Data Challenge

Arjun Mishra

February 19, 2017

In this analysis, we are going to use the publicly available dataset for Green Taxis running in New York City in the month of September 2015. This data has been collected by the New York City Taxi and Limousine commission. Green Taxis (as opposed to yellow ones) are taxis that are not allowed to pick up passengers inside of the densely populated areas of Manhattan.

We will start by loading in the data and performing some exploratory analysis on it.

Q.1.

```
# Reading in the dataset from the online source
# Naming variable considering possible future addition of yellow cab data
# Link for data obtained from http://www.nyc.gov/html/tlc/html/about/trip_rec
ord_data.shtml

gtaxi_data = fread("https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata
_2015-09.csv")

##
Read 5.4% of 1494926 rows
Read 14.0% of 1494926 rows
Read 22.1% of 1494926 rows
Read 30.1% of 1494926 rows
Read 34.8% of 1494926 rows
Read 42.8% of 1494926 rows
Read 50.8% of 1494926 rows
Read 58.9% of 1494926 rows
Read 66.2% of 1494926 rows
Read 74.3% of 1494926 rows
Read 82.3% of 1494926 rows
Read 90.3% of 1494926 rows
Read 98.3% of 1494926 rows
Read 1494926 rows and 21 (of 21) columns from 0.223 GB file in 00:00:15

#Checking the dimensions of the dataset
dim(gtaxi_data) # 1494926 x 21 data table

## [1] 1494926      21
```

The data was obtained using the link for the Green Taxi data for the month of September 2015 (on the website http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml).

The fread function in the data.table package allows us to read data directly from the internet. This will ensure that the data is up to date whenever it is read.

The dataset comprises of 1494926 rows and 21 columns.

Continuing with the initial exploration of the data:

```
# Checking types of the data fields and example data values
str(gtaxi_data)
```

```
## 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>
```

```
# Except for Store_and_fwd_flag (string), all other data fields are numerical
/integer
# Some data fields classified as integers should be factors - payment type, t
rip type and VendorID
gtaxi_data$VendorID = as.factor(gtaxi_data$VendorID)
gtaxi_data$Trip_type = as.factor(gtaxi_data$Trip_type)
gtaxi_data$Payment_type = as.factor(gtaxi_data$Payment_type)
gtaxi_data$RateCodeID = as.factor(gtaxi_data$RateCodeID)

# Only NAs can be seen for Ehail_fee - checking for values in the field
print(unique(gtaxi_data$Ehail_fee))
```

```
## [1] NA
```

```r
# Only NA values are contained in the field. The field not contained in data dictionary either
# Removing NA Ehail_fee column
gtaxi_data$Ehail_fee = NULL

# Looking at the statistical summary of the data
summary(gtaxi_data)
```

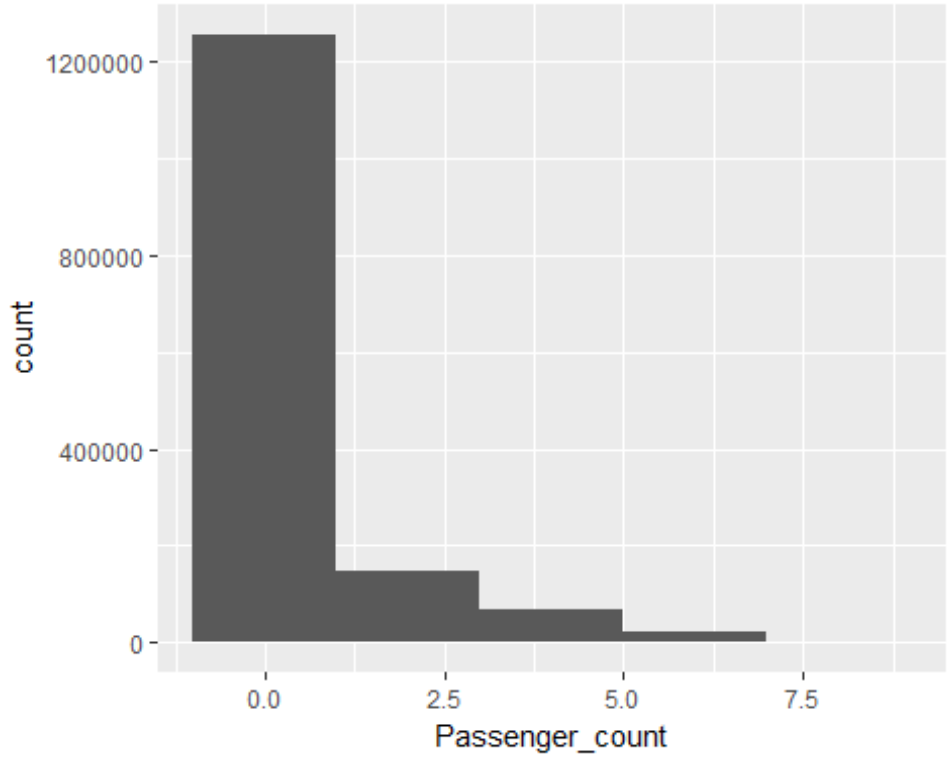
```
##  VendorID    lpep_pickup_datetime Lpep_dropoff_datetime Store_and_fwd_flag
##  1: 325827   Length:1494926       Length:1494926        Length:1494926
##  2:1169099   Class :character     Class :character      Class :character
##              Mode  :character     Mode  :character      Mode  :character
##
##
##
##
##  RateCodeID   Pickup_longitude Pickup_latitude Dropoff_longitude
##  1 :1454464   Min.   :-83.32   Min.   : 0.00   Min.   :-83.43
##  2 :   4435   1st Qu.:-73.96   1st Qu.:40.70   1st Qu.:-73.97
##  3 :   1117   Median :-73.95   Median :40.75   Median :-73.95
##  4 :    925   Mean   :-73.83   Mean   :40.69   Mean   :-73.84
##  5 :  33943   3rd Qu.:-73.92   3rd Qu.:40.80   3rd Qu.:-73.91
##  6 :     36   Max.   :  0.00   Max.   :43.18   Max.   :  0.00
##  99:      6
##  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
##
##  improvement_surcharge  Total_amount     Payment_type Trip_type
##  Min.   :-0.3000       Min.   :-475.00   1:701287     1   :1461506
##  1st Qu.: 0.3000       1st Qu.:   8.16   2:783699     2   :  33416
##  Median : 0.3000       Median :  11.76   3:  5498     NA's:      4
##  Mean   : 0.2921       Mean   :  15.03   4:  4368
##  3rd Qu.: 0.3000       3rd Qu.:  18.30   5:    74
##  Max.   : 0.3000       Max.   : 581.30
##
```

```
# Summary shows some Latitude and Longitude data has 0's which is not right f
or NYC.
# There are negative fares, tips and total_amounts. Either bad data or meanin
g unclear.
# Might need data cleaning based on what appears in graphs and which fields w
ill be needed
```

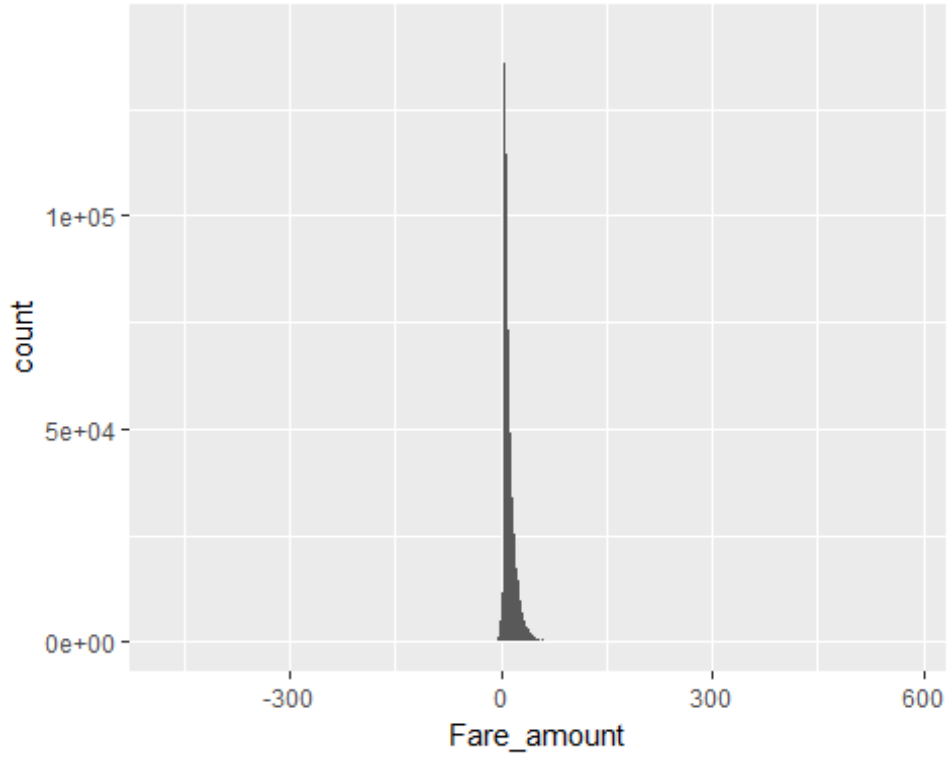
After removing the Ehail_fee variable, which consisted of only NA values, we have 20 columns remaining. It is better to clean the data at the start as it may reduce the complexity of the dataset and the further analysis.

The summary of the data suggests that we need to look at the variables visually too (individually). There might be bad data or we might find some interesting trends.

```
#Plotting the histogram of number of passengers to see distribution
qplot(Passenger_count, data=gtaxi_data,
      geom="histogram", binwidth = 2)
```




```
#Distribution makes sense - the maximum frequency is of the bin <= 2

#Plotting the the Fare Amount to look at how many negative values are present
qplot(Fare_amount, data=gtaxi_data,
      geom="histogram", binwidth = 1)
```




```
# The plot is highly skewed due to outliers. We still cannot say if the negat
ive values are correct or not

#Checking the number of negative values and the number of positive outliers:
print("The number of Fare Amounts in the negative:")

## [1] "The number of Fare Amounts in the negative:"

print(sum(gtaxi_data$Fare_amount < 0)) #There are 2417

## [1] 2417

#Looking at a few negative entries to see what is different
gtaxi_data[gtaxi_data$Fare_amount <= -400, ]

##    VendorID lpep_pickup_datetime Lpep_dropoff_datetime Store_and_fwd_flag
## 1:        2  2015-09-01 13:43:39   2015-09-01 13:43:52                  N
## 2:        2  2015-09-08 14:16:51   2015-09-08 14:20:30                  N
## 3:        2  2015-09-20 23:18:39   2015-09-21 23:18:03                  N
## 4:        2  2015-09-20 23:22:58   2015-09-20 23:23:16                  N
## 5:        2  2015-09-28 19:50:52   2015-09-28 19:54:39                  N
##    RateCodeID Pickup_longitude Pickup_latitude Dropoff_longitude
## 1:          5        -73.83154        40.86857         -73.82864
## 2:          5        -73.78350        40.66738         -73.78350
## 3:          5        -73.96163        40.69819         -73.96164
## 4:          5        -73.96163        40.69817         -73.98080
## 5:          5        -73.99134        40.61805         -73.99094
##    Dropoff_latitude Passenger_count Trip_distance Fare_amount Extra
```

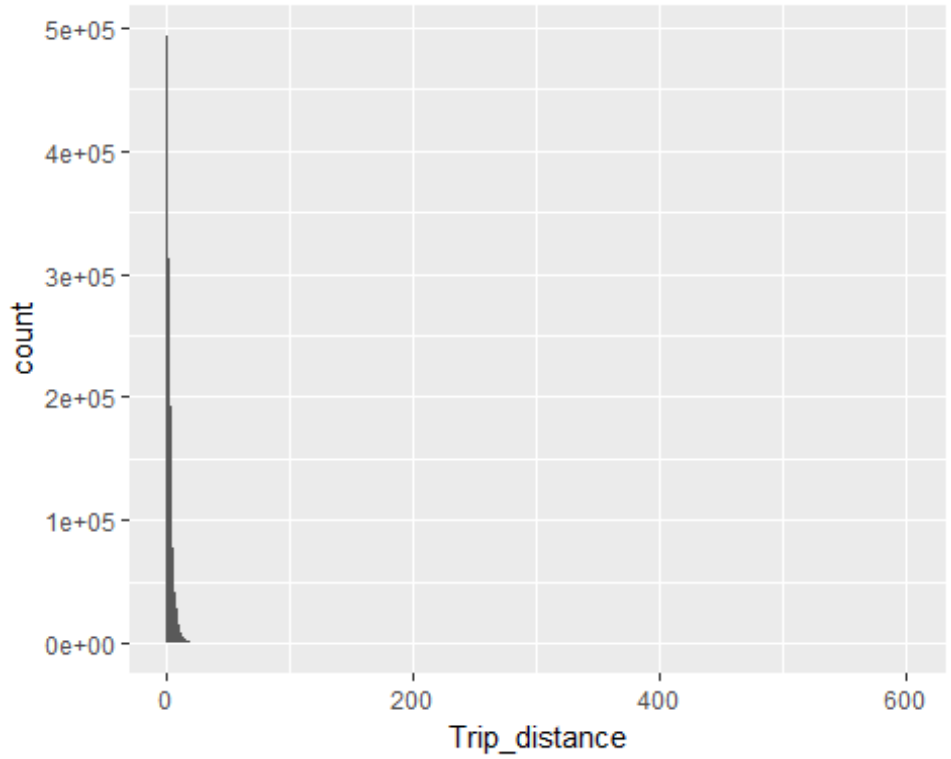
```
## 1:         40.86863                2          0.06        -400     0
## 2:         40.66738                3          0.00        -400     0
## 3:         40.69818                1          0.00        -475     0
## 4:         40.69674                1          0.83        -475     0
## 5:         40.61842                4          0.09        -450     0
##    MTA_tax Tip_amount Tolls_amount improvement_surcharge Total_amount
## 1:       0          0            0                     0         -400
## 2:       0          0            0                     0         -400
## 3:       0          0            0                     0         -475
## 4:       0          0            0                     0         -475
## 5:       0          0            0                     0         -450
##    Payment_type Trip_type
## 1:            4         2
## 2:            4         2
## 3:            3         2
## 4:            3         2
## 5:            3         2

#The negative entries are only of the payment type 3, 4 - no charge and dispu
te
#Confirming this statement
gtaxi_data %>%
    filter(Fare_amount < 0)%>%
    group_by(Payment_type)%>%
    summarize(type_count = n())

## # A tibble: 4 × 2
##   Payment_type type_count
##          <fctr>      <int>
## 1            1          3
## 2            2        205
## 3            3       1338
## 4            4        871

#There are 208 values which are of the cash and credit card type.
```

The negative fare amounts are 2417 in number and mostly for no charge and dispute trips. As the maximum amount are for no charge this does mean that the amounts are not correct. For some of the extreme amounts (-450, -400, -475), the distance traveled was less than a mile. Thus this seems to be bad data.

Q.2.

Now, we will analyze the trip distance for the Green taxis:

```
#Plotting a histogram of the trip distance
qplot(Trip_distance, data=gtaxi_data,
      geom="histogram", binwidth = 1)
```




```
# Again the plot is highly skewed due to the outliers.
#The bulk of the data points are below 50 miles

#Checking number of extreme values for trip_distance
print("The number of trips with trip_distance > 50:")
```

```
## [1] "The number of trips with trip_distance > 50:"
```

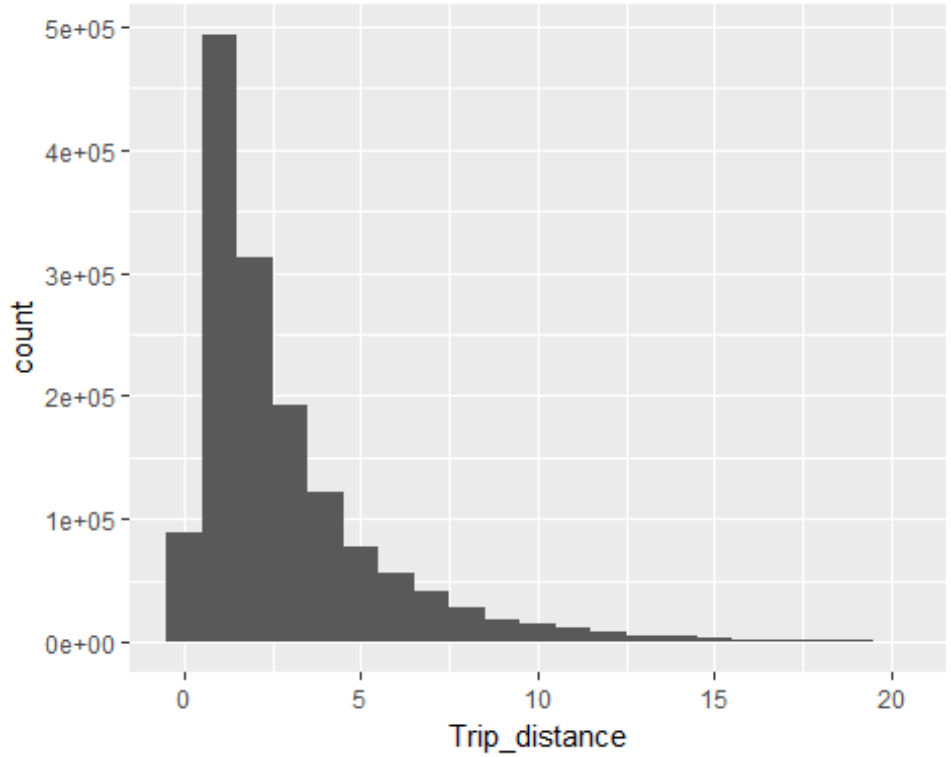
```
sum(gtaxi_data$Trip_distance > 50) #Only 68 trips out of 1494926 greater than
50 miles
```

```
## [1] 68
```

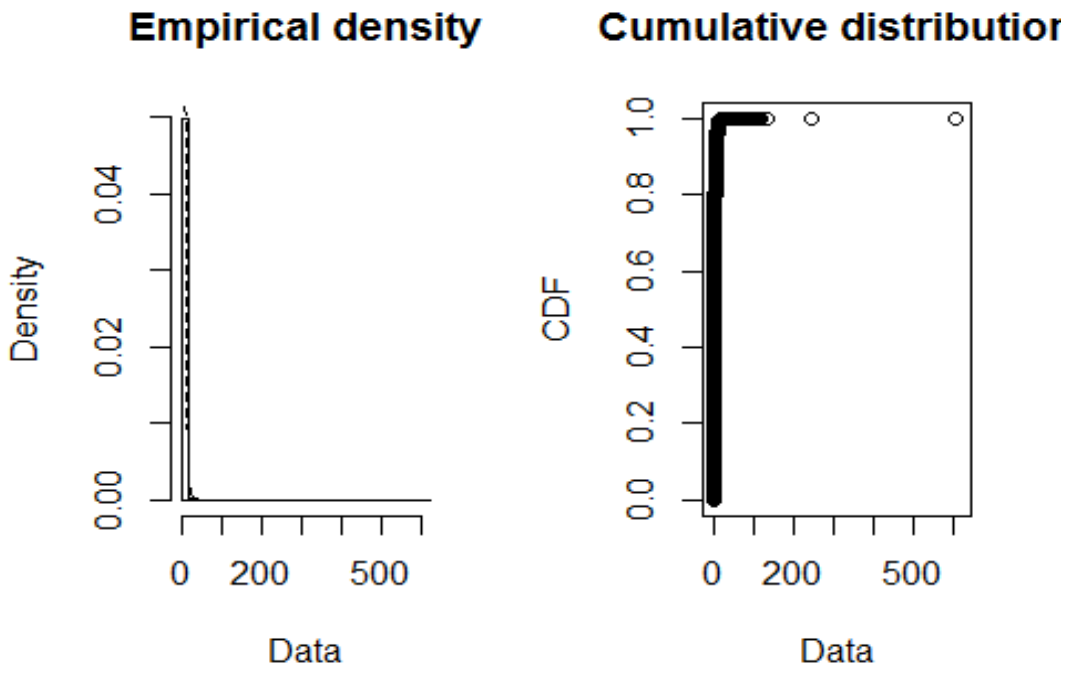
```
sum(gtaxi_data$Trip_distance > 20) #Only 3364 trips out of 1494926 greater th
an 20 miles
```

```
## [1] 3364
```

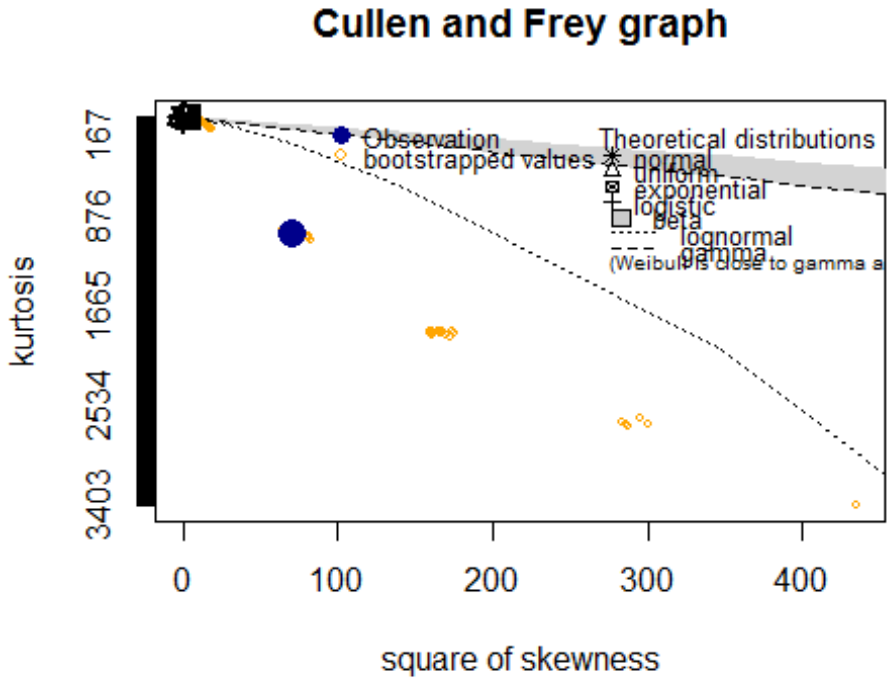
```r
#Plotting the subset dataset for better look at the distribution
qplot(Trip_distance, data=gtaxi_data[gtaxi_data$Trip_distance < 20, ],
      geom="histogram", binwidth = 1)
```



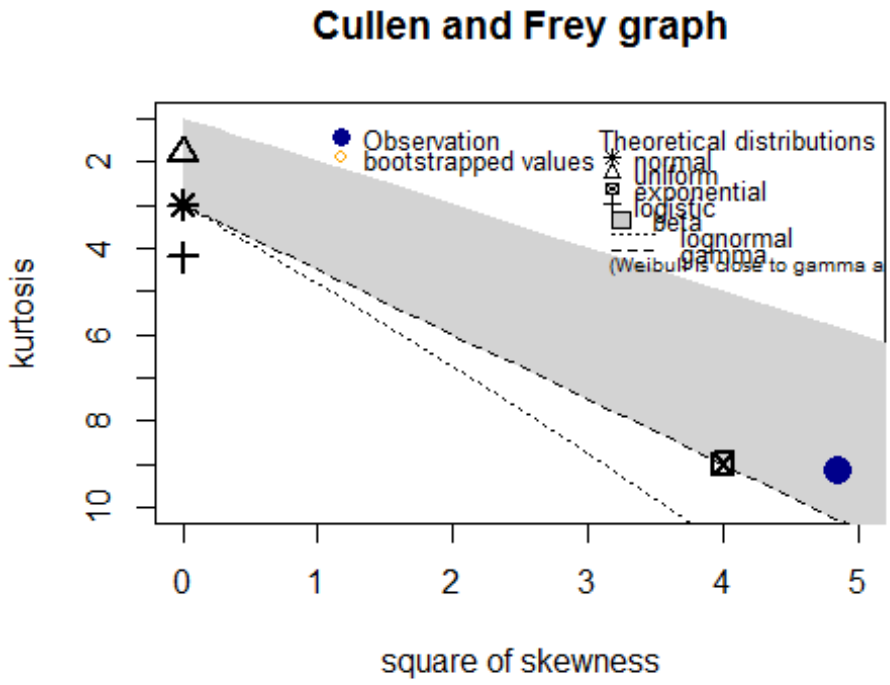

```r
# We want to look at the distribution that best fits this variable to get a b
etter understanding
# We will also plot the CDF and PDF first
# These functions use the fitdistrplus package
plotdist(gtaxi_data$Trip_distance, histo = TRUE, demp = TRUE)
```




```
# Now we will plot the Cullen Frey graph which helps us to figure out which d
istribution fits
# the data in the best manner
descdist(gtaxi_data$Trip_distance, discrete = FALSE, boot = 100)
```

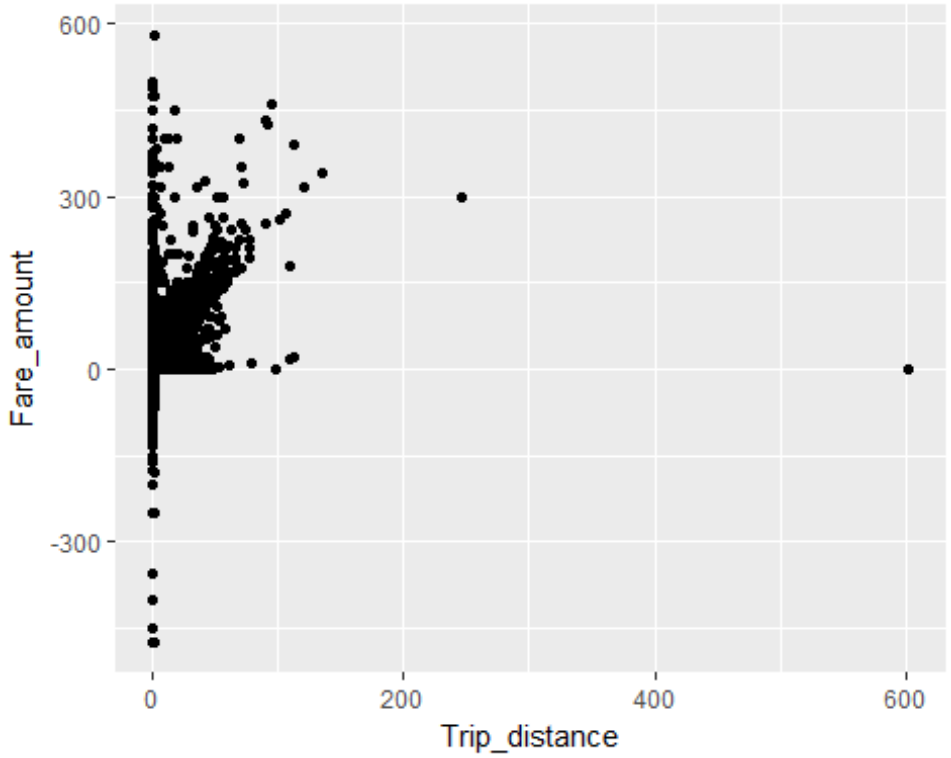


```
## summary statistics
## ------
## min:  0   max:  603.1
## median:  1.98
## mean:  2.968141
## estimated sd:  3.076621
## estimated skewness:  8.442681
## estimated kurtosis:  1025.481

# The bootstrap allows to normalize the random variability in the sample
# The distribution does not necessarily fit any parametric distribution

#Plotting the Cullen Frey for the subset of the data
descdist(gtaxi_data$Trip_distance[gtaxi_data$Trip_distance < 20], discrete =
FALSE, boot = 100)
```

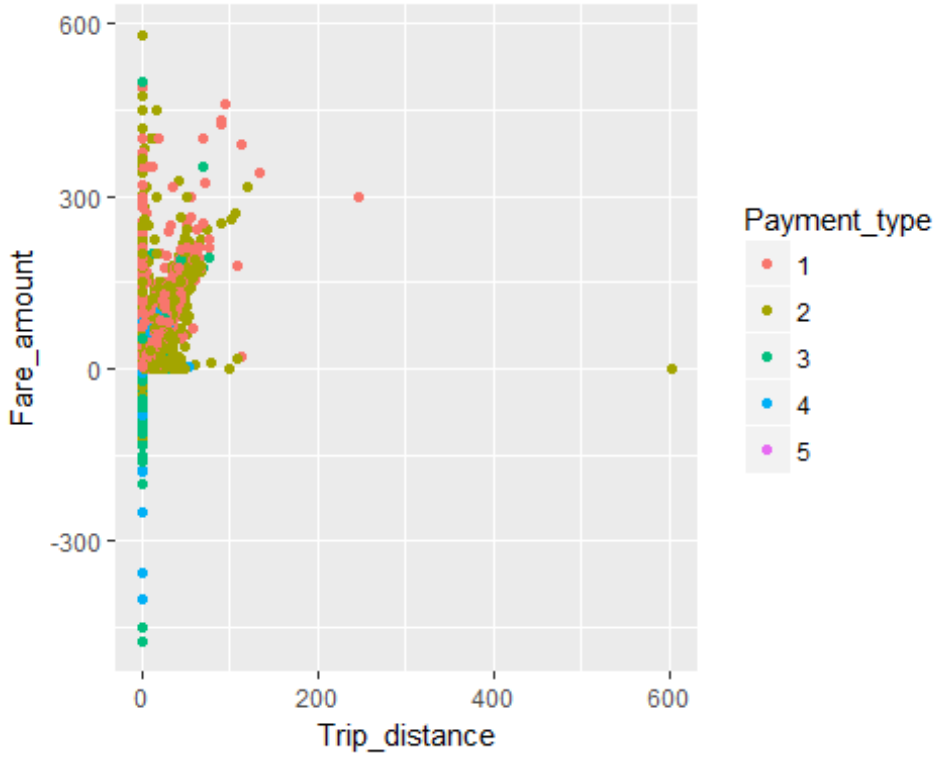



```
## summary statistics
## ------
## min:  0   max:  19.99
## median:  1.97
## mean:  2.916113
## estimated sd:  2.806185
## estimated skewness:  2.201725
## estimated kurtosis:  9.142652

# This dsitribution does not fit any parametric dist. closely either but it d
oes resemble the
```

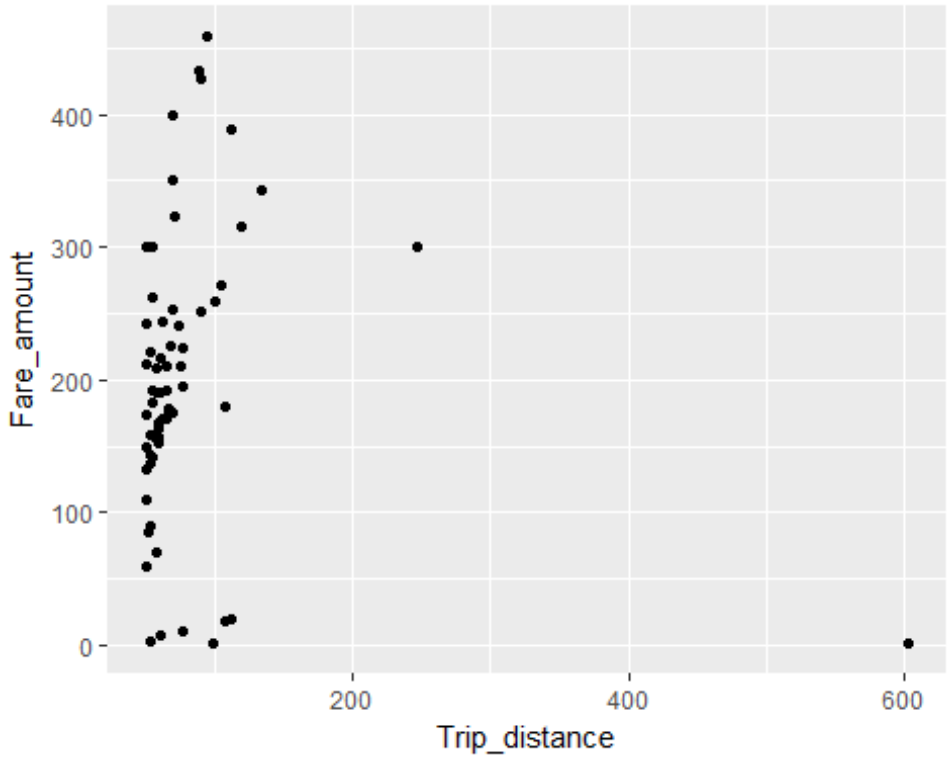
```
# beta and lognormal distributions.

#For a bivariate analysis, we can look at the scatter of trip distance with f
are amount
#to see if the outliers were actual data points or bad data

qplot(Trip_distance, Fare_amount, data = gtaxi_data)
```



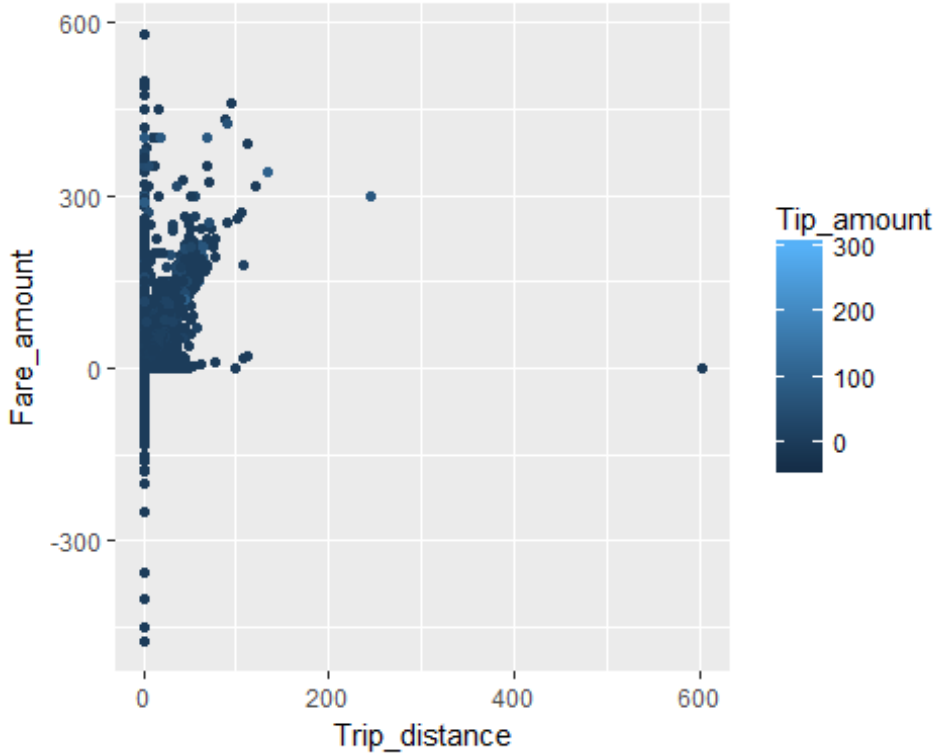

```
# There is a linearly increasing trend among the data but there are alot of p
oints joined to
# the axes as well. These points indicate issues with the data or special cas
es/disputes.

# We can also look at this bivariate analysis in terms of the payment types
qplot(Trip_distance, Fare_amount, data = gtaxi_data, colour = Payment_type)
```




```
qplot(Trip_distance, Fare_amount, data = gtaxi_data[Trip_distance > 50, ])
```



```
#The largest trip with trip distance ~ 600 has 0 fare amount indicating it is
bad data.

# Another indicator to ensure if some points are bad data or not is to see th
e tip amounts for
# the trips as well
qplot(Trip_distance, Fare_amount, data = gtaxi_data, colour = Tip_amount)
```



From the analysis of the visualizations of the variable Trip_distance, we can infer the following things:

1. There are some extreme outliers in the data which are causing the distribution of the variable to be highly right skewed.

2. These outliers can be of two types:

- Either these are bad data entries
- These can be trips that are really long distance and not just within the city.

To distinguish between the two types of outliers, we can see other associated data with it like the fare amount, tip amount and type of payment.

3. There are only ~3000 trips out of 1494926 that have a trip distance greater than 20 miles.

4. The trip distance distribution does not fit any particular parametric distributions closely. We will have to be careful about predicting this variable if using models that assume gaussian distributions.

5. From the bivariate analysis, we do observe a linear trend between the trip distance and fare amount, as can be obviously expected. However, this trend is not as clean as it should be due to points concentrated towards the axes, negative fare amounts etc.

Another analysis that can be done here is:

We can exclude some of the extreme outliers that we know are bad data (negative fares, 0 fares for trip distances > 0) and create a column that is the average fare amount per trip distance, computed separtely for each RateCodeID. This can help us get the average per mile cost of the trips and we can further weed out erroneous data. Our final motive of performing such an analysis is to get as close as we can to the actual distribution and actual data so that we can further model it/use it in other ways.

Q.3.

Now we are going to perform some analysis on the trips in terms of the time variables:

```
# We have two time variables - the time of pickup and dropoff.
# We can use both of these to extract information about the taxi activity in
the City.

# Will use the package lubridate to extract variables from the drop and picku
p variables

##### Prospective variables: date of month, day of week, hour of day, trip ti
me

# We will consider the pickup time as the primary time variable

# Date of month
gtaxi_data$month_date = mday(gtaxi_data$lpep_pickup_datetime)
unique(gtaxi_data$month_date) #Check to see conversion has happened

##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30

# Day of week
gtaxi_data$weekday = wday(gtaxi_data$lpep_pickup_datetime) # Sunday is 1
unique(gtaxi_data$weekday)

## [1] 3 4 5 6 7 1 2

gtaxi_data$weekend = 0
gtaxi_data$weekend[gtaxi_data$weekday %in% c(7, 1)] = 1
```

```r
gtaxi_data$weekday = as.factor(gtaxi_data$weekday)

# Hour of day
gtaxi_data$pickup_hour = hour(gtaxi_data$lpep_pickup_datetime)
unique(gtaxi_data$pickup_hour)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
## [24] 23
```

```r
gtaxi_data$ride_duration = difftime(gtaxi_data$Lpep_dropoff_datetime,
                                    gtaxi_data$lpep_pickup_datetime, units =
"hours")

range(gtaxi_data$ride_duration)
```

```
## Time differences in hours
## [1]  0.00000 23.99833
```

```r
# Now we have all the time variables we need. Reporting mean and median trip
distance by hour

mean_trip = gtaxi_data %>%
              group_by(pickup_hour) %>%
              summarise(mean_distance = mean(Trip_distance))

# The trips are shorter from 9 AM to 9 PM and longest very early morning at 5
and 6.
print("The mean trip distances grouped by the hour of the day are:")
```

```
## [1] "The mean trip distances grouped by the hour of the day are:"
```

```r
print(mean_trip)
```

```
## # A tibble: 24 × 2
##    pickup_hour mean_distance
##          <int>         <dbl>
## 1            0      3.115276
## 2            1      3.017347
## 3            2      3.046176
## 4            3      3.212945
## 5            4      3.526555
## 6            5      4.133474
## 7            6      4.055149
## 8            7      3.284394
## 9            8      3.048450
## 10           9      2.999105
## # ... with 14 more rows
```

```r
median_trip = gtaxi_data %>%
                group_by(pickup_hour) %>%
                summarise(median_distance = median(Trip_distance))
```

```r
# The median reveals the same info - shorter trips from 9-9 and longest early
morning.
print("The median trip distances grouped by the hour of the day are:")
```

```
## [1] "The median trip distances grouped by the hour of the day are:"
```

```r
print(median_trip)
```

```
## # A tibble: 24 × 2
##    pickup_hour median_distance
##          <int>           <dbl>
## 1            0            2.20
## 2            1            2.12
## 3            2            2.14
## 4            3            2.20
## 5            4            2.36
## 6            5            2.90
## 7            6            2.84
## 8            7            2.17
## 9            8            1.98
## 10           9            1.96
## # ... with 14 more rows
```

```r
#We now need to look at trip to and from the NYC airport areas.

#For this, we have the variable RateCodeID that tells us if the trip was to o
r
#from one of the airports

# RateCodeID 2, 3 and 4 correspond to NYC airports.
# LaGuardia is not included in this. Maybe Green Taxis not allowed as
# this is within the city??

# The numerical summary of trips to the airports
airport_trips = gtaxi_data %>%
  filter(RateCodeID %in% c(2, 3, 4)) %>%
  group_by(RateCodeID) %>%
  summarise(number_of_trips = n(), avg_fare = mean(Fare_amount),
            avg_dist = mean(Trip_distance),
            passengers = mean(Passenger_count), avg_tip = mean(Tip_amount))

print("A summary of the trips made to the airports around NYC is:")
```

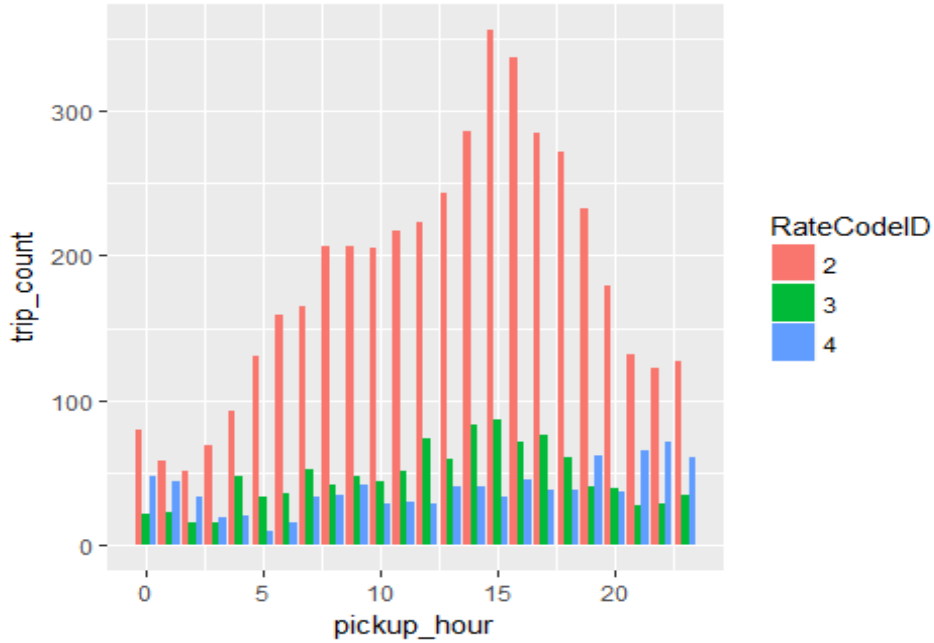
```
## [1] "A summary of the trips made to the airports around NYC is:"
```

```r
print(airport_trips)
```

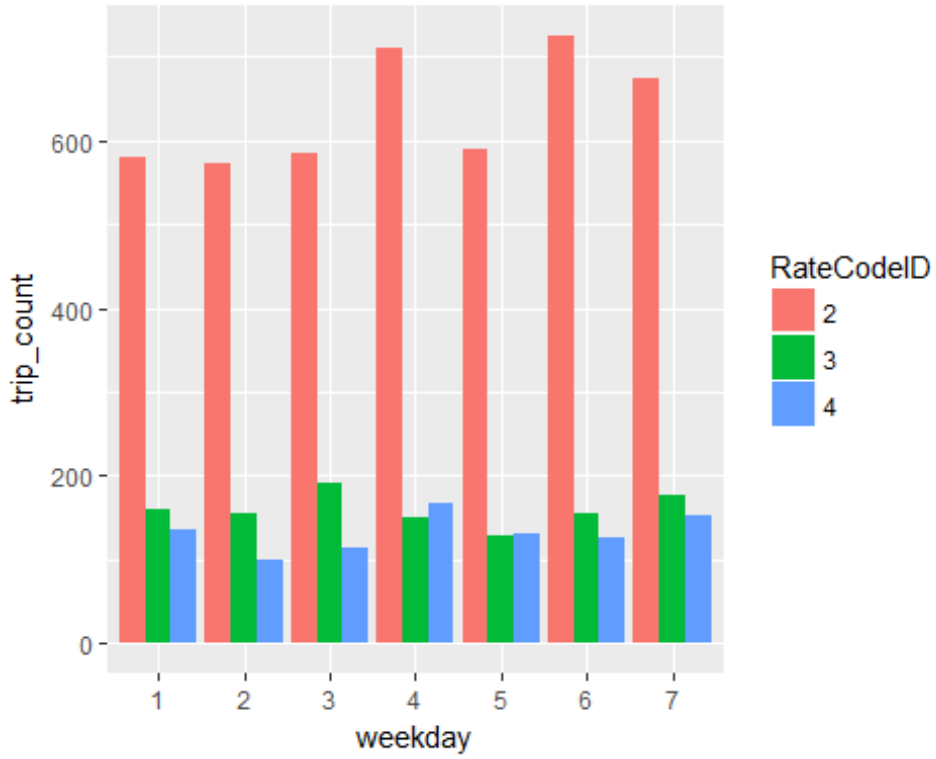
```
## # A tibble: 3 × 6
##   RateCodeID number_of_trips avg_fare avg_dist passengers  avg_tip
##       <fctr>           <int>    <dbl>    <dbl>      <dbl>    <dbl>
```

```
## 1           2            4435 49.02187 10.24478   1.338670 4.080232
## 2           3            1117 48.79857 10.90790   1.396598 5.438577
## 3           4             925 60.16489 14.90240   1.455135 5.284973
```

```
# We can also look at the payment methods for each airport
airport_payments = gtaxi_data %>%
  filter(RateCodeID %in% c(2, 3, 4)) %>%
  group_by(RateCodeID, Payment_type) %>%
  summarise(trip_count = n())

airport_payments = airport_payments %>%
  group_by(RateCodeID) %>%
  mutate(totals = sum(trip_count)) %>%
  mutate(pay_prop = trip_count/totals)

# As expected, most payments were cash or card. The proportion of
# disputes and no charge seems to be higher for Newark compared to JFK. Also,
more JFK flights
# are paid by cash.

# Finally, we can look at the travel times and days for each airport
airport_hour = gtaxi_data %>%
  filter(RateCodeID %in% c(2, 3, 4)) %>%
  group_by(RateCodeID, pickup_hour) %>%
  summarise(trip_count = n())

# To get a better look , we can plot this
ggplot(airport_hour, aes(x = pickup_hour, y = trip_count,
                         fill = RateCodeID)) +
  geom_bar(stat = "identity", position="dodge")
```



```r
#Looking by day of week
airport_day = gtaxi_data %>%
  filter(RateCodeID %in% c(2, 3, 4)) %>%
  group_by(RateCodeID, weekday) %>%
  summarise(trip_count = n())

#Plotting
ggplot(airport_day, aes(x = weekday, y = trip_count,
                        fill = RateCodeID)) +
  geom_bar(stat = "identity", position="dodge")
```

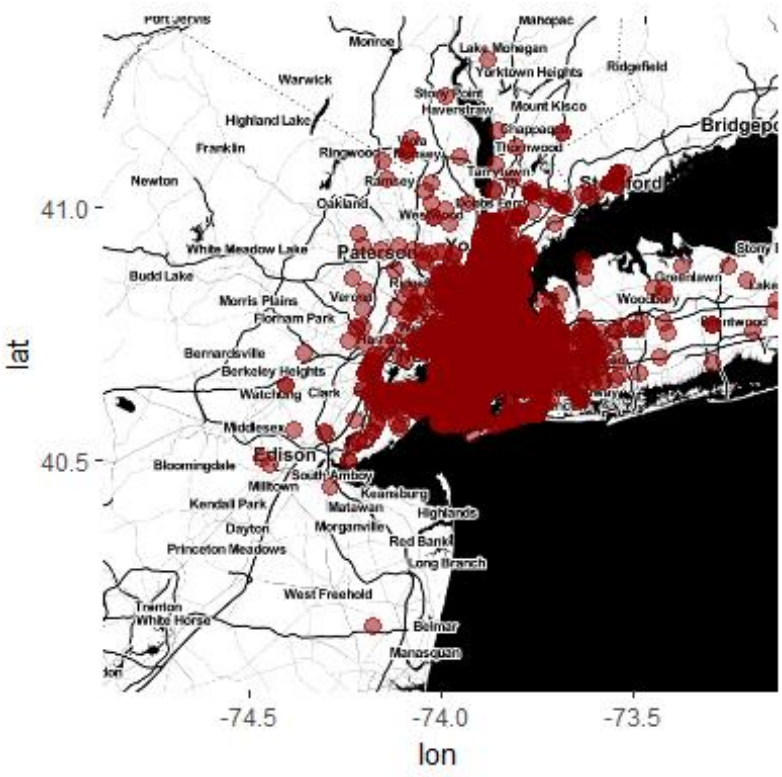



```r
#Using maps to see where all the pickups are located

############ Reference: https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/ggmap/ggmapCheatsheet.pdf


NY_Map = ggmap(get_map(location = 'new york', zoom = 9, source = 'stamen',
                       maptype="toner"))

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=new+york&zoom=9&size=640x640&scale=2&maptype=terrain&sensor=false

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

## Map from URL : http://tile.stamen.com/toner/9/149/191.png
```

```
## Map from URL : http://tile.stamen.com/toner/9/150/191.png

## Map from URL : http://tile.stamen.com/toner/9/151/191.png

## Map from URL : http://tile.stamen.com/toner/9/149/192.png

## Map from URL : http://tile.stamen.com/toner/9/150/192.png

## Map from URL : http://tile.stamen.com/toner/9/151/192.png

## Map from URL : http://tile.stamen.com/toner/9/149/193.png

## Map from URL : http://tile.stamen.com/toner/9/150/193.png

## Map from URL : http://tile.stamen.com/toner/9/151/193.png
```

```
allpickups_Map = NY_Map + geom_point(data = gtaxi_data, aes(x = Pickup_longit
ude , y = Pickup_latitude), alpha = .5, color="darkred", size = 3)

print(allpickups_Map)
```

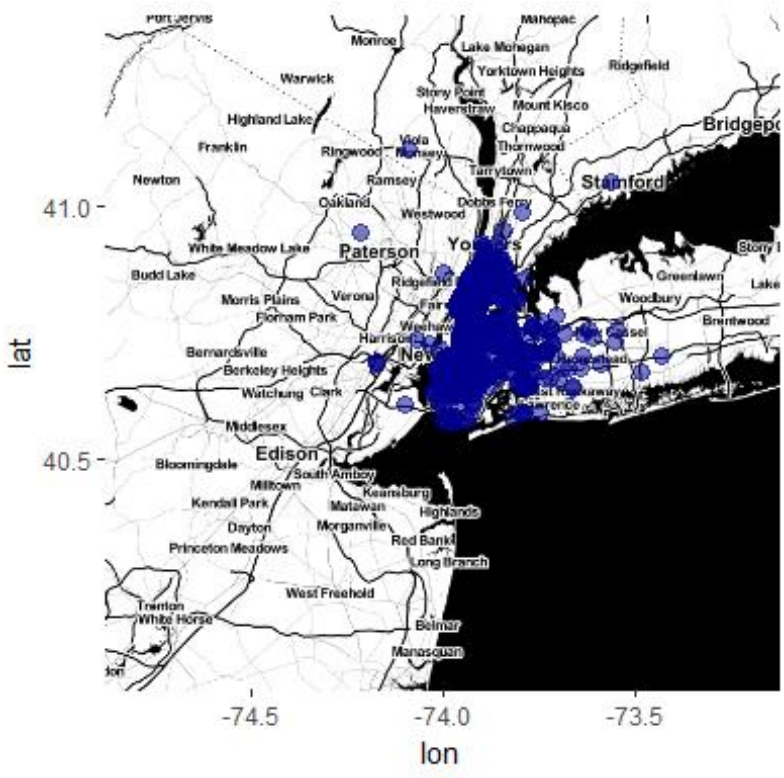
```
## Warning: Removed 2138 rows containing missing values (geom_point).
```




```
# Now plotting only airport pickups

airportpick_Map = NY_Map + geom_point(data = gtaxi_data[RateCodeID %in% c(2,
3, 4), ], aes(x = Pickup_longitude , y = Pickup_latitude), alpha = .5, color=
"darkblue", size = 3)
```

```
print(airportpick_Map)
```

```
## Warning: Removed 32 rows containing missing values (geom_point).
```




```
#The two maps don't show a lot of information that we can use to distinguish
```

From the number of trips for each airport, we can clearly see that JFK has the maximum number of taxi traffic, followed by Newark. The average fare is highest for Westchester/Nassau airports as the average distance travelled for those airports is highest. However, the average number of miles is 50% more than JFK/Newark but the fare is only 20% more compared to those airports. JFK also has more number of solo passengers travelling to the airport compared to the other two airports. Also, the average tip amount is more than a dollar lower than the average tips for Newark and Westchester/Nassau.

Cash is the more preferred mode of payment for JFK and Newark whereas it is card for Westchester/Nassau airports. Also, the proportion of dispute and no charge cases is the highest for Newark aiport trips compared to the other two.

Filtering by the RateCodeID only tells us that the trip was charged at the Airport rate. It does not house information about whether the trip is to or from the airport. As we can see from plotting only the PICKUP LATITUDE and LONGITUDE on a map, the pickups are all around the city and not just from the airports. ------- This is reason I have taken the pickup time as being representative of airport trips. In actuality, the same analysis should be run on the dropoff time as well.

From the pickup_hour VS the trip_count graph, we can see that the number of trips peak between 2PM and 4PM for JFK and Newark. However, the peaks for the Westchester/Nassau airports occur between 9PM to 12PM. For the weekdays, there is no stark trend that we can observe. JFK sees peaked taxi traffic mid week on Wednesday and end of week on Friday and Saturday.

An alternative and much better approach to this problem is that we define a vicinity/neighbouhood of each airport in NYC according to the Latitude and Longitude. If the latitude and longitude for any Pickup or Drop-off for a trip occur within those vicinities, we categorize the trip as one from the airports. The RateCodeID variable does not define a code for Laguardia airport whereas the the map information for Green Taxi trips show a lot of activity in that area.

Once these trips are defined, we can further analyze the drop off and pickup locations for the taxi rides and figure out which neighbourhoods people come from. Based on the residing neighbourhoods of the passengers, we can estimate some of their demographics. Hence, we can define clusters/areas of high tipping customers and low tipping customers. This can further help us prioritize premium jobs versus others. A lot of other analysis can be done based on this information.

Q.4.

Now, we will make a predictive model for a derived variable:

```
# There are 0s for Total_amount variable. We will put tip amount as 0 for tho
se cases.
gtaxi_data$Tip_percent = 0

# Calculate for other cases
gtaxi_data$Tip_percent[gtaxi_data$Total_amount > 0] =
  (gtaxi_data$Tip_amount[gtaxi_data$Total_amount > 0] /
      gtaxi_data$Total_amount[gtaxi_data$Total_amount > 0]) * 100

summary(gtaxi_data)
```

```
##  VendorID    lpep_pickup_datetime Lpep_dropoff_datetime Store_and_fwd_flag
##  1: 325827   Length:1494926       Length:1494926        Length:1494926
##  2:1169099   Class :character     Class :character      Class :character
##              Mode  :character     Mode  :character      Mode  :character
##
##
##
##
##  RateCodeID   Pickup_longitude Pickup_latitude Dropoff_longitude
##  1 :1454464   Min.   :-83.32   Min.   : 0.00   Min.   :-83.43
##  2 :   4435   1st Qu.:-73.96   1st Qu.:40.70   1st Qu.:-73.97
##  3 :   1117   Median :-73.95   Median :40.75   Median :-73.95
```

```
##  4 :    925   Mean   :-73.83   Mean   :40.69   Mean   :-73.84
##  5 :  33943   3rd Qu.:-73.92   3rd Qu.:40.80   3rd Qu.:-73.91
##  6 :     36   Max.   :  0.00   Max.   :43.18   Max.   :  0.00
##  99:      6
##  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
##
##  improvement_surcharge  Total_amount     Payment_type Trip_type
##  Min.   :-0.3000       Min.   :-475.00   1:701287     1   :1461506
##  1st Qu.: 0.3000       1st Qu.:   8.16   2:783699     2   :  33416
##  Median : 0.3000       Median :  11.76   3:  5498     NA's:      4
##  Mean   : 0.2921       Mean   :  15.03   4:  4368
##  3rd Qu.: 0.3000       3rd Qu.:  18.30   5:    74
##  Max.   : 0.3000       Max.   : 581.30
##
##    month_date    weekday       weekend        pickup_hour
##  Min.   : 1.00   1:220675   Min.   :0.0000   Min.   : 0.00
##  1st Qu.: 8.00   2:165396   1st Qu.:0.0000   1st Qu.: 9.00
##  Median :16.00   3:210256   Median :0.0000   Median :15.00
##  Mean   :15.49   4:224804   Mean   :0.3214   Mean   :13.53
##  3rd Qu.:23.00   5:191555   3rd Qu.:1.0000   3rd Qu.:19.00
##  Max.   :30.00   6:222507   Max.   :1.0000   Max.   :23.00
##                  7:259733
##  ride_duration      Tip_percent
##  Length:1494926    Min.   :  0.000
##  Class :difftime   1st Qu.:  0.000
##  Mode  :numeric    Median :  0.000
##                    Mean   :  6.634
##                    3rd Qu.: 16.667
##                    Max.   :100.000
##

#Based on the summary, only trip type has 4 NA's. We will make those 1 (highe
r perentage).
gtaxi_data$Trip_type[is.na(gtaxi_data$Trip_type)] = 1

# Locations will be important in our predictions.
```

```r
# There are 0's in these fields. We will input the means into these fields.
# I will take into account both pickup and drop to compute mean

avg_lat = mean(c(gtaxi_data$Pickup_latitude[gtaxi_data$Pickup_latitude > 0],
                 gtaxi_data$Dropoff_latitude[gtaxi_data$Dropoff_latitude > 0]))

avg_lon = mean(c(gtaxi_data$Pickup_longitude[gtaxi_data$Pickup_longitude < 0]
,
                 gtaxi_data$Dropoff_longitude[gtaxi_data$Dropoff_longitude < 0]
))

# Replacing 0's with mean values
gtaxi_data$Pickup_latitude[gtaxi_data$Pickup_latitude == 0] = avg_lat
gtaxi_data$Dropoff_latitude[gtaxi_data$Dropoff_latitude == 0] = avg_lat

gtaxi_data$Pickup_longitude[gtaxi_data$Pickup_longitude == 0] = avg_lon
gtaxi_data$Dropoff_longitude[gtaxi_data$Dropoff_longitude == 0] = avg_lon

# To validate a new sample, all the created fields will be needed
# Creating a function for preparing the data

data_prep = function(new_data){

  new_data$VendorID = as.factor(new_data$VendorID)
  new_data$Trip_type = as.factor(new_data$Trip_type)
  new_data$Payment_type = as.factor(new_data$Payment_type)
  new_data$RateCodeID = as.factor(new_data$RateCodeID)

  new_data$Ehail_fee = NULL

  new_data$month_date = mday(new_data$lpep_pickup_datetime)

  # Day of week
  new_data$weekday = wday(new_data$lpep_pickup_datetime) # Sunday is 1
  new_data$weekend = 0
  new_data$weekend[new_data$weekday %in% c(7, 1)] = 1
  new_data$weekday = as.factor(new_data$weekday)

  # Hour of day
  new_data$pickup_hour = hour(new_data$lpep_pickup_datetime)

  # Ride duration
  new_data$ride_duration = difftime(new_data$Lpep_dropoff_datetime,
                                    new_data$lpep_pickup_datetime, units =
"hours")

  new_data$Tip_percent = 0

  # Calculate for other cases
```

```
  new_data$Tip_percent[new_data$Total_amount > 0] =
    (new_data$Tip_amount[new_data$Total_amount > 0] /
        new_data$Total_amount[new_data$Total_amount > 0]) * 100

  new_data = dplyr::select(new_data, -VendorID, - lpep_pickup_datetime,
                           - Lpep_dropoff_datetime, - Store_and_fwd_flag, - Ti
p_amount,
                           - MTA_tax, - Extra,
                           - Tolls_amount, - improvement_surcharge, - Total_am
ount)

  return(new_data)

}

set.seed(11)
split_index = createDataPartition(gtaxi_data$Tip_percent, p = 0.8, list = FAL
SE, times = 1)

gtaxi_train = gtaxi_data[split_index, ]
gtaxi_test = gtaxi_data[-split_index, ]

gtaxi_train = dplyr::select(gtaxi_train, -VendorID, - lpep_pickup_datetime,
                           - Lpep_dropoff_datetime, - Store_and_fwd_flag, - Ti
p_amount,
                           - MTA_tax, - Extra,
                           - Tolls_amount, - improvement_surcharge, - Total_am
ount)

gtaxi_test = dplyr::select(gtaxi_test, -VendorID, - lpep_pickup_datetime,
                           - Lpep_dropoff_datetime, - Store_and_fwd_flag, - Ti
p_amount,
                           - MTA_tax, - Extra,
                           - Tolls_amount, - improvement_surcharge, - Total_am
ount)



# We will build a simple regression tree for this problem.
# It does a good job of segregating the set on variable boundaries

# anova method is used for continuous predictions
tree_model = rpart(Tip_percent ~ . , data = gtaxi_train, method = "anova")

predict_test = predict(tree_model, newdata = gtaxi_test)
predictions_table = data.table(actual = gtaxi_test$Tip_percent, predicted = p
redict_test)

err = rmse(predictions_table$actual, predictions_table$predicted)
```

```
# This error term can only be evaluated relatively. We will need more samples
and compute this
# error again to get a sense of what is the right amount.
```

Additional analysis that can be implemented here:

-- The data is not completely clean. For example, the negative values of Fare_Amount have not been dealt with. After careful evaluation, those data fields should either be removed or the values should be replaced.

-- Model parameters should be decided and validated using cross validation first.

-- A lot of other modeling options should also be tried out. Most kaggle competitions are won with boosting methods and ensembles.

-- More features should be added for a better model. We can think about adding interaction terms, try out further feature engineering or add some external data sources.

Q.5.

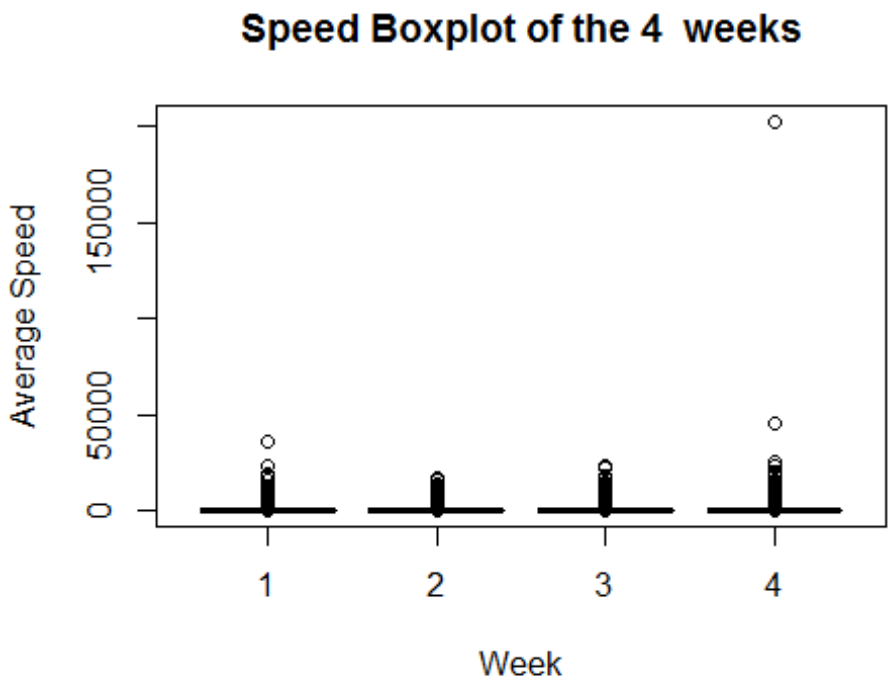
Option A: Distributions o Build a derived variable representing the average speed over the course of a trip.

o Can you perform a test to determine if the average trip speeds are materially the same in all weeks of September? If you decide they are not the same, can you form a hypothesis regarding why they differ?

o Can you build up a hypothesis of average trip speed as a function of time of day?

```
gtaxi_data$avg_speed = 0

#Calculating the speed in miles per hour for ride durations > 0
gtaxi_data$avg_speed[gtaxi_data$ride_duration > 0] =
  gtaxi_data$Trip_distance[gtaxi_data$ride_duration > 0] /
as.numeric(gtaxi_data$ride_duration[gtaxi_data$ride_duration > 0])

# To evaluate whether the speeds are the same in all the weeks of september,
# we will perform a one way anova test.
# Encoding a week variable -  we will add 1 extra day to the 3rd and 4th week
s
gtaxi_data$week = 0
gtaxi_data$week[gtaxi_data$month_date < 31] = 4
gtaxi_data$week[gtaxi_data$month_date < 23] = 3
gtaxi_data$week[gtaxi_data$month_date < 15] = 2
gtaxi_data$week[gtaxi_data$month_date < 8] = 1

# We can first explore difference in the 4 weeks by juts looking at a simple
```

```
box plot
boxplot(gtaxi_data$avg_speed ~ as.factor(gtaxi_data$week),
        xlab = "Week", ylab = "Average Speed")
title(main = "Speed Boxplot of the 4  weeks")
```




```
# The box plot is of no use because of the extreme outliers.

# We can perform a sanity check here - the average speeds in and around new y
ork
# would rarely exceed 100 mph. Checking to see how many such cases are there:
sum(gtaxi_data$avg_speed > 100) # 2919
```

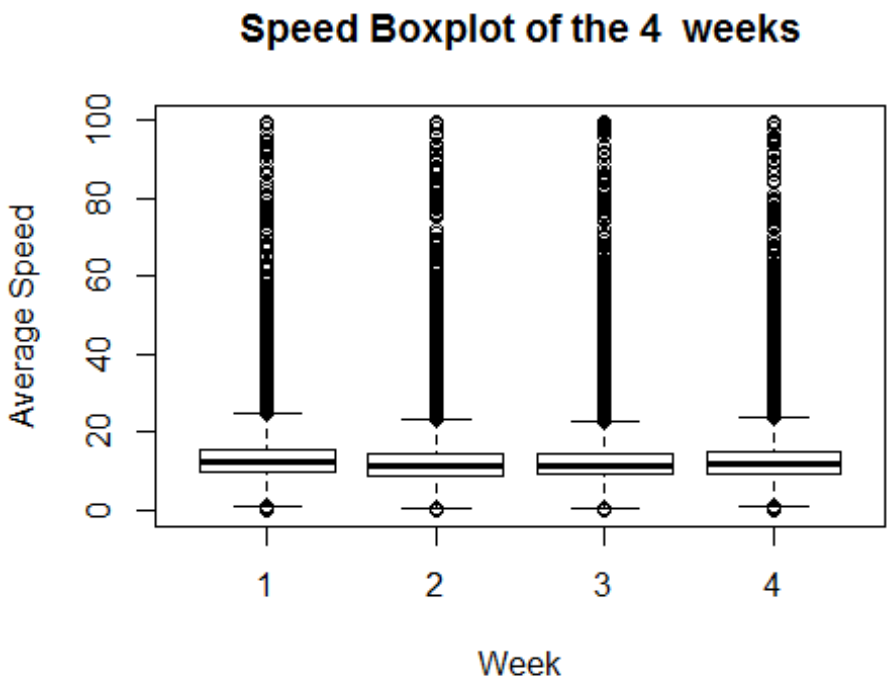
```
## [1] 2919
```

```
sum(gtaxi_data$avg_speed > 150) # 2543
```

```
## [1] 2543
```

```
# This info might just indicate that the speed was very high but not as high
as captured.
# I will remove these from the data for the hypothesis test so that they don'
t skew the results
gtaxi_data1 = gtaxi_data[gtaxi_data$avg_speed < 100, ]

# Plotting the box plot again
boxplot(gtaxi_data1$avg_speed ~ as.factor(gtaxi_data1$week),
```

```
        xlab = "Week", ylab = "Average Speed")
title(main = "Speed Boxplot of the 4  weeks")
```




```
# From the boxplot, there appears to be very little to distinguish between th
e 4 weeks

# To validate this, we will perform the ANOVA test
anova_model = aov(gtaxi_data1$avg_speed ~ as.factor(gtaxi_data1$week))
summary(anova_model)

##                                 Df   Sum Sq Mean Sq F value Pr(>F)
## as.factor(gtaxi_data1$week)      3   220300   73433    2024 <2e-16 ***
## Residuals                  1491999 54142373      36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# The ANOVA model suggests that the means of the groups are different for sur
e.
# The F-value is very huge and hence the p-value is very small

# To check which two pairs are different, we can compute Tukey's pairwise com
parison test.
pair_comp = TukeyHSD(anova_model, which = 'as.factor(gtaxi_data1$week)', conf
.level = 0.95)
pair_comp
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = gtaxi_data1$avg_speed ~ as.factor(gtaxi_data1$week))
##
## $`as.factor(gtaxi_data1$week)`
##           diff         lwr        upr    p adj
## 2-1 -0.9773150 -1.01428978 -0.9403402 0.00e+00
## 3-1 -0.9118043 -0.94781261 -0.8757960 0.00e+00
## 4-1 -0.4646187 -0.50096745 -0.4282700 0.00e+00
## 3-2  0.0655107  0.03004255  0.1009789 1.24e-05
## 4-2  0.5126963  0.47688258  0.5485100 0.00e+00
## 4-3  0.4471856  0.41237055  0.4820006 0.00e+00
```

```
# From this test, we can see that all the weeks are statistically different f
rom each other.
```

```
############## Making a hypothesis of average trip speed as a function of time
of day
```

```
# First, we will calculate the correlation between these two fields
# We are going to take the hour as the proxy for time of day
cor(gtaxi_data1$avg_speed, gtaxi_data1$pickup_hour)
```

```
## [1] -0.1051558
```

```
# Not a very high correlation but it is a negative one.

# To look at the avg_speed as a function of pickup_hour, we will run a simple
regression

lin_model = lm(avg_speed ~ pickup_hour, data = gtaxi_data1)
summary(lin_model)
```

```
##
## Call:
## lm(formula = avg_speed ~ pickup_hour, data = gtaxi_data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.087  -3.488  -1.032   2.203  87.635
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.0870600  0.0109488  1286.6   <2e-16 ***
## pickup_hour -0.0933584  0.0007228  -129.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 6.003 on 1492001 degrees of freedom
## Multiple R-squared:  0.01106,    Adjusted R-squared:  0.01106
## F-statistic: 1.668e+04 on 1 and 1492001 DF,  p-value: < 2.2e-16

# The summary of the model suggests that the regression model is highly signi
ficant.

print(lin_model$coefficients)

## (Intercept) pickup_hour
## 14.08706004 -0.09335836
```

Ou ANOVA and Tukey tests suggest that the means of the average speed for the 4 weeks of September are different than each other. From a purely statistical point of view, this difference occurs due to the extremely high variance of the speeds throughout the weeks. It is unlikely that the average speeds of the taxis go upto 100 miles per hour in the areas they operate in. It is mostly due to the data reported by the taxis that we are geting numbers this high. Apart from the statistical view, these numbers can be different through the weeks because: (My supposition is that longer trips/trips outside the city should have higher average speeds) -- Universities opening in the first week of September can lead to increased demand of long trips. -- Labor day is also in the first week of September. It lead to an extended weekend in the year 2015 as it was on a Monday. This can lead to more people travelling to their homes and thus higher airport traffic as well as longer trips outside the city. Hence the increased average speed. -- The same logic as above can be applied to the last week of September. There are jewish holidays in the last week which might promote long distance trips. -- From a business point of view, the first and last weeks of each month are the ones when a lot of activity takes place. This can also be a reason of different riding patterns during the 4 weeks.

From the regression of Average Speed versus hour of the day, we see that the hour of the day has considerable significance in explaining the variability in the average speed. The equation suggests that as the hour increases in magnitude, the average speed drops. This makes intuitive sense as the traffic increases with increasing hour throughout the day and it drops after midnight. The coefficient is small as this trend is not a strict policy but captures the overall picture.