

electric-vehicle-project

October 8, 2024

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

Requirement already satisfied: plotly in c:\users\owner\desktop\a\lib\site-packages (5.9.0)

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

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

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

1 Step 2 : Reading the csv data into a dataframe

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

```
[4]: nyc.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	3FA6POSU1K	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]: nyc.shape
```

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

```
[6]: nyc["Electric Vehicle Type"].unique()
```

```
[6]: array(['Plug-in Hybrid Electric Vehicle (PHEV)',
        'Battery Electric Vehicle (BEV)'], dtype=object)
```

```
[7]: nyc["Make"].unique()
```

```
[7]: array(['TOYOTA', 'CHEVROLET', 'NISSAN', 'FORD', 'TESLA', 'KIA', 'AUDI',
        'FIAT', 'BMW', 'PORSCH', 'CADILLAC', 'HONDA', 'MITSUBISHI',
        'CHRYSLER', 'RIVIAN', 'HYUNDAI', 'VOLVO', 'VOLKSWAGEN',
        'MERCEDES-BENZ', 'JEEP', 'MINI', 'SMART', 'SUBARU', 'POLESTAR',
        'LUCID MOTORS', 'LINCOLN', 'JAGUAR', 'FISKER', 'LAND ROVER',
        'LEXUS', 'TH!NK', 'GENESIS', 'BENTLEY', 'AZURE DYNAMICS'],
        dtype=object)
```

```
[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	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

```
[9]: nyc.isnull().sum()
```

```
[9]: VIN (1-10)          0
County                 0
City                   0
State                  0
Postal Code            0
Model Year             0
Make                   0
Model                  20
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
```

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

```
[11]: nyc.isnull().sum()
```

```
[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

```

```
[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
```

2 Step 3: Import required library - plotly.express

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

3 Step 4: Scatter plot using plotly.express

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

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

5 Step 5 - Box Plot using plotly.express

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

6 Step 6 - Pie Chart Plot using plotly.express

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

```
[17]: nyc["State"].unique()
```

```
[17]: array(['WA'], dtype=object)
```

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

```
[18]:
```

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	\
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	
4	3FA6POSU1K	Snohomish	Everett	WA	98201	2019	FORD	
5	5YJ3E1EB5J	Snohomish	Bothell	WA	98021	2018	TESLA	
6	1N4AZ0CP4D	Snohomish	Everett	WA	98203	2013	NISSAN	

	Model	Electric Vehicle Type	\
2	LEAF	Battery Electric Vehicle (BEV)	
3	BOLT EV	Battery Electric Vehicle (BEV)	
4	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	
5	MODEL 3	Battery Electric Vehicle (BEV)	
6	LEAF	Battery Electric Vehicle (BEV)	

	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	\
2	Clean Alternative Fuel Vehicle Eligible	73	
3	Clean Alternative Fuel Vehicle Eligible	238	

4	Not eligible due to low battery range	26
5	Clean Alternative Fuel Vehicle Eligible	215
6	Clean Alternative Fuel Vehicle Eligible	75

	Base MSRP	Legislative District	DOL Vehicle ID \
2	0	15.0	218972519
3	0	39.0	186750406
4	0	38.0	2006714
5	0	1.0	475635324
6	0	38.0	253546023

	Vehicle Location	Electric Utility	2020 Census Tract
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	POINT (-122.18384 47.8031)	PUGET SOUND ENERGY INC	53061051916
6	POINT (-122.23019 47.94949)	PUGET SOUND ENERGY INC	53061040900

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

```
[20]: nyc.shape
```

```
[20]: (112152, 17)
```

```
[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
```

```
[21]:
```

	Census Tract	EV Count
0	53001950100	10
1	53001950200	3
2	53001950301	4
3	53001950302	1
4	53001950303	4
...
1755	53077940002	6
1756	53077940003	6
1757	53077940005	1
1758	53077940006	3
1759	53077940007	1

```
[1760 rows x 2 columns]
```

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

```
[23]: state_nyc= ev_count_by_pincode[ev_count_by_pincode['State'] == 'WA']
```

7 Choropleth Map

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

```
[25]: fig = px.choropleth_mapbox(state_nyc,
                                geojson=r"C:\Users\Owner\Downloads\dataset.csv",
                                locations='Postal Code',
                                color='Number_of_EV_Vehicles',
                                featureidkey="properties.ZCTA5CE10",
                                mapbox_style="carto-positron",
                                zoom=5,
                                center={"lat": 47.7511, "lon": -120.7401},
                                title="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={"r": 0, "t": 0, "l": 0, "b": 0})

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

C:\Users\Owner\Desktop\a\Lib\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 warning.

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

9 Racing Bar Plot

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

Requirement already satisfied: bar-chart-race in
c:\users\owner\desktop\a\lib\site-packages (0.1.0)
Requirement already satisfied: pandas>=0.24 in
c:\users\owner\desktop\a\lib\site-packages (from bar-chart-race) (2.2.1)

Requirement already satisfied: matplotlib>=3.1 in
c:\users\owner\desktop\a\lib\site-packages (from bar-chart-race) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.0.5)
Requirement already satisfied: cycler>=0.10 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (23.0)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\owner\desktop\a\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\owner\desktop\a\lib\site-packages (from pandas>=0.24->bar-chart-race) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\owner\desktop\a\lib\site-packages (from pandas>=0.24->bar-chart-race) (2024.1)
Requirement already satisfied: six>=1.5 in c:\users\owner\desktop\a\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.1->bar-chart-race) (1.16.0)

10 Step 2 - Import required library - bar_chart_race

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

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

```
[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]
)

```

```

[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_font=dict(size=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()

```

```
[ ]:
```