

untitled-1

October 4, 2024

```
[1]: # import all necessary libraries
import pandas as pd
import plotly.express as ex
import numpy as np
```

```
[2]: # provide the path for dataset
PATH=r"C:\Users\LOKESH\Downloads\dataset.csv"
```

```
[3]: # load the data into dataframe
df=pd.read_csv(PATH)
```

```
[4]: # load the top 5 values from dataframe
df.head()
```

```
[4]:
```

	VIN (1-10)	County	City	State	Postal Code	Model	Year	Make	\
0	JTMEB3FV6N	Monroe	Key West	FL	33040		2022	TOYOTA	
1	1G1RD6E45D	Clark	Laughlin	NV	89029		2013	CHEVROLET	
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901		2011	NISSAN	
3	1G1FW6S08H	Skagit	Concrete	WA	98237		2017	CHEVROLET	
4	3FA6P0SU1K	Snohomish	Everett	WA	98201		2019	FORD	

	Model	Electric Vehicle Type	\
0	RAV4 PRIME	Plug-in Hybrid Electric Vehicle (PHEV)	
1	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	
2	LEAF	Battery Electric Vehicle (BEV)	
3	BOLT EV	Battery Electric Vehicle (BEV)	
4	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	

	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	\
0	Clean Alternative Fuel Vehicle Eligible	42	
1	Clean Alternative Fuel Vehicle Eligible	38	
2	Clean Alternative Fuel Vehicle Eligible	73	
3	Clean Alternative Fuel Vehicle Eligible	238	
4	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

```
[5]: # get the number of rows and columns
df.shape
```

```
[5]: (112634, 17)
```

```
[6]: # get the names of all the columns
df.columns
```

```
[6]: Index(['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'],
          dtype='object')
```

```
[7]: # get the data types and count of non null values for all the columns
df.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  int64
5   Model Year                                    112634 non-null  int64
6   Make                                           112634 non-null  object
7   Model                                          112614 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  int64
11  Base MSRP                                     112634 non-null  int64
```

```

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

```

Observation:

The dataset contains the details about the electric vehicles. There are 112634 rows and 17 columns in the dataframe. There are no special characters in the column names. The datatypes of the columns are appropriate as per the values in the column. There are some missing values in the “Model”, “Legislative District”, “Vehicle Location”, “Electric Utility”. The missing values are treated below using the missing values treatment.

Missing Values Treatment

```

[8]: # replace the missing values in Model with mode
df['Model']=df['Model'].fillna(df['Model'].mode())

```

```

[9]: # find mean
df['Legislative District'].mean()

```

```

[9]: 29.805604016092854

```

```

[10]: # find median
df['Legislative District'].median()

```

```

[10]: 34.0

```

```

[11]: # replace the missing values in Legislative District with median as there is
      ↪ variation between mean and median
df['Legislative District']=df['Legislative District'].fillna(df['Legislative_
      ↪ District'].median())

```

```

[12]: # find mode
df['Vehicle Location'].mode()

```

```

[12]: 0    POINT (-122.13158 47.67858)
      Name: Vehicle Location, dtype: object

```

```

[13]: # replace Nan values with np.nan
df['Vehicle Location']=df['Vehicle Location'].replace('NaN',np.nan)

```

```

[14]: # find sum of np.nan values
df['Vehicle Location'].isna().sum()

```

```

[14]: 24

```

```
[15]: # replace np.nan value with mode
df['Vehicle Location']=df['Vehicle Location'].fillna(df['Vehicle Location'].
↳mode()[0])
```

```
[16]: # replace Nan values with np.nan
df['Electric Utility']=df['Electric Utility'].replace('NaN',np.nan)
```

```
[17]: # find sum of np.nan values
df['Electric Utility'].isna().sum()
```

```
[17]: 443
```

```
[18]: # find mode
df['Electric Utility'].mode()
```

```
[18]: 0    PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
      Name: Electric Utility, dtype: object
```

```
[19]: # replace np.nan values with mode
df['Electric Utility']=df['Electric Utility'].fillna(df['Electric Utility'].
↳mode()[0])
```

```
[20]: df.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  int64
5   Model Year                              112634 non-null  int64
6   Make                                    112634 non-null  object
7   Model                                   112614 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  int64
11  Base MSRP                               112634 non-null  int64
12  Legislative District                     112634 non-null  float64
13  DOL Vehicle ID                           112634 non-null  int64
14  Vehicle Location                         112634 non-null  object
15  Electric Utility                         112634 non-null  object
16  2020 Census Tract                       112634 non-null  int64
dtypes: float64(1), int64(6), object(10)
```

memory usage: 14.6+ MB

Task1: Univariate and Bivariate Analysis using Plotly Express

Univariate

```
[21]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='VIN (1-10)', title='Count Plot of VIN (1-10)')

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

```
[22]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='County', title='Count Plot of County')

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

```
[23]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='City', title='Count Plot of City')

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

```
[24]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='State', title='Count Plot of State')

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

```
[25]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Postal Code', title='Count Plot of Postal Code')

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

```
[26]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Model Year', title='Count Plot of Model Year')

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

```
[27]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Make', title='Count Plot of Make')

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

```
[28]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Model', title='Count Plot of Model')

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

```
[29]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Electric Vehicle Type', title='Count Plot of Electric_
↳Vehicle Type')

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

```
[30]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Clean Alternative Fuel Vehicle (CAFV) Eligibility',
↳title='Count Plot of Clean Alternative Fuel Vehicle (CAFV) Eligibility')

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

```
[32]: # box plot
fig = ex.box(df, x="Electric Range")
fig.show()
```

```
[33]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Vehicle Location', title='Count Plot of Vehicle_
↳Location')

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

```
[34]: # Create the count plot using Plotly Express
fig = ex.histogram(df, x='Electric Utility', title='Count Plot of Electric_
↳Utility')

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

Bivariate Analysis:

Electric Range vs Model Year:

```
[35]: # scatter plot
fig = ex.scatter(x=df['Electric Range'],y=df['Model Year'])
fig.show()
```

```
[36]: fig = ex.scatter(x=df['Base MSRP'],y=df['Model Year'])
fig.show()
```

```
[37]: # box plot
fig=ex.box(df,x='Electric Vehicle Type',y='Base MSRP')
fig.show()
```

```
[38]: # box plot
fig=ex.box(df,x='Electric Vehicle Type',y='Electric Range')
fig.show()
```

```
[39]: # box plot
fig=ex.box(df,x='Clean Alternative Fuel Vehicle (CAFV) Eligibility',y='Base_
↳MSRP')
fig.show()
```

```
[40]: # Create a stacked bar plot
fig = ex.box(df, y='Legislative District', x='Electric Vehicle Type',
            title='Legislative District vs. Electric Vehicle Type')

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

Task2:Choropleth to display the number of EV vehicles based on location

```
[41]: # Aggregate electric vehicle count by State
electric_vehicle_count_by_state = df.groupby(['State','Model Year']).size().
↳reset_index(name='EV Count')

# Create the Choropleth plot
fig = ex.choropleth(
    electric_vehicle_count_by_state,
    locations='State',
    locationmode='USA-states',
    color='EV Count',
    scope='usa',
    color_continuous_scale='Viridis',
    labels={'EV Count': 'Electric Vehicle Count'},
    hover_name='State',
    hover_data=['EV Count'],
    title='Electric Vehicle Count by State and Model Year',
    animation_frame='Model Year'
)

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

Observation:

From the above plot, we can observe the number of electric vehicles in thousands in each state in the particular Model year. The highest number of electric vehicles are found in WA state in 2017 with the count of 8620. Similarly the count of the electric vehicles in each state can be determined in every model year.

Task3:Racing Bar Plot to display the animation of EV Make and its count each year.

```
[42]: !pip install bar-chart-race
```

```
Collecting bar-chart-race
```

```
Obtaining dependency information for bar-chart-race from https://files.pythonhosted.org/packages/09/01/f6d1a1a0978b39560843c54be7349804d7d2faef0a869acd7c8a6fc920b0/bar_chart_race-0.1.0-py3-none-any.whl.metadata
```

```
Downloading bar_chart_race-0.1.0-py3-none-any.whl.metadata (4.2 kB)
```

```
Requirement already satisfied: pandas>=0.24 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from bar-chart-race) (2.0.3)
```

```
Requirement already satisfied: matplotlib>=3.1 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from bar-chart-race) (3.7.2)
```

```
Requirement already satisfied: contourpy>=1.0.1 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.0.5)
```

```
Requirement already satisfied: cycler>=0.10 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (0.11.0)
```

```
Requirement already satisfied: fonttools>=4.22.0 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (4.25.0)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.4.4)
```

```
Requirement already satisfied: numpy>=1.20 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.24.3)
```

```
Requirement already satisfied: packaging>=20.0 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (23.1)
```

```
Requirement already satisfied: pillow>=6.2.0 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (10.0.1)
```

```
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (3.0.9)
```

```
Requirement already satisfied: python-dateutil>=2.7 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (2.8.2)
```

```
Requirement already satisfied: pytz>=2020.1 in
```

```
c:\users\lokes\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
```



```
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\lokesesh\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race)
(2023.3)
Requirement already satisfied: six>=1.5 in c:\users\lokesesh\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.1->bar-chart-race) (1.16.0)
Downloading bar_chart_race-0.1.0-py3-none-any.whl (156 kB)
----- 0.0/156.8 kB ? eta -:-:--
-- ----- 10.2/156.8 kB ? eta -:-:--
----- 30.7/156.8 kB 325.1 kB/s eta 0:00:01
----- 133.1/156.8 kB 871.5 kB/s eta 0:00:01
----- 156.8/156.8 kB 852.2 kB/s eta 0:00:00
Installing collected packages: bar-chart-race
Successfully installed bar-chart-race-0.1.0
```

```
[47]: import bar_chart_race as bcr
import warnings
```

```
[53]: pip install imageio[ffmpeg]
```

```
Requirement already satisfied: imageio[ffmpeg] in
c:\users\lokesesh\anaconda3\lib\site-packages (2.26.0)
Requirement already satisfied: numpy in c:\users\lokesesh\anaconda3\lib\site-
packages (from imageio[ffmpeg]) (1.24.3)
Requirement already satisfied: pillow>=8.3.2 in
c:\users\lokesesh\anaconda3\lib\site-packages (from imageio[ffmpeg]) (10.0.1)
Collecting imageio-ffmpeg (from imageio[ffmpeg])
  Obtaining dependency information for imageio-ffmpeg from https://files.pythonh
  osted.org/packages/a9/1c/1b9c72bf839def47626436ea5ebaf643404f7850482c5fafd71a3de
  eaa94/imageio_ffmpeg-0.5.1-py3-none-win_amd64.whl.metadata
  Downloading imageio_ffmpeg-0.5.1-py3-none-win_amd64.whl.metadata (1.6 kB)
Requirement already satisfied: psutil in c:\users\lokesesh\anaconda3\lib\site-
packages (from imageio[ffmpeg]) (5.9.0)
Requirement already satisfied: setuptools in c:\users\lokesesh\anaconda3\lib\site-
packages (from imageio-ffmpeg->imageio[ffmpeg]) (68.0.0)
Downloading imageio_ffmpeg-0.5.1-py3-none-win_amd64.whl (22.6 MB)
----- 0.0/22.6 MB ? eta -:-:--
----- 0.0/22.6 MB ? eta -:-:--
----- 0.0/22.6 MB ? eta -:-:--
----- 0.0/22.6 MB 262.6 kB/s eta 0:01:27
----- 0.1/22.6 MB 525.1 kB/s eta 0:00:43
----- 0.3/22.6 MB 1.5 MB/s eta 0:00:15
- ----- 0.8/22.6 MB 3.2 MB/s eta 0:00:07
- ----- 0.9/22.6 MB 3.6 MB/s eta 0:00:06
-- ----- 1.6/22.6 MB 4.5 MB/s eta 0:00:05
--- ----- 2.0/22.6 MB 5.4 MB/s eta 0:00:04
---- ----- 2.5/22.6 MB 5.9 MB/s eta 0:00:04
----- 2.9/22.6 MB 6.0 MB/s eta 0:00:04
```

```

----- 3.3/22.6 MB 6.2 MB/s eta 0:00:04
----- 3.9/22.6 MB 6.7 MB/s eta 0:00:03
----- 4.3/22.6 MB 6.9 MB/s eta 0:00:03
----- 4.7/22.6 MB 7.2 MB/s eta 0:00:03
----- 5.4/22.6 MB 7.4 MB/s eta 0:00:03
----- 5.8/22.6 MB 7.6 MB/s eta 0:00:03
----- 6.4/22.6 MB 7.8 MB/s eta 0:00:03
----- 6.9/22.6 MB 8.1 MB/s eta 0:00:02
----- 7.4/22.6 MB 8.3 MB/s eta 0:00:02
----- 7.9/22.6 MB 8.4 MB/s eta 0:00:02
----- 8.4/22.6 MB 8.5 MB/s eta 0:00:02
----- 8.9/22.6 MB 8.6 MB/s eta 0:00:02
----- 9.3/22.6 MB 8.6 MB/s eta 0:00:02
----- 9.9/22.6 MB 8.8 MB/s eta 0:00:02
----- 10.4/22.6 MB 10.2 MB/s eta 0:00:02
----- 10.9/22.6 MB 10.4 MB/s eta 0:00:02
----- 11.4/22.6 MB 10.7 MB/s eta 0:00:02
----- 11.9/22.6 MB 10.6 MB/s eta 0:00:02
----- 12.5/22.6 MB 10.9 MB/s eta 0:00:01
----- 13.0/22.6 MB 10.9 MB/s eta 0:00:01
----- 13.5/22.6 MB 10.9 MB/s eta 0:00:01
----- 14.0/22.6 MB 10.9 MB/s eta 0:00:01
----- 14.6/22.6 MB 11.1 MB/s eta 0:00:01
----- 15.1/22.6 MB 11.3 MB/s eta 0:00:01
----- 15.6/22.6 MB 11.1 MB/s eta 0:00:01
----- 16.2/22.6 MB 10.9 MB/s eta 0:00:01
----- 16.8/22.6 MB 11.1 MB/s eta 0:00:01
----- 17.3/22.6 MB 11.1 MB/s eta 0:00:01
----- 17.9/22.6 MB 11.1 MB/s eta 0:00:01
----- 18.4/22.6 MB 11.1 MB/s eta 0:00:01
----- 18.9/22.6 MB 11.1 MB/s eta 0:00:01
----- 19.5/22.6 MB 11.3 MB/s eta 0:00:01
----- 20.0/22.6 MB 11.3 MB/s eta 0:00:01
----- 20.5/22.6 MB 11.3 MB/s eta 0:00:01
----- 21.0/22.6 MB 11.3 MB/s eta 0:00:01
----- 21.5/22.6 MB 11.3 MB/s eta 0:00:01
----- 22.1/22.6 MB 11.3 MB/s eta 0:00:01
----- 22.5/22.6 MB 11.3 MB/s eta 0:00:01
----- 22.6/22.6 MB 11.1 MB/s eta 0:00:01
----- 22.6/22.6 MB 10.2 MB/s eta 0:00:00

```

Installing collected packages: imageio-ffmpeg

Successfully installed imageio-ffmpeg-0.5.1

Note: you may need to restart the kernel to use updated packages.

```
[56]: EXTENSION = {
      '.png': 'PNG',
      '.jpg': 'JPEG',
```

```

    '.jpeg': 'JPEG',
    '.bmp': 'BMP',
    '.gif': 'GIF',
    '.tiff': 'TIFF',
    '.mp4': 'MP4', # Add this line if it's missing
}

```

```
[61]: conda install -c conda-forge ffmpeg
```

```

Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

```

```
## Package Plan ##
```

```
environment location: c:\Users\LOKESH\anaconda3
```

```

added / updated specs:
- ffmpeg

```

The following packages will be downloaded:

package	build		
ca-certificates-2024.8.30	h56e8100_0	155 KB	conda-forge
certifi-2024.8.30	pyhd8ed1ab_0	160 KB	conda-forge
ffmpeg-4.3.1	ha925a31_0	26.2 MB	conda-forge
Total:		26.5 MB	

The following NEW packages will be INSTALLED:

```
ffmpeg                conda-forge/win-64::ffmpeg-4.3.1-ha925a31_0
```

The following packages will be UPDATED:

```

ca-certificates      pkgs/main::ca-certificates-2023.12.12~ --> conda-forge::ca-
certificates-2024.8.30-h56e8100_0
certifi              pkgs/main/win-64::certifi-2023.11.17~~ --> conda-
forge/noarch::certifi-2024.8.30-pyhd8ed1ab_0

```

Downloading and Extracting Packages

```
ca-certificates-2024 | 155 KB | 0%
```

ffmpeg-4.3.1	26.2 MB		0%
certifi-2024.8.30	160 KB		0%
certifi-2024.8.30	160 KB	#	10%
ffmpeg-4.3.1	26.2 MB		0%
ca-certificates-2024	155 KB	#	10%
ffmpeg-4.3.1	26.2 MB	2	3%
ca-certificates-2024	155 KB	#####	100%
ca-certificates-2024	155 KB	#####	100%
certifi-2024.8.30	160 KB	#####	100%
certifi-2024.8.30	160 KB	#####	100%
ffmpeg-4.3.1	26.2 MB	5	6%
ffmpeg-4.3.1	26.2 MB	#	10%
ffmpeg-4.3.1	26.2 MB	#7	17%
ffmpeg-4.3.1	26.2 MB	##1	21%
ffmpeg-4.3.1	26.2 MB	##5	25%
ffmpeg-4.3.1	26.2 MB	##9	30%
ffmpeg-4.3.1	26.2 MB	###3	34%
ffmpeg-4.3.1	26.2 MB	###7	38%
ffmpeg-4.3.1	26.2 MB	####2	42%
ffmpeg-4.3.1	26.2 MB	####6	46%
ffmpeg-4.3.1	26.2 MB	#####	50%
ffmpeg-4.3.1	26.2 MB	#####4	55%
ffmpeg-4.3.1	26.2 MB	#####8	59%
ffmpeg-4.3.1	26.2 MB	#####2	63%

ffmpeg-4.3.1	26.2 MB	#####6	67%
ffmpeg-4.3.1	26.2 MB	#####	71%
ffmpeg-4.3.1	26.2 MB	#####5	75%
ffmpeg-4.3.1	26.2 MB	#####9	79%
ffmpeg-4.3.1	26.2 MB	#####3	83%
ffmpeg-4.3.1	26.2 MB	#####7	87%
ffmpeg-4.3.1	26.2 MB	#####1	91%
ffmpeg-4.3.1	26.2 MB	#####5	95%
ffmpeg-4.3.1	26.2 MB	#####	100%
ffmpeg-4.3.1	26.2 MB	#####	100%

Preparing transaction: ...working... done
 Verifying transaction: ...working... done
 Executing transaction: ...working... done

Note: you may need to restart the kernel to use updated packages.

==> WARNING: A newer version of conda exists. <==
 current version: 23.7.4
 latest version: 24.9.1

Please update conda by running

```
$ conda update -n base -c defaults conda
```

Or to minimize the number of packages updated during conda update use

```
conda install conda=24.9.1
```

```
[63]: # Group the data by Model Year and Make to get the count of each EV Make for
      ↪ each year
ev_make_count_by_year = df.groupby(['Model Year', 'Make']).size().
      ↪ reset_index(name='Count')

ev_make_count_pivot = ev_make_count_by_year.pivot(index='Model Year',
      ↪ columns='Make', values='Count')

# Create the Racing Bar Plot
bcr.bar_chart_race(df=ev_make_count_pivot,
      ↪ filename='ev_make_count_racing_bar_plot.mp4')
```

```
[64]: from IPython.display import HTML, Video
```

```
[66]: video_path = '/content/ev_make_count_racing_bar_plot.mp4'
      Video(video_path, embed=True)
```

```
[66]: <IPython.core.display.Video object>
```