Download (right-click, save target as ...) this page as a jupyterlab notebook from: ES07

Matplotlib, Pandas, and Numpy for Dataframe Management and Plotting

ENGR 1330 | Computational Thinking with Data Science | Texas Tech University

Developed By: Samshul Arefeen @ Texas Tech University

1) Pandas and Matplotlib libraries

Exercise-1: Import all the libraries - numpy, pandas, and matplotlib so that we do not have to worry about importing the libraries later on in this assignment

(POINTS: 6)

In [ ]:

```

```

Exercise-2: The file http://54.243.252.9/engr-1330-webroot/2-Homework/ES07/Electric_Vehicle_Population_Data.csv contains the database of electric vehicles registered and operated in different cities and states (primarily Washington State) of the United States. It also contains the vehicle details like make, model, model year, vehicle type, electric range, base retail price and fields that are mostly self explanatory.

> *Get the file by either download from link above, or use the script below to automatically get the file; you may need to install the requests module `! pip install requests` will sometimes work when run from the notebook, but not always*

(POINTS: 54 - each task in this exercise carries 9 points)

In [ ]:

```
! pip install requests # magic function to install requests into ipython kernel
```

In [ ]:

```
# Get the database -- use the Get Data From URL Script
# import needed modules to interact with the internet
import requests
# make the connection to the remote file (actually its implementing "bash curl -O http://fqdn/path ...")
remote_url = 'http://54.243.252.9/engr-1330-webroot/2-Homework/ES07/Electric_Vehicle_Population_Data.csv' # an E
response = requests.get(remote_url) # Gets the file contents puts into an object
output = open('Electric_Vehicle_Population_Data.csv', 'wb') # Prepare a destination, local
output.write(response.content) # write contents of object to named local file
output.close() # close the connection
# check that database is downloaded and non-empty
```

Task-1: Read the **Electric_Vehicle_Population_Data.csv** file and store in a variable name **ev_pop**. After reading, display the first 10 rows of the dataframe **ev_pop** as the output.

In [ ]:

```
#GIVE YOUR ANSWER FOR TASK-1 IN THIS CELL
```

Task-2: Drop the columns 'Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Legislative District', 'DOL Vehicle ID', 'Vehicle Location' from the dataframe **ev_pop**. After indexing, display the first 5 rows of the dataframe **ev_pop** as the output. [Hint:The drop() function with 'columns' and 'inplace' argument may be used]

In [ ]:

```
#GIVE YOUR ANSWER FOR TASK-2 IN THIS CELL
```

Let's say we first want to see whether the EV purchase has generally shown an growth trend over the years. The strategy to visualize it is to have the 'Model Year' (in an ordered manner) in the x-axis and the value counts of that 'Model Year' in the y-axis as a scatter plot.

Task-3: Extract the value counts for each 'Model Year' and save the pandas Series in the variable name 'model_year'

In [ ]:

```
#GIVE YOUR ANSWER FOR TASK-3 IN THIS CELL
```

At this point model_year is a pandas Series having the 'Model Year' values as its index and count of values as the Series.

If we explore the dataset we shall see the 'Base MSRP' column inexplacably contain a good number of values equal to 'zero' and in some case unusually high values (can be disregarded as outlier) which are greater than 100,000.

Task-4: Plot a scatter plot with the Model Year values in the x-axis and their value counts in the y-axis. Make sure the plot has proper labels and title. [Hint: You can use Series.index attribute to have model year as a data sequence]. Comment on your observations in a following markdown cell.

In [ ]:

```
#GIVE YOUR ANSWER FOR TASK-4 IN THIS CELL
```

Student Answer (Expected): The observation is the EV usage or purchase shown a general trend of exponential growth which might be slowed down in last couple of years due to COVID and other challenging situations.

Let's work towards plotting a histogram for 'MRSP Base Price' value of the entire EV Population database with the following two tasks.

Task-5: Clean up the ev_pop dataframe by selecting only the rows of the dataframe where the two conditions: a) the Base MSRP is greater than 0 b) the base MSRP is less than or equal to 100000 are simultaneously met (**and** operation).Save the modified dataframe as **ev_pop_cleaned**

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-5 IN THIS CELL
```

Task-6: Plot the Histogram of the 'Base MSRP' series of the **ev_pop_cleaned** dataframe

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-6 IN THIS CELL
```

Exercise-2: In this exercise, we shall visualize the data from the manufacturer's perspective through following four tasks"

(POINTS: 40 - each task in this exercise carries 10 points)

Task-1: Using the **ev_pop** dataframe, extract the top ten makers of electric vehicles by the dataset. Print the names of the makers and their corresponding value counts (no. of vehicles by the maker). [Hint: Use the **value_counts()** and **nlargest()** functions in tendem to extract the series of top ten makers]

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-1 IN THIS CELL
```

Task-2: Make a bar plot for the top ten makers and their vehicle counts in the **ev_pop** database. Use all the proper plotting practices for labels and title. For better readability make sure the x labels are rotated 90 degrees.

Hint: Use the xticks.rotation() function for the label rotation. Use pyplot.show() to avoid unwanted texts above the bar graph.

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-2 IN THIS CELL
```

Now we want to observe the sales (revenue) trend of the top two manufacturers through the following tasks.

Task-3: Extract two separate dataframes for the top two manufacturers where the 'Make' column value of **ev_pop** matches with the manufacturer. Sort the two dataframes by the ascending values for 'Model Year' and save the two sorted dataframes as **sorted_Tesla** and **sorted_Nissan**. Apply the **groupby()** and **sum()** function together for grouping the sorted dataframes by 'Model Year' and get the summation for numerical fields. It will extract two dataframes (name them **tesla** and **nissan** respectively) which will provide the yearly sum of numerical columns like 'Electric Range' and 'Base MSRP'.

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-3 IN THIS CELL
```

Task-4: Plot an overlaid line plot for 'Model Year' (vs.) 'Revenue (=summed up Base MSRP)' for each of the top two manufacturers. The plot must display an x-label, a y-label, a title, and a legend.

Note-1: To increase the line width of the line plot, use the argument `lw`. For example: lw = 3.

Note-2: To give your own labels for each color that corresponds to one of the 4 countries, use the argument `label`. For example: label = 'Tesla'.

In [ ]:
```
#GIVE YOUR ANSWER FOR TASK-4 IN THIS CELL
```

In [ ]:
```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js