DATA ANALYSIS AND VISUALIZATION FOR CAB TRIPS

CSE3020 - DATA VISUALIZATION

By

Lagan Gupta - 19BCE2057

Akanksha Jagdish - 19BCE2228

Crish Karthik P - 18BCE2171

S R Praveen - 19BCE0363

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT

In recent years, many countries have witnessed speedy urbanization, improvements in living standards and significant growths in economic activities. As a consequence, there has been a substantial increase in the need to commute.

Not everyone can afford a private vehicle, so they turn towards the next best option- cabs. The increased cab demand levels have led to an increase in the overall numbers of cab trips, which has contributed to increased traffic congestion, energy consumption, and air pollution, particularly in big cities.

An increasing number of cabs drive on roads and produce a massive volume of trajectory data. These data provide a new opportunity for understanding people's travel behavior.

With the help of visualization, cities can avail the benefit of understanding the complex data and gain insights that would help them to craft decisions.

INTRODUCTION

We have collected the data set ‘taxi trip duration’ from Kaggle website which gives information like number of trips by a particular vehicle, idle time, pick-up datetime, drop datetime, no. of passengers and trip duration.

Uber TLC FOIL Response:

This directory contains data on over 4.5 million Uber pickups in New York City from April to September 2014, and 14.3 million more Uber pickups from January to June 2015. Trip-level data on 10 other for-hire vehicle (FHV) companies, as well as aggregated data for 329 FHV companies, is also included. All the files are as they were received on August 3, Sept. 15 and Sept. 22, 2015.

This data was used for four FiveThirtyEight stories: Uber Is Serving New York’s Outer Boroughs More Than Taxis Are, Public Transit Should Be Uber’s New Best Friend, Uber Is Taking Millions Of Manhattan Rides Away From Taxis, and Is Uber Making NYC Rush-Hour Traffic Worse?.

The Data:

The dataset contains:

1. Uber trip data from 2014 (April - September), separated by month, with detailed location information

Uber trip data from 2014:

There are six files of raw data on Uber pickups in New York City from April to September 2014.

The files are separated by month and each has the following columns:

1. Date/Time: The date and time of the Uber pickup
2. Lat: The latitude of the Uber pickup
3. Lon: The longitude of the Uber pickup
4. Base: The TLC base company code affiliated with the Uber pickup

These files are named:

1. uber-raw-data-apr14.csv
2. uber-raw-data-aug14.csv
3. uber-raw-data-jul14.csv
4. uber-raw-data-jun14.csv
5. uber-raw-data-may14.csv
6. uber-raw-data-sep14.csv

LITERATURE SURVEY

Big Data Trip Classification on the New York City Taxi and Uber Sensor Network

They first employ big data technologies to analyze this vast dataset: Apache Spark is used for data processing and classification, Apache Hive is used for data storage, and MapReduce is used for data profiling. Since taxis and Uber are equipped with GPS sensors, they then visualize a mobile sensor network over New York City separated into fine-sized regions each acting as a mobile sensing node.

Each location on the network falls into a region and is classified into one of three categories based on which service dominates the particular region: Yellow taxi, Green taxi, or Uber. They utilize logistic regression to classify a region into one of the three categories.

Their classification algorithm is then used to analyze the interaction between taxi and Uber. Finally, they propose a trip recommendation system for users using classification results together with a web service application.

Analysing Uber Trips using PySpark

In this venture they analyze the Daily, Monthly and Yearly Uber Pickups in New York City. This mission is primarily based on Data Visualization that will inform us toward use of ggplot2 library for perceiving the data.

They made use of programs like ggplot2 that allowed them to plot a number of sorts of visualizations that pertained to a number of time-frames of the year. With this, they conclude how time affected client trips. Finally, they made a geo plot of New York that furnished us with the small print of how a variety of customers made journeys from one of a kind bases.

Investigating the effect crime has on Uber and Yellow Taxi pickups in NYC

They use the data from April to September in 2014 for both Uber and Yellow Taxi pickups, and socio-economic data in 2014 by census tract to develop several econometric models treating each census tract as an individual observation. They create some heatmaps since the question they are interested in is highly related to geographical differences, and heatmaps can highlight some features of the data that they can hardly capture otherwise.

Exploring the Taxi and Uber Demands in New York City: An Empirical Analysis and Spatial Modeling

This study aims to investigate the impact of the emerging app-based for-hire vehicles on taxi industry through quantitative analyses of Uber and taxi demands for neighborhoods of New York City (NYC). Demand forecasting models, which can account for the spatial dependence of Uber and taxi trips are developed.

In the empirical analysis, they explore the spatio-temporal patterns of Uber and taxi pick-up data. Results of the Moran’s I tests confirm the spatial dependence of both taxi and Uber demands.

Linear models, spatial error models, and spatial lag models are developed to estimate the taxi and Uber demands of each neighborhood using socio-economical and transportation-related characteristics.

Analysis of the passenger pick-up pattern for taxi location recommendation

This paper analyzes a pick-up pattern of taxi service in Jeju area based on the real-life location history data collected from the Taxi Telematics system, aiming at obtaining useful background data necessary to design a location recommendation service for empty taxis.

To decide a reasonable granularity of location recommendation, refined clustering is performed by means of the well-known k-means method supported by the E-Miner statistics software package. First the irrelavant fields are removed from the dataset. Then using the SAS E-Miner, first trace is implemented.

Now, the close pick-up points are to be grouped to build a meaningful recommendable space entity. E-Miner provides the geographical clustering function over the 2-dimensional space, so we run this program for the pick-up records. This step employs k-means clustering.

This paper has built an analysis framework for real-life movement history data collected in the Jeju Taxi Telematics system, extracted information on the passenger pick-up pattern by probing the state diagram, and analyzed its spatial and temporal distribution.

Taxi Hotspots Identification through Origin and Destination Analysis of Taxi Trips using K-means Clustering and H-indexing

In this study, the taxi origin and destination hotspots are determined by first clustering the available O-D pairs from empirical mobility traces. The validity of these formed clusters is determined by utilizing the silhouette analysis. Finally, hotspots are located by measuring the cluster's h-index.

Simulation results reveal that more clusters tend to provide unreliable silhouette values due to the fact that origin/destination GPS points are very close to each other. For a given number of clusters, the h-index tend to locate clusters that can be considered as hotspots.

The algorithm devised first samples the Taxi mobility traces with the dataset mainly consisting of latitude, longitude, time stamp of the corresponding GPS location. Then the GPS distances are converted to kilometers and determination of origin and destination of each trip is done. Then using Silhouette Analysis, an optimal value for k is found which is further used to create clusters.

Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips

The data are complex, containing geographical and temporal components in addition to multiple variables associated with each trip. Consequently, it is hard to specify exploratory queries and to perform comparative analyses (e.g., compare different regions over time).

This problem is compounded due to the size of the data—there are on average 500,000 taxi trips each day in NYC. The paper allows the users to search up the taxi routes using a visual query system.

The model is able to express a wide range of spatio-temporal queries, and it is also flexible in that not only can queries be composed but also different aggregations and visual representations can be applied, allowing users to explore and compare results.

Analysis and Visualization for Hot Spot Based Route Recommendation Using Short-Dated Taxi GPS Traces

With the paper's approach, hot spots for loading and unloading passenger(s) are extracted using an improved DBSCAN algorithm after data pre-processing including cleaning and filtering.

Then, this paper describes the start-end point-based similar trajectory method to get coarse-level trajectories clusters, together with the density-based ε distance trajectory clustering algorithm to identify recommended potential routes.

A weighted tree is defined including such factors as driving time, velocity, distance and endpoint attractiveness for optimal route evaluation from vacant to occupied hot spots. The taxi GPS data used in this paper is downloaded from the website of Datatang with about 7600 taxis in the city of Nanjing from September 1–2, 2010.

Visualizing Hidden Themes of Taxi Movement with Semantic Transformation

A new methodology is developed to discover and analyze the hidden knowledge of massive taxi trajectory data within a city. This approach creatively transforms the geographic coordinates (i.e. latitude and longitude) to street names reflecting contextual semantic information.

Consequently, the movement of each taxi is studied as a document consisting of the taxi traversed street names, which enables semantic analysis of massive taxi data sets as document corpora. Hidden themes, namely taxi topics, are identified through textual topic modeling techniques.

The taxi topics reflect urban mobility patterns and trends, which are displayed and analyzed through a visual analytics system. The system integrates interactive visualization tools, including taxi topic maps, topic routes, street clouds and parallel coordinates, to visualize the probability-based topical information.

Urban planners, administration, travelers, and drivers can conduct their various knowledge discovery tasks with direct semantic and visual assists. The effectiveness of this approach is illustrated by case studies using a large taxi trajectory data set acquired from 21, 360 taxis in a city.

Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster

Customer mapping is a grouping of customer profiling to facilitate analysis and policy of SMEs in the production of goods, especially batik sales. Researchers will use a combination of K-Means method with elbow to improve efficient and effective k-means performance in processing large amounts of data.

K-Means Clustering is a localized optimization method that is sensitive to the selection of the starting position from the midpoint of the cluster. So choosing the starting position from the midpoint of a bad cluster will result in K-Means Clustering algorithm resulting in high errors and poor cluster results.

The K-means algorithm has problems in determining the best number of clusters. Based on the results obtained from the process in determining the best number of clusters with elbow method can produce the same number of clusters K on the amount of different data.

The result of determining the best number of clusters with elbow method will be the default for characteristic process based on case study. Measurement of k-means value of k-means has resulted in the best clusters based on SSE values on 500 clusters of batik visitors.

PROPOSED WORK

The demand for cabs at any time of the day in any month is huge. The base stations may not have enough number of drivers which can lead to cancellation of rides or delay in cabs reaching or dropping customers.

This happens especially during the peak hours. The surge in cab demands also happens to ask for more number of cabs or the cabs to reach quicker to the one hiring it. Thus, it becomes important to station cab stations at such areas which maximise the avavilability and minimise the time to get a cab.

Data Abstraction

<u>Method of Abstraction:</u>

The method we have adapted for data abstraction is based on the prime incentive of the attributes that are essential for traffic congestion analysis.

As our project title is "Data analysis and visualization for cab trips", our main motive revolves around where the cabs availed the maximum (Position/Location), at what hours is it peaking during the day(time) and lastly how many days in a month is the usage of cabs the most(days). These are some basic attributes that we came across which has resulted in choosing the following attributes and items:

The attributes that we have decided for:

1. Position: longitude and latitude
2. Peaking hours: Time
3. Days in a month: Days

Data Representation:

We have used the Table dataset to represent our data with latitude and longitude of the destinations, peaking hours(date/time), and days of the 6 months we have taken into consideration being the attributes.

The reason we have chosen the table dataset is because:

1. The display will be used to look up individual values.
2. It will be used to compare individual values but not entire series of values to one another.
3. Precise values are required.
4. The quantitative information to be communicated involves more than one unit of measure.
5. Both summary and detailed values are included.
6. It is possible to analyze how the different values for a set of independent variables affect a dependent variable.

Attribute Semantics:

This project focuses on ontology-based approach to support the process of visualizing urban mobility data. The approach consists of building a visualization-oriented urban mobility ontology, focused on themes such as ridership, vehicle flows and many others.
The ontology also allows characterizing visualization techniques with human perception factors, so that they can be used to automatically infer recommended techniques for a dataset. 2D Heat Map is appropriate for analyzing travel by cab.

Since we have longitude and latitude as attributes, we have not categorized them as ordinal, categorical or sequential, although the visualization can be technically support both.

Target Identification:

Our target audience are the cities' traffic police and officers in charge of keeping the city pollution level low.

We visualize for them how the most used service in the city -cab, is causing congestion at traffic signals. We provide them with visualized comparisons after comparing data of 6 months like:

1. Number of cab trips everyday
2. Number of cab trips per month
3. Number of cab trips every hour in each month

We intend to help the officers make decisions wise for the city and the world by analyzing and visualizing cab trip frequency that can be related to air pollution, which will impact all of us in the long run.

Task Abstraction

Tasks:

1. To plot cab trip usage by using different visualization techniques on an hourly, weekly and monthly basis.
2. To determine the peaking hours of availed cabs in a day.
3. To find the month with maximum cab utilization between April-September.

Actions:

Actions are used to define user goals. As our project is based on analysis, we have decided to go with analyse-consume existing data. Under consume, we have 3 steps that we need to follow to analyse information already existing.

1. Discover: We consume the existing data, and with that we try to discover something new. We derived the number of base stations needed to be employed by the UBER and the placing(location) of those base stations in the NYC, with the view to maximize the profit.
2. Present: We present the data in a user understandable visualization. This makes it easier for users not familiar with R and python to understand and comprehend the data.
3. Enjoy: For the users indulgence, we focus on common interests. That is maximum car availability during the month or day.

Visual Idioms:

A visualization idiom is a distinct approach to creating and manipulating visual representations.

Data: Data are the types and hierarchical salience of the information to represent. In this project we have collected a dataset in the form of 6 months from April to September. Each data set has millions of data.

Design: The design part is where we visually encode and organize choices. We went through various graphically visual techniques and used the most comprehensible techniques to visualize the data available. Implementations like pie-charts, bar-graphs, heat-maps, and line graphs are used so that any individual irrespective of their knowledge about the data set or the technique can understand clearly.

K-MEANS CLUSTERING ALGORITHM:

Clustering is the process of dividing the datasets into groups, consisting of similar data-points". Clustering is a type of unsupervised machine learning, which is used when you have unlabeled data. The K-means clustering algorithm's main goal is to group similar elements or data points into a cluster. The "K" in K-means represents the number of clusters.

We applied K-means clustering on this dataset to understand the trips taken with Uber and to identify rides in different places within New York. We consume the existing data, and with that we try to discover something new. We derived the number of base stations needed to be employed by the UBER and the placing(location) of those base stations in New York City, with the view to maximize the profit.

WHAT DO WE DERIVE USING K-MEANS ALGORITHM IN OUR DATASET:

Attributes involved in implementing K-MEANS CLUSTERING:

- Latitude of the pickup location
- Longitude of the pickup location

Uber can use the centroids of the clusters we have derived as their new base stations. Whenever Uber gets a new ride request from the customer, they can check the closeness of the pickup location (in terms of the location's latitude and longitude values) against each centroids (denotes new base station) we obtained using K-MEANS CLUSTERING algorithm. Whichever base station is nearer, Uber can direct the vehicle from that particular base station to the customer location.

Uber has many drivers and offers services in many locations. If Uber knows the base station, and if they are getting a lot of ride requests from one particular base station in comparison with the other base stations, then uber can strategically place their drivers in better locations wherein their probability of receiving a ride request is higher. This will help Uber to serve their customers faster as vehicles are placed closer to the pickup location and also help to grow their business.

RESULTS

Visualization using R:

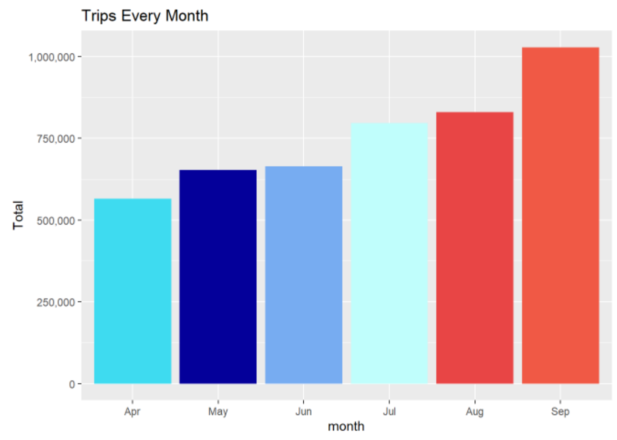
Dashboard

Taxi Trips visualization Dashboard

trips relating to month | Trips relating to days

Trips Every Month | Trips by Hour and Month

```
ggplot(month_group, aes(month, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips Every Month") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma) +
  scale_fill_manual(values = colors2)
```



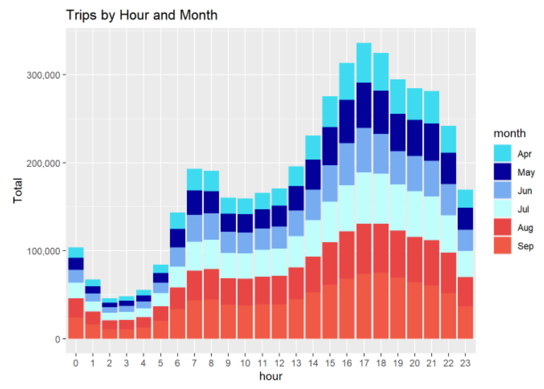
Dashboard

Taxi Trips visualization Dashboard

trips relating to month | Trips relating to days

Trips Every Month | Trips by Hour and Month

```
ggplot(month_hour, aes(hour, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Hour and Month") +
  scale_y_continuous(labels = comma) +
  scale_fill_manual(values = colors2)
```



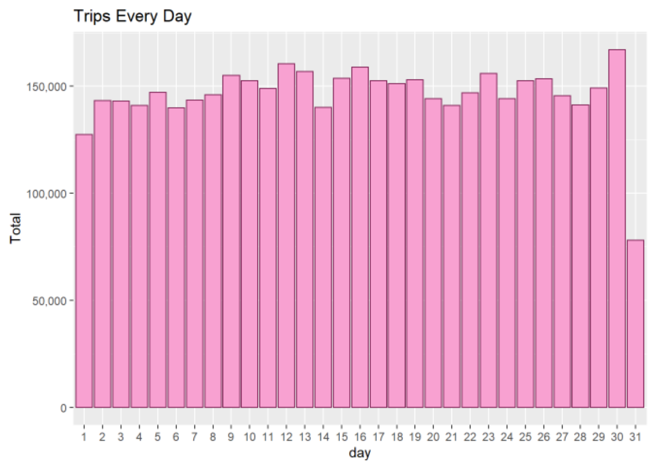
Dashboard

Taxi Trips visualization Dashboard

trips relating to month | Trips relating to days

Trips Every Day | Trips by Day and Month

```
ggplot(day_group, aes(day, Total)) +
  geom_bar( stat = "identity", fill = "#f8a1d1", color = "#822659") +
  ggtitle("Trips Every Day") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma)
```



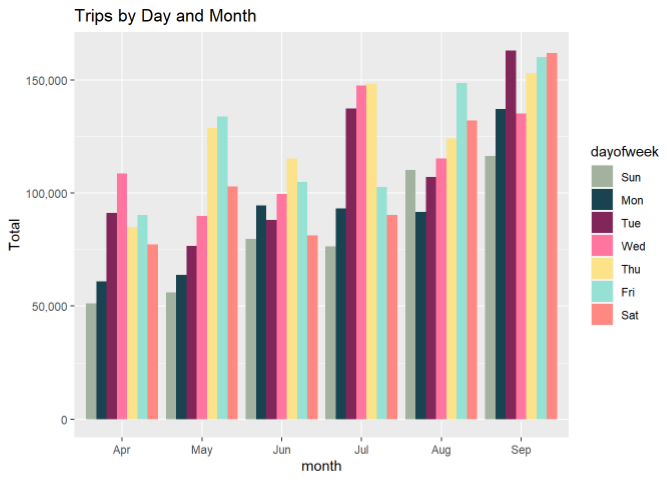
Dashboard

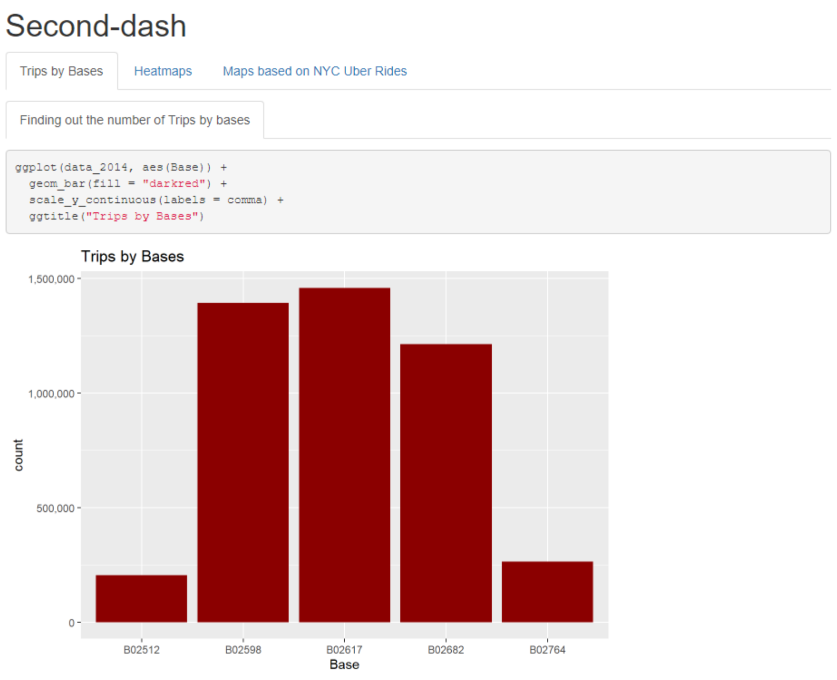
Taxi Trips visualization Dashboard

trips relating to month | Trips relating to days

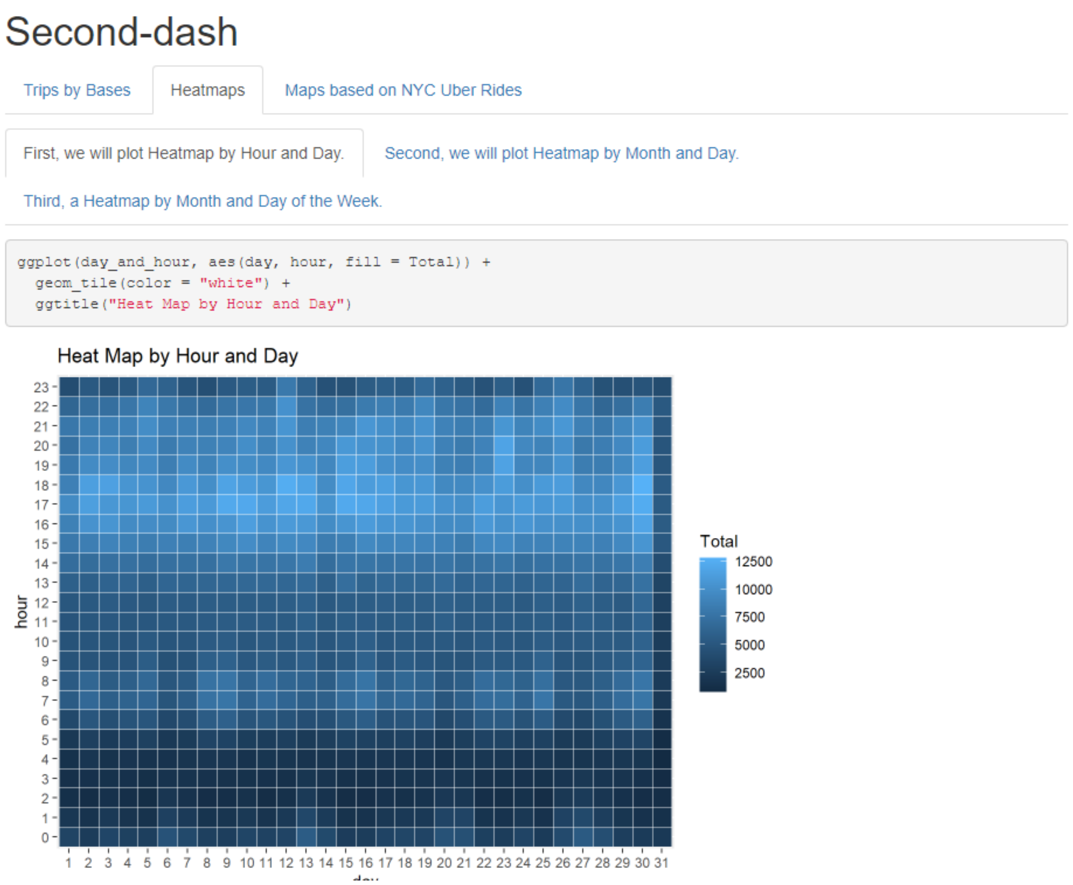
Trips Every Day | Trips by Day and Month

```
ggplot(month_weekday, aes(month, Total, fill = dayofweek)) +
  geom_bar( stat = "identity", position = "dodge") +
  ggtitle("Trips by Day and Month") +
  scale_y_continuous(labels = comma) +
  scale_fill_manual(values = colors3)
```





```
ggplot(data_2014, aes(Base)) +
  geom_bar(fill = "darkred") +
  scale_y_continuous(labels = comma) +
  ggtitle("Trips by Bases")
```



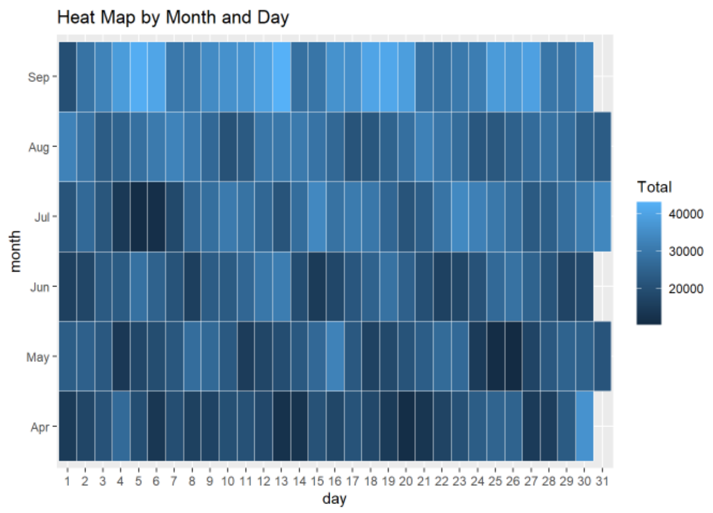
```
ggplot(day_and_hour, aes(day, hour, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Hour and Day")
```

Second-dash

Trips by Bases | Heatmaps | Maps based on NYC Uber Rides

First, we will plot Heatmap by Hour and Day. | Second, we will plot Heatmap by Month and Day. | Third, a Heatmap by Month and Day of the Week.

```
ggplot(day_month_group, aes(day, month, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Month and Day")
```

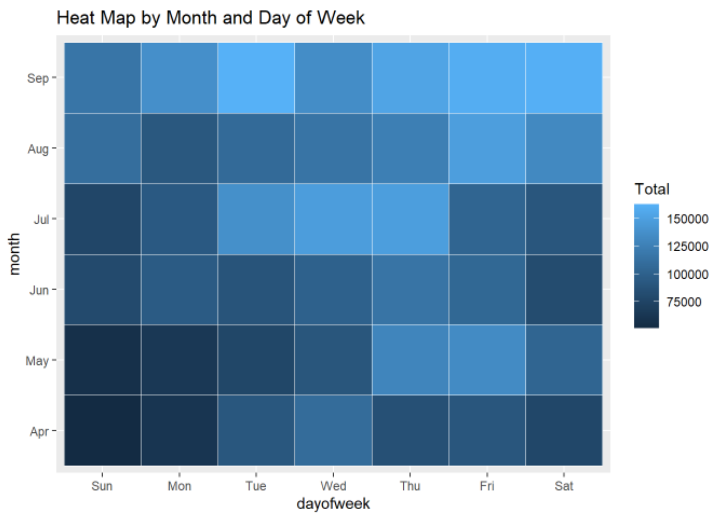


Second-dash

Trips by Bases | Heatmaps | Maps based on NYC Uber Rides

First, we will plot Heatmap by Hour and Day. | Second, we will plot Heatmap by Month and Day. | Third, a Heatmap by Month and Day of the Week.

```
ggplot(month_weekday, aes(dayofweek, month, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Month and Day of Week")
```



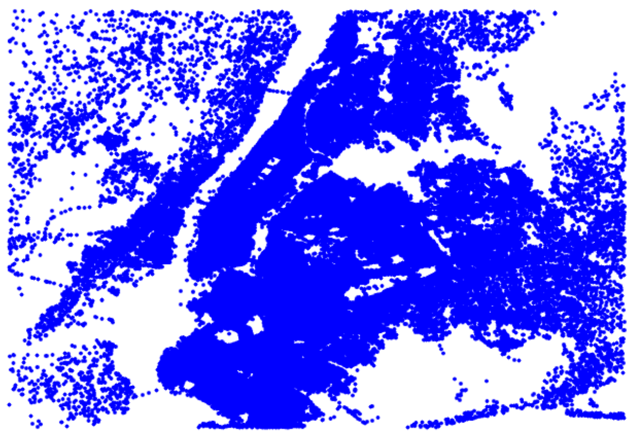
Second-dash

Trips by Bases | Heatmaps | Maps based on NYC Uber Rides

```r
min_lat <- 40.5774
max_lat <- 40.9176
min_long <- -74.15
max_long <- -73.7004
ggplot(data_2014, aes(x=Lon, y=Lat)) +
  geom_point(size=1, color = "blue") +
  scale_x_continuous(limits=c(min_long, max_long)) +
  scale_y_continuous(limits=c(min_lat, max_lat)) +
  theme_map() +
  ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)")
```

```
## Warning: Removed 71701 rows containing missing values (geom_point).
```

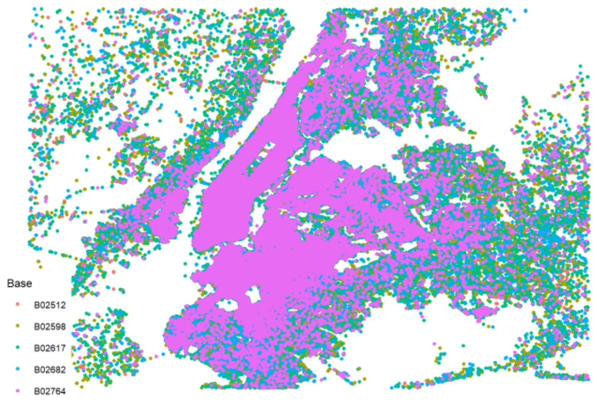
NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)



```r
ggplot(data_2014, aes(x=Lon, y=Lat, color = Base)) +
  geom_point(size=1) +
  scale_x_continuous(limits=c(min_long, max_long)) +
  scale_y_continuous(limits=c(min_lat, max_lat)) +
  theme_map() +
  ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by BASE")
```

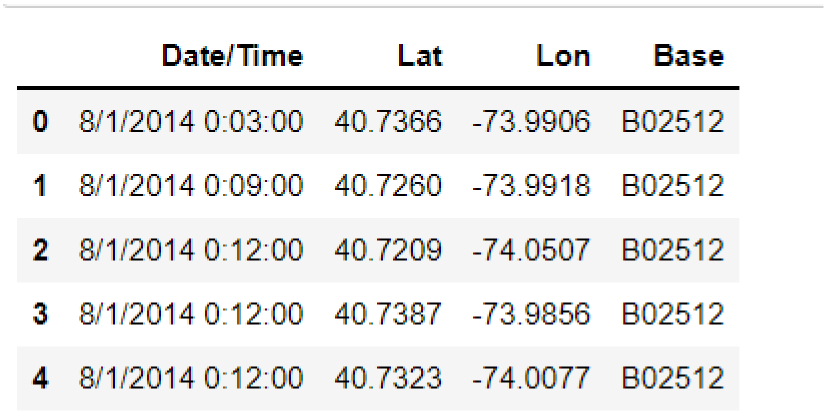
```
## Warning: Removed 71701 rows containing missing values (geom_point).
```

NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by BASE



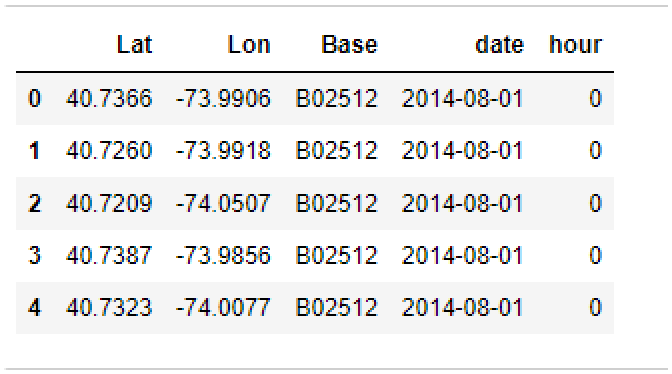
K-MEANS CLUSTERING MODEL IMPLEMENTED IN OUR PROJECT (USING PYTHON):

First five rows of our dataset:



	Date/Time	Lat	Lon	Base
0	8/1/2014 0:03:00	40.7366	-73.9906	B02512
1	8/1/2014 0:09:00	40.7260	-73.9918	B02512
2	8/1/2014 0:12:00	40.7209	-74.0507	B02512
3	8/1/2014 0:12:00	40.7387	-73.9856	B02512
4	8/1/2014 0:12:00	40.7323	-74.0077	B02512

First five rows of the dataset after we have separated the original “Date/Time” attribute into Date and Hour respectively:

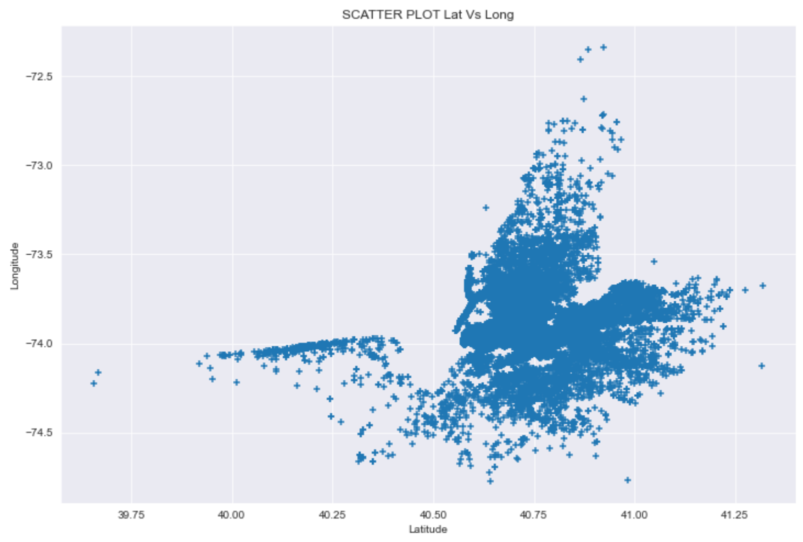


	Lat	Lon	Base	date	hour
0	40.7366	-73.9906	B02512	2014-08-01	0
1	40.7260	-73.9918	B02512	2014-08-01	0
2	40.7209	-74.0507	B02512	2014-08-01	0
3	40.7387	-73.9856	B02512	2014-08-01	0
4	40.7323	-74.0077	B02512	2014-08-01	0

SCATTER PLOT

The relationship between Latitude and Longitude plotted with the help of SCATTERPLOT to analyze the distribution of Cab Bookings in NEW-YORK

In [7]:
```
plt.figure(figsize=(12,8))
plt.title('SCATTER PLOT Lat Vs Long')
plt.ylabel('Longitude')
plt.xlabel('Latitude')
plt.scatter(df['Lat'],df['Lon'],marker='+')
plt.show()
```

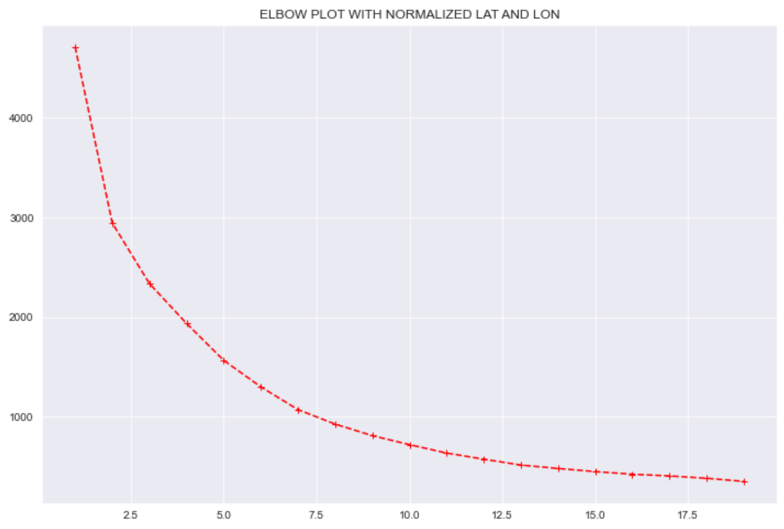


ELBOW PLOT

To find a correct value of K based on Sum of Squared Errors versus Number of Clusters to create a model model which is neither overfitted nor underfitted.

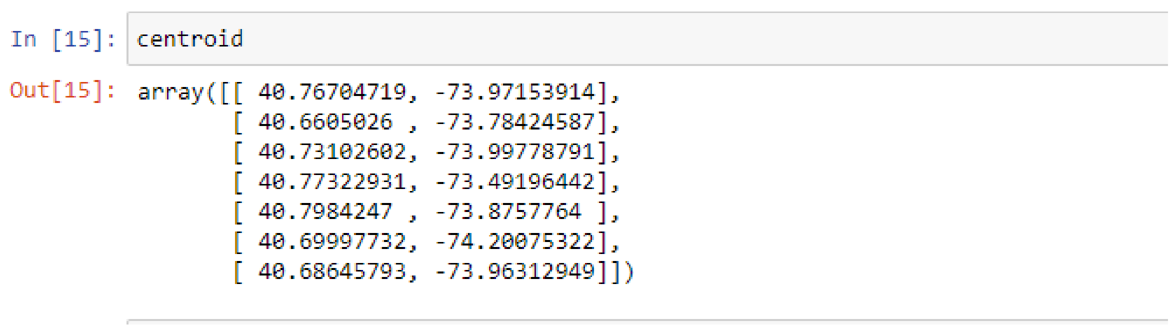
The value of K should be taken at the point of rapid change.(the shape of the plot around this region will be bent like a elbow)

In [11]:
```
plt.figure(figsize=(12,8))
plt.title('ELBOW PLOT WITH NORMALIZED LAT AND LON')
plt.plot(range(1,20),sse,'r+--')
plt.show()
```



All the visualization of the remaining plots are done after using K-MEANS CLUSTERING MODEL on our dataset. The base stations derived from K-MEANS CLUSTERING are assigned the numbers from 0 to 6. Also the dataset we have used is updated with the new base station information.

Location of the new base stations obtained (centroids of the cluster)



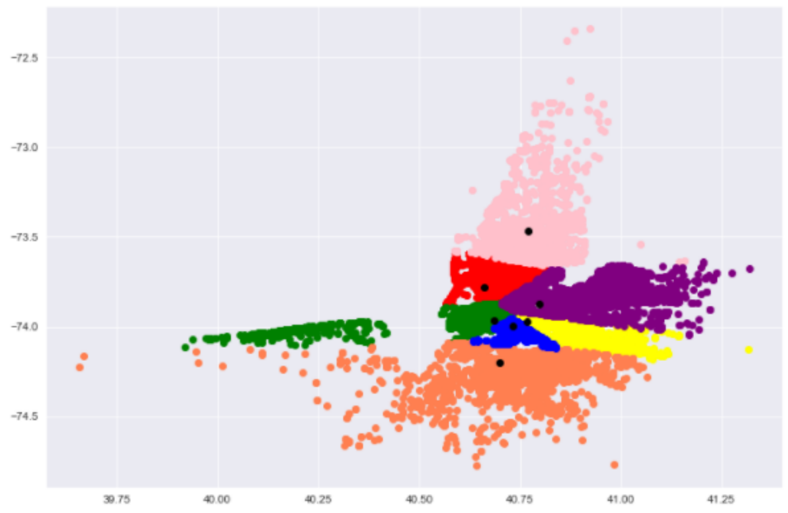
```
In [15]: centroid

Out[15]: array([[ 40.76704719, -73.97153914],
                [ 40.6605026 , -73.78424587],
                [ 40.73102602, -73.99778791],
                [ 40.77322931, -73.49196442],
                [ 40.7984247 , -73.8757764 ],
                [ 40.69997732, -74.20075322],
                [ 40.68645793, -73.96312949]])
```

SCATTER PLOT AFTER CLUSTERING.

THE NUMBER OF CLUSTERS,K=7 IS CHOSEN. THE 7 CLUSTERS ARE REPRESENTED WITH DIFFERENT COLORS IN THE SCATTERPLOT PROVIDED BELOW.

THE CENTROIDS OF EACH CLUSTERS ARE MARKED WITH BLACK(CIRCLE) DOTS.

In [14]:
```
a=df[df['BASE']==0]
b=df[df['BASE']==1]
c=df[df['BASE']==2]
d=df[df['BASE']==3]
e=df[df['BASE']==4]
f=df[df['BASE']==5]
g=df[df['BASE']==6]
plt.figure(figsize=(12,8))
plt.scatter(a['Lat'],a['Lon'],color='red')
plt.scatter(b['Lat'],b['Lon'],color='yellow')
plt.scatter(c['Lat'],c['Lon'],color='green')
plt.scatter(d['Lat'],d['Lon'],color='coral')
plt.scatter(e['Lat'],e['Lon'],color='blue')
plt.scatter(f['Lat'],f['Lon'],color='pink')
plt.scatter(g['Lat'],g['Lon'],color='purple')
for i in range (7):
    plt.scatter(centroid[i][0],centroid[i][1],color='black')
plt.show()
```

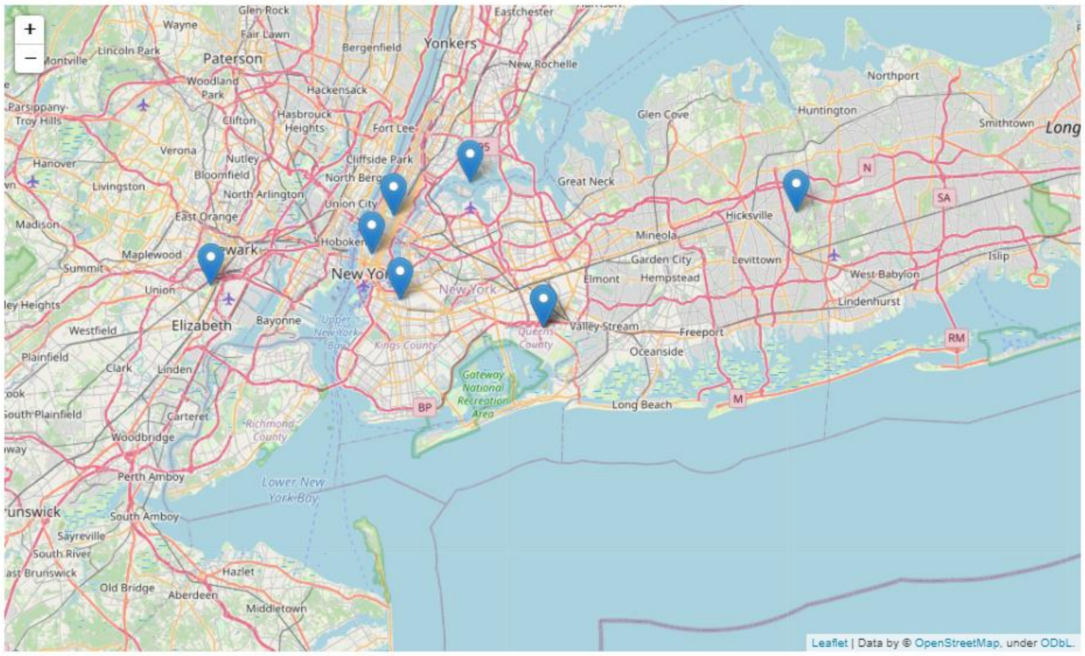


REAL WORLD MAP

THE NEW BASES (ie the new centroids of each clusters) are mapped in world map using folium library to get a better understanding of our new found bases.

```
In [32]: #LOCATION OF THE NEW BASES
         import folium
         map=folium.Map(location=[centroid[0][0],centroid[0][1]],zoom_start=10)
         map.add_child(folium.Marker(location=[centroid[0][0],centroid[0][1]],popup='tkd'))
         for i in range(7):
             map.add_child(folium.Marker(location=[centroid[i][0],centroid[i][1]],popup='tkd'))
         map
```

Out[32]:



Predicting the base station from which the uber cab has to be sent to pick-up the customer based on the pick-up location

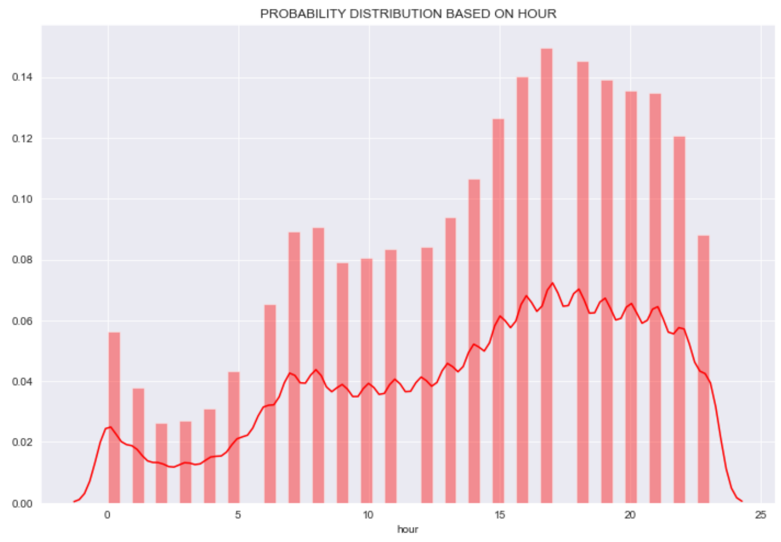
```
In [17]: #ENTER LOCATION(latitude and longitude) IN THE FORM OF 2D-ARRAY
         def get_base(location):
             x=kmeans.predict(location)
             print(f'CLUSTER NUMBER : {x}  CLUSTER CO-ORDINATE : {centroid[x]}')
```

```
In [18]: get_base([[40.1,-73.2]])

         CLUSTER NUMBER : [3]  CLUSTER CO-ORDINATE : [[ 40.77322931 -73.49196442]]
```

DISTRIBUTION PLOT WITH KERNAL DENSITY ESTIMATION (KDE)

Histographic representation of Cab Bookings based on Hour

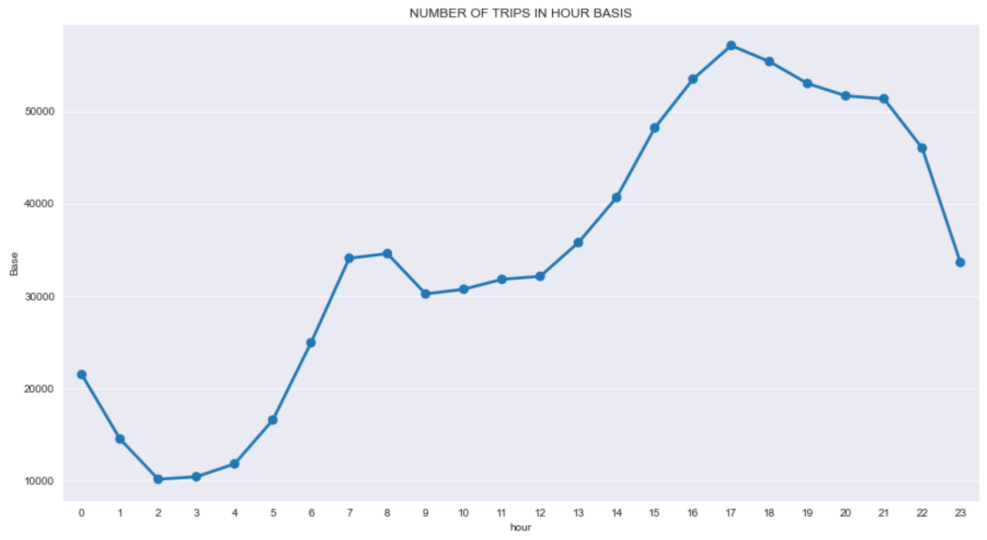
In [20]:
```
plt.figure(figsize=(12,8))
plt.title('PROBABILITY DISTRIBUTION BASED ON HOUR')
sns.distplot(df['hour'],color='red')
plt.show()
```



LINE PLOT : TRIPS VS HOUR

From the line plot we can infer that the maximum number of trips occurs from evening 4pm to night 10 pm.

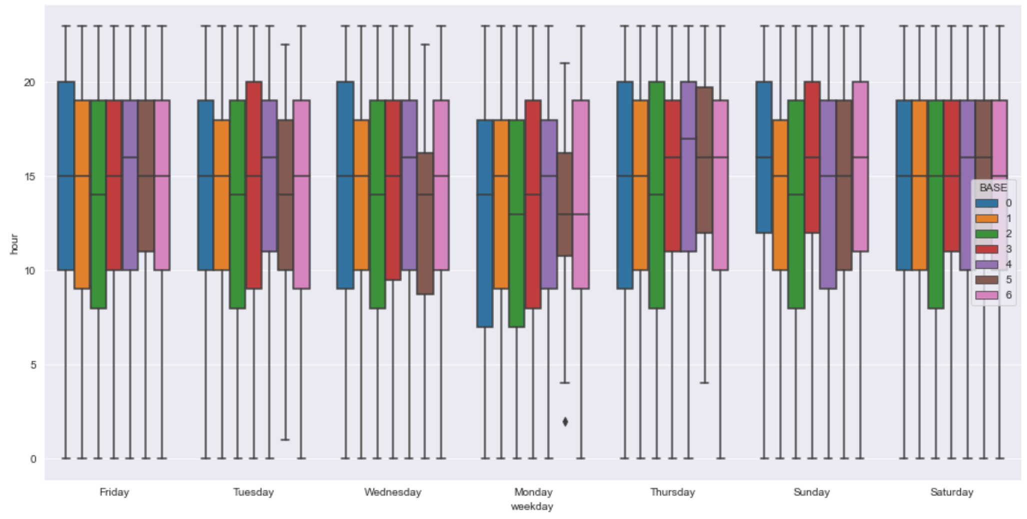
In [21]:
```
plt.figure(figsize=(15,8))
y=df.groupby(['hour']).count()['Base']
plt.title('NUMBER OF TRIPS IN HOUR BASIS')
sns.pointplot(x=y.reset_index()['hour'],y=y)
plt.show()
```



BOX PLOT

TO GET AN ACCURATE DESCRIPTION OF TRIPS ,THE BOX PLOT TAKES HOUR,BASE AND WEEKDAYS INTO CONSIDERATION. THE MEDIAN OF TRIP TIME IS AROUND 3PM AND MAXIMUM TRIPS OCCUR AFTER 5.

```
In [22]: plt.figure(figsize=(16,8))
         sns.boxplot(x=df['weekday'],y=df['hour'],hue=df['BASE'])
         plt.show()
```

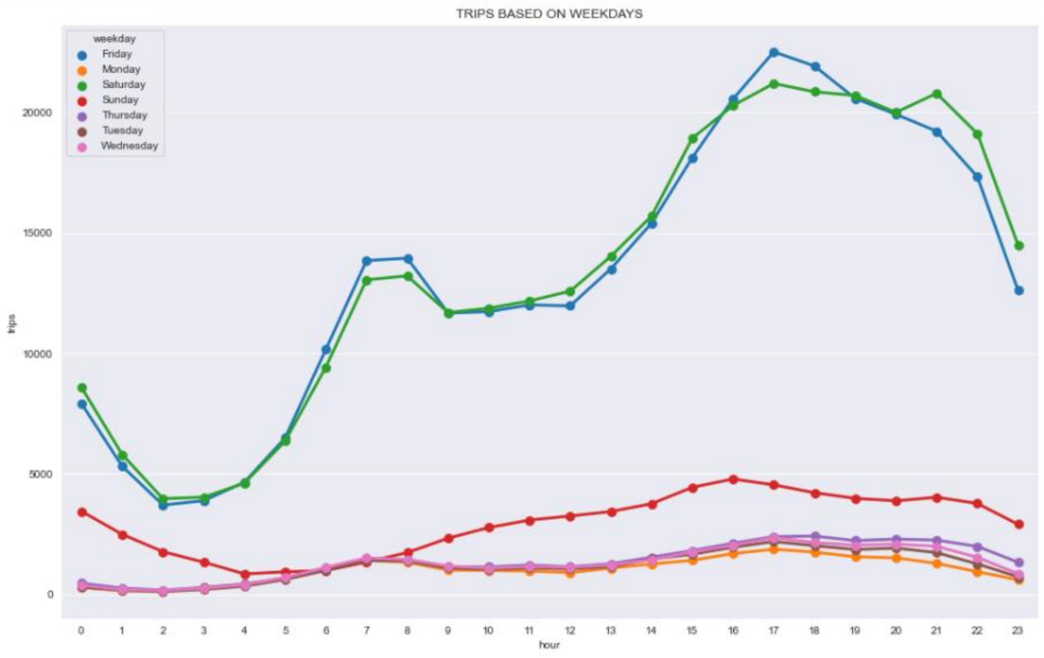


LINE PLOT : TRIPS VS HOUR (HUE= DAY)

IT IS OBVIOUS FROM THE GRAPH THAT FRIDAY AND SATURDAY HAS HIGHER NUMBER OF TRIPS.

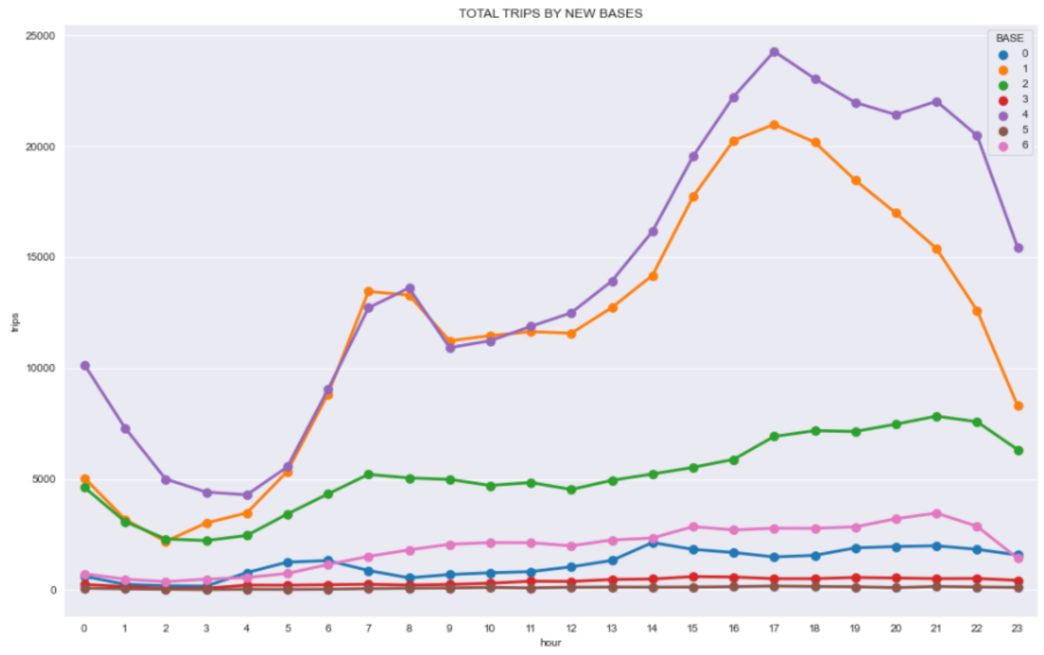
4PM-10PM : THE TRIPS ON SATURDAYS AND FRIDAYS ARE 4 TIMES HIGHER WHEN COMPARED TO REST OF THE WEEK.

```
In [23]: plt.figure(figsize=(16,10))
         y=df.groupby(['hour','weekday'])['date'].count().reset_index()
         plt.title('TRIPS BASED ON WEEKDAYS')
         y['trips']=y['date']
         sns.pointplot(x=y.hour,y=y.trips,hue=y.weekday)
         plt.show()
```

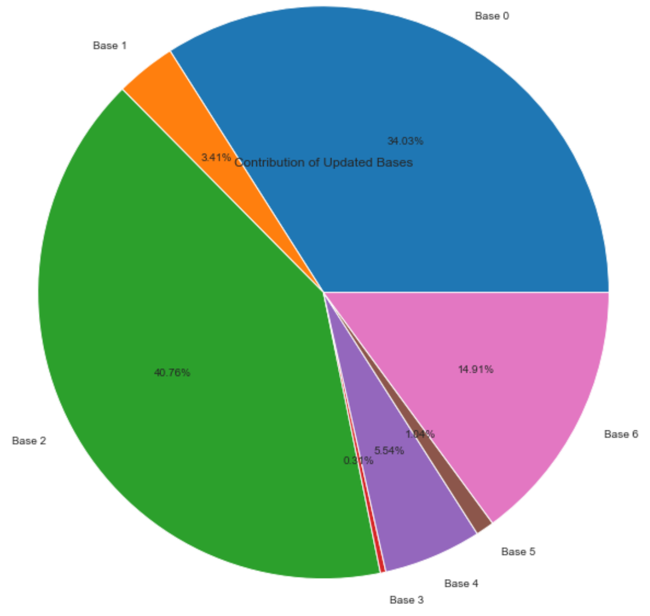


LINE PLOT : TRIPS VS HOUR (HUE= BASES)

```
In [25]: plt.figure(figsize=(16,10))
         y=df.groupby(['hour','BASE'])['date'].count().reset_index()
         plt.title('TOTAL TRIPS BY NEW BASES')
         y['trips']=y['date']
         sns.pointplot(x=y.hour,y=y.trips,hue=y.BASE)
         plt.show()
```



```
In [27]: l=[]
         for i in range(0,7):
             l.append(df.loc[df['BASE']==i].count()['hour'])
         plt.title('Contribution of Updated Bases')
         plt.pie(l,autopct='%0.2f%%',labels=['Base 0','Base 1','Base 2','Base 3','Base 4','Base 5','Base 6'],radius=3,
                explode=[0,0,0,0,0,0,0])
         plt.show()
```



CONCLUSION

Using the visualization, the basic aim of the project is fulfilled which was to analyze and study the patterns of the Uber Cab trips in New York. The different visualizations portrayed help us identify different times of the day, days of the week or areas made by trips to analyze what is the most suitable time and day when the cabs are to be increased. We have also implemented the k-means algorithms to identify clusters. All of these clusters have a centroid, which is the proposed location to be set up as a base for maximum efficiency of cabs. These clusters will help the agency to be more central to the cab requests in their respective clusters. Finally we have visualized the data of what might be the outcomes after setting up the bases in new locations. This research has incorporated data visualization and analysis by implementing K-means clustering which is an unsupervised machine learning algorithm to enhance the productivity of cab trips for Uber.

Future work in this area might include implementation of the same technique using various other machine learning algorithms like Gaussian Mixture Model, DBScan, Mean Shift clustering, etc. On comparing the results of these algorithms, we can continue to provide much better and efficient results.

REFERENCES

1. Huiyu Sun, Siyuan Hu, Suzanne McIntosh, Yi Cao, "*Big Data Trip Classification on the New York City Taxi and Uber Sensor Network*," Journal of Internet Technology, vol. 19, no. 2 , pp. 591-598, Mar. 2018.
2. *Analysing Uber Trips using PySpark*, Rahul Pradhan et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1119 012013
3. Li, Kebing, "*Investigating the effect crime has on Uber and Yellow Taxi pickups in NYC*" (2019). Honors Theses. Paper 924.
4. Diego Correa Ph.D., Kun Xie Ph.D., Kaan Ozbay Ph.D. (January, 2017), "*Exploring the Taxi and Uber Demands in New York City: An Empirical Analysis and Spatial Modeling"*
5. Junghoon Lee, Inhye Shin, Gyung-Leen Park (2008), "*Analysis of the passenger pick-up pattern for taxi location recommendation"*
6. Elmer R. Magsino et al (24-25 May 2021), "*Taxi Hotspots Identification through Origin and Destination Analysis of Taxi Trips using K-means Clustering and H-indexing",* Journal of Physics: Conference Series
7. Nivan Ferreira, Jorge Poco, Huy T. Vo, Juliana Freire, and Cl´audio T. Silva (Dec. 2013), "*Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips"*, IEEE Transactions on Visualization and Computer Graphics (Volume: 19, Issue: 12)

8. Ying Shen, Ligang Zhao and Jing Fan (2015), "*Analysis and Visualization for Hot Spot Based Route Recommendation Using Short-Dated Taxi GPS Traces",* Special Issue Intelligent Data Analysis
9. Ding Chu, David A Sheets, Ye Zhao, Yingyu Wu (March 2014), "*Visualizing Hidden Themes of Taxi Movement with Semantic Transformation"*, 2014 IEEE Pacific Visualization Symposium
10. M A Syakur, B K Khotimah, E M S Rochman and B D Satoto (2018), "*Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster"*, IOP Conference Series: Materials Science and Engineering (Vol. 336)

APPENDIX A

Implemented Code

Code for visualization using flexdashboard in R:

~~~
---

title: "Dashboard"

output:

flexdashboard::flex_dashboard:

        orientation: columns

        vertical_layout: fill

---


```{r setup, include=FALSE}

library(flexdashboard)

library(ggplot2)

library(ggthemes)

library(lubridate)

library(dplyr)

library(tidyr)

library(DT)

library(scales)


apr_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-

apr14.csv",header=TRUE,sep=",")

attach(apr_data)
~~~

```
may_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-
may14.csv",header=TRUE,sep=",")
attach(may_data)
jun_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-
jun14.csv",header=TRUE,sep=",")
attach(jun_data)
jul_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-
jul14.csv",header=TRUE,sep=",")
attach(jul_data)
aug_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-
aug14.csv",header=TRUE,sep=",")
attach(aug_data)
sep_data <- read.delim(file="D:/VIT/5th Sem/DataViz/Project/uber-dataset/uber-raw-data-
sep14.csv",header=TRUE,sep=",")
attach(sep_data)


colors1 = c("#CC1011", "#665555", "#05a399", "#cfcaca", "#f5e840", "#0683c9", "#e075b0")
colors2 = c("#3edbf0", "#04009a", "#77acf1", "#c0fefc", "#e84545", "#f05945", "#007580")
colors3 = c("#a2b29f", "#194350", "#822659", "#ff75a0", "#fce38a", "#95e1d3", "#ff8882")

data_2014 <- rbind(apr_data, may_data, jun_data, jul_data, aug_data, sep_data)

data_2014$Date.Time <- as.POSIXct(data_2014$Date.Time, format = "%m/%d/%Y%H:%M:%S")

data_2014$Time <- format(as.POSIXct(data_2014$Date.Time, format = "%m/%d/%Y%H:%M:%S"),
format="%H:%M:%S")
```

~~~
data_2014$Date.Time <- ymd_hms(data_2014$Date.Time)


data_2014$day <- factor(day(data_2014$Date.Time))

data_2014$month <- factor(month(data_2014$Date.Time, label = TRUE))


data_2014$year <- factor(year(data_2014$Date.Time))


data_2014$dayofweek <- factor(wday(data_2014$Date.Time, label = TRUE))


data_2014$hour <- factor(hour(hms(data_2014$Time)))


data_2014$minute <- factor(minute(hms(data_2014$Time)))


data_2014$second <- factor(second(hms(data_2014$Time)))


hour_data <- data_2014 %>% group_by(hour) %>%
  dplyr::summarize(Total = n())


month_hour <- data_2014 %>% group_by(month, hour) %>% dplyr::summarize(Total = n())


day_group <- data_2014 %>% group_by(day) %>% dplyr::summarize(Total = n())


day_month_group <- data_2014 %>% group_by(month, day) %>%  dplyr::summarize(Total = n())
month_group <- data_2014 %>% group_by(month) %>% dplyr::summarize(Total = n())
month_weekday <- data_2014 %>%  group_by(month, dayofweek) %>%  dplyr::summarize(Total = n())
day_and_hour <- data_2014 %>%  group_by(day, hour) %>%  dplyr::summarize(Total = n())
```
~~~

~~~
Taxi Trips visualization Dashboard{.tabset}
==============================


trips relating to month{.tabset}
---------------------------------------------------------------


### Trips Every Month
```{r}
ggplot(month_group, aes(month, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips Every Month") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma) +
  scale_fill_manual(values = colors2)
```


### Trips by Hour and Month
```{r}
ggplot(month_hour, aes(hour, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Hour and Month") +
  scale_y_continuous(labels = comma) +
  scale_fill_manual(values = colors2)
```


Trips relating to days {.tabset}
--------------------------------------------------------------------
~~~

~~~
### Trips Every Day

```{r}

ggplot(day_group, aes(day, Total)) +

  geom_bar( stat = "identity", fill = "#f8a1d1", color = "#822659") +

  ggtitle("Trips Every Day") +

  theme(legend.position = "none") +

  scale_y_continuous(labels = comma)

```


### Trips by Day and Month

```{r}

ggplot(month_weekday, aes(month, Total, fill = dayofweek)) +

  geom_bar( stat = "identity", position = "dodge") +

  ggtitle("Trips by Day and Month") +

  scale_y_continuous(labels = comma) +

  scale_fill_manual(values = colors3)

```


Second-dash {.tabset}

=============================


Trips by Bases {.tabset}

---------------------------------------------------------------------


### Finding out the number of Trips by bases

```{r}

ggplot(data_2014, aes(Base)) +
~~~

~~~
  geom_bar(fill = "darkred") +
  scale_y_continuous(labels = comma) +
  ggtitle("Trips by Bases")
```


Heatmaps {.tabset}
---------------------------------------------------------------------


### First, we will plot Heatmap by Hour and Day.
```{r}
ggplot(day_and_hour, aes(day, hour, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Hour and Day")
```


### Second, we will plot Heatmap by Month and Day.
```{r}
ggplot(day_month_group, aes(day, month, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Month and Day")
```



### Third, a Heatmap by Month and Day of the Week.
```{r}
ggplot(month_weekday, aes(dayofweek, month, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Month and Day of Week")
~~~

~~~
```


###Fourth, a Heatmap that delineates Month and Bases.
```{r}
month_base <-  data_2014 %>%
  group_by(Base, month) %>%
  dplyr::summarize(Total = n())
day0fweek_bases <-  data_2014 %>%
  group_by(Base, dayofweek) %>%
  dplyr::summarize(Total = n())
ggplot(month_base, aes(Base, month, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Month and Bases")
```


###Fifth, plot the heatmap, by bases and day of the week.
```{r}
ggplot(day0fweek_bases, aes(Base, dayofweek, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Bases and Day of Week")
```


Maps based on NYC Uber Rides {.tabset}
---------------------------------------------------------------------


```{r}
min_lat <- 40.5774
max_lat <- 40.9176
~~~

```
min_long <- -74.15

max_long <- -73.7004

ggplot(data_2014, aes(x=Lon, y=Lat)) +

  geom_point(size=1, color = "blue") +

  scale_x_continuous(limits=c(min_long, max_long)) +

  scale_y_continuous(limits=c(min_lat, max_lat)) +

  theme_map() +

  ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)")
```

```{r}
ggplot(data_2014, aes(x=Lon, y=Lat, color = Base)) +

  geom_point(size=1) +

  scale_x_continuous(limits=c(min_long, max_long)) +

  scale_y_continuous(limits=c(min_lat, max_lat)) +

  theme_map() +

  ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by BASE")
```

Code for implementation of K-means clustering using python:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df = pd.read_csv('uber.csv')

df.head()

df['time']=df['Date/Time'].apply(lambda x : x[-8:-1])
```

```
df['month']=df['Date/Time'].apply(lambda x : x[0])

df['day']=df['Date/Time'].apply(lambda x : x[2])

df['day']=df['day'].apply(lambda x : '-'+x)

df['month']=df['month'].apply(lambda x : '-'+x)

df['year']='2014'

df['date']=df['year']+df['month']+df['day']

df.drop(columns=['day','month','year'],inplace=True)

df.head()

df.drop(columns=['Date/Time'],inplace=True)

df['hour']=df['time'].apply(lambda x : x[0:2])

df['hour']=df['hour'].astype('int64')

df.drop(columns=['time'],inplace=True)

df['date']=df['date'].astype('datetime64')

df.head()

df.info()

sns.set_style('darkgrid')

plt.figure(figsize=(12,8))

plt.title('SCATTER PLOT Lat Vs Long')

plt.ylabel('Longitude')

plt.xlabel('Latitude')

plt.scatter(df['Lat'],df['Lon'],marker='+')

plt.show()

from sklearn.cluster import KMeans

df.head()

sse=[]

for k in range(1,20):

    kmeans=KMeans(n_clusters=k)

    kmeans.fit_predict(df[['Lat','Lon']])
    sse.append(kmeans.inertia_)
plt.figure(figsize=(12,8))
plt.title('ELBOW PLOT WITH NORMALIZED LAT AND LON')
plt.plot(range(1,20),sse,'r+--')
plt.show()
kmeans=KMeans(n_clusters=7)
a=kmeans.fit_predict(df[['Lat','Lon']])
centroid= kmeans.cluster_centers_
df['BASE']=a
df.head()
a=df[df['BASE']==0]
b=df[df['BASE']==1]
c=df[df['BASE']==2]
d=df[df['BASE']==3]
e=df[df['BASE']==4]
f=df[df['BASE']==5]
g=df[df['BASE']==6]
plt.figure(figsize=(12,8))
plt.scatter(a['Lat'],a['Lon'],color='red')
plt.scatter(b['Lat'],b['Lon'],color='yellow')
plt.scatter(c['Lat'],c['Lon'],color='green')
plt.scatter(d['Lat'],d['Lon'],color='coral')
plt.scatter(e['Lat'],e['Lon'],color='blue')
plt.scatter(f['Lat'],f['Lon'],color='pink')
plt.scatter(g['Lat'],g['Lon'],color='purple')
for i in range (7):
    plt.scatter(centroid[i][0],centroid[i][1],color='black')
plt.show()
```

```
centroid

#LOCATION OF THE NEW BASES

import folium

map=folium.Map(location=[centroid[0][0],centroid[0][1]],zoom_start=10)

map.add_child(folium.Marker(location=[centroid[0][0],centroid[0][1]],popup='tkd'))

for i in range(7):

    map.add_child(folium.Marker(location=[centroid[i][0],centroid[i][1]],popup='tkd'))

map

#ENTER LOCATION(latitude and longitude) IN THE FORM OF 2D-ARRAY

def get_base(location):

    x=kmeans.predict(location)

    print(f'CLUSTER NUMBER : {x}  CLUSTER CO-ORDINATE : {centroid[x]}')

get_base([[40.1,-73.2]])

df['weekday']=df['date'].dt.day_name()

df.sort_values('hour',ascending=True,inplace=True)

df.head()

plt.figure(figsize=(12,8))

plt.title('PROBABILITY DISTRIBUTION BASED ON HOUR')

sns.distplot(df['hour'],color='red')

plt.show()

plt.figure(figsize=(15,8))

y=df.groupby(['hour']).count()['Base']

plt.title('NUMBER OF TRIPS IN HOUR BASIS')

sns.pointplot(x=y.reset_index()['hour'],y=y)

plt.show()

plt.figure(figsize=(16,8))

sns.boxplot(x=df['weekday'],y=df['hour'],hue=df['BASE'])

plt.show()
```

```
df['weekday'].value_counts()

plt.figure(figsize=(16,10))

y=df.groupby(['hour','weekday'])['date'].count().reset_index()

plt.title('TRIPS BASED ON WEEKDAYS')

y['trips']=y['date']

sns.pointplot(x=y.hour,y=y.trips,hue=y.weekday)

plt.show()

df["BASE"].value_counts()

plt.figure(figsize=(16,10))

y=df.groupby(['hour','BASE'])['date'].count().reset_index()

plt.title('TOTAL TRIPS BY NEW BASES')

y['trips']=y['date']

sns.pointplot(x=y.hour,y=y.trips,hue=y.BASE)

plt.show()

l=[]

for i in range(0,7):

    l.append(df.loc[df['BASE']==i].count()['hour'])

plt.title('Contribution of Updated Bases')

plt.pie(l,autopct='%0.2f%%',labels=['Base 0','Base 1','Base 2','Base 3','Base 4','Base 5','Base 6'],radius=3,

      explode=[0,0,0,0,0,0,0])

plt.show()
```