

```
In [1]: ! pip install plotly

Requirement already satisfied: plotly in c:\users\owner\desktop\alib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\owner\desktop\alib\site-packages (from plotly) (8.2.3)

In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

C:\Users\Owner\Desktop\alib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently install
ed).
  from pandas.core import (
```

Step 2 : Reading the csv data into a dataframe

```
In [3]: nyc = pd.read_csv(r"C:\Users\Owner\Downloads\dataset.csv")

In [4]: nyc.head()

Out[4]:
```

	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	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
0	JTMEB3FVN	Monroe	Key West	FL	33040	2022	TOYOTA	RAV4 PRIME	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	42	0	NaN	198968248	POINT (-81.80023 24.5546)	NaN	12087972100
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	NaN	5204412	POINT (-114.57245 35.16815)	NaN	32003006702
2	JN1A20CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	218972519	POINT (-120.50721 46.60446)	PACIFICORP	53077001602
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	186750406	POINT (-121.7515 48.53892)	PUGET SOUND ENERGY INC	53057951101
4	3FAP6PSUJK	Snohomish	Everett	WA	98201	2019	FORD	FUSION	Battery Electric Vehicle (PHEV)	Not eligible due to low battery range	26	0	38.0	2006714	POINT (-122.20596 47.97659)	PUGET SOUND ENERGY INC	53061041500

```
In [5]: nyc.shape
Out[5]: (112634, 17)

In [6]: nyc["Electric Vehicle Type"].unique()
Out[6]: array(['Plug-in Hybrid Electric Vehicle (PHEV)',
              'Battery Electric Vehicle (BEV)', dtype=object)

In [7]: nyc["Make"].unique()
Out[7]: array(['TOYOTA', 'CHEVROLET', 'NISSAN', 'FORD', 'TESLA', 'KIA', 'AUDI',
              'FIAT', 'BMW', 'Porsche', 'CADILLAC', 'HONDA', 'MITSUBISHI',
              'CHRYSLER', 'RIVIAN', 'HYUNDAI', 'VOLVO', 'VOLKSWAGEN',
              'MERCEDES-BENZ', 'JEEP', 'MINI', 'SMART', 'SUBARU', 'POLESTAR',
              'LUCID MOTORS', 'LINCOLN', 'JAGUAR', 'FISHER', 'LAND ROVER',
              'LEXUS', 'TRINX', 'GENESIS', 'BENTLEY', 'AZURE DYNAMICS'],
              dtype=object)

In [8]: nyc.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 object
5   Model Year          112634 non-null object
6   Make                112634 non-null object
7   Model               112634 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 float64
11  Base MSRP            112634 non-null object
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 [9]: nyc.isnull().sum()
Out[9]:
VIN (1-10)      0
County          0
City            0
State           0
Postal Code     0
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    286
DOL Vehicle ID        0
Vehicle Location      24
Electric Utility      443
2020 Census Tract     0
dtype: int64

In [10]: nyc.dropna(inplace=True)

In [11]: nyc.isnull().sum()
Out[11]:
VIN (1-10)      0
County          0
City            0
State           0
Postal Code     0
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    0
DOL Vehicle ID        0
Vehicle Location      0
Electric Utility      0
2020 Census Tract     0
dtype: int64

In [12]: nyc.info()

<class 'pandas.core.frame.DataFrame'>
Index: 112152 entries, 2 to 112633
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  -
0   VIN (1-10)          112152 non-null object
1   County              112152 non-null object
2   City                112152 non-null object
3   State               112152 non-null object
4   Postal Code         112152 non-null int64
5   Model Year          112152 non-null int64
6   Make                112152 non-null object
7   Model               112152 non-null object
8   Electric Vehicle Type 112152 non-null object
9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 112152 non-null object
10  Electric Range       112152 non-null int64
11  Base MSRP            112152 non-null int64
12  Legislative District  112152 non-null float64
13  DOL Vehicle ID       112152 non-null int64
14  Vehicle Location     112152 non-null object
15  Electric Utility      112152 non-null object
16  2020 Census Tract    112152 non-null int64
dtypes: float64(1), int64(6), object(10)
memory usage: 15.4+ MB
```

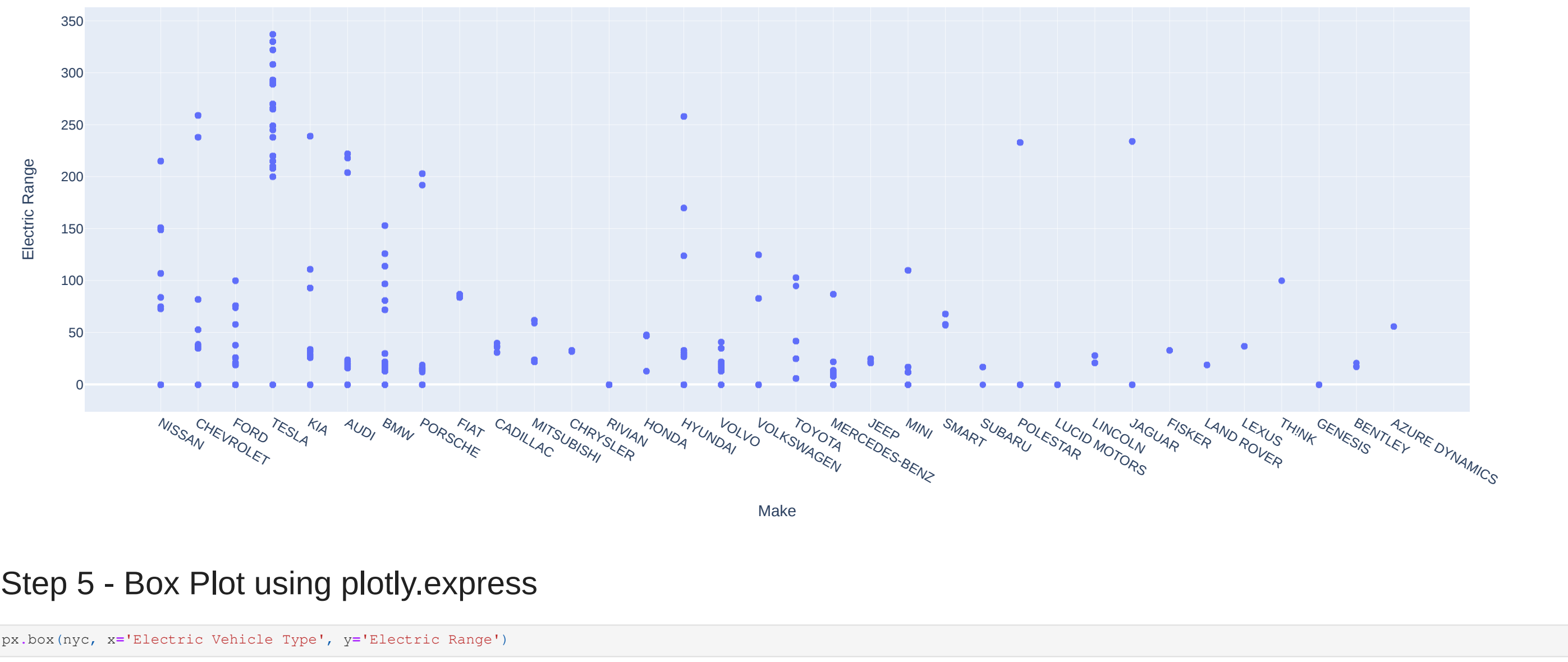
Step 3: Import required library - plotly.express

```
In [13]: import plotly.express as px
```

Step 4: Scatter plot using plotly.express

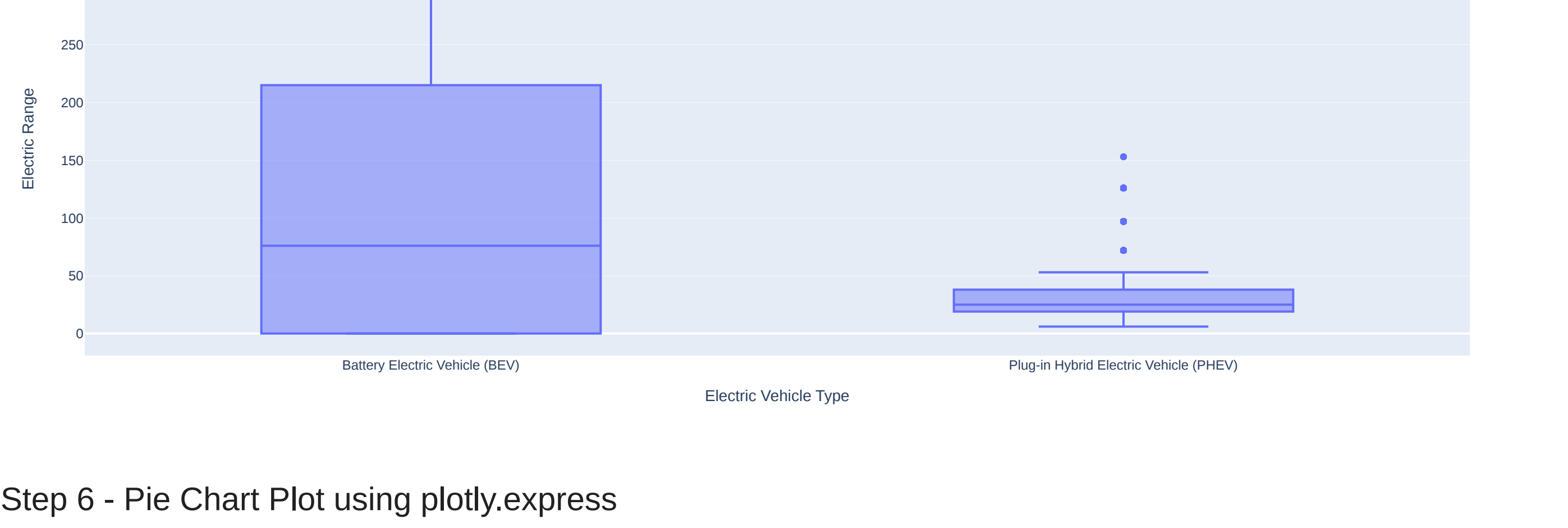
Task 1: This is an open ended problem. Apply Exploratory Data Analysis (Univariate and Bivariate) on the dataset available above.

```
In [14]: px.scatter(nyc, x = "Make", y = "Electric Range")
```



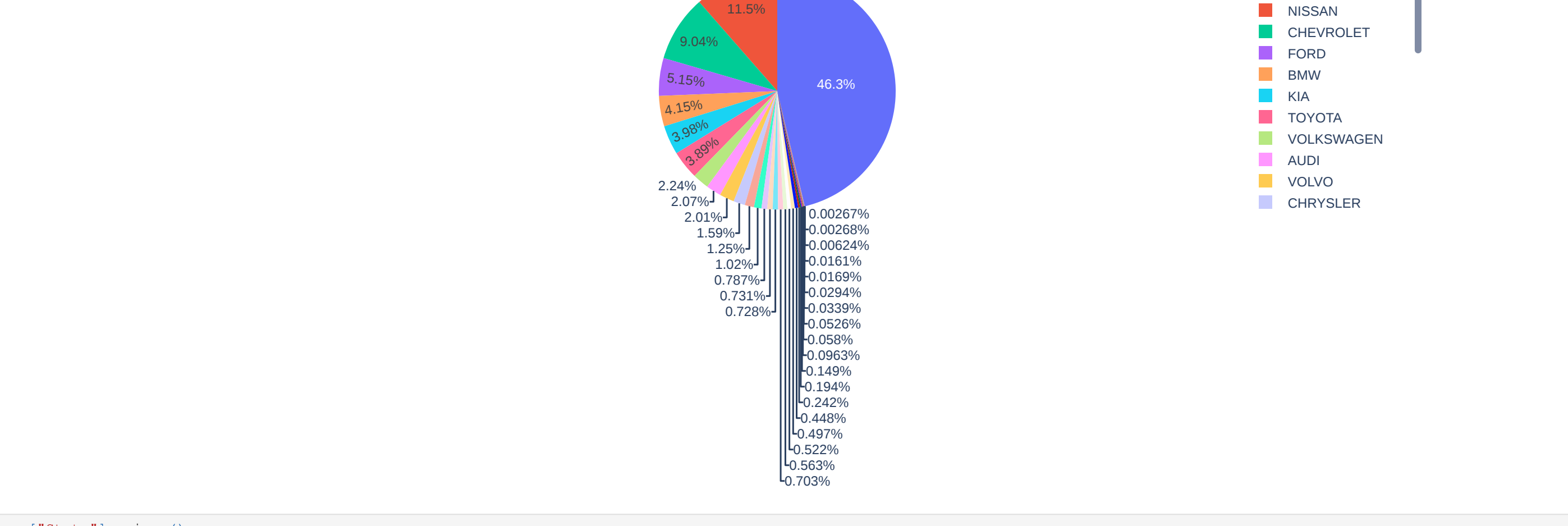
Step 5 - Box Plot using plotly.express

```
In [15]: px.box(nyc, x="Electric Vehicle Type", y="Electric Range")
```



Step 6 - Pie Chart Plot using plotly.express

```
In [16]: px.pie(nyc, names = "Make", values="2020 Census Tract")
```



```
In [17]: nyc["State"].unique()
Out[17]: array(['WA'], dtype=object)
```

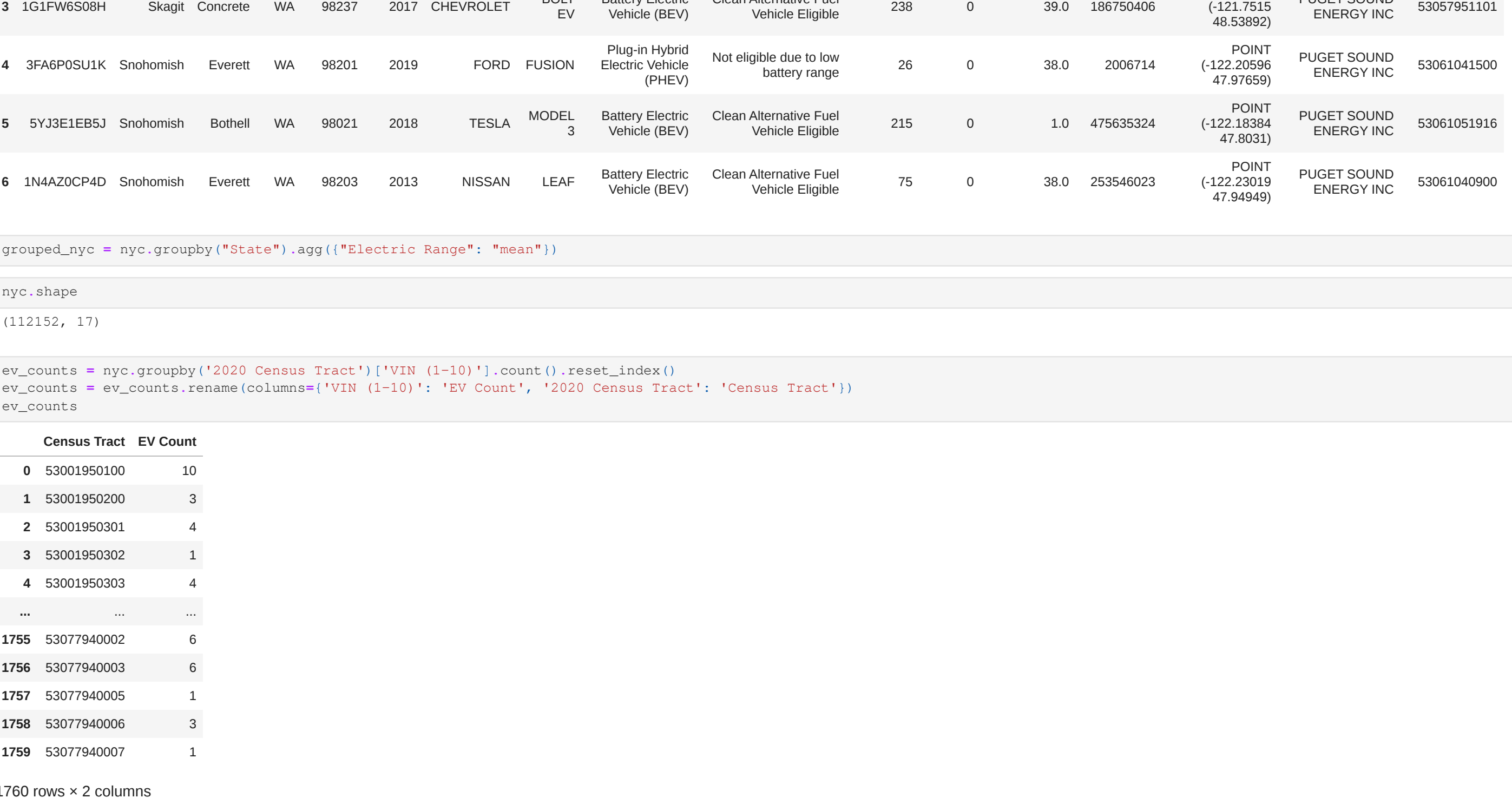
```
In [18]: nyc.head()
Out[18]:
```

	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	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
2	JN1A20CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	218972519	POINT (-120.50721 46.60446)	PACIFICORP	53077001602
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	186750406	POINT (-121.7515 48.53892)	PUGET SOUND ENERGY INC	53057951101
4	3FAP6PSUJK	Snohomish	Everett	WA	98201	2019	FORD	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	26	0	38.0	2006714	POINT (-122.20596 47.97659)	PUGET SOUND ENERGY INC	53061041500
5	5Y3E1EB5J	Snohomish	Bothell	WA	98021	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	1.0	475635324	POINT (-122.18384 47.8031)	PUGET SOUND ENERGY INC	53001051916
6	1N4A20CP4D	Snohomish	Everett	WA	98203	2013	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	75	0	38.0	253549023	POINT (-122.20019 47.94940)	PUGET SOUND ENERGY INC	53061040900

```
In [19]: grouped_nyc = nyc.groupby(["State"]).agg({"Electric Range": "mean"})
```

```
In [20]: nyc.shape
Out[20]: (112152, 17)
```

```
In [21]: ev_counts = nyc.groupby('2020 Census Tract')['VIN (1-10)'].count().reset_index()
ev_counts = ev_counts.rename(columns={'VIN (1-10)': 'EV Count', '2020 Census Tract': 'Census Tract'})
ev_counts
```



```
In [22]: ev_count_by_pincode = nyc.groupby(['Postal Code', 'Model Year', 'State']).size().reset_index(name='Number_of_EV_Vehicles')
In [23]: state_nyc= ev_count_by_pincode[ev_count_by_pincode['State'] == 'WA']
```

Choropleth Map

```
In [24]: import json
```

```
In [25]: fig = px.choropleth_mapbox(state_nyc,
                                geojson=C:\Users\Owner\Downloads\dataset.csv",
                                locations="Postal Code",
                                color="Number_of_EV_Vehicles",
                                featureidkey="properties.XCPACZ10",
                                mapbox_style="carto-positron",
                                zoom=5,
                                center=[lat: 47.7511, "lon": -120.7401],
                                titles="Number of EV Vehicles based on location Washington Over Time",
                                animation_frame="Model Year",
                                color_continuous_scale="Viridis",
                                hover_data=["Number_of_EV_Vehicles"])

# Update layout for aesthetics
fig.update_layout(margin={"t": 0, "b": 0, "l": 0, "r": 0})

# Show the animated map
fig.show()
```

C:\Users\Owner\Desktop\alib\site-packages\plotly\express_core.py:1979: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass '(name,)' instead of 'name' to silence this warnin



Task 3: Create a Racing Bar Plot to display the animation of EV Make and its count each year.

Racing Bar Plot

```
In [26]: !pip install bar-chart-race
```

Requirement already satisfied: bar-chart-race in c:\users\owner\desktop\alib\site-packages (0.1.0)

Requirement already satisfied: pandas>=0.24 in c:\users\owner\desktop\alib\site-packages (from bar-chart-race) (2.2.1)

Requirement already satisfied: matplotlib>3.1 in c:\users\owner\desktop\alib\site-packages (from bar-chart-race) (3.7.1)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (1.0.5)

Requirement already satisfied: numpy>=1.20 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (0.11.0)

Requirement already satisfied: packaging>=20.0 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (4.25.0)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (4.25.0)

Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (1.4.4)

Requirement already satisfied: pillow>=6.2.0 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (9.4.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (3.0.9)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\owner\desktop\alib\site-packages (from matplotlib>3.1->bar-chart-race) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\owner\desktop\alib\site-packages (from pandas>=0.24->bar-chart-race) (2024.1)

Requirement already satisfied: tzdata>=2022.7 in c:\users\owner\desktop\alib\site-packages (from pandas>=0.24->bar-chart-race) (2024.1)

Requirement already satisfied: six>=1.5 in c:\users\owner\desktop\alib\site-packages (from python-dateutil>=2.7->matplotlib>3.1->bar-chart-race) (1.16.0)

Step 2 - Import required library - bar_chart_race

```
In [27]: import bar_chart_race as bcr
```

```
In [28]: df = nyc.groupby(['Make', 'Model Year']).size().reset_index(name='Number_of_Vehicles')
```

```
In [29]: # Create the animated racing bar plot with annotations
fig = px.bar(df,
             y="Make", # Place Make on y-axis
             x="Number_of_Vehicles", # Place the count of EV vehicles on the x-axis
             color="Make", # Color each make differently
             animation_frame="Model Year", # Create animation by year
             orientation="h", # Horizontal bar chart
             title="EV Makes and their Count Over the Years",
             labels={"Number_of_Vehicles": "Number of EV Vehicles"},
             range_x=(0, 3000)
             )
```

```
In [30]: fig.update_traces(texttemplate='%{x}', # Display the actual x-axis values (Number_of_Vehicles)
                        textposition='outside', # Place the text outside the bars
                        textfont_size=17) # Adjust the font size for better readability
```

Adjust the layout for improved visibility and emphasis on movement

fig.update_layout(
 xaxis=dict(showgrid=True, gridcolor='LightGray'), # Show grid for better visibility
 yaxis_title='EV Makes',
 xaxis_title='Number of EV Vehicles',
 showlegend=False, # Hide legend as it's not necessary for this chart
 title_x=0.5, # Center title
 title_fontsize=20, # Increase title font size
 margin=dict(l=50, r=50, t=50, b=50), # Adjust margins
 width=800, # Set a fixed width
 height=600 # Set a fixed height
)

Show the plot
fig.show()



```
In [ ]:
```