In [1]:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import numpy as np

# For animation
from IPython.display import HTML
```

In [2]:

```
file_path = r'C:\Users\P.DEEPIKA\Downloads\dataset.csv'

dataset = pd.read_csv(file_path)
```

In [3]:

```
dataset.head()
```

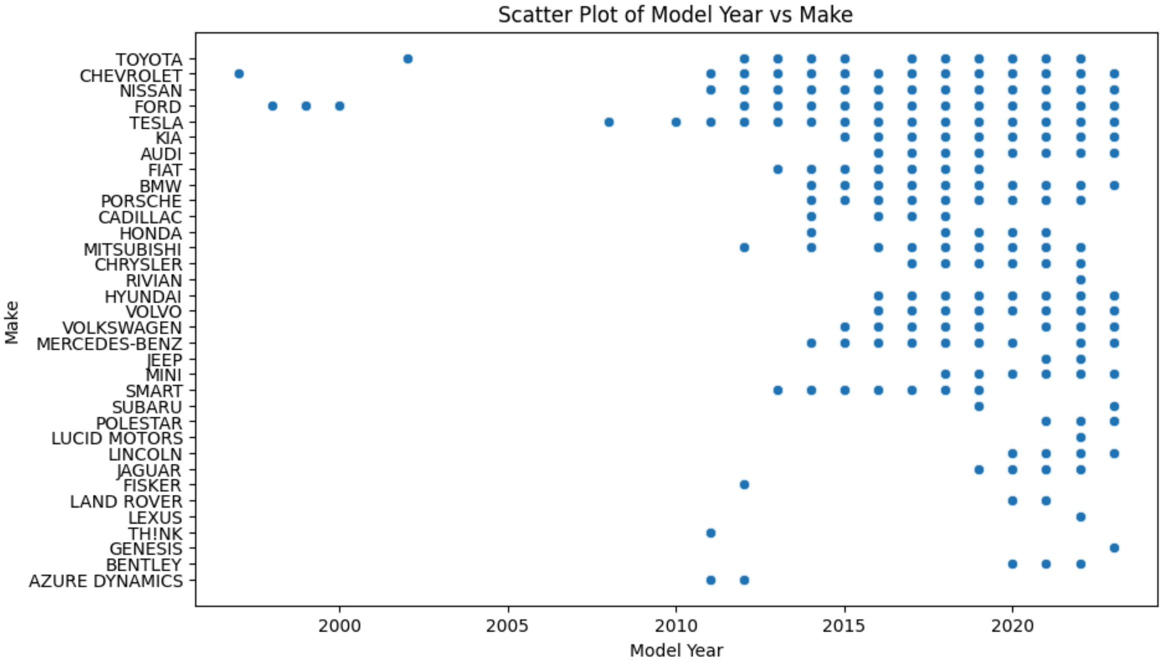
Out[3]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Elect Vehi Ty
0	JTMEB3FV6N	Monroe	Key West	FL	33040	2022	TOYOTA	RAV4 PRIME	Plug Hyb Elect Vehi (PHE
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug Hyb Elect Vehi (PHE
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Batte Elect Vehi (BE
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Batte Elect Vehi (BE
4	3FA6P0SU1K	Snohomish	Everett	WA	98201	2019	FORD	FUSION	Plug Hyb Elect Vehi (PHE

In [4]:

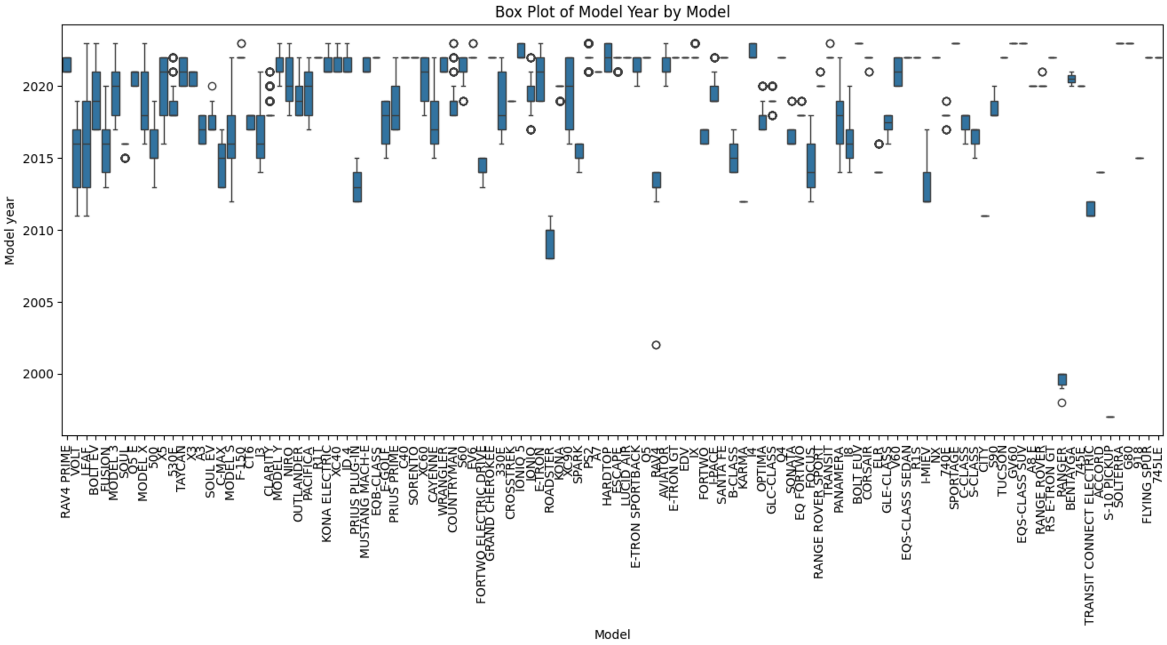
```
# Scatter plot for Model Year vs Make
plt.figure(figsize=(10,6))
sns.scatterplot(x='Model Year', y='Make', data=dataset)
```

```
plt.title('Scatter Plot of Model Year vs Make')
plt.xlabel('Model Year')
plt.ylabel('Make')
plt.show()
```



In [5]:

```
# Box plot for Model Year by Model
plt.figure(figsize=(16,6))
sns.boxplot(x='Model', y='Model Year', data=dataset)
plt.xticks(rotation=90)
plt.title('Box Plot of Model Year by Model')
plt.xlabel('Model')
plt.ylabel('Model year')
plt.show()
```



In [6]:

```
# Pie chart showing distribution of EV Make
make_count = dataset['Make'].value_counts()

# Plotting pie chart
fig = go.Figure(data=[go.Pie(labels=make_count.index, values=make_count.values)]
fig.update_layout(title_text='Distribution of Electric Vehicle Makes')
fig.show()
```

In [7]:

```
dataset['State'].nunique()
```

Out[7]:

```
45
```

In [8]:

```
ev_count_by_state = dataset.groupby('State').size().reset_index(name='EV_Count')
```

In [15]:

```
fig = px.choropleth(
    ev_count_by_state,
    locations='State',
    locationmode='USA-states',
    color='EV_Count',
    scope='usa',
    labels={'EV-Count': 'Number of EVs'},
    title='Number of Electric Vehicles by State'
)

fig.update_layout(
    title_x=0.5,
)

fig.show()
```

In [16]:

```
ev_count_by_pincode = dataset.groupby(['Postal Code', 'Model Year', 'State']).si
state_data = ev_count_by_pincode[ev_count_by_pincode['State'] == 'WA']

fig = px.choropleth_mapbox(
    state_data,
    geojson="https://raw.githubusercontent.com/OpenDataDE/State-zip-code-GeoJSON
    locations='Postal Code',
    color='Number_of_EV_Vehicles',
    featureidkey="properties.ZCTA5CE10",
    mapbox_style="carto-positron",
    zoom=5,
    center={"lat": 47.7511, "lon": -120.7401},
    animation_frame="Model Year",
    color_continuous_scale="Cividis",
    hover_data=['Number_of_EV_Vehicles']
)

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

fig.show()
```

In [76]:

```
import bar_chart_race as bcr
```

In [77]:

```
df = dataset.groupby(['Make','Model Year']).size().reset_index(name='Number_of_V
```

In [78]:

```
# Creating  the animated Racing Bar Plot
fig=px.bar(df,
           y='Make',
           x='Number_of_Vehicles',
           color='Make',
           animation_frame='Model Year',
           orientation='h',
           title='EV Makes and their Count Over the Years',
           labels={'Number_of_Vehicles': 'Number of EV Vehicles'},
           range_x=[0,3000])
```

In [79]:

```
fig.update_layout(xaxis=dict(showgrid=True,  gridcolor='LightGray'),
                  yaxis_title='EV Makes',
                  xaxis_title='Number of EV Vehicles',
                  showlegend=False,
                  title_x=0.9,
                  title_font=dict(size=20),
                  margin=dict(l=50, r=50, t=50, b=50),
                  width=800,
                  height=600)

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