Problem solving and exploratory data analysis using R

Sinchan Bhattacharya (50096808) and Harsha HS (50098099)
(sinchanb@buffalo.edu) *(hhassans@buffalo.edu)*
Data Intensive Computing, University at Buffalo

1. Abstract:

Petabytes of data is being produced every day through social networks, blogs, business, research or engineering. There is an urgent need to handle this amount of data efficiently. Data Intensive does just that. It facilitates understanding of complex problems that must process massive amounts of data. Through the development of new classes of software, algorithms, and hardware, data-intensive applications can provide timely and meaningful analytical results in response to exponentially growing data complexity and associated analysis requirements. This emerging area brings many challenges that are different from traditional high-performance computing.

In this project we are planning to do Exploratory Data Analysis which is the building block and the most critical part of a model. Firstly we are going to explore the statistical modelling of data and do some hands on experiments and query writing using the questions and examples given our text book. Then we will be bringing our creative skills and work on a large dataset and yield for some patterns. We will be using flight tracking data set which consists of all the information of all the domestic flights within the United States.

2. Project Objectives:

Problem solving and exploratory data analysis using R will meet the following objectives:

- Learn and explore statistical modeling.
- Learn R Language for data analysis.
- Find interesting correlations between flights and the destination cities which we usually fail to notice in day to day life.
- Hands on investigation on existing Flight System using R Studio.

3. Project Approach:

Since statistical analysis is completely new for both of us, we stepped on our first stone by solving the examples and questions given in our text book. We first struggled to get going as we avoided seeing the solutions but gradually got hold of it. We then used the knowledge gained from the two cases by incorporating it to the real time data of Flight Tracking System.

4. Chapter 2: NY Times Data set and outcomes:

We were provided with 31 datasets from the New York Times database where each dataset represented one day's worth of ads shown and clicks recorded on the *New York Times* home page in May 2012. Each row represented a single user. There are five columns: age, gender, number impressions, number clicks and logged in.

We were provided with a set of questions:

The data from the individual csv files could be read using "read.table" command:
data1 <- read.table ("C:/Users/Sinchan/Desktop/Data IntensiveComputing/doing_data_science-master/dds_datasets/nyt1.csv", header = T,sep = ',')

• Create a new variable, age_group, that categorizes users as "<18", "18-24", "25-34", "35-44", "45-54", "55-64", and "65+".

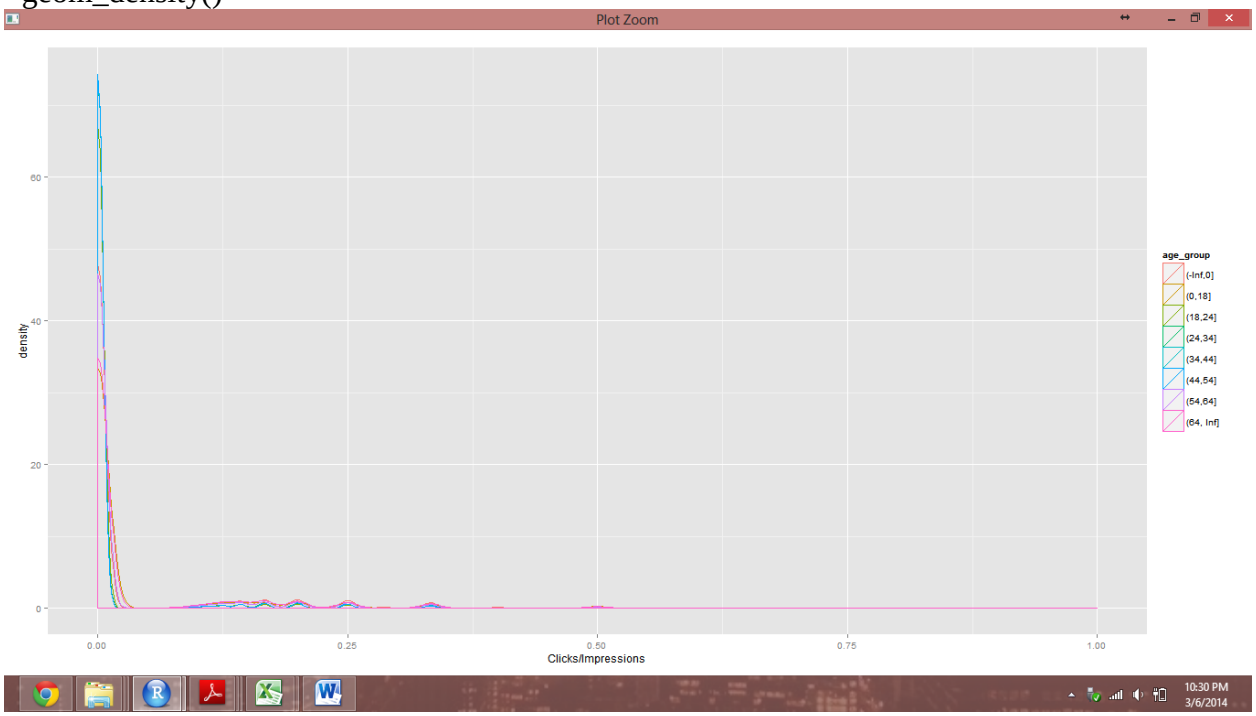
We used if-else function to first categorize the age groups:

```
data1 <-
        transform(
        data1,
        age_group =
        ifelse (
                Age %in% seq (0, 17),
                "<18",
        ifelse (
                Age %in% seq (18, 24),
                "18 - 24",
        ifelse (
                Age %in% seq (25, 34),
                "25-34",
        ifelse (
                Age %in% seq (35, 44),
                "35 - 44",
        ifelse (
                Age %in% seq (45, 54),
                "45 - 54",
                "> 65"
        )
        )
        )
        )
        )
)
```

Or we can simple use the command:
data1$agecat <-cut(data1$Age,c(-Inf,0,18,24,34,44,54,64,Inf))

• For a single day:
• Plot the distributions of number of impressions and clickthrough-rate (CTR=# clicks/# Impressions) for these six age categories.

```
ggplot(subset(data1, Impressions>0), aes(x=Clicks/Impressions,colour=age_group))
+geom_density()
```



Through this density graph we observe that the probability of getting CTR is most in the range of [0, 0.04] and the most effected age group is between (44, 54].
We conclude that the reason being that most of the men and women have reduced concentration in their work and so they tend to deviate from the task at hand. Therefore the marketing companies can target this age group if they want to promote their products.

• Define a new variable to segment or categorize users based on their click behavior.

We have added a new segment in the age group called “Unknown”. We used the following information after observation that for a person who is not signed in, it is impossible to know his/her age. Therefore the gender cannot be concluded as 0 since there is a difference between not knowing the gender and assigning the gender as female.

```
#create categories
data1$HeOrShe1[data1$Gender==0]<-"Female"
data1$HeOrShe1[data1$Gender>0]<-"Male"
data1$HeOrShe1[data1$Signed_In==0]<-"UnKnown"
data1$HeOrShe1<-factor(data1$HeOrShe1)
head(data1)
```

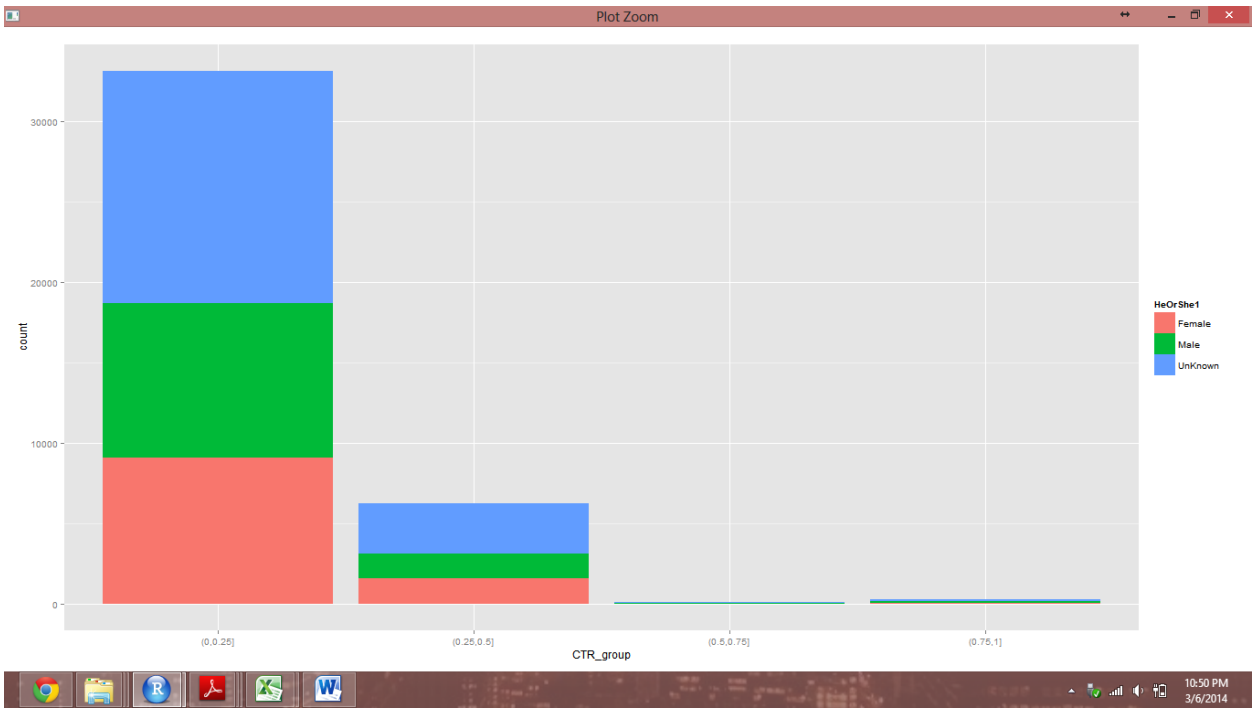
We also categorized users who had a CTR in a certain range. We used:
#Define a new variable to segment or categorize users based on their click behavior
data1$CTR_group<-cut(data1$Clicks/data1$Impressions, c(-Inf,0,0.25,0.5,0.75,1,Inf))

And finally, we categorized Users who produced an Impression or Not.
#create categories based on impression
data1$scode[data1$Impressions==0]<-"No_Impressions"
data1$scode[data1$Impressions>0]<-"Impressionss"
data1$scode[data1$Clicks>0]<-"Clicks"

The output is as follows:

	Age	Gender	Impressions	Clicks	Signed_In	agecat	age_group	he_or_she	HeOrShe1	CTR_group
1	36	0	3	0	1	(34,44]	(34,44]	(-Inf,0]	Female	(-Inf,0]
2	73	1	3	0	1	(64, Inf]	(64, Inf]	(0,1]	Male	(-Inf,0]
3	30	0	3	0	1	(24,34]	(24,34]	(-Inf,0]	Female	(-Inf,0]
4	49	1	3	0	1	(44,54]	(44,54]	(0,1]	Male	(-Inf,0]
5	47	1	11	0	1	(44,54]	(44,54]	(0,1]	Male	(-Inf,0]
6	47	0	11	1	1	(44,54]	(44,54]	(-Inf,0]	Female	(0,0.25]

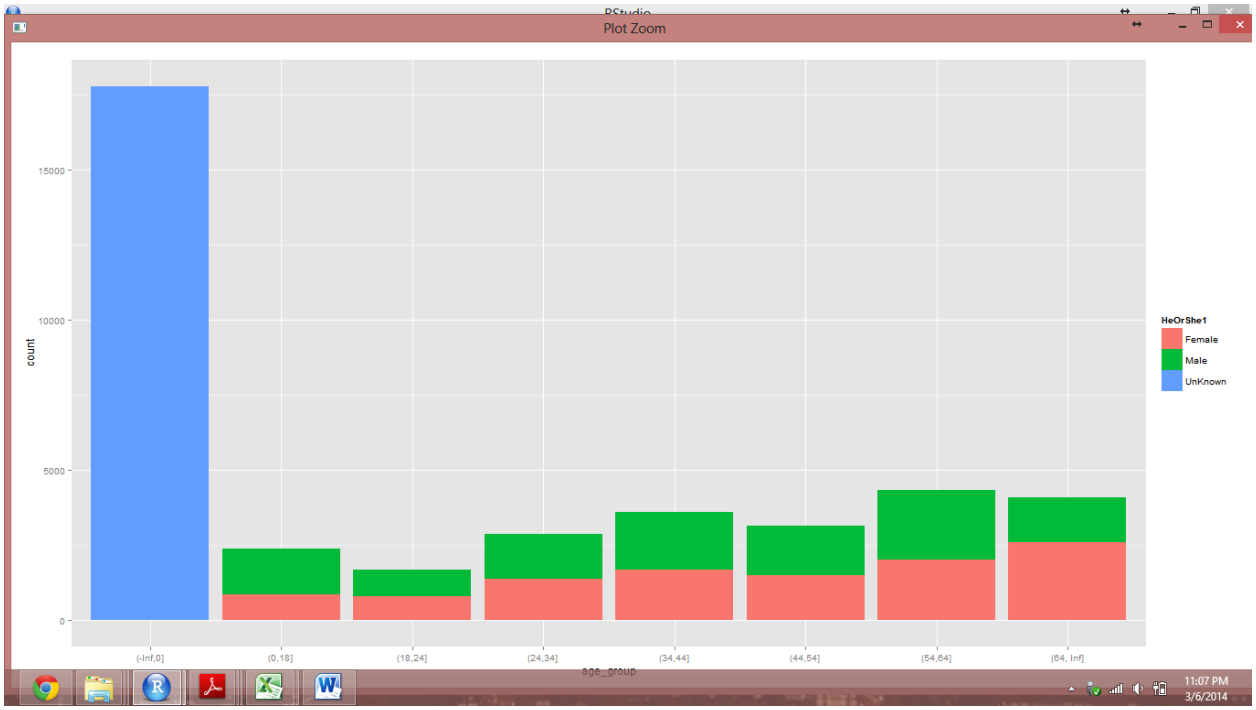
Using:
ggplot(subset(data1, Clicks>0), aes(fill=HeOrShe1, x=CTR_group))
+geom_histogram(binwidth=1),
We generated this histogram and it is evident that majority if the people who click have a CTR of 0.25. Very few people click on the impression which pops up on the web site.



• Explore the data and make visual and quantitative comparisons across user segments/demographics (<18-year-old males versus < 18-year-old females or logged-in versus not, for example).

We have categorized users based on the number of clicks. We have filtered them if they have not clicked on the impressions.

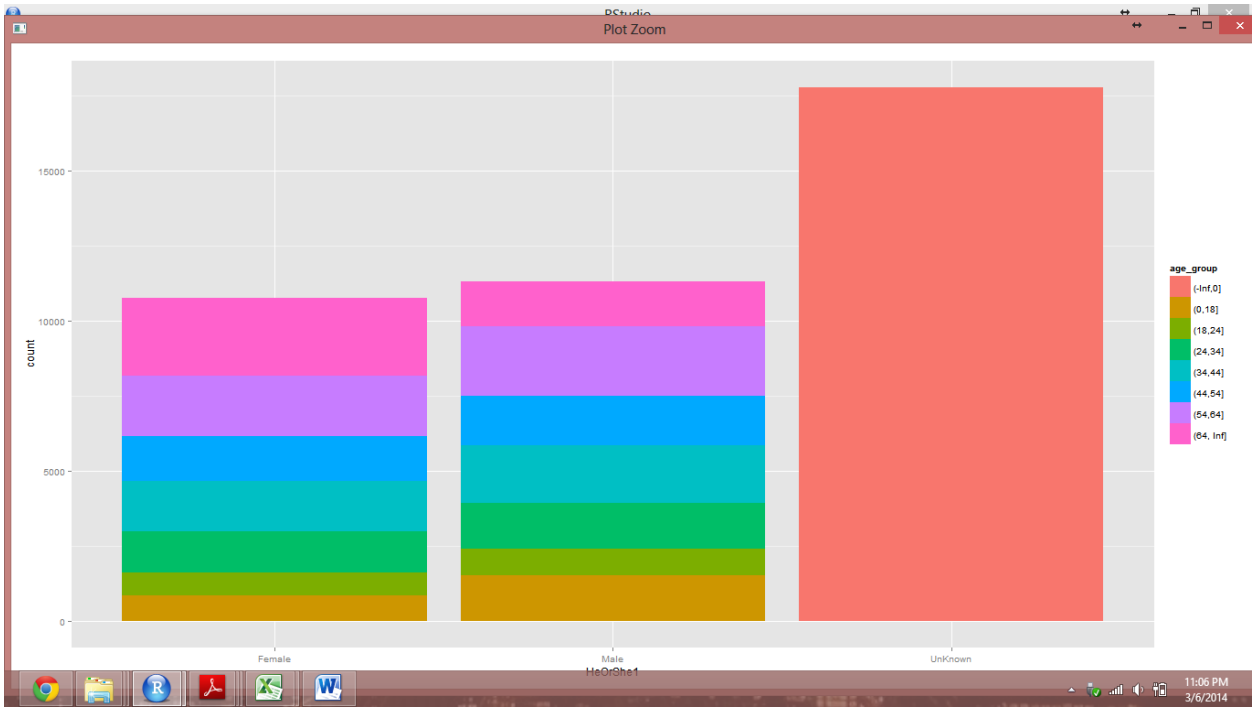
ggplot(subset(data1, Clicks > 0), aes(x=age_group, fill=HeOrShe1))
+geom_histogram(binwidth=1)



Through this plot, we observe that the people who click the most are the senior citizens who are above the age of 55. So the marketing agencies can target that particular group or can improve their advertisements so as to attract more of the young generation.

We used another command:
ggplot(subset(data1, Clicks>0), aes(x=HeOrShe1, fill=age_group))
+geom_histogram(binwidth=1)

This is just another representation where we observe how Male/ Female react to the impressions. We see that there is hardly any difference in the two genders. While the boys less than 18 years click more often than the girls of same age, on the other hand women above 65 are more tempted to click on the impressions than their counterparts.



• Create metrics/measurements/statistics that summarize the data. Think about what will be important to track over time—what will compress the data, but still capture user behavior.
As stated before one metric is the Clicks through ratio (CTR) of which we have already plotted the graphs.
We also used:

```
data1$scode <- factor(data1$scode)
head(data1)
clen <- function(x){c(length(x))}
etable<-summaryBy(Impressions~scode+Gender+agecat,
        data = data1, FUN=clen)
```

The output of the following is:

```
scode Gender age_group Impressions.clen
1  Clicks      0  (-Inf,0]                2
2  Clicks      0   (44,54]                1
3    Imps      0  (-Inf,0]                2
4    Imps      0    (0,18]                1
5    Imps      0   (18,24]                1
6    Imps      0   (24,34]                2
7    Imps      0   (34,44]                1
8    Imps      0   (44,54]                2
9    Imps      0   (54,64]                1
10   Imps      1   (24,34]                1
11   Imps      1   (34,44]                1
12   Imps      1   (44,54]                2
13   Imps      1 (64, Inf]                1
```

As we can observe in the final Summary, Clicks can be observed only in the users between the age group of –Inf to 0, who are Unsigned or the users whom we do not have the information about and between the age group of 44 to 54 years. In the other age groups we

could hardly observe any clicks but there is varying distribution of the number of Impressions.

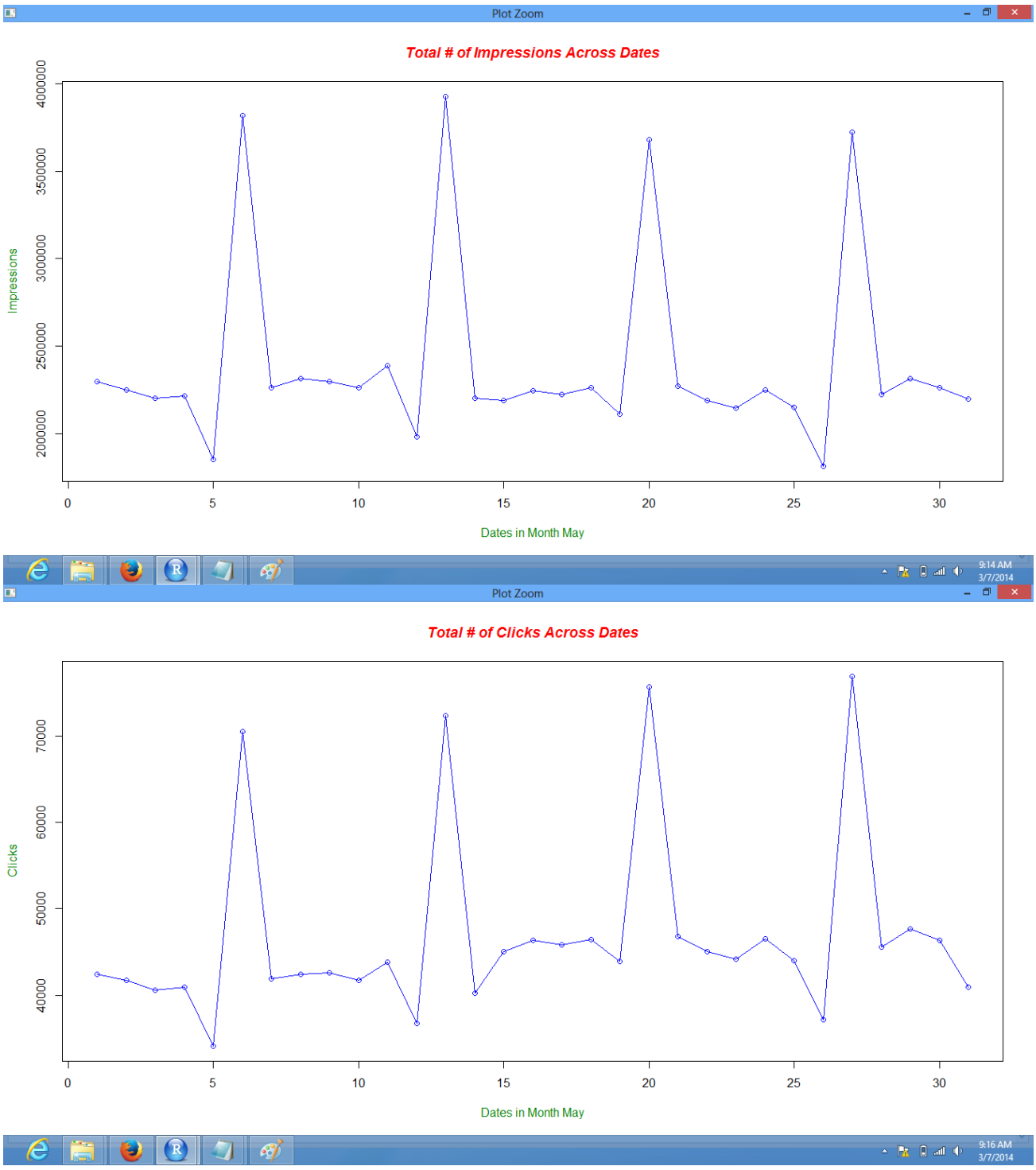
• Now extend your analysis across days. Visualize some metrics and distributions over time.

First, we imported all the csv files into the variables and then assigned an unique variable using cbind() to recognize each file uniquely.

```
data1<-read.csv("nyt1.csv")
data2<-read.csv("nyt2.csv")
data3<-read.csv("nyt3.csv")
data4<-read.csv("nyt4.csv")
data5<-read.csv("nyt5.csv")
data6<-read.csv("nyt6.csv")
data7<-read.csv("nyt7.csv")
data8<-read.csv("nyt8.csv")
data9<-read.csv("nyt9.csv")
data10<-read.csv("nyt10.csv")
data11<-read.csv("nyt11.csv")
data12<-read.csv("nyt12.csv")
data13<-read.csv("nyt13.csv")
data14<-read.csv("nyt14.csv")
data15<-read.csv("nyt15.csv")
data16<-read.csv("nyt16.csv")
data17<-read.csv("nyt17.csv")
data18<-read.csv("nyt18.csv")
data19<-read.csv("nyt19.csv")
data20<-read.csv("nyt20.csv")
data21<-read.csv("nyt21.csv")
data22<-read.csv("nyt22.csv")
data23<-read.csv("nyt23.csv")
data24<-read.csv("nyt24.csv")
data25<-read.csv("nyt25.csv")
data26<-read.csv("nyt26.csv")
data27<-read.csv("nyt27.csv")
data28<-read.csv("nyt28.csv")
data29<-read.csv("nyt29.csv")
data30<-read.csv("nyt30.csv")
data31<-read.csv("nyt31.csv")


#Adds a extra field indicating date and differentiates between
#sets like nyt1, nyt2 .....
data1<-cbind(Date = 1, data1)
data2<-cbind(Date = 2, data2)
data3<-cbind(Date = 3, data3)
```

```
data4<-cbind(Date = 4, data4)
data5<-cbind(Date = 5, data5)
data6<-cbind(Date = 6, data6)
data7<-cbind(Date = 7, data7)
data8<-cbind(Date = 8, data8)
data9<-cbind(Date = 9, data9)
data10<-cbind(Date = 10, data10)
data11<-cbind(Date = 11, data11)
data12<-cbind(Date = 12, data12)
data13<-cbind(Date = 13, data13)
data14<-cbind(Date = 14, data14)
data15<-cbind(Date = 15, data15)
data16<-cbind(Date = 16, data16)
data17<-cbind(Date = 17, data17)
data18<-cbind(Date = 18, data18)
data19<-cbind(Date = 19, data19)
data20<-cbind(Date = 20, data20)
data21<-cbind(Date = 21, data21)
data22<-cbind(Date = 22, data22)
data23<-cbind(Date = 23, data23)
data24<-cbind(Date = 24, data24)
data25<-cbind(Date = 25, data25)
data26<-cbind(Date = 26, data26)
data27<-cbind(Date = 27, data27)
data28<-cbind(Date = 28, data28)
data29<-cbind(Date = 29, data29)
data30<-cbind(Date = 30, data30)
data31<-cbind(Date = 31, data31)


data_final<-rbind(data1, data2, data3, data4, data5, data6, data7, data8,
                  data9, data10, data11, data12, data13, data14, data15,
                  data16, data17, data18, data19, data20, data21, data22,
                  data23, data24, data25, data26, data27, data28, data29,
                  data30, data31)

data_final<-cbind(CTR=as.numeric(data_final$Clicks/data_final$Impressions), data_final)

data_final$age_group<-cut(data_final$Age, c(-Inf,0,18,24,34,44,54,64,Inf))
data_final$he_or_she<-cut(data_final$Gender, c(-Inf,0,1,Inf))


#Plot Total Impressions Across Dates
sum_Imps<-aggregate(Impressions~Date, data=data_final, FUN =sum)
plot(sum_Imps$Impressions, type='o', col="blue", ann=FALSE)
title(main="Total # of Impressions Across Dates", col.main="red", font.main=4)
```

```
title(xlab="Dates in Month May", col.lab=rgb(0,0.5,0))
title(ylab="Impressions", col.lab=rgb(0,0.5,0))

#Plot Total Clicks Across Dates
sum_Clicks<-aggregate(Clicks~Date, data=data_final, FUN=sum)
plot(sum_Clicks$Clicks, type='o', col="blue", ann=FALSE)
title(main="Total # of Clicks Across Dates", col.main="red", font.main=4)
title(xlab="Dates in Month May", col.lab=rgb(0,0.5,0))
title(ylab="Clicks", col.lab=rgb(0,0.5,0))

#Plot Click Through Rate Across Dates
mean_CTR<-aggregate(CTR~Date, data=data_final, FUN=mean)
plot(mean_CTR$CTR, type='o', col="red", ann=FALSE)
title(main="Click Through Rate Across Dates", col.main="red", font.main=4)
title(xlab="Dates in Month May", col.lab=rgb(0,0.5,0))
title(ylab="Average Click Through Rates", col.lab=rgb(0,0.5,0))
```



Here we see that the number of impressions is constant. We can deduce this fact from the height if the plot, whereas for clicks, we observe that the number of clicks after 15th of May increases by a large extent as the graph has shifted up somewhat. So, we can infer that after15th of May, the CTR should increase by a considerable amount. Now, we plot CTR across the month of May.

Observation: It is a very important observation which is exactly as we anticipated. After 15th of May, the Clicks through Rate drastically shot up. The above plots shows that the Clicks and Impressions almost follow the same pattern but there is an improvement in the CTR due to the improvement in the number of clicks and constant nature of impressions

Now we plot a histogram to see the change in the number of people who clicked based on their age group.

ggplot(data_final, aes(x=Date, fill=age_group))+geom_histogram(binwidth=1)

We observe that on 6th, 13th, 20st and 27th of May the number of clicks considerably spiked up. On cross checking with the calendar for May, 2012, we observe that these are Sundays. So we can conclude safely that people used the website the most on Sundays as a result they see the impressions and clicks on Sundays the most.

Using:
ggplot(data_final, aes(x=Date, fill=he_or_she))+geom_histogram(binwidth=1)

We also observe that, the majority of the clicks come from female. So the advertising companies can target products more attractive to the male population so as to increase their sales.

5. Chapter 2: RealDirect data set and outcomes:

In this case study we were provided with the Sales of Bronx Rolling Sales File for a fiscal year from August 2012 - August 2013 for six parts of the New York City. We are asked to first clean the data and then do some EDA and improve the way people buy or sell houses in the city region.

Stage1: Data Cleaning:

```
library("gdata")
bk<-read.xls("rollingsales_ manhattan.xls",pattern="BOROUGH")
getwd()
head(bk)
summary(bk)
```

Suppose we have the vector:

```
x <- c("23455","23456" , "abc")
```

But we do not want “abc” to be printed. So we use,

```
y<-as.numeric(gsub("[^[:digit:]]", "", x))
```

This will show the result:

```
23455 23456    NA
```

Thus we need to clean all the data in the Sales Price, Square Feet and Date so that they only take in the respective integers and clears out all the anomalous data. Therefore we used the commands:

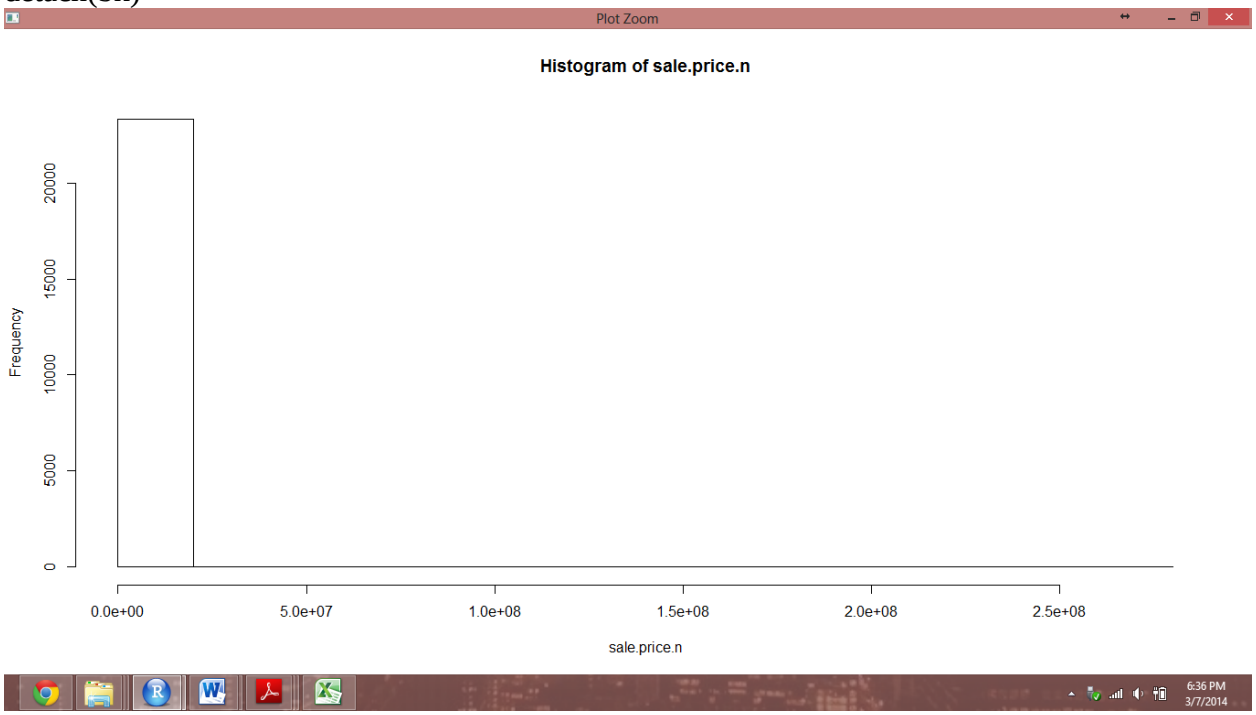
```
bk$sale.price.n<- as.numeric(gsub("[^[:digit:]]","",bk$sale.price))
bk$gross.sqft<-as.numeric(gsub("[^[:digit:]]","",bk$gross.square.feet))
bk$land.sqft<-as.numeric(gsub("[^[:digit:]]","",bk$land.square.feet))
bk$sale.date<-as.Date(bk$sale.date)
bk$year.built<-as.numeric(as.character(bk$year.built))


count(is.na(bk$SALE.PRICE.N))
names(bk)<-tolower(name(bk))

count(is.na(bk$SALE.PRICE.N))
names(bk)<-tolower(names(bk))
```

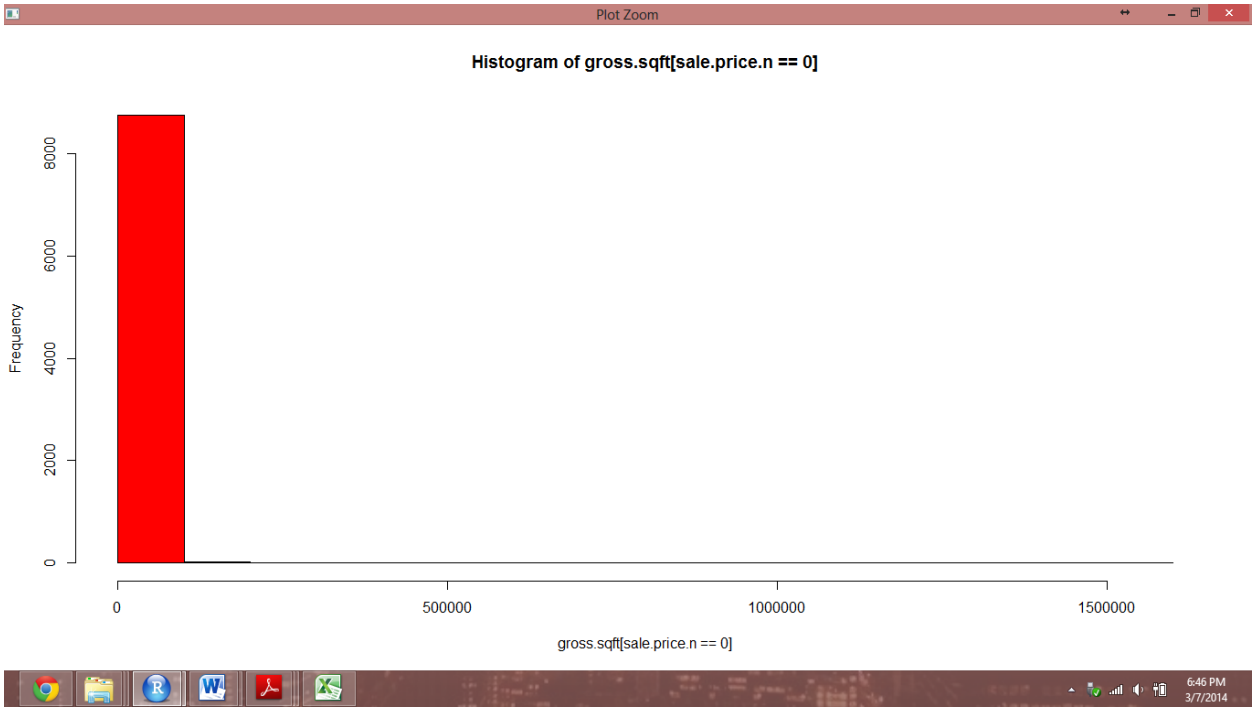
Stage2: EDA

First we plot a regular histogram between sales price and its respective counts in three different scenarios

```
attach(bk)
hist(sale.price.n)
hist(sale.price.n[sale.price.n>0])
```

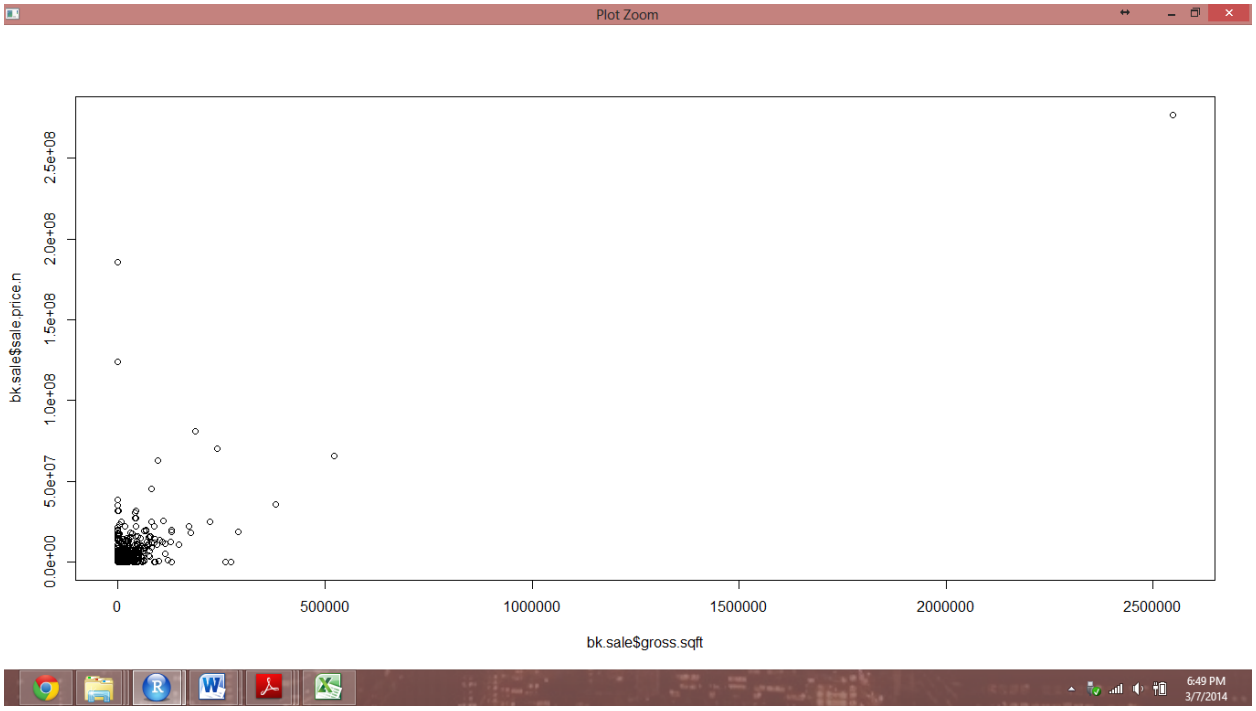
```
hist(gross.sqft[sale.price.n==0])
detach(bk)
```





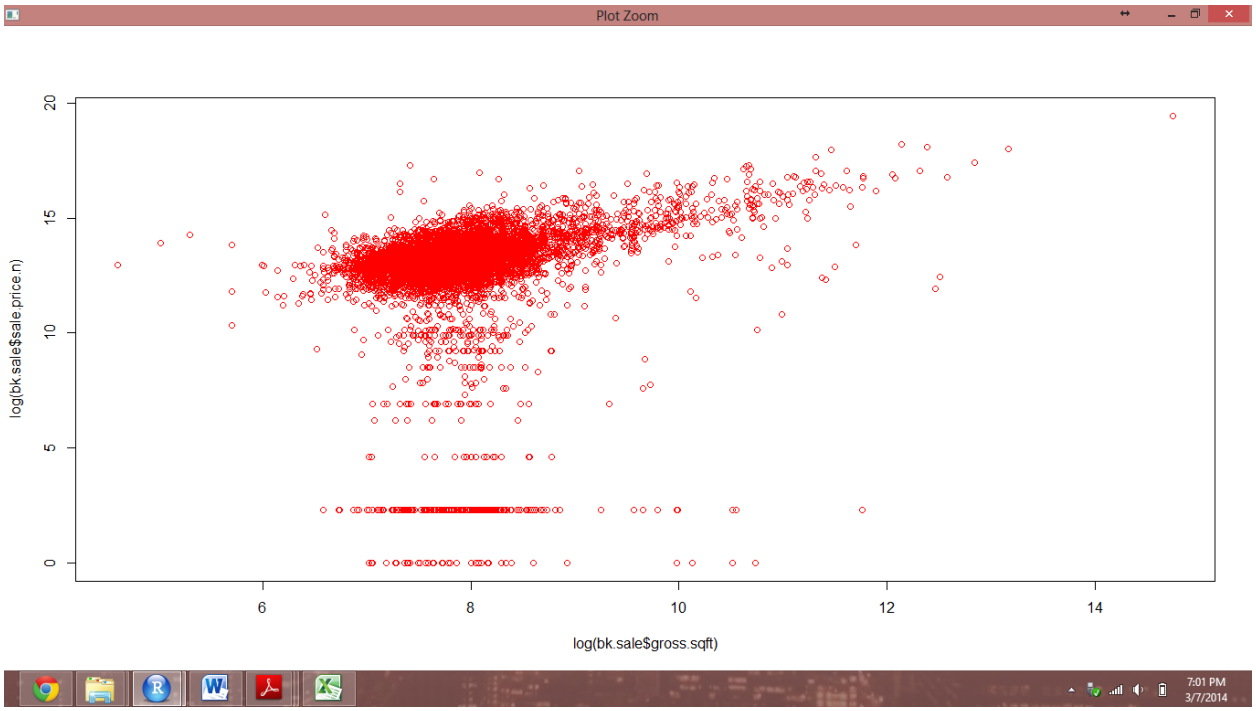
Now we plot the distribution of sqare feet versus sale price.

```
bk.sale<-bk[bk$sale.price.n!=0,]
plot (bk.sale$gross.sqft,bk.sale$sale.price.n)
```



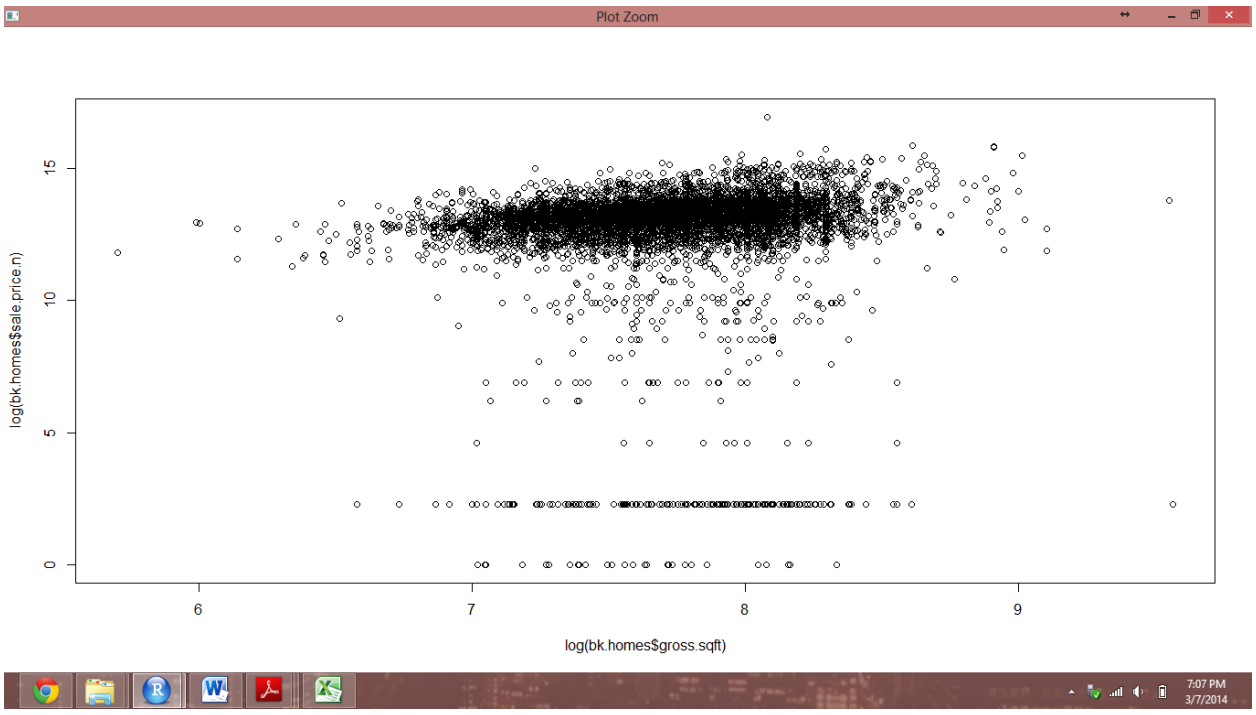
Here we observe that there is some anamolous behaviour between the two. Because for 0 sqaure feet, the sale price cannot be as high as 1.7×10^8 $. This is wrong and hence we can conclude that the data is noisy.

Taking a closer look at the data, we adjust the scale by taking log on both the sides of the axises and hence we use:
plot(log(bk.sale$gross.sqft),log(bk.sale$sale.price.n), col = 'red')



This plot shows how exactly the points are plotted between axises.

Now let us see how the plot only the square feet which contribute to the 'FAMILY' type of the Building Class Category.
We use the where clause.
So, using the command:
bk.homes <- bk.sale[which(grepl("FAMILY",bk.sale$building.class.category)),]
plot (log(bk.homes$gross.sqft),log(bk.homes$sale.price.n))
We observe that there is no unusual behavior of the plot between gross square feet and sales price. It is almost similar to the plot of the sqaure feet of all the building category verses sales, only just shifted towards the right.



6. Own Data:

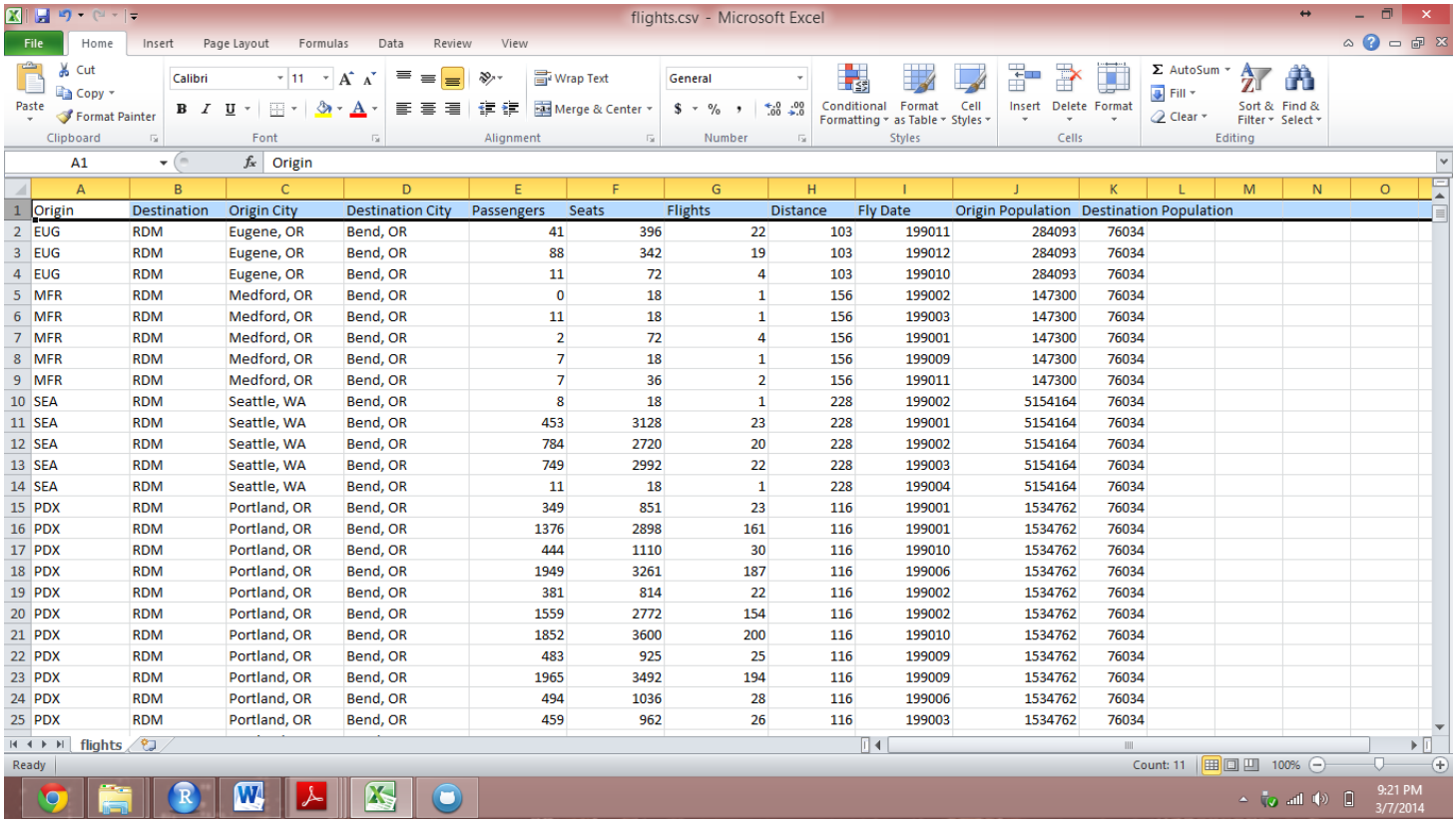
6.1 Data Set Name and Source:

We have used a datasets which tracks all the domestic US Flights between all the major States in the United States. It also maintains the census details of the origin and the destination cities, seats available in each flight and the distances between the origin and the destination cities. This data is available at:
https://github.com/sinchan15/Flight_Details.

Explanation of the data set:
Origin:
It shows the Airport code of the origin Airport.
Destination:
It shows the Airport code of the destination Airport.
Origin City:
It shows the city where the origin Airport is located along with the State code.
Destination City:
It shows the city where the destination Airport is located along with the State code.
Passengers:
It gives the passenger count of a single flight.

Seats:
It gives the count of the number of total available seats for a particular flight.
Flights:
It gives us the number of flights operational between the origin and the destination airport.
Distance:
It gives the total flight distance between the source and destinations.
Fly Date:
The date in which the all the flights took off in that particular month. It is in the format of YYYYMM. For example: 199003.
Origin Population:
It gives the total population of the origin city.
Destination Population:
It gives the total population of the origin city.

We have provided you the snapshot of the dataset for your perusal.



Origin	Destination	Origin City	Destination City	Passengers	Seats	Flights	Distance	Fly Date	Origin Population	Destination Population
EUG	RDM	Eugene, OR	Bend, OR	41	396	22	103	199011	284093	76034
EUG	RDM	Eugene, OR	Bend, OR	88	342	19	103	199012	284093	76034
EUG	RDM	Eugene, OR	Bend, OR	11	72	4	103	199010	284093	76034
MFR	RDM	Medford, OR	Bend, OR	0	18	1	156	199002	147300	76034
MFR	RDM	Medford, OR	Bend, OR	11	18	1	156	199003	147300	76034
MFR	RDM	Medford, OR	Bend, OR	2	72	4	156	199001	147300	76034
MFR	RDM	Medford, OR	Bend, OR	7	18	1	156	199009	147300	76034
MFR	RDM	Medford, OR	Bend, OR	7	36	2	156	199011	147300	76034
SEA	RDM	Seattle, WA	Bend, OR	8	18	1	228	199002	5154164	76034
SEA	RDM	Seattle, WA	Bend, OR	453	3128	23	228	199001	5154164	76034
SEA	RDM	Seattle, WA	Bend, OR	784	2720	20	228	199002	5154164	76034
SEA	RDM	Seattle, WA	Bend, OR	749	2992	22	228	199003	5154164	76034
SEA	RDM	Seattle, WA	Bend, OR	11	18	1	228	199004	5154164	76034
PDX	RDM	Portland, OR	Bend, OR	349	851	23	116	199001	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	1376	2898	161	116	199001	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	444	1110	30	116	199010	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	1949	3261	187	116	199006	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	381	814	22	116	199002	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	1559	2772	154	116	199002	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	1852	3600	200	116	199010	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	483	925	25	116	199009	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	1965	3492	194	116	199009	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	494	1036	28	116	199006	1534762	76034
PDX	RDM	Portland, OR	Bend, OR	459	962	26	116	199003	1534762	76034

6.2 Experiments, Plots and Interpretations:

6.2.1. GMap Plotting of the Domestic Airports in the United States:

Inorder to plot the maps, we first need to Convert the Origin City to a unique list of city and corresponding state. We use:

```
list <- unique(strsplit(as.character(flights$Origin.City), ","))
origin<-ldply(list)
colnames(origin)<-c("City", "State")
```

Similarly, we need to convert the Destination City to a unique list of city and corresponding state.

```
list <- unique(strsplit(as.character(flights$Destination.City), ","))
destination<-ldply(list)
colnames(destination)<-c("City", "State")
```

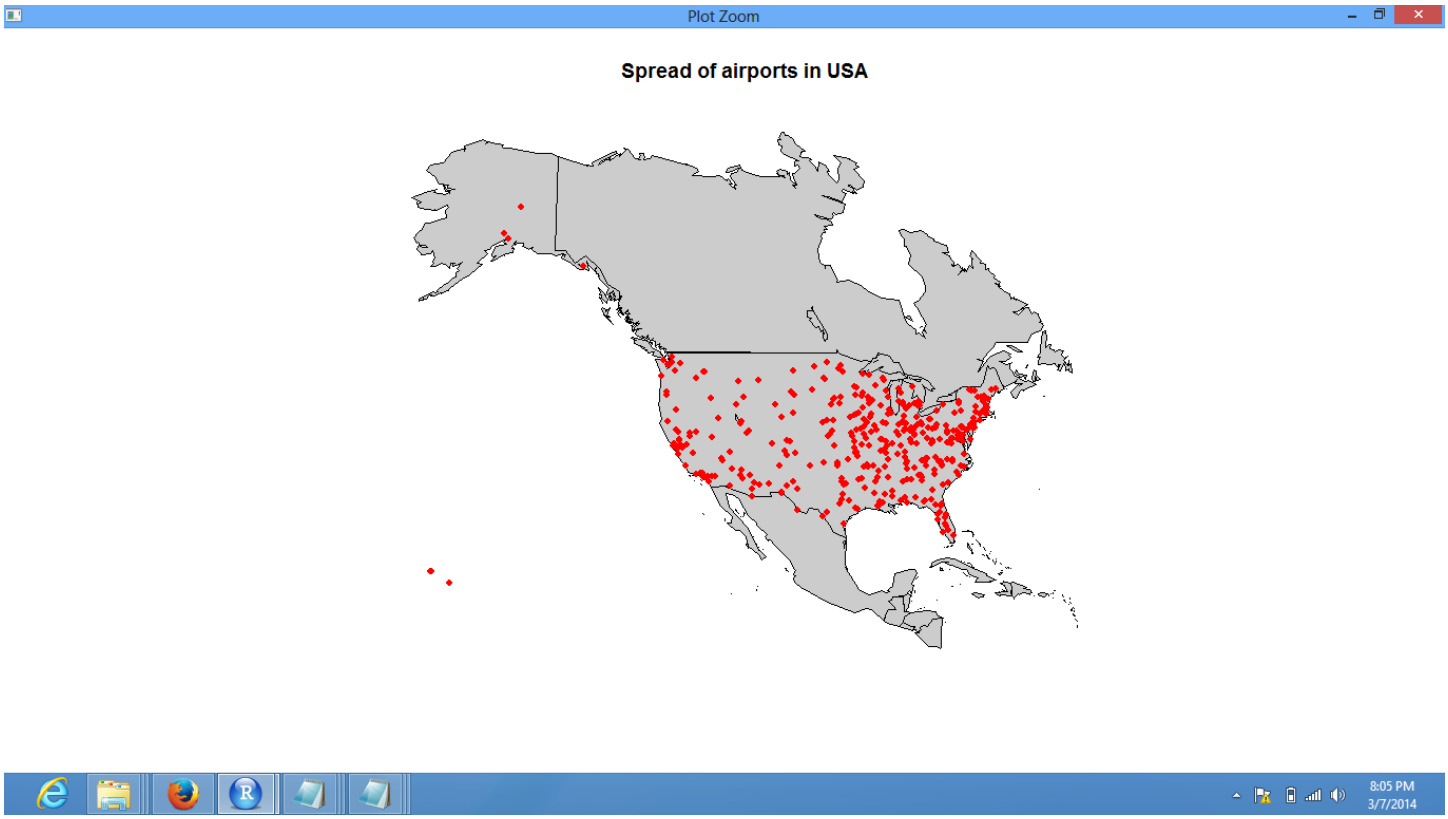
Now we extracted the latitude and longitude of the airports by calling the API like a web service and getting the value since it is not available readily in the dataset. However, this is a one time job. We can just fetch the latitude and longitude details and persist it and need to be updated when a new airport is added or introduced.
Therefore we use the ‘stringr’ library.

```
library(stringr)
ads1 <- unique(origin$City)
ads2 <- unique(origin$State)
ads <- paste(ads1, ads2,"USA",sep = ",")
ads <- str_trim(ads)
lonlat<-geocode(ads)
origin<-cbind(origin, lonlat)
```

We then plot the map using the map() method.

```
map(database= "world", xlim = c(-139.3, -58.8), ylim = c(13.5, 55.7), col="grey80",
fill=TRUE, projection="gilbert", orientation= c(90,0,225))
title("Spread of airports in USA")
lon<-lonlat$lon
lat<-lonlat$lat
coord <- mapproject(lon, lat, proj="gilbert", orientation=c(90, 0, 225))
points(coord, pch=20, cex=1.2, col="red")
```

Here we note that all the airports have been plotted in Red. Another observation which can be drawn is that the concentration of airports in the east coast is slightly more than the concentration in the west coast. The reason being that in the east coast there are more number of metropolitan cities.



6.2.2. Demand Distribution Graph:

Now we plot the distribution of demand and need of all the domestic airports.
We calculate demand on the basis of the ratio of passengers to available seats. We categorized demands as:

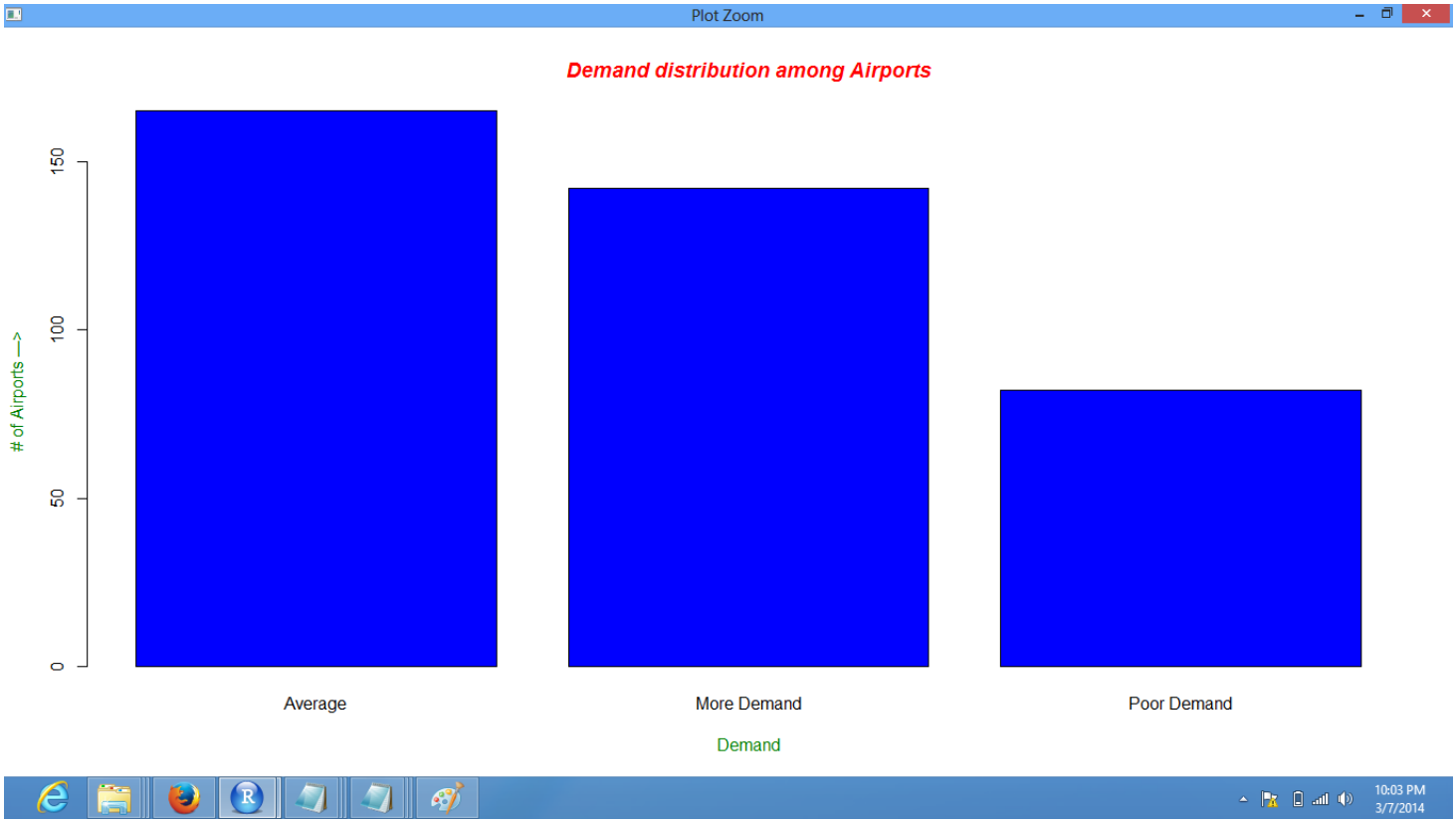
- More Demand: It means that the passenger to seat ratio approaches 1. It means that the number of passengers are meeting the number of allocated seats and sometimes even exceeding the number of available seats. This means that there needs to be an increased number of flights between the airports. We have provided the threshold to be [0, 0.6]. Hence there is more demands of such flights between the two airports.
- Average Demand: It means that the available seats are considerably greater than the number of passengers per flight. The range for passenger to seat ratio is defined to be between (0.3, 0.6). It suggests that there is adequate demand and supply of the flights and no actions are required in order to increase the profits.
- Less Demand: When the flight almost flies empty, that is when the passenger to available seats is very very low, there is wastage of fuel and money for the airlines as there is no need for such flights to take off. Thus when the passenger to flight ratio becomes less than 0.3, we consider that the demand for such flight is very less.

Using R, we coded the following and categorized the demands

```
#create categories
pas_seat_ratio$demand[pas_seat_ratio$pasToSeatRatio>=0.6]<-"More Demand"
pas_seat_ratio$demand[pas_seat_ratio$pasToSeatRatio>0.3 &
pas_seat_ratio$pasToSeatRatio<0.6]<-"Average"
pas_seat_ratio$demand[pas_seat_ratio$pasToSeatRatio>=0 &
pas_seat_ratio$pasToSeatRatio<=0.3]<-"Less Demand"

#Convert the column to a factor
pas_seat_ratio$demand<-factor(pas_seat_ratio$demand)
```

We have now plotted this Demand distribution among Airports graph through a simple histogram.

```
plot(pas_seat_ratio$demand, type='o', col="blue", ann=FALSE)
title(main="Demand distribution among Airports", col.main="red", font.main=4)
title(ylab="# of Airports --->", col.lab=rgb(0,0.5,0))
title(xlab="Demand", col.lab=rgb(0,0.5,0))
```



There are a total of 367 airports out of which 160 airports have Average demand. Similarly, 140 airports have high/more demand and the remaining 67 have less/low demand.

6.2.3. Average number of airports per demand category:

Ultimately after the analysis, the final observation is that the demand or the Passenger to Seats ratio has been taken care of and more number of flights has been allocated from the airport or the cities from which the ratio of Passenger to Seats is more and this is an expected behavior to assess the areas of varying demand distributions.

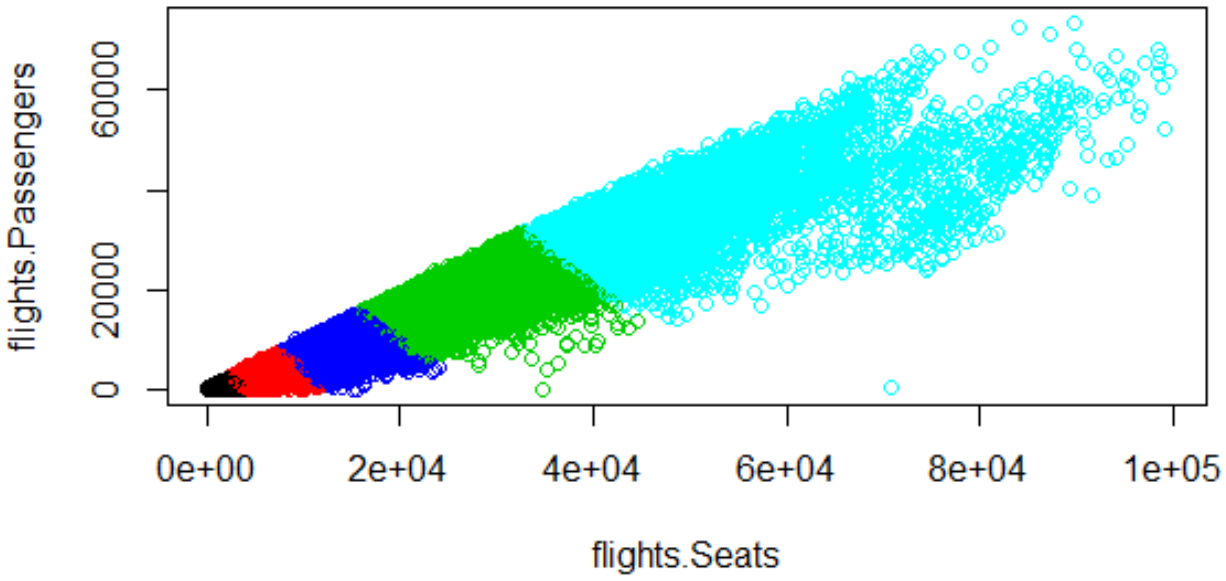
This was confirmed by the summaryBy() method:
The output is shown below:

```
mean_Flights
       demand FlightRatio.mean
1     Average          29.29924
2 More Demand          34.47161
3 Poor Demand          16.77001
```

6.2.4. K-Means Clustering of Available Seats to the number of Passengers:

Here we observe that we had made 5 clusters based on the Passengers to Seat ratio. Here we can conclude that as the Number of Passengers increases, seats also increase and thus, such coordinates of passengers and flights have been clustered into one. There are a few exceptions in the case but even those where handled by the kmeans() method.
We used the following code:

```
library(ggplot2)
flights$ <-cut(data1$Age,c(-Inf,0,18,24,34,44,54,64,Inf))
x <- data.frame(flights$Seats, flights$Passengers)

mat = as.matrix(x)

kclus <- kmeans(mat,5)

#plotcluster(mat,kclus$cluster)

plot(mat,col = kclus$cluster)
```



7. Lessons Learned:

One thing I would like to say about this project is, it is a completely different experience and one of its kind since this project emphasized more on our analytical skill rather than our technical capability.

We learnt mostly about looking into the data and drawing useful insights from it. We used to learn more about the tools to analyze the data and process it but this project thought us the importance of those tools and why exactly they are used and what is the importance of that data the tools are used to process.

Deriving some observable patterns from the data and drawing some conclusions from it and also thinking in the business perspective was a biggest challenge since we were not used to it. I think it added more capability for us to even design and develop the tools since now we are aware of how it is used in the real business scenarios.

Skills Added: R Programming, Using open source libraries, Data Analysis and Interpretation, Machine Learning Algorithms