**Module 5 Challenge Report**

**Matplotlib Practice**

**Background**

A key component in the process of data analysis is visualization, which is a graphical representation of the information or the interpretation of the result of findings. It is a fundamental part of story-telling. Matplotlib is a Python Pandas library that offers techniques of constructing graphical representation that communicate data.

**Objective**

The objective is to demonstrate the process involved in analyzing a given sets of data using Python Pandas Matplotlib set of functions built into the Matplotlib library. In this exercise, we will create summary statistics of key metrics and produce the graphical representation.

**Development Environment** Python Pandas Matplotlib Jupyter Notebook Data file (city\_data.csv, ride\_data.csv)

**Code Plan**

This exercise use a specific procedure and logic to guide the code development. That procedure is used as a guide as outlined in the following. Note that two data files were provided (city\_data.csv, ride\_data.csv).

Part I

1. Import dependencies and libraries for Pandas and Matplotlib.

2. Download data files (city\_data.csv, ride\_data.csv)

3. Set path to the location of the two data files

4. Read in the or load the files using Pandas pd.read\_csv()

5. Inspect the data. Get unique values for each objects of the data.

6. Merge the two data files using pd.merge() function

7. Get average Fare per Ride using groupby()

8. Get the sum of the fares for each city type using groupby() and sum()

9. Get the sum of the drivers for each city type using groupby() and sum()

10. Get the sum of the Rides for each city type using groupby() and sum()

11. Calculate Total Average Fare per Driver

12. Create a summary DataFrame with items 7 through 11

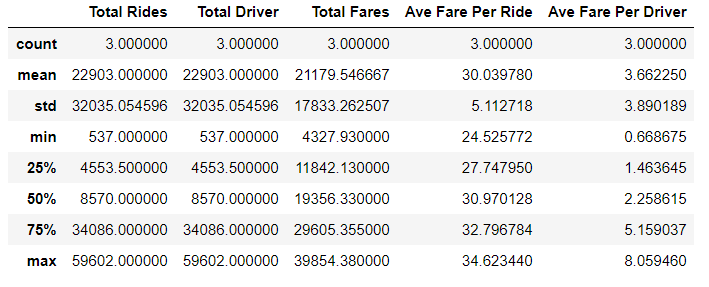
Part II

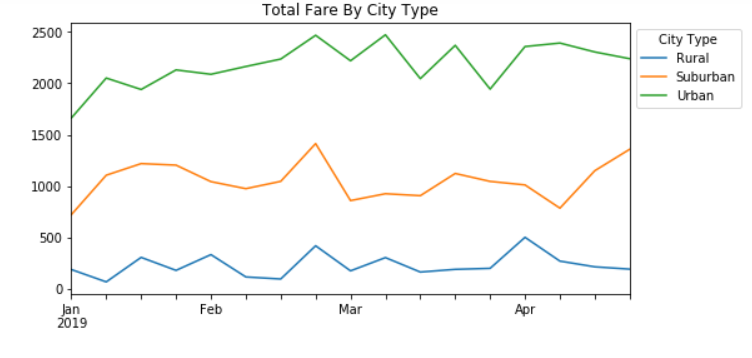
The second part requires some data manipulation before we get to plotting.

1. Rename columns {'city':'City', 'date':'Date','fare':'Fare', 'ride\_id': 'Ride Id','driver\_count': 'No. Drivers', 'type':'City Type'
2. Set the index to the Date column
3. Create a new DataFrame,for fares, and include only the City Type and Fare columns using the copy() method on # the merged DataFrame
4. Set the index to the datetime data type
5. Calculate the sum() of fares by the type of city and date using groupby() to create a Series.
6. Convert the groupby Series into a DataFrame in one step.
7. Reset the index, then create a pivot table DataFrame with the Date as the index and columns = 'City Type'. The Fare for each Date should appear in each row.
8. Create a new DataFrame from the pivot table DataFrame on the given dates, '2019-01-01':'2019-04-28', using loc. Hint: df.loc['start':'end']
9. Create a new DataFrame by setting the DataFrame you created in Step 8 with resample() in weekly bins, and calculate the sum() of the fares for each week in the resampled data.
10. Using the object-oriented interface method, plot the DataFrame you created in Step 9 using the df.plot() function.

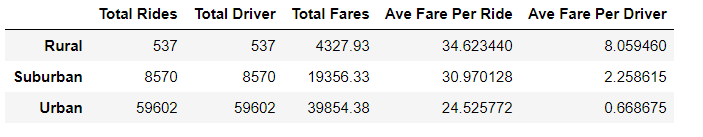
**Summary**

* In this analysis we merged two data files with different aspects of information on ride-sharing. We classified the information into three main categories by “City Type”. These are Rural, Suburban and Urban
* We extract and rearrange the data to allow the analysis based of the three classes of city types, categorically comparing the Fares for each
* The summary DataFrame objects produced are: Total Rides, Total Driver, Total Fares, Average Fare Per Ride and Average Fare Per Driver.
* The Key-Metrics for the dataset are:





**Plot of Total Fare By City Type**

****

* The result of this analysis shows the average fare per ride is highest in Rural City Type at about $35.00, followed by Suburban City Type at $31.00 and the lowest ride fare for the Urban City Type at $25.00**.**
* The analysis also suggests that there are more rides scheduled in the Urban City Type than any other.
* Fewer rides in the Rural suggests less demand, which accounts for the higher fare.