Group 4 R Script Submission for Group Assignment

Steven Spence, Marianne Marjanovic, Stanley Omeike, and Manar Qutub

11/20/2018

# Load the orginial data set for analysis into the document  
  
Churn.Training.Data.Original <- read.csv("Churn Training Data Original.csv")

The following R codes will deal with data exploration and preparation.

# Review the data types for all of the variables  
  
str(Churn.Training.Data.Original)

## 'data.frame': 3333 obs. of 20 variables:  
## $ state : Factor w/ 51 levels "AK","AL","AR",..: 34 12 8 12 36 25 28 39 13 16 ...  
## $ account\_length : int 125 108 82 NA 83 89 135 28 86 65 ...  
## $ area\_code : Factor w/ 3 levels "area\_code\_408",..: 3 2 2 1 2 2 2 2 1 2 ...  
## $ international\_plan : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ voice\_mail\_plan : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...  
## $ number\_vmail\_messages : int 0 0 0 30 0 0 0 0 0 0 ...  
## $ total\_day\_minutes : num 2013 292 300 110 337 ...  
## $ total\_day\_calls : int 99 99 109 71 120 81 81 87 115 137 ...  
## $ total\_day\_charge : num 28.7 49.6 51 18.8 57.4 ...  
## $ total\_eve\_minutes : num 1108 221 181 182 227 ...  
## $ total\_eve\_calls : int 107 93 100 108 116 74 114 92 112 83 ...  
## $ total\_eve\_charge : num 14.9 18.8 15.4 15.5 19.3 ...  
## $ total\_night\_minutes : num 243 229 270 184 154 ...  
## $ total\_night\_calls : int 92 110 73 88 114 120 82 112 95 111 ...  
## $ total\_night\_charge : num 10.95 10.31 12.15 8.27 6.93 ...  
## $ total\_intl\_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...  
## $ total\_intl\_calls : int 7 9 4 8 7 4 6 3 7 6 ...  
## $ total\_intl\_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...  
## $ number\_customer\_service\_calls: int 0 2 0 2 0 1 1 3 2 4 ...  
## $ churn : Factor w/ 2 levels "no","yes": 1 2 2 1 2 1 1 1 1 2 ...

From this review, we can see that the “churn” variable is coded as a factor with “yes” being the second value.

# Review the summary of the data set  
  
summary(Churn.Training.Data.Original)

## state account\_length area\_code international\_plan  
## WV : 106 Min. :-209.00 area\_code\_408: 838 no :3010   
## MN : 84 1st Qu.: 72.00 area\_code\_415:1655 yes: 323   
## NY : 83 Median : 100.00 area\_code\_510: 840   
## AL : 80 Mean : 97.32   
## OH : 78 3rd Qu.: 127.00   
## OR : 78 Max. : 243.00   
## (Other):2824 NA's :501   
## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## no :2411 Min. :-10.000 Min. : 0.0 Min. : 0.0   
## yes: 922 1st Qu.: 0.000 1st Qu.: 149.3 1st Qu.: 87.0   
## Median : 0.000 Median : 190.5 Median :101.0   
## Mean : 7.333 Mean : 418.9 Mean :100.3   
## 3rd Qu.: 16.000 3rd Qu.: 237.8 3rd Qu.:114.0   
## Max. : 51.000 Max. :2185.1 Max. :165.0   
## NA's :200 NA's :200 NA's :200   
## total\_day\_charge total\_eve\_minutes total\_eve\_calls total\_eve\_charge  
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.00   
## 1st Qu.:24.45 1st Qu.: 170.5 1st Qu.: 87.0 1st Qu.:14.14   
## Median :30.65 Median : 209.9 Median :100.0 Median :17.09   
## Mean :30.63 Mean : 324.3 Mean :100.1 Mean :17.08   
## 3rd Qu.:36.84 3rd Qu.: 257.6 3rd Qu.:114.0 3rd Qu.:20.00   
## Max. :59.64 Max. :1244.2 Max. :170.0 Max. :30.91   
## NA's :200 NA's :301 NA's :200 NA's :200   
## total\_night\_minutes total\_night\_calls total\_night\_charge  
## Min. : 23.2 Min. : 33.0 Min. : 1.040   
## 1st Qu.:167.3 1st Qu.: 87.0 1st Qu.: 7.530   
## Median :201.4 Median :100.0 Median : 9.060   
## Mean :201.2 Mean :100.1 Mean : 9.054   
## 3rd Qu.:235.3 3rd Qu.:113.0 3rd Qu.:10.590   
## Max. :395.0 Max. :175.0 Max. :17.770   
## NA's :200 NA's :200   
## total\_intl\_minutes total\_intl\_calls total\_intl\_charge  
## Min. : 0.00 Min. : 0.00 Min. :0.000   
## 1st Qu.: 8.50 1st Qu.: 3.00 1st Qu.:2.300   
## Median :10.30 Median : 4.00 Median :2.780   
## Mean :10.23 Mean : 4.47 Mean :2.762   
## 3rd Qu.:12.10 3rd Qu.: 6.00 3rd Qu.:3.270   
## Max. :20.00 Max. :20.00 Max. :5.400   
## NA's :200 NA's :301 NA's :200   
## number\_customer\_service\_calls churn   
## Min. :0.000 no :2850   
## 1st Qu.:1.000 yes: 483   
## Median :1.000   
## Mean :1.561   
## 3rd Qu.:2.000   
## Max. :9.000   
## NA's :200

From the summary above, we can see there may be some noise/incorrect values within the dataset:

account\_length: contains negative values and missing values

number\_vmail\_messages: contains negative values and missing values

total\_day\_minutes: missing values

total\_day\_calls: missing values

total\_day\_charge: missing values

total\_eve\_minutes: missing values

total\_eve\_calls: missing values

total\_eve\_charge: missing values

total\_night\_minutes: missing values

total\_night\_charge: missing values

total\_intl\_minutes: missing values

total\_intl\_calls: missing values

total\_intl\_charge: missing values

total\_customer\_service\_calls: missing values

# Return the percentage for each row that is missing data values  
  
rowMeans(is.na(Churn.Training.Data.Original))

# Return percentage of each dimension that is missing values  
  
colMeans(is.na(Churn.Training.Data.Original))

## state account\_length   
## 0.00000000 0.15031503   
## area\_code international\_plan   
## 0.00000000 0.00000000   
## voice\_mail\_plan number\_vmail\_messages   
## 0.00000000 0.06000600   
## total\_day\_minutes total\_day\_calls   
## 0.06000600 0.06000600   
## total\_day\_charge total\_eve\_minutes   
## 0.06000600 0.09030903   
## total\_eve\_calls total\_eve\_charge   
## 0.06000600 0.06000600   
## total\_night\_minutes total\_night\_calls   
## 0.06000600 0.00000000   
## total\_night\_charge total\_intl\_minutes   
## 0.06000600 0.06000600   
## total\_intl\_calls total\_intl\_charge   
## 0.09030903 0.06000600   
## number\_customer\_service\_calls churn   
## 0.06000600 0.00000000

The following code below will help with visualizing the missing values throughout the data set.

# Loading the MICE package  
  
library(mice)

## Warning: package 'mice' was built under R version 3.4.4

## Loading required package: lattice

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(VIM)

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

## Warning: package 'data.table' was built under R version 3.4.4

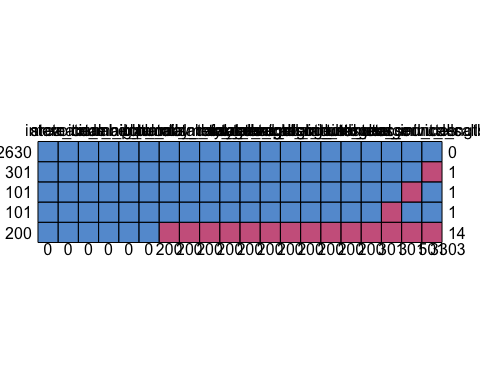
## VIM is ready to use.   
## Since version 4.0.0 the GUI is in its own package VIMGUI.  
##   
## Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##   
## Attaching package: 'VIM'

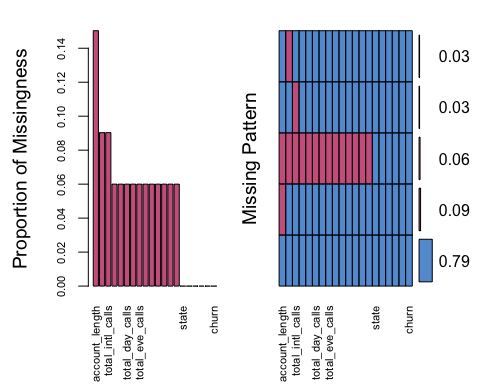
## The following object is masked from 'package:datasets':  
##   
## sleep

library(lattice)  
  
# Used to understand the missing value pattern  
  
md.pattern(Churn.Training.Data.Original)



## state area\_code international\_plan voice\_mail\_plan total\_night\_calls  
## 2630 1 1 1 1 1  
## 301 1 1 1 1 1  
## 101 1 1 1 1 1  
## 101 1 1 1 1 1  
## 200 1 1 1 1 1  
## 0 0 0 0 0  
## churn number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## 2630 1 1 1 1  
## 301 1 1 1 1  
## 101 1 1 1 1  
## 101 1 1 1 1  
## 200 1 0 0 0  
## 0 200 200 200  
## total\_day\_charge total\_eve\_calls total\_eve\_charge total\_night\_minutes  
## 2630 1 1 1 1  
## 301 1 1 1 1  
## 101 1 1 1 1  
## 101 1 1 1 1  
## 200 0 0 0 0  
## 200 200 200 200  
## total\_night\_charge total\_intl\_minutes total\_intl\_charge  
## 2630 1 1 1  
## 301 1 1 1  
## 101 1 1 1  
## 101 1 1 1  
## 200 0 0 0  
## 200 200 200  
## number\_customer\_service\_calls total\_eve\_minutes total\_intl\_calls  
## 2630 1 1 1  
## 301 1 1 1  
## 101 1 1 0  
## 101 1 0 1  
## 200 0 0 0  
## 200 301 301  
## account\_length   
## 2630 1 0  
## 301 0 1  
## 101 1 1  
## 101 1 1  
## 200 0 14  
## 501 3303

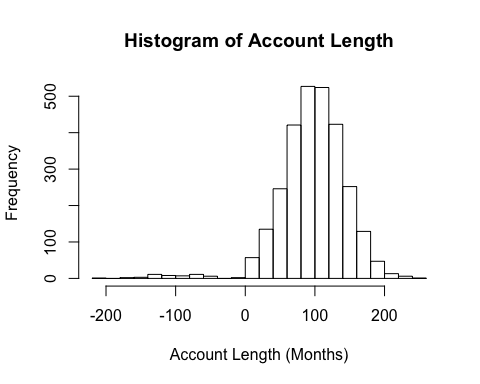
# Plot the missing values  
  
aggr(Churn.Training.Data.Original, col = mdc(1:2), numbers = TRUE, sortVars = TRUE, labels = names(Churn.Training.Data.Original), cex.axis = .7, gap = 3, ylab = c("Proportion of Missingness", "Missing Pattern"))



##   
## Variables sorted by number of missings:   
## Variable Count  
## account\_length 0.15031503  
## total\_eve\_minutes 0.09030903  
## total\_intl\_calls 0.09030903  
## number\_vmail\_messages 0.06000600  
## total\_day\_minutes 0.06000600  
## total\_day\_calls 0.06000600  
## total\_day\_charge 0.06000600  
## total\_eve\_calls 0.06000600  
## total\_eve\_charge 0.06000600  
## total\_night\_minutes 0.06000600  
## total\_night\_charge 0.06000600  
## total\_intl\_minutes 0.06000600  
## total\_intl\_charge 0.06000600  
## number\_customer\_service\_calls 0.06000600  
## state 0.00000000  
## area\_code 0.00000000  
## international\_plan 0.00000000  
## voice\_mail\_plan 0.00000000  
## total\_night\_calls 0.00000000  
## churn 0.00000000

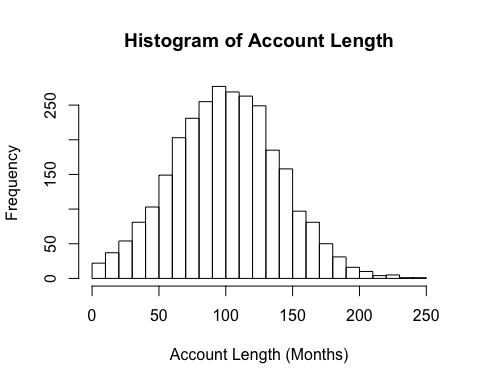
The histograms below will show the values of the data set oringinally entered into the file. This will allow us to explore the data and see where outliers and missing values may be present.

# Create a histogram of the "account\_length" variable  
  
hist(Churn.Training.Data.Original$account\_length, breaks = 25, main = "Histogram of Account Length", xlab = "Account Length (Months)", ylab = "Frequency")



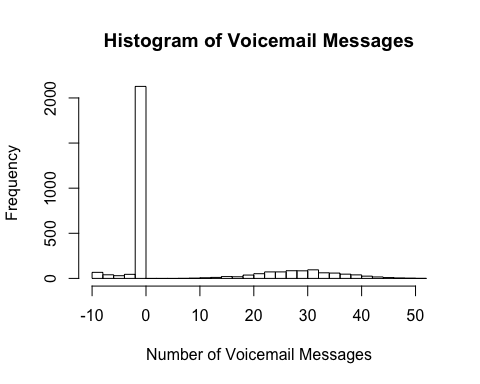
From this chart, as well as the previous summary, we can see that there are negative values that do not make sense in the context of the business problem. We will begin our analysis by assuming these were incorrectly added as negative values, when they should have been positive values.

# Convert the negative values to postive values for account length  
  
Churn.Training.Data.Original$account\_length <- abs(Churn.Training.Data.Original$account\_length)  
  
# Plot of the treated values for account length  
  
hist(Churn.Training.Data.Original$account\_length, breaks = 25, main = "Histogram of Account Length", xlab = "Account Length (Months)", ylab = "Frequency")



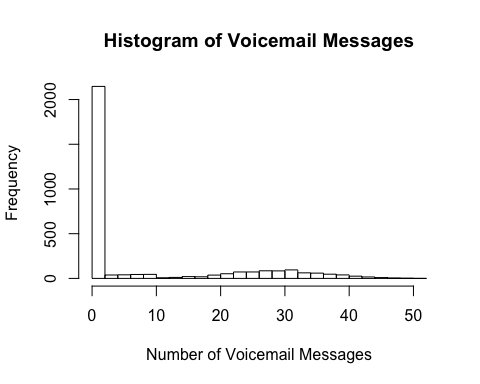
We can see that the data appears to be normally distributed after treating the negative values.

# Create a histogram of the "number\_vmail\_messages" variable  
  
hist(Churn.Training.Data.Original$number\_vmail\_messages, breaks = 25, main = "Histogram of Voicemail Messages", xlab = "Number of Voicemail Messages", ylab = "Frequency")



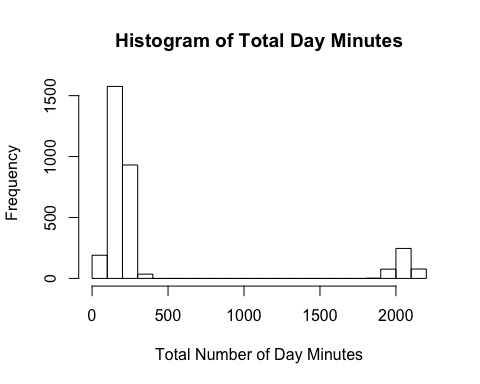
From this chart, as well as the previous summary, we can see that there are negative values that do not make sense in the context of the business problem. We will begin our analysis by assuming these were incorrectly added as negative values, when they should have been positive values.

# Convert the negative values to postive values for number of voicemail messages  
  
Churn.Training.Data.Original$number\_vmail\_messages <- abs(Churn.Training.Data.Original$number\_vmail\_messages)  
  
# Plot of the treated values for number of voicemail messages  
  
hist(Churn.Training.Data.Original$number\_vmail\_messages, breaks = 25, main = "Histogram of Voicemail Messages", xlab = "Number of Voicemail Messages", ylab = "Frequency")



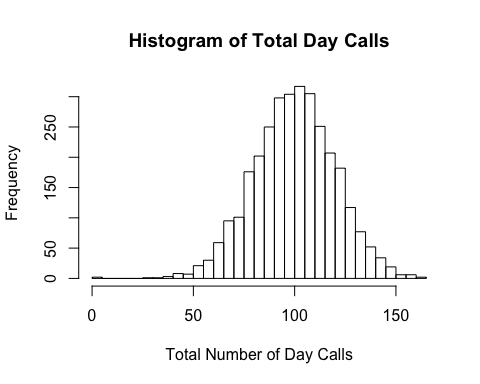
The data is still significantly skewed after treating the negative values. Additional transformation of log transforming should be used, if possible, to treat the skewness of the data.

# Create a histogram of the "total\_day\_minutes" variable  
  
hist(Churn.Training.Data.Original$total\_day\_minutes, breaks = 25, main = "Histogram of Total Day Minutes", xlab = "Total Number of Day Minutes", ylab = "Frequency")



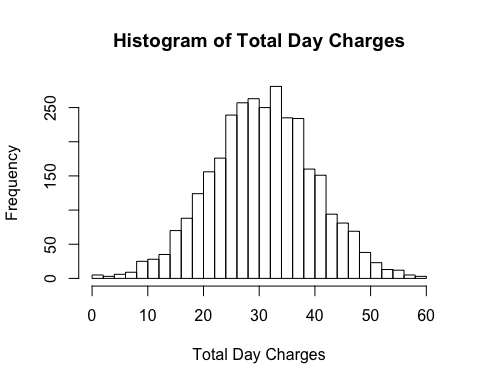
From the chart above we can see that there is a bimodal relationship present within the data set in this variable. We will attempt to treat these values by making discrete variables to describe “low” and “high” frequency users.

# Create a histogram of the "total\_day\_calls" variable  
  
hist(Churn.Training.Data.Original$total\_day\_calls, breaks = 25, main = "Histogram of Total Day Calls", xlab = "Total Number of Day Calls", ylab = "Frequency")

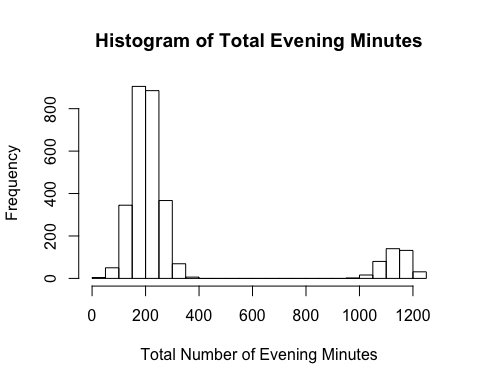


We can see a fairly normal distribution; however, there looks to be a few outliers at the value of 0.

# Create a histogram of the "total\_day\_charge" variable  
  
hist(Churn.Training.Data.Original$total\_day\_charge, breaks = 25, main = "Histogram of Total Day Charges", xlab = "Total Day Charges", ylab = "Frequency")

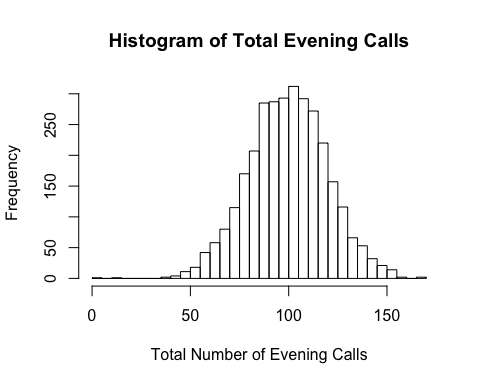


# Create a histogram of the "total\_eve\_minutes" variable  
  
hist(Churn.Training.Data.Original$total\_eve\_minutes, breaks = 25, main = "Histogram of Total Evening Minutes", xlab = "Total Number of Evening Minutes", ylab = "Frequency")



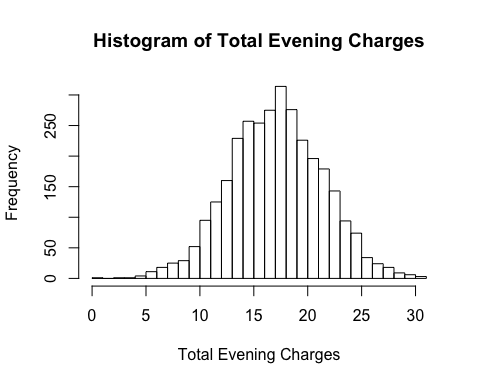
From the chart above we can see that there is a bimodal relationship present within the data set in this variable. We will attempt to treat these values by making discrete variables to describe “low” and “high” frequency users.

# Create a histogram of the "total\_eve\_calls" variable  
  
hist(Churn.Training.Data.Original$total\_eve\_calls, breaks = 25, main = "Histogram of Total Evening Calls", xlab = "Total Number of Evening Calls", ylab = "Frequency")



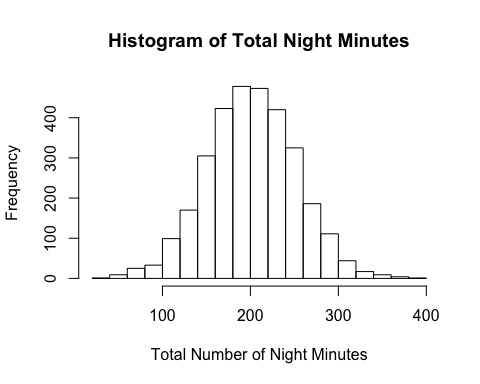
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_eve\_charge" variable  
  
hist(Churn.Training.Data.Original$total\_eve\_charge, breaks = 25, main = "Histogram of Total Evening Charges", xlab = "Total Evening Charges", ylab = "Frequency")



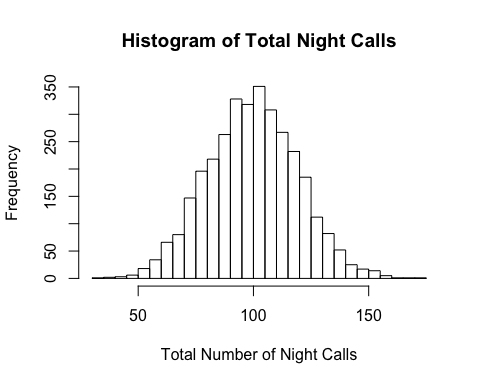
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_night\_minutes" variable  
  
hist(Churn.Training.Data.Original$total\_night\_minutes, breaks = 25, main = "Histogram of Total Night Minutes", xlab = "Total Number of Night Minutes", ylab = "Frequency")



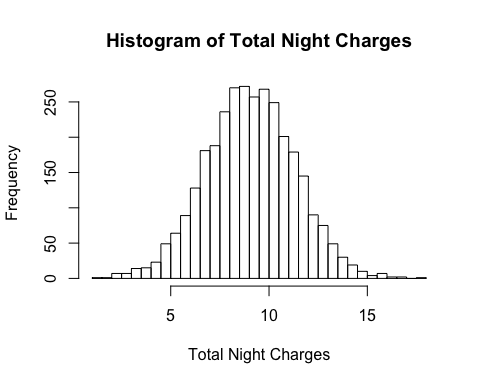
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_night\_calls" variable  
  
hist(Churn.Training.Data.Original$total\_night\_calls, breaks = 25, main = "Histogram of Total Night Calls", xlab = "Total Number of Night Calls", ylab = "Frequency")



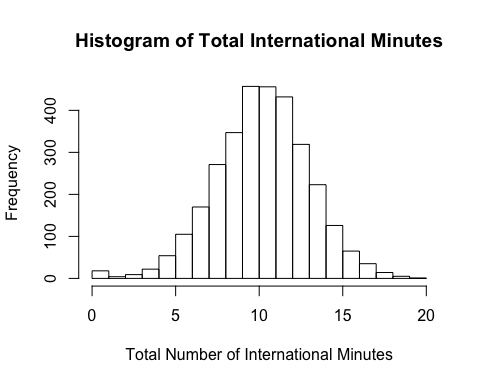
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_night\_charge" variable  
  
hist(Churn.Training.Data.Original$total\_night\_charge, breaks = 25, main = "Histogram of Total Night Charges", xlab = "Total Night Charges", ylab = "Frequency")



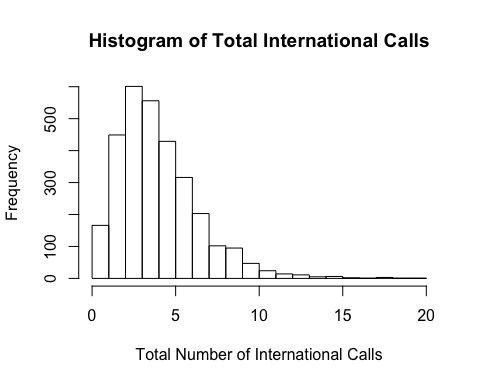
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_intl\_minutes" variable  
  
hist(Churn.Training.Data.Original$total\_intl\_minutes, breaks = 25, main = "Histogram of Total International Minutes", xlab = "Total Number of International Minutes", ylab = "Frequency")



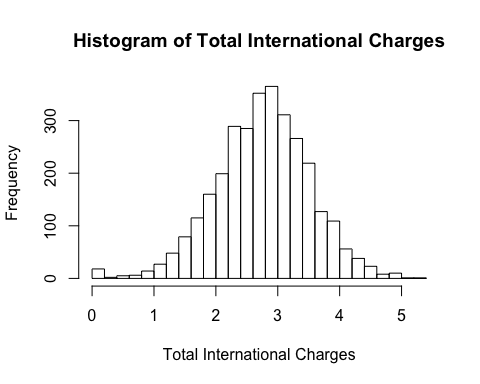
This variable appears to be normally distributed and little to no outliers present.

# Create a histogram of the "total\_intl\_calls" variable  
  
hist(Churn.Training.Data.Original$total\_intl\_calls, breaks = 25, main = "Histogram of Total International Calls", xlab = "Total Number of International Calls", ylab = "Frequency")



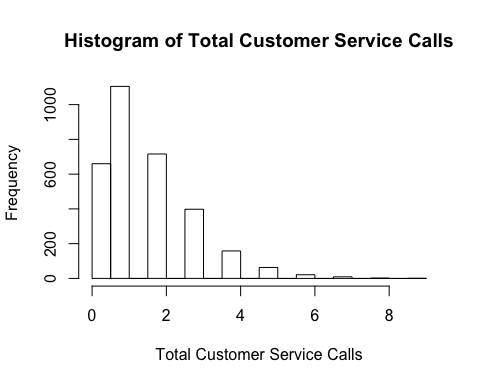
This data set appears to be slightly skewed and may need to be treated to account for the skewness.

# Create a histogram of the "total\_intl\_charge" variable  
  
hist(Churn.Training.Data.Original$total\_intl\_charge, breaks = 25, main = "Histogram of Total International Charges", xlab = "Total International Charges", ylab = "Frequency")



This variable appears to be normally distributed and little to no outliers present. Slight peak of values at the 0 value.

# Create a histogram of the "number\_customer\_service\_calls" variable  
  
hist(Churn.Training.Data.Original$number\_customer\_service\_calls, breaks = 25, main = "Histogram of Total Customer Service Calls", xlab = "Total Customer Service Calls", ylab = "Frequency")



This variable appears to be slightly skewed and may need to be treated prior to creating the model.

Code below is to impute the missing values throughout the data set:

The “MICE” library was used to treat the missing values throughout the data set. The code for this algorithm is presented below.

# Calling the mice library to impute the missing values throughout the data set.  
  
library(mice)  
  
Churn.Training.Set.Imputes <- mice(Churn.Training.Data.Original, m=5, maxit = 40)

##   
##

Imputed.Churn.Training.Set <- complete(Churn.Training.Set.Imputes, 5)

# Return percentage of each dimension that is missing values  
  
colMeans(is.na(Imputed.Churn.Training.Set))

Additional Columns / Variables Calculated to Add into Model

# Adds a new column that is the sum of all the total calls taken by a customer  
  
Imputed.Churn.Training.Set$total\_number\_calls <- rowSums(Imputed.Churn.Training.Set[,c("total\_day\_calls","total\_eve\_calls","total\_night\_calls","total\_intl\_calls")])

# Adds a new column that is the sum of all the minutes taken by a customer  
  
Imputed.Churn.Training.Set$total\_number\_minutes <- rowSums(Imputed.Churn.Training.Set[,c("total\_day\_minutes","total\_eve\_minutes","total\_night\_minutes","total\_intl\_minutes")])

# Adds a new column that is the sum of all the charges by a customer  
  
Imputed.Churn.Training.Set$total\_number\_charges <- rowSums(Imputed.Churn.Training.Set[,c("total\_day\_charge","total\_eve\_charge","total\_night\_charge","total\_intl\_charge")])

# Adds a new column that is the average cost per call by customer  
  
Imputed.Churn.Training.Set$average\_call\_cost <- (Imputed.Churn.Training.Set$total\_number\_charges/Imputed.Churn.Training.Set$total\_number\_calls)

# Adds a new column that is the average cost per minute by customer  
  
Imputed.Churn.Training.Set$average\_minute\_cost <- (Imputed.Churn.Training.Set$total\_number\_charges / Imputed.Churn.Training.Set$total\_number\_minutes)

# Adds a new column that is the average number of calls the customer completed a day  
  
Imputed.Churn.Training.Set$average\_calls\_per\_day <- (Imputed.Churn.Training.Set$total\_number\_calls/Imputed.Churn.Training.Set$account\_length)

# Adds a new column that is the average number of customer service calls per day  
  
Imputed.Churn.Training.Set$average\_service\_calls\_per\_day <- (Imputed.Churn.Training.Set$number\_customer\_service\_calls/Imputed.Churn.Training.Set$account\_length)

# Adds a new column that is the average charge of customer per day  
  
Imputed.Churn.Training.Set$average\_cost\_per\_day <- (Imputed.Churn.Training.Set$total\_number\_charges/Imputed.Churn.Training.Set$account\_length)

# Adds a new column that is the percentage of calls performed during the day  
  
Imputed.Churn.Training.Set$percent\_day\_calls <- (Imputed.Churn.Training.Set$total\_day\_calls/Imputed.Churn.Training.Set$total\_number\_calls)

# Adds a new column that is the percentage of calls performed during the evening  
  
Imputed.Churn.Training.Set$percent\_eve\_calls <- (Imputed.Churn.Training.Set$total\_eve\_calls/Imputed.Churn.Training.Set$total\_number\_calls)

# Adds a new column that is the percentage of calls performed during the night  
  
Imputed.Churn.Training.Set$percent\_night\_calls <- (Imputed.Churn.Training.Set$total\_night\_calls/Imputed.Churn.Training.Set$total\_number\_calls)

# Adds a new column that is the percentage of calls performed international  
  
Imputed.Churn.Training.Set$percent\_intl\_calls <- (Imputed.Churn.Training.Set$total\_intl\_calls/Imputed.Churn.Training.Set$total\_number\_calls)

# Adds a new column that is the percentage of minutes performed during the day  
  
Imputed.Churn.Training.Set$percent\_day\_minutes <- (Imputed.Churn.Training.Set$total\_day\_minutes/Imputed.Churn.Training.Set$total\_number\_minutes)

# Adds a new column that is the percentage of minutes performed during the evening  
  
Imputed.Churn.Training.Set$percent\_eve\_minutes <- (Imputed.Churn.Training.Set$total\_eve\_minutes/Imputed.Churn.Training.Set$total\_number\_minutes)

# Adds a new column that is the percentage of minutes performed during the night  
  
Imputed.Churn.Training.Set$percent\_night\_minutes <- (Imputed.Churn.Training.Set$total\_night\_minutes/Imputed.Churn.Training.Set$total\_number\_minutes)

# Adds a new column that is the percentage of minutes internationally  
  
Imputed.Churn.Training.Set$percent\_intl\_minutes <- (Imputed.Churn.Training.Set$total\_intl\_minutes/Imputed.Churn.Training.Set$total\_number\_minutes)

# Adds a new column that is the percentage of charges during the day  
  
Imputed.Churn.Training.Set$percent\_day\_charge <- (Imputed.Churn.Training.Set$total\_day\_charge/Imputed.Churn.Training.Set$total\_number\_charges)

# Adds a new column that is the percentage of charges during the evening  
  
Imputed.Churn.Training.Set$percent\_eve\_charge <- (Imputed.Churn.Training.Set$total\_eve\_charge/Imputed.Churn.Training.Set$total\_number\_charges)

# Adds a new column that is the percentage of charges during the night  
  
Imputed.Churn.Training.Set$percent\_night\_charge <- (Imputed.Churn.Training.Set$total\_night\_charge/Imputed.Churn.Training.Set$total\_number\_charges)

# Adds a new column that is the percentage of charges internationally  
  
Imputed.Churn.Training.Set$percent\_intl\_charge <- (Imputed.Churn.Training.Set$total\_intl\_charge/Imputed.Churn.Training.Set$total\_number\_charges)

The following code splits the data into training and testing data. This is done to prevent the overfitting of the model to the data set.

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

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

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

# This code requires "dplyr" to be run  
  
Imputed.Churn.Training.Set$id <- 1:nrow(Imputed.Churn.Training.Set)  
  
# Create a training data set that takes 80% of the data  
  
Imputed.Churn.Training <- Imputed.Churn.Training.Set %>%  
 sample\_frac(0.80)  
  
# Creates a testing data set from the remaining 20% of the data  
  
Imputed.Churn.Testing <- anti\_join(Imputed.Churn.Training.Set, Imputed.Churn.Training, by = 'id' )

Logistic regression model performed numerous times to find the best fit model. Code for the logistic regression model is found below:

# Create first logistic model for the prediction of churn  
  
Logistic\_Model\_11 <- glm(churn ~ international\_plan +  
 voice\_mail\_plan +  
 total\_day\_charge +  
 number\_customer\_service\_calls +  
 percent\_day\_calls +  
 percent\_eve\_calls +  
 percent\_night\_calls +  
 percent\_day\_minutes +  
 percent\_eve\_minutes +  
 percent\_night\_minutes +  
 percent\_day\_charge, family = "binomial", data = Imputed.Churn.Training)  
  
summary(Logistic\_Model\_11)

##   
## Call:  
## glm(formula = churn ~ international\_plan + voice\_mail\_plan +   
## total\_day\_charge + number\_customer\_service\_calls + percent\_day\_calls +   
## percent\_eve\_calls + percent\_night\_calls + percent\_day\_minutes +   
## percent\_eve\_minutes + percent\_night\_minutes + percent\_day\_charge,   
## family = "binomial", data = Imputed.Churn.Training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9521 -0.4934 -0.3182 -0.1851 3.1576   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 30.09129 15.41332 1.952 0.050904 .   
## international\_planyes 2.07624 0.17041 12.184 < 2e-16 \*\*\*  
## voice\_mail\_planyes -0.87748 0.16703 -5.253 1.49e-07 \*\*\*  
## total\_day\_charge 0.21682 0.01637 13.245 < 2e-16 \*\*\*  
## number\_customer\_service\_calls 0.51721 0.04733 10.928 < 2e-16 \*\*\*  
## percent\_day\_calls 32.47468 9.01910 3.601 0.000317 \*\*\*  
## percent\_eve\_calls 31.42544 9.06703 3.466 0.000528 \*\*\*  
## percent\_night\_calls 32.26506 9.08709 3.551 0.000384 \*\*\*  
## percent\_day\_minutes -62.65211 12.63633 -4.958 7.12e-07 \*\*\*  
## percent\_eve\_minutes -65.16183 13.19724 -4.938 7.91e-07 \*\*\*  
## percent\_night\_minutes -63.98341 13.18266 -4.854 1.21e-06 \*\*\*  
## percent\_day\_charge -16.92774 1.86570 -9.073 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2049.5 on 2509 degrees of freedom  
## Residual deviance: 1551.5 on 2498 degrees of freedom  
## (156 observations deleted due to missingness)  
## AIC: 1575.5  
##   
## Number of Fisher Scoring iterations: 6

Code below is used to check the performance of the model with the AUC values.

# Checking the accuracy of the model against the testing set of data pulled from the model.  
  
library(pROC)

## Warning: package 'pROC' was built under R version 3.4.4

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

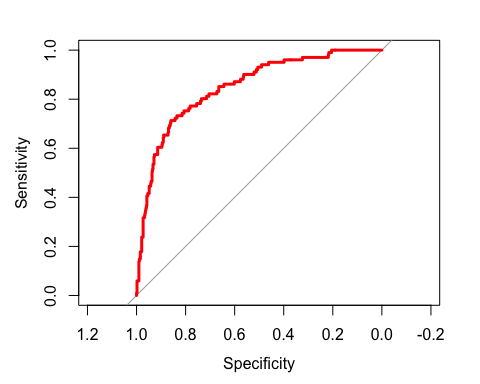
## The following object is masked from 'package:colorspace':  
##   
## coords

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

Imputed.Churn.Testing$Predicted\_Values <- predict(Logistic\_Model\_11, newdata = Imputed.Churn.Testing, type = "response")  
  
roc(Imputed.Churn.Testing$churn, Imputed.Churn.Testing$Predicted\_Values)

##   
## Call:  
## roc.default(response = Imputed.Churn.Testing$churn, predictor = Imputed.Churn.Testing$Predicted\_Values)  
##   
## Data: Imputed.Churn.Testing$Predicted\_Values in 522 controls (Imputed.Churn.Testing$churn no) < 101 cases (Imputed.Churn.Testing$churn yes).  
## Area under the curve: 0.8498

plot(roc(Imputed.Churn.Testing$churn, Imputed.Churn.Testing$Predicted\_Values), col = "red", lwd = 3)



Business Insights of Significant Variables:

International Plan

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,international\_plan), count = n(), percent\_churn = 100\*(sum(churn=="yes")/(sum(churn=="yes")+sum(churn=="no")))) %>%  
 arrange(desc(percent\_churn))

## Warning: package 'bindrcpp' was built under R version 3.4.4

## # A tibble: 2 x 3  
## international\_plan count percent\_churn  
## <fct> <int> <dbl>  
## 1 yes 323 42.4  
## 2 no 3010 11.5

We can see that customers with an international plan appear to churn 4x the rate of customers without an international plan.

Voicemail Plan

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,voice\_mail\_plan), count = n(), percent\_churn = 100\*(sum(churn=="yes")/(sum(churn=="yes")+sum(churn=="no")))) %>%  
 arrange(desc(percent\_churn))

## # A tibble: 2 x 3  
## voice\_mail\_plan count percent\_churn  
## <fct> <int> <dbl>  
## 1 no 2411 16.7   
## 2 yes 922 8.68

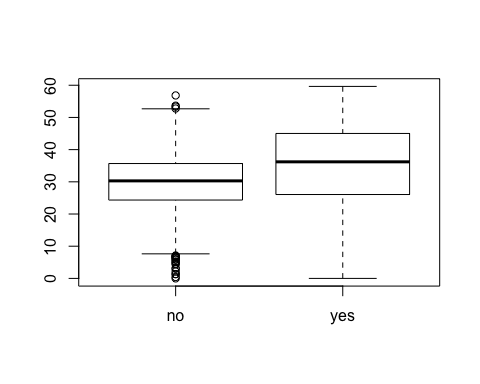
From review of the data, we see that customers without a voicemail plan have a churn rate 2x that of customers with a voicemail plan.

Total Day Charges

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_day\_charge = mean(total\_day\_charge), median\_day\_charges = median(total\_day\_charge))

## # A tibble: 2 x 4  
## churn count mean\_day\_charge median\_day\_charges  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 29.8 30.3  
## 2 yes 483 35.1 36.2

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$total\_day\_charge)



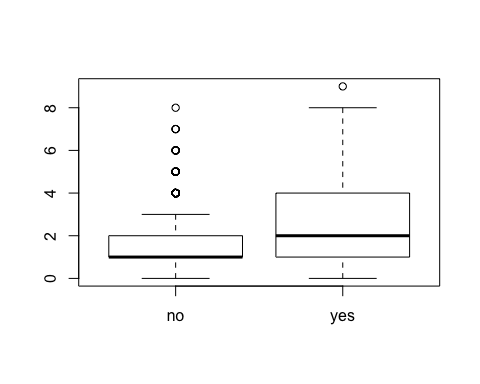
From review of the data, we see that customers that churn have a higher total day charge rate on average. The median total day charge is approximately 22.5% higher for customers that churn.

Number Customer Service Calls:

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_customer\_service = mean(number\_customer\_service\_calls), median\_customer\_service = median(number\_customer\_service\_calls))

## # A tibble: 2 x 4  
## churn count mean\_customer\_service median\_customer\_service  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 1.45 1  
## 2 yes 483 2.23 2

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$number\_customer\_service\_calls)



summarise(group\_by(Imputed.Churn.Training.Set,number\_customer\_service\_calls), count = n(), percent\_churn = 100\*(sum(churn=="yes")/(sum(churn=="yes")+sum(churn=="no"))))

## # A tibble: 10 x 3  
## number\_customer\_service\_calls count percent\_churn  
## <int> <int> <dbl>  
## 1 0 696 13.1   
## 2 1 1190 10.3   
## 3 2 753 12.1   
## 4 3 426 9.62  
## 5 4 167 44.3   
## 6 5 66 62.1   
## 7 6 22 68.2   
## 8 7 10 60   
## 9 8 2 50   
## 10 9 1 100

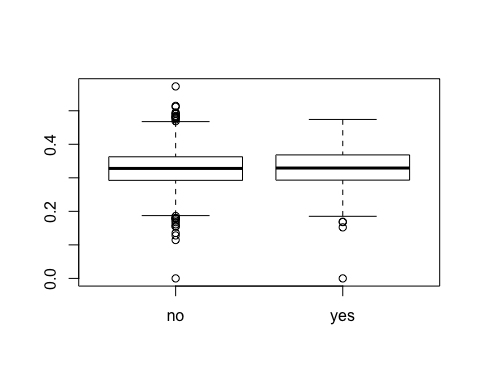
After review of the data, customers are much more likely to churn if they’ve called customer service 4 times or more. Churn % increases from 10% to at least 40% after the 3rd customer service call.

Percent Day Calls

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_day\_percentage = mean(percent\_day\_calls), median\_day\_calls = median(percent\_day\_calls))

## # A tibble: 2 x 4  
## churn count mean\_day\_percentage median\_day\_calls  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.328 0.328  
## 2 yes 483 0.330 0.329

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_day\_calls)



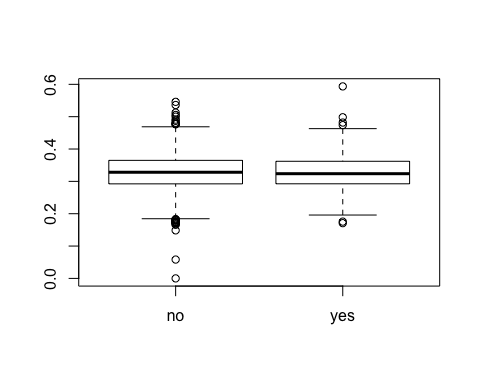
Do not see an obvious negative/positive correlation between percentage of day calls and churn.

Percent Evening Calls

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_eve\_percentage = mean(percent\_eve\_calls), median\_eve\_calls = median(percent\_eve\_calls))

## # A tibble: 2 x 4  
## churn count mean\_eve\_percentage median\_eve\_calls  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.329 0.328  
## 2 yes 483 0.328 0.324

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_eve\_calls)



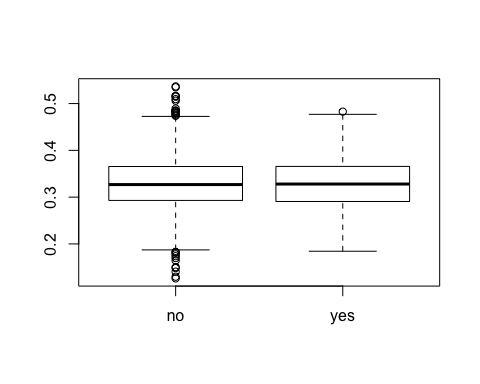
Do not see an obvious negative/positive correlation between percentage of eve calls and churn.

Percent Night Calls

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_night\_percentage = mean(percent\_night\_calls), median\_night\_calls = median(percent\_night\_calls))

## # A tibble: 2 x 4  
## churn count mean\_night\_percentage median\_night\_calls  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.329 0.327  
## 2 yes 483 0.328 0.328

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_night\_calls)



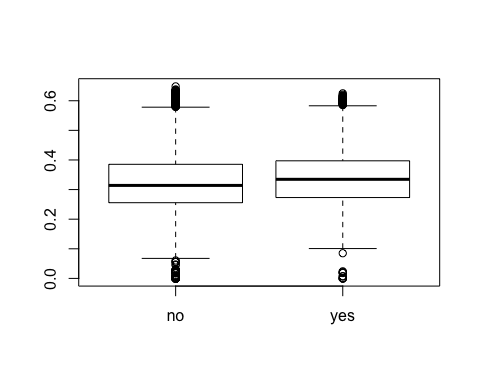
Do not see an obvious negative/positive correlation between percentage of eve calls and churn.

Percent Day Minutes

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_day\_percent\_minutes = mean(percent\_day\_minutes), median\_day\_percent\_minutes = median(percent\_day\_minutes))

## # A tibble: 2 x 4  
## churn count mean\_day\_percent\_minutes median\_day\_percent\_minutes  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.335 0.314  
## 2 yes 483 0.342 0.335

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_day\_minutes)



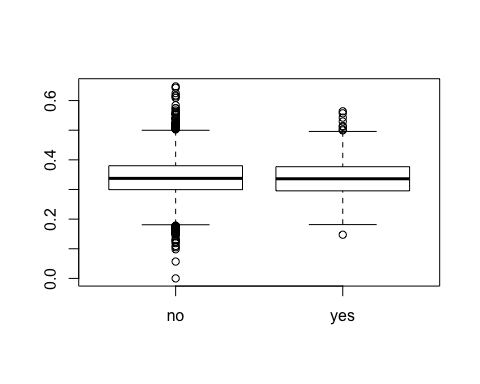
Customers churning have a slightly higher day usage percentage than customers that do not (~ 1%)

Percent Evening Minutes

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_eve\_percent\_minutes = mean(percent\_eve\_minutes), median\_eve\_percent\_minutes = median(percent\_eve\_minutes))

## # A tibble: 2 x 4  
## churn count mean\_eve\_percent\_minutes median\_eve\_percent\_minutes  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.340 0.337  
## 2 yes 483 0.337 0.336

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_eve\_minutes)



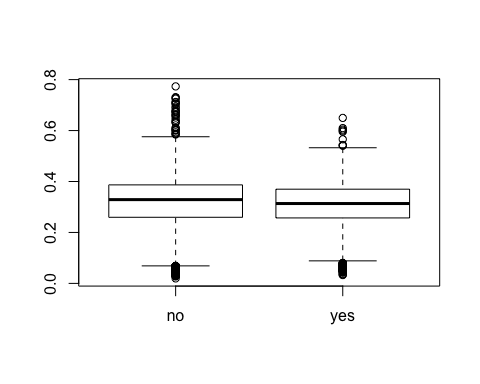
Do not see an obvious negative/positive correlation between percentage of eve minutes and churn.

Percent Night Minutes

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_night\_percent\_minutes = mean(percent\_night\_minutes), median\_night\_percent\_minutes = median(percent\_night\_minutes))

## # A tibble: 2 x 4  
## churn count mean\_night\_percent\_minutes median\_night\_percent\_minutes  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 0.309 0.329  
## 2 yes 483 0.304 0.313

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_night\_minutes)



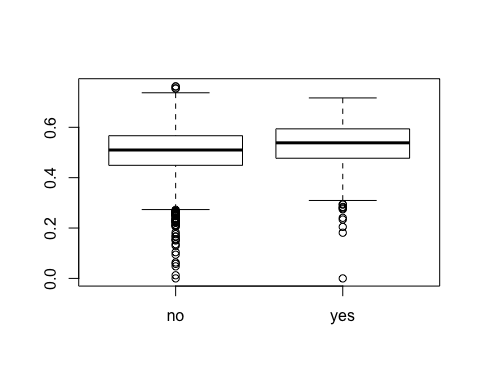
Customers churning have a slightly less night percentage of minutes used than those customers not churning (~1%)

Percent Day Charge

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,churn), count = n(), mean\_day\_charge\_percent = mean(percent\_day\_charge), median\_day\_charge\_percent = median(percent\_day\_charge))

## # A tibble: 2 x 4  
## churn count mean\_day\_charge\_percent median\_day\_charge\_percent  
## <fct> <int> <dbl> <dbl>  
## 1 no 2850 NA NA  
## 2 yes 483 NA NA

plot(Imputed.Churn.Training.Set$churn,Imputed.Churn.Training.Set$percent\_day\_charge)



Customers churning have a slightly higher percent of day charges than those customers not churning.

State

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,state), count = n(), percent\_churn = 100\*(sum(churn=="yes")/(sum(churn=="yes")+sum(churn=="no")))) %>%  
 arrange(desc(percent\_churn))

## # A tibble: 51 x 3  
## state count percent\_churn  
## <fct> <int> <dbl>  
## 1 CA 34 26.5  
## 2 NJ 68 26.5  
## 3 TX 72 25   
## 4 MD 70 24.3  
## 5 SC 60 23.3  
## 6 MI 73 21.9  
## 7 MS 65 21.5  
## 8 NV 66 21.2  
## 9 WA 66 21.2  
## 10 ME 62 21.0  
## # ... with 41 more rows

Area Code

library(dplyr)  
  
summarise(group\_by(Imputed.Churn.Training.Set,area\_code), count = n(), percent\_churn = 100\*(sum(churn=="yes")/(sum(churn=="yes")+sum(churn=="no")))) %>%  
 arrange(desc(percent\_churn))

## # A tibble: 3 x 3  
## area\_code count percent\_churn  
## <fct> <int> <dbl>  
## 1 area\_code\_510 840 14.9  
## 2 area\_code\_408 838 14.6  
## 3 area\_code\_415 1655 14.3

The code below is for predicting the values of the 1,000 customers provided in the class project.

# Adds a new column that is the sum of all the total calls taken by a customer  
  
Customers\_To\_Predict$total\_number\_calls <- rowSums(Customers\_To\_Predict[,c("total\_day\_calls","total\_eve\_calls","total\_night\_calls","total\_intl\_calls")])

# Adds a new column that is the sum of all the minutes taken by a customer  
  
Customers\_To\_Predict$total\_number\_minutes <- rowSums(Customers\_To\_Predict[,c("total\_day\_minutes","total\_eve\_minutes","total\_night\_minutes","total\_intl\_minutes")])

# Adds a new column that is the sum of all the charges by a customer  
  
Customers\_To\_Predict$total\_number\_charges <- rowSums(Customers\_To\_Predict[,c("total\_day\_charge","total\_eve\_charge","total\_night\_charge","total\_intl\_charge")])

# Adds a new column that is the average cost per call by customer  
  
Customers\_To\_Predict$average\_call\_cost <- (Customers\_To\_Predict$total\_number\_charges/Customers\_To\_Predict$total\_number\_calls)

# Adds a new column that is the average cost per minute by customer  
  
Customers\_To\_Predict$average\_minute\_cost <- (Customers\_To\_Predict$total\_number\_charges / Customers\_To\_Predict$total\_number\_minutes)

# Adds a new column that is the average number of calls the customer completed a day  
  
Customers\_To\_Predict$average\_calls\_per\_day <- (Customers\_To\_Predict$total\_number\_calls/Customers\_To\_Predict$account\_length)

# Adds a new column that is the average number of customer service calls per day  
  
Customers\_To\_Predict$average\_service\_calls\_per\_day <- (Customers\_To\_Predict$number\_customer\_service\_calls/Customers\_To\_Predict$account\_length)

# Adds a new column that is the average charge of customer per day  
  
Customers\_To\_Predict$average\_cost\_per\_day <- (Customers\_To\_Predict$total\_number\_charges/Customers\_To\_Predict$account\_length)

# Adds a new column that is the percentage of calls performed during the day  
  
Customers\_To\_Predict$percent\_day\_calls <- (Customers\_To\_Predict$total\_day\_calls/Customers\_To\_Predict$total\_number\_calls)

# Adds a new column that is the percentage of calls performed during the evening  
  
Customers\_To\_Predict$percent\_eve\_calls <- (Customers\_To\_Predict$total\_eve\_calls/Customers\_To\_Predict$total\_number\_calls)

# Adds a new column that is the percentage of calls performed during the night  
  
Customers\_To\_Predict$percent\_night\_calls <- (Customers\_To\_Predict$total\_night\_calls/Customers\_To\_Predict$total\_number\_calls)

# Adds a new column that is the percentage of calls performed international  
  
Customers\_To\_Predict$percent\_intl\_calls <- (Customers\_To\_Predict$total\_intl\_calls/Customers\_To\_Predict$total\_number\_calls)

# Adds a new column that is the percentage of minutes performed during the day  
  
Customers\_To\_Predict$percent\_day\_minutes <- (Customers\_To\_Predict$total\_day\_minutes/Customers\_To\_Predict$total\_number\_minutes)

# Adds a new column that is the percentage of minutes performed during the evening  
  
Customers\_To\_Predict$percent\_eve\_minutes <- (Customers\_To\_Predict$total\_eve\_minutes/Customers\_To\_Predict$total\_number\_minutes)

# Adds a new column that is the percentage of minutes performed during the night  
  
Customers\_To\_Predict$percent\_night\_minutes <- (Customers\_To\_Predict$total\_night\_minutes/Customers\_To\_Predict$total\_number\_minutes)

# Adds a new column that is the percentage of minutes internationally  
  
Customers\_To\_Predict$percent\_intl\_minutes <- (Customers\_To\_Predict$total\_intl\_minutes/Customers\_To\_Predict$total\_number\_minutes)

# Adds a new column that is the percentage of charges during the day  
  
Customers\_To\_Predict$percent\_day\_charge <- (Customers\_To\_Predict$total\_day\_charge/Customers\_To\_Predict$total\_number\_charges)

# Adds a new column that is the percentage of charges during the evening  
  
Customers\_To\_Predict$percent\_eve\_charge <- (Customers\_To\_Predict$total\_eve\_charge/Customers\_To\_Predict$total\_number\_charges)

# Adds a new column that is the percentage of charges during the night  
  
Customers\_To\_Predict$percent\_night\_charge <- (Customers\_To\_Predict$total\_night\_charge/Customers\_To\_Predict$total\_number\_charges)

# Adds a new column that is the percentage of charges internationally  
  
Customers\_To\_Predict$percent\_intl\_charge <- (Customers\_To\_Predict$total\_intl\_charge/Customers\_To\_Predict$total\_number\_charges)

# Creating new variable "churn\_prob" to capture the probability values that a customer may churn.  
  
Churn\_Prob <- predict(Logistic\_Model\_11, newdata = Customers\_To\_Predict, type = "response")

