Churn Prediction

The churn rate, also known as the rate of attrition, is the percentage of subscribers to a service who discontinue their subscriptions to that service within a given time period. For a company to expand its clientele, its growth rate, as measured by the number of new customers, must exceed its churn rate.

|  |  |
| --- | --- |
| Attribute Name | Description |
| State | categorical, for the 50 states and the District of Columbia |
| VoiceMail Plan | dichotomous categorical, yes or no |
| Account length | integer-valued, how long account has been active |
| Number of voice mail messages | integer-valued |
| Area code | categorical |
| Total day minutes | continuous, minutes customer used service during the day |
| Phone number | essentially a surrogate for customer ID |
| Total day calls | integer-valued |
| International Plan | dichotomous categorical, yes or no |
| Total day charge | continuous, perhaps based on foregoing two variables |
| Total evening minutes | continuous, minutes customer used service during the evening. |
| Total night charge | continuous, perhaps based on foregoing two variables |
| Total evening calls | integer-valued |
| Total international minutes | continuous, minutes customer used service to make international calls. |
| Total evening charge | continuous, perhaps based on foregoing two variables |
| Total international calls | integer-valued |
| Total night minutes | continuous, minutes customer used service during the night |
| Total international charge | continuous, perhaps based on foregoing two variables |
| Total night calls | integer-valued |
| Number of calls to customer service | integer-valued |
| Churn | Label indicating if customer churned |

#-------------------------Package Requirements--------------------------------------------  
#required packages: ggplot2, randomForest, RWeka, dplyr  
installedPackages = installed.packages()  
installedPackages = installedPackages[,1]  
requiredPackages = as.matrix(c('ggplot2','randomForest','RWeka','dplyr'))  
installPackages<-function(package){  
 searchResult<- grep(paste(package,"$",sep = ""),installedPackages)  
 #print(length(searchResult))  
 if(length(searchResult) == 0){  
 print (paste(package,"not installed"))  
 print("Downloading and Installing the package")  
 install.packages(package)  
 }  
}  
loadPackages<-function(package){  
 print (paste("Loading",package))  
 require(package,character.only = TRUE)  
}  
installingPackages <- apply(requiredPackages, 1, installPackages)  
loadingPackages <- apply(requiredPackages, 1, loadPackages)

## [1] "Loading ggplot2"

## Loading required package: ggplot2

## [1] "Loading randomForest"

## Loading required package: randomForest

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## [1] "Loading RWeka"

## Loading required package: RWeka

## [1] "Loading dplyr"

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Check for the Directory of the R language  
getwd()

## [1] "C:/Users/shant/OneDrive/Documents/Fall 2016/Survey of Programming Languages/Churn-Prediction"

#copy the churn\_tel.csv to this location, view the files present in the directory  
dir()

## [1] "boxplots\_CallsChargeMin.R" "catAttrExploration.R"   
## [3] "Churn-Prediction.Rproj" "churn\_tel.csv"   
## [5] "churnPrediction.html" "churnPrediction.pdf"   
## [7] "churnPrediction.Rmd" "contAttrExploration.R"   
## [9] "exploreData.R" "firstforest.csv"   
## [11] "installAndLoadPackages.R" "j48decisionTree.R"   
## [13] "loadData.R" "project\_spl.R"   
## [15] "randomForestTest.R" "README.md"   
## [17] "scatterplot\_ggally.R" "tTest.R"

#import the file in the above directory, then read it here  
churn <- read.csv("churn\_tel.csv")  
# Compactly Display the Structure of churn dataset  
str(churn)

## 'data.frame': 3333 obs. of 21 variables:  
## $ State : Factor w/ 51 levels "AK","AL","AR",..: 17 36 32 36 37 2 20 25 19 50 ...  
## $ Account.Length: int 128 107 137 84 75 118 121 147 117 141 ...  
## $ Area.Code : int 415 415 415 408 415 510 510 415 408 415 ...  
## $ Phone : Factor w/ 3333 levels "327-1058","327-1319",..: 1927 1576 1118 1708 111 2254 1048 81 292 118 ...  
## $ Int.l.Plan : Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 1 2 ...  
## $ VMail.Plan : Factor w/ 2 levels "no","yes": 2 2 1 1 1 1 2 1 1 2 ...  
## $ VMail.Message : int 25 26 0 0 0 0 24 0 0 37 ...  
## $ Day.Mins : num 265 162 243 299 167 ...  
## $ Day.Calls : int 110 123 114 71 113 98 88 79 97 84 ...  
## $ Day.Charge : num 45.1 27.5 41.4 50.9 28.3 ...  
## $ Eve.Mins : num 197.4 195.5 121.2 61.9 148.3 ...  
## $ Eve.Calls : int 99 103 110 88 122 101 108 94 80 111 ...  
## $ Eve.Charge : num 16.78 16.62 10.3 5.26 12.61 ...  
## $ Night.Mins : num 245 254 163 197 187 ...  
## $ Night.Calls : int 91 103 104 89 121 118 118 96 90 97 ...  
## $ Night.Charge : num 11.01 11.45 7.32 8.86 8.41 ...  
## $ Intl.Mins : num 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...  
## $ Intl.Calls : int 3 3 5 7 3 6 7 6 4 5 ...  
## $ Intl.Charge : num 2.7 3.7 3.29 1.78 2.73 1.7 2.03 1.92 2.35 3.02 ...  
## $ CustServ.Calls: int 1 1 0 2 3 0 3 0 1 0 ...  
## $ Churn. : Factor w/ 2 levels "False.","True.": 1 1 1 1 1 1 1 1 1 1 ...

#Names of all the attributes in the data set  
names(churn)

## [1] "State" "Account.Length" "Area.Code" "Phone"   
## [5] "Int.l.Plan" "VMail.Plan" "VMail.Message" "Day.Mins"   
## [9] "Day.Calls" "Day.Charge" "Eve.Mins" "Eve.Calls"   
## [13] "Eve.Charge" "Night.Mins" "Night.Calls" "Night.Charge"   
## [17] "Intl.Mins" "Intl.Calls" "Intl.Charge" "CustServ.Calls"  
## [21] "Churn."

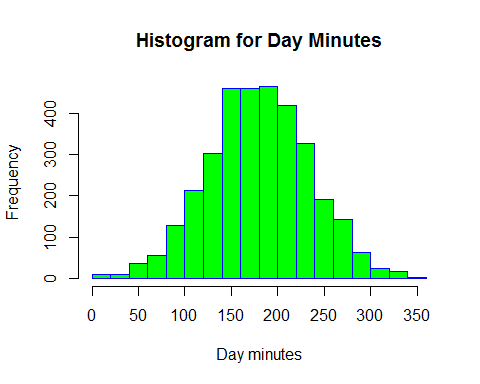
#summary of the dataset  
summary(churn)

## State Account.Length Area.Code Phone Int.l.Plan  
## WV : 106 Min. : 1.0 Min. :408.0 327-1058: 1 no :3010   
## MN : 84 1st Qu.: 74.0 1st Qu.:408.0 327-1319: 1 yes: 323   
## NY : 83 Median :101.0 Median :415.0 327-3053: 1   
## AL : 80 Mean :101.1 Mean :437.2 327-3587: 1   
## OH : 78 3rd Qu.:127.0 3rd Qu.:510.0 327-3850: 1   
## OR : 78 Max. :243.0 Max. :510.0 327-3954: 1   
## (Other):2824 (Other) :3327   
## VMail.Plan VMail.Message Day.Mins Day.Calls   
## no :2411 Min. : 0.000 Min. : 0.0 Min. : 0.0   
## yes: 922 1st Qu.: 0.000 1st Qu.:143.7 1st Qu.: 87.0   
## Median : 0.000 Median :179.4 Median :101.0   
## Mean : 8.099 Mean :179.8 Mean :100.4   
## 3rd Qu.:20.000 3rd Qu.:216.4 3rd Qu.:114.0   
## Max. :51.000 Max. :350.8 Max. :165.0   
##   
## Day.Charge Eve.Mins Eve.Calls Eve.Charge   
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.00   
## 1st Qu.:24.43 1st Qu.:166.6 1st Qu.: 87.0 1st Qu.:14.16   
## Median :30.50 Median :201.4 Median :100.0 Median :17.12   
## Mean :30.56 Mean :201.0 Mean :100.1 Mean :17.08   
## 3rd Qu.:36.79 3rd Qu.:235.3 3rd Qu.:114.0 3rd Qu.:20.00   
## Max. :59.64 Max. :363.7 Max. :170.0 Max. :30.91   
##   
## Night.Mins Night.Calls Night.Charge Intl.Mins   
## Min. : 23.2 Min. : 33.0 Min. : 1.040 Min. : 0.00   
## 1st Qu.:167.0 1st Qu.: 87.0 1st Qu.: 7.520 1st Qu.: 8.50   
## Median :201.2 Median :100.0 Median : 9.050 Median :10.30   
## Mean :200.9 Mean :100.1 Mean : 9.039 Mean :10.24   
## 3rd Qu.:235.3 3rd Qu.:113.0 3rd Qu.:10.590 3rd Qu.:12.10   
## Max. :395.0 Max. :175.0 Max. :17.770 Max. :20.00   
##   
## Intl.Calls Intl.Charge CustServ.Calls Churn.   
## Min. : 0.000 Min. :0.000 Min. :0.000 False.:2850   
## 1st Qu.: 3.000 1st Qu.:2.300 1st Qu.:1.000 True. : 483   
## Median : 4.000 Median :2.780 Median :1.000   
## Mean : 4.479 Mean :2.765 Mean :1.563   
## 3rd Qu.: 6.000 3rd Qu.:3.270 3rd Qu.:2.000   
## Max. :20.000 Max. :5.400 Max. :9.000   
##

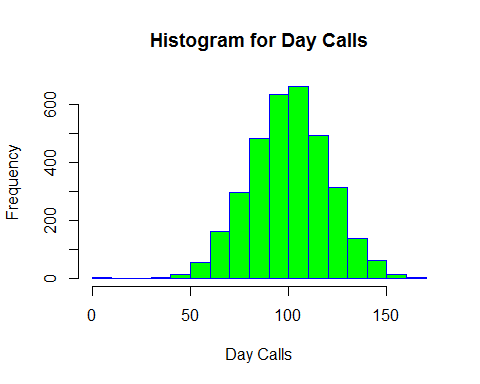
#Omit and row which has missing value(there are none), it returns 0 rows  
churn[!complete.cases(churn),]

## [1] State Account.Length Area.Code Phone   
## [5] Int.l.Plan VMail.Plan VMail.Message Day.Mins   
## [9] Day.Calls Day.Charge Eve.Mins Eve.Calls   
## [13] Eve.Charge Night.Mins Night.Calls Night.Charge   
## [17] Intl.Mins Intl.Calls Intl.Charge CustServ.Calls  
## [21] Churn.   
## <0 rows> (or 0-length row.names)

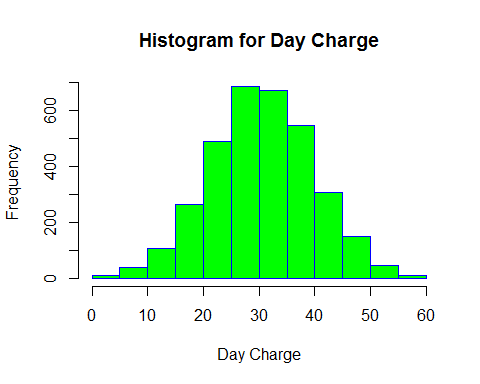
#Histograms (run them one by one)  
  
hist(  
 churn$Day.Mins,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Day Minutes",  
 xlab = "Day minutes"  
)



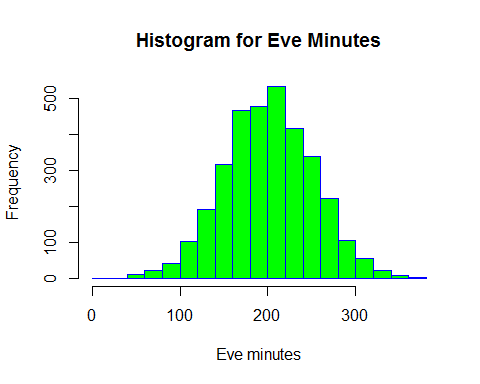
hist(  
 churn$Day.Calls,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Day Calls",  
 xlab = "Day Calls"  
)



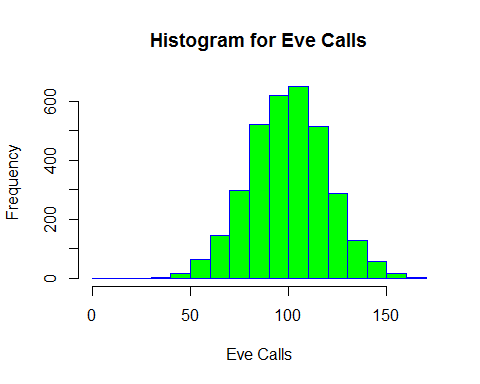
hist(  
 churn$Day.Charge,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Day Charge",  
 xlab = "Day Charge"  
)



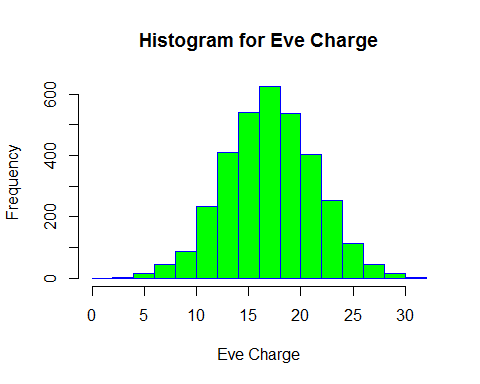
hist(  
 churn$Eve.Mins,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Eve Minutes",  
 xlab = "Eve minutes"  
)



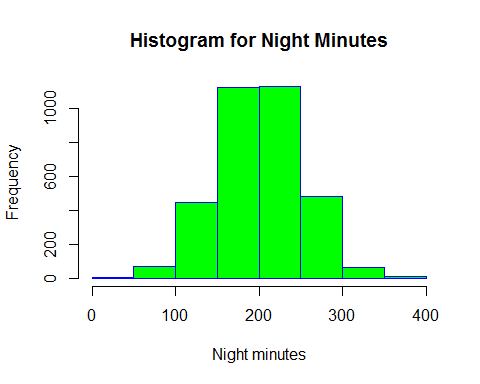
hist(  
 churn$Eve.Calls,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Eve Calls",  
 xlab = "Eve Calls"  
)



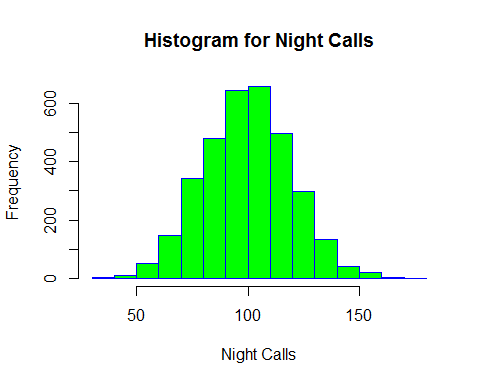
hist(  
 churn$Eve.Charge,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Eve Charge",  
 xlab = "Eve Charge"  
)



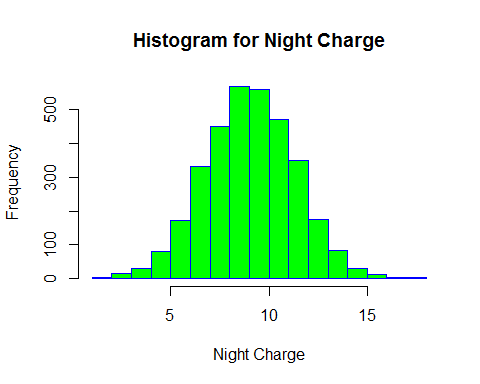
hist(  
 churn$Night.Mins,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Minutes",  
 xlab = "Night minutes"  
)



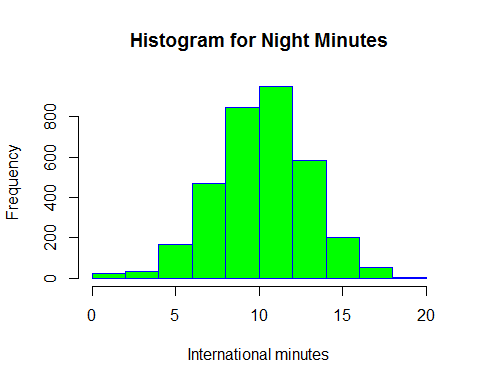
hist(  
 churn$Night.Calls,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Calls",  
 xlab = "Night Calls"  
)



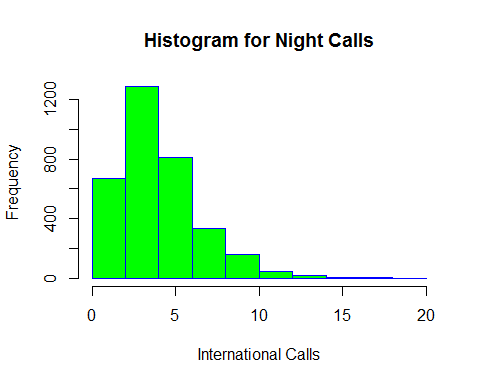
hist(  
 churn$Night.Charge,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Charge",  
 xlab = "Night Charge"  
)



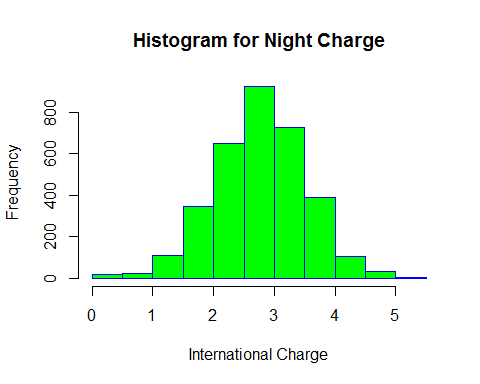
hist(  
 churn$Intl.Mins,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Minutes",  
 xlab = "International minutes"  
)



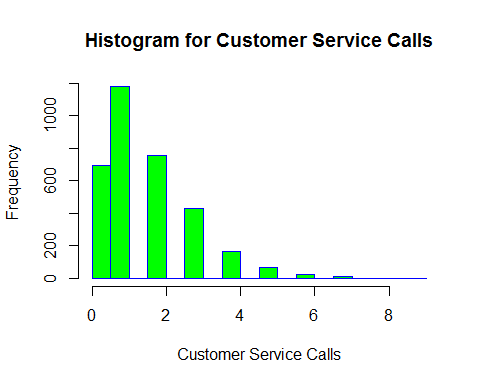
hist(  
 churn$Intl.Calls,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Calls",  
 xlab = "International Calls"  
)



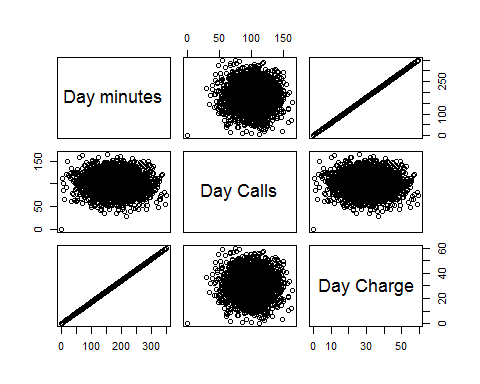
hist(  
 churn$Intl.Charge,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Night Charge",  
 xlab = "International Charge"  
)



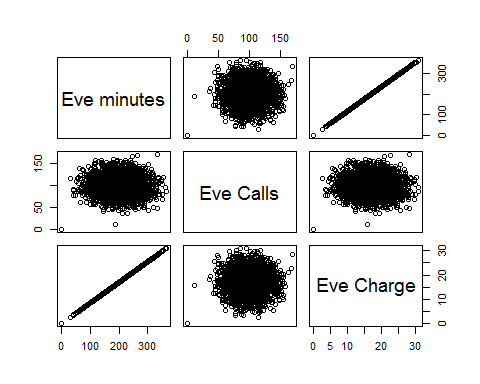
hist(  
 churn$CustServ.Calls,  
 border = "blue",  
 col = "green",  
 main = "Histogram for Customer Service Calls",  
 xlab = "Customer Service Calls"  
)



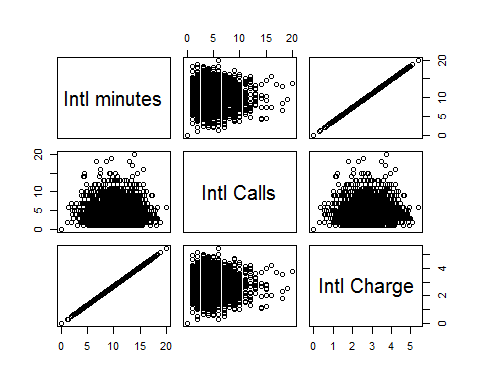
#scatter plot amongst the seemingly similar variables(continuous)  
  
churnScatter1 <- churn[, c("Day.Mins", "Day.Calls", "Day.Charge")]  
colnames(churnScatter1) <-  
 c("Day minutes", "Day Calls", "Day Charge")  
plot(churnScatter1)



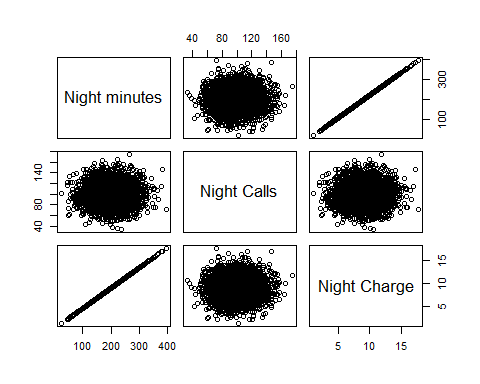
churnScatter2 <- churn[, c("Eve.Mins", "Eve.Calls", "Eve.Charge")]  
colnames(churnScatter2) <-  
 c("Eve minutes", "Eve Calls", "Eve Charge")  
plot(churnScatter2)



churnScatter3 <-  
 churn[, c("Intl.Mins", "Intl.Calls", "Intl.Charge")]  
colnames(churnScatter3) <-  
 c("Intl minutes", "Intl Calls", "Intl Charge")  
plot(churnScatter3)



churnScatter4 <-  
 churn[, c("Night.Mins", "Night.Calls", "Night.Charge")]  
colnames(churnScatter4) <-  
 c("Night minutes", "Night Calls", "Night Charge")  
plot(churnScatter4)



#Correlation matrix (description in the document)  
#On the basis of corelation we eliminate 4 variables, since there were a linear  
#function of other 4 variables  
  
  
cor(churnScatter1[sapply(churnScatter1, is.numeric)])

## Day minutes Day Calls Day Charge  
## Day minutes 1.000000000 0.006750414 0.999999952  
## Day Calls 0.006750414 1.000000000 0.006752962  
## Day Charge 0.999999952 0.006752962 1.000000000

cor(churnScatter2[sapply(churnScatter2, is.numeric)])

## Eve minutes Eve Calls Eve Charge  
## Eve minutes 1.00000000 -0.01143011 0.99999978  
## Eve Calls -0.01143011 1.00000000 -0.01142289  
## Eve Charge 0.99999978 -0.01142289 1.00000000

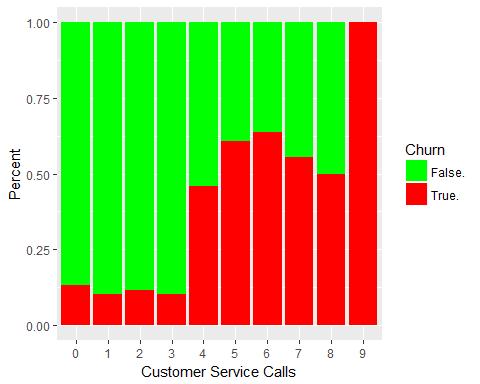
cor(churnScatter3[sapply(churnScatter2, is.numeric)])

## Intl minutes Intl Calls Intl Charge  
## Intl minutes 1.00000000 0.03230388 0.99999274  
## Intl Calls 0.03230388 1.00000000 0.03237215  
## Intl Charge 0.99999274 0.03237215 1.00000000

cor(churnScatter4[sapply(churnScatter2, is.numeric)])

## Night minutes Night Calls Night Charge  
## Night minutes 1.00000000 0.01120386 0.99999921  
## Night Calls 0.01120386 1.00000000 0.01118782  
## Night Charge 0.99999921 0.01118782 1.00000000

#---------------------Graphical Evidence to retain ------------------------------  
#---------------------above variables(Customer Service Call)---------------------  
#Customer Service Calls vs Churn  
  
ggplot() +  
 geom\_bar(data = churn,  
 aes(  
 x = factor(churn$CustServ.Calls),  
 fill = factor(churn$Churn)  
 ),  
 position = "fill") +  
 scale\_x\_discrete("Customer Service Calls") +  
 scale\_y\_continuous("Percent") +  
 guides(fill = guide\_legend(title = "Churn")) +  
 scale\_fill\_manual(values = c("green", "red"))



#Conclusion: Customer Service Calls is predictive of churn

#If the p-value is greater than .1, it will not be predictive of churn  
  
t.test(churn$Intl.Calls ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Intl.Calls by churn$Churn  
## t = 2.9604, df = 640.64, p-value = 0.003186  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.1243807 0.6144620  
## sample estimates:  
## mean in group False. mean in group True.   
## 4.532982 4.163561

#Retain International Calls  
  
t.test(churn$Day.Calls ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Day.Calls by churn$Churn  
## t = -1.0024, df = 627.17, p-value = 0.3165  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.113677 1.009186  
## sample estimates:  
## mean in group False. mean in group True.   
## 100.2832 101.3354

#Eliminate Day Calls Calls  
  
t.test(churn$Night.Calls ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Night.Calls by churn$Churn  
## t = -0.34882, df = 647.95, p-value = 0.7273  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.262875 1.580194  
## sample estimates:  
## mean in group False. mean in group True.   
## 100.0582 100.3996

#Eliminate Night Calls  
  
t.test(churn$Eve.Calls ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Eve.Calls by churn$Churn  
## t = -0.53739, df = 660.41, p-value = 0.5912  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.431568 1.386607  
## sample estimates:  
## mean in group False. mean in group True.   
## 100.0386 100.5611

#Eliminate Eve Calls  
  
t.test(churn$Day.Mins ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Day.Mins by churn$Churn  
## t = -9.6846, df = 571.51, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -38.17516 -25.30148  
## sample estimates:  
## mean in group False. mean in group True.   
## 175.1758 206.9141

#Retain Day Minutes  
  
t.test(churn$Eve.Mins ~ churn$Churn)

##   
## Welch Two Sample t-test  
##   
## data: churn$Eve.Mins by churn$Churn  
## t = -5.2724, df = 645.99, p-value = 1.839e-07  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -18.345214 -8.388479  
## sample estimates:  
## mean in group False. mean in group True.   
## 199.0433 212.4101

#retain Eve minutes  
  
t.test(churn$Night.Mins ~ churn$Churn)

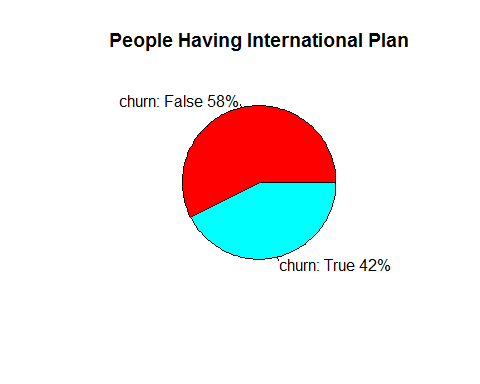
##   
## Welch Two Sample t-test  
##   
## data: churn$Night.Mins by churn$Churn  
## t = -2.1709, df = 688.58, p-value = 0.03028  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -9.7097014 -0.4872667  
## sample estimates:  
## mean in group False. mean in group True.   
## 200.1332 205.2317

#retain Night minutes  
  
t.test(churn$CustServ.Calls ~ churn$Churn)

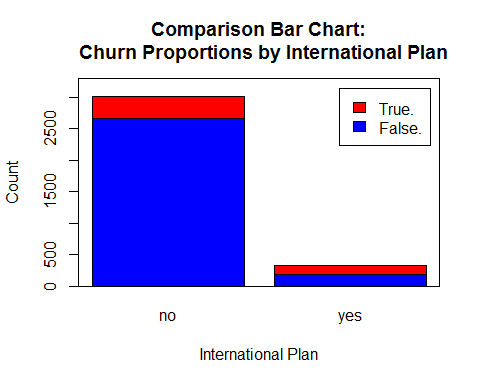
##   
## Welch Two Sample t-test  
##   
## data: churn$CustServ.Calls by churn$Churn  
## t = -8.9551, df = 548.17, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.9510789 -0.6088993  
## sample estimates:  
## mean in group False. mean in group True.   
## 1.449825 2.229814

#retain Customer Service Calls  
  
#conclusion: Retain Intl Calls, Eve minutes, Nigh, Eve, Day Minutes  
#, Customer Service Calls

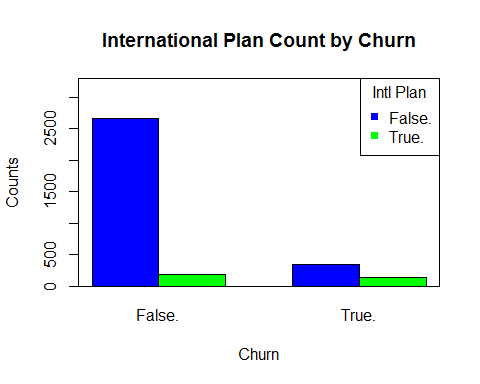
#Intl Plan  
  
#table for counts of Churn and International Plan  
countsIntlPlan <- table(churn$Churn,  
 churn$Int.l.Plan,  
 dnn = c("Churn", "International Plan"))  
  
  
#Pie chart wich shows that people who have international plan may churn  
slices <- c(countsIntlPlan[1, 2] , countsIntlPlan[2, 2])  
lbls <- c("churn: False", "churn: True")  
pct <- round(slices / sum(slices) \* 100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls, "%", sep = "") # ad % to labels  
pie(slices,  
 labels = lbls,  
 col = rainbow(length(lbls)),  
 main = "People Having International Plan")



#Overlayed bar chart  
barplot(  
 countsIntlPlan,  
 legend = rownames(countsIntlPlan),  
 col = c("blue", "red"),  
 ylim = c(0, 3300),  
 ylab = "Count",  
 xlab = "International Plan",  
 main = "Comparison Bar Chart:  
 Churn Proportions by International Plan"  
)  
box(which = "plot",  
 lty = "solid",  
 col = "black")



#Clustered Bar Chart of Churn and Intl Plan with legend  
barplot(  
 t(countsIntlPlan),  
 col = c("blue", "green"),  
 ylim = c(0, 3300),  
 ylab = "Counts",  
 xlab = "Churn",  
 main = "International Plan Count by Churn",  
 beside = TRUE  
)  
legend(  
 "topright",  
 c(rownames(countsIntlPlan)),  
 col = c("blue", "green"),  
 pch = 15,  
 title = "Intl Plan"  
)  
box(which = "plot",  
 lty = "solid",  
 col = "black")



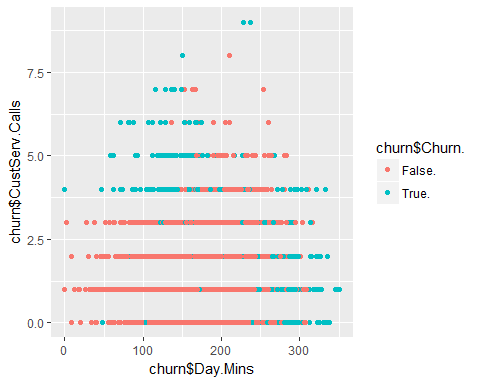
#Vmail Plan  
#weak evidence, but still vmail plan may be predictive  
#'cause we can see int row.margin[2,1] and row.margin[2,2]  
#that the people who dont have the vmail plan and will churn % = 84  
#have vmail and will churn % = 16  
  
countsVmailPlan <- table(churn$Churn, churn$VMail.Plan,  
 dnn = c("Churn", "Vmail Plan"))  
  
  
row.margin <- round(prop.table(countsVmailPlan, margin = 1), 4)\*100  
row.margin

## Vmail Plan  
## Churn no yes  
## False. 70.46 29.54  
## True. 83.44 16.56

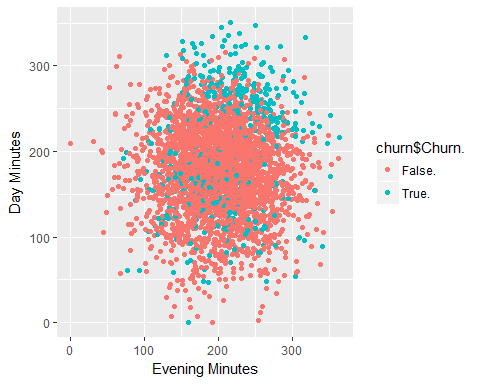
#Vmail message's histogram gives us a spike  
#For the analysis we say that If Voice Mail Messages > 0 then VoiceMailMessages\_Flag = 1;  
#otherwise VoiceMailMessages\_Flag = 0  
#it reveals that it is similar to the vmal plan, hence we can eliminate vmail message  
  
churn$flag[churn$VMail.Message>0] <- 1  
churn$flag[churn$VMail.Message<=0] <- 0  
table(churn$flag,churn$Churn)

##   
## False. True.  
## 0 2008 403  
## 1 842 80

#------------------Multivariate relationships-------------------------------------  
  
#cust serv calls vs day calls  
#Conclusion: hiher the  
qplot(churn$Day.Mins,  
 churn$CustServ.Calls,  
 data = churn,  
 colour = churn$Churn.)



#Day min Vs Eve min  
#conclusion: Higher the day min and evening min, more the churn  
qplot(churn$Eve.Mins,churn$Day.Mins,  
 data = churn,  
 colour = churn$Churn., xlab = "Evening Minutes",  
 ylab= "Day Minutes")



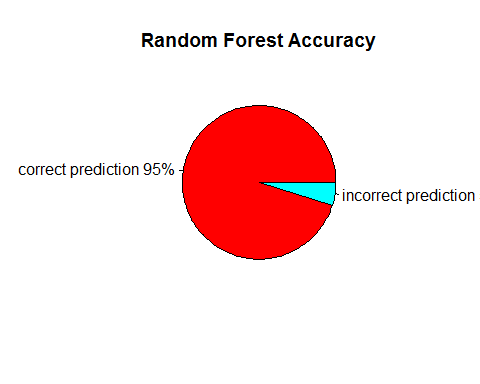
#creates a new file firstforest to test against the test data  
#----------------------------------------------------------------------------------  
  
#churn <- read.csv("churn\_tel.csv")  
churnTrain <- churn[800:3300, ]  
churnTest <- churn[1:500, ]  
  
  
fit <-  
 randomForest(  
 as.factor(Churn.) ~ Int.l.Plan + VMail.Plan + CustServ.Calls + Day.Mins +  
 Eve.Mins + VMail.Message + Night.Mins + Intl.Mins + Intl.Calls,  
 data = churnTrain,  
 importance = TRUE,  
 ntree = 500,  
 nodesize = 3  
 )  
Prediction <- predict(fit, churnTest)  
print(nrow(churnTest))

## [1] 500

submit <- data.frame(id = churnTest$Phone, Churn = Prediction)  
write.csv(submit, file = "firstforest.csv", row.names = FALSE)  
fr <- read.csv("firstforest.csv")  
count <- table(churnTest$Churn, fr$Churn)  
count

##   
## False. True.  
## False. 427 5  
## True. 19 49

#Accuracy of Random Forest(using pie chart)  
  
  
slices <- c(count[1, 1] + count[2, 2], count[1, 2] + count[2, 1])  
lbls <- c("correct prediction", "incorrect prediction")  
pct <- round(slices / sum(slices) \* 100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls, "%", sep = "") # ad % to labels  
pie(slices,  
 labels = lbls,  
 col = rainbow(length(lbls)),  
 main = "Random Forest Accuracy")



#creates a new file decisionTree to test against the test data  
#----------------------------------------------------------------------------------  
  
#churn <- read.csv("churn\_tel.csv")  
churnTrain2 <- churn[800:3300, ]  
churnTest2 <- churn[1:500, ]  
  
  
decisionTree <- J48(`Churn.` ~ ., data = churnTrain2)  
prediction\_tree <- predict(decisionTree, churnTest2)  
count2 <- table(churnTest2$Churn, prediction\_tree)  
count2

## prediction\_tree  
## False. True.  
## False. 418 14  
## True. 19 49

#Accuracy of J48(using pie chart)  
  
  
slices <-  
 c(count2[1, 1] + count2[2, 2], count2[1, 2] + count2[2, 1])  
lbls <- c("correct prediction", "incorrect prediction")  
pct <- round(slices / sum(slices) \* 100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls, "%", sep = "") # ad % to labels  
pie(slices,  
 labels = lbls,  
 col = rainbow(length(lbls)),  
 main = "J48 Decision Tree Accuracy")

