

**Computer Science and Engineering**

**Big Data Analytics Project**

**Exploring interactions between urban components**

**Project Report**

* Team Name: GoAsians
* Team Members:

|  |  |
| --- | --- |
| **Names** | **Net-ID** |
| Vaibhav Aggarwal | [va771@nyu.edu](mailto:Va771@nyu.edu) |
| Jitao Guo | jg4397@nyu.edu |
| Palak Arora | pa1154@nyu.edu |

**TABLE OF CONTENTS**

Table of Contents

[ABSTRACT 4](#_Toc451119224)

[Visualizations: 5](#_Toc451119225)

[14. Payment Mode for Yellow Taxi Users Analysis 18](#_Toc451119226)

[Scripts/Code: 21](#_Toc451119227)

[Steps on how to run the scripts: 22](#_Toc451119228)

[On Amazon AWS. 22](#_Toc451119229)

[**Steps for Local** 22](#_Toc451119230)

[**Steps for visualization of graph** 22](#_Toc451119231)

[Experimental Setup 23](#_Toc451119232)

[Cluster Configuration: 23](#_Toc451119233)

[**Tools Used**: 23](#_Toc451119234)

[Optimizations applied to the script: 23](#_Toc451119235)

[Individual contributions: 23](#_Toc451119236)

**Exploring interactions between urban components**

**Topic:** Exploring interactions between urban components

**Dataset:** Weather and Citi Bike

* **Weather Dataset from NYC (year of 2015)**

Available at: https://nyu.box.com/s/4lkrxs9rdsfjzpu1gh9nwen89jxtc9dd

Metadata available at: https://nyu.box.com/s/461edjve0obbsefcpaygb1h9z8yc49c4

Temporal data is in GMT (hour resolution).

Single dataset for the entire city.

* **2015 Yellow Taxi Dataset**

Available at http://www.nyc.gov/html/tlc/html/about/trip\_record\_data.shtml

Collect all the data files from 2015 (yellow taxi only)

Metadata available at: http://www.nyc.gov/html/tlc/downloads/pdf/data\_dictionary\_trip\_records\_yellow.pdf

Temporal data is in EST (second resolution).

Spatial data is in GPS.

* **Citi Bike Trip Histories**

Available at: https://www.citibikenyc.com/system-data ("Citi Bike Trip Histories" section)

Temporal data is in EST (second resolution).

Spatial data is in GPS.

# ABSTRACT

Urban data is unique in that it captures the behavior of the different components of a city, namely its citizens, existing infrastructure (physical and policies), the environment (e.g.: weather), and interactions between these elements. The availability of these data makes it possible to not only better understand the individual components but also obtain insights into how they interact.

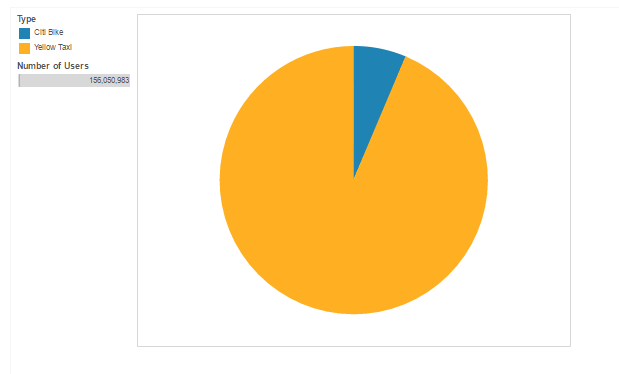
The Project analyses the Dataset from Weather, Citi Bike Data and Yellow Taxi Data.

The various datasets are joined, and analyzed for various attributes of weather.

The Visualizations are done for various types of joins. Hadoop MapReduce is used to join the tables, and form various use cases.

# Visualizations:

1. **Pie Chart for Citi Bike and Yellow Taxi Users in 2015**



The Pie Chart depicts the total number of users for Yellow taxi and Citi Bike in the year 2015.

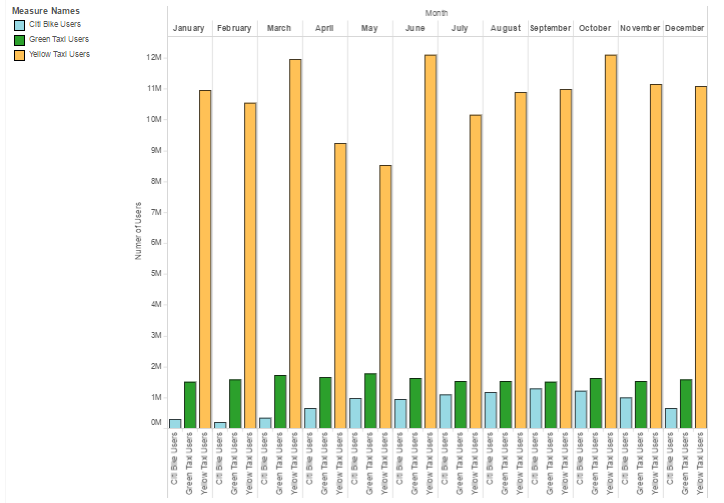
**MAPPER:**

Mapper prints the user details as key and 1 as count for both Citi Bike and Yellow Taxi Tables. The Mapper takes the enitre dataset of Citi Bike and Yellow Taxi as input.

**REDUER:**

Reducer counts the number of users belonging to the same key and prints the output.

1. **User count for Yellow taxi and Citi bike per month**



The Side-By-Side Bar Graph depicts the total number of Citi Bike and Yellow Taxi Users per month. The X-axis denotes the month and the Y-axis denotes the number of Citi Bike and Yellow Taxi Users.

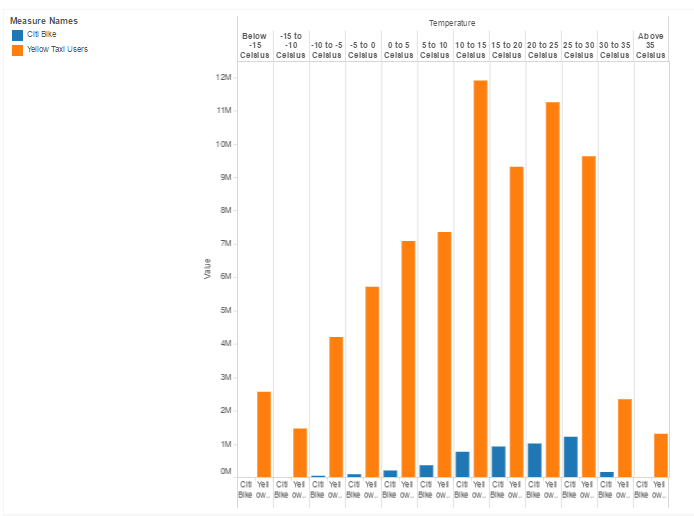
**MAPPER**:

The Mapper takes the Citi Bike and Yellow Taxi data as input and prints the key-value pairs. The Key is the month number and Citi/Yellow tag. Example: 01Citi for January Citi Bike User. The count is 1.

**REDUER:**

The Reducer takes the key-value pairs and counts the number of keys and prints them. The output looks like: January Citi 12345, January Yellow 4658273.

1. **Temperature vs Total Citi bike and Yellow Taxi Users for Year 2015**



The Side-By-Side Bar Graph depicts the total number of users for Yellow taxi and Citi Bike Users for various Temperature Ranges. The X-axis denotes the temperature scale with a range span of 5 Fahrenheit and the Y-axis denotes the number of users for Yellow taxi and Citi bike within that particular temperature range.

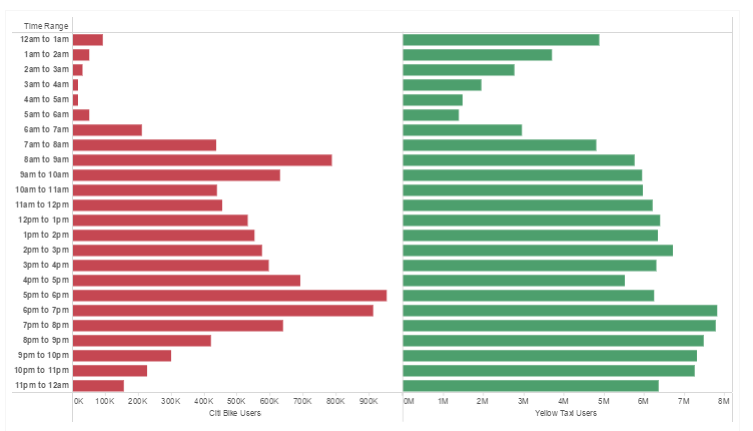
**MAPPER:**

The Mapper takes the joined tables of Citi Bike and Weather and Yellow Taxi and Weather. Primary key is YYYYMMDDHH. Temperature is extracted and printed along with Citi Bike or Yellow Taxi depending on whether the row belongs to Citi Bike or Yellow Taxi data.

**REDUCER:**

The Reducer then counts the number of Citi Bike Users and Yellow Taxi Users for different temperature ranges and prints them.

1. **Citi Bike/ Yellow Taxi users during various Time Slots**



The Bar graph depicts the number of Citi bike Users and Yellow Taxi users for a particular time slot. The time range taken is 1 hour.

From the above graph, we can observe that most Citi Bike users are most likely to ride a bicycle during evening, around 5pm to 7pm. The reason is most likely that people get done with their jobs and ride bicycle when they get free. Whereas, for Yellow Taxi Users, the peak time is noted between 6pm to 7pm. This clearly indicates that the office time gets over and people leave for their homes via Yellow Taxi.

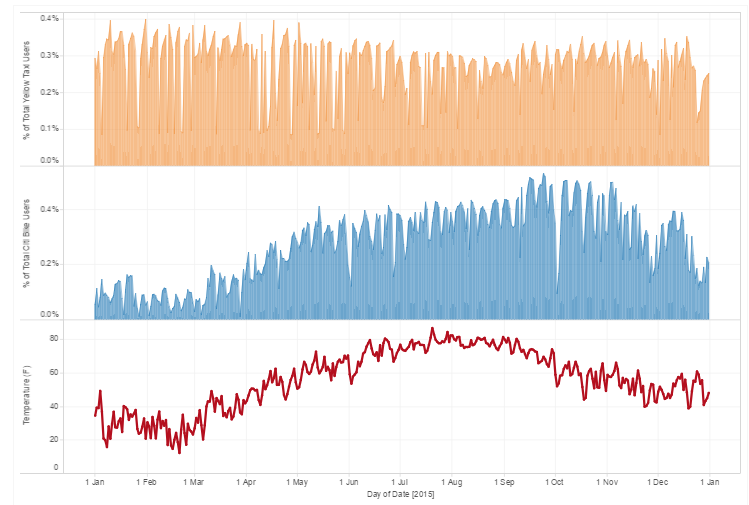
**MAPPER:**

Mapper extracts the time(only hour) from the primary key(Date Time YYYYMMDDHH) from the yellow taxi and citi bike table and prints the desired time slot.

**REDUCER:**

Reducer counts the number of citi bike and yellow taxi users for every time slot and prints it.

1. **User Percentage vs Day graph**



The Area Graph depicts the Percentage of total number Yellow taxi, and Citi Bike users against Temperature (in F) for each day in 2015. The X-axis denotes each day of year 2015 and the Y-axis denotes the temperature and the Percentage of total users.

From the graph, we can observe that at low temperatures, New Yorkers prefer Yellow Taxi over Citi Bike. We all know how cold it gets during winters. Hence, it’s almost impossible to ride a bicycle during winters which explains why New Yorkers prefer Yellow Taxi over Citi Bike during cold.

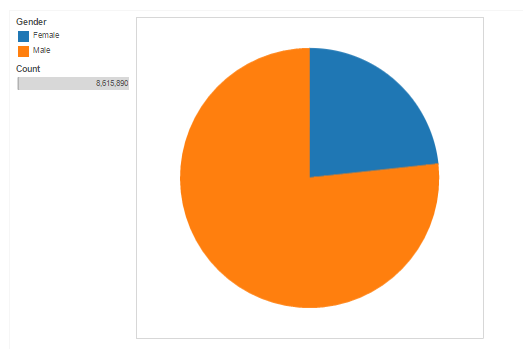
**MAPPER:**

The Mapper the takes joined data of Citi Bike and Yellow Taxi and sets the key as YYYYMMDD. The Mapper prints the users for every day along with temperature for each and every day of 2015.

**REDUCER:**

The Reducer counts the number of Citi Bike Users and Yellow Taxi Users for every day. Also, average temperature (in F) is calculated for every day which is printed along with the total number of users per day.

1. **Pi Chart for Total number of Males and Female Citi bike users**



The Pie-Chart graph depicts the number of Citi Bike Users on the basis of Gender. From the graph, it is very clear that the males are more likely to use a Citi Bike than Female. New York City’s population is 53% female and 47% male. Even with more number of females in the NYC, the number of male users for Citi Bike are more.

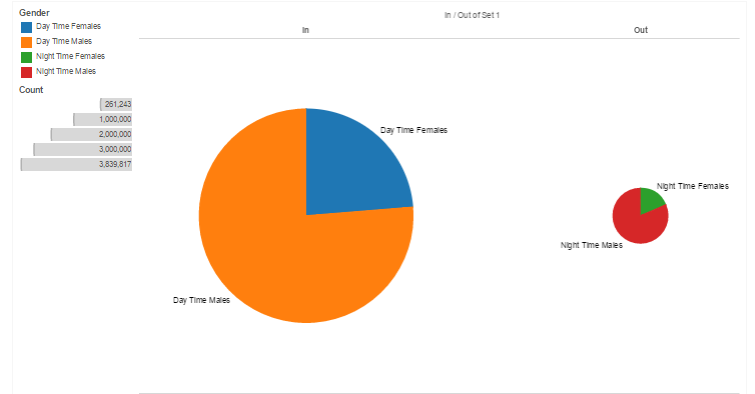
**MAPPER:**

The Mapper takes the Citi Bike data as input and extracts the gender field from every row. It prints the key as the gender and the value as count which is set to 1.

**REDUCER:**

The Reducer counts the number of Citi Bike Users on the basis of Gender. Hence, we get the number of males and females who ride Citi Bike.

1. **Citi Bike – Gender and Time**



The above Pie-Charts depicts the number of Citi Bike users based on the gender and the time period. The Night Time is taken from 10pm to 5am, whereas the Day time is taken from 5am till 10pm at night. The graph shows that Male Users are active both during the day time as well as the night time. Whereas, female users are more likely to be using Citi Bikes during Day Time rather than Night Time.

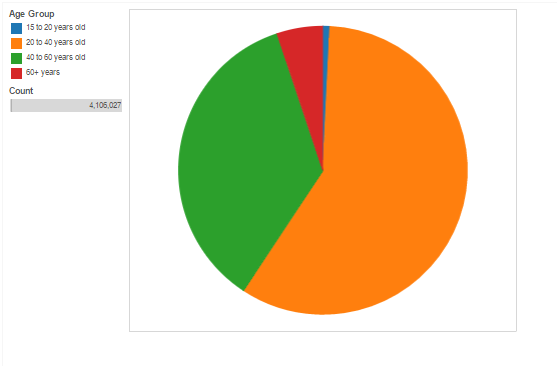
**MAPPER:**

The Mapper takes the Citi Bike Data as input and extracts the time and the gender from the data. For day time, it separates the users on the basis of gender. For example, if it is day time, and the gender is Male (i.e. 1), then the key generated is “Day Male”. The count is set to 1.

**REDUCER:**

The Reducer counts the number of users based on the gender and the time shift. There are 4 unique keys generated, namely “Day Males”, “Night Males”, “Day Females”, and “Night Females”. The reducer gives the number of required users for all the unique keys.

1. **Citi Bike users count on basis of Age**

****

The Pie-Chart Graph depicts the number of Citi bike users on the basis of age. From the above graph, we can observe that the people between the age group of 20-40 years are most athletic. Surprisingly, the next followed age group is 40-60 years old. New York is full of cyclists. Irrespective of the age, everyone loves to cycle around in New York.

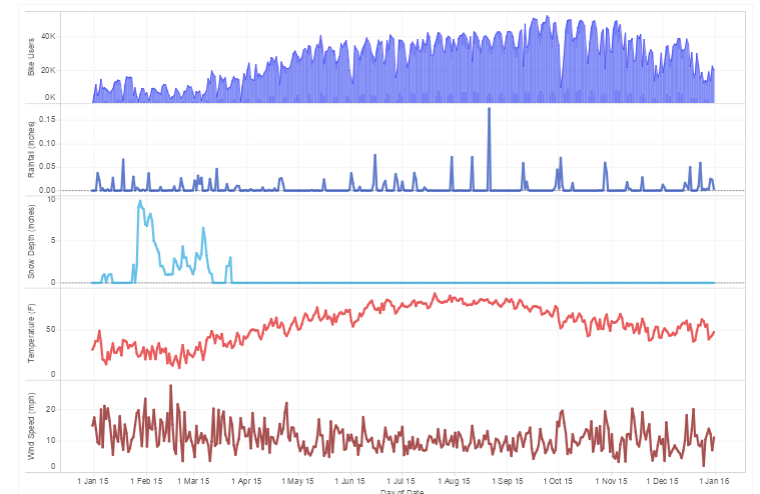
**MAPPER:**

The Mapper takes the Citi Bike Data as input and prints the age group of every user. The key is set to the Age Group for every age. For example, for age 22 years old, the key is “20 to 40 years old” and so on.

**REDUCER:**

The Reducer counts the number of Citi Bike Uses on the basis of different age group. Here, age groups act as unique keys. A counter keep on incrementing for every unique key. Hence, we get the total number of Citi Bike Users according to various age groups.

1. **Bike user Per day analysis**



The Area Graph depicts the number of Citi Bike Users per day. The Line Graph represents various weather attributes such as Temperature (in F), Rainfall (in inches), Snowfall Depth (in inches), and Wind Speed (in mph). From the graph, we can observe that the number of Citi Bike Users decrease tremendously whenever there was Snowfall. Also, the number of Citi Bike Users decreased with the decrease in the temperature (in F). Although, there was no such observation for Wind Speed and Rainfall. The number of Citi Bike Users were not affected by Rainfall and Wind Speed.

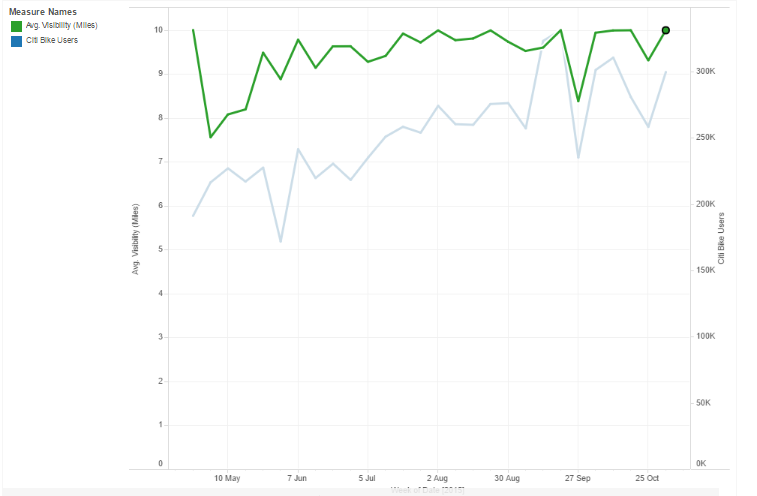
**MAPPER:**

The Mapper extracts the date (YYYYMMDD) from the primary key YYYYMMDDHH from the join output between weather and Citi bike table and prints the corresponding user along with temperature(F), Rainfall(inches), Snowfall(inches) and Wind Speed(Mph).

**REDUCER:**

The Reducer counts the number of users per day and prints the number of Citi Bike Users along with the temperature, wind speed, rainfall and snowfall for that particular day.

1. **Visibility vs Citi Bike:**



This graph depicts the effects of Visibility on the Citi Bike Users. As we can see that as the Visibility (in miles) increases the Citi Bike Users also increases. The range for the above graph is from May 1 to November 1, since the number of Citi Bike Users depends on various other weather factors during other months.

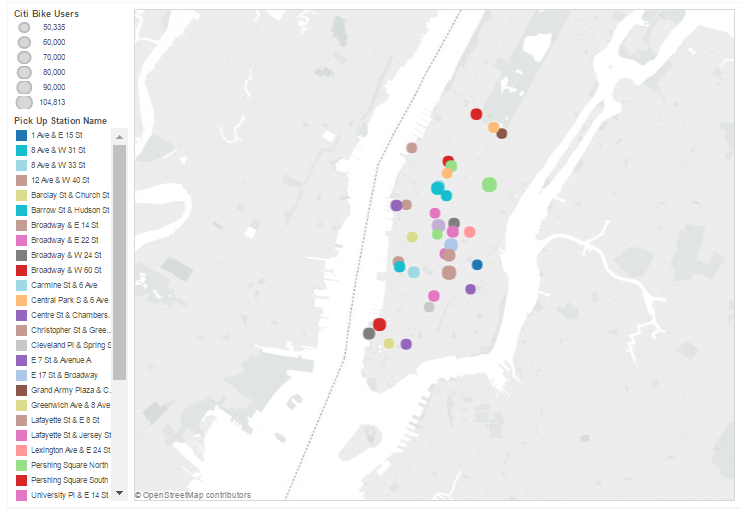
**MAPPER:**

The Mapper takes the output of the join between the Citi Bike Data and the Weather table as input and prints out the Citi bike user and corresponding visibility (in miles) for each day in 2015.

**REDUCER:**

The Reducer counts the number of Citi bike users per day along with the average visibility of that particular day.

1. **Busiest Pick up Locations in NYC**



The Map Graph depicts the top 20 busiest pick-up Citi Bike Stations in New York City. The graph shows that the widely used stations are in the Midtown Manhattan.

**MAPPER:**

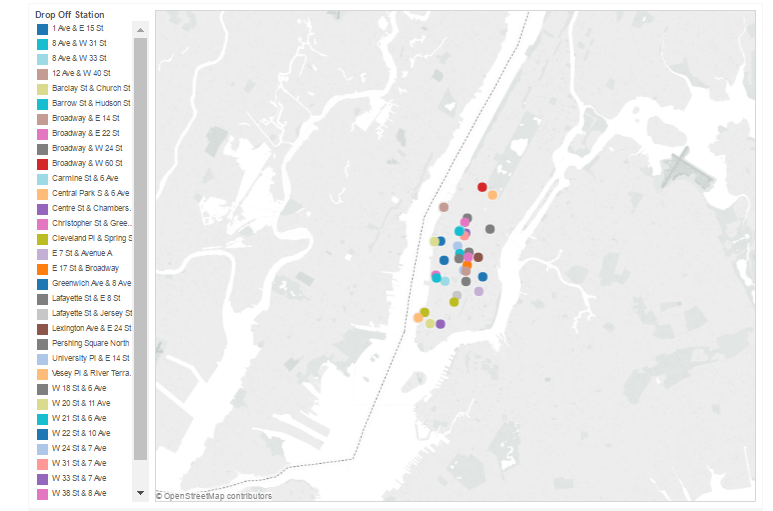
The Mapper takes the Citi Bike Data as input and extracts the Latitude and Longitude along with the Station Name from each and every row. The key is set to the Station Name and the count to 1.

**REDUCER:**

The Reducer counts the number of Citi Bike Users who pick-up their bikes at the stations. The Reucer then prints the names of the Stations with most number of pick-ups along with the latitude and longitude.

With the output, we are able to plot each station on the Manhattan Map.

1. **Busiest drop-off stations in NYC**



The Map Graph depicts the top 20 busiest drop-off Citi Bike Stations in New York City. The graph shows that the widely used stations are in the Midtown Manhattan.

**MAPPER:**

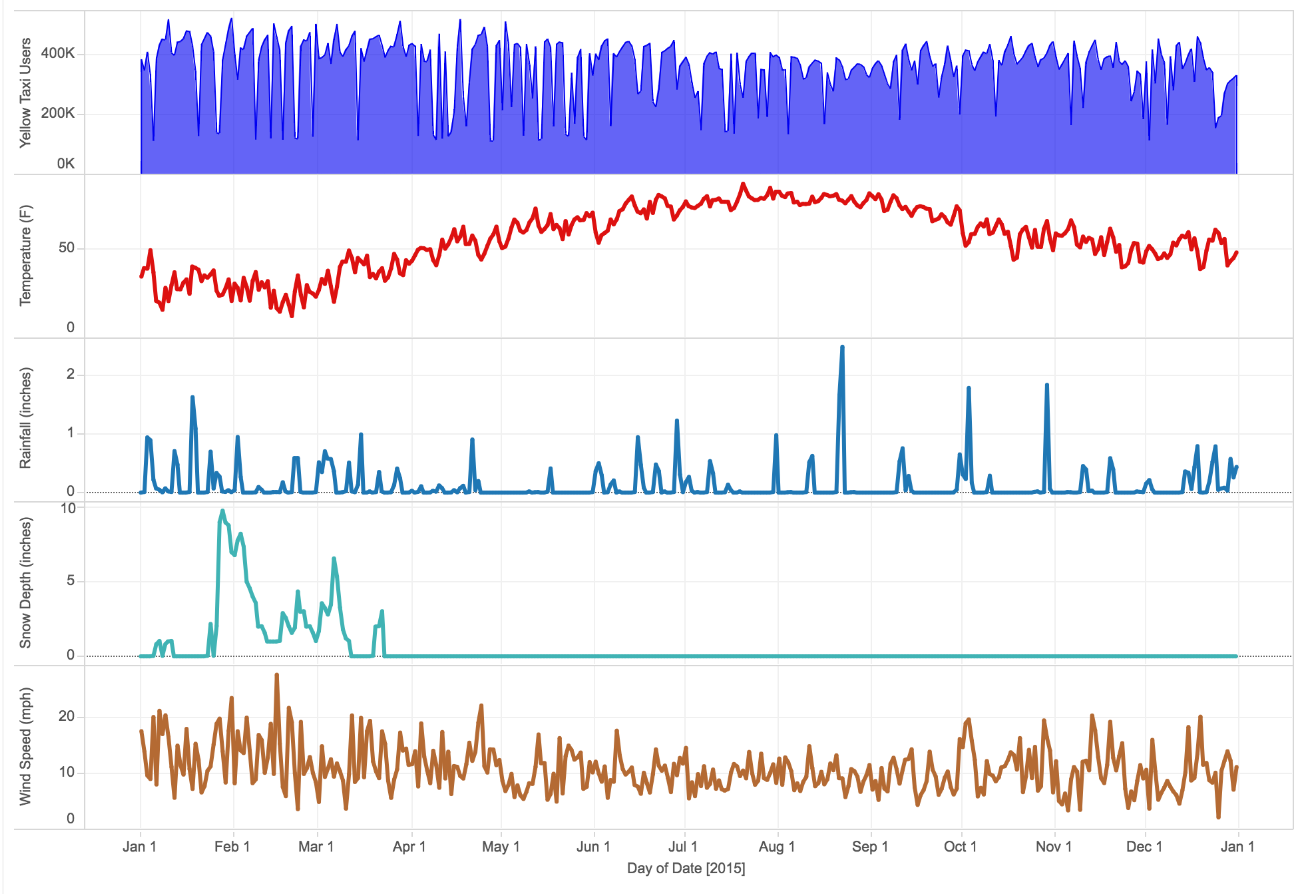
The Mapper takes the Citi Bike Data as input and extracts the Latitude and Longitude along with the Station Name from each and every row. The key is set to the Station Name and the count to 1.

**REDUCER:**

The Reducer counts the number of Citi Bike Users who drop-off their bikes at the stations. The Reucer then prints the names of the Stations with most number of drop-offs along with the latitude and longitude.

With the output, we are able to plot each station on the Manhattan Map.

1. **Yellow Taxi Users Analysis on the basis of Temperature, Rainfall and Snowfall**

****

The Area Graph shows the number of Yellow Taxi Users per day, whereas the Line Graphs shows the Temperature (F), Rainfall (inches), Wind Speed(mph) and Snowfall Depth (inches) for every day in year 2015. The graph shows the relation between number of Yellow Taxi Users and the weather attributes.

We can notice from the graph that Wind Speed and Temperature does not affect the number of Yellow Taxi Users that much, but the amount of rainfall and snowfall does affect. With the increase in Snowfall Depth, the number of Yellow Taxi Users increases.

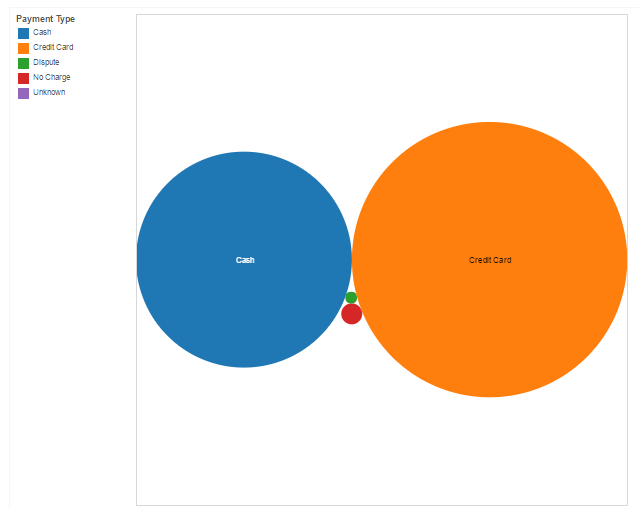
**MAPPER:**

The Mapper takes the joined data of Yellow Taxi and Weather data as input and sets the prints the primary key (YYYYMMDDHH), temperature, rainfall, wind speed, and snowfall for every row.

**REDUCER:**

The Reducer counts the number of users per day and prints the output as number of users per day, along with the average values of Temperature, Rainfall, Wind Speed, and Snowfall.

## Payment Mode for Yellow Taxi Users Analysis



The Pie Chart graph depicts the type of payment mode for Yellow Taxi users. The graph clearly shows that maximum users prefer Credit Card as a mode of payment. New Yorkers don’t generally carry too much of cash with them. Although, the next preferred mode pf payment is Cash.

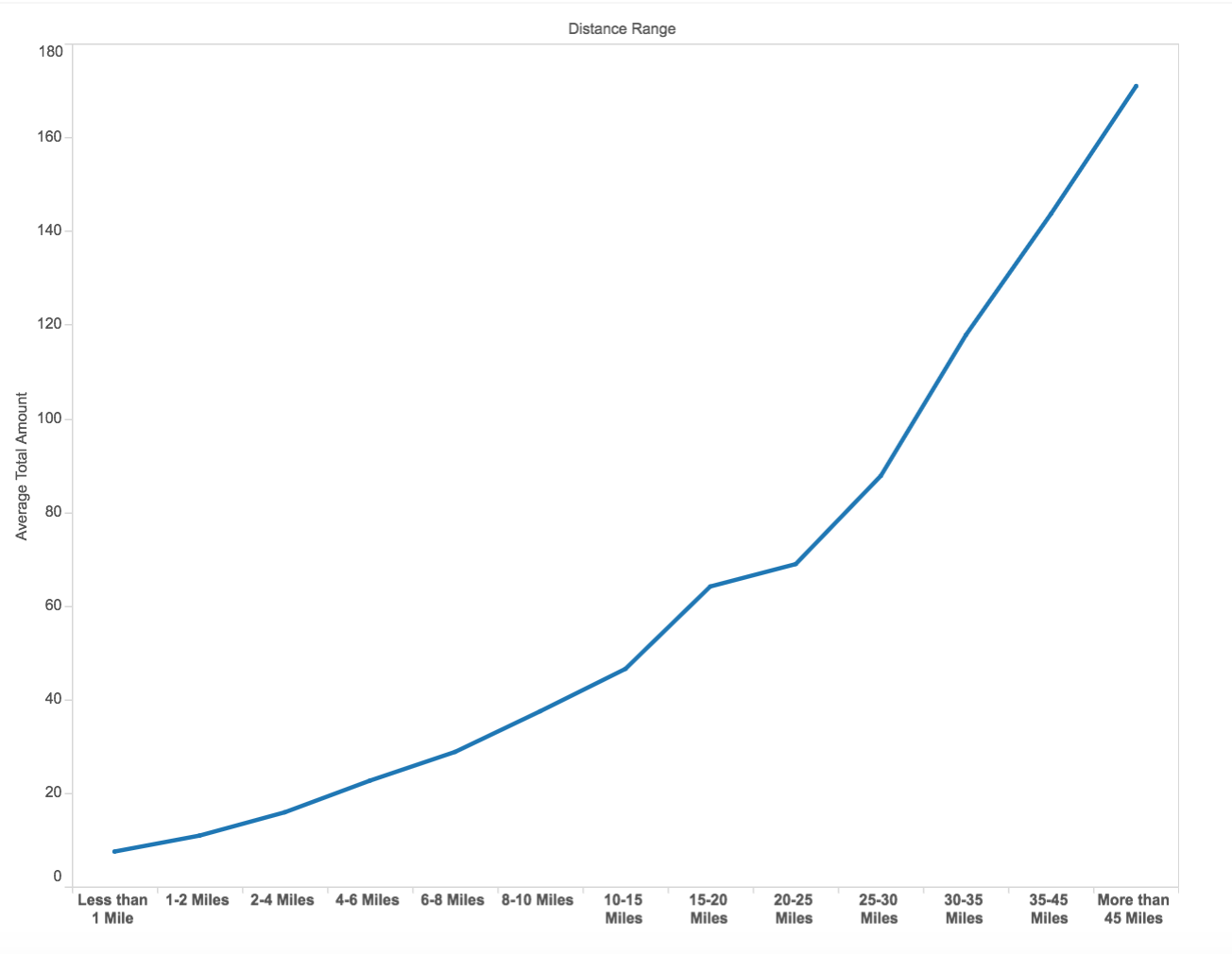
**MAPPER:**

The Mapper takes the Yellow Taxi Data as input and sets the key as the mode of payment. For example, if the mode of payment is 1, then key is set to “Credit Card”. The count is set to 1.

**REDUCER:**

The Reducer counts the number of unique keys and prints them. Hence, we get the output as the counts for each and every mode of payment.

1. **Distance Vs Average Total Fare Amount for Yellow Taxi**

****

The Line Graph depicts the Average Total Fare and the distance covered in Yellow Taxi. The X-axis shows the distance range, and Y-axis shows the Average Total Fare Amount in Dollars.

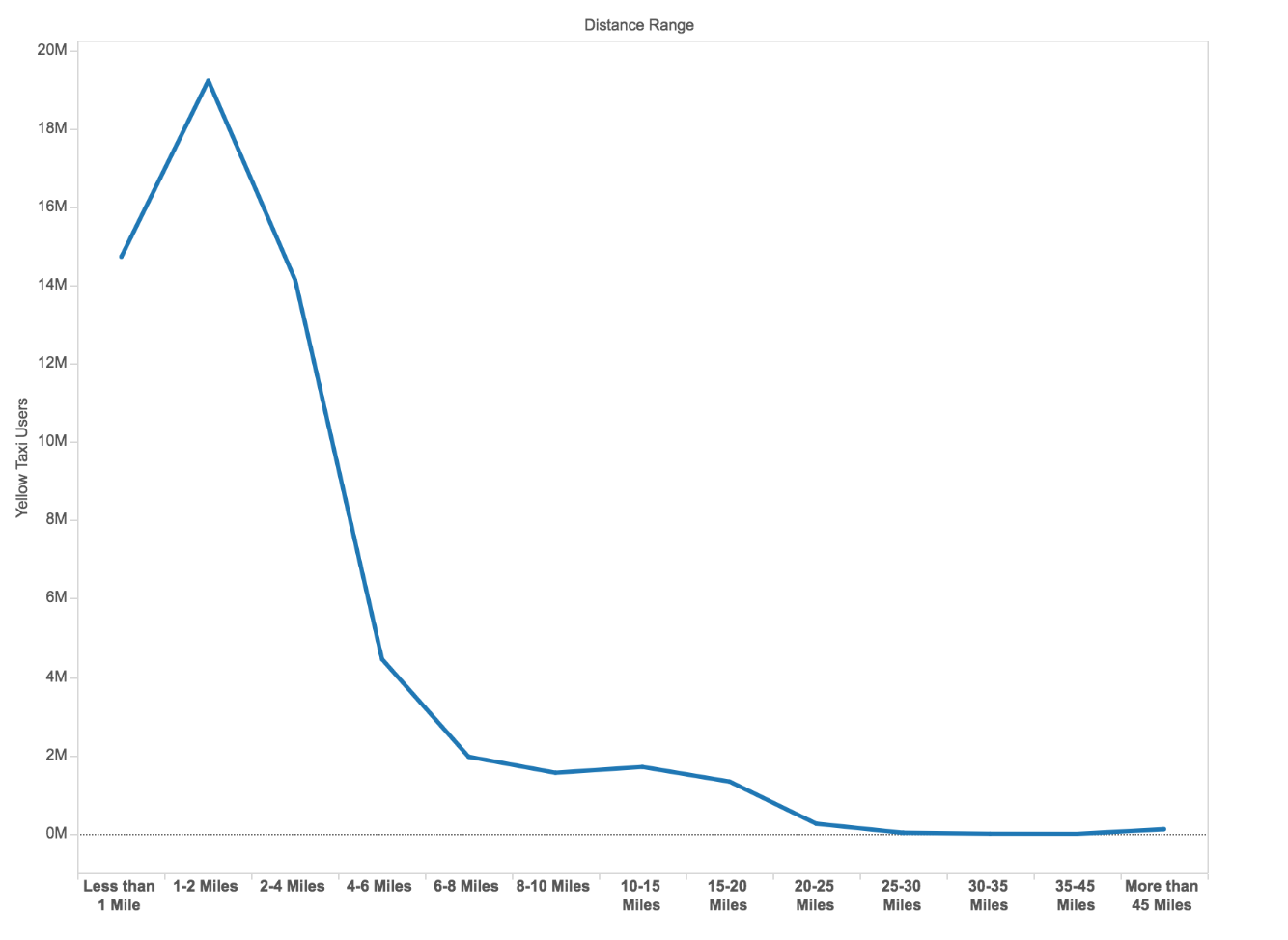
**MAPPER:**

The Mapper takes the Yellow Taxi Data as Input in which the key is set to distance range. For Example, for distance less than 1 mile, key is “Less than 1 Mile” and so on. The value is set to the total amount of the particular row. The Count is set to 1 for every key-value pair.

**REDUCER:**

The Reducer counts the number of unique keys and keeps on adding the value (i.e. total fare amount). In the end, it calculates the average of Total Fare Amount. The total number of unique keys are 13, namely Less than 1 Mile, 1-2 Miles, and so on.

1. **Distance Vs Total Yellow Taxi Users**

****

The Line Graph shows the Total Yellow Taxi users for various distance range. The graph shows that maximum Yellow Taxi Users travel in the Yellow Taxi for short distances only. In the graph, we can observe that between 1-2 Miles, maximum number of Yellow Taxi are hired.

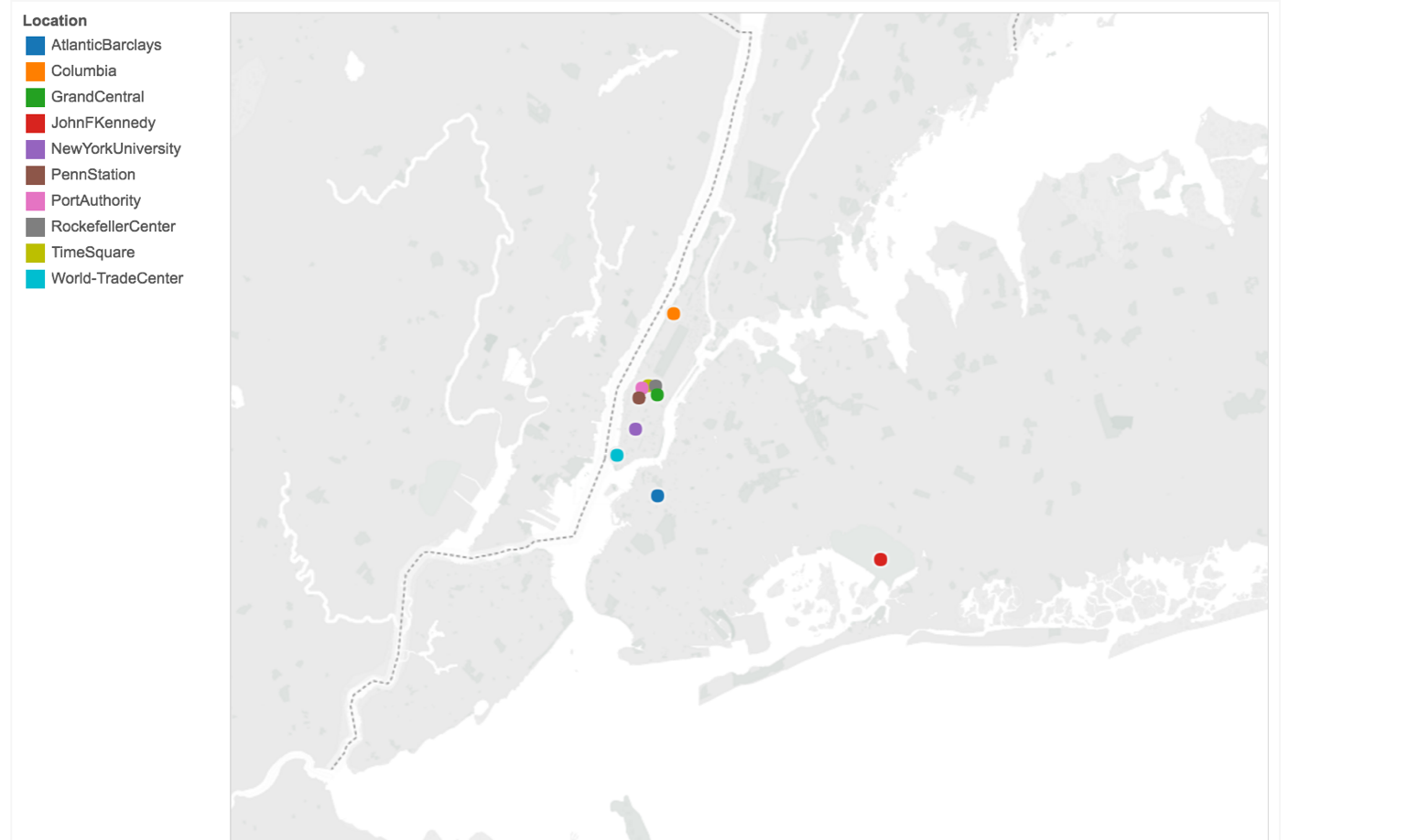
**MAPPER:**

The Mapper takes the Yellow Taxi Data as Input in which the key is set to distance range. For Example, for distance less than 1 mile, key is “Less than 1 Mile” and so on. The value is 1.

**REDUCER:**

The Reducer counts the number of unique keys and thus prints the number of Yellow Taxi Users for every distance range.

1. **Yellow Taxi with most number of Pick-Up Locations**



The Map Graph various locations where there were maximum number of Pick-Up Locations. Some of these locations include JFK Airport, Time Square, Port Authority etc.

**MAPPER:**

The Mapper takes the Yellow Taxi Data as input and extracts the latitude and longitude of all the entries. The Spatial index is used to check whether a particular point lies within the polygon.

**REDUCER:**

The Reducer takes in the input from Mapper and counts the number of pick-ups from all the latitude and longitude.

# Scripts/Code:

Github: https://github.com/GoAsians/GoAsiansBigData.git

# Steps on how to run the scripts:

## On Amazon AWS.

* Login on aws.amazon.com
* Open S3 - Create Cluster - Create Bucket
* Upload all the files- MapReduce, Input Data (Taxi Data, Citi Bike)
* Go to EMR
* Create Cluster - select the EC2 key-pair - click Create
* Now go to Steps - Add Steps
* In step type - Choose Streaming Program
* In name - Output file
* In mapper - Mapper input file
* In Reducer - Reducer Input file
* Input Location - folder
* Make sure that the Taxi/Weather are in the same folder inside the s3 bucket.
* Browse to your project Directory.
* Output - <Bucket name> / output.
* Leave the argument box empty.
* Click ADD
* After the successful operation of the map/reduce task
* Keep refreshing until the status changes from running to completed.
* Once the status turns to completely go-to the output folder to check.
* You’ll find list of files that is the output of the Map/Reduce Progress.

### **Steps for Local**

On MAC terminal:

* Open Terminal and browse to the folder containing the Map/Reduce Program and type the following command -
* chmod +x <name of mapperfile.py>
* chmod +x <name of Reducerfile.py>
* Cat <name of i/p file> | <Mapperfile> | sort | < name of reduce file>

### **Steps for visualization of graph**

* Import the output file in the Tableau software and drag & drop the required fields to Rows and columns click on the type of graph you want to choose. Example: Bar graph, Pie Chart, Histogram, etc.

# Experimental Setup

## Cluster Configuration:

Master: 1

Slave Nodes: 3

**Number of Mappers/Reducers**: 17

### **Tools Used**:

Amazon AWS, Python, Tableau, Github, Javascripts, HTML, CSS, Material Design Lite, Git.

Github: https://github.com/GoAsians/GoAsiansBigData.git

# Optimizations applied to the script:

Program was taking a lot of time so we changed the one stage algorithm to the two stage algorithm.

# Individual contributions:

Vaibhav was responsible for writing MapReduce programs and Visualizations.

Jitao was responsible for PySpark and MapReduce programs.

Palak was responsible for Web page development and Documentation and Visualization.