}, animation_frame='Model Year')

fig.update_layout(margin={"r": 0, "t": 0, "l": 0, "b": 0}).show()
```

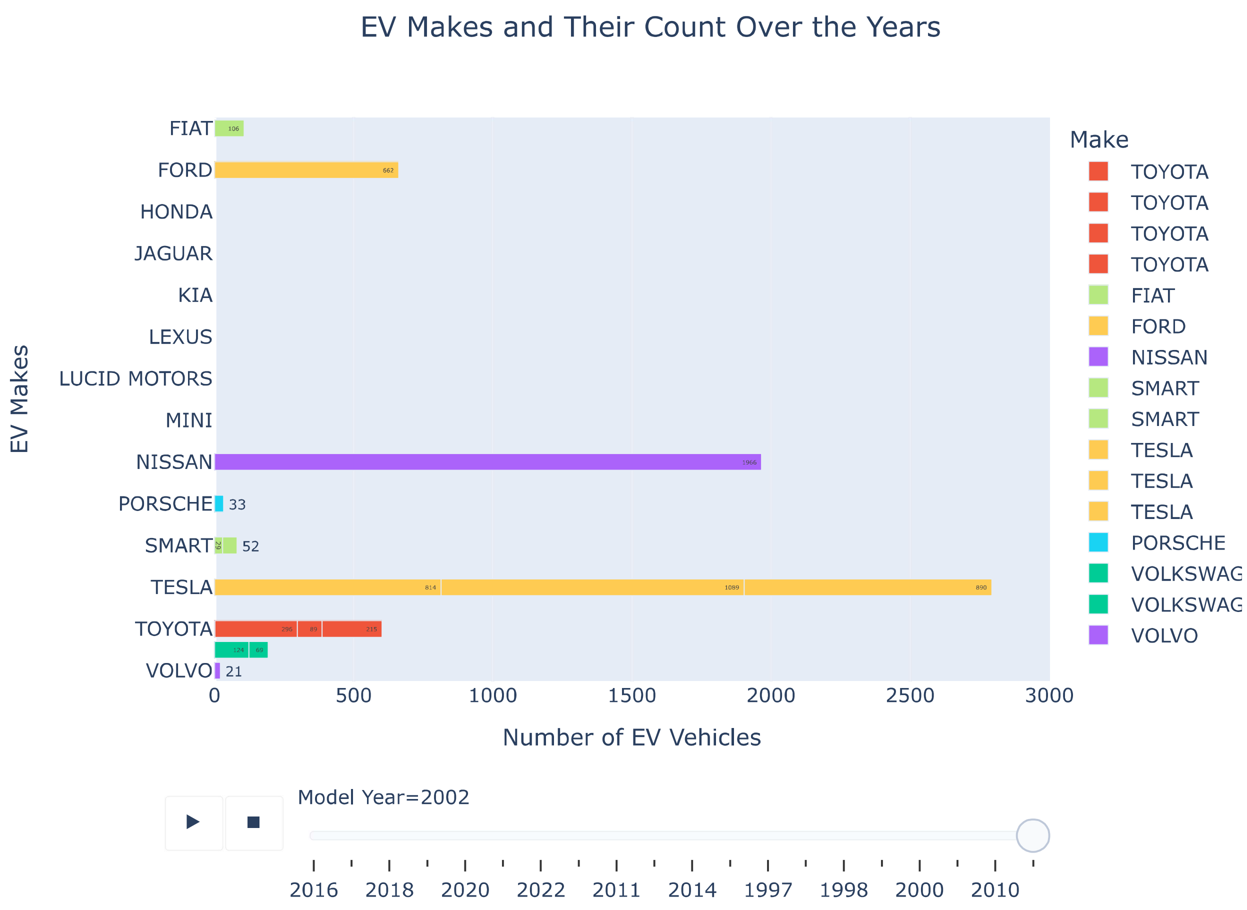


In [ ]:

```
# pip install bar_chart_race
```

In [21]:

```python
import pandas as pd
import plotly.express as px

df = pd.read_csv("dataset.csv")
d = df.groupby(['Make', 'Model Year']).size().reset_index(name='Number_of_Vehicles')

px.bar(d, x='Number_of_Vehicles', y='Make', color='Make', animation_frame='Model Year', orientation
       title='EV Makes and Their Count Over the Years', range_x=[0, 3000]).update_traces(
    texttemplate='%{x}', textposition='outside').update_layout(
    yaxis_title='EV Makes', xaxis_title='Number of EV Vehicles', title_x=0.5, width=800, height=600
```



In [ ]: