DATA Challenge

Bhanuja Nagore

ANALYSIS REPORT

Contents

[NYC Green Taxi 2](#_Toc475385987)

[Overview: 2](#_Toc475385988)

[Tools Used: 2](#_Toc475385989)

[Solutions: 2](#_Toc475385990)

[Question 1 2](#_Toc475385991)

[1.1 Programmatically download and load into your favorite analytical tool the trip data for September 2015. 2](#_Toc475385992)

[1.2 Report how many rows and columns of data you have loaded 3](#_Toc475385993)

[Question 2 5](#_Toc475385994)

[2.1 Plot a histogram of the number of the trip distance ("Trip Distance") 5](#_Toc475385995)

[2.2 Report any structure you find and any hypotheses you have about that structure 7](#_Toc475385996)

[Data Cleaning: 9](#_Toc475385997)

[Question 3 15](#_Toc475385998)

[Report mean and median trip distance grouped by hour of day 15](#_Toc475385999)

[We'd like to get a rough sense of identifying trips that originate or terminate at one of the NYC area airports. Can you provide a count of how many transactions fit this criteria, the average fair, and any other interesting characteristics of these trips? 17](#_Toc475386000)

[Question 4 17](#_Toc475386001)

[4.1 Build a derived variable for tip as a percentage of the total fare. 17](#_Toc475386002)

[Build a predictive model for tip as a percentage of the total fare. Use as much of the data as you like (or all of it). We will validate a sample. 18](#_Toc475386003)

[REGRESSION: 18](#_Toc475386004)

[DECISION TREE (Prediction Tree) 20](#_Toc475386005)

[Appendix A 26](#_Toc475386006)

[Codes: 26](#_Toc475386007)

[C1 - Main Source Code: 26](#_Toc475386008)

[C2 – Source Code: To predict model for Tip percentage Q4 part 2: 34](#_Toc475386009)

[C3 – Source Code: To create Histograms and bar plot: 38](#_Toc475386010)

[C4 – Source Code: Used for clustering in Question 2 To define Structure to the data 38](#_Toc475386011)

[C5 – Source Code: Used for Question 3.2 To calculate various values like total fare, hours, duration for airport and all trips 40](#_Toc475386012)

[C6 - Main Source code for Visualization – To generate excel that is used in Tableau to create graphs/plots/charts 42](#_Toc475386013)

[Appendix B 44](#_Toc475386014)

[Additional Figures 44](#_Toc475386015)

# NYC Green Taxi

## Overview:

New York City's Taxi and Limousine Commission

The City of New York created the Taxi and Limousine Commission in 1971 to oversee the

operations of a number of public transportation services in New York City. With over 74,000

vehicles and over 130,000 drivers who operate said vehicles (New York Taxi and Limousine

Commission, 2014), the importance of the TLC in New York cannot be understated.

The transportation services the TLC provides are numerous. Medallion taxicabs consist

of 13,635 vehicles in the TLCs vehicle pool (New York Taxi and Limousine Commission,

2014).

TLC also created a new license for Boro Taxis, which provide service for the boroughs

of New York and are known for their green color.

Furthermore, the TLC has also made a push to license commuter vans that

can be pre-arranged to move larger groups of passengers. In total, these 78,000 vehicles

provide transport for over 1.5 million passengers every day (New York Taxi and Limousine

Commission, 2014).

As private companies have exploded into the taxi industry, the TLC has also reacted

to this sudden new pressure. Following the rise of companies such as Uber, the TLC

implemented its own E-Hail program that allowed riders to hail their taxi through an app

on their mobile phone (New York Taxi and Limousine Commission, 2014).

Tools Used:

I have used the following tools for the NYC Green Taxi dataset:

1. R (R-studio) – read/retrieve the data, data cleaning, data munging, analysis and modelling
2. Microsoft excel – To export data into Tableau
3. Tableau – Data Visualization

***Quick Facts:***

I used data as table in R. Directly downloaded the csv file from the given url

<'https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv'>

## Solutions:

### Question 1

### 1.1 Programmatically download and load into your favorite analytical tool the trip data for September 2015.

#### Approach:

I used data as table in R. Directly downloaded the csv file from the given url

<'https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv'>

#### Code:

# read the NYC Green Taxi Data for the month of Sept’15 from the given path ()

library(data.table)

mydat <- fread('https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv')

head(mydat)

str(mydat)

write.csv(mydat, file = "greenTaxi.csv", row.names = FALSE)

summary(mydat)

is.data.frame(mydat)

### 1.2 Report how many rows and columns of data you have loaded

#### Rows: 1494926

#### Columns: 21

|  |
| --- |
| Read 1494926 rows and 21 (of 21) columns from 0.223 GB file in 00:00:14  Classes ‘data.table’ and 'data.frame': 1494926 obs. of 21 variables:  $ VendorID : int 2 2 2 2 2 2 2 2 2 2 ...  $ lpep\_pickup\_datetime : chr "2015-09-01 00:02:34" "2015-09-01 00:04:20" "2015-09-01 00:01:50" "2015-09-01 00:02:36" ...  $ Lpep\_dropoff\_datetime: chr "2015-09-01 00:02:38" "2015-09-01 00:04:24" "2015-09-01 00:04:24" "2015-09-01 00:06:42" ...  $ Store\_and\_fwd\_flag : chr "N" "N" "N" "N" ...  $ RateCodeID : int 5 5 1 1 1 1 1 1 1 1 ...  $ Pickup\_longitude : num -74 -74 -73.9 -73.9 -74 ...  $ Pickup\_latitude : num 40.7 40.9 40.8 40.8 40.7 ...  $ Dropoff\_longitude : num -74 -74 -73.9 -73.9 -73.9 ...  $ Dropoff\_latitude : num 40.7 40.9 40.8 40.8 40.7 ...  $ Passenger\_count : int 1 1 1 1 1 1 1 1 1 1 ...  $ Trip\_distance : num 0 0 0.59 0.74 0.61 1.07 1.43 0.9 1.33 0.84 ...  $ Fare\_amount : num 7.8 45 4 5 5 5.5 6.5 5 6 5.5 ...  $ Extra : num 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...  $ MTA\_tax : num 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...  $ Tip\_amount : num 1.95 0 0.5 0 0 1.36 0 0 1.46 0 ...  $ Tolls\_amount : num 0 0 0 0 0 0 0 0 0 0 ...  $ Ehail\_fee : logi NA NA NA NA NA NA ...  $ improvement\_surcharge: num 0 0 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 ...  $ Total\_amount : num 9.75 45 5.8 6.3 6.3 8.16 7.8 6.3 8.76 6.8 ...  $ Payment\_type : int 1 1 1 2 2 1 1 2 1 2 ...  $ Trip\_type : int 2 2 1 1 1 1 1 1 1 1 ...  - attr(\*, ".internal.selfref")=<externalptr> |

> summary(mydat)

VendorID lpep\_pickup\_datetime Lpep\_dropoff\_datetime Store\_and\_fwd\_flag

Min. :1.000 Length:1494926

Length:1494926

1st Qu.:2.000 Class :character Class :character Class :character

Median :2.000 Mode :character Mode :character Mode :character

Mean :1.782

3rd Qu.:2.000

Max. :2.000

RateCodeID Pickup\_longitude Pickup\_latitude Dropoff\_longitude

Min. : 1.000 Min. :-83.32 Min. : 0.00 Min. :-83.43

1st Qu.: 1.000 1st Qu.:-73.96 1st Qu.:40.70 1st Qu.:-73.97

Median : 1.000 Median :-73.95 Median :40.75 Median :-73.95

Mean : 1.098 Mean :-73.83 Mean :40.69 Mean :-73.84

3rd Qu.: 1.000 3rd Qu.:-73.92 3rd Qu.:40.80 3rd Qu.:-73.91

Max. :99.000 Max. : 0.00 Max. :43.18 Max. : 0.00

Dropoff\_latitude Passenger\_count Trip\_distance Fare\_amount

Min. : 0.00 Min. :0.000 Min. : 0.000 Min. :-475.00

1st Qu.:40.70 1st Qu.:1.000 1st Qu.: 1.100 1st Qu.: 6.50

Median :40.75 Median :1.000 Median : 1.980 Median : 9.50

Mean :40.69 Mean :1.371 Mean : 2.968 Mean : 12.54

3rd Qu.:40.79 3rd Qu.:1.000 3rd Qu.: 3.740 3rd Qu.: 15.50

Max. :42.80 Max. :9.000 Max. :603.100 Max. : 580.50

Extra MTA\_tax Tip\_amount Tolls\_amount

Min. :-1.0000 Min. :-0.5000 Min. :-50.000 Min. :-15.2900

1st Qu.: 0.0000 1st Qu.: 0.5000 1st Qu.: 0.000 1st Qu.: 0.0000

Median : 0.5000 Median : 0.5000 Median : 0.000 Median : 0.0000

Mean : 0.3513 Mean : 0.4866 Mean : 1.236 Mean : 0.1231

3rd Qu.: 0.5000 3rd Qu.: 0.5000 3rd Qu.: 2.000 3rd Qu.: 0.0000

Max. :12.0000 Max. : 0.5000 Max. :300.000 Max. : 95.7500

Ehail\_fee improvement\_surcharge Total\_amount Payment\_type

Mode:logical Min. :-0.3000 Min. :-475.00 Min. :1.000

NA's:1494926 1st Qu.: 0.3000 1st Qu.: 8.16 1st Qu.:1.000

Median : 0.3000 Median : 11.76 Median :2.000

Mean : 0.2921 Mean : 15.03 Mean :1.541

3rd Qu.: 0.3000 3rd Qu.: 18.30 3rd Qu.:2.000

Max. : 0.3000 Max. : 581.30 Max. :5.000

Trip\_type

Min. :1.000

1st Qu.:1.000

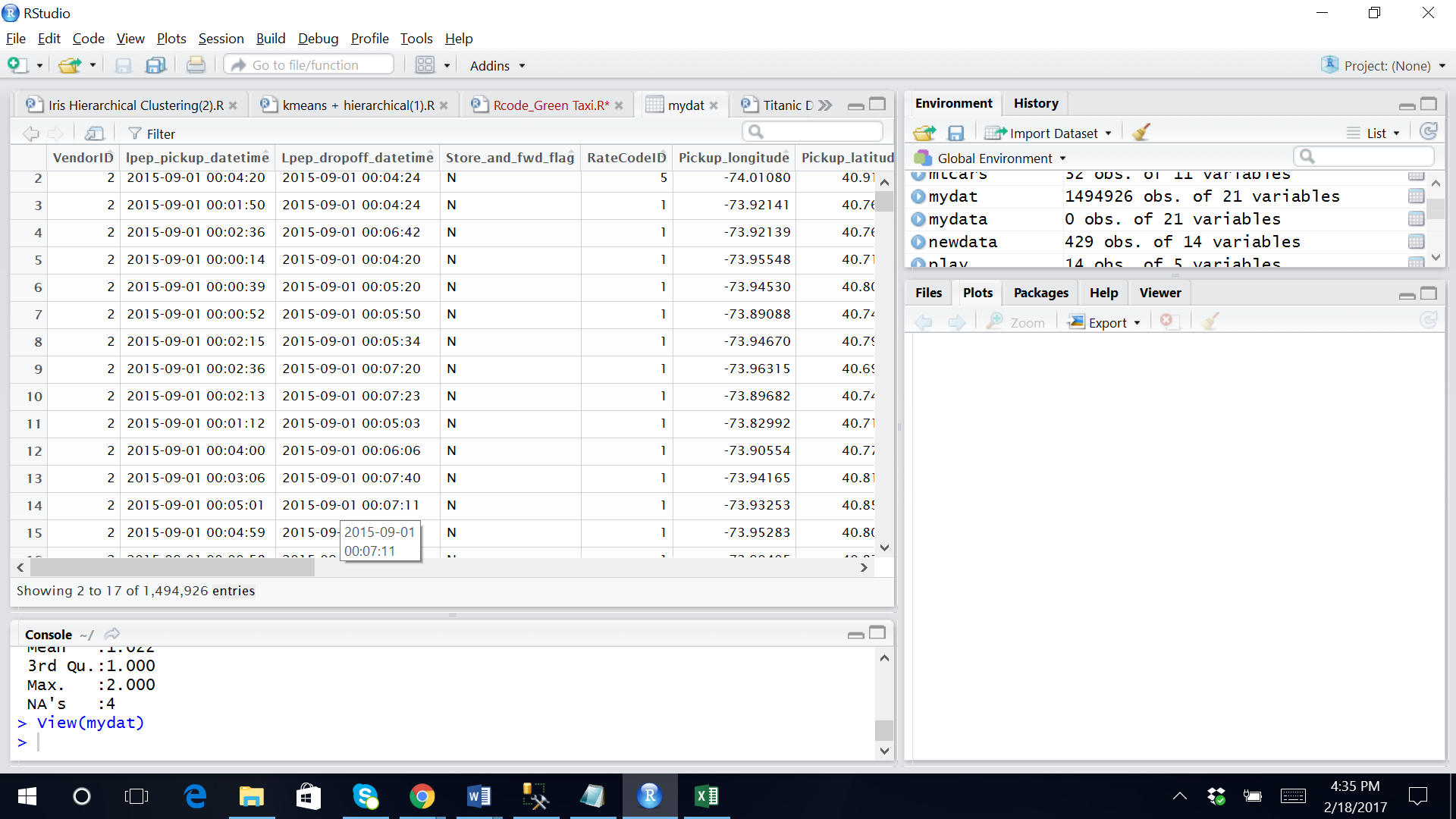
Median :1.000

Mean :1.022

3rd Qu.:1.000

Max. :2.000

NA's :4



### Question 2

### 2.1 Plot a histogram of the number of the trip distance ("Trip Distance")

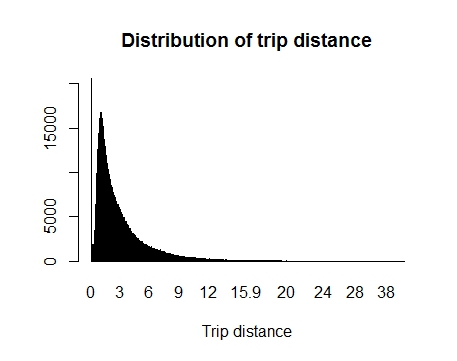
#### Solution:

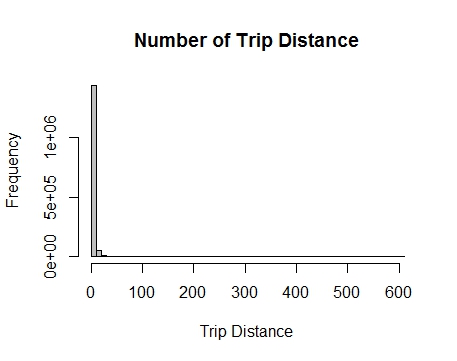
#### Plotted the Histogram in R

hist(mydat$Trip\_distance, breaks = 50, xlab = "Trip Distance", col = "gray", main = "Number of Trip Distance" )

hist(mydat$Trip\_distance, main = "Number of Trip Distance", xlab = "Trip Distance",

col="darkgreen")

****

****

Source code:

set1 = cleanData[cleanData$Tip\_amount>0]

set2 = cleanData[cleanData$Tip\_amount==0]

#Total passenger paying Tip

nrow(set1)

#Total passenger not paying Tip

nrow(set2)

hist(cleanData$Tip\_amount, main="All trips", xlab = "Tip Amount", col="gray")

hist(set1$Tip\_amount, main="All trips with tips", xlab = "Tip Amount", col="gray")

require(graphics)

cleanData$Tip\_Percentage[cleanData$Tip\_Percentage=="NA"]<-0

offer<-sample(c(Payment\_type),size = 500,replace = T)

amountPur<-sample(c(Total\_amount),size = 500,replace = T)

offertest<-data.frame(offer=as.factor(offer))

model<-aov(amountPur ~ offer, data = offertest)

summary(model)

require(graphics)

offer<-sample(c(Payment\_type,Trip\_distance),size = 500,replace = T)

amountPur<-sample(c(Total\_amount),size = 500,replace = T)

offertest<-data.frame(offer=as.factor(offer))

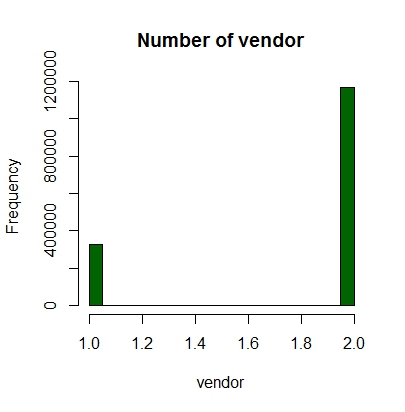
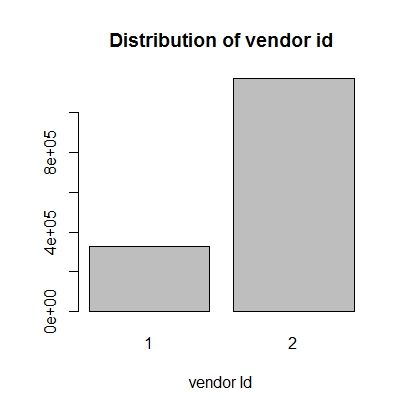
model<-aov(amountPur ~ offer, data = offertest)

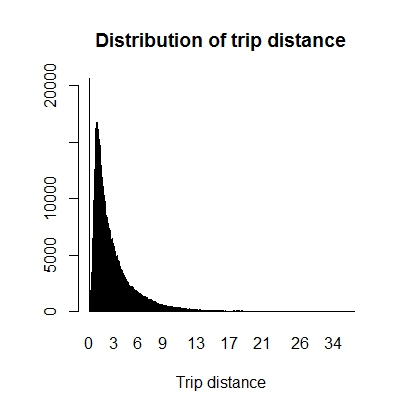
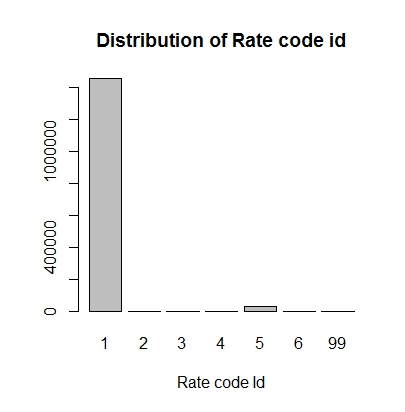
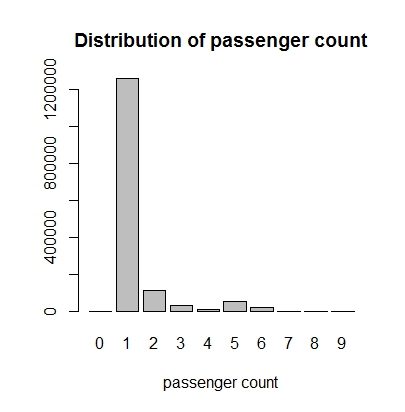
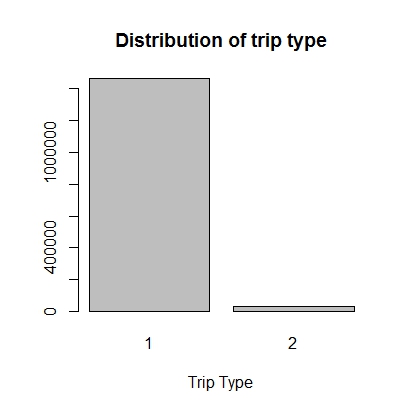
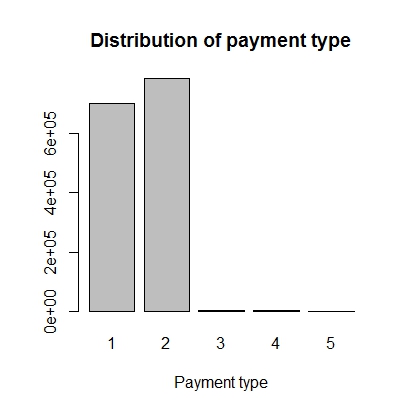
summary(model)

### 2.2 Report any structure you find and any hypotheses you have about that structure

#### Solution:

**Plots to find the Structure, Trend and pattern of the data**

** **

****

#### Inference:

* The record provided are maximum from VeriFone Inc. Vendor.
* The payment type is cash and credit hence cash dominates the payment type. Therefore, many records are not captured for Tip Amount as most of the times the payment is made by cash.
* Mostly 1-2 or 5 passengers travel with standard rate. And the trip distance is within 3miles most of the times.

**Before Figuring out the structure, started with cleaning out the data. Few steps are:**

## Data Cleaning:

1. **As the basic fare is 2.5$, so the data whose Total amount was less than 2.5$, changed to 2.5$**
2. **For all the fare amount less than 0, changed it to 0**
3. **All the improvement\_surcharge less than 0, changed it to 0**
4. **All the Tip amount less than 0, changed it to 0**
5. **All the Tip type == NaN, NA, changed it to 1 (Most frequent)**
6. **For all the rate code ID ==99, changed to 2 (Most frequent)**
7. **Deleted Ehail fee as all the value was NULL**
8. **All the Extra less than 0, changed it to 0**
9. **All the duration ==NA, changed it to 0**
10. **All the speed equal to NaN,NA,Inf, changed it to 0**

#### Clean Data Summary:

Classes ‘data.table’ and 'data.frame': 1494926 obs. of 28 variables:

$ VendorID : int 2 2 2 2 2 2 2 2 2 2 ...

$ lpep\_pickup\_datetime : POSIXct, format: "2015-09-01 00:02:34" "2015-09-01 00:04:20" ...

$ Lpep\_dropoff\_datetime: POSIXct, format: "2015-09-01 00:02:38" "2015-09-01 00:04:24" ...

$ Store\_and\_fwd\_flag : chr "N" "N" "N" "N" ...

$ RateCodeID : num 5 5 1 1 1 1 1 1 1 1 ...

$ Pickup\_longitude : num -74 -74 -73.9 -73.9 -74 ...

$ Pickup\_latitude : num 40.7 40.9 40.8 40.8 40.7 ...

$ Dropoff\_longitude : num -74 -74 -73.9 -73.9 -73.9 ...

$ Dropoff\_latitude : num 40.7 40.9 40.8 40.8 40.7 ...

$ Passenger\_count : int 1 1 1 1 1 1 1 1 1 1 ...

$ Trip\_distance : num 0 0 0.59 0.74 0.61 1.07 1.43 0.9 1.33 0.84 ...

$ Fare\_amount : num 7.8 45 4 5 5 5.5 6.5 5 6 5.5 ...

$ Extra : num 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

$ MTA\_tax : num 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

$ Tip\_amount : num 1.95 0 0.5 0 0 1.36 0 0 1.46 0 ...

$ Tolls\_amount : num 0 0 0 0 0 0 0 0 0 0 ...

$ improvement\_surcharge: num 0 0 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 ...

$ Total\_amount : num 9.75 45 5.8 6.3 6.3 8.16 7.8 6.3 8.76 6.8 ...

$ Payment\_type : int 1 1 1 2 2 1 1 2 1 2 ...

$ Trip\_type : num 2 2 1 1 1 1 1 1 1 1 ...

$ Tip\_Percentage : num 20 0 8.62 0 0 ...

$ month : int 9 9 9 9 9 9 9 9 9 9 ...

$ wday : int 3 3 3 3 3 3 3 3 3 3 ...

$ hour : atomic 0 0 0 0 0 0 0 0 0 0 ...

..- attr(\*, "levels")= chr "0-6am" "0-6am" "0-6am" "0-6am" ...

$ DropOffhour : int 0 0 0 0 0 0 0 0 0 0 ...

$ duration : num 0 0 2 4 4 4 4 3 4 5 ...

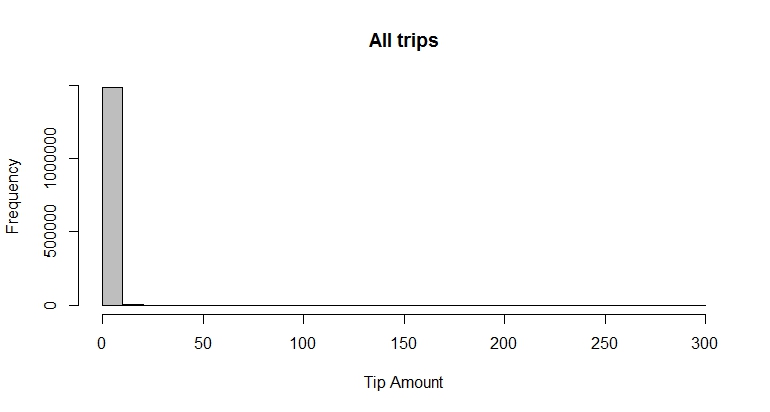
$ speed : num 0 0 17.7 11.1 9.15 ...

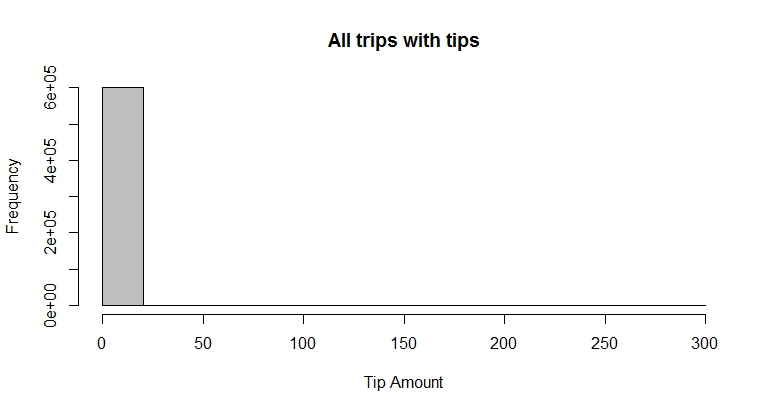
$ BySpeed : logi TRUE TRUE TRUE TRUE TRUE TRUE ...

- attr(\*, ".internal.selfref")=<externalptr>

#### Approach:

* Firstly, after analyzing the dirty data, plotted various graphs to read the large chunk of data. This comes under the discovery phase so as to plan for the model.
* To develop initial Hypothesis, Visualization is useful for data exploration and presentation but stats is crucial because may exist throughout the entire data analytics Lifecycle.
* To apply the regression model, it is important to test the hypothesis as The variable will not affect the outcome because its coefficient is zero for null hypothesis to be true.
* I started with ANOVA as there are more than two populations. The impact of various factors like payment type and Total amount that affect/not affect the tip amount. The goal is to determine which strategy is more effective.





> #Total passenger paying Tip

> nrow(set1)

[1] 602732

> #Total passenger not paying Tip

> nrow(set2)

[1] 892194

>

|  |  |  |
| --- | --- | --- |
| |  | | --- | | require(graphics)  > cleanData$Tip\_Percentage[cleanData$Tip\_Percentage=="NA"]<-0  > offer<-sample(c(Payment\_type),size = 500,replace = T)  > amountPur<-sample(c(Total\_amount),size = 500,replace = T)  > offertest<-data.frame(offer=as.factor(offer))  > model<-aov(amountPur ~ offer, data = offertest)  > summary(model)  Df Sum Sq Mean Sq F value Pr(>F)  offer 3 678356 226119 0.409 0.747  Residuals 496 274450890 553328  >  > require(graphics)  > offer<-sample(c(Payment\_type,Trip\_distance),size = 500,replace = T)  > amountPur<-sample(c(Total\_amount),size = 500,replace = T)  > offertest<-data.frame(offer=as.factor(offer))  > model<-aov(amountPur ~ offer, data = offertest)  > summary(model)  Df Sum Sq Mean Sq F value Pr(>F)  offer 179 86105684 481037 0.762 0.978  Residuals 320 201935944 631050 | |  |  Analysis: ONE WAY ANOVA  * The sample size is 500 * H0 – Null Hypothesis: The means are equal of all the populations * H1 – Alternative Hypothesis: The means are not equal * The first test for offer degree of freedom is 3, the degree of freedom for residual is 496. * The F test (0.409) and (0.762) are less than 1 and p value near 1. * Thus, null hypothesis that the means are equal should be accepted. * Hence p value is close to 1, suggest that payment type and trip distance are significantly different from each other * in the group of transactions with tips compared to the group with no tip. Therefore, this variable * would use to train the classification model. * Only offer was executed because the influence of one factor(offers), its one-way ANOVA. * The goal is to analyze one factor such as payment type hence One-way ANOVA. |
|  |

#### Second Structure applied: Clustering

* To group the similar objects and to discover hidden structure, using R to perform K means analysis.
* The task is to group all the passengers based on these areas "Payment type", "speed", "Trip type", "Total amount", "duration".
* In the graph the large circle represents location of the cluster means. The small dot represents the passengers to the corresponding clusters by assigned color: red, blue, green.

**#Clustering based on Payment type and Rate code ID or Trip Type**

> kmdata\_orig[1:10,]

Payment\_type speed Trip\_type Total\_amount duration

[1,] 1 15.050000 1 15.80 12

[2,] 1 14.287500 1 18.80 16

[3,] 2 13.000000 1 8.30 6

[4,] 1 13.162500 1 14.30 16

[5,] 2 8.666667 1 8.80 9

[6,] 1 7.600000 1 15.95 15

[7,] 2 7.800000 1 6.80 6

[8,] 1 8.068966 1 24.95 29

[9,] 2 10.320000 1 9.80 10

[10,] 2 18.000000 1 25.30 24

> wss<-numeric(15)

> for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

> km = kmeans(kmdata\_orig, 3, nstart = 25)

> km

K-means clustering with 3 clusters of sizes 1, 16, 12

Cluster means:

Payment\_type speed Trip\_type Total\_amount duration

1 1.000 8.719149 1.000000 47.81000 47.00000

2 1.625 11.593933 1.000000 8.50625 6.68750

3 1.250 14.140681 1.083333 18.55917 17.41667

Clustering vector:

[1] 3 3 2 3 2 3 2 3 2 3 2 2 2 3 3 2 2 2 2 3 2 3 2 2 2 1 2 3 3

Within cluster sum of squares by cluster:

[1] 0.0000 367.7776 1178.8350

(between\_SS / total\_SS = 71.9 %)

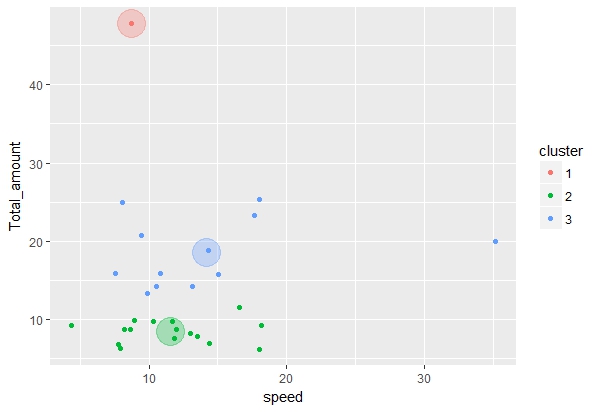
Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size" "iter"

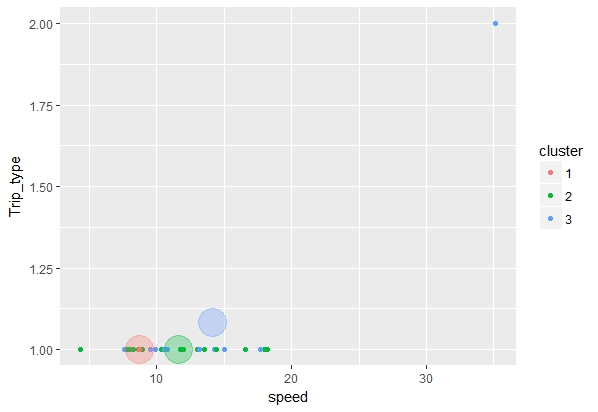
[9] "ifault"

> c(wss[3], sum(km$withinss))

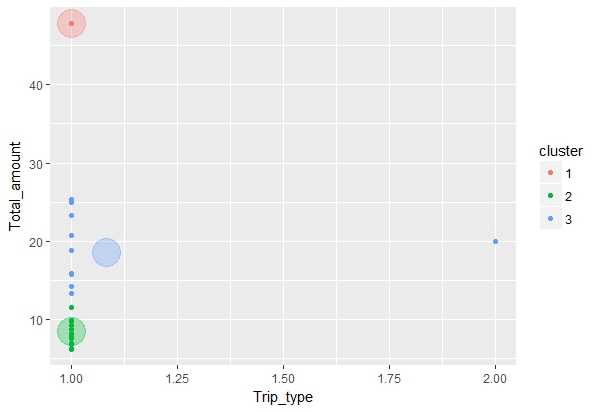
[1] 1546.613 1546.613



This shows passengers who pay tip are divided into 3 clusters with respect to Total amount charged for trip versus Speed



This shows passengers who pay tip are divided into 3 clusters with respect to Trip type versus Speed



This shows passengers who pay tip are divided into 3 clusters with respect to Trip type versus Total amount charged for trip

As shown by the figure the clusters where not made properly hence removed duration and again applied clustering to improve the model.

> kmdata\_orig[1:10,]

Payment\_type speed Trip\_type Total\_amount

[1,] 1 15.050000 1 15.80

[2,] 1 14.287500 1 18.80

[3,] 2 13.000000 1 8.30

[4,] 1 13.162500 1 14.30

[5,] 2 8.666667 1 8.80

[6,] 1 7.600000 1 15.95

[7,] 2 7.800000 1 6.80

[8,] 1 8.068966 1 24.95

[9,] 2 10.320000 1 9.80

[10,] 2 18.000000 1 25.30

> wss<-numeric(15)

> for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

> km = kmeans(kmdata\_orig, 3, nstart = 25)

> km

K-means clustering with 3 clusters of sizes 7, 21, 1

Cluster means:

Payment\_type speed Trip\_type Total\_amount

1 1.285714 16.815747 1.142857 21.272857

2 1.523810 11.308613 1.000000 9.995238

3 1.000000 8.719149 1.000000 47.810000

Clustering vector:

[1] 1 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 3 2 1 2

Within cluster sum of squares by cluster:

[1] 551.7692 445.6175 0.0000

(between\_SS / total\_SS = 67.0 %)

Available components:

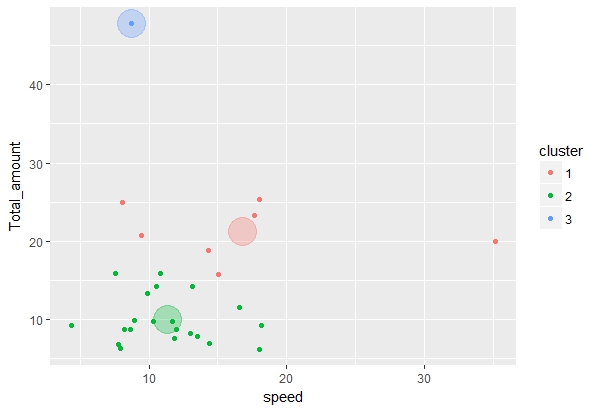
[1] "cluster" "centers" "totss" "withinss"

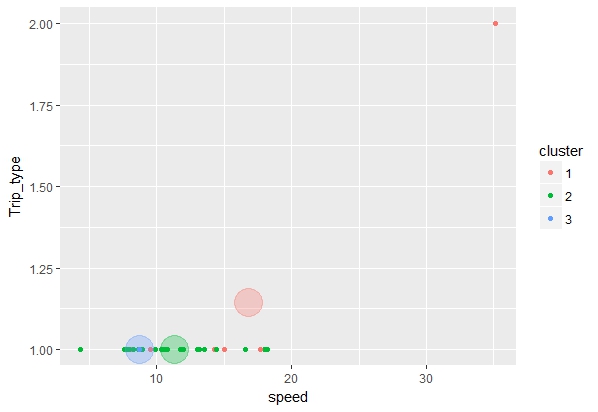
[5] "tot.withinss" "betweenss" "size" "iter"

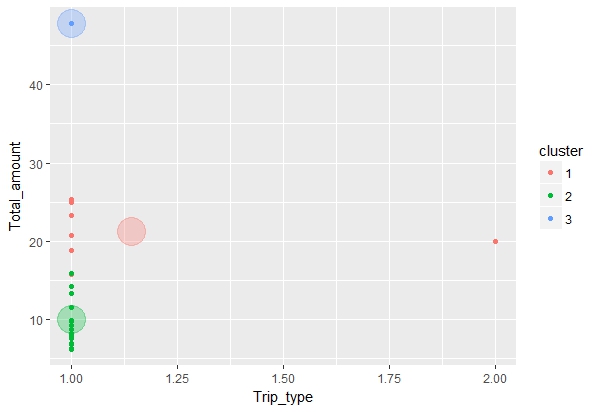
[9] "ifault"

> c(wss[3], sum(km$withinss))

[1] 997.3867 997.3867







From the above plots, we can infer that the three cluster formed – the factors most prominently helps to determine model and have strong effect are Trip type and Total amount. For the standard rate the blue cluster pays the high total amount and they appreciate low speed.

### Question 3

### Report mean and median trip distance grouped by hour of day

|  |
| --- |
| Hour N mean sd se  1 0 67158 3.115276 2.963919 0.011437142  2 1 53773 3.017347 2.889377 0.012460118  3 2 41196 3.046176 2.968572 0.014625812  4 3 31640 3.212945 4.631416 0.026037274  5 4 26424 3.526555 3.535880 0.021751954  6 5 16700 4.133474 3.969791 0.030719163  7 6 22667 4.055149 3.973244 0.026390538  8 7 41978 3.284394 3.272076 0.015970281  9 8 58965 3.048450 3.101140 0.012770979  10 9 62027 2.999105 3.075405 0.012348434  11 10 57468 2.944482 3.054072 0.012739906  12 11 56791 2.912015 3.091721 0.012973603  13 12 57828 2.903065 3.075712 0.012790177  14 13 57477 2.878294 3.112318 0.012981859  15 14 66664 2.864304 3.086522 0.011954287  16 15 73777 2.857040 3.119719 0.011485633  17 16 79157 2.779852 2.999746 0.010662029  18 17 88022 2.679114 2.858260 0.009633990  19 18 97245 2.653222 2.724282 0.008736118  20 19 96141 2.715597 2.707118 0.008730778  21 20 90785 2.777052 2.717110 0.009017793  22 21 86543 2.999189 3.051589 0.010373135  23 22 84705 3.185394 3.099036 0.010648100  24 23 79795 3.191538 3.017084 0.010680695 |

|  |
| --- |
| > summary(cdata) |
| Hour N mean sd se |
| 0 : 1 Min. :16700 Min. :2.653 Min. :2.707 Min. :0.008731 |
| 1 : 1 1st Qu.:50824 1st Qu.:2.862 1st Qu.:2.967 1st Qu.:0.010659 |
| 2 : 1 Median :60496 Median :2.999 Median :3.076 Median :0.012404 |
| 3 : 1 Mean :62289 Mean :3.074 Mean :3.171 Mean :0.014080 |
| 4 : 1 3rd Qu.:81023 3rd Qu.:3.187 3rd Qu.:3.114 3rd Qu.:0.013393 |
| 5 : 1 Max. :97245 Max. :4.133 Max. :4.631 Max. :0.030719 |
| (Other):18 |
| > str(cdata) |
| 'data.frame': 24 obs. of 5 variables: |
| $ Hour: Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ... |
| $ N : int 67158 53773 41196 31640 26424 16700 22667 41978 58965 62027 ... |
| $ mean: num 3.12 3.02 3.05 3.21 3.53 ... |
| $ sd : num 2.96 2.89 2.97 4.63 3.54 ... |
| $ se : num 0.0114 0.0125 0.0146 0.026 0.0218 ... |
| > cdata |

**Source Code:**

#Report mean and median trip distance grouped by hour of day.

library(ggplot2)

library(dplyr)

library(plyr)

cdata <- ddply(cleanData, c("Hour"), summarise,

N = length(Trip\_distance),

mean = mean(Trip\_distance),

sd = sd(Trip\_distance),

se = sd / sqrt(N))

summary(cdata)

str(cdata)

cdata

### We'd like to get a rough sense of identifying trips that originate or terminate at one of the NYC area airports. Can you provide a count of how many transactions fit this criteria, the average fair, and any other interesting characteristics of these trips?

#### Analysis:

* Number of trips that originate or terminate at one of the NYC area airports: 5558 trips
  + (Rate code ID ==2 OR 3)
* Average fare : 50.1308$ and Average total fare for NYC area airports : 58.44823$
* Average Fare amount : 12.55982$ for all the Trip and
* Average Total amount for all the Trip : 15.06217$

Trips distribution by trip distances and hour of the day

* Average Airport Distance: 10.36699 miles
* Average Total Distance: 2.698141 miles
* Average Airport Hour (Average Hour for airport area trips) : 13.00468 hrs
* Average Total Hour (all trips): 13.53407 hrs
* Average Airport Duration (Average Duration for airport area trips): 30.09464
* Average Total Duration (Average Duration for all trips): 19.77605
* Average Airport Tip Amount (Average Tip amount for airport area trips given by passengers): 4.348816$
* Average Airport Tip Amount (Average Tip Amount for all trips) : 1.235815$

### Question 4

### 4.1 Build a derived variable for tip as a percentage of the total fare.

#### Analysis:

|  |  |
| --- | --- |
| summary(Tip\_Percentage)  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.000 0.000 0.000 6.634 16.670 100.000   |  | | --- | | str(Tip\_Percentage)  num [1:1494926] 20 0 8.62 0 0 ... | |

Source code:

Tip\_Percentage<- (100\*mydat$Tip\_amount)/mydat$Total\_amount

mydat$Tip\_Percentage <- Tip\_Percentage

Tip\_Percentage

### Build a predictive model for tip as a percentage of the total fare. Use as much of the data as you like (or all of it). We will validate a sample.

#### Approach:

To build the predictive model the first step was to start with regression analysis that influence that a set of variables has on the outcome depends on the other variables.

**As it is useful in answering what tip should a passenger should give as a percentage of total fare.**

### REGRESSION:

Regression is a useful explanatory tool that identify the input variables that have the great statistical influence on the outcome.

In this case it can be found that payment type, total amount can be an excellent predictors of the Tip amount success for passengers of Green taxi.

The use case: A simple linear regression analysis is used to model Tip prices as a function of the passengers trip in Green taxi. Such a model helps us to determine the tip amount payed by the passenger. If the model was to determine that passenger should pay tip or not then would have applied logistic regression.

The linear regression was applied twice to further improve by excluding some factors which are not contributing significantly for calculating Tip percentage.

The data was a huge chuck so build the model with 2% of the whole data as used for Training the model.

Model 1

The independent factors used are –

* Total\_amount
* Passenger\_count
* speed
* Tolls\_amount
* Month
* Trip\_distance
* Duration
* Hour
* Wday
* Extra
* Payment\_type

Dependent Factor – Tip Percentage

|  |
| --- |
| results <-lm(formula = Tip\_Percentage ~ |
| Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type |
| , data = trainn) |
|  |

Call:

lm(formula = Tip\_Percentage ~ Total\_amount + Passenger\_count +

speed + Tolls\_amount + month + Trip\_distance + duration +

hour + wday + Extra + Payment\_type, data = trainn)

Residuals:

Min 1Q Median 3Q Max

-53.666 -0.742 -0.463 2.772 88.916

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.384e+01 1.755e-01 135.884 < 2e-16 \*\*\*

Total\_amount 2.216e-01 6.034e-03 36.719 < 2e-16 \*\*\*

Passenger\_count 2.442e-03 2.983e-02 0.082 0.93476

speed 6.634e-02 5.200e-03 12.757 < 2e-16 \*\*\*

Tolls\_amount -3.193e-01 3.844e-02 -8.307 < 2e-16 \*\*\*

month NA NA NA NA

Trip\_distance -7.483e-01 2.288e-02 -32.707 < 2e-16 \*\*\*

duration -8.866e-04 3.432e-04 -2.583 0.00979 \*\*

hour 9.243e-03 4.767e-03 1.939 0.05252 .

wday -1.146e-02 1.530e-02 -0.749 0.45395

Extra -1.879e-02 8.837e-02 -0.213 0.83162

Payment\_type -1.250e+01 6.253e-02 -199.960 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.394 on 29887 degrees of freedom

Multiple R-squared: 0.6235, Adjusted R-squared: 0.6234

F-statistic: 4949 on 10 and 29887 DF, p-value: < 2.2e-16

The dropping of factors resulted in a change to the estimates of the remaining parameters and their statistical significance.

Refined the model further by excluding the factors Passenger count, hour, wday and Extra.

Final Model:

|  |
| --- |
| results <-lm(formula = Tip\_Percentage ~ |
| Total\_amount +speed +Tolls\_amount+Trip\_distance+duration+Payment\_type |
| , data = trainn) |
| summary(results) |

Call:

lm(formula = Tip\_Percentage ~ Total\_amount + speed + Tolls\_amount +

Trip\_distance + duration + Payment\_type, data = trainn)

Residuals:

Min 1Q Median 3Q Max

-53.679 -0.723 -0.472 2.783 88.838

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.393e+01 1.432e-01 167.129 < 2e-16 \*\*\*

Total\_amount 2.216e-01 6.030e-03 36.753 < 2e-16 \*\*\*

speed 6.561e-02 5.167e-03 12.698 < 2e-16 \*\*\*

Tolls\_amount -3.192e-01 3.841e-02 -8.311 < 2e-16 \*\*\*

Trip\_distance -7.485e-01 2.285e-02 -32.754 < 2e-16 \*\*\*

duration -9.006e-04 3.431e-04 -2.625 0.00867 \*\*

Payment\_type -1.251e+01 6.252e-02 -200.023 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.394 on 29891 degrees of freedom

Multiple R-squared: 0.6234, Adjusted R-squared: 0.6234

F-statistic: 8248 on 6 and 29891 DF, p-value: < 2.2e-16

#### Analysis:

In this model the p value of 2.2e-16 is small, which indicates that the null hypothesis should be rejected. R-square can be increased by adding more factors but can lead to overfitting.

R Square is closer to 1 hence the model is better in explaining the data.

### DECISION TREE (Prediction Tree)

#### The second model applied –

To specify the sequence of decisions and consequences. The goal was to predict a response that is the Tip amount percentage. The inpur variables used are Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type.

Then gradually refining the tree by dropping off the variables and predicting the factors which will help in making the decision. Constructed a Tree with various test point and branches.

Every test point was testing a particular variable and the branch represented the decision.

fit <- rpart(Tip\_Percentage ~

Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type

,data = trainn, method="anova")

summary(fit)

rpart.plot(fit)

plot(fit)

text(fit)

Call:

rpart(formula = Tip\_Percentage ~ Total\_amount + Passenger\_count +

speed + Tolls\_amount + month + Trip\_distance + duration +

hour + wday + Extra + Payment\_type, data = trainn, method = "anova")

n= 29898

CP nsplit rel error xerror xstd

1 0.6423802 0 1.0000000 1.0000166 0.01500683

2 0.0100000 1 0.3576198 0.3577034 0.01235409

Variable importance

Payment\_type Total\_amount Trip\_distance duration

67 16 8 6

speed Tolls\_amount

2 1

Node number 1: 29898 observations, complexity param=0.6423802

mean=6.600872, MSE=77.24142

left son=2 (15919 obs) right son=3 (13979 obs)

Primary splits:

Payment\_type < 1.5 to the right, improve=0.642380200, (0 missing)

Total\_amount < 9.33 to the left, improve=0.071506630, (0 missing)

Trip\_distance < 2.195 to the left, improve=0.012877190, (0 missing)

duration < 11.5 to the left, improve=0.009321382, (0 missing)

speed < 10.6641 to the left, improve=0.005193137, (0 missing)

Surrogate splits:

Total\_amount < 12.805 to the left, agree=0.645, adj=0.241, (0 split)

Trip\_distance < 3.045 to the left, agree=0.586, adj=0.114, (0 split)

duration < 12.5 to the left, agree=0.575, adj=0.090, (0 split)

speed < 15.10333 to the left, agree=0.546, adj=0.030, (0 split)

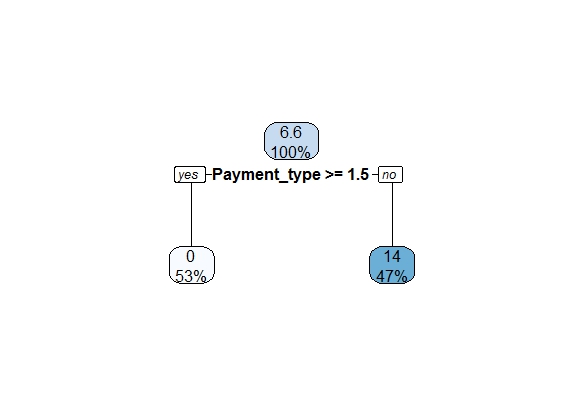
Tolls\_amount < 3.535 to the left, agree=0.540, adj=0.016, (0 split)

Node number 2: 15919 observations

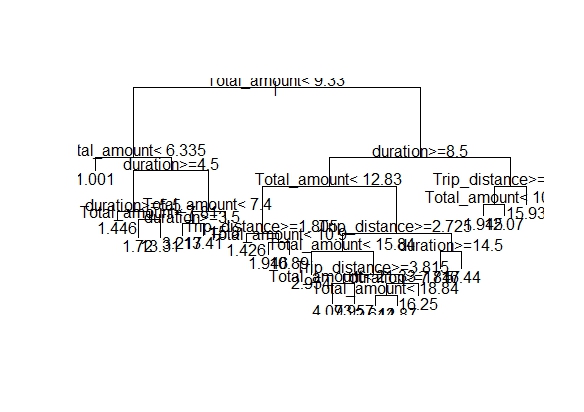
mean=0, MSE=0

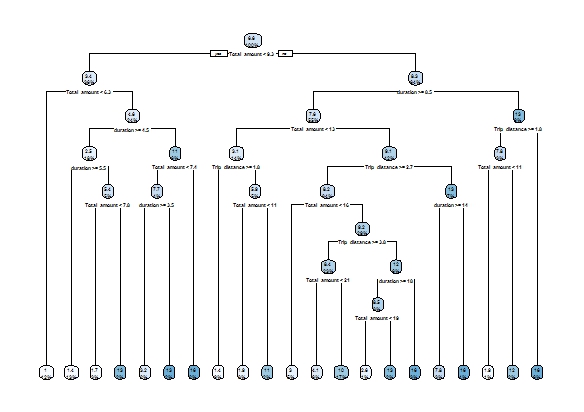
Node number 3: 13979 observations

mean=14.11781, MSE=59.07964



Only payment type was the major factor for information gain. When refined the model and after many fit and trials, excluded payment type and the decision tree formed





Too many decisions and branches were obtained, then tried more refining and came to the final model

Call:

|  |
| --- |
| rpart(formula = Tip\_Percentage ~ Total\_amount + Trip\_distance,  data = trainn, method = "anova")  n= 29898 |

CP nsplit rel error xerror xstd

1 0.07150663 0 1.0000000 1.0000739 0.01500804

2 0.02995429 1 0.9284934 0.9286249 0.01453868

3 0.02228220 3 0.8685848 0.8693095 0.01365870

4 0.01974427 5 0.8240204 0.8248483 0.01351661

5 0.01415897 7 0.7845319 0.7870333 0.01349179

6 0.01265796 9 0.7562139 0.7600386 0.01347232

7 0.01119631 10 0.7435560 0.7516366 0.01346693

8 0.01000000 11 0.7323596 0.7431132 0.01347029

Variable importance

Total\_amount Trip\_distance

52 48

Node number 1: 29898 observations, complexity param=0.07150663

mean=6.600872, MSE=77.24142

left son=2 (10630 obs) right son=3 (19268 obs)

Primary splits:

Total\_amount < 9.33 to the left, improve=0.07150663, (0 missing)

Trip\_distance < 2.195 to the left, improve=0.01287719, (0 missing)

Surrogate splits:

Trip\_distance < 1.525 to the left, agree=0.893, adj=0.7, (0 split)

Node number 2: 10630 observations, complexity param=0.0222822

mean=3.436774, MSE=48.99828

left son=4 (3448 obs) right son=5 (7182 obs)

Primary splits:

Total\_amount < 6.335 to the left, improve=0.05815318, (0 missing)

Trip\_distance < 1.185 to the right, improve=0.04124712, (0 missing)

Surrogate splits:

Trip\_distance < 0.735 to the left, agree=0.82, adj=0.446, (0 split)

Node number 3: 19268 observations, complexity param=0.02995429

mean=8.34648, MSE=84.25251

left son=6 (17016 obs) right son=7 (2252 obs)

Primary splits:

Trip\_distance < 1.565 to the right, improve=0.03208822, (0 missing)

Total\_amount < 22.845 to the left, improve=0.01936158, (0 missing)

Surrogate splits:

Total\_amount < 9.76 to the right, agree=0.899, adj=0.136, (0 split)

Node number 4: 3448 observations

mean=1.000556, MSE=17.22868

Node number 5: 7182 observations, complexity param=0.0222822

mean=4.606376, MSE=60.03314

left son=10 (5046 obs) right son=11 (2136 obs)

Primary splits:

Trip\_distance < 0.915 to the right, improve=0.16844460, (0 missing)

Total\_amount < 6.795 to the right, improve=0.05307003, (0 missing)

Surrogate splits:

Total\_amount < 7.275 to the right, agree=0.758, adj=0.187, (0 split)

Node number 6: 17016 observations, complexity param=0.02995429

mean=7.748317, MSE=75.58598

left son=12 (4516 obs) right son=13 (12500 obs)

Primary splits:

Total\_amount < 12.83 to the left, improve=0.06706684, (0 missing)

Trip\_distance < 1.755 to the right, improve=0.00284349, (0 missing)

Surrogate splits:

Trip\_distance < 2.395 to the left, agree=0.86, adj=0.471, (0 split)

Node number 7: 2252 observations

mean=12.86617, MSE=126.6052

Node number 10: 5046 observations

mean=2.537421, MSE=35.55252

Node number 11: 2136 observations

mean=9.493992, MSE=83.86412

Node number 12: 4516 observations, complexity param=0.01265796

mean=4.002448, MSE=51.38544

left son=24 (2713 obs) right son=25 (1803 obs)

Primary splits:

Trip\_distance < 1.965 to the right, improve=0.12596850, (0 missing)

Total\_amount < 10.34 to the left, improve=0.03760121, (0 missing)

Surrogate splits:

Total\_amount < 10.795 to the right, agree=0.684, adj=0.209, (0 split)

Node number 13: 12500 observations, complexity param=0.01974427

mean=9.101624, MSE=77.4284

left son=26 (10338 obs) right son=27 (2162 obs)

Primary splits:

Trip\_distance < 2.725 to the right, improve=0.04423448, (0 missing)

Total\_amount < 12.97 to the right, improve=0.01695495, (0 missing)

Surrogate splits:

Total\_amount < 14.17 to the right, agree=0.871, adj=0.253, (0 split)

Node number 24: 2713 observations

mean=1.928375, MSE=27.55331

Node number 25: 1803 observations, complexity param=0.01119631

mean=7.123335, MSE=71.0331

left son=50 (709 obs) right son=51 (1094 obs)

Primary splits:

Total\_amount < 10.34 to the left, improve=0.20188810, (0 missing)

Trip\_distance < 1.805 to the right, improve=0.02940138, (0 missing)

Node number 26: 10338 observations, complexity param=0.01974427

mean=8.255293, MSE=74.61877

left son=52 (1381 obs) right son=53 (8957 obs)

Primary splits:

Total\_amount < 15.32 to the left, improve=0.06271748, (0 missing)

Trip\_distance < 2.905 to the right, improve=0.00247195, (0 missing)

Surrogate splits:

Trip\_distance < 2.775 to the left, agree=0.868, adj=0.01, (0 split)

Node number 27: 2162 observations

mean=13.14851, MSE=71.06083

Node number 50: 709 observations

mean=2.419291, MSE=31.31847

Node number 51: 1094 observations

mean=10.17193, MSE=73.13669

Node number 52: 1381 observations

mean=2.745912, MSE=32.77479

Node number 53: 8957 observations, complexity param=0.01415897

mean=9.104736, MSE=75.66887

left son=106 (7461 obs) right son=107 (1496 obs)

Primary splits:

Trip\_distance < 3.555 to the right, improve=0.03156058, (0 missing)

Total\_amount < 22.845 to the left, improve=0.02483758, (0 missing)

Surrogate splits:

Total\_amount < 16.75 to the right, agree=0.858, adj=0.152, (0 split)

Node number 106: 7461 observations, complexity param=0.01415897

mean=8.412748, MSE=75.55793

left son=212 (2389 obs) right son=213 (5072 obs)

Primary splits:

Total\_amount < 21.345 to the left, improve=0.078060660, (0 missing)

Trip\_distance < 3.815 to the right, improve=0.002014569, (0 missing)

Surrogate splits:

Trip\_distance < 4.825 to the left, agree=0.814, adj=0.418, (0 split)

Node number 107: 1496 observations

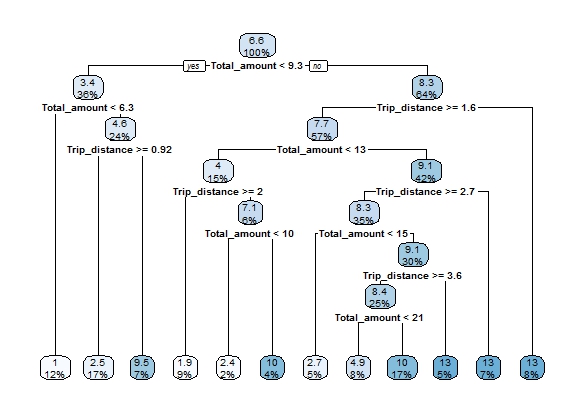
mean=12.55589, MSE=61.92357

Node number 212: 2389 observations

mean=4.874098, MSE=52.75391

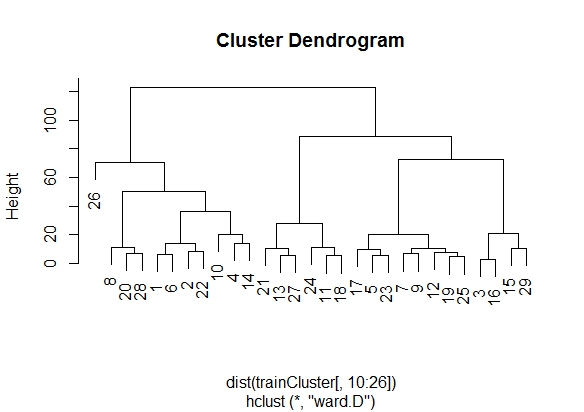
Node number 213: 5072 observations

mean=10.07951, MSE=77.62281



Hence at the first split, the decision tree chooses the Total amount attribute followed by Trip distance. There are 36% paasengers who have paid total amount less than 6.3 17% who have travelled less than 1 mile hence for them giving tip amount does not make much sense.

Then applied Cluster on the Rate code ID-



# Appendix A

## Codes:

## C1 - Main Source Code:

# Prepare Data

library(data.table)

mydat <- fread('https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv')

head(mydat)

str(mydat)

#write.csv(mydat, file = "greenTaxi.csv", row.names = FALSE)

summary(mydat)

is.data.frame(mydat)

#Cleaning the data for the predictive model for tip as a percentage of the total fare

#Clean data for derived variable for tip as a percentage of the total fare.

mydat$Total\_amount[mydat$Total\_amount<2.5] <-2.5

mydat$Fare\_amount[mydat$Fare\_amount<0] <-0

mydat$improvement\_surcharge[mydat$improvement\_surcharge<0] <-0.30

mydat$Tip\_amount[mydat$Tip\_amount<0] <-0

mydat$Trip\_type[mydat$Trip\_type=="NaN"]<-1

mydat$Trip\_type[mydat$Trip\_type=="NA"]<-1

mydat$RateCodeID[mydat$RateCodeID==99] <-2

mydat$Ehail\_fee<- NULL

mydat$Extra[mydat$Extra<0] <-0

Tip\_Percentage<- (100\*mydat$Tip\_amount)/mydat$Total\_amount

mydat$Tip\_Percentage <- Tip\_Percentage

Tip\_Percentage

summary(Tip\_Percentage)

str(Tip\_Percentage)

#write.csv(mydat, file = "greenTaxicleanDataTip.csv", row.names = FALSE)

summary(mydat)

str(mydat)

cleanData<-mydat

# Change the format of datetime from string to POSIXct objects

cleanData$lpep\_pickup\_datetime <- as.POSIXct(cleanData$lpep\_pickup\_datetime,format='%Y-%m-%d %H:%M:%S')

cleanData$Lpep\_dropoff\_datetime <- as.POSIXct(cleanData$Lpep\_dropoff\_datetime,format='%Y-%m-%d %H:%M:%S')

cleanData$month <- month(cleanData$lpep\_pickup\_datetime)

cleanData$wday <- wday(cleanData$lpep\_pickup\_datetime)

cleanData$hour <- hour(cleanData$lpep\_pickup\_datetime)

cleanData$DropOffhour <- hour(cleanData$Lpep\_dropoff\_datetime)

summary(cleanData)

cleanData$duration <- floor(as.double(cleanData$Lpep\_dropoff\_datetime-cleanData$lpep\_pickup\_datetime)/60.0)

#cleanData$TipPercentOnFareAmount <- (cleanData$Tip\_amount/cleanData$Fare\_amount) \* 100.0

cleanData$speed<- (60\* cleanData$Trip\_distance/cleanData$duration)

cleanData$speed[cleanData$speed=="NaN"]<-0

cleanData$speed[cleanData$speed=="Inf"]<-0

cleanData$duration[cleanData$duration=="NA"]<-0

#cleanData$BySpeed<-(cleanData$Payment\_type==1 && cleanData$speed<80 && cleanData$Tip\_Percentage<50)

#write.csv(cleanData, file = "greenTaxi1.csv", row.names = FALSE)

newLevel<-c(rep('0-6am',6),rep('6-9am',3),rep('9am-4pm',7),rep('4-7pm',3),rep('7-12pm',5))

levels(cleanData$hour)<-newLevel

levels

str(cleanData)

summary(cleanData)

#Applying Regression

library(lattice)

#splom(~cleanData[c(10,11,15)], groups = NULL, data = cleanData, axis.line.tck =0,axis.text.alpha=0)

VendorID<-as.factor(cleanData$VendorID)

Passenger\_count<-as.factor(cleanData$Passenger\_count)

Trip\_distance<-as.factor(cleanData$Trip\_distance)

Total\_amount<-as.factor(cleanData$Total\_amount)

Payment\_type<-as.factor(cleanData$Payment\_type)

Hour<-as.factor(cleanData$hour)

Week<-as.factor(cleanData$wday)

Month\_day<-as.factor(cleanData$month)

duration<-as.factor(cleanData$duration)

Speed\_mph<-as.factor(cleanData$speed)

Tolls\_amount<-as.factor(cleanData$Tolls\_amount)

Extra<-as.factor(cleanData$Extra)

## 2% of the sample size

smp\_size <- floor(0.02 \* nrow(cleanData))

## set the seed to make your partition reproductible

set.seed(123)

train\_ind <- sample(seq\_len(nrow(cleanData)), size = smp\_size)

trainn <- cleanData[train\_ind, ]

test <- cleanData[-train\_ind, ]

summary(trainn)

str(trainn)

results <-lm(formula = Tip\_Percentage ~

Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type

, data = trainn)

results <-lm(formula = Tip\_Percentage ~

Total\_amount +speed +Tolls\_amount+Trip\_distance+duration+Payment\_type

, data = trainn)

summary(results)

library("rpart")

library("rpart.plot")

tip\_decision<-trainn

tip\_decision

summary(tip\_decision)

fit <- rpart(Tip\_Percentage ~

Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type

,data = trainn, method="anova")

summary(fit)

rpart.plot(fit)

plot(fit)

text(fit)

Fit2 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Tolls\_amount+Trip\_distance+duration+Payment\_type

, data = trainn,method="anova")

summary(Fit2)

rpart.plot(Fit2)

plot(Fit2)

text(Fit2)

Fit3 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Trip\_distance+duration+Payment\_type

, data = trainn,method="anova")

summary(Fit3)

rpart.plot(Fit3)

plot(Fit3)

text(Fit3)

Fit4 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Trip\_distance+duration

, data = trainn,method="anova")

summary(Fit4)

rpart.plot(Fit4)

plot(Fit4)

text(Fit4)

Fit5 <-rpart(Tip\_Percentage ~

Total\_amount +Trip\_distance+duration

, data = trainn,method="anova")

summary(Fit5)

rpart.plot(Fit5)

plot(Fit5)

text(Fit5)

Fit6 <-rpart(Tip\_Percentage ~

Total\_amount +Trip\_distance

,data = trainn,method="anova")

summary(Fit6)

rpart.plot(Fit6)

plot(Fit6)

text(Fit6)

#write.csv(trainn, file = "mytaxidectree.csv", row.names = FALSE)

summary(Fit6)

rpart.plot(Fit6, type = 4,extra = 0)

library(ggplot2)

## 2% of the sample size

s <- floor(0.00002 \* nrow(cleanData))

## set the seed to make your partition reproducible

set.seed(1234)

train\_ind <- sample(seq\_len(nrow(cleanData)), size = s)

trainCluster <- cleanData[train\_ind, ]

summary(trainCluster)

str(trainCluster)

clusterPlot <- function(type) {

clusters <- hclust(dist(trainCluster[, 10:26]), method = type)

plot(clusters)

clusterCut <- cutree(clusters, 3)

show(table(clusterCut, trainCluster$RateCodeID)) # show required, else will not print

}

d <- dist(trainCluster, method = "euclidean") # distance matrix

fit <- hclust(d, method="ward")

plot(fit) # display dendogram

clusterPlot('ward.D')

#write.csv(trainn, file = "mygreen.csv", row.names = FALSE)

################################

#Report mean and median trip distance grouped by hour of day.

library(ggplot2)

library(dplyr)

library(plyr)

cdata <- ddply(cleanData, c("Hour"), summarise,

N = length(Trip\_distance),

mean = mean(Trip\_distance),

sd = sd(Trip\_distance),

se = sd / sqrt(N))

summary(cdata)

str(cdata)

cdata

############################################

cleanData$airport\_trips <- data((cleanData$RateCodeID==2) | (cleanData$RateCodeID==3))

summary(cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

Total\_aiport\_trips<-summary(cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

Total\_aiport\_trips

Total\_trips<- filter(cleanData, cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

count(Total\_trips)

nrow(Total\_trips)

####Average fare and total fare

AverageFareAmount <- mean(Total\_trips$Fare\_amount)

#Average Fare amount for Airport Trip

AverageFareAmount

AverageTotalAmount <-mean(Total\_trips$Total\_amount)

#Average Total amount for Airport Trip

AverageTotalAmount

AverageFareAmountallTrips <- mean(cleanData$Fare\_amount)

#Average Fare amount for all the Trip

AverageFareAmountallTrips

AverageTotalAmountallTrips <-mean(cleanData$Total\_amount)

#Average Total amount for all the Trip

AverageTotalAmountallTrips

#Trips distribution by trip distances and hour of the day

# Airport Trip Distance

AirportDist = Total\_trips$Trip\_distance # airport trips

AvgAirportDist = mean(AirportDist)

#Average Airport Distance

AvgAirportDist

#Average Total Distance

AverageDist = mean(cleanData$Trip\_distance)

AverageDist

# Airport Trip Hour

AirportHour = Total\_trips$hour # airport trips

AvgAirportHour = mean(AirportHour)

#Average Airport Hour

AvgAirportHour

#Average Total Hour

AverageHour = mean(cleanData$hour)

AverageHour

# Airport Trip Duration

AirportDuration = Total\_trips$duration # airport trips

AvgAirportDuration = mean(AirportDuration)

#Average Airport Duration

AvgAirportDuration

#Average Total Duration

AverageDuration = mean(cleanData$duration)

AverageDuration

# Airport Trip Tip amount

AirportTip = Total\_trips$Tip\_amount # airport trips

AvgAirportTip = mean(AirportTip)

#Average Airport Tip Amount

AvgAirportTip

#Average Total Tip Amount

AverageTip = mean(cleanData$Tip\_amount)

AverageTip

#Report any structure you find and any hypotheses you have about that structure

set1 = cleanData[cleanData$Tip\_amount>0]

set2 = cleanData[cleanData$Tip\_amount==0]

#Total passenger paying Tip

nrow(set1)

#Total passenger not paying Tip

nrow(set2)

require(graphics)

cleanData$Tip\_Percentage[cleanData$Tip\_Percentage=="NA"]<-0

offer<-sample(c(Payment\_type),size = 500,replace = T)

amountPur<-sample(c(Total\_amount),size = 500,replace = T)

offertest<-data.frame(offer=as.factor(offer))

model<-aov(amountPur ~ offer, data = offertest)

summary(model)

require(graphics)

offer<-sample(c(Payment\_type,Trip\_distance),size = 500,replace = T)

amountPur<-sample(c(Total\_amount),size = 500,replace = T)

offertest<-data.frame(offer=as.factor(offer))

model<-aov(amountPur ~ offer, data = offertest)

summary(model)

#Clustering based on Payment type and Rate code ID or Trip Type

library(plyr)

library(ggplot2)

library(cluster)

library(lattice)

library(grid)

library(gridExtra)

kmdata\_orig<-as.matrix(trainCluster[,c("Payment\_type","speed", "Trip\_type","Total\_amount","duration")])

kmdata\_orig[1:10,]

wss<-numeric(15)

for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

km = kmeans(kmdata\_orig, 3, nstart = 25)

km

c(wss[3], sum(km$withinss))

#preparation of the data and clustering results

df = as.data.frame(kmdata\_orig[,2:4])

df$cluster = factor(km$cluster)

centers=as.data.frame(km$centers)

g1=ggplot(data = df, aes(x=speed, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g1))

g1

g2=ggplot(data = df, aes(x=speed, y =Trip\_type, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Trip\_type, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g2))

g2

g3=ggplot(data = df, aes(x=Trip\_type, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=Trip\_type, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g3))

g3

#Clustering based on Payment type and Rate code ID or Trip Type

library(plyr)

library(ggplot2)

library(cluster)

library(lattice)

library(grid)

library(gridExtra)

kmdata\_orig<-as.matrix(trainCluster[,c("Payment\_type","speed", "Trip\_type","Total\_amount")])

kmdata\_orig[1:10,]

wss<-numeric(15)

for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

km = kmeans(kmdata\_orig, 3, nstart = 25)

km

c(wss[3], sum(km$withinss))

#preparation of the data and clustering results

df = as.data.frame(kmdata\_orig[,2:4])

df$cluster = factor(km$cluster)

centers=as.data.frame(km$centers)

g1=ggplot(data = df, aes(x=speed, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g1))

g1

g2=ggplot(data = df, aes(x=speed, y =Trip\_type, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Trip\_type, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g2))

g2

g3=ggplot(data = df, aes(x=Trip\_type, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=Trip\_type, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g3))

g3

## C2 – Source Code: To predict model for Tip percentage Q4 part 2:

Source Code for Model prediction:

# Prepare Data

library(data.table)

mydat <- fread('https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv')

head(mydat)

str(mydat)

#write.csv(mydat, file = "greenTaxi.csv", row.names = FALSE)

summary(mydat)

is.data.frame(mydat)

#Cleaning the data for the predictive model for tip as a percentage of the total fare

#Clean data for derived variable for tip as a percentage of the total fare.

mydat$Total\_amount[mydat$Total\_amount<2.5] <-2.5

mydat$Fare\_amount[mydat$Fare\_amount<0] <-0

mydat$improvement\_surcharge[mydat$improvement\_surcharge<0] <-0.30

mydat$Tip\_amount[mydat$Tip\_amount<0] <-0

mydat$Trip\_type[mydat$Trip\_type=="NaN"]<-1

mydat$Trip\_type[mydat$Trip\_type=="NA"]<-1

mydat$RateCodeID[mydat$RateCodeID==99] <-2

mydat$Ehail\_fee<- NULL

mydat$Extra[mydat$Extra<0] <-0

Tip\_Percentage<- (100\*mydat$Tip\_amount)/mydat$Total\_amount

mydat$Tip\_Percentage <- Tip\_Percentage

Tip\_Percentage

summary(Tip\_Percentage)

str(Tip\_Percentage)

#write.csv(mydat, file = "greenTaxicleanDataTip.csv", row.names = FALSE)

summary(mydat)

str(mydat)

cleanData<-mydat

# Change the format of datetime from string to POSIXct objects

cleanData$lpep\_pickup\_datetime <- as.POSIXct(cleanData$lpep\_pickup\_datetime,format='%Y-%m-%d %H:%M:%S')

cleanData$Lpep\_dropoff\_datetime <- as.POSIXct(cleanData$Lpep\_dropoff\_datetime,format='%Y-%m-%d %H:%M:%S')

cleanData$month <- month(cleanData$lpep\_pickup\_datetime)

cleanData$wday <- wday(cleanData$lpep\_pickup\_datetime)

cleanData$hour <- hour(cleanData$lpep\_pickup\_datetime)

cleanData$DropOffhour <- hour(cleanData$Lpep\_dropoff\_datetime)

summary(cleanData)

cleanData$duration <- floor(as.double(cleanData$Lpep\_dropoff\_datetime-cleanData$lpep\_pickup\_datetime)/60.0)

#cleanData$TipPercentOnFareAmount <- (cleanData$Tip\_amount/cleanData$Fare\_amount) \* 100.0

cleanData$speed<- (60\* cleanData$Trip\_distance/cleanData$duration)

cleanData$speed[cleanData$speed=="NaN"]<-0

cleanData$speed[cleanData$speed=="Inf"]<-0

cleanData$duration[cleanData$duration=="NA"]<-0

#cleanData$BySpeed<-(cleanData$Payment\_type==1 && cleanData$speed<80 && cleanData$Tip\_Percentage<50)

#write.csv(cleanData, file = "greenTaxi1.csv", row.names = FALSE)

newLevel<-c(rep('0-6am',6),rep('6-9am',3),rep('9am-4pm',7),rep('4-7pm',3),rep('7-12pm',5))

levels(cleanData$hour)<-newLevel

levels

str(cleanData)

summary(cleanData)

#Applying Regression

library(lattice)

#splom(~cleanData[c(10,11,15)], groups = NULL, data = cleanData, axis.line.tck =0,axis.text.alpha=0)

VendorID<-as.factor(cleanData$VendorID)

Passenger\_count<-as.factor(cleanData$Passenger\_count)

Trip\_distance<-as.factor(cleanData$Trip\_distance)

Total\_amount<-as.factor(cleanData$Total\_amount)

Payment\_type<-as.factor(cleanData$Payment\_type)

Hour<-as.factor(cleanData$hour)

Week<-as.factor(cleanData$wday)

Month\_day<-as.factor(cleanData$month)

duration<-as.factor(cleanData$duration)

Speed\_mph<-as.factor(cleanData$speed)

Tolls\_amount<-as.factor(cleanData$Tolls\_amount)

Extra<-as.factor(cleanData$Extra)

## 2% of the sample size

smp\_size <- floor(0.02 \* nrow(cleanData))

## set the seed to make your partition reproductible

set.seed(123)

train\_ind <- sample(seq\_len(nrow(cleanData)), size = smp\_size)

trainn <- cleanData[train\_ind, ]

test <- cleanData[-train\_ind, ]

summary(trainn)

str(trainn)

results <-lm(formula = Tip\_Percentage ~

Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type

, data = trainn)

results <-lm(formula = Tip\_Percentage ~

Total\_amount +speed +Tolls\_amount+Trip\_distance+duration+Payment\_type

, data = trainn)

summary(results)

#Decision Tree

library("rpart")

library("rpart.plot")

tip\_decision<-trainn

tip\_decision

summary(tip\_decision)

fit <- rpart(Tip\_Percentage ~

Total\_amount +Passenger\_count +speed +Tolls\_amount+month+Trip\_distance+duration+hour+wday+Extra+Payment\_type

,data = trainn, method="anova")

summary(fit)

rpart.plot(fit)

plot(fit)

text(fit)

Fit2 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Tolls\_amount+Trip\_distance+duration+Payment\_type

, data = trainn,method="anova")

summary(Fit2)

rpart.plot(Fit2)

plot(Fit2)

text(Fit2)

Fit3 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Trip\_distance+duration+Payment\_type

, data = trainn,method="anova")

summary(Fit3)

rpart.plot(Fit3)

plot(Fit3)

text(Fit3)

Fit4 <-rpart(Tip\_Percentage ~

Total\_amount +speed +Trip\_distance+duration

, data = trainn,method="anova")

summary(Fit4)

rpart.plot(Fit4)

plot(Fit4)

text(Fit4)

Fit5 <-rpart(Tip\_Percentage ~

Total\_amount +Trip\_distance+duration

, data = trainn,method="anova")

summary(Fit5)

rpart.plot(Fit5)

plot(Fit5)

text(Fit5)

Fit6 <-rpart(Tip\_Percentage ~

Total\_amount +Trip\_distance

,data = trainn,method="anova")

summary(Fit6)

rpart.plot(Fit6)

plot(Fit6)

text(Fit6)

#write.csv(trainn, file = "mytaxidectree.csv", row.names = FALSE)

summary(Fit6)

rpart.plot(Fit6, type = 4,extra = 0)

library(ggplot2)

## 2% of the sample size

s <- floor(0.00002 \* nrow(cleanData))

## set the seed to make your partition reproductible

set.seed(1234)

train\_ind <- sample(seq\_len(nrow(cleanData)), size = s)

trainCluster <- cleanData[train\_ind, ]

summary(trainCluster)

str(trainCluster)

clusterPlot <- function(type) {

clusters <- hclust(dist(trainCluster[, 10:26]), method = type)

plot(clusters)

clusterCut <- cutree(clusters, 3)

show(table(clusterCut, trainCluster$RateCodeID)) # show required, else will not print

}

d <- dist(trainCluster, method = "euclidean") # distance matrix

fit <- hclust(d, method="ward")

plot(fit) # display dendogram

clusterPlot('ward.D')

## C3 – Source Code: To create Histograms and bar plot:

hist(mydat$VendorID, main = "Number of vendor", xlab = "vendor",

col="darkgreen")

barplot(table(mydat$VendorID),main = "Distribution of vendor id" , xlab = "vendor Id")

barplot(table(mydat$RateCodeID), main = "Distribution of Rate code id" , xlab = "Rate code Id")

barplot(table(mydat$Passenger\_count),main = "Distribution of passenger count" , xlab = "passenger count")

barplot(table(mydat$Trip\_type),main = "Distribution of trip type" , xlab = "Trip Type")

barplot(table(mydat$Payment\_type),main = "Distribution of payment type" , xlab = "Payment type")

barplot(table(mydat$Trip\_distance),main = "Distribution of trip distance" , xlab = "Trip distance")

## C4 – Source Code: Used for clustering in Question 2 To define Structure to the data

#Clustering based on Payment type and Rate code ID or Trip Type

library(plyr)

library(ggplot2)

library(cluster)

library(lattice)

library(grid)

library(gridExtra)

kmdata\_orig<-as.matrix(trainCluster[,c("Payment\_type","speed", "Trip\_type","Total\_amount","duration")])

kmdata\_orig[1:10,]

wss<-numeric(15)

for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

km = kmeans(kmdata\_orig, 3, nstart = 25)

km

c(wss[3], sum(km$withinss))

#preparation of the data and clustering results

df = as.data.frame(kmdata\_orig[,2:4])

df$cluster = factor(km$cluster)

centers=as.data.frame(km$centers)

g1=ggplot(data = df, aes(x=speed, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g1))

g1

g2=ggplot(data = df, aes(x=speed, y =Trip\_type, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Trip\_type, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g2))

g2

g3=ggplot(data = df, aes(x=Trip\_type, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=Trip\_type, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g3))

g3

#Clustering based on Payment type and Rate code ID or Trip Type

library(plyr)

library(ggplot2)

library(cluster)

library(lattice)

library(grid)

library(gridExtra)

kmdata\_orig<-as.matrix(trainCluster[,c("Payment\_type","speed", "Trip\_type","Total\_amount")])

kmdata\_orig[1:10,]

wss<-numeric(15)

for(k in 1:15) wss[k]<-sum(kmeans(kmdata\_orig, centers = k, nstart = 25)$withinss)

km = kmeans(kmdata\_orig, 3, nstart = 25)

km

c(wss[3], sum(km$withinss))

#preparation of the data and clustering results

df = as.data.frame(kmdata\_orig[,2:4])

df$cluster = factor(km$cluster)

centers=as.data.frame(km$centers)

g1=ggplot(data = df, aes(x=speed, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g1))

g1

g2=ggplot(data = df, aes(x=speed, y =Trip\_type, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=speed, y =Trip\_type, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g2))

g2

g3=ggplot(data = df, aes(x=Trip\_type, y =Total\_amount, color=cluster ))+

geom\_point()+theme(legend.position = "right")+

geom\_point(data=centers, aes(x=Trip\_type, y =Total\_amount, color=as.factor(c(1,2,3))),

size=10,alpha=.3,show.legend =FALSE)

tmp=ggplot\_gtable(ggplot\_build(g3))

g3

## C5 – Source Code: Used for Question 3.2 To calculate various values like total fare, hours, duration for airport and all trips

**Source Code:**

############################################

cleanData$airport\_trips <- data((cleanData$RateCodeID==2) | (cleanData$RateCodeID==3))

summary(cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

Total\_aiport\_trips<-summary(cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

Total\_aiport\_trips

Total\_trips<- filter(cleanData, cleanData$RateCodeID==2 | cleanData$RateCodeID==3)

count(Total\_trips)

nrow(Total\_trips)

####Average fare and total fare

AverageFareAmount <- mean(Total\_trips$Fare\_amount)

#Average Fare amount for Airport Trip

AverageFareAmount

AverageTotalAmount <-mean(Total\_trips$Total\_amount)

#Average Total amount for Airport Trip

AverageTotalAmount

AverageFareAmountallTrips <- mean(cleanData$Fare\_amount)

#Average Fare amount for all the Trip

AverageFareAmountallTrips

AverageTotalAmountallTrips <-mean(cleanData$Total\_amount)

#Average Total amount for all the Trip

AverageTotalAmountallTrips

#Trips distribution by trip distances and hour of the day

# Airport Trip Distance

AirportDist = Total\_trips$Trip\_distance # airport trips

AvgAirportDist = mean(AirportDist)

#Average Airport Distance

AvgAirportDist

#Average Total Distance

AverageDist = mean(cleanData$Trip\_distance)

AverageDist

# Airport Trip Hour

AirportHour = Total\_trips$hour # airport trips

AvgAirportHour = mean(AirportHour)

#Average Airport Hour

AvgAirportHour

#Average Total Hour

AverageHour = mean(cleanData$hour)

AverageHour

# Airport Trip Duration

AirportDuration = Total\_trips$duration # airport trips

AvgAirportDuration = mean(AirportDuration)

#Average Airport Duration

AvgAirportDuration

#Average Total Duration

AverageDuration = mean(cleanData$duration)

AverageDuration

# Airport Trip Tip amount

AirportTip = Total\_trips$Tip\_amount # airport trips

AvgAirportTip = mean(AirportTip)

#Average Airport Tip Amount

AvgAirportTip

#Average Total Tip Amount

AverageTip = mean(cleanData$Tip\_amount)

AverageTip

## C6 - Main Source code for Visualization – To generate excel that is used in Tableau to create graphs/plots/charts

library(data.table)

library(spatial)

library(jsonlite)

library(geojsonio)

nycjson<-geojson\_read("https://raw.githubusercontent.com/dwillis/nyc-maps/master/boroughs.geojson",what='sp')

#'http://catalog.civicdashboards.com/dataset/c3555efe-cb95-48f5-8816-9083d1f30c3d/resource/57356e9e-e43c-44c0-9536-6e07ab9e2e75/download/ec25edd692b24248a2b70c95d0ed85fbtemp.geojson',what='sp')

Boroughs = c('Manhattan', 'Bronx', 'Brooklyn', 'Queens', 'Staten Island')

summary(Boroughs)

nycZone<-nycjson

nycZone[,2:3]<-NULL #only keep the borough code

taxiGreen <- fread('https://s3.amazonaws.com/nyc-tlc/trip+data/green\_tripdata\_2015-09.csv',stringsAsFactors = F)

taxiGreen$dummy1 <- NULL #handle the excess ',' in all the rows of the csv files

taxiGreen$dummy2 <- NULL

names(taxiGreen) = c("VendorID" , "Pickup\_datetime" ,"Dropoff\_datetime" ,"Store\_and\_fwd\_flag" ,

"RateCodeID" , "Pickup\_longitude" , "Pickup\_latitude" , "Dropoff\_longitude" ,

"Dropoff\_latitude" , "Passenger\_count" , "Trip\_distance" , "Fare\_amount" ,

"Extra" , "MTA\_tax" , "Tip\_amount" , "Tolls\_amount" ,

"Ehail\_fee" , "improvement\_surcharge" ,"Total\_amount" , "Payment\_type" ,

"Trip\_type" )

summary(taxiGreen)

library(sp)

library(rgdal)

PickupArea<-SpatialPoints(cbind(taxiGreen$Pickup\_longitude,taxiGreen$Pickup\_latitude))

PickupArea@proj4string <- nycjson@proj4string

pickupBoroughCodes<-PickupArea %over% nycZone

summary(pickupBoroughCodes)

taxiGreen$boroughCode\_p <- pickupBoroughCodes$BoroName

#taxiGreen$boroughCode<-pickupBoroughCodes$name

DropoffPts<-SpatialPoints(cbind(taxiGreen$Dropoff\_longitude,taxiGreen$Dropoff\_latitude))

DropoffPts@proj4string <- nycjson@proj4string

dropoffBoroughCodes<-DropoffPts %over% nycZone

taxiGreen$boroughCodeDrop<-dropoffBoroughCodes$BoroName

taxiGreen$month <- month(taxiGreen$Pickup\_datetime)

taxiGreen$wday <- wday(taxiGreen$Pickup\_datetime)

taxiGreen$hour <- hour(taxiGreen$Pickup\_datetime)

# Change the format of datetime from string to POSIXct objects

taxiGreen$Pickup\_datetime <- as.POSIXct(taxiGreen$Pickup\_datetime,format='%Y-%m-%d %H:%M:%S')

taxiGreen$Dropoff\_datetime <- as.POSIXct(taxiGreen$Dropoff\_datetime,format='%Y-%m-%d %H:%M:%S')

taxiGreen$duration <- floor(as.double(taxiGreen$Dropoff\_datetime-taxiGreen$Pickup\_datetime)/60.0)

taxiGreen$percent <- taxiGreen$Tip\_amount/taxiGreen$Fare\_amount \* 100.0

taxiGreen$speed <- taxiGreen$Trip\_distance/taxiGreen$duration \*60

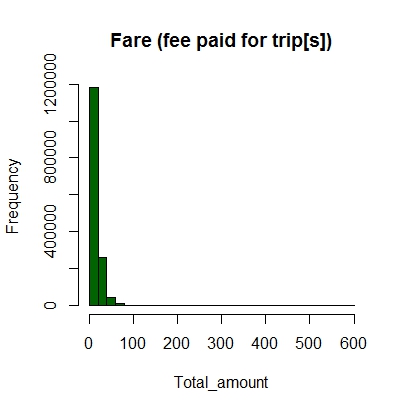
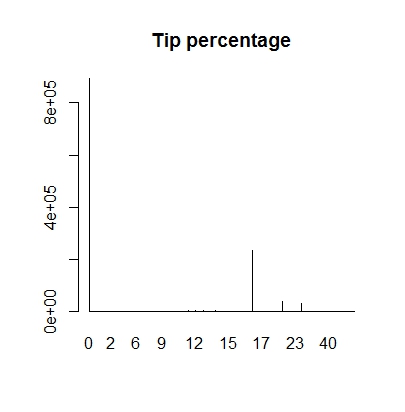
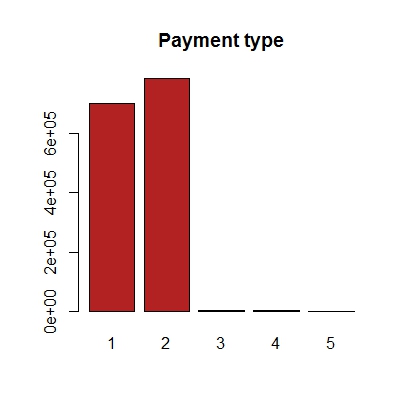
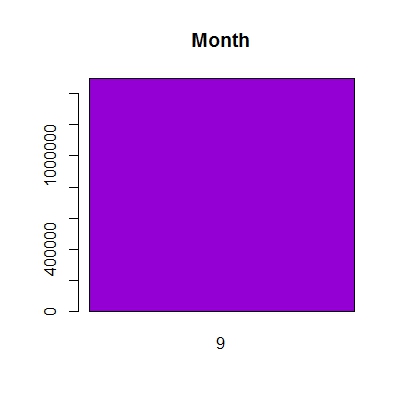
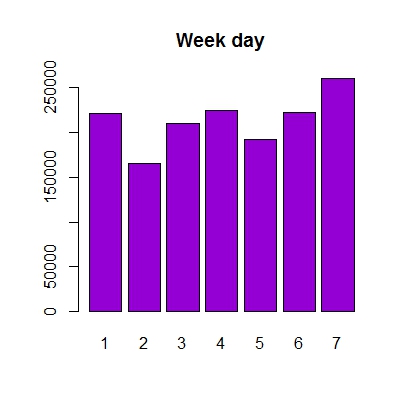
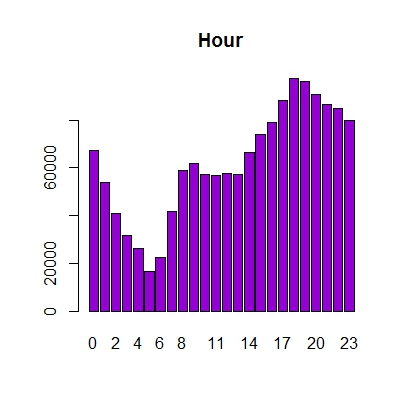
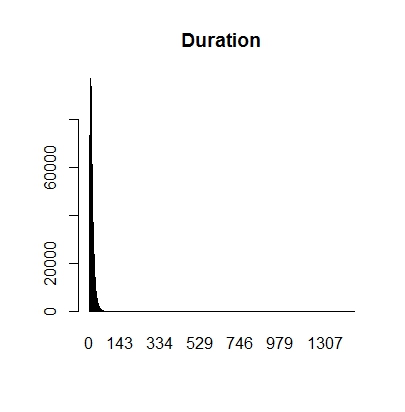
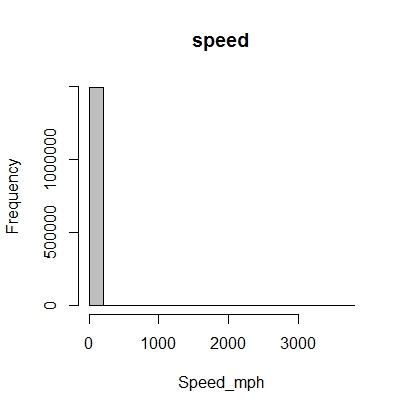
taxiGreen$speed[taxiGreen$speed=="NaN"]<-0

# usage in Tableau for visualization

write.csv(taxiGreen, file = "greenLocationBoro.csv", row.names = TRUE)

# Appendix B

## Additional Figures

****