## Uber Trips Analysis in Python

**Description -**

Data collected for Uber trips can educate the technicians with valuable learning tools and practice hub, enabling them to fine-tune their services and customer satisfaction in the long run. This project undertakes the utilization of Uber ride data in the Python which thereby seeks to establish some associations among numerous elements.

Firstly, the Uber trip data is loaded from the VM into the Python environment. The library infrastructure including pandas in this case helps in having quick data reading and editing, thus dataframe analysis is rapid.

The second step data importation is then followed by the checking procedure involving checking the data set in detail. This comprises of the examination of data construction whether there is a gap, spotting values where values are missing, and ascertaining variable composition. Descriptive statistics will be used to concisely portray central inclinations and variance in data and to disclose main distributions.

Next, we transmit data relationships which is the following topic. This implies that the relations between numerical variables related one to another are studied to make sure of any related or moving trends. Moreover, we consider the ways in which nominal variables operate in reference to each other through studies of dependencies and correlations.

These libraries help to demonstrate how the relations used in the data are laid out visually by libraries such as Seaborn or Matplotlib. Graphical features are designed in such a way that it provides ability to draw different kinds of charts such as scatter plot, histogram and box plot that help in analyzing patterns and trends. Additionally, these of the overall distribution.

The exploration in a profound state tries to detect whether there is a assignable outlier or a deviation individual actor whose actions require a further investigation. This example of a split-route case can give us invaluable information about an individual's riding habits or maybe the presence of some data bugs that have to be fixed.

A comparison of Uber’s trip data will in all likelihood give away some of the important sleep-deeper truths that fuel the process of enhancing services and creating a better cure for clients. A Python-based profound analytics give a clear look at the data, detecting subtle relation between different variables. In this way, better decisions are made.

A principle important fact is, therefore, to establish riding patterns of corporations. The idea in this is to see the blood plasma concentration as well as properly time it in hours and places of peak demand. By doing that analysis, conclusions will be drawn from them to be utilized by allocating resources and setting prices, as well as improving serviceholder efficiency.

Additionally, the data is enabling the marketers to target particular event-based activity patterns and information specific to a particular demographic is driving the modification of marketing campaigns and services. Нарастывание к многопараметрическому анализу внутри данных даёт возможность выявить скрытый шаблон и, в конечном итоге, позволяет проводить卡чественные улучшения.

Visualization is a strong tool because it simplifies the complicated analysis process by presenting information in a graphical form. For instance, a charts like a scatter plot, heat maps, and graphs explaining spatiotemporal dynamics are just among the numerous visuals that can provide decision-makers with factual arguments to build their decisions on.

And then, it‘s time to 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

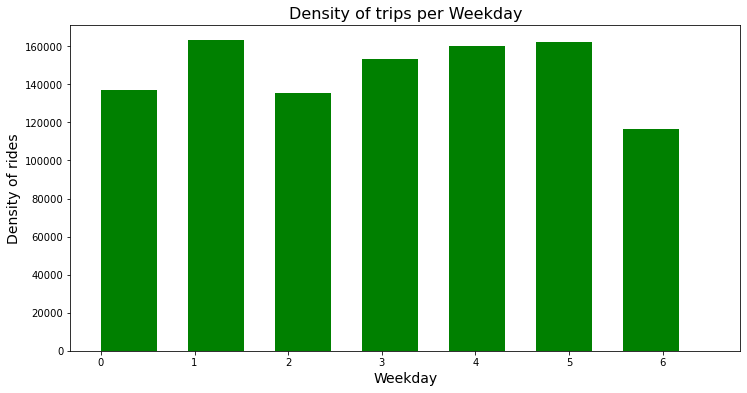
fig,ax = plt.subplots(figsize = (12,6))

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

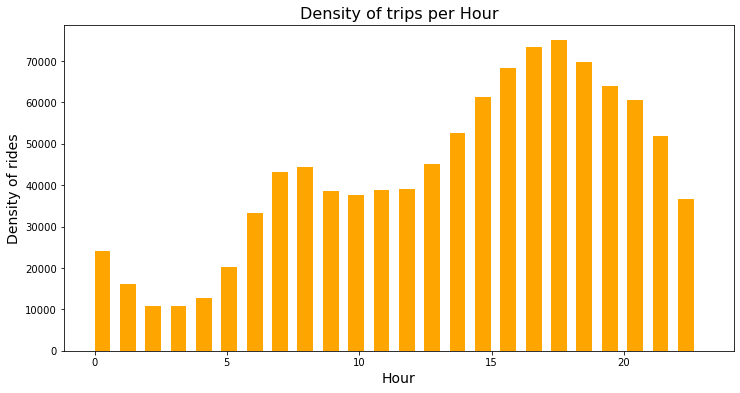
fig,ax = plt.subplots(figsize = (12,6))

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.