Churn data modeling: Exploring categorical and numerical variables in churn data(contd..)

24/07/2021

Lecture 4(Practical)

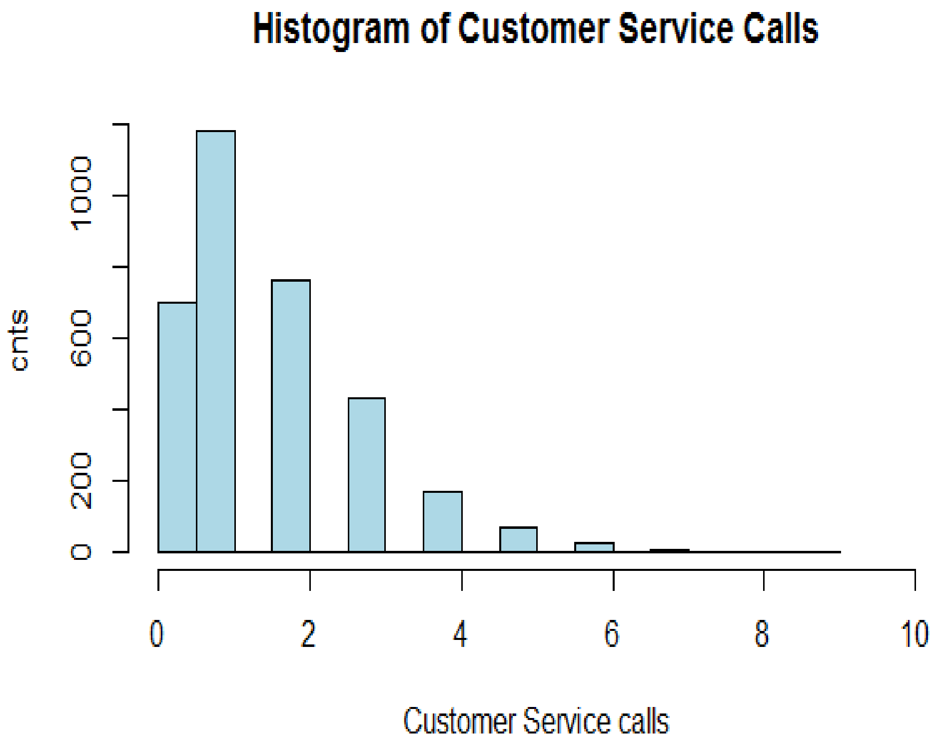
Exploring Numerical variables
Histogram of non-overlayed Customer Service Calls

- hist(churn$CustServ.Calls, xlim=c(0,10),col="lightblue", ylab="cnts", xlab="Customer Service calls",

  main= "Histogram of Customer Service Calls")

Histogram of CSC with no overlay

The distribution is right skewed with a mode of 1 call.



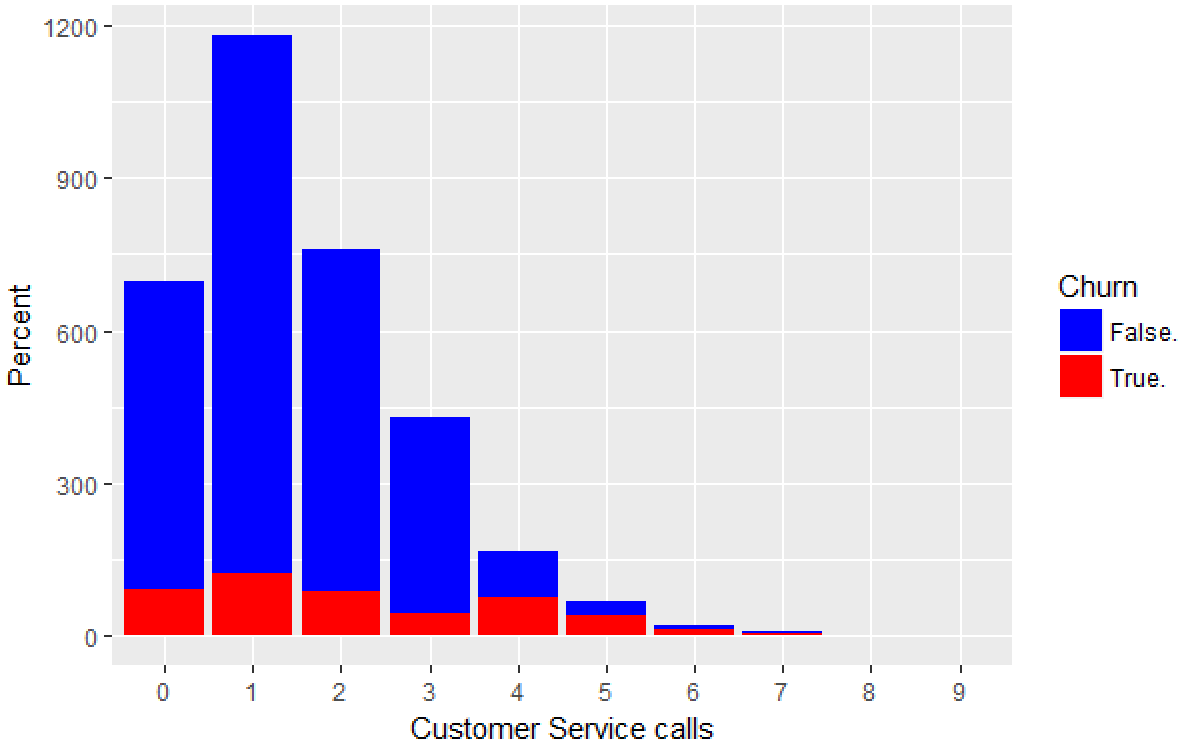
Overlayed Barcharts

- Library(ggplot2)
- ggplot()+geom_bar(data=churn, aes(x=factor(churn$CustServ.Calls),fill=factor(churn$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"))

Is churn by CSC?

Churn % may be higher for higher numbers of CSC.



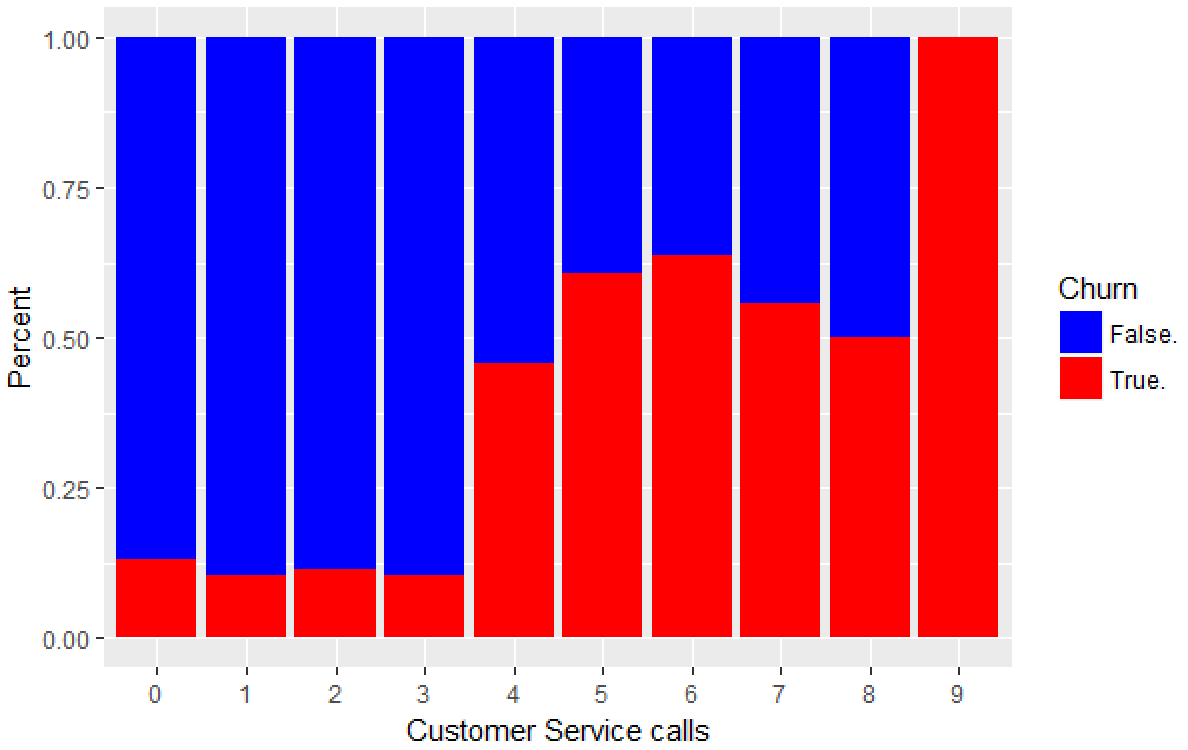
Normalized Histogram

- 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("blue","red"))

- Normalized histogram are useful for teasing out the relationship between a numerical predictor and the target.

Proportion of Churners versus Non churners



Binning based on predictive value

- Consider the variable Day Minutes
- range(churn$Day.Mins)
- #will get the range of values of the variable Day Minutes
- breaks <- c(0,50,100,150,200,250,300,350,400)
- # specify interval/bin labels
- tags <- c("[0-50)","[50-100)", "[100-150)", "[150-200)", "[200-250)", "[250-300)","[300-350)", "[350-400)")
- # bucketing values into bins
- group_tags <- cut(churn$Day.Mins,

  breaks=breaks,

  include.lowest=TRUE,

  right=FALSE,

  labels=tags)
- # inspect bins
- summary(group_tags)

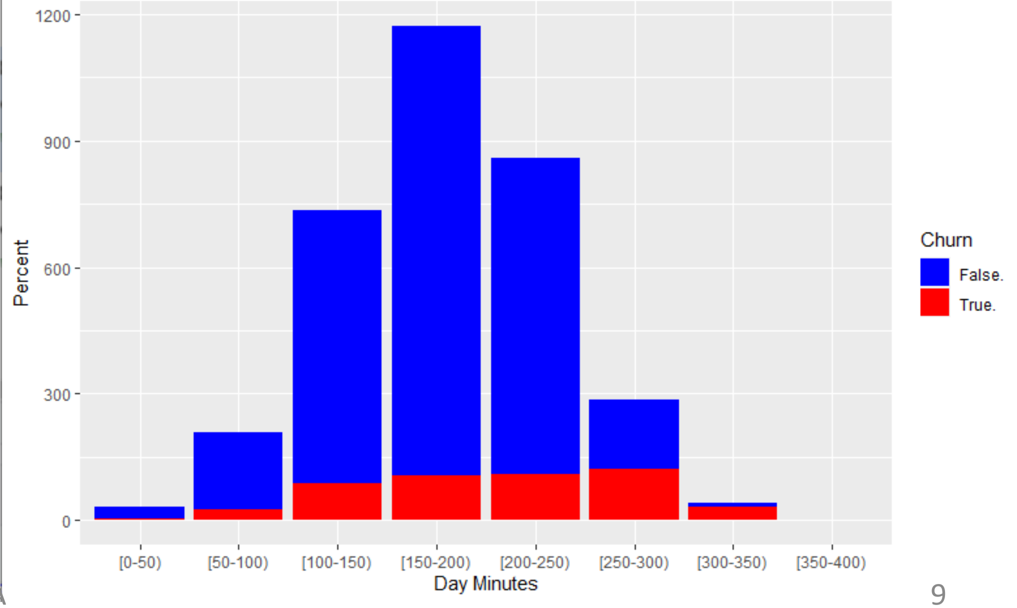
```
> summary(group_tags)
   [0-50)  [50-100) [100-150) [150-200) [200-250) [250-300) [300-350)
       30       208       735      1173       859       285        42
[350-400)
        1
```

Non-standardized histogram of Day Minutes

- dayminutes_groups <- factor(group_tags, ordered = TRUE)

- ggplot()+geom_bar(data=churn,aes(x=factor(group_tags),fill=factor(churn$Churn)),position= "stack")+ scale_x_discrete("Day Minutes")+scale_y_continuous("Percent")+ guides(fill=guide_legend(title ="Churn"))+scale_fill_manual (values=c("blue","red"))

- High day users tend to churn at a higher rate.



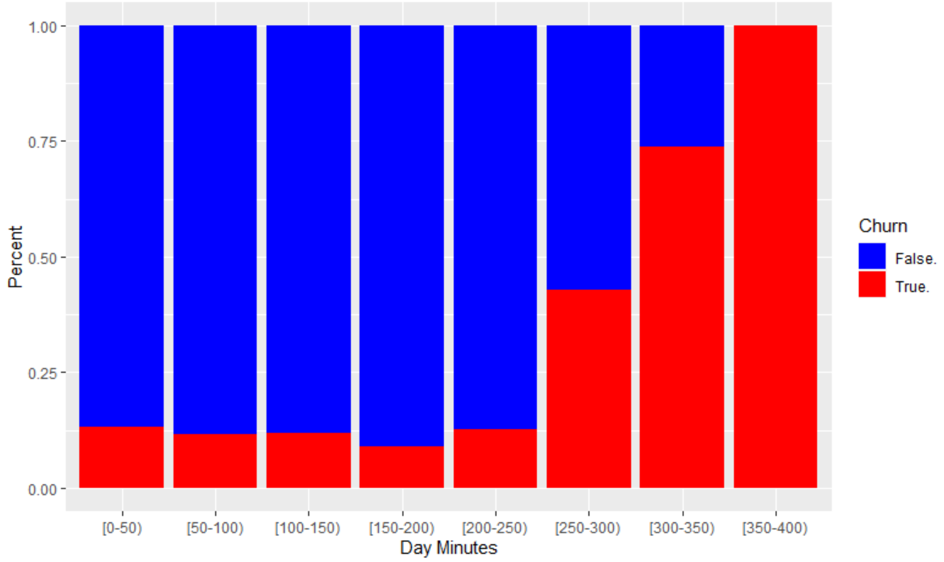
Standardized histogram of Day Minutes

- ggplot()+geom_bar(data=churn, aes(x=factor(group_tags),fill=factor(churn$Churn)),position= "fill")+

  scale_x_discrete("Day Minutes")+scale_y_continuous("Percent")+

  guides(fill=guide_legend(title ="Churn"))+scale_fill_manual (values=c("blue","red"))

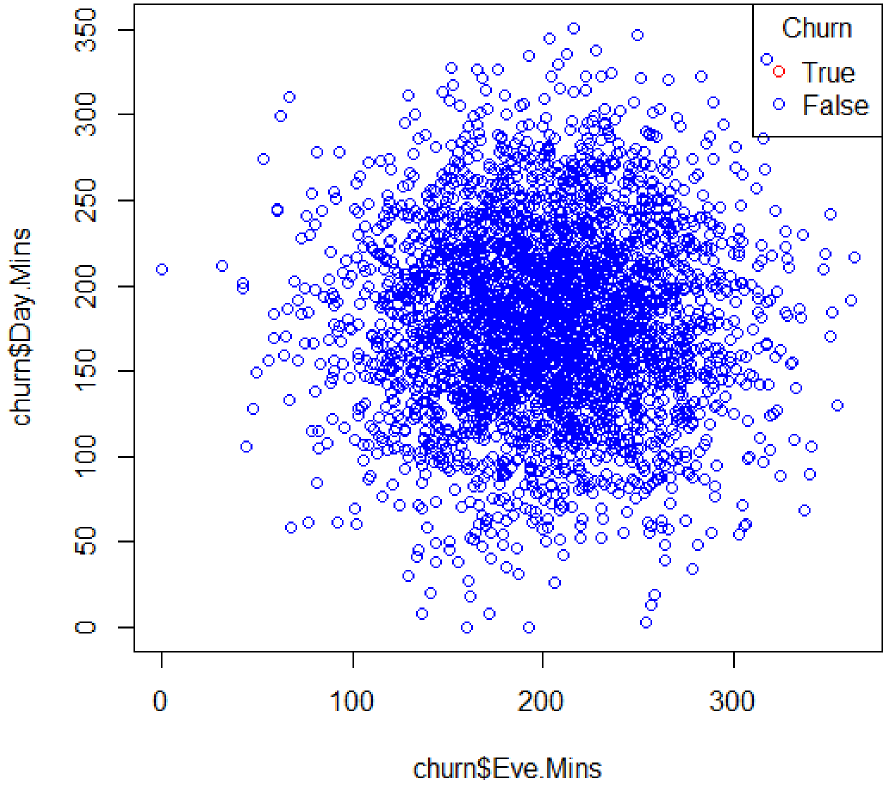
- The company should investigate why heavy day users are tempted to leave. As the number of day minutes passes 200, the company should consider special incentives.
- The model will include day minutes as a predictor of churn.



Exploring Multivariate Relationships

- Possible multivariate associations of numeric variables with churn using scatter plot
- Scatter plot of Evening Minutes and Day Minutes colored by Churn.
- plot(churn$Eve.Mins,churn$Day.Mins, col=ifelse(churn$Churn=="True","red","blue"))
- legend("topright",c("True","False"),col=c("red","blue"),pch = 1, title= "Churn")

Scatter plot of Evening Minutes and Day Minutes colored by Churn

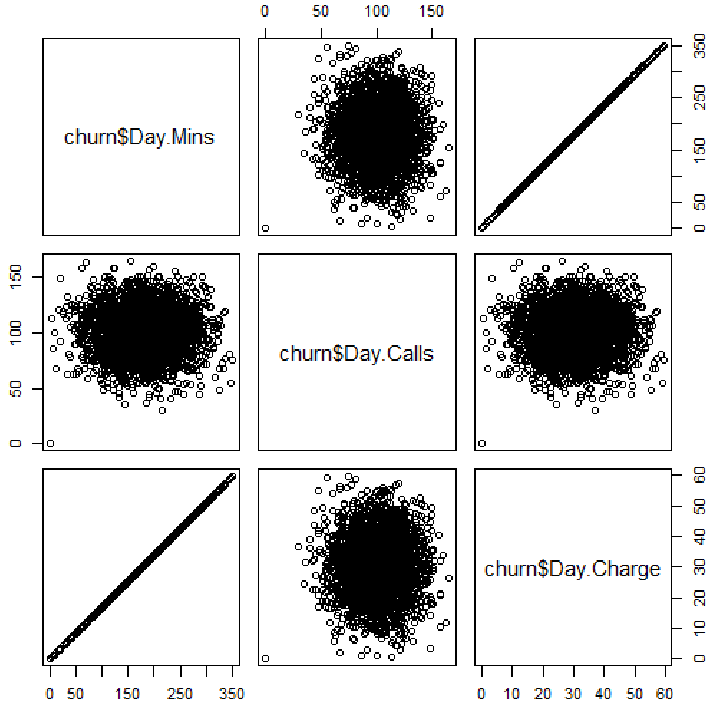


Strategy for handling correlated predictor variables

- Two variables X and Y are linearly correlated if an increase in X is associated with either an increase in Y or a decrease in Y.
- The correlation coefficient *r* quantifies the strength and direction of the linear relationship between X and Y.
- Identify any variables that are *perfectly correlated* (i.e., r=1 or r=-1). Do not retain both variables in the model, but rather omit one.
- Identify groups of variables that are correlated with each other. Then later during modeling phase, apply dimension reduction methods such as PCA to these variables.

Scatter Plot Matrix:
Using EDA to investigate correlated predictor variables

- pairs(~churn$Day.Mins+churn$Day.Calls+churn$Day.Charge)



Correlation values with p values

- days<-cbind(churn$Day.Mins,churn$Day.Calls,churn$Day.Charge)
- MinsCallsTest<-cor.test(churn$Day.Mins,churn$Day.Calls)
- MinsChargeTest<-cor.test(churn$Day.Mins,churn$Day.Charge)
- CallsChargeTest<-cor.test(churn$Day.Calls,churn$Day.Charge)
- round(cor(days),4)

```
       [,1]   [,2]   [,3]
[1,] 1.0000 0.0068 1.0000
[2,] 0.0068 1.0000 0.0068
[3,] 1.0000 0.0068 1.0000
```

- MinsCallsTest$p.value

  0.6968515 (No correlation between Day.Mins and Day.Calls)
- MinsChargeTest$p.value

  0 (Correlation between Day.Mins and Day.Charge)
- CallsChargeTest$p.value

  0.6967428 (No correlation between Day.Calls and Day.Charge)