Open in Colab

Assignment 7

Weeks 8 & 9 - Pandas

- In this homework assignment, you will explore and analyze a public dataset of your choosing. Since this assignment is “open-ended” in nature, you are free to expand upon the requirements below. However, you must meet the minimum requirments as indicated in each section.
- You must use Pandas as the **primary tool** to process your data.
- The preferred method for this analysis is in a .ipynb file. Feel free to use whichever platform of your choosing.
  - https://www.youtube.com/watch?v=inN8seMm7UI (Getting started with Colab).
- Your data should need some "work", or be considered "dirty". You must show your skills in data cleaning/wrangling.

Some data examples:

- https://www.data.gov/
- https://opendata.cityofnewyork.us/
- https://datasetsearch.research.google.com/
- https://archive.ics.uci.edu/ml/index.php

Resources:

- https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html
- https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

Headings or comments

You are required to make use of comments, or headings for each section. You must explain what your code is doing, and the results of running your code. Act as if you were giving this assignment to your manager - you must include clear and descriptive information for each section.

You may work as a group or indivdually on this assignment.

Introduction

In this section, please describe the dataset you are using. Include a link to the source of this data. You should also provide some explanation on why you choose this dataset.

Description

The data i chose to work with for this assignment is found here (https://catalog.data.gov/dataset/electric-vehicle-population-data) and the data covers information on the electric vehicles within Washington state. According to the description online the data "shows the Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) that are currently registered through Washington State Department of Licensing (DOL)".

Data Exploration

Import your dataset into your .ipynb, create dataframes, and explore your data.

Include:

- Summary statistics means, medians, quartiles,
- Missing value information
- Any other relevant information about the dataset.

In [40]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [ ]:
```
## Reading in the data from a local file
df = pd.read_csv("./data/Electric_Vehicle_Population_Data.csv")
print(df.head())
```

In [11]:
```
## Taking a look at the raw data
#### Analysis notes, for the point of this analysis i think taking limiting the sco
#### -  Limiting to Battery Electric Vehicles, no hybrids.
#### - County level count analysis by make
#### - Electric range by make
### Limiting to the relevant columns based on the analysis decisions.
lim_df = df[['County','Model Year','Make', 'Model','Electric Range']][df['Electric
print(lim_df.head())
```

```
     County  Model Year     Make    Model  Electric Range
0      King        2019    TESLA  MODEL 3           220.0
1    Kitsap        2020    TESLA  MODEL Y           291.0
2    Kitsap        2023  HYUNDAI  IONIQ 5             0.0
5  Thurston        2012    TESLA  MODEL S           265.0
6      King        2017      BMW       I3            81.0
```

In [16]:

```
## Now that we limited to the columns we would want for the proper analysis, we are
## Taling a look at the unqique values in each columnd. The one numberic value colu

## Getting dtype and other info.
print(lim_df.info())
print('----')
print(lim_df.describe())
print('----')
# Unique Values of non numeric
print("County")
print(lim_df["County"].unique())
print("Make")
print(lim_df["Make"].unique())
print("Model")
print(lim_df["Model"].unique())
## NO null values to deal with, moving on to aggregating for numbers to chart / loo
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 186998 entries, 0 to 235691
Data columns (total 5 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   County          186996 non-null  object
 1   Model Year      186998 non-null  int64
 2   Make            186998 non-null  object
 3   Model           186998 non-null  object
 4   Electric Range  186998 non-null  float64
dtypes: float64(1), int64(1), object(3)
memory usage: 8.6+ MB
None
----
          Model Year  Electric Range
count  186998.000000   186998.000000
mean     2021.635665       50.163799
std         2.784636       93.661931
min      2000.000000        0.000000
25%      2021.000000        0.000000
50%      2023.000000        0.000000
75%      2024.000000       73.000000
max      2025.000000      337.000000
----
County
['King' 'Kitsap' 'Thurston' 'Yakima' 'Snohomish' 'Island' 'Skagit' 'Grant'
 'Chelan' 'Whitman' 'Kittitas' 'Walla Walla' 'Stevens' 'Spokane'
 'Okanogan' 'Clark' 'Jefferson' 'Cowlitz' 'Clallam' 'Klickitat' 'Franklin'
 'Whatcom' 'Pierce' 'Benton' 'Skamania' 'San Juan' 'Grays Harbor'
 'Wahkiakum' 'Mason' 'Lewis' 'Douglas' 'Pacific' 'Asotin' 'San Mateo'
 'Lincoln' 'Pend Oreille' 'Adams' 'Howard' 'Beaufort' 'Wake' 'San Diego'
 'Calvert' 'Columbia' 'Santa Clara' 'Los Angeles' 'District of Columbia'
 'Meade' 'DeKalb' 'Fairfax' 'Hardin' 'Anne Arundel' 'Kings' 'Lee' 'Ferry'
 'Loudoun' 'Brevard' 'Currituck' 'Orange' 'Maricopa' 'Hamilton' 'Stafford'
 'Hennepin' 'Ventura' 'Lake' 'Monterey' 'Placer' 'Montgomery' 'Doña Ana'
 'Suffolk' 'Allegheny' 'Solano' "St. Mary's" 'Jackson' 'Leavenworth'
 'Middlesex' 'Collin' 'Kootenai' 'San Francisco' 'Bell' nan 'Alameda'
 'Geary' 'Bristol' 'Contra Costa' 'Duval' "Prince George's" 'Bexar'
 'Pettis' 'Chesterfield' 'Prince George' 'Tarrant' 'Maui' 'Virginia Beach'
 'Plaquemines' 'Rockdale' 'Northampton' 'Texas' 'Arapahoe' 'Yuba'
 'Anchorage' 'Riverside' 'York' 'Sacramento' 'Cumberland' 'St. Charles'
 'Camden' 'Cook' 'Alexandria' 'Charles' 'Providence' 'St. Louis'
 'New London' 'Chesapeake' 'Allen' 'San Bernardino' 'El Paso' 'Pulaski'
 'New York' 'James City' 'Davidson' 'Wise' 'Greene' 'Larimer' 'Macomb'
 'Washoe' 'Dallas' 'Rockingham' 'Sarasota' 'Frederick' 'Newport'
 'Hillsborough' 'Galveston' 'Forsyth' 'Harnett' 'Falls Church' 'Sussex'
 'Horry' 'Harford' 'Arlington' 'Baltimore' 'Madison' 'Johnson' 'Moore'
 'Gwinnett' 'Laramie' 'Sarpy' 'Essex' 'Hartford' 'Honolulu' 'Miami-Dade'
 'Osceola' 'Shelby' 'Hoke' 'Travis' 'Multnomah' 'Muscogee' 'Volusia'
 'Kent' 'Fredericksburg' 'Marion' 'Garfield' 'Nueces' 'Harris' 'Kern'
 'Marin' 'Polk' 'Pima' 'Brown' 'Prince William' 'New Castle' 'Atlantic'
 'Autauga' 'Albemarle' 'Saratoga' 'Houston' 'Richmond' 'Berkeley' 'Pinal'
 'Palm Beach' 'Cuyahoga' 'Medina' 'Hudson' 'Williamson' 'Tooele']
Make
['TESLA' 'HYUNDAI' 'BMW' 'NISSAN' 'POLESTAR' 'CHEVROLET' 'FIAT' 'KIA'
 'RIVIAN' 'TOYOTA' 'VOLKSWAGEN' 'FORD' 'AUDI' 'PORSCHE' 'VOLVO'
 'MITSUBISHI' 'JAGUAR' 'SMART' 'LEXUS' 'MERCEDES-BENZ' 'GMC' 'MINI'
 'SUBARU' 'CADILLAC' 'ACURA' 'HONDA' 'GENESIS' 'LUCID' 'FISKER' 'VINFAST'
 'MAZDA' 'MULLEN AUTOMOTIVE INC.' 'BRIGHTDROP' 'TH!NK' 'AZURE DYNAMICS'
```

```
 'ROLLS-ROYCE' 'JEEP' 'RAM']
Model
['MODEL 3' 'MODEL Y' 'IONIQ 5' 'MODEL S' 'I3' 'LEAF' 'MODEL X' 'PS2'
 'BOLT EV' 'SPARK' '500' 'IONIQ' 'SOUL' 'NIRO' 'R1S' 'BZ4X' 'EV6' 'E-GOLF'
 'F-150' 'E-TRON' 'I4' 'IX' 'SOUL EV' 'TAYCAN' 'KONA' 'BOLT EUV' 'R1T'
 'MUSTANG MACH-E' 'XC40' 'I-MIEV' 'I-PACE' 'EDV' 'FOCUS' 'FORTWO' '500E'
 'BLAZER EV' 'Q4' 'E-TRON GT' 'RZ' 'B-CLASS' 'ARIYA' 'HUMMER EV SUV'
 'ID.4' 'KONA ELECTRIC' 'COUNTRYMAN' 'IONIQ 6' 'CYBERTRUCK' 'SOLTERRA'
 'LYRIQ' 'EQE-CLASS SUV' 'EQS-CLASS SEDAN' 'RAV4' 'HUMMER EV PICKUP' 'ZDX'
 'PROLOGUE' 'MACAN' 'HARDTOP' 'SILVERADO EV' 'EV9' 'FORTWO ELECTRIC DRIVE'
 'RS E-TRON GT' 'Q6' 'EQUINOX EV' 'GV60' 'Q8' 'I5' 'EQS-CLASS SUV' 'GV70'
 'AIR' 'C40' 'I7' 'EQE-CLASS SEDAN' 'E-TRON SPORTBACK' 'OCEAN' 'TRANSIT'
 'EQB-CLASS' 'RANGER' 'EX30' 'SQ8' 'IONIQ 5 N' 'ROADSTER' 'EX90'
 'ID. BUZZ' 'VF 8' 'POLESTAR 3' 'EQ FORTWO' 'OPTIQ' 'EX40' 'MX-30'
 'G-CLASS' 'G80' 'ONE' 'ESPRINTER' 'ZEVO' 'CITY' 'SIERRA EV'
 'TRANSIT CONNECT ELECTRIC' 'SPECTRE' 'WAGONEER S' 'MIRAI' 'SQ6'
 'PROMASTER 3500' 'BRIGHTDROP 400']
```

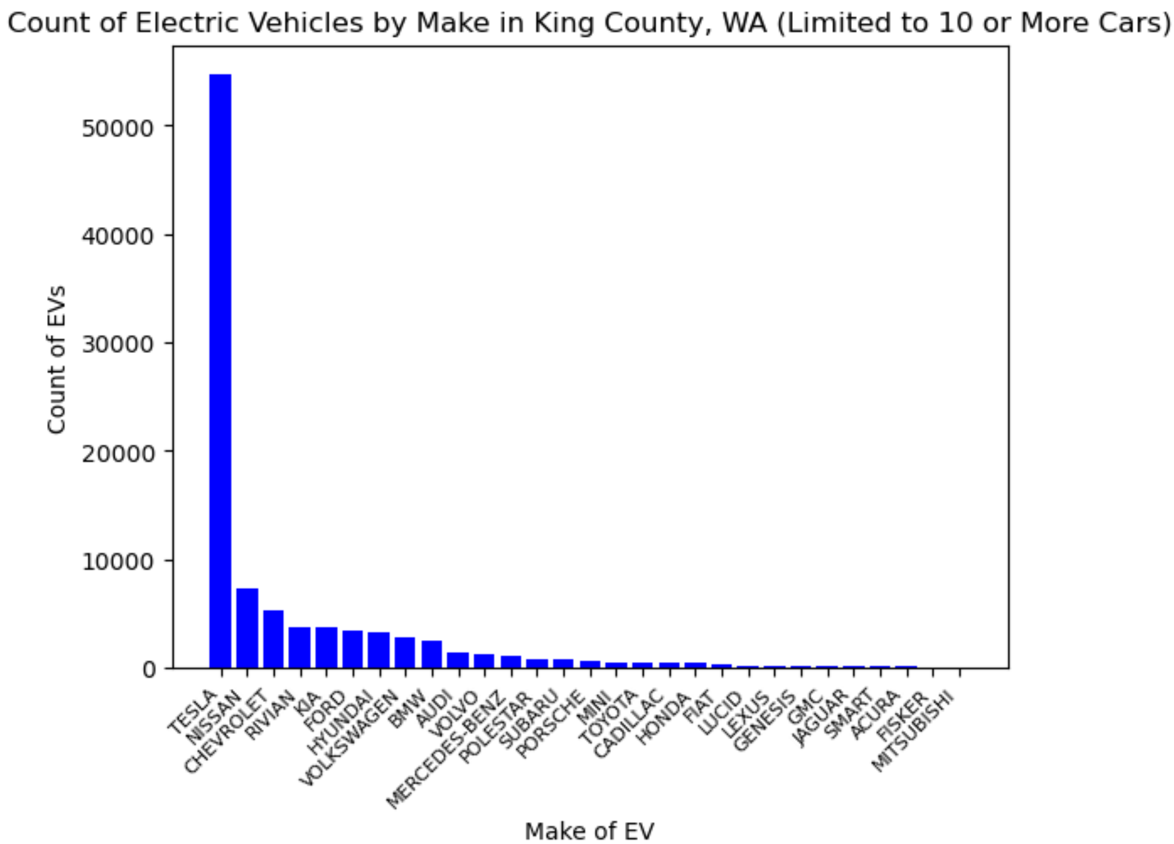
In [22]:
```python
## Initial Group by to get foudnational numbers, will agg more to get different sum
step1 = lim_df.groupby(['County','Make',"Model","Electric Range"]).agg({"Model Year
step1 = step1.rename(columns={"Model Year":"EV_Count_ModelLevel"})
```

In [54]:
```python
## County Make Breakdown
step1["County"] = step1["County"].astype(str).str.upper()
county_make = step1.groupby(['County',"Make"]).agg({'EV_Count_ModelLevel':'sum',}).
county_make = county_make.rename(columns={"EV_Count_ModelLevel":"EV_count"})
## Filtering for King county to see the popularity fo EVs by Make for Seattle.
df_king = county_make[(county_make["County"]=="KING")&(county_make["EV_count"]>=10)
print(df_king)
```

	County	Make	EV_count
439	KING	TESLA	54709
431	KING	NISSAN	7395
413	KING	CHEVROLET	5352
435	KING	RIVIAN	3824
423	KING	KIA	3740
416	KING	FORD	3434
420	KING	HYUNDAI	3333
442	KING	VOLKSWAGEN	2858
410	KING	BMW	2437
408	KING	AUDI	1426
443	KING	VOLVO	1215
427	KING	MERCEDES-BENZ	1106
432	KING	POLESTAR	822
438	KING	SUBARU	746
433	KING	PORSCHE	647
428	KING	MINI	554
441	KING	TOYOTA	542
412	KING	CADILLAC	517
419	KING	HONDA	448
414	KING	FIAT	401
425	KING	LUCID	235
424	KING	LEXUS	223
417	KING	GENESIS	197
418	KING	GMC	139
421	KING	JAGUAR	121
437	KING	SMART	108
407	KING	ACURA	103
415	KING	FISKER	80
429	KING	MITSUBISHI	18

In [56]:

```
## Plotting
plt.bar(df_king['Make'], df_king['EV_count'], color='blue')
plt.title("Count of Electric Vehicles by Make in King County, WA (Limited to 10 or
plt.xlabel('Make of EV')
plt.ylabel("Count of EVs")
## FIxing the labels on X b/c illegible
plt.xticks(rotation=45,ha='right',fontsize=8)
plt.show()
```


Count of Electric Vehicles by Make in King County, WA (Limited to 10 or More Cars)
Count of EVs
0
10000
20000
30000
40000
50000
TESLA
NISSAN
CHEVROLET
RIVIAN
KIA
FORD
HYUNDAI
VOLKSWAGEN
BMW
AUDI
VOLVO
MERCEDES-BENZ
POLESTAR
SUBARU
PORSCHE
MINI
TOYOTA
CADILLAC
HONDA
FIAT
LUCID
LEXUS
GENESIS
GMC
JAGUAR
SMART
ACURA
FISKER
MITSUBISHI
Make of EV

Data Wrangling (CHECK LIST VERSION)

Create a subset of your original data and perform the following.

1. Modify multiple column names.
   - Edited multiple column names stemming from group by needs.
2. Look at the structure of your data – are any variables improperly coded? Such as strings or characters? Convert to correct structure if needed.
   - There are seemingly no improper data types. no real need to convert, also no encoding issues.
3. Fix missing and invalid values in data.
   - There were no invalid or null values in the datasetl.
4. Create new columns based on existing columns or calculations.
   - Did this via the group by sums for different levels of aggregation.
5. Drop column(s) from your dataset.
   - Dropped multiple columns by selecting subset when starting the analysis. kept the columns i needed.
6. Drop a row(s) from your dataset.
   - Dropped rows via the selection of "Battery Electric Vehicle (BEV)" for the type of EV we wanted to look at.
7. Sort your data based on multiple variables.
   - Sorted the semi final results by county adn make.
8. Filter your data based on some condition.
   - Filtered data via the selection of "Battery Electric Vehicle (BEV)" for the type of EV we wanted to look at.
9. Convert all the string values to upper or lower cases in one column.
   - Converted the county names to uppercase.
10. Check whether numeric values are present in a given column of your dataframe.
    - Did this with desribe and info in the begining. Only Year and the Electric range values. Did group by for more number counts.
11. Group your dataset by one column, and get the mean, min, and max values by group.
    - Groupby()
    - agg() or .apply()
    - Grouped by for coutns.
12. Group your dataset by two columns and then sort the aggregated results within the groups.
    - Did this. Mentioned above.

You are free (and should) to add on to these questions. Please clearly indicate in your assignment your answers to these questions.

```
In [ ]:
```

Conclusions

After exploring your dataset, provide a short summary of what you noticed from this dataset. What would you explore further with more time?

After taking a look at parts of this data set, one can see that Tesla's are by far the most popular Battery Powered EV in King County Washington, which is home to the city of Seattle. THe second and third most popular EV brand, are Nissan and Chevrolet, resepctively. With more time, i would see how this break down of Electric Vehicle preference by make at the county level varies for each county. Additionally, i would want to see this over time. particularly now, with the controversy surrounding Elon Musk and Tesla. I would be curious to see if this data shifts over time, however for this we would need more data. As a proxy, without new data, we could see which year the more popular EV models were purchased and identify if it was before or after the controversy.