## Uber Trips Analysis in Python

**Description -**

Looking at Uber trips could provide technicians with necessary information to see and make best services thereby making the customers happy. In this project we how we can explore the ride data from Uber to detect the association and patterns between different variables using Python.

For the analysis, first, we have to load the Uber trip data into our Python environment. Will apply library frameworks like pandas for reading and transforming data significantly faster. This entails the DataFrame loading the dataset into the pandas where analysis is quickened as well as the data manipulation.

Subsequently, imporation e of data will be followed by the verification stage, where we will start working on it. Firstly, checking of the structured nature of the dataset, completing missing values and comprehending the variable types are among our concerns too. Descriptive statistics will be used to summarize what the phenomenon is, thereby showing the metrics that represent the midpoint and variability within the data.

After that, we will be investigating the mutual connection among the things in our data set. This require computation of correlations of numerical variables to check if there are any related or moving directions. In this way, too what is going to happen in terms of how the category variables have an impact on each one another and if there is any dependency or any association, will also be looked at.

To demonstrate the relationships forms in the data, we will use some libraries like Matplotlib or Seaborn, which do the plotting for us. Therefore, we are able to set different types of graphs, such as scatter plots, histograms and box plots, which are often used for pattern and trend analysis and identification of an overall distribution.

As we progress into the exploration, we will inspect for any possible trendsetters or odd ones that may need more in-depth study. Such atypical cases could be a cornucopia of information that highlights certain unique ride patterns or potential data problematic that need to be resolved.

**Analysis of patterns and trends -**

Examination of Uber trip data will help us find essential patterns and tendencies which could be utilized to streamline services and enrich user experiences/be the tool to optimize services and improve customer experience. Through Python-based analytics which subsequently get involved in the data, we discover more intricate patterns.

The main part of the analysis includes determining the riding pattern corporate. Here, we consider plasma concentration monitoring along with specific peaks of the hours and locations where the need might be at its top, which in turn could inform decision making about resource allocation and the strategy on pricing. For instance, we could spot that weekday evenings in the city are usually the time drivers are in the highest demand. Hence, this could tell that there is a need for more drivers to be on the roads during that time especially in these zones.

Moreover, having deeper insight into the data from Uber trips makes it possible for us to see the prevailant patterns in user activities. For instance, this could answer the question whether a given demographic would like and use specific types of rides or travel destinations more often rather than others, and hence the marketing campaigns could be directed to that specific group or the service can be modified accordingly. On the other hand, we could learn that the youth users tend to choose ride within city centres which are usually shorter, while older users are likely to travel in longer ways to suburban areas.

Second, we will find out that cross analysis of multiple parameters in the data set may lead to the revealing of deeper relationships. Combining downwards data, for instance where the trips are made, when and how much time it takes them, we can unveil hidden patterns that in turn facilitate the improvement of services. In this way, it may be revealed that longer trips are rather more easy to be made at off-peak times, therefore, a loyalty or discount scheme could be designed for drivers operating during those periods.

Visualization is a perfect tool to analyze these relations in a more transparent and intuitive manner with the help of Python tools. But charts such as scatter plots, histograms, and heat maps can represent the spatial and time series dynamics, and then those facilitate for the decision makers to take decisions.

**Now Let’s look at the code snippets -**

To read the data set we will be importing pandas as pd

import pandas as pd

to visualize the dataset we will be importing matplot library

import matplotlib.pyplot as plt

Reading the Dataset

uber\_df= pd.read\_csv("uber-raw-data-sep14.csv")

Now , Displaying the first 5 records of the Dataset from starting and from ending

uber\_df.head(5)

uber\_df.tail()

Finding the shape of the dataset:

the number of rows and the number of columns in the DataFrame uber\_df.

uber\_df.shape

Now retrieving the Information the Dataset Such as Datatype of each column , number of non null values , memory usage etc .

uber\_df.info()

Now changing the data type of the Date/Time in the column from string to Datetime

uber\_df['Date/Time']= pd.to\_datetime(uber\_df['Date/Time'])

this line of code extracts the day component from the "Date/Time" column and assigns it to a new column named "Day"

ber\_df["Day"] = uber\_df["Date/Time"].apply(lambda x: x.day)

This line of code extracts the hour component from the "Date/Time" column and assigns it to a new column named "Hour"

uber\_df["Hour"] = uber\_df["Date/Time"].apply(lambda x: x.hour)

This line of code extracts the weekday component from the "Date/Time" column and assigns it to a new column named "Weekday"

uber\_df["Weekday"] = uber\_df["Date/Time"].apply(lambda x: x.weekday())

Now again displaying first 5 entries of the dataset

uber\_df.head(5)

Generating a histogram to visualize the density of Uber trips per day

fig,ax = plt.subplots(figsize = (12,6))

plt.hist(uber\_df.Day, width= 0.6, bins= 30)

plt.title("Density of trips per Day", fontsize=16)

plt.xlabel("Day", fontsize=14)

plt.ylabel("Density of rides", fontsize=14)

Generating a histogram to visualize the density of Uber trips per weekday as we know the highest rides are during the weekday

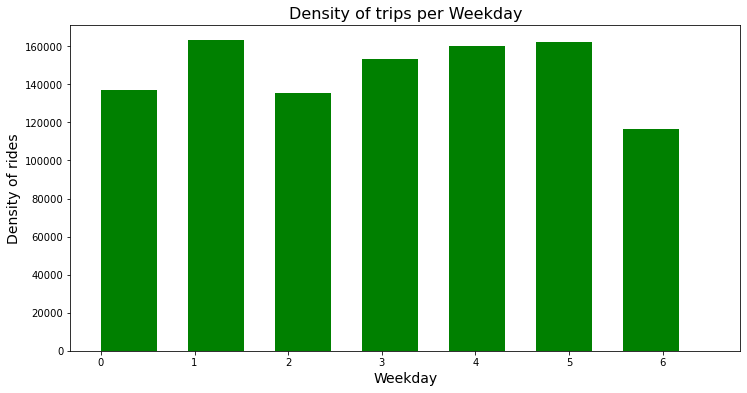
fig,ax = plt.subplots(figsize = (12,6))

plt.hist(uber\_df.Weekday, width= 0.6, range= (0, 6.5), bins=7, color= "green")

plt.title("Density of trips per Weekday", fontsize=16)

plt.xlabel("Weekday", fontsize=14)

plt.ylabel("Density of rides", fontsize=14)



Through the above histogram we got to know that The busiest day in the week for Uber is Monday. On the other hand, Saturday is the day with the least number of rides.

Generating a histogram to visualize the density of Uber trips per hour

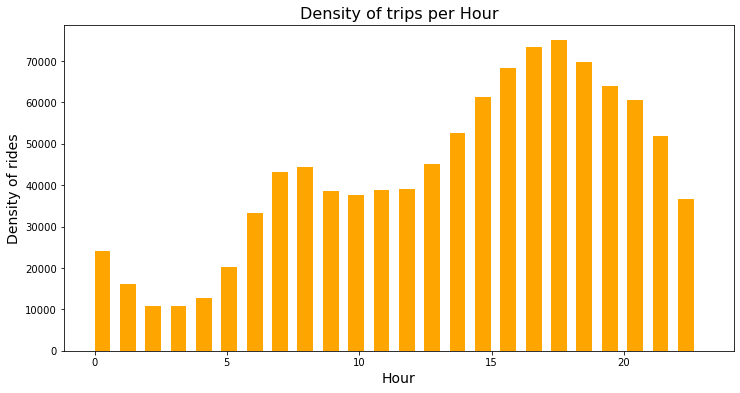
fig,ax = plt.subplots(figsize = (12,6))

plt.hist(uber\_df.Hour, width= 0.6, bins=24, color= "orange")

plt.title("Density of trips per Hour", fontsize=16)

plt.xlabel("Hour", fontsize=14)

plt.ylabel("Density of rides", fontsize=14)



the number of rides decrease gradually from 1 AM to 4 PM and then increases starting from 5 AM onward till it reaches 6 PM which is the hour with the highest number of rides.

Creating a scatter plot to visualize the density of Uber trips based on longitude and latitude coordinates.

fig,ax = plt.subplots(figsize = (12,6))

x= uber\_df.Lon

y= uber\_df.Lat

plt.scatter(x, y, color= "purple")

plt.title("Density of trips per Hour", fontsize=16)

plt.xlabel("Hour", fontsize=14)

plt.ylabel("Density of rides", fontsize=14)



The region with the highest density of rides is near Manhattan and Newburgh. While the region with the lowest density is near New Jersey.