Exploratory.Data.Analysis

By Saba Akram

2/17/2021

churn <- read.csv("D:/M.Sc in Banking and Financial Analytics/Sem 3/Data Analytic and Machine learning/Data Mining and Predictive Analysis/Data sets/churn.txt", stringsAsFactors=TRUE)  
churn[1:10,] # Showing the first 10 records.

## State Account.Length Area.Code Phone Int.l.Plan VMail.Plan VMail.Message  
## 1 KS 128 415 382-4657 no yes 25  
## 2 OH 107 415 371-7191 no yes 26  
## 3 NJ 137 415 358-1921 no no 0  
## 4 OH 84 408 375-9999 yes no 0  
## 5 OK 75 415 330-6626 yes no 0  
## 6 AL 118 510 391-8027 yes no 0  
## 7 MA 121 510 355-9993 no yes 24  
## 8 MO 147 415 329-9001 yes no 0  
## 9 LA 117 408 335-4719 no no 0  
## 10 WV 141 415 330-8173 yes yes 37  
## Day.Mins Day.Calls Day.Charge Eve.Mins Eve.Calls Eve.Charge Night.Mins  
## 1 265.1 110 45.07 197.4 99 16.78 244.7  
## 2 161.6 123 27.47 195.5 103 16.62 254.4  
## 3 243.4 114 41.38 121.2 110 10.30 162.6  
## 4 299.4 71 50.90 61.9 88 5.26 196.9  
## 5 166.7 113 28.34 148.3 122 12.61 186.9  
## 6 223.4 98 37.98 220.6 101 18.75 203.9  
## 7 218.2 88 37.09 348.5 108 29.62 212.6  
## 8 157.0 79 26.69 103.1 94 8.76 211.8  
## 9 184.5 97 31.37 351.6 80 29.89 215.8  
## 10 258.6 84 43.96 222.0 111 18.87 326.4  
## Night.Calls Night.Charge Intl.Mins Intl.Calls Intl.Charge CustServ.Calls  
## 1 91 11.01 10.0 3 2.70 1  
## 2 103 11.45 13.7 3 3.70 1  
## 3 104 7.32 12.2 5 3.29 0  
## 4 89 8.86 6.6 7 1.78 2  
## 5 121 8.41 10.1 3 2.73 3  
## 6 118 9.18 6.3 6 1.70 0  
## 7 118 9.57 7.5 7 2.03 3  
## 8 96 9.53 7.1 6 1.92 0  
## 9 90 9.71 8.7 4 2.35 1  
## 10 97 14.69 11.2 5 3.02 0  
## Churn.  
## 1 False.  
## 2 False.  
## 3 False.  
## 4 False.  
## 5 False.  
## 6 False.  
## 7 False.  
## 8 False.  
## 9 False.  
## 10 False.

sum.churn<-summary(churn$Churn.)  
sum.churn

## False. True.   
## 2850 483

churn.count=0  
for (i in 1:length(churn$Churn.)) {  
 if(churn$Churn.[i]=="True.")churn.count=churn.count+1  
 else(churn.count=churn.count+0)  
   
}  
churn.count

## [1] 483

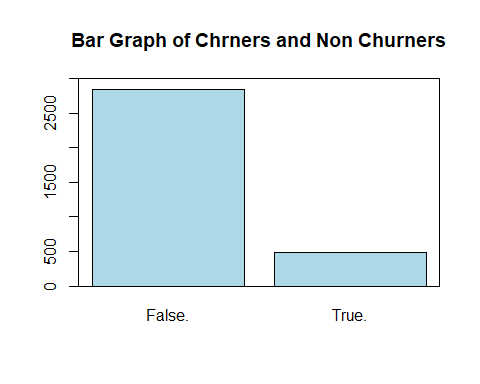
prop.churn<-(churn.count)/length(churn$Churn.)  
prop.churn

## [1] 0.1449145

prop.churn\_or<-sum(churn$Churn.=="True.")/length(churn$Churn.) ## Other method  
prop.churn\_or

## [1] 0.1449145

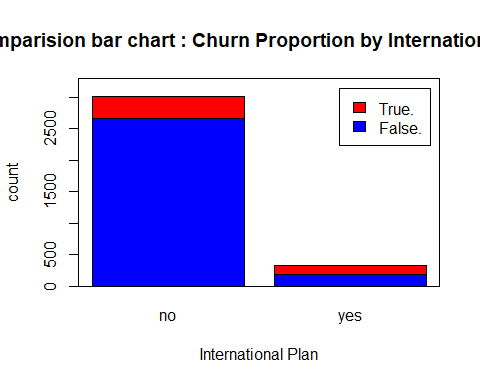
## creating bar chart  
barplot(sum.churn,ylim = c(0,3000),main = "Bar Graph of Chrners and Non Churners",col = "lightblue")  
box(which = "plot",lty = "solid",col="black")



## We will try to make a table for the counts of churn and the international plan  
  
count<-table(churn$Churn.,churn$Int.l.Plan,dnn = c("Churn","International Plans"))  
count

## International Plans  
## Churn no yes  
## False. 2664 186  
## True. 346 137

## Using the above data to make the overlaid bar graph.  
  
barplot(count,legend=rownames(count),col = c("blue","red"),ylim = c(0,3300),ylab = "count",xlab = "International Plan",main = "Comparision bar chart : Churn Proportion by International Plan")  
box(which = "plot",lty = "solid",col="black")



## If I use beside=TRUE then the bar chart will become clustered. otherwise it will be overlayed.

sumtable<-addmargins(count,FUN = sum)

## Margins computed over dimensions  
## in the following order:  
## 1: Churn  
## 2: International Plans

sumtable

## International Plans  
## Churn no yes sum  
## False. 2664 186 2850  
## True. 346 137 483  
## sum 3010 323 3333

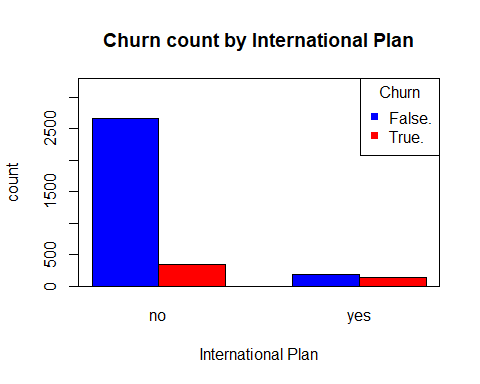
## creating a table of proportions over rows.  
row.margin<-round(prop.table(count,margin = 1),4)\*100 # Margin=1 means taking proportions in rows. whereas taking 2 means taking proportions in the column. 4 = number of digits.  
row.margin

## International Plans  
## Churn no yes  
## False. 93.47 6.53  
## True. 71.64 28.36

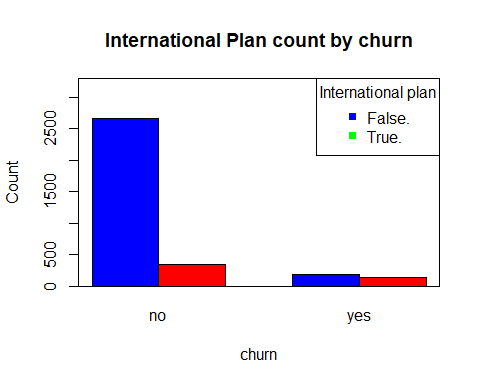
## Creating a table of proportions over column  
## we will just take margin equals 2.  
  
col.margin<-round(prop.table(count,margin = 2),4)\*100  
col.margin

## International Plans  
## Churn no yes  
## False. 88.50 57.59  
## True. 11.50 42.41

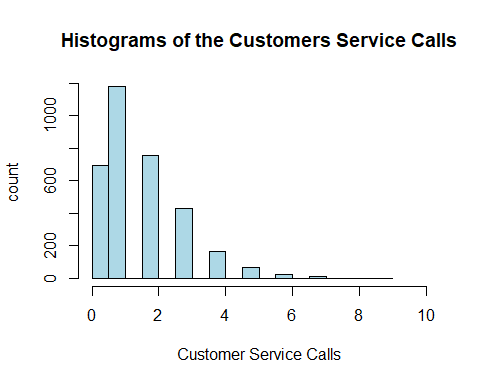
## Lets try a clustered bar chart of the international plan with churn as legend with different means.  
  
barplot(count,col = c("blue","red"),ylim = c(0,3300),ylab = "count",xlab = "International Plan",main = "Churn count by International Plan",beside = TRUE)  
legend("topright",c(rownames(count)),col = c("blue","red"),pch = 15,title = "Churn")  
box(which = "plot",lty = "solid",col="black")



## Lets try a clustered bar chart of the churn with international plan as legend with different means.  
  
barplot(count,col = c("blue","red"),ylim = c(0,3300),ylab = "Count",xlab = "churn",main = "International Plan count by churn",beside = TRUE)  
legend("topright",c(rownames(count)),col = c("blue","green"),pch = 15,title = "International plan")  
box(which = "plot",lty = "solid",col="black")

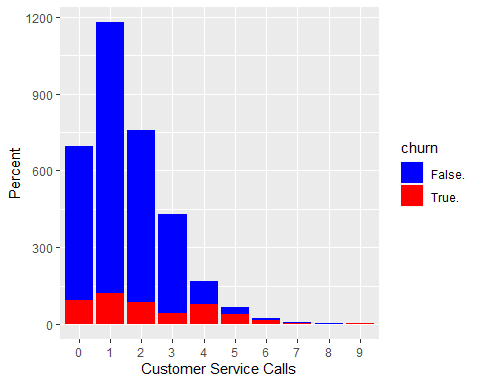


## Histogram of non overlayed customer service calls  
  
hist(churn$CustServ.Calls,xlim = c(0,10),col = "lightblue",ylab = "count",xlab = "Customer Service Calls",main = "Histograms of the Customers Service Calls")

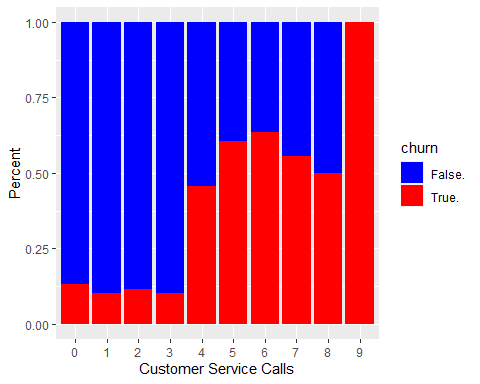


## Let's try the EDA using the package ggplot2  
library(ggplot2)

## Ovrlayed bar chart  
  
ggplot()+geom\_bar(data = churn,aes(x=factor(CustServ.Calls),fill=factor(Churn.)),position = "stack")+scale\_x\_discrete("Customer Service Calls")+scale\_y\_continuous("Percent")+guides(fill=guide\_legend(title = "churn"))+scale\_fill\_manual(values = c("blue","red"))



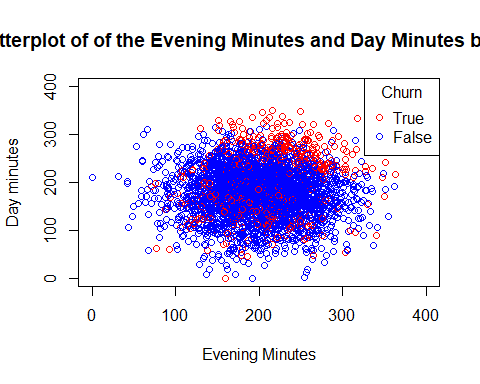
## Using position =fill instead of position="stack" will provide us a homogenous view of the same bar graph   
  
ggplot()+geom\_bar(data = churn,aes(x=factor(CustServ.Calls),fill=factor(Churn.)),position = "fill")+scale\_x\_discrete("Customer Service Calls")+scale\_y\_continuous("Percent")+guides(fill=guide\_legend(title = "churn"))+scale\_fill\_manual(values = c("blue","red"))



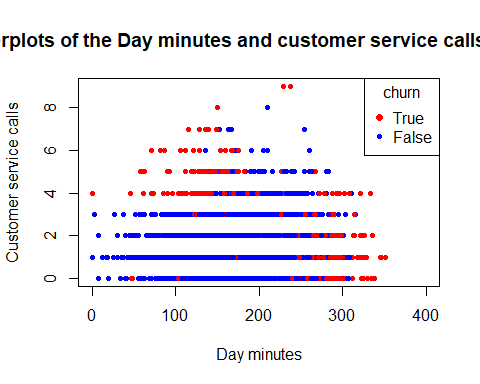
## Now we will use two sample t test  
## First we will partition the data.  
  
churn.false<-subset(churn,churn$Churn.=="False.")  
churn.true<-subset(churn,churn$Churn.=="True.")  
  
## Now we run the test  
  
t.test(churn.false$Intl.Calls,churn.true$Intl.Calls)

##   
## Welch Two Sample t-test  
##   
## data: churn.false$Intl.Calls and churn.true$Intl.Calls  
## 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 of x mean of y   
## 4.532982 4.163561

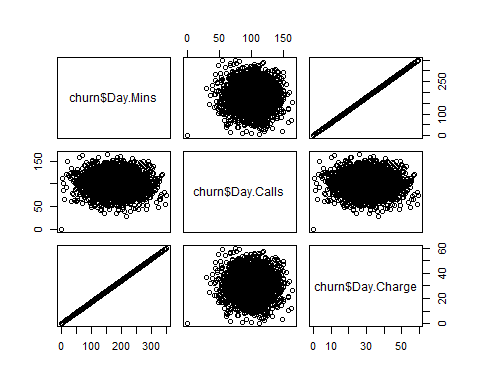
## Scatter plot of the Evening minutes and day minutes coloured by churn.  
plot(churn$Eve.Mins,churn$Day.Mins,xlim = c(0,400),ylim = c(0,400),xlab = "Evening Minutes",ylab = "Day minutes",main = "Scatterplot of of the Evening Minutes and Day Minutes by churn",col=ifelse(churn$Churn.=="True.","red","blue"))  
legend("topright",c("True","False"),col = c("red","blue"),pch = 1,title = "Churn")



## Scatter plot of day minutes and customer service calls, coloured by churn.  
  
plot(churn$Day.Mins,churn$CustServ.Calls,xlim = c(0,400),xlab = "Day minutes",ylab = "Customer service calls",main = "Scatterplots of the Day minutes and customer service calls by churn",col=ifelse(churn$Churn.=="True.","red","blue"),pch=ifelse(churn$Churn.=="True",16,20))  
legend("topright",c("True","False"),col = c("red","blue"),pch = c(16,20),title = "churn")



## Scatter plot matrix  
pairs(~churn$Day.Mins+churn$Day.Calls+churn$Day.Charge)



## From the last matrix plot we observed that there is a linear relationship between day minutes and day charge.  
## Let's run the regression analysis on these.  
  
fit<-lm(churn$Day.Charge~churn$Day.Mins)  
fit

##   
## Call:  
## lm(formula = churn$Day.Charge ~ churn$Day.Mins)  
##   
## Coefficients:  
## (Intercept) churn$Day.Mins   
## 0.0006134 0.1699996

summary(fit)

##   
## Call:  
## lm(formula = churn$Day.Charge ~ churn$Day.Mins)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0045935 -0.0025391 0.0004326 0.0024587 0.0045224   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.134e-04 1.711e-04 3.585e+00 0.000341 \*\*\*  
## churn$Day.Mins 1.700e-01 9.108e-07 1.866e+05 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.002864 on 3331 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 3.484e+10 on 1 and 3331 DF, p-value: < 2.2e-16

## Here we observe that as we increase day minutes by 1 unit, there will be increase of Rs. 0.1699996.

## Correlation values with p values  
  
days<-cbind(churn$Day.Mins,churn$Day.Calls,churn$Day.Charge)  
MinsCallsTest<-cor.test(churn$Day.Mins,churn$Day.Calls)  
MinsChageTest<-cor.test(churn$Day.Mins,churn$Day.Charge)  
CallsChargeTest<-cor.test(churn$Day.Calls,churn$Day.Charge)   
round(cor(days))

## [,1] [,2] [,3]  
## [1,] 1 0 1  
## [2,] 0 1 0  
## [3,] 1 0 1

MinsCallsTest$p.value

## [1] 0.6968515

MinsChageTest$p.value

## [1] 0

CallsChargeTest$p.value

## [1] 0.6967428

## Now we will apply correlation values and p values in matrix form.  
  
## We will collect variables of interest.  
  
corrdata<-cbind(churn$Account.Length,churn$VMail.Plan,churn$Day.Mins,churn$Day.Calls,churn$CustServ.Calls)  
round(cor(corrdata))

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 0 0 0 0  
## [2,] 0 1 0 0 0  
## [3,] 0 0 1 0 0  
## [4,] 0 0 0 1 0  
## [5,] 0 0 0 0 1

## After creating variables of interest we will declare the matrix.  
  
corrpvalues<-matrix(rep(0,25),ncol = 5) ## Created a zero matrix.  
corrpvalues

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0 0 0 0 0  
## [2,] 0 0 0 0 0  
## [3,] 0 0 0 0 0  
## [4,] 0 0 0 0 0  
## [5,] 0 0 0 0 0

## Now we fill the matrix with the correlations  
  
for (i in 1:4) {  
 for (j in (i+1):5) {  
 corrpvalues[i,j]<-corrpvalues[j,i]<-round(cor.test(corrdata[,i],corrdata[,j])$p.value,3)  
 }  
}  
corrpvalues

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.000 0.866 0.720 0.026 0.827  
## [2,] 0.866 0.000 0.923 0.522 0.304  
## [3,] 0.720 0.923 0.000 0.697 0.439  
## [4,] 0.026 0.522 0.697 0.000 0.274  
## [5,] 0.827 0.304 0.439 0.274 0.000