RootMetrics Coding Challenge

The dummy data set contains 5 variables and 18743 observations. It contains 4 metrics (download speed, upload speed, sms speed, dropped calls) of 4 different carriers (A, B, C, D). We can also notice that the first three speed metrics are numeric varibles and "dropped_call" is a binary(categorical) variable. For this coding practice, I visualized my findings using R programming environment, with the help of two packages: ggplot2 and gridExtra. Below, I first list the steps taken to read the data, to check the data structure and to get number of missing values in each column.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.4
```

```
#########################   Load Data   #############################
setwd("~/Documents/interview/rootmetrics_coding")
mydata <- read.csv('dummy_data.csv')
str(mydata)
```

```
## 'data.frame':    18743 obs. of  5 variables:
##  $ carrier       : Factor w/ 4 levels "Carrier A","Carrier B",..: 1 1 1 2 2 2 2 2 1 1 ...
##  $ download_speed: int  NA NA NA 47 NA NA NA 585 NA NA ...
##  $ upload_speed  : int  NA NA NA NA 46 NA NA NA NA NA ...
##  $ sms_speed     : int  NA 5494 NA NA NA 3719 NA NA 2275 NA ...
##  $ dropped_call  : int  0 NA 0 NA NA NA 0 NA NA 0 ...
```

```
###############  Check Num of Missing Values in Each Column #########
missing <- apply(is.na(mydata),2,sum)
missing
```

```
##        carrier download_speed   upload_speed      sms_speed   dropped_call
##              0          14041          14111          13963          14114
```

We want to vasualize "download_speed", "upload_speed", "sms_speed", and "dropped_call" on a carrier level. Here, I mainly use 4 different types of plots to represent the metrics.

- Pie Chart: Illustrate numerical proportion.
- Histogram: Represent distribution of numerical data.
- Bar Chart: Compare grouped data.
- Box Plot: Depict groups of numerical data through their quartiles.

Below is a function I wrote to improve the pie charts in *ggplot2* environment. With this function, I can generate more informative pie charts.

```
ggpie <- function(dat, by, totals) {
  ggplot(dat, aes_string(x = factor(1), y=totals, fill=by)) +
  geom_bar(stat='identity', color='black') +
  # removes black borders from legend
  guides(fill=guide_legend(override.aes=list(colour=NA))) +
    coord_polar(theta='y') +
      theme(axis.ticks=element_blank(),
            axis.text.y=element_blank(),
            axis.text.x=element_text(colour='black'),
            axis.title=element_blank()) +
      scale_y_continuous(breaks=cumsum(dat[[totals]]) -
                          dat[[totals]] / 2, labels=dat[[by]])
}
```

Representations/Visualizations of Speed Metrics

Below is a function I use to generate all the descriptive plots for the speed metrics. For each *speed metric* I create 9 different plots to fully describe the data.

- Four pie charts:
  - Get 0.25, 0.5 and 0.75 quantiles for the speed metric in all carriers.
  - For each carrier subset, based on the quantile range(for all carriers) the metric values belong to, we assign a label for each metric value: "< 0.25", "0.25 - 0.5", "0.5 - 0.75" or "> 0.75", e.g. For speed metric "download_speed", we have 25% quantile 3367, median 9691 and 75% quantile 20085.75 in all carriers. For subset of carrier A, if we have a download_speed value of 12857, then it will be labeled as "0.5 - 0.75" because the value sits between 50% and 75% quantiles. This will help us understand the distribution of download_speed of carrier A in a bigger picture.
  - Create a pie plot based on the above labels of the speed metric. Thus we have 4 pie charts for 4 different carriers.
- Three histogram plots:
  - For each *speed metric*, create a histogram plot for each carrier respectively. The vertical line represents the median.
  - Create an overlaid histogram plot with medians for all carriers to compare the distributions of different carriers.
  - Create a density plot with medians for all carriers.
- Bar plot of medians: create a bar plot of medians. The horizonal bash line represents the median of the metric in all carriers.
- Box plot of carriers: depict a summary(quantiles, range, variance) of metric for 4 carriers.

```
Get_Numeric_Description <- function(input, bin){
  ###  Generate descriptive plots for numeric variables   ###############
  ##  input: data set input                                #############
  ##  bin: bin width in histograms                         #############
  ##  Return: 9 figures to describe the speed metrics      #############
  ######################################################################

  ###############  Set Lab Name and Titles   ###########################
  xlab_name <- gsub("_", " ", colnames(input)[2])
  xlab_name <- toupper(xlab_name)
  title1 <- paste0("CARRIER_A ", xlab_name,
```

```
                  " DISTRIBUTION \n IN ALL-CARRIERS QUANTILES")
title2 <- paste0("CARRIER_B ", xlab_name,
                  " DISTRIBUTION \n IN ALL-CARRIERS QUANTILES")
title3 <- paste0("CARRIER_C ", xlab_name,
                  " DISTRIBUTION \n IN ALL-CARRIERS QUANTILES")
title4 <- paste0("CARRIER_D ", xlab_name,
                  " DISTRIBUTION \n IN ALL-CARRIERS QUANTILES")
title5 <- paste0("HISTOGRAM WITH MEDIANS OF ", xlab_name)
title6 <- paste0("OVERLAID HISTOGRAM WITH MEDIANS OF ",xlab_name)
title7 <- paste0("DENSITY PLOT WITH MEDIANS OF ",xlab_name)
title8 <- paste0("BAR PLOT FOR MEDIANS OF ",xlab_name)
title9 <- paste0("BOX PLOT OF ",xlab_name)
###########  Get median of the metric for each carrier    #############
median_data <- aggregate(input[,2], by = list(input$carrier),
                         FUN = 'median')
median_data <- as.data.frame(median_data)
colnames(median_data) <- c("carrier", 'median')
##########  Pie Chart of the distribution of each carrier   ############
quantiles <- quantile(input[,2], probs = seq(0, 1, 0.25))
count_carrier <- function(carrier, quantiles){
  count_carrier <- matrix(0,4,2)
  count_carrier[,1] <- c("<0.25","0.25 - 0.5",'0.5 - 0.75',">0.75")
  for(k in 1:4){
    count_carrier[k,2] <- length(which(quantiles[k]< input[,2] &
                                      input[,2] < quantiles[k+1] &
                                          input[,1] == carrier))
  }
  count_carrier <- as.data.frame(count_carrier)
  count_carrier[,2] <- as.numeric(as.character(count_carrier[,2]))
  colnames(count_carrier) <- c("Overall_Quantiles",'count')
  return(count_carrier)
}
count_A <- count_carrier('Carrier A', quantiles)
count_B <- count_carrier('Carrier B', quantiles)
count_C <- count_carrier('Carrier C', quantiles)
count_D <- count_carrier('Carrier D', quantiles)
plot1 <- ggpie(count_A, by='Overall_Quantiles', totals='count') +
  ggtitle(title1)
plot2 <- ggpie(count_B, by='Overall_Quantiles', totals='count') +
  ggtitle(title2)
plot3 <- ggpie(count_C, by='Overall_Quantiles', totals='count')+
  ggtitle(title3)
plot4 <- ggpie(count_D, by='Overall_Quantiles', totals='count')+
  ggtitle(title4)
###############   Histogram for Each Carrier  #####################
input <- input[order(input[,1]),]
plot5 <- ggplot(input, aes(x = input[ ,2], fill = carrier)) +
  geom_histogram(binwidth=bin, alpha=0.5) +
  ggtitle(title5) + xlab(xlab_name) + ylab("COUNT") +
  geom_vline(data=median_data, aes(xintercept = median,col = carrier),
             linetype="dashed", size=0.5) + facet_grid(.~ carrier)
########  Overlaid Histogram For Each Carrier   ##################
plot6 <- ggplot(input, aes(x = input[,2], fill = carrier)) +
```

```
    geom_histogram(binwidth = bin, alpha=0.5, position="identity") +
    ggtitle(title6) + xlab(xlab_name) + ylab("COUNT") +
    geom_vline(data=median_data, aes(xintercept = median, col = carrier),
               linetype="dashed", size=0.5)
  ##############   Density Plot with Medians   #####################
  plot7 <- ggplot(input, aes(x = input[,2], col = carrier)) +
    geom_density() + ggtitle(title7) + xlab(xlab_name) + ylab("DENSITY") +
    geom_vline(data=median_data, aes(xintercept = median, col=carrier),
               linetype="dashed", size=0.5)
  #############   Bar Plot of Medians of Each Carrier  #############
  plot8 <- ggplot(data=median_data, aes(x=carrier, y=median_data[,2],
                                      fill=carrier,
                                      label = median_data[,2])) +
    geom_bar(stat="identity", width=.5) +
    geom_text(position = position_dodge(0.9), vjust = -0.25) +
    geom_hline(yintercept = median(input[,2]),
               linetype="dashed", size=0.5) +
    ylab(paste(xlab_name,'MEDIAN')) + xlab('CARRIER') + ggtitle(title8)
  #####################  Box Plot of Each Carrier  ##################
  plot9 <- ggplot(input, aes(x=carrier, y=input[,2], fill=carrier)) +
    geom_boxplot() + ggtitle(title9) + ylab(xlab_name) +
    xlab("CARRIER")
  return(list(plot1,plot2,plot3,plot4,plot5,plot6,plot7,plot8,plot9))
}
```

Then we can run above function on our speed metrics to obtain the descriptive plots. Before we do so, for each metric we need to do outlier checking first. I define the threshold of outliers by 3*IQR method, where IQR = 75% quantile - 25% quantile. If the metric value > 3*IQR + 75% quantile or less than 25% quantile - 3*IQR, then we can say that the metirc value is an outlier.

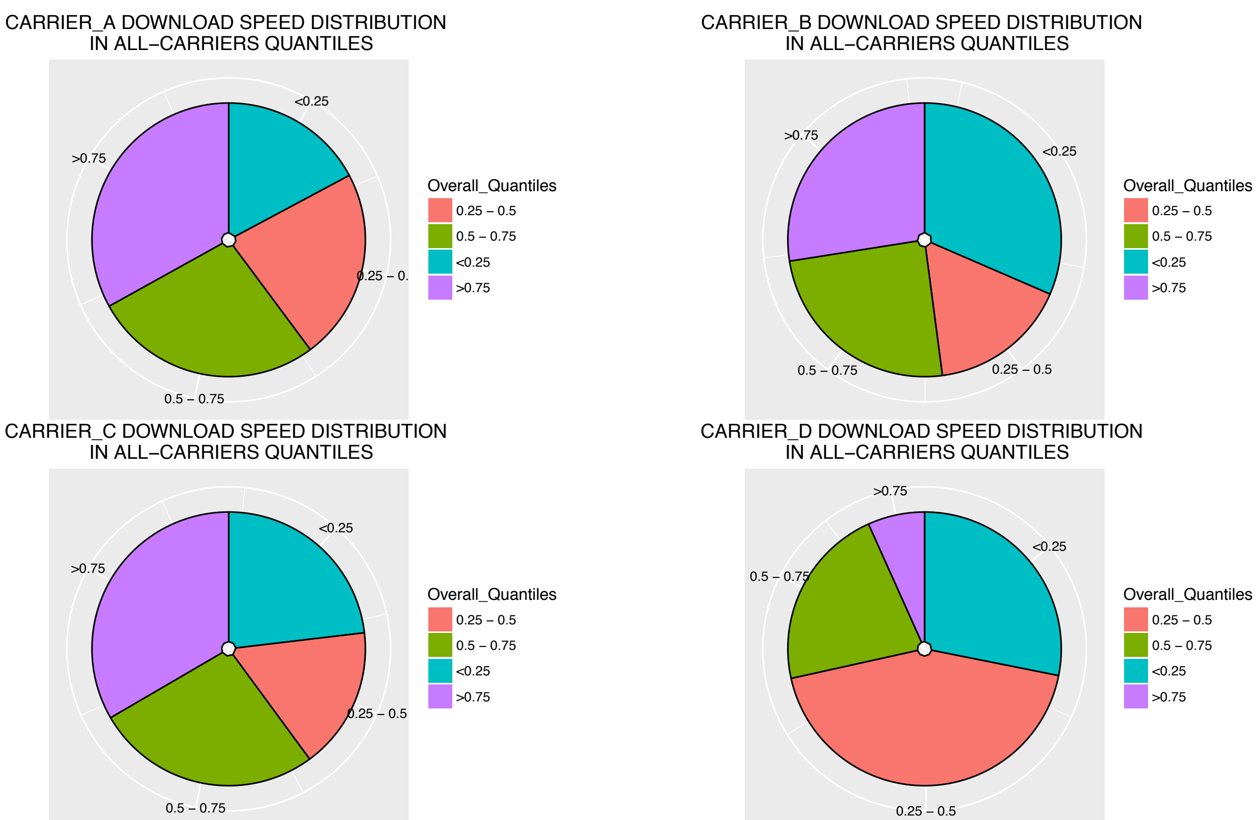
- If a metric has less than 20 outliers, then we can simply remove the outliers and build descriptive plots on the remain.
- If a metric has over 20 outliers, then we'll implement above function on both outlier data set and the data set without outliers(so we get 18 plots for that metric).

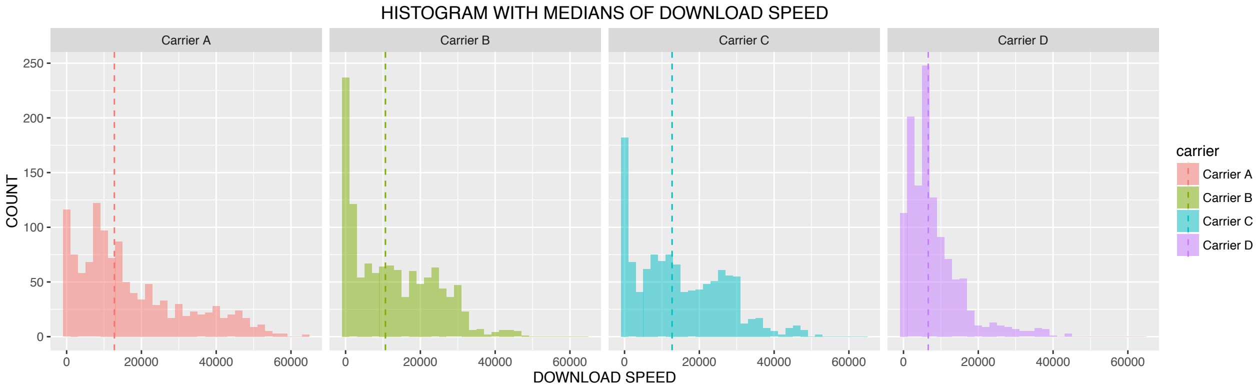
We find that sms_speed has 138 outliers, so I implement above function on both sms_speed outliers and the sms_speed data after removing outliers. The other speed metrics all have less than 20 outliers so the above function is only run on the data set without outliers.

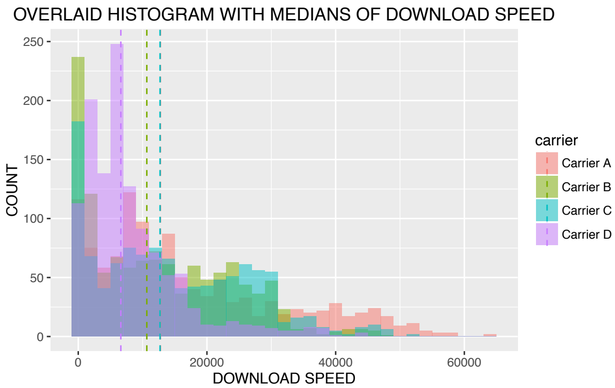
```
##########  Generate Description Plots for Each Int Variable ########
for(i in 2:4){
  cat("\n")
  print(paste0("VISUALIZATIONS OF ", toupper(colnames(mydata)[i])))
  mydata_fix <- mydata[which(is.na(mydata[,i]) == FALSE), c(1,i)]
  ###################  Get Outliers by 3*IQR   #####################
  ##########  IQR = 75% quantile - 25% quantile    ##################
  lowerq = quantile(mydata_fix[,2])[2]
  upperq = quantile(mydata_fix[,2])[4]
  iqr = upperq - lowerq
  threshold.upper = (iqr * 3) + upperq
  threshold.lower = lowerq - (iqr * 3)
  outliers <- mydata_fix[which(mydata_fix[,2] > threshold.upper),]
  colnames(outliers)[2] = paste0(colnames(outliers)[2],' Outliers')
```

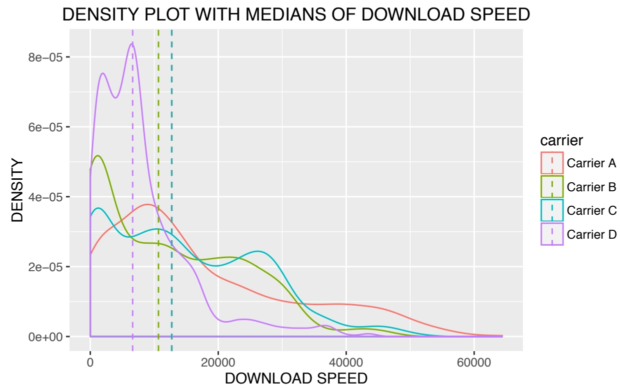
```
  ###############  Remove Outliers and Generate Plots  ##############
  mydata_fix <- mydata_fix[which(mydata_fix[,2] < threshold.upper),]
  plots <- Get_Numeric_Description(mydata_fix,2000)
  plot1_2 <- grid.arrange(plots[[1]], plots[[2]], ncol = 2)
  plot3_4 <- grid.arrange(plots[[3]], plots[[4]], ncol = 2)
  plot(plots[[5]])
  plot6_7 <- grid.arrange(plots[[6]], plots[[7]], ncol = 2)
  plot8_9 <- grid.arrange(plots[[8]], plots[[9]], ncol = 2)
  #####  Generate Plots for Outliers if Num of Outliers > 20  #######
  if(nrow(outliers) > 20){
     cat("\n")
     print(paste0("VISUALIZATIONS OF ",
                  toupper(colnames(mydata)[i]), " OUTLIERS"))
     plots <- Get_Numeric_Description(outliers,5000)
     plot1_2 <- grid.arrange(plots[[1]], plots[[2]], ncol = 2)
     plot3_4 <- grid.arrange(plots[[3]], plots[[4]], ncol = 2)
     plot(plots[[5]])
     plot6_7 <- grid.arrange(plots[[6]], plots[[7]], ncol = 2)
     plot8_9 <- grid.arrange(plots[[8]], plots[[9]], ncol = 2)
  }
}
```

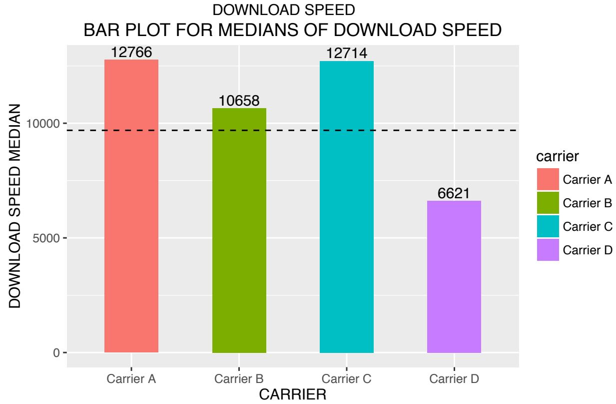
```
##
## [1] "VISUALIZATIONS OF DOWNLOAD_SPEED"
```

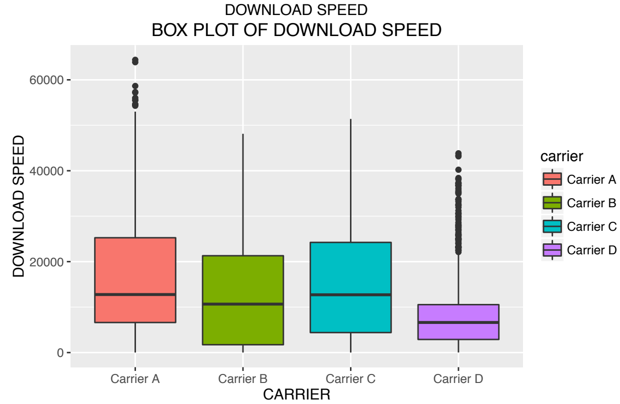




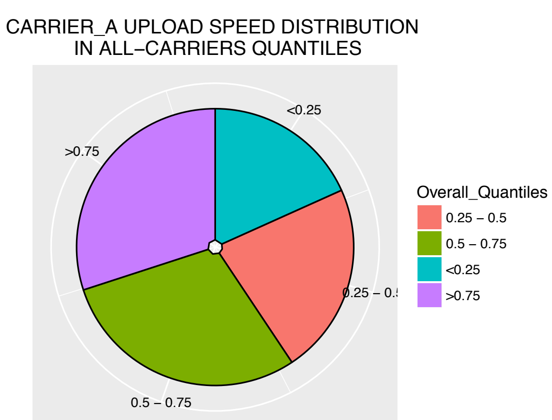


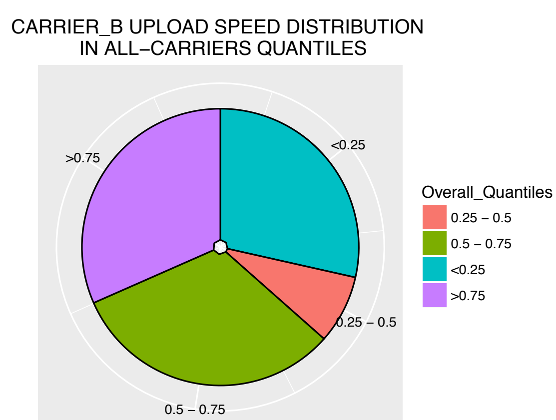


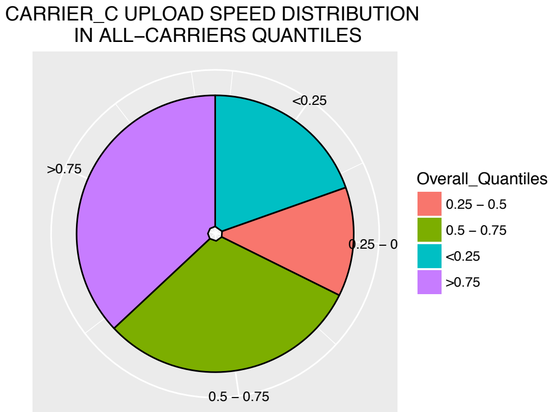


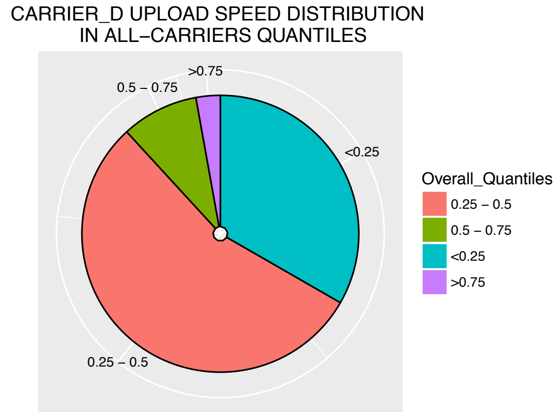


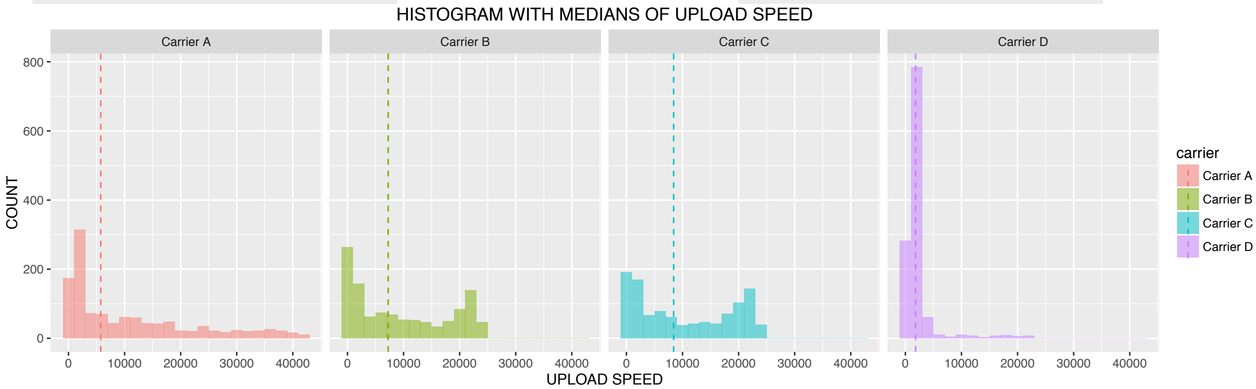

```
##
## [1] "VISUALIZATIONS OF UPLOAD_SPEED"
```

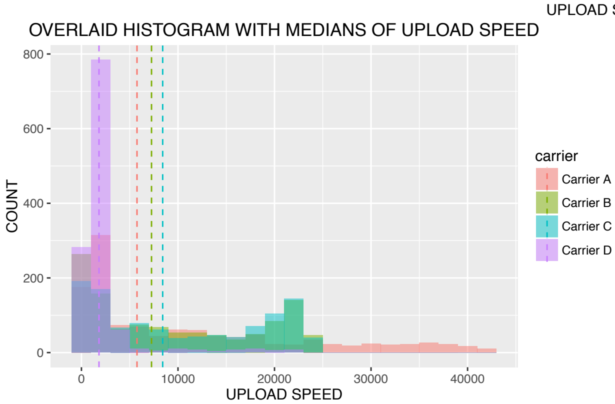


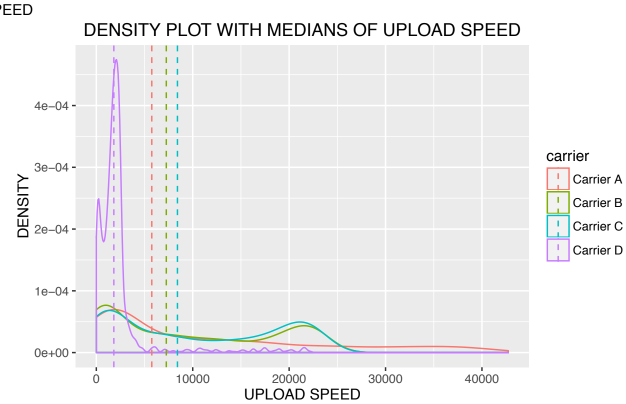


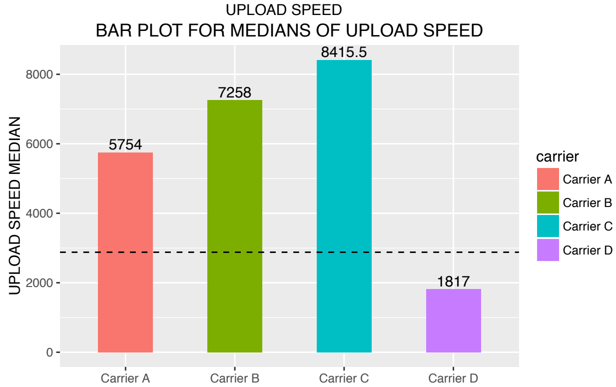


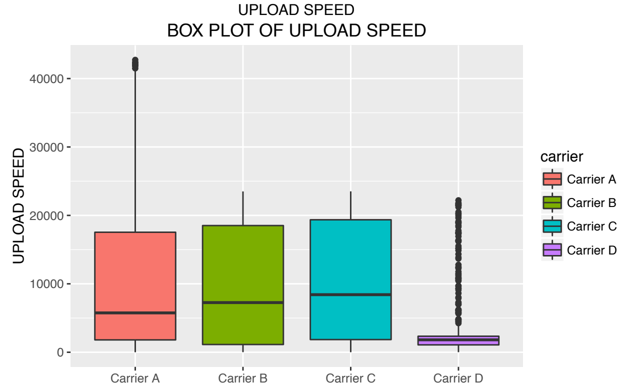




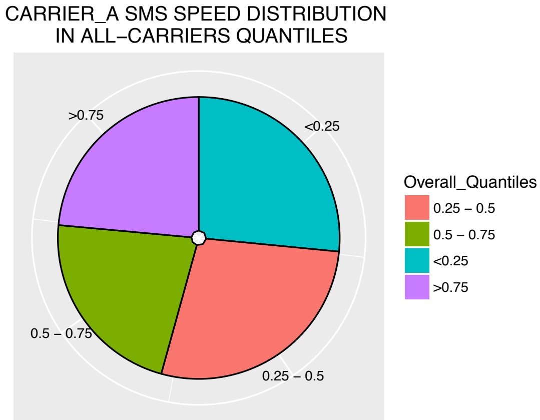


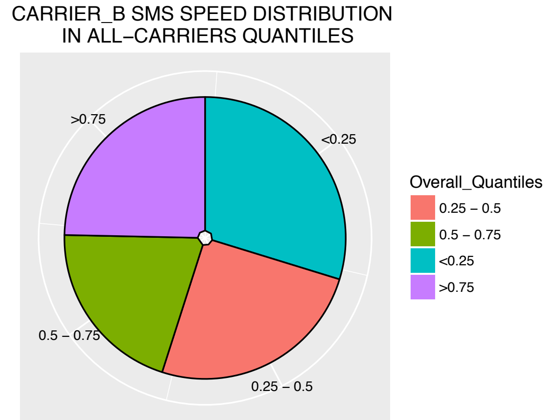


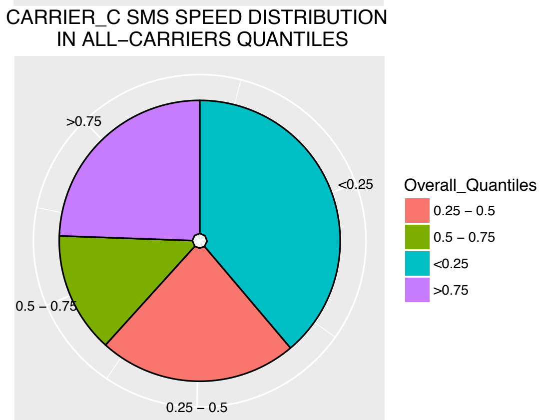


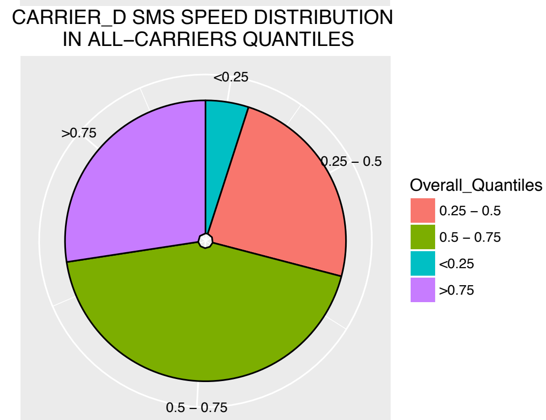


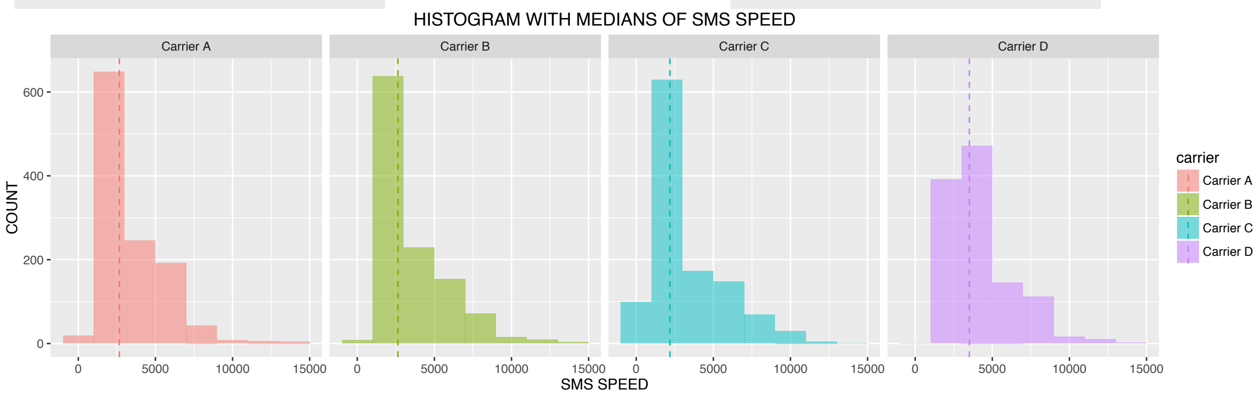

```
##
## [1] "VISUALIZATIONS OF SMS_SPEED"
```

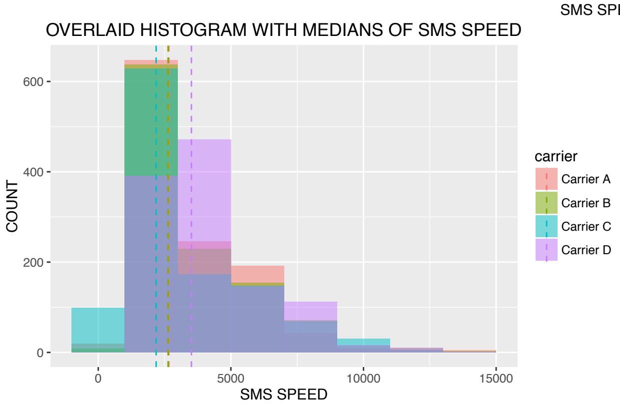

CARRIER_A SMS SPEED DISTRIBUTION
IN ALL–CARRIERS QUANTILES
>0.75
<0.25
0.5 – 0.75
0.25 – 0.5
Overall_Quantiles
0.25 – 0.5
0.5 – 0.75
<0.25
>0.75

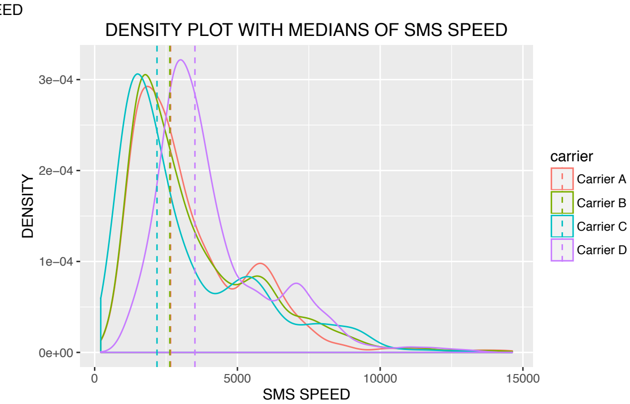

CARRIER_B SMS SPEED DISTRIBUTION
IN ALL–CARRIERS QUANTILES
>0.75
<0.25
0.5 – 0.75
0.25 – 0.5
Overall_Quantiles
0.25 – 0.5
0.5 – 0.75
<0.25
>0.75

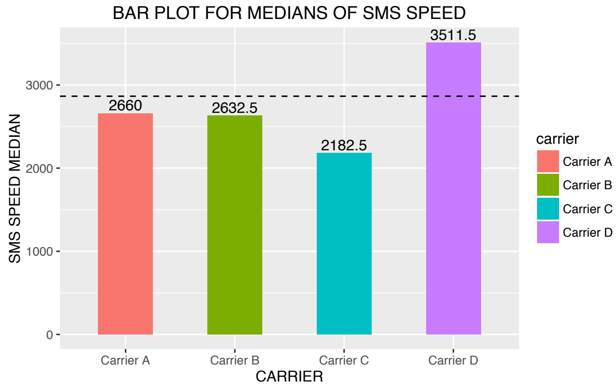

CARRIER_C SMS SPEED DISTRIBUTION
IN ALL–CARRIERS QUANTILES
>0.75
<0.25
0.5 – 0.75
0.25 – 0.5
Overall_Quantiles
0.25 – 0.5
0.5 – 0.75
<0.25
>0.75

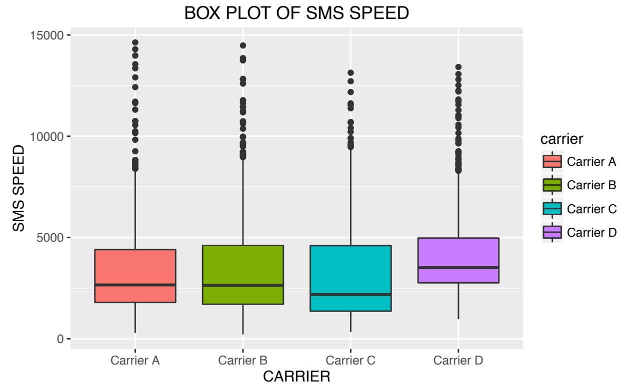

CARRIER_D SMS SPEED DISTRIBUTION
IN ALL–CARRIERS QUANTILES
<0.25
>0.75
0.25 – 0.5
0.5 – 0.75
Overall_Quantiles
0.25 – 0.5
0.5 – 0.75
<0.25
>0.75


HISTOGRAM WITH MEDIANS OF SMS SPEED
Carrier A
Carrier B
Carrier C
Carrier D
COUNT
600
400
200
0
0
5000
10000
15000
SMS SPEED
carrier
Carrier A
Carrier B
Carrier C
Carrier D

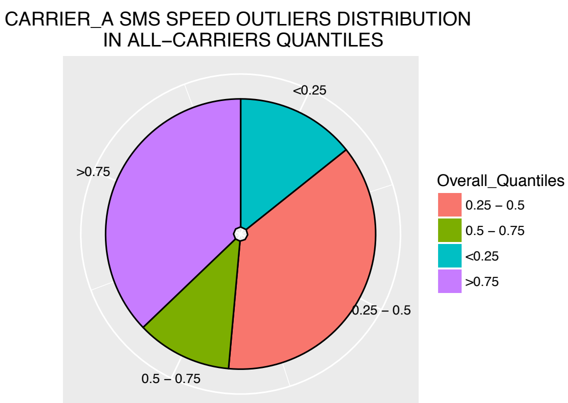

OVERLAID HISTOGRAM WITH MEDIANS OF SMS SPEED
COUNT
600
400
200
0
0
5000
10000
15000
SMS SPEED
carrier
Carrier A
Carrier B
Carrier C
Carrier D

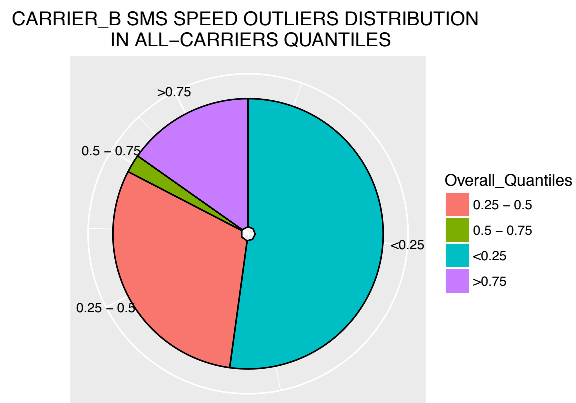

DENSITY PLOT WITH MEDIANS OF SMS SPEED
DENSITY
3e–04
2e–04
1e–04
0e+00
0
5000
10000
15000
SMS SPEED
carrier
Carrier A
Carrier B
Carrier C
Carrier D

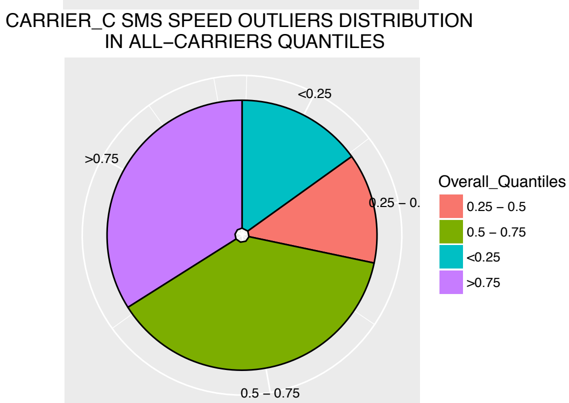


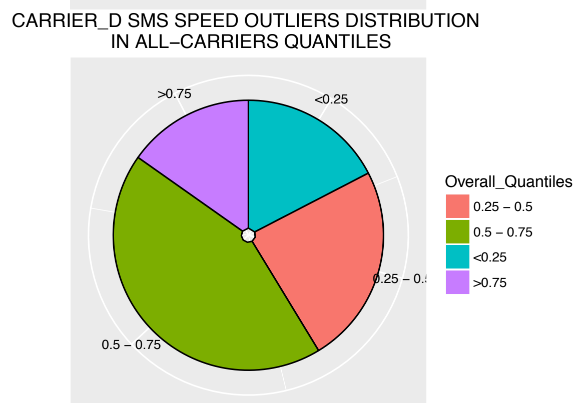


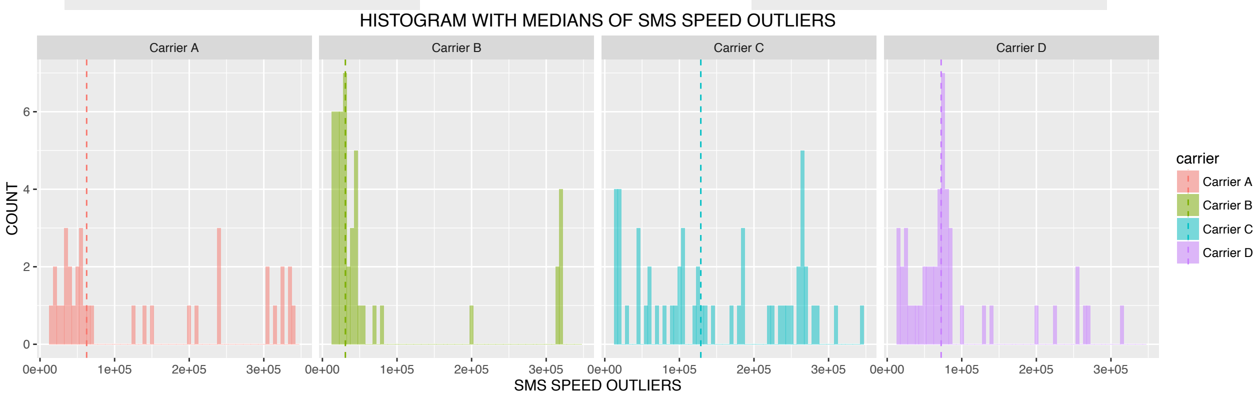

```
##
## [1] "VISUALIZATIONS OF SMS_SPEED OUTLIERS"
```

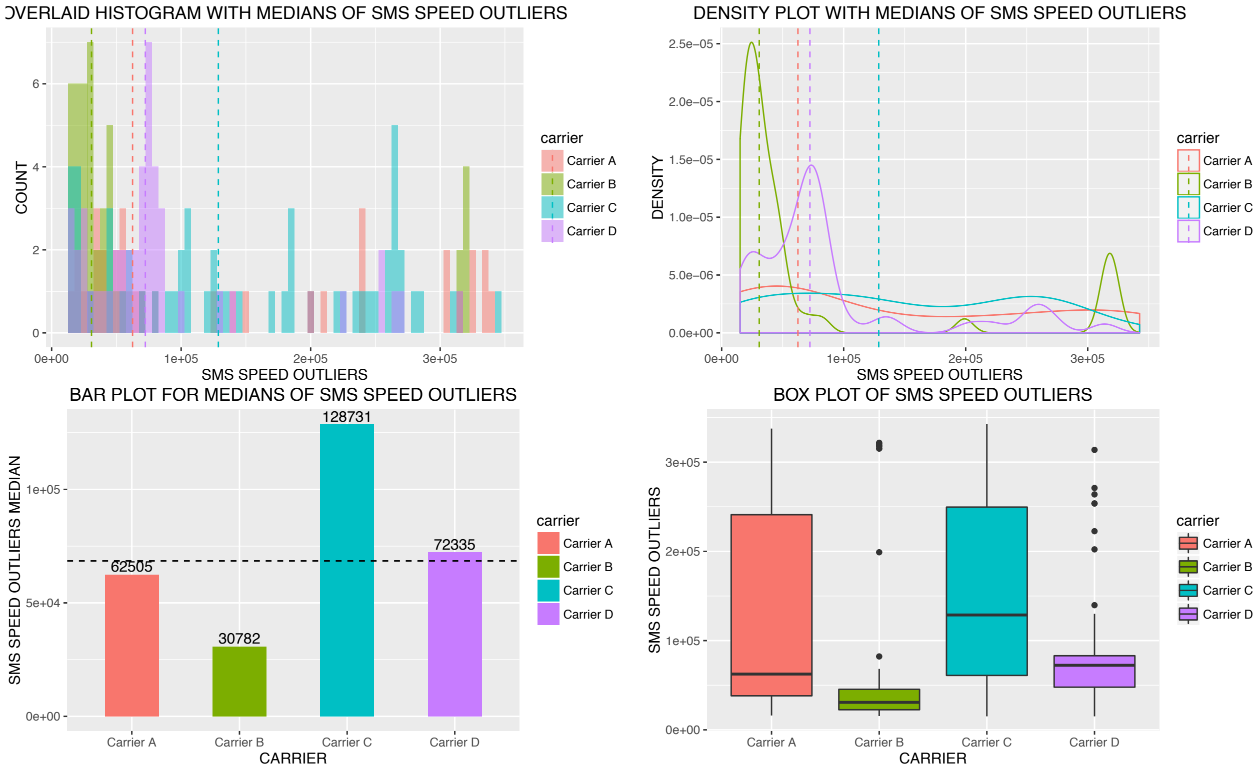












Representations of Binary Metric(“dropped call”)

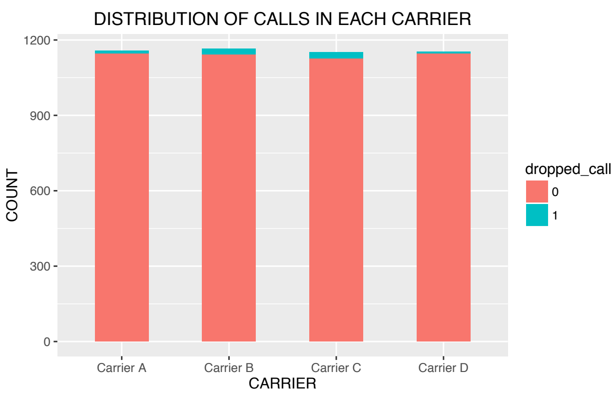
For the binary metric “dropped call”, I create two plots based on the frequency table of “dropped call”. First, I use below queries to generate a frequency table with percentage.

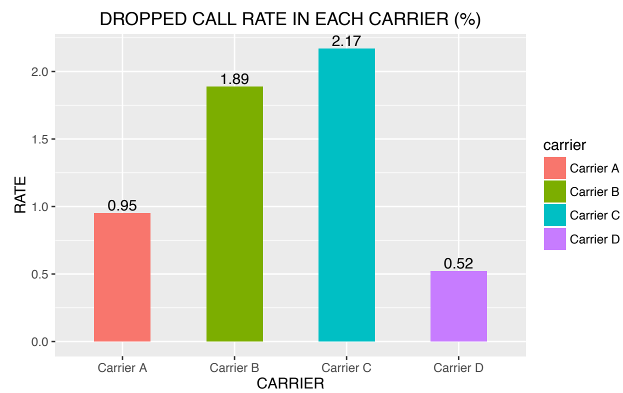
```
#####################  Get Percentage of dropped call in Each Carrier   ##################
mydata_fix <- mydata[which(is.na(mydata[,5]) == FALSE), c(1,5)]
freq_table <- aggregate(mydata_fix, by = list(mydata_fix$carrier,mydata_fix$dropped_call),
                        FUN = length)
freq_table <- freq_table[,1:3]
freq_table[,2] <- as.factor(freq_table[,2])
tmp_table <- aggregate(freq_table[,3], by = list(freq_table[,1]),
          FUN = prop.table)[,2]
percent <- c(tmp_table[,1],tmp_table[,2]) * 100
freq_table[,4] <- percent
colnames(freq_table) <- c("carrier", "dropped_call", "freq","rate")
freq_table
```

```
##     carrier dropped_call freq       rate
## 1 Carrier A            0 1146 99.0492653
## 2 Carrier B            0 1144 98.1132075
## 3 Carrier C            0 1127 97.8298611
## 4 Carrier D            0 1148 99.4800693
## 5 Carrier A            1   11  0.9507347
## 6 Carrier B            1   22  1.8867925
## 7 Carrier C            1   25  2.1701389
## 8 Carrier D            1    6  0.5199307
```

I create two bar plots to vasualize the "dropped call" metric. One represents the distribution of calls in each carrier. The other compares the dropped call rate in each carrier.

```
##############  DISTRIBUTION OF CALLS IN EACH CARRIER  #########################
plot10 <- ggplot(freq_table,aes(x=carrier,y=freq,fill=dropped_call), color=dropped_call) +
  stat_summary(fun.y=mean,position="stack",geom="bar",width = 0.5) +
  ggtitle("DISTRIBUTION OF CALLS IN EACH CARRIER") +
  xlab("CARRIER") + ylab("COUNT")
##############   PERCENTAGE OF DROPPED CALL IN EACH CARRIER     ######################
plot11 <- ggplot(freq_table[5:8,],aes(x = carrier, y = rate, fill = carrier)) +
  geom_bar(stat = "identity", width=.5) +
  geom_text(aes(label=round(rate,2)), position = position_dodge(0.9), vjust = -0.25) +
  ggtitle("DROPPED CALL RATE IN EACH CARRIER (%)") +
  xlab("CARRIER") + ylab("RATE")
plot10_11 <- grid.arrange(plot10, plot11, ncol = 2)
```





Summary

Since there are lots of plots to read in this report, I attach my findings beside each graph as a note, you can find the contents by clicking the note. Please let me know if it does not render well in your pdf reader. Below is a summary of conclusions from above visualizations:

- Carrier A: High download speed, Intermediate upload speed, Intermediate sms speed, Low dropped call rate.
- Carrier B: Intermideate download speed, High upload speed, Intermediate sms speed, High dropped call rate.
- Carrier C: High download speed, High upload speed, Low sms speed, High dropped call rate.
- Carrier D: Low download speed, Low upload speed, High sms speed, low dropped call rate.