NYC Taxi Project

Tejas Bawaskar

tejasbawaskar@gmail.com

NYC Taxis Tip Prediction

Background:

I was taking a cab on my way to my house from the airport, being unaware that tipping was customary here in the US, I initially refused to pay the driver a tip. After much contemplation, I gave him a dollar on $59 fare amount. I realized this was not correct and was rude on my behalf. The culture here is completely different from where I grew up. After having several conversations with cab drivers, I came to realize that they usually aren't tipped well even after giving a perfect ride to riders.

Proposal:

I want to help cab drivers, earn a bit more on tips for their hard work and time spent driving. Hence, I started working on a dataset for NYC taxi drivers that would help them analyze and predict how much tip they would receive from drivers given their status (such as location, time of day, day of the week, distance, time traveled etc.).

```
df <- read.csv("C:/Users/tejas/Documents/NYC/nyc_taxi.csv")

nrow(df) #number of rows in the dataset

## [1] 1494926

ncol(df) #number of columns in the dataset

## [1] 21
```

```
colnames(df) #column names

##  [1] "VendorID"              "lpep_pickup_datetime"
##  [3] "Lpep_dropoff_datetime" "Store_and_fwd_flag"
##  [5] "RateCodeID"            "Pickup_longitude"
##  [7] "Pickup_latitude"       "Dropoff_longitude"
##  [9] "Dropoff_latitude"      "Passenger_count"
## [11] "Trip_distance"         "Fare_amount"
## [13] "Extra"                 "MTA_tax"
## [15] "Tip_amount"            "Tolls_amount"
## [17] "Ehail_fee"             "improvement_surcharge"
## [19] "Total_amount"          "Payment_type"
## [21] "Trip_type"

summary(df) #summary of the dataset

##     VendorID              lpep_pickup_datetime
##  Min.   :1.000   2015-09-20 02:00:32:      9
##  1st Qu.:2.000   2015-09-05 14:57:48:      8
##  Median :2.000   2015-09-10 17:43:49:      8
##  Mean   :1.782   2015-09-13 00:27:28:      8
##  3rd Qu.:2.000   2015-09-13 01:06:29:      8
##  Max.   :2.000   2015-09-26 22:48:40:      8
##                  (Other)            :1494877
##          Lpep_dropoff_datetime Store_and_fwd_flag   RateCodeID
##  2015-09-28 00:00:00:    172   N:1486192          Min.   : 1.000
##  2015-09-13 00:00:00:    153   Y:   8734          1st Qu.: 1.000
##  2015-09-19 00:00:00:    141                      Median : 1.000
##  2015-09-14 00:00:00:    126                      Mean   : 1.098
##  2015-09-21 00:00:00:    125                      3rd Qu.: 1.000
##  2015-09-12 00:00:00:    119                      Max.   :99.000
##  (Other)            :1494090
##  Pickup_longitude Pickup_latitude Dropoff_longitude Dropoff_latitude
##  Min.   :-83.32   Min.   : 0.00   Min.   :-83.43    Min.   : 0.00
##  1st Qu.:-73.96   1st Qu.:40.70   1st Qu.:-73.97    1st Qu.:40.70
##  Median :-73.95   Median :40.75   Median :-73.95    Median :40.75
##  Mean   :-73.83   Mean   :40.69   Mean   :-73.84    Mean   :40.69
##  3rd Qu.:-73.92   3rd Qu.:40.80   3rd Qu.:-73.91    3rd Qu.:40.79
##  Max.   :  0.00   Max.   :43.18   Max.   :  0.00    Max.   :42.80
##
##  Passenger_count Trip_distance      Fare_amount          Extra
##  Min.   :0.000   Min.   :  0.000   Min.   :-475.00   Min.   :-1.0000
##  1st Qu.:1.000   1st Qu.:  1.100   1st Qu.:   6.50   1st Qu.: 0.0000
##  Median :1.000   Median :  1.980   Median :   9.50   Median : 0.5000
##  Mean   :1.371   Mean   :  2.968   Mean   :  12.54   Mean   : 0.3513
##  3rd Qu.:1.000   3rd Qu.:  3.740   3rd Qu.:  15.50   3rd Qu.: 0.5000
##  Max.   :9.000   Max.   :603.100   Max.   : 580.50   Max.   :12.0000
##
##     MTA_tax          Tip_amount       Tolls_amount      Ehail_fee
##  Min.   :-0.5000   Min.   :-50.000   Min.   :-15.2900   Mode:logical
##  1st Qu.: 0.5000   1st Qu.:  0.000   1st Qu.:  0.0000   NA's:1494926
##  Median : 0.5000   Median :  0.000   Median :  0.0000
##  Mean   : 0.4866   Mean   :  1.236   Mean   :  0.1231
##  3rd Qu.: 0.5000   3rd Qu.:  2.000   3rd Qu.:  0.0000
##  Max.   : 0.5000   Max.   :300.000   Max.   : 95.7500
##
##  improvement_surcharge  Total_amount      Payment_type     Trip_type
##  Min.   :-0.3000       Min.   :-475.00   Min.   :1.000   Min.   :1.000
##  1st Qu.: 0.3000       1st Qu.:   8.16   1st Qu.:1.000   1st Qu.:1.000
##  Median : 0.3000       Median :  11.76   Median :2.000   Median :1.000
##  Mean   : 0.2921       Mean   :  15.03   Mean   :1.541   Mean   :1.022
##  3rd Qu.: 0.3000       3rd Qu.:  18.30   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   : 0.3000       Max.   : 581.30   Max.   :5.000   Max.   :2.000
##                                                          NA's   :4
```

```
# checking for NA
colSums(is.na(df[,]))
```

```
##              VendorID  lpep_pickup_datetime Lpep_dropoff_datetime
##                     0                     0                     0
##    Store_and_fwd_flag            RateCodeID      Pickup_longitude
##                     0                     0                     0
##       Pickup_latitude     Dropoff_longitude      Dropoff_latitude
##                     0                     0                     0
##       Passenger_count         Trip_distance           Fare_amount
##                     0                     0                     0
##                 Extra               MTA_tax            Tip_amount
##                     0                     0                     0
##          Tolls_amount             Ehail_fee improvement_surcharge
##                     0               1494926                     0
##          Total_amount          Payment_type             Trip_type
##                     0                     0                     4

# We can see Ehail_fee has no data hence we will eliminate it
df <- subset(df,select = -c(Ehail_fee))

# We can also see that only 4 observations are missing in Trip_type of 1.49 million observations
df <- df[complete.cases(df),]

# Checking the datatypes of each variable in the dataframe
sapply(df, class)

##              VendorID  lpep_pickup_datetime Lpep_dropoff_datetime
##             "integer"              "factor"              "factor"
##    Store_and_fwd_flag            RateCodeID      Pickup_longitude
##              "factor"             "integer"             "numeric"
##       Pickup_latitude     Dropoff_longitude      Dropoff_latitude
##             "numeric"             "numeric"             "numeric"
##       Passenger_count         Trip_distance           Fare_amount
##             "integer"             "numeric"             "numeric"
##                 Extra               MTA_tax            Tip_amount
##             "numeric"             "numeric"             "numeric"
##          Tolls_amount improvement_surcharge          Total_amount
##             "numeric"             "numeric"             "numeric"
##          Payment_type             Trip_type
##             "integer"             "integer"
```

This gives us an understanding that the data types for ‘lpep_pickup_datetime’ and ‘Lpep_dropoff_datetime’ aren’t in the format that we want. Rest of the variables are fine.

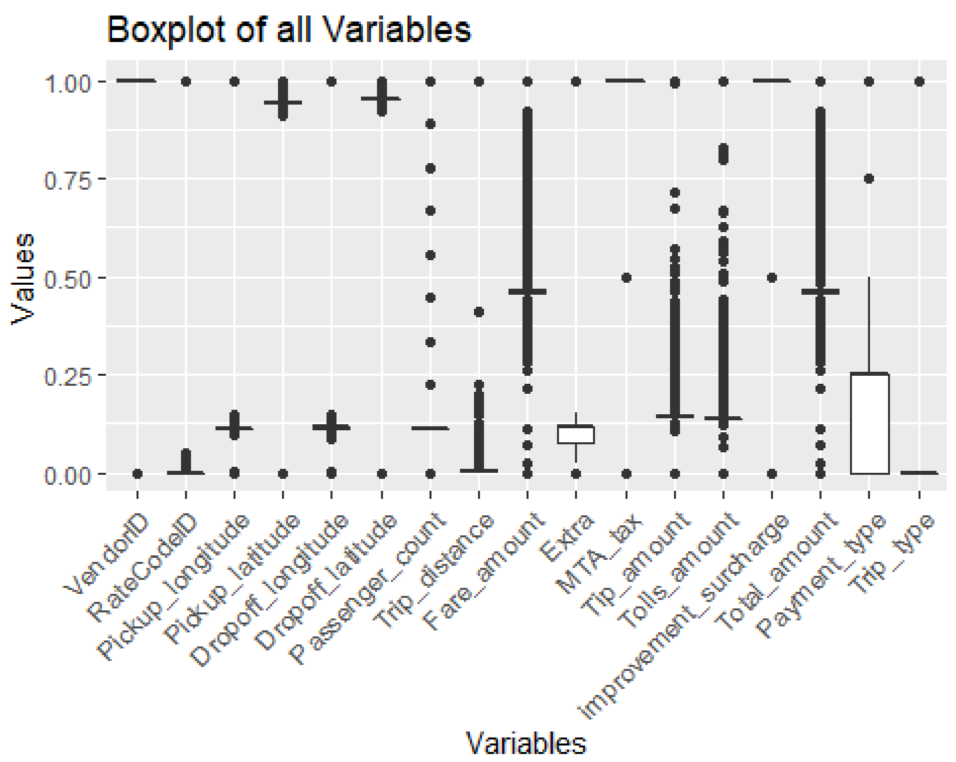
Data Preprocessing and Exploratory Data Analysis

Check for outliers

Since a normal boxplot without normalizing wasn’t visually easily to interpret with all the variables simultaneously. All the numerical variables are plotted after normalizing them between 0 and 1.

*The values were normalized between 0 and 1, as unscaled variables didn’t give a proper a visualization.

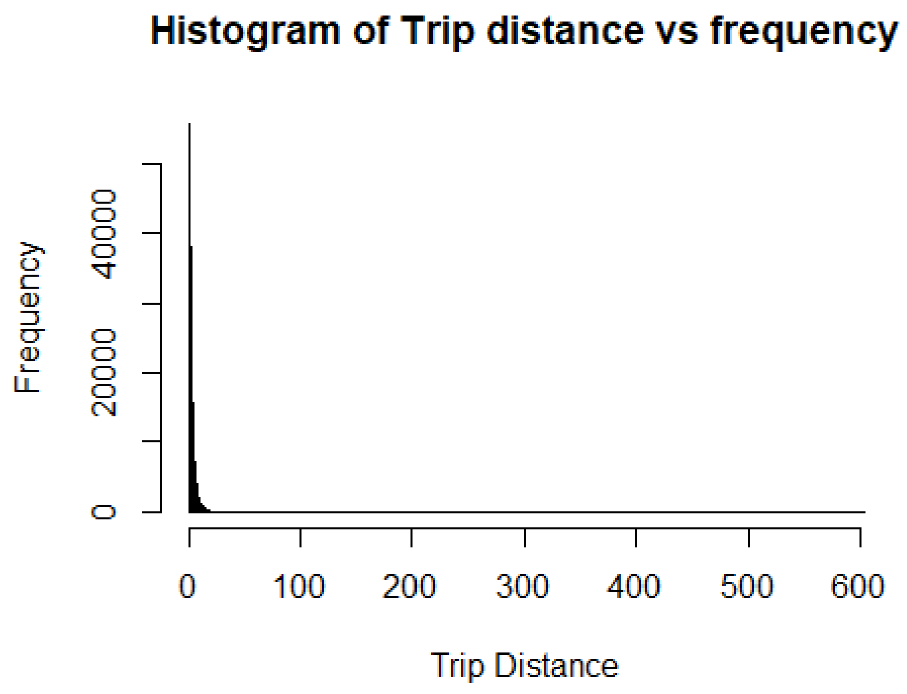
```
# variables were then scaled to better understand their behavior
nums <- sapply(df, is.numeric)
num.df <- df[ , nums]
norm.df <- normalize(num.df, method = 'range', range = c(0,1))
#boxplot(norm.df, names=colnames(norm.df), las = 2)
ggplot(stack(norm.df), aes(x = ind, y = values)) +
    geom_boxplot() + theme(axis.text.x =
                               element_text(size  = 10,
                                            angle = 45,
                                            hjust = 1,
                                            vjust = 1)) + labs (x = 'Variables',
                                                                y = 'Values',
                                                                title = 'Boxplot of all Variables')
```



This plot shows us that, there are a number of outliers for variables like, trip distance, fare amount...

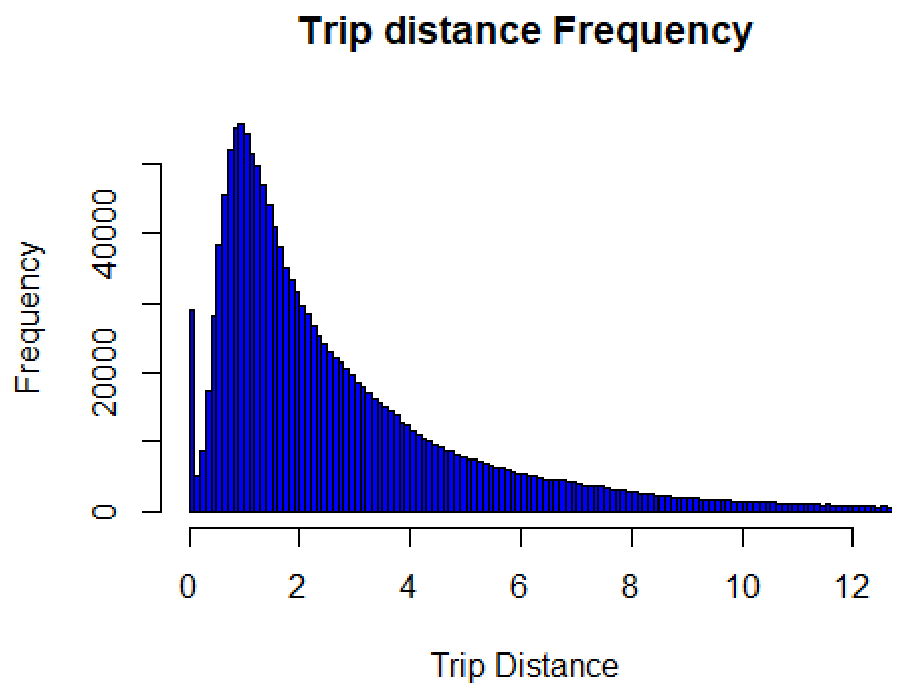
We will look into Trip Distance as it is the first variable that has a high number of outliers.

```
rm(num.df)
gc()

##              used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   4348485 232.3    7714147  412.0   6106474  326.2
## Vcells 102685788 783.5  426260836 3252.2 532109598 4059.7

#Lets look at Trip Distance as it shows a high percentage of outliers
hist(df$Trip_distance, breaks = 5000, col = 'black',xlab = 'Trip Distance'
     ,main = 'Histogram of Trip distance vs frequency')
```



Since the figure is quite vague, for visualization purposes, all the outliers are excluded by limiting the x values to x = mean + standard deviation*3 to give a clear idea about the distribution.

```
hist (df$Trip_distance
            ,breaks = 5000
            ,col = "blue"
            ,xlab = 'Trip Distance'
            ,main = 'Trip distance Frequency'
            ,xlim = c(0,mean(df$Trip_distance) + 3*sd(df$Trip_distance))
            ,freq = TRUE)
```



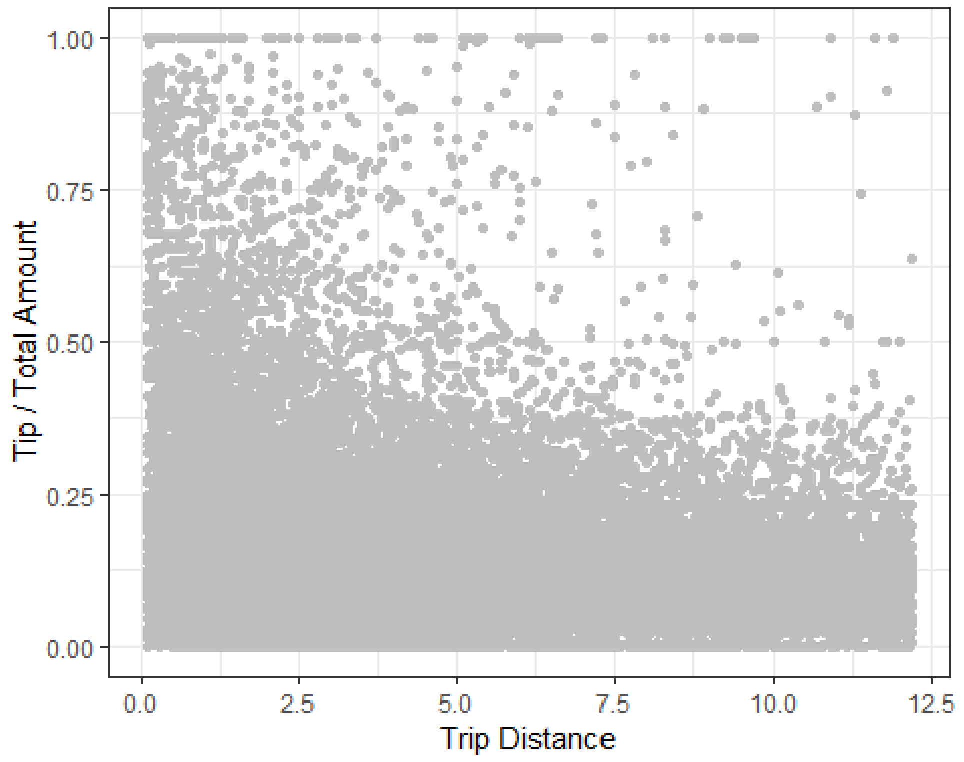
It can be inferred that the graph is non-normal i.e. right skewed and the mode < median < mean. Here the t-statistic will hold valid because of its robustness to deviations from normality. The skewness can be explained as:

1) The relation might be affected by the fact that people that travel long distances can't afford to pay the fare for a cab. Instead they use the public transportations system such as bus, metro etc. People who go to work on a daily basis would prefer a cheaper option.

2) Moreover the cab drivers wouldn't want to travel to a remote place from where they wouldn't get a return fare.

*It can be pointed out that there is an anomaly seen in the graph. There are number of rides where the distance covered is zero which is incorrect. We thus need to clean the data through for better predictions/results.

Plot - Trip Distance v/s Tip/Total amt.



From the graph, we notice that a good percentage of tip to total amt ratio is high in case of short trips. Drivers are better of driving short distances to let the tips keep coming.

Another anamoly is how tips match the total amount paid.

Airport trips - According to the data dictionary, it is mentioned that the rate code ids are

1 = Standard rate

2 = JFK

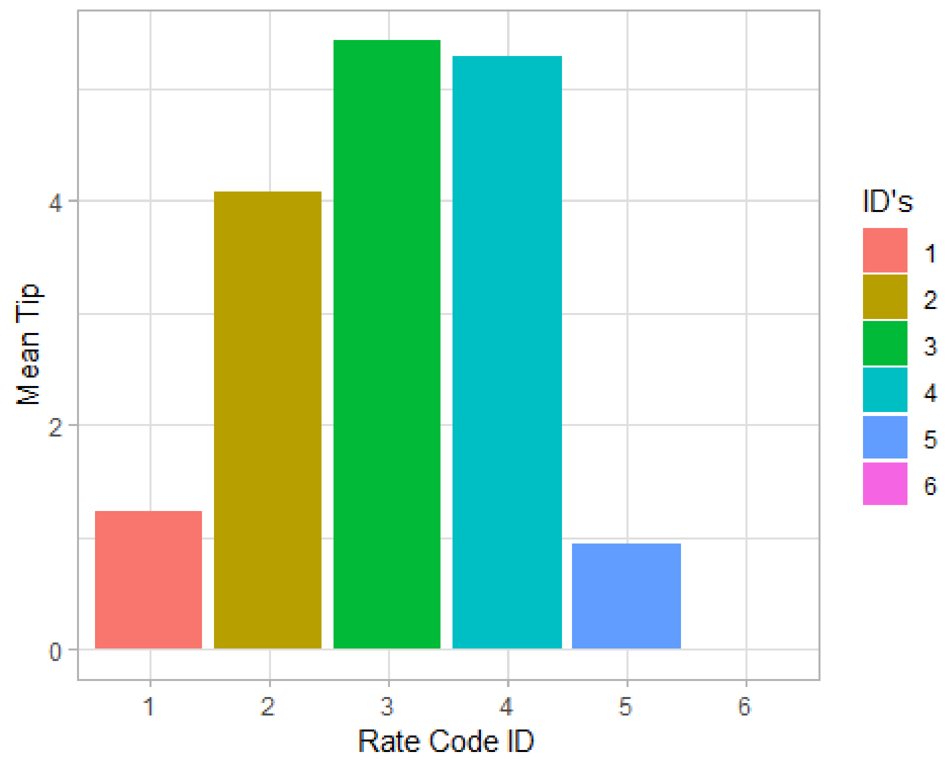
3 = Newark

4 = Westchester

5 = Negotiated fare

6 = Group ride

```
plyr::count(df, 'RateCodeID')

##   RateCodeID    freq
## 1          1 1454464
## 2          2    4435
## 3          3    1117
## 4          4     925
## 5          5   33943
## 6          6      36
## 7         99       2

# Upon analysis it was found that ratecode id 99 was an error
df <- subset(df,df$RateCodeID < 7)

ggplot(df, aes(y = Tip_amount, x = factor(RateCodeID), fill = factor(RateCodeID))) +
  theme_light() +
  stat_summary(fun.y = mean, # calc mean of all observations for the month
               geom = "bar") +
  scale_color_manual('') +
  labs(x = 'Rate Code ID', y = 'Mean Tip', fill = "ID's")
```



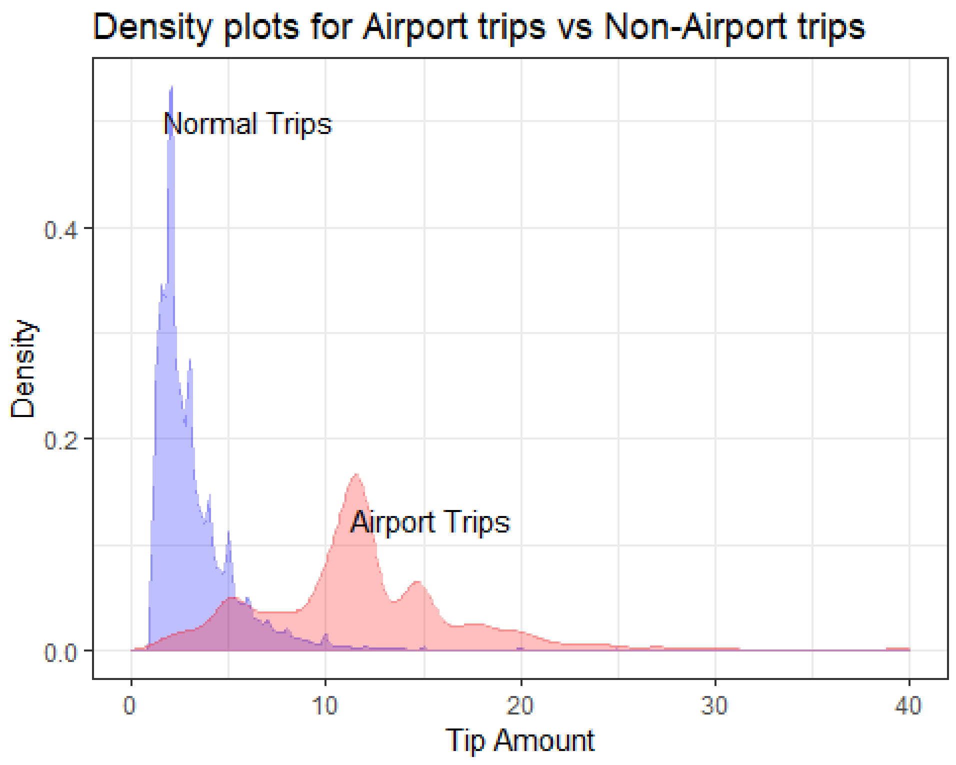
*The graph plotted above is a filtered dataset with observations that have payment type as Credit Card. Later, it was revealled that other payment types had an anamoly that needs to be fixed.

This plot shows the average tip recieved from differnt rate code ID's. This gives us a better idea of how drivers can earn a bit more than their usual earnings. Tips recieved from airports are nearly as 10x their standard tip. This could be accounted for a number of reasons

1) The local taxes are levied on riders for catching a cab from the airport, thus contributing to the total amount and subsequently higher tips.

2) Tourists who come to visit, might be unaware of how much to tip the driver, driving the mean tip even higher.

Lets look at the density plots for airport trips vs normal trips. And try to find any significance difference between them.



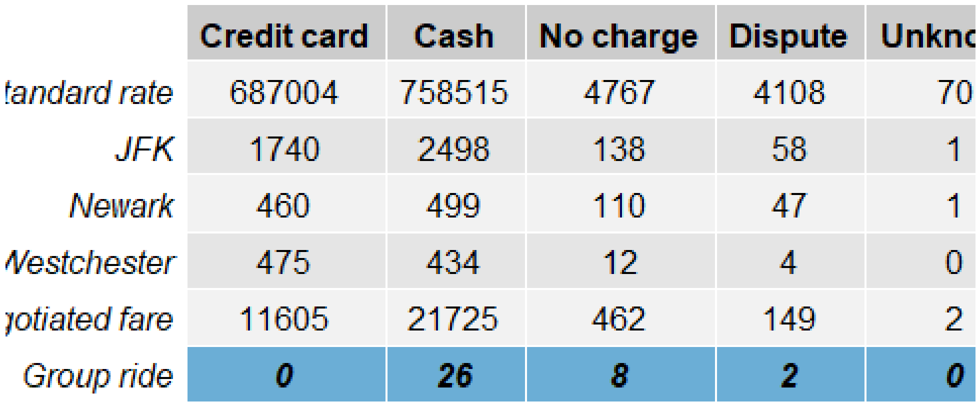
Looking at the plots of the individual density plots it becomes clear that the data is highly skewed. To compare the average/mean tips per ride type it makes sense to use a non-parametric method. Although there has been research done on parametric methods that say that the Welch Two Sample t-test for independent variables with different standard deviations is insensitive to the distribution if a min of 30 (thumb rule, could be 50 as well) data points are available. We have sufficient datapoints to perform this test which would be independent of the of spread of the data. To conclude Welch tests can be performed on skewed and non-normal data when the min. number of points is 30

```
t.test(df[((df$RateCodeID == 1) | (df$RateCodeID == 5) | (df$RateCodeID == 6)),]$Tip_amount,
            df[((df$RateCodeID == 2) | (df$RateCodeID == 3) | (df$RateCodeID == 4)),]$Tip_amount, var.equal
= F)

##
##  Welch Two Sample t-test
##
## data:  df[((df$RateCodeID == 1) | (df$RateCodeID == 5) | (df$RateCodeID ==  and df[((df$RateCodeID == 2)
| (df$RateCodeID == 3) | (df$RateCodeID ==     6)), ]$Tip_amount and     4)), ]$Tip_amount
## t = -31.302, df = 6480.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.469428 -3.060481
## sample estimates:
## mean of x mean of y
##  1.221586  4.486540
```

With a p-value less than 0.05 we can reject the null hypothesis that the means are equal (or the difference is insignificant) and conclude that the difference in means are indeed significant for different ride types.

Going back to the rate code id's. Rate code ID 6 though showing some distance travlled, is showing almost zero tip recieved. ID 6 corresponds to group ride. Let's dive into it a bit, by looking the type of payment for each rate code ID. First

```
temp <- with(df,table(df$RateCodeID, df$Payment_type))
rownames(temp) <- c('Standard rate','JFK','Newark','Westchester','Negotiated fare','Group ride')
colnames(temp) <- c('Credit card','Cash','No charge','Dispute','Unknown')

t1 <- ttheme_default(core=list(
  fg_params=list(fontface=c(rep("plain", 5), "bold.italic")),
  bg_params = list(fill=c(rep(c("grey95", "grey90"),
                              length.out=5), "#6BAED6"),
                   alpha = rep(c(1,0.5), each=10))
))

## Cross table of the count of rides w.r.t the payment type and rate code id.
grid.table(temp, theme = t1)
```

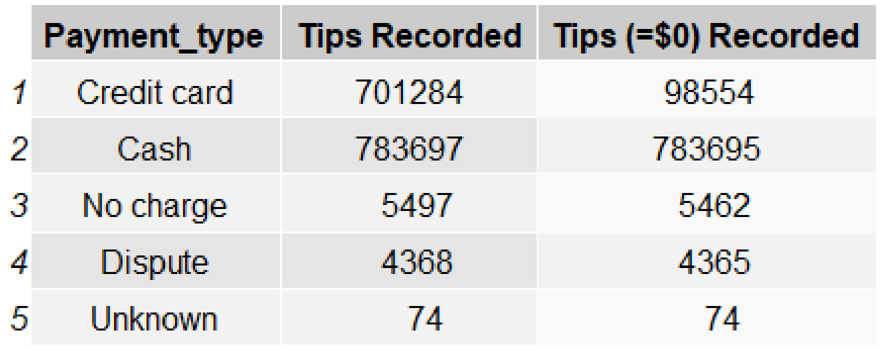


	Credit card	Cash	No charge	Dispute	Unknc
tandard rate	687004	758515	4767	4108	70
JFK	1740	2498	138	58	1
Newark	460	499	110	47	1
Vestchester	475	434	12	4	0
gotiated fare	11605	21725	462	149	2
Group ride	***0***	***26***	***8***	***2***	***0***

Since a majority of the payment done in group rides (rate id 6) is done by cash, the tips aren't recorded.

Upon further analysis, I found that, the tip amount recorded is zero for all cases but one. All the transactions which are not electronic have zero tips.

```
temp1 <- plyr::count(df, vars = 'Payment_type')
temp2 <- plyr::count(df[df$Tip_amount == 0,], vars = 'Payment_type')

temp <- merge(temp1,temp2,'Payment_type')
colnames(temp) = c('Payment_type','Tips Recorded', 'Tips (=$0) Recorded')
temp$Payment_type <- c('Credit card','Cash','No charge','Dispute','Unknown')

grid.table(temp, theme=t1)
```



	Payment_type	Tips Recorded	Tips (=$0) Recorded
1	Credit card	701284	98554
2	Cash	783697	783695
3	No charge	5497	5462
4	Dispute	4368	4365
5	Unknown	74	74

It is understood that if passengers pay by cash, their tips won't be recorded by drivers so as to avoid taxes. Ironically, all the other types of trips also have zero tips. It would be sensible to remove those points as they will have no good impact on our predictions.

From the data provided, we can derive a few variables of our own.

```
#Converting Trip Duratians to secs
x1 <- strptime(df$lpep_pickup_datetime, "%Y-%m-%d %H:%M:%OS")
x2 <- strptime(df$Lpep_dropoff_datetime, "%Y-%m-%d %H:%M:%OS")
df$trip_duration <- as.numeric(x2-x1,units="secs") #this is a derived feature

#dividing into hours
time.category <- with(df, ifelse(trip_duration <= (4*3600), 1,
                                 ifelse(trip_duration >= 5*3600 & trip_duration <= 24*3600, 2, 3))
)
aggregate(df$trip_duration,by=list(time.category),FUN=length)

##   Group.1       x
## 1       1 1486351
## 2       2    8382
## 3       3     187

gc()

##             used  (Mb) gc trigger (Mb)  max used   (Mb)
## Ncells  4391753 234.6    7714147  412   7714147  412.0
## Vcells 96729841 738.0  341175089 2603 532109598 4059.7
```

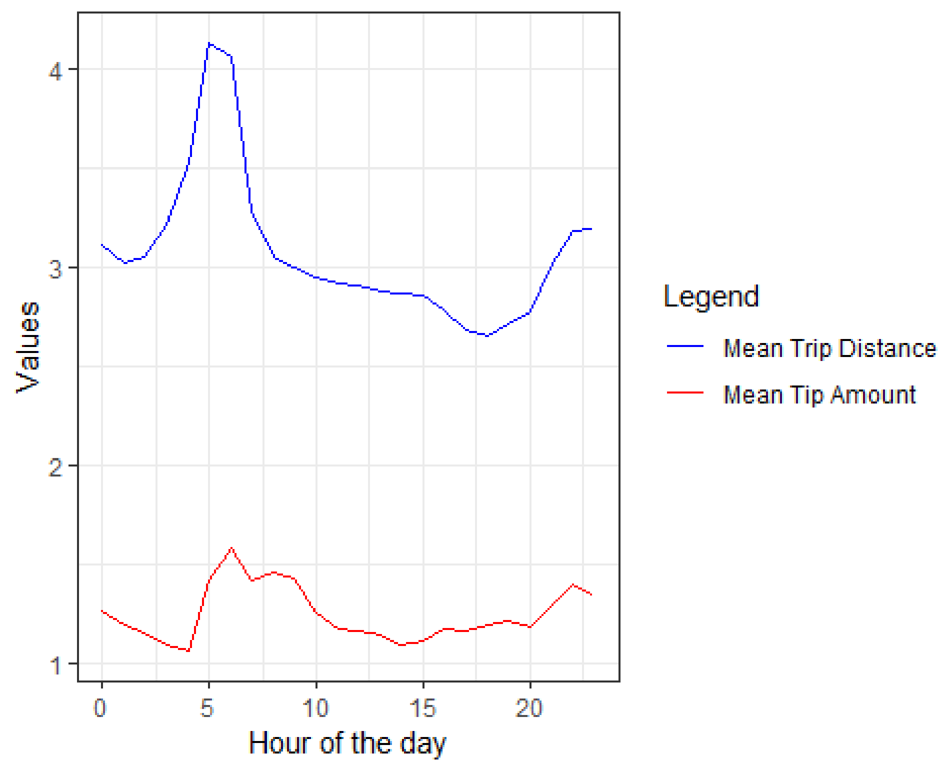
As can be seen 8382 rides are between 5 hrs and 24 hrs and 187 rides are above 24 hours. These are obviously outliers as passengers don't travel usually for more than 2 hours, as an exception I will consider 4 hours to be the upper limit while building my model.

Lets see if the time of the day has any impact on the distance covered.

```
df$hour <- as.POSIXlt(anytime(as.factor(df$lpep_pickup_datetime)))$hour
temp <- aggregate(.~ hour, data = df,mean)[,c('hour','Trip_distance','Tip_amount')]
temp
```

```
##    hour Trip_distance Tip_amount
## 1     0      3.115276   1.260581
## 2     1      3.017347   1.189363
## 3     2      3.046176   1.149693
## 4     3      3.212945   1.089771
## 5     4      3.526555   1.058802
## 6     5      4.133722   1.414896
## 7     6      4.055686   1.579561
## 8     7      3.284394   1.411359
## 9     8      3.048450   1.454360
## 10    9      2.999105   1.421786
## 11   10      2.944482   1.250880
## 12   11      2.912015   1.172081
## 13   12      2.903115   1.163272
## 14   13      2.878294   1.136609
## 15   14      2.864304   1.090176
## 16   15      2.857040   1.109829
## 17   16      2.779852   1.170272
## 18   17      2.679114   1.163597
## 19   18      2.653222   1.186836
## 20   19      2.715597   1.212864
## 21   20      2.777052   1.177906
## 22   21      2.999223   1.286050
## 23   22      3.185394   1.394593
## 24   23      3.191538   1.343764
```

```r
ggplot(data = temp, aes(x = hour)) +
  geom_line(aes(y = Trip_distance, colour = 'Mean Trip Distance')) +
  geom_line(aes(y = Tip_amount, colour = 'Mean Tip Amount')) +
  theme_bw() +
  labs(x = "Hour of the day", y = 'Values') +
  scale_colour_manual('Legend ',breaks = c("Mean Trip Distance", "Mean Tip Amount")
                       ,values=c('Mean Trip Distance'="blue",'Mean Tip Amount'="red"))
```



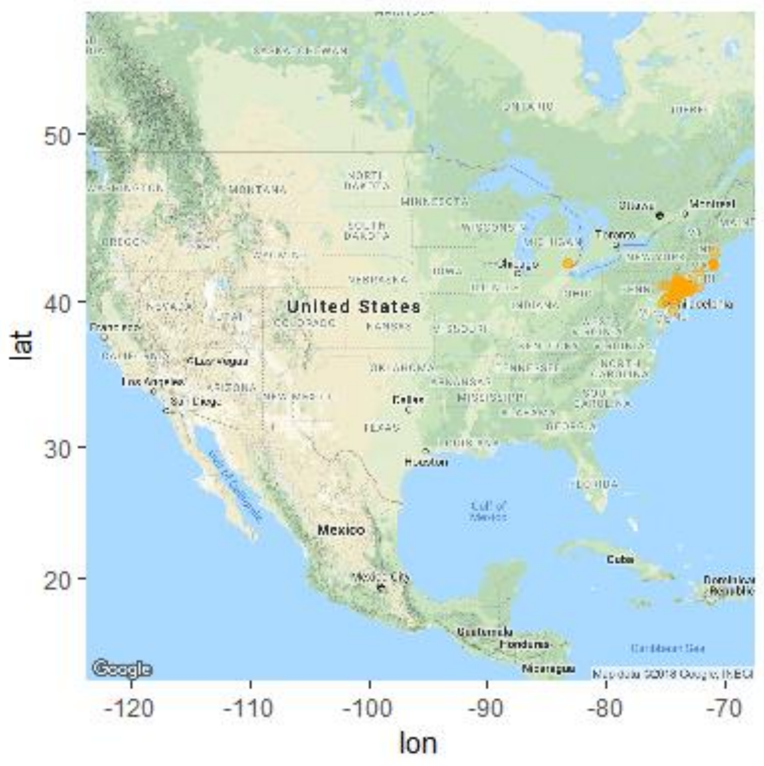
An interesting observation can be interpreted here.

The distance covered here during the 5th and 6th hour are highest along with the tip recieved. This could be true as people travelling early in the morning to work would need to get on time. While coming back from work they wouldn't mind cathing a local transportation system. Also subsequently it reduces through out the day utill it reaches late night till the morning. I believe this would be high mainly beacuse of the number of rides seen on weekends, which definetly would have an impact on these late night rides.

```
gc()
```

```
##             used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   4398138 234.9    7714147  412.0   7714147  412.0
## Vcells  88586238 675.9  272940071 2082.4 532109598 4059.7
```

```
#latitude and longtitude other than the area covered by Green Taxis
temp <- data.frame(df$Pickup_longitude, df$Pickup_latitude)
colnames(temp) = c('lon','lat')

usa_center = as.numeric(geocode("United States"))

## Information from URL :
http://maps.googleapis.com/maps/api/geocode/json?address=United%20States&sensor=false

USAMap = ggmap(get_googlemap(center=usa_center, scale=2, zoom=4), extent="panel")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=37.09024,-
95.712891&zoom=4&size=640x640&scale=2&maptype=terrain&sensor=false

USAMap +
  geom_point(aes(x=lon, y=lat), data=temp, col="orange", alpha=0.4)
```



From this visualization it is clear that some of the coordinates fall out of the actual area of service. These coordinates need to be scrapped off and for future purposes these also need to be investigated. While plotting, it was noticed that 1985 observations had coordinates outside of USA. Interesting?

Coordinates outside of area of service (bounding box) are set to NA (reference taken from https://www.maptechnica.com/city-map/New%20York/NY/3651000). According to this, the bounding box limits are latitude=40.917577, longitude=-74.259090 at the northwest corner, and latitude=40.477399, longitude=-73.700272 at the southeast corner

```
ndf <- data.frame(Dropoff_longitude=df$Dropoff_longitude
                 ,Dropoff_latitude=df$Dropoff_latitude
                 ,Pickup_longitude=df$Pickup_longitude
                 ,Pickup_latitude=df$Pickup_latitude)
```

```r
nw <- list(lat = 40.917577, lon = -74.259090)
se <- list(lat = 40.477399, lon = -73.700272)

ind <- which(df$Dropoff_longitude < nw$lon | df$Dropoff_longitude > se$lon)
ndf$Dropoff_longitude[ind] <- NA

ind <- which(df$Pickup_longitude < nw$lon | df$Pickup_longitude > se$lon)
ndf$Pickup_longitude[ind] <- NA

ind <- which(df$Dropoff_latitude < se$lat | df$Dropoff_latitude > nw$lat)
ndf$Dropoff_latitude[ind] <- NA

ind <- which(df$Pickup_latitude < se$lat | df$Pickup_latitude > nw$lat)
ndf$Pickup_latitude[ind] <- NA

na_count <- sapply(ndf, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)
na_count

##                     na_count
## Dropoff_longitude     2972
## Dropoff_latitude      3223
## Pickup_longitude      2312
## Pickup_latitude       2358
```

As it can be seen there are latitudes and longtitudes outside of the area of service.

```r
# 0 passenger count
nrow(df[df$Passenger_count == 0,])

## [1] 436
```

Here we notice that the passenger count in some of these rides were zero. This is obviously incorrect, but instead of deleting these rows, it would be better to replace 0 with the median number of passengers, since there are fares recorded with those rides and tips to. Also, the median was chosen as the its histogram was skewed.

```r
# More than 7 passenger count
nrow(df[df$Passenger_count > 8,])
```

```
## [1] 16
```

Another observation was that since the number of max passengers that can travel is only 7, we can exclude those rows that have more than 7 passengers

Also there were a few more anomalies that were observed which needed to be cleaned up. After reviewing the rates (from "http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml"), it was observed that the minimum fare for a ride is $2.5. From the data it can be observed that some of the rides were less than the minimum amount.

```
# Fare Amount less than 2.5
nrow(df[df$Fare_amount < 2.50,])

## [1] 7455
```

7455 rows have fare amount < 2.5 which can't be possible.

Looking at a bit deeper, fare amount from different vendors, Creative Mobile Technologies and VeriFone Inc.

```
nrow(df[df$Fare_amount < 0 & df$VendorID == 2,])

## [1] 2417

nrow(df[df$Fare_amount < 0 & df$VendorID == 1,])

## [1] 0
```

I see that vendor ID 2 has a an issue recording fares at times. All the fares that have values less than 0 (wrongly recorded negative values) are derived only from the second vendor (i.e. Vendor ID = 2, VeriFone Inc). This should be further investigated, to avoid loss and erors in data. As of now, I'll convert all the negative data to positive to avoid loss in data.

```
neg.vars <- c('Fare_amount','Extra','improvement_surcharge','Total_amount','MTA_tax','Tip_amount')

df[df$Fare_amount < 0,][neg.vars] <- df[df$Fare_amount < 0,][neg.vars]*-1

#Removing Fare amount less than 2.5
df <- subset(df,df[,'Fare_amount'] >= 2.5)

# Distances greater then 0
```

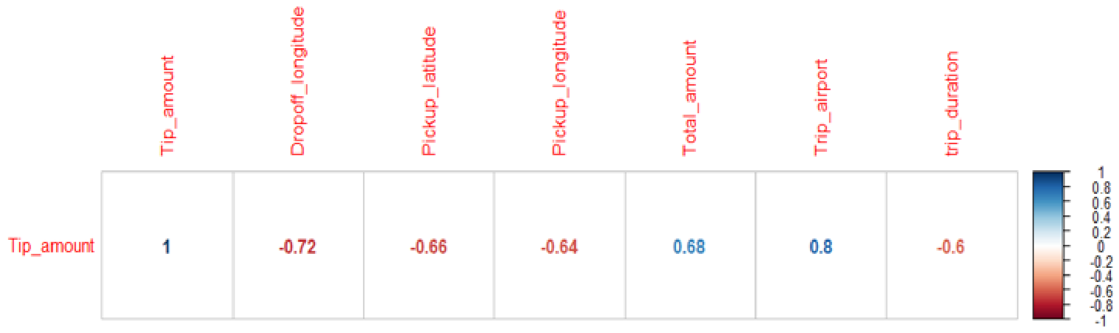
```
df <- subset(df,df[,11] > 0)
nrow(df)

## [1] 1470312

#Trip Durations greater then 4 hrs
df <- subset(df,df[,21] < (4*3600))

#remove trip Durations less then 2min
df <- subset(df,df[,21] > (2*60))
nrow(df)

## [1] 1431210

# set coordinates outside of NYC bounding box to NA(reference taken from
# https://www.maptechnica.com/city-map/New%20York/NY/3651000)
nw <- list(lat = 40.917577, lon = -74.259090)
se <- list(lat = 40.477399, lon = -73.700272)
ind <- which(df$Dropoff_longitude < nw$lon | df$Dropoff_longitude > se$lon)
df$Dropoff_longitude[ind] <- NA
ind <- which(df$Pickup_longitude < nw$lon | df$Pickup_longitude > se$lon)
df$Pickup_longitude[ind] <- NA
ind <- which(df$Dropoff_latitude < se$lat | df$Dropoff_latitude > nw$lat)
df$Dropoff_latitude[ind] <- NA
ind <- which(df$Pickup_latitude < se$lat | df$Pickup_latitude > nw$lat)
df$Pickup_latitude[ind] <- NA
nrow(df)

## [1] 1431210

df <- df[complete.cases(df),]
nrow(df)

## [1] 1427671

# passengers < 7
df <- subset(df,df[,10] < 7)
```

```r
# Replace passengers with zero count with the median value i.e. 1
df$Passenger_count[df$Passenger_count == 0] <- 1

# Since payment types other than credit card have zero tips 99% of the time
df <- subset(df, Payment_type == 1)

# Convert to non-airport and airport trips
df[df$RateCodeID==1|df$RateCodeID==5|df$RateCodeID==6,]$RateCodeID <- 0
df[df$RateCodeID==2|df$RateCodeID==3|df$RateCodeID==4,]$RateCodeID <- 1
```

Collinearity plot

Looking at relations among variables

```r
nums <- sapply(df, is.numeric) #taking only numeric class
num.df <- df[ , nums]
gc()
#display only tip_percent
corrplot(cor(num.df[,-16])[,18], method = "number",tl.cex = 1,type="lower",diag=FALSE)
```



Relations

According to this correlation plot,

Tips

Negatively correlated:

1) Dropoff Longitude

2) Pickup Longitude

3) Pickup Latitude

4) Trip Duration

Positively correlated:

1) Total Amount

2) Airport Trip

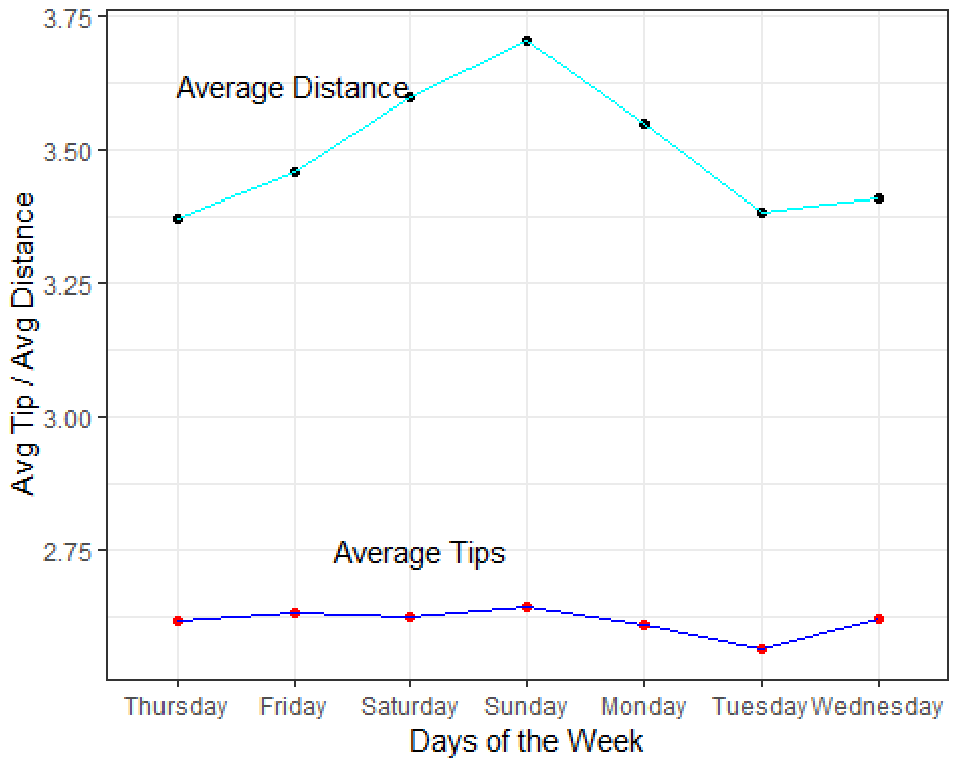
Feature engineering: new derived features

```
df$Tip_percent <- (df$Tip_amount/df$Total_amount)*100

clean_datetime <- df %>%
  mutate(lpep_pickup_datetime = ymd_hms(lpep_pickup_datetime)) %>%
  mutate(Lpep_dropoff_datetime = ymd_hms(Lpep_dropoff_datetime)) %>%
  mutate(weekday_pickup = weekdays(lpep_pickup_datetime)) %>%
  mutate(weekday_dropoff= weekdays(Lpep_dropoff_datetime))%>%
  mutate(hpick = hour(lpep_pickup_datetime)) %>%
  mutate(date1 = date(lpep_pickup_datetime))

#from the above code we get derived features such as weekday pickup,hour of pickup
```

```r
temp <- clean_datetime %>%
  group_by(weekday_pickup) %>%
  summarize(Count_Trips = n(), avg_dist = mean(Trip_distance),
            avg_passengers = mean(Passenger_count),
            avg_price = mean(Total_amount),
            avg_Tip = mean(Tip_amount),
            Total_tip = sum(Tip_amount))

temp[c(2,5,3,4,1,6,7),] <- temp[c(1,2,3,4,5,6,7),]
temp$ratio <- temp$avg_Tip/temp$avg_dist

head(temp,7)
```

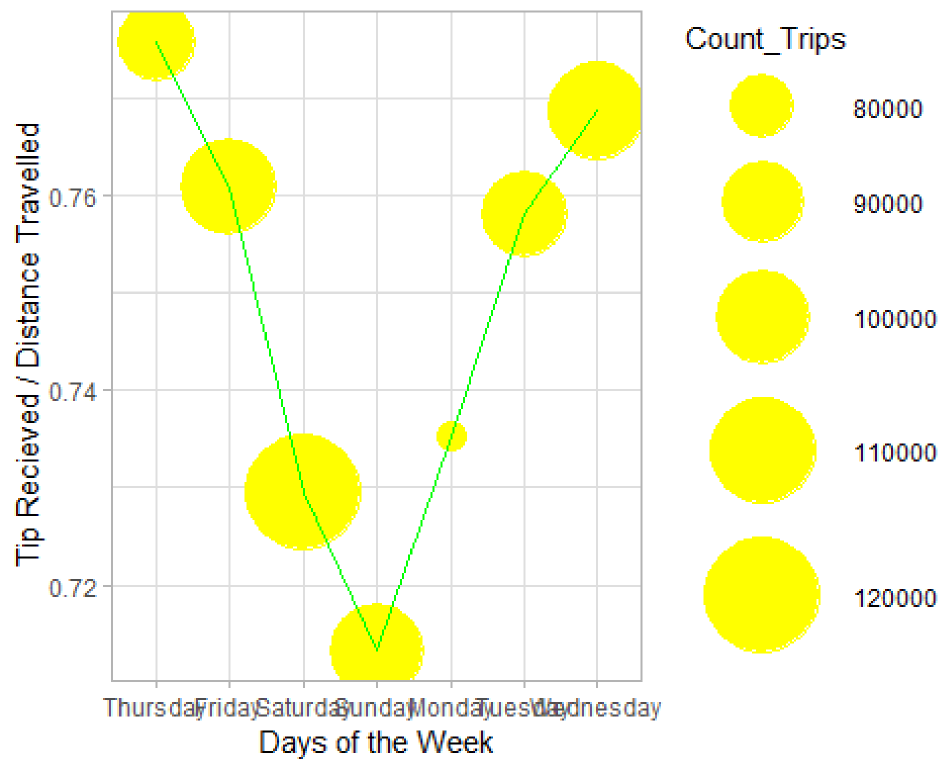
```
## # A tibble: 7 x 8
##   weekday_pickup Count_Trips avg_dist avg_passengers avg_price avg_Tip
##   <chr>                <int>    <dbl>          <dbl>     <dbl>   <dbl>
## 1 Thursday             88274     3.37           1.35      18.1    2.61
## 2 Friday              101211     3.46           1.36      18.3    2.63
## 3 Saturday            120402     3.60           1.40      18.1    2.62
## 4 Sunday               99571     3.70           1.40      18.1    2.64
## 5 Monday               72832     3.55           1.36      18.0    2.61
## 6 Tuesday              93683     3.38           1.35      17.8    2.56
## 7 Wednesday           102930     3.41           1.35      18.1    2.62
## # ... with 2 more variables: Total_tip <dbl>, ratio <dbl>
```

```r
ggplot(temp, aes(x = factor(weekday_pickup),group = 1)) +
  geom_point(aes(y = avg_Tip), color = 'red') +
  geom_line(aes(y = avg_Tip), color = 'blue') +
  geom_point(aes(y = avg_dist), color = 'black') +
  geom_line(aes(y = avg_dist), color = 'cyan') +
  theme_bw() +
  xlab('Days of the Week') +
  ylab('Avg Tip / Avg Distance') +
  annotate("text", x = 2.0, y = 3.62, label = "Average Distance") +
  annotate("text", x = 3.1, y = 2.75, label = "Average Tips") +
  scale_x_discrete(limits=temp$weekday_pickup)
```



From the above graph and table above we notice The maximum distance was on Sunday and the maximum avg tip/ total tip amount recieved was on Saturday followed by Friday. There's a straight dip in tips on Monday. This can be accounted as people like to go out on weekends and won't hesitate to spend a little more, on the other hand they wouldn't do the same on a weekday. Let's compare the ratio's of the distance to tip.

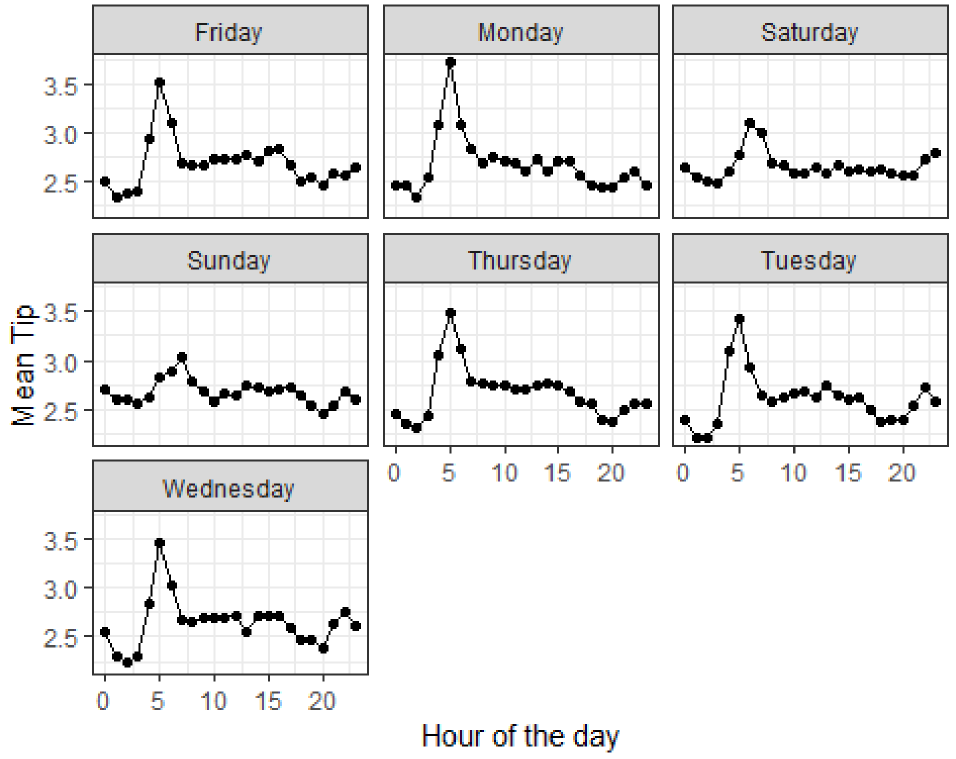
```
p1 <- ggplot(temp, aes(x = weekday_pickup
                 ,group = 1)) +
  geom_point(aes(y = ratio, size = Count_Trips), color = 'yellow') +
  geom_line(aes(y = ratio), color = 'green') +
  theme_light() +
  xlab('Days of the Week') +
  ylab('Tip Recieved / Distance Travelled') +
  scale_x_discrete(limits=temp$weekday_pickup) +
  scale_size_continuous(range = c(5,20)) +
 theme(legend.position="right")

p1
```



There's a massive dip on Sunday's, which drivers might be unaware of. They don't get a good return on their trips on Sunday. The intensity of each point describes the average number of rides given at that day. From the graph, we see that Thursday and Wednesday have the best ratio's, inturn giving the best returns, although it doesn't graph doesn't show us the total tips recieved that day. If one looks to save time and wants the best on its retun, they should work through the weekdays (Wednesday, Thursday); else, if the driver has time and wants to earn a bit more, it would be smarter to work on weekends as the ratio difference betwwen Thurs and Sunday isn't that high.

Combining this knowledge recieved and our our previous plot on tip recieved by the hour, we can get a detailed explaination on when and what time can one maximize their tips

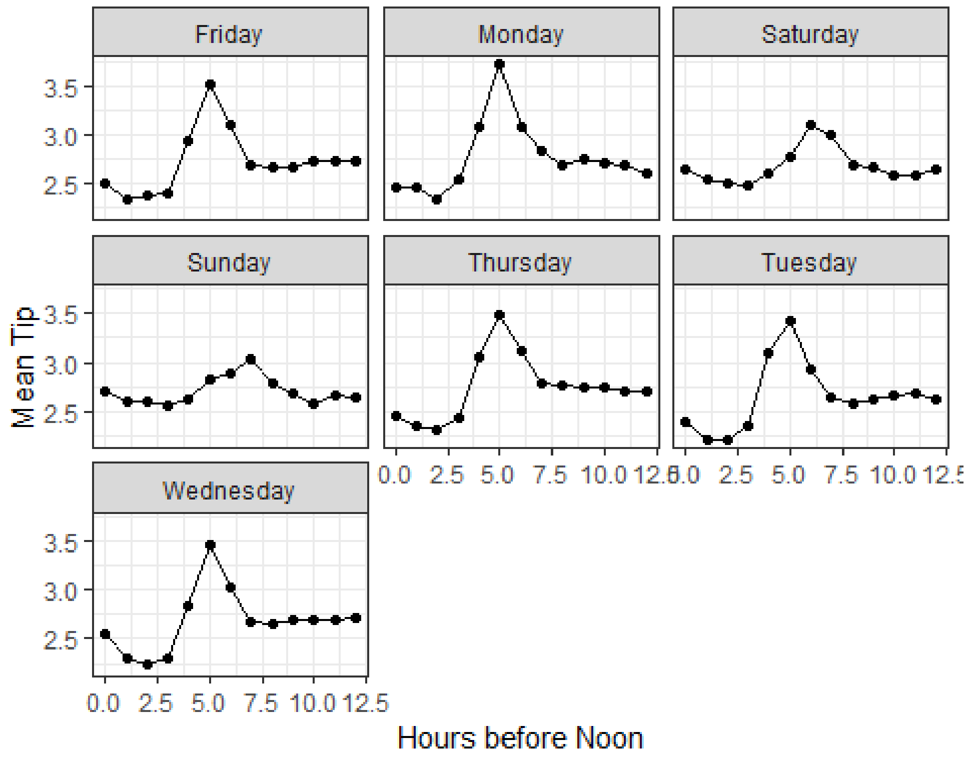
```
temp <- clean_datetime %>%
  group_by(hour,weekday_pickup) %>%
  summarise(Mean_Tip = mean(Tip_amount))

ggplot(temp, aes(hour, Mean_Tip)) +
  geom_point() +
  geom_line() +
  theme_bw() +
  facet_wrap(~factor(weekday_pickup)) +
  labs(x = 'Hour of the day', y = 'Mean Tip')
```



A closer look into these plots

Tips recieved before Noon during the week

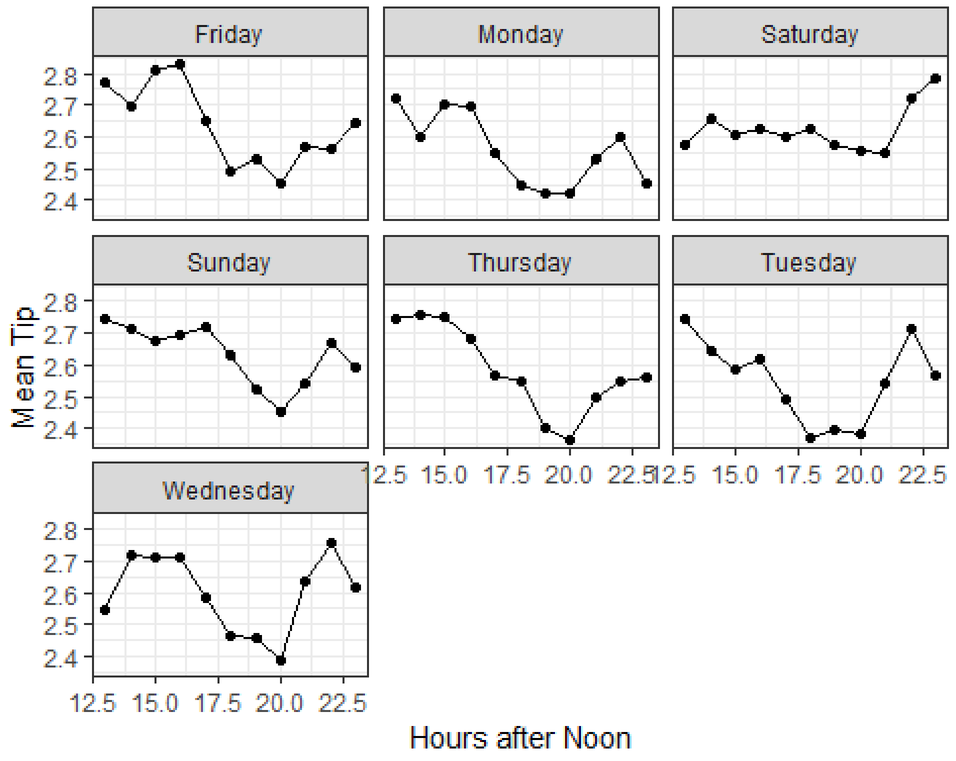
```
ggplot(temp[temp$hour <= 12,], aes(hour, Mean_Tip)) +
  geom_point() +
  geom_line() +
    theme_bw() +
  facet_wrap(~factor(weekday_pickup)) +
  labs(x = 'Hours before Noon', y = 'Mean Tip', Title = 'Tip recieved before 12pm')
```



As it can be see, morning 5 pm can be a good time to earn some tips on days like Friday, Monday, Thursday, Tuesday and Wednesday.

Tips recieved after Noon during the week

```
ggplot(temp[temp$hour > 12,], aes(hour, Mean_Tip)) +
  geom_point() +
  geom_line() +
    theme_bw() +
  facet_wrap(~factor(weekday_pickup)) +
    labs(x = 'Hours after Noon', y = 'Mean Tip')
```



```r
df$weekday_num<-as.integer(format(as.Date(df$lpep_pickup_datetime),"%w"))
```

From this we can confirm the best time to work at night after 9 pm is on Saturday, followed by wednesday and Tuesday (Surprising?)