Predict the customer churn

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

plot(cars)



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

#### 1.Importing Libraries

library(plyr) # for data manipulation  
library(dplyr) # for data-preprocessing and data manipulation  
library(tidyverse) # for dplyr,ggplot  
library(ggplot2) #for ggplots  
library(caret) #for data splitting , modeling tuning , pre-processing  
library(party) # for decision tree  
library(ggcorrplot) # for correlations.  
library(stats) # for the stepwise search.  
library(rpart) # for decision tree  
library(rpart.plot) #for decision tree  
library(mice) # for Imputation for NA values  
library(VIM) # for plotting the amount of missing values

#### 2. Loading and Reading the Dataset

#setwd("~/Documents/BusinessAnalyticsGroupProject/R code and Script")   
#Loading the training data to analyze and build model  
Churn\_Train <- read\_csv("Churn\_Train.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_double(),  
## state = col\_character(),  
## area\_code = col\_character(),  
## international\_plan = col\_character(),  
## voice\_mail\_plan = col\_character(),  
## churn = col\_character()  
## )  
## i Use `spec()` for the full column specifications.

#Loading the file containing the list of consumers that we need to predict their future churn  
load("Customers\_To\_Predict.RData")  
# removed the "area\_code\_" part of the string in "area\_code" variable.  
Churn\_Train$area\_code <- as.factor(sub("area\_code\_", "", Churn\_Train$area\_code))   
Customers\_To\_Predict$area\_code <- as.factor(sub("area\_code\_", "", Customers\_To\_Predict$area\_code))

#### 3. Analysing Count of the NA Values in the Dataset

summary(Churn\_Train)

## state account\_length area\_code international\_plan  
## Length:3333 Min. :-209.00 408: 838 Length:3333   
## Class :character 1st Qu.: 72.00 415:1655 Class :character   
## Mode :character Median : 100.00 510: 840 Mode :character   
## Mean : 97.32   
## 3rd Qu.: 127.00   
## Max. : 243.00   
## NA's :501   
## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## Length:3333 Min. :-10.000 Min. : 0.0 Min. : 0.0   
## Class :character 1st Qu.: 0.000 1st Qu.: 149.3 1st Qu.: 87.0   
## Mode :character 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 total\_intl\_minutes  
## Min. : 23.2 Min. : 33.0 Min. : 1.040 Min. : 0.00   
## 1st Qu.:167.3 1st Qu.: 87.0 1st Qu.: 7.530 1st Qu.: 8.50   
## Median :201.4 Median :100.0 Median : 9.060 Median :10.30   
## Mean :201.2 Mean :100.1 Mean : 9.054 Mean :10.23   
## 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   
## NA's :200 NA's :200 NA's :200   
## total\_intl\_calls total\_intl\_charge number\_customer\_service\_calls  
## Min. : 0.00 Min. :0.000 Min. :0.000   
## 1st Qu.: 3.00 1st Qu.:2.300 1st Qu.:1.000   
## Median : 4.00 Median :2.780 Median :1.000   
## Mean : 4.47 Mean :2.762 Mean :1.561   
## 3rd Qu.: 6.00 3rd Qu.:3.270 3rd Qu.:2.000   
## Max. :20.00 Max. :5.400 Max. :9.000   
## NA's :301 NA's :200 NA's :200   
## churn   
## Length:3333   
## Class :character   
## Mode :character   
##   
##   
##   
##

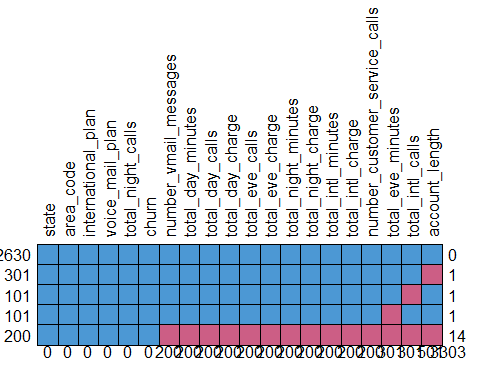
sapply(Churn\_Train, function(x) sum(is.na(x))) # NA data

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

sapply(Customers\_To\_Predict, function(x) sum(is.na(x))) # no NA data

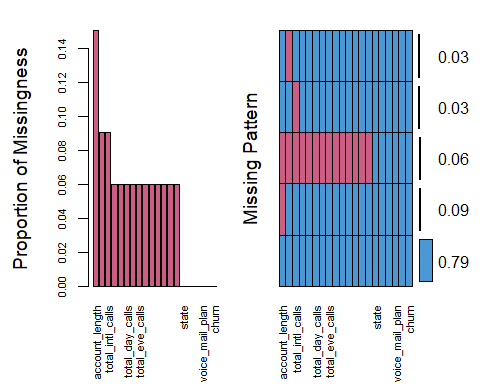
## state account\_length   
## 0 0   
## area\_code international\_plan   
## 0 0   
## voice\_mail\_plan number\_vmail\_messages   
## 0 0   
## total\_day\_minutes total\_day\_calls   
## 0 0   
## total\_day\_charge total\_eve\_minutes   
## 0 0   
## total\_eve\_calls total\_eve\_charge   
## 0 0   
## total\_night\_minutes total\_night\_calls   
## 0 0   
## total\_night\_charge total\_intl\_minutes   
## 0 0   
## total\_intl\_calls total\_intl\_charge   
## 0 0   
## number\_customer\_service\_calls   
## 0

md.pattern(Churn\_Train,rotate.names = TRUE) # See the missing value pattern



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

# Plot the missing values  
aggr(Churn\_Train, col = mdc(1:2), numbers = TRUE, sortVars = TRUE, labels = names(Churn\_Train), 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

As we can see that the current dataset Churn\_Train has lots of NA values .Removing them from the dataset will be lead to the loss of useful information and can imapct data analysis and model predictions.

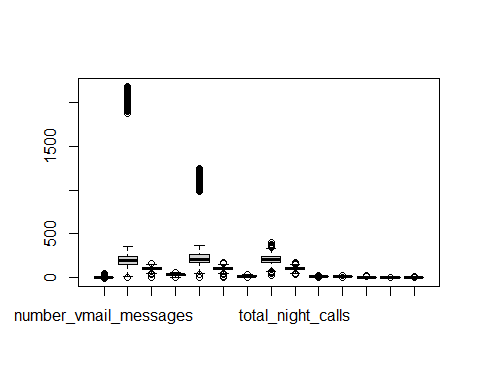
Therefore , we can use mice()to impute NA values. It creates multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values.

#### 4.Imputing Missing Values

## Warning: Number of logged events: 354

#### 5. Exploratory Data Analysis

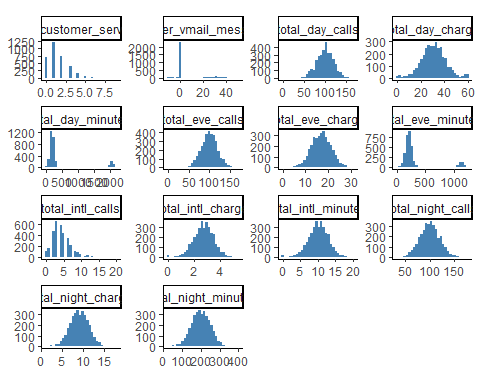
boxplot(Churn\_Train[, 6:19])

 Interpretation of Boxplot:

Based on the boxplot chart , we can see that most of the variables in the Churn\_Train dataset are normally distributed with an exception of “total day minutes” and “total evening minutes” with outliers.

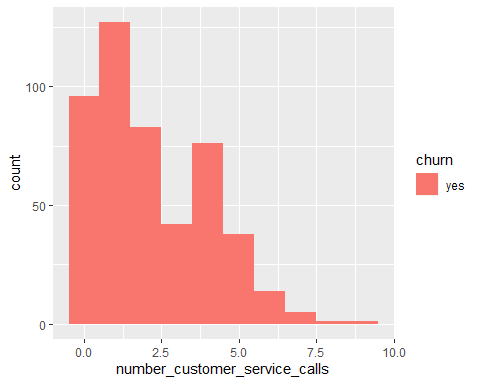
Similarly, we can see below the individual graphs displaying distribution for each variable .

Churn\_Train[, 6:19] %>%   
 gather(key = Variable, value = Value) %>%   
 ggplot() +  
 geom\_histogram(aes(x = Value), fill = "steelblue") +  
 facet\_wrap(~Variable, scales='free') +  
 theme\_classic() +  
 theme(aspect.ratio = 0.5, axis.title = element\_blank(), panel.grid = element\_blank())

 Interpretation : We can clearly see the beautiful bell curve distribution of data for most of the variables.

“Total day minutes” and “total evening minutes” have some small no. of outliers.Also,“Customer service calls” data is also rightly skewed.

Churn\_Train %>%   
 filter(churn == "yes") %>%  
 ggplot(mapping = aes(x = number\_customer\_service\_calls)) +  
 geom\_histogram(aes(fill = churn), binwidth = 1) # Showing the number of customer service calls per churned customer.



Churn\_Train %>%   
 group\_by(churn) %>%   
 tally(churn == "yes") # total churned in data set.

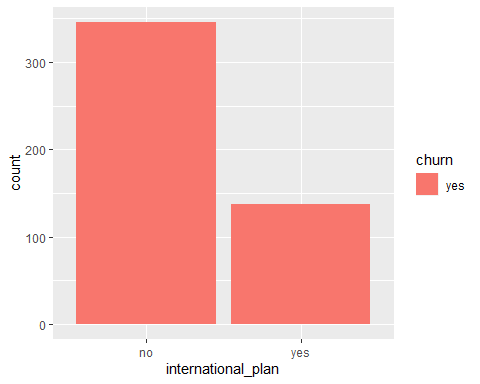
## # A tibble: 2 x 2  
## churn n  
## <chr> <int>  
## 1 no 0  
## 2 yes 483

Churn\_Train %>%   
 filter(churn == "yes" & number\_customer\_service\_calls >= 1 & number\_customer\_service\_calls <= 4) %>%   
 tally()/483 # 67% of all the customers who churned made 1 to 4 calls to customer service.

## n  
## 1 0.679089

Churn\_Train %>%   
 filter(churn == "yes") %>%  
 ggplot(mapping = aes(x = international\_plan)) +  
 geom\_histogram(aes(fill = churn), stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad

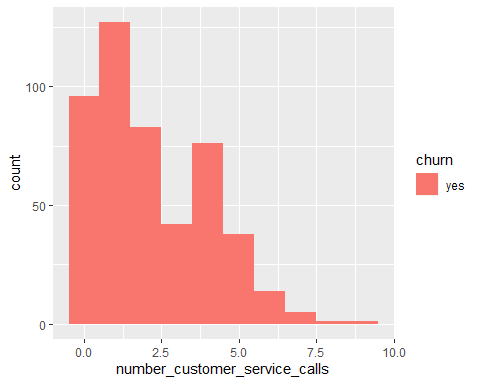


Churn\_Train %>%   
 group\_by(international\_plan) %>%   
 filter(churn == "yes") %>%   
 #select(international\_plan) %>%   
 dplyr:: summarise("Churn Count" =n(), "Percent" = n()/483)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 3  
## international\_plan `Churn Count` Percent  
## <chr> <int> <dbl>  
## 1 no 346 0.716  
## 2 yes 137 0.284

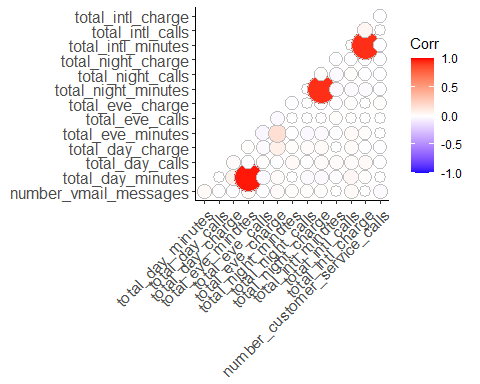
# 28% of all international plan subscribers will churn.  
  
Churn\_Train %>%   
 filter(churn == "yes") %>%  
 ggplot(mapping = aes(x = number\_customer\_service\_calls)) +  
 geom\_histogram(aes(fill = churn), binwidth = 1)

 #### 5.2 Correlation between variables of Train\_Churn and dataset.

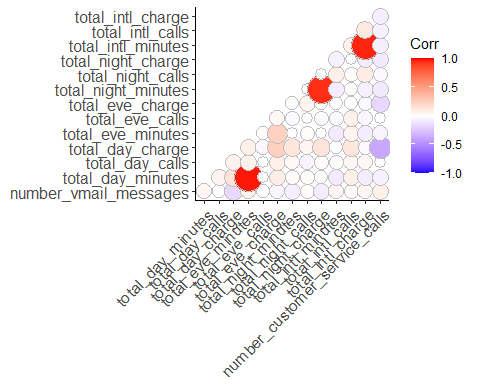
Here, we will analyze the correlation of variables with the following condition:

1. Complete Train\_Churn dataset and
2. when churn== Yes

Churn\_Train %>%   
 filter(churn=="yes") -> churn  
cor(Churn\_Train[, 6:19]) -> cc  
cor(churn[, 6:19]) ->cc2  
# ggplot to determine the correlation between variables in the Churn\_Train dataset  
ggcorrplot(cc, method = "circle", type = "lower", ggtheme = theme\_classic)



# ggplot to determine the correlation between variables when the customer have churn  
ggcorrplot(cc2, method = "circle", type = "lower", ggtheme = theme\_classic)

 Interpretation of the correlation chart:

#####Complete Churn\_Train dataset:-

#### Positive Relation :

1. total evening minutes and total day minutes
2. total evening charges and total evening minutes
3. total night charges and total night minutes
4. total international charge and total international minutes

##### When (Churn==""Yes)

#### Positive Correlation :

1.total evening minutes and total day minutes

2.total night charge and total night minutes

3.total international charge and total international minutes

#### Negative Correlation :

1.number customer service calls and total day charge,total evening charge,total night minutes,total international calls and charges

2.total day charge and number of voice mail messages

3.total evening charges and total evening charge

4.total night charge and total day charge

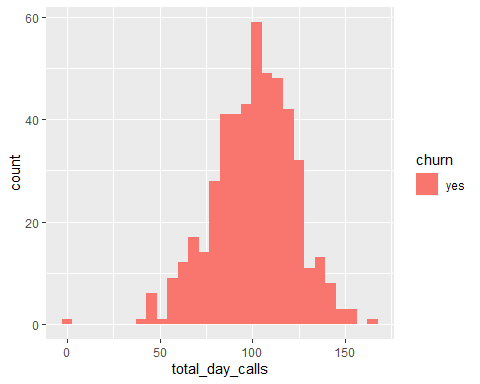
Strong negative correlation between total day minutes and total evening minutes. Meaning that the as the evening minutes increase the total day minutes decrease. Also, a slight negative correlation between the total evening minutes and the total evening charges.

Looking at the correlation of just the people who churned, some possible interesting information appeared. There is a strong correlation between the totals day charges and the number of Customer Service Calls. The higher the charges the more calls were made. The same was true for customer service calls and total evening charges although less of a relationship compared to day charges.

Lets , Analyze data in more detail for total day calls , number of customer service calls and total day charge.

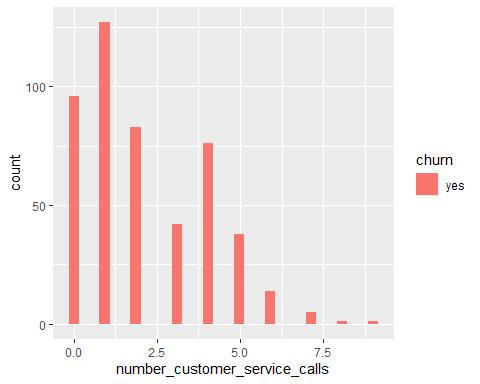
Churn\_Train %>%   
 filter(churn=="yes") %>%   
 ggplot(mapping = aes(x = total\_day\_calls)) +  
 geom\_histogram(aes(fill = churn))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



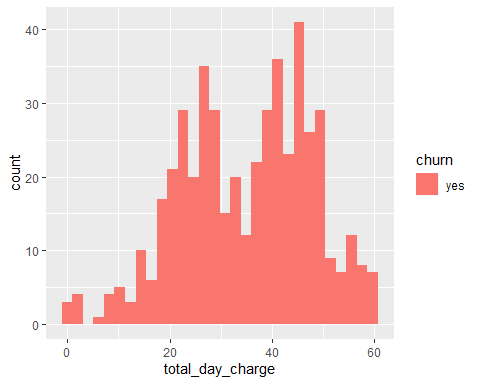
Churn\_Train %>%   
 filter(churn=="yes") %>%   
 ggplot(mapping = aes(x = number\_customer\_service\_calls)) +  
 geom\_histogram(aes(fill = churn))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Churn\_Train %>%   
 filter(churn=="yes") %>%   
 ggplot(mapping = aes(x = total\_day\_charge)) +  
 geom\_histogram(aes(fill = churn))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Most of the people seem to churn between 75 to 125 calls per day, making 1 to 5 customer service calls, and when the charges are between 10 and 60 per day.

Based on the above, I might suggest that the reason people are churning is that the cost of daily phone call charges during the day are too much. FYI I think this data is really old as I remember when Cell Phone companies used to charge more for calls made during the day than the evening.

# 6.Data Pre-Processing

## 1.Updating the values of churn to 1 or 0  
Churn\_Train$churn<- ifelse(Churn\_Train$churn=="yes",1,0)  
##2.Factorization of Churn\_Train data  
Churn\_Train$area\_code<- as.factor(Churn\_Train$area\_code) # added because of decision trees  
Churn\_Train$state<- as.factor(Churn\_Train$state)  
Churn\_Train$international\_plan<-as.factor(Churn\_Train$international\_plan)  
Churn\_Train$voice\_mail\_plan <-as.factor(Churn\_Train$voice\_mail\_plan)  
Churn\_Train$churn<- as.factor(Churn\_Train$churn)  
## 3.Validating the structure of the Churn\_Train data  
str(Churn\_Train)

## '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 : num 125 108 82 82 83 89 135 28 86 65 ...  
## $ area\_code : Factor w/ 3 levels "408","415","510": 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 : num 0 0 0 30 0 0 0 0 0 0 ...  
## $ total\_day\_minutes : num 2013 292 300 110 337 ...  
## $ total\_day\_calls : num 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 : num 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 : num 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 : num 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: num 0 2 0 2 0 1 1 3 2 4 ...  
## $ churn : Factor w/ 2 levels "0","1": 1 2 2 1 2 1 1 1 1 2 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. state = col\_character(),  
## .. account\_length = col\_double(),  
## .. area\_code = col\_character(),  
## .. international\_plan = col\_character(),  
## .. voice\_mail\_plan = col\_character(),  
## .. number\_vmail\_messages = col\_double(),  
## .. total\_day\_minutes = col\_double(),  
## .. total\_day\_calls = col\_double(),  
## .. total\_day\_charge = col\_double(),  
## .. total\_eve\_minutes = col\_double(),  
## .. total\_eve\_calls = col\_double(),  
## .. total\_eve\_charge = col\_double(),  
## .. total\_night\_minutes = col\_double(),  
## .. total\_night\_calls = col\_double(),  
## .. total\_night\_charge = col\_double(),  
## .. total\_intl\_minutes = col\_double(),  
## .. total\_intl\_calls = col\_double(),  
## .. total\_intl\_charge = col\_double(),  
## .. number\_customer\_service\_calls = col\_double(),  
## .. churn = col\_character()  
## .. )

## 7.Choice of Models:

Decision trees and logistic regression are two very popular algorithms and can be used to customer churn prediction with strong predictive performance and good comprehensibility.

Therefore, we will be using classification model such as a logistic regression and decision tree model to determine the predictive ability for each model. Based on the results , we will choose one model to predict Customer churn probability.

## 8.Determining the predictive ability of Logistic regression and Decision trees models :

##### Steps:

1. Partitioning the Churn\_Train data into train\_data and validation\_data.
2. Building Decision Tree model and Predicting the results on the validation dataset and using confusion matrix to validate the performance.
3. Building Logistic Regression model and Predicting the results on the validation data set and using confusion matrix to validate the performance of the model.
4. Comparing the results and Selecting model.

#### 8.1 Churn Train data partitioning (60%,40%)

set.seed(2020)  
partition<- createDataPartition(Churn\_Train$churn,p=0.6,list=FALSE)  
train\_data<- Churn\_Train[partition,]  
validation\_data<- Churn\_Train[-partition,]

#### 8.2 Building Decision tree model:

# Decision Tree   
DecisionTree\_model <- ctree(churn~ ., train\_data[,-1]) #not including state column  
pred\_tree <- predict(DecisionTree\_model, validation\_data)  
#table  
table(pred\_tree)

## pred\_tree  
## 0 1   
## 1179 154

#### 8.3 Confusion matrix for decision trees

#confusion matrix  
confusionMatrix(pred\_tree,validation\_data$churn) ## without states

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1112 67  
## 1 28 126  
##   
## Accuracy : 0.9287   
## 95% CI : (0.9136, 0.942)  
## No Information Rate : 0.8552   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6859   
##   
## Mcnemar's Test P-Value : 9.67e-05   
##   
## Sensitivity : 0.9754   
## Specificity : 0.6528   
## Pos Pred Value : 0.9432   
## Neg Pred Value : 0.8182   
## Prevalence : 0.8552   
## Detection Rate : 0.8342   
## Detection Prevalence : 0.8845   
## Balanced Accuracy : 0.8141   
##   
## 'Positive' Class : 0   
##

#### 8.4 Building Logistic Regression Model:

#Note:Model performance got improved after removing "states"  
## Applying logistic regression model   
Logistic\_Model <- glm(churn ~ .,family=binomial(link="logit"),data=train\_data[,-1])  
summary(Logistic\_Model)

##   
## Call:  
## glm(formula = churn ~ ., family = binomial(link = "logit"), data = train\_data[,   
## -1])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9413 -0.5232 -0.3579 -0.2195 3.1225   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.5111846 0.8989306 -7.243 4.38e-13 \*\*\*  
## account\_length -0.0024384 0.0014274 -1.708 0.08757 .   
## area\_code415 -0.0296486 0.1724127 -0.172 0.86347   
## area\_code510 -0.2058340 0.2003333 -1.027 0.30421   
## international\_planyes 1.9492120 0.1847137 10.553 < 2e-16 \*\*\*  
## voice\_mail\_planyes -0.9899579 0.3610243 -2.742 0.00611 \*\*   
## number\_vmail\_messages -0.0017586 0.0120610 -0.146 0.88408   
## total\_day\_minutes -0.0023771 0.0024530 -0.969 0.33253   
## total\_day\_calls 0.0009815 0.0036103 0.272 0.78574   
## total\_day\_charge 0.0663555 0.0143298 4.631 3.65e-06 \*\*\*  
## total\_eve\_minutes 0.0045165 0.0048369 0.934 0.35042   
## total\_eve\_calls 0.0005137 0.0035216 0.146 0.88401   
## total\_eve\_charge 0.0106394 0.0591397 0.180 0.85723   
## total\_night\_minutes 0.0062339 0.0042252 1.475 0.14010   
## total\_night\_calls -0.0025007 0.0036391 -0.687 0.49198   
## total\_night\_charge -0.0662782 0.0930372 -0.712 0.47623   
## total\_intl\_minutes 0.0051183 0.0688572 0.074 0.94075   
## total\_intl\_calls -0.0718708 0.0304122 -2.363 0.01812 \*   
## total\_intl\_charge 0.3608120 0.2542138 1.419 0.15580   
## number\_customer\_service\_calls 0.4567319 0.0497370 9.183 < 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: 1655.7 on 1999 degrees of freedom  
## Residual deviance: 1334.4 on 1980 degrees of freedom  
## AIC: 1374.4  
##   
## Number of Fisher Scoring iterations: 5

## Predicting churn results based on the logistic model  
predict\_validation<-predict(Logistic\_Model,newdata = validation\_data,type='response')  
## Categorizing the result based on the cutoff value(0.5)  
resultcheck<-ifelse(predict\_validation>0.5,1,0)

#### 8.5 Building Improvised Logistic Regression model

Logistic\_Model2 <-glm(formula = churn ~ account\_length + area\_code + international\_plan +   
 voice\_mail\_plan + number\_vmail\_messages + total\_day\_minutes +   
 total\_day\_calls + total\_day\_charge + total\_eve\_minutes +   
 total\_eve\_charge + total\_night\_minutes + total\_night\_charge +   
 total\_intl\_minutes + total\_intl\_calls + number\_customer\_service\_calls +   
 total\_day\_charge:number\_customer\_service\_calls + total\_day\_charge:total\_eve\_charge +   
 voice\_mail\_plan:total\_day\_charge + international\_plan:total\_intl\_minutes +   
 international\_plan:number\_customer\_service\_calls + total\_eve\_charge:number\_customer\_service\_calls +   
 total\_day\_charge:total\_night\_charge + international\_plan:total\_intl\_calls +   
 area\_code:number\_vmail\_messages + voice\_mail\_plan:total\_intl\_calls +   
 total\_intl\_calls:number\_customer\_service\_calls + total\_day\_calls:total\_eve\_charge +   
 number\_vmail\_messages:total\_intl\_calls + international\_plan:total\_day\_calls +   
 voice\_mail\_plan:total\_night\_charge + total\_night\_minutes:number\_customer\_service\_calls +   
 total\_eve\_charge:total\_intl\_calls + voice\_mail\_plan:total\_eve\_charge +   
 total\_eve\_charge:total\_night\_minutes + total\_day\_charge:total\_intl\_calls +   
 area\_code:total\_day\_minutes + international\_plan:total\_eve\_minutes +   
 international\_plan:total\_day\_minutes + international\_plan:total\_eve\_charge +   
 total\_night\_minutes:total\_night\_charge, family = binomial(link = "logit"),   
 data = train\_data)  
#summary  
summary(Logistic\_Model2)

##   
## Call:  
## glm(formula = churn ~ account\_length + area\_code + international\_plan +   
## voice\_mail\_plan + number\_vmail\_messages + total\_day\_minutes +   
## total\_day\_calls + total\_day\_charge + total\_eve\_minutes +   
## total\_eve\_charge + total\_night\_minutes + total\_night\_charge +   
## total\_intl\_minutes + total\_intl\_calls + number\_customer\_service\_calls +   
## total\_day\_charge:number\_customer\_service\_calls + total\_day\_charge:total\_eve\_charge +   
## voice\_mail\_plan:total\_day\_charge + international\_plan:total\_intl\_minutes +   
## international\_plan:number\_customer\_service\_calls + total\_eve\_charge:number\_customer\_service\_calls +   
## total\_day\_charge:total\_night\_charge + international\_plan:total\_intl\_calls +   
## area\_code:number\_vmail\_messages + voice\_mail\_plan:total\_intl\_calls +   
## total\_intl\_calls:number\_customer\_service\_calls + total\_day\_calls:total\_eve\_charge +   
## number\_vmail\_messages:total\_intl\_calls + international\_plan:total\_day\_calls +   
## voice\_mail\_plan:total\_night\_charge + total\_night\_minutes:number\_customer\_service\_calls +   
## total\_eve\_charge:total\_intl\_calls + voice\_mail\_plan:total\_eve\_charge +   
## total\_eve\_charge:total\_night\_minutes + total\_day\_charge:total\_intl\_calls +   
## area\_code:total\_day\_minutes + international\_plan:total\_eve\_minutes +   
## international\_plan:total\_day\_minutes + international\_plan:total\_eve\_charge +   
## total\_night\_minutes:total\_night\_charge, family = binomial(link = "logit"),   
## data = train\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7695 -0.4349 -0.2458 -0.1185 4.0131   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -8.159e-01 3.163e+00  
## account\_length -2.061e-03 1.639e-03  
## area\_code415 2.157e-01 2.476e-01  
## area\_code510 2.139e-01 2.895e-01  
## international\_planyes 5.632e+00 2.011e+00  
## voice\_mail\_planyes 4.031e+00 1.379e+00  
## number\_vmail\_messages 3.988e-02 2.811e-02  
## total\_day\_minutes -7.808e-04 3.062e-03  
## total\_day\_calls -3.016e-02 1.787e-02  
## total\_day\_charge -3.957e-02 5.098e-02  
## total\_eve\_minutes 1.999e-03 6.092e-03  
## total\_eve\_charge -5.105e-01 1.678e-01  
## total\_night\_minutes -2.594e-03 1.050e-02  
## total\_night\_charge -4.674e-01 2.214e-01  
## total\_intl\_minutes 5.580e-02 3.220e-02  
## total\_intl\_calls 9.109e-02 1.761e-01  
## number\_customer\_service\_calls 3.275e+00 4.013e-01  
## total\_day\_charge:number\_customer\_service\_calls -5.306e-02 5.991e-03  
## total\_day\_charge:total\_eve\_charge 8.478e-03 1.814e-03  
## voice\_mail\_planyes:total\_day\_charge -1.103e-01 2.236e-02  
## international\_planyes:total\_intl\_minutes 4.922e-01 1.031e-01  
## international\_planyes:number\_customer\_service\_calls -3.550e-01 1.598e-01  
## total\_eve\_charge:number\_customer\_service\_calls -3.199e-02 1.380e-02  
## total\_day\_charge:total\_night\_charge 8.157e-03 3.845e-03  
## international\_planyes:total\_intl\_calls -4.451e-01 9.761e-02  
## area\_code415:number\_vmail\_messages -9.042e-03 1.757e-02  
## area\_code510:number\_vmail\_messages -2.850e-03 2.051e-02  
## voice\_mail\_planyes:total\_intl\_calls 2.246e-01 1.260e-01  
## total\_intl\_calls:number\_customer\_service\_calls -6.654e-02 2.454e-02  
## total\_day\_calls:total\_eve\_charge 2.113e-03 9.914e-04  
## number\_vmail\_messages:total\_intl\_calls -7.692e-03 4.349e-03  
## international\_planyes:total\_day\_calls -2.352e-02 1.102e-02  
## voice\_mail\_planyes:total\_night\_charge -8.319e-02 9.285e-02  
## total\_night\_minutes:number\_customer\_service\_calls -4.933e-04 1.186e-03  
## total\_eve\_charge:total\_intl\_calls -7.751e-03 7.203e-03  
## voice\_mail\_planyes:total\_eve\_charge -1.068e-01 5.444e-02  
## total\_eve\_charge:total\_night\_minutes 8.941e-04 4.109e-04  
## total\_day\_charge:total\_intl\_calls 5.276e-03 3.305e-03  
## area\_code415:total\_day\_minutes -4.115e-04 3.245e-04  
## area\_code510:total\_day\_minutes -9.636e-04 4.217e-04  
## international\_planyes:total\_eve\_minutes 4.278e-02 7.644e-03  
## international\_planyes:total\_day\_minutes -2.085e-02 3.820e-03  
## international\_planyes:total\_eve\_charge -4.918e-01 9.744e-02  
## total\_night\_minutes:total\_night\_charge 6.354e-05 5.299e-04  
## z value Pr(>|z|)   
## (Intercept) -0.258 0.79644   
## account\_length -1.257 0.20857   
## area\_code415 0.871 0.38363   
## area\_code510 0.739 0.46000   
## international\_planyes 2.800 0.00510 \*\*   
## voice\_mail\_planyes 2.924 0.00346 \*\*   
## number\_vmail\_messages 1.419 0.15595   
## total\_day\_minutes -0.255 0.79875   
## total\_day\_calls -1.687 0.09156 .   
## total\_day\_charge -0.776 0.43760   
## total\_eve\_minutes 0.328 0.74283   
## total\_eve\_charge -3.041 0.00235 \*\*   
## total\_night\_minutes -0.247 0.80481   
## total\_night\_charge -2.111 0.03475 \*   
## total\_intl\_minutes 1.733 0.08316 .   
## total\_intl\_calls 0.517 0.60495   
## number\_customer\_service\_calls 8.161 3.33e-16 \*\*\*  
## total\_day\_charge:number\_customer\_service\_calls -8.856 < 2e-16 \*\*\*  
## total\_day\_charge:total\_eve\_charge 4.674 2.95e-06 \*\*\*  
## voice\_mail\_planyes:total\_day\_charge -4.933 8.08e-07 \*\*\*  
## international\_planyes:total\_intl\_minutes 4.774 1.81e-06 \*\*\*  
## international\_planyes:number\_customer\_service\_calls -2.221 0.02634 \*   
## total\_eve\_charge:number\_customer\_service\_calls -2.319 0.02039 \*   
## total\_day\_charge:total\_night\_charge 2.122 0.03387 \*   
## international\_planyes:total\_intl\_calls -4.560 5.11e-06 \*\*\*  
## area\_code415:number\_vmail\_messages -0.515 0.60680   
## area\_code510:number\_vmail\_messages -0.139 0.88948   
## voice\_mail\_planyes:total\_intl\_calls 1.783 0.07459 .   
## total\_intl\_calls:number\_customer\_service\_calls -2.711 0.00671 \*\*   
## total\_day\_calls:total\_eve\_charge 2.131 0.03306 \*   
## number\_vmail\_messages:total\_intl\_calls -1.768 0.07698 .   
## international\_planyes:total\_day\_calls -2.135 0.03277 \*   
## voice\_mail\_planyes:total\_night\_charge -0.896 0.37028   
## total\_night\_minutes:number\_customer\_service\_calls -0.416 0.67745   
## total\_eve\_charge:total\_intl\_calls -1.076 0.28192   
## voice\_mail\_planyes:total\_eve\_charge -1.962 0.04975 \*   
## total\_eve\_charge:total\_night\_minutes 2.176 0.02957 \*   
## total\_day\_charge:total\_intl\_calls 1.596 0.11045   
## area\_code415:total\_day\_minutes -1.268 0.20481   
## area\_code510:total\_day\_minutes -2.285 0.02229 \*   
## international\_planyes:total\_eve\_minutes 5.596 2.19e-08 \*\*\*  
## international\_planyes:total\_day\_minutes -5.459 4.78e-08 \*\*\*  
## international\_planyes:total\_eve\_charge -5.047 4.48e-07 \*\*\*  
## total\_night\_minutes:total\_night\_charge 0.120 0.90456   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1655.7 on 1999 degrees of freedom  
## Residual deviance: 1055.7 on 1956 degrees of freedom  
## AIC: 1143.7  
##   
## Number of Fisher Scoring iterations: 6

#Predicting the validation data based on the improvised logistic Regression model  
predict\_validation2<-predict(Logistic\_Model2,newdata = validation\_data,type='response')  
#Classify the data based on the value greater than 0.5 and saving into a folder.  
resultcheck2<-ifelse(predict\_validation2>0.5,1,0)  
class(resultcheck2)

## [1] "numeric"

str(resultcheck)

## Named num [1:1333] 0 0 0 0 0 0 0 0 0 0 ...  
## - attr(\*, "names")= chr [1:1333] "1" "2" "10" "14" ...

#### 8.7 Accuracy check for both logistic regression models

##Logistic method  
#resultcheck!=validation\_data$churn  
error<-mean(resultcheck!=validation\_data$churn)  
accuracy<-1-error  
print(accuracy)

## [1] 0.8522131

#improvised model for logistic regression  
error2<-mean(resultcheck2!=validation\_data$churn)  
accuracy2<-1-error2  
print(accuracy2)

## [1] 0.8912228

Result: Accuracy of the improvised model using the step() function has better results with Accuracy around 90%.

# 8.8 ROC for logistic regression

library(pROC)

## 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

#ROC Curve for validation Data set with Logistic Model  
roc(validation\_data$churn, predict\_validation)

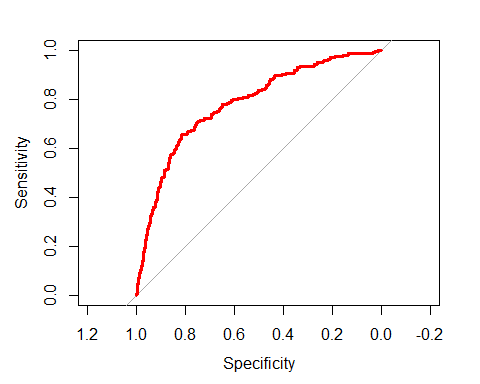
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = validation\_data$churn, predictor = predict\_validation)  
##   
## Data: predict\_validation in 1140 controls (validation\_data$churn 0) < 193 cases (validation\_data$churn 1).  
## Area under the curve: 0.7805

plot.roc(validation\_data$churn,predict\_validation,col = "red", lwd = 3)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases



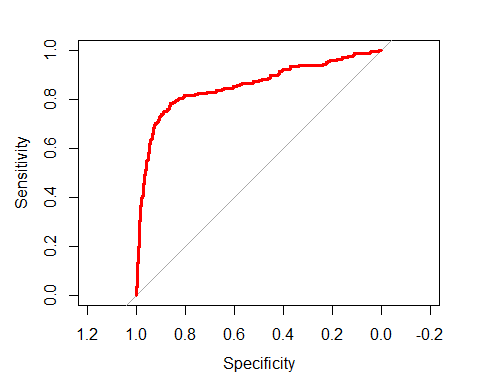
#ROC Curve for validation Data set with Improvised Logistic Model  
roc(validation\_data$churn, predict\_validation2)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = validation\_data$churn, predictor = predict\_validation2)  
##   
## Data: predict\_validation2 in 1140 controls (validation\_data$churn 0) < 193 cases (validation\_data$churn 1).  
## Area under the curve: 0.8546

plot.roc(validation\_data$churn,predict\_validation2,col = "red", lwd = 3)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases



#### 8.9 Let’s make a confusion matrix for the logistic regression performed above.

# Logistic Regression Confusion Matrix  
resultcheck<- as.factor(resultcheck)  
confusionMatrix(resultcheck,validation\_data$churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1109 166  
## 1 31 27  
##   
## Accuracy : 0.8522   
## 95% CI : (0.832, 0.8708)  
## No Information Rate : 0.8552   
## P-Value [Acc > NIR] : 0.6399   
##   
## Kappa : 0.1589   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9728   
## Specificity : 0.1399   
## Pos Pred Value : 0.8698   
## Neg Pred Value : 0.4655   
## Prevalence : 0.8552   
## Detection Rate : 0.8320   
## Detection Prevalence : 0.9565   
## Balanced Accuracy : 0.5564   
##   
## 'Positive' Class : 0   
##

#Improvised Logistic Regression Model Confusion Matrix  
resultcheck2<- as.factor(resultcheck2)  
confusionMatrix(resultcheck2, validation\_data$churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1110 115  
## 1 30 78  
##   
## Accuracy : 0.8912   
## 95% CI : (0.8733, 0.9074)  
## No Information Rate : 0.8552   
## P-Value [Acc > NIR] : 6.420e-05   
##   
## Kappa : 0.4624   
##   
## Mcnemar's Test P-Value : 3.041e-12   
##   
## Sensitivity : 0.9737   
## Specificity : 0.4041   
## Pos Pred Value : 0.9061   
## Neg Pred Value : 0.7222   
## Prevalence : 0.8552   
## Detection Rate : 0.8327   
## Detection Prevalence : 0.9190   
## Balanced Accuracy : 0.6889   
##   
## 'Positive' Class : 0   
##

# Anova  
anova(Logistic\_Model,Logistic\_Model2, test="Chisq")

## Analysis of Deviance Table  
##   
## Model 1: churn ~ account\_length + area\_code + international\_plan + voice\_mail\_plan +   
## number\_vmail\_messages + total\_day\_minutes + total\_day\_calls +   
## total\_day\_charge + total\_eve\_minutes + total\_eve\_calls +   
## total\_eve\_charge + total\_night\_minutes + total\_night\_calls +   
## total\_night\_charge + total\_intl\_minutes + total\_intl\_calls +   
## total\_intl\_charge + number\_customer\_service\_calls  
## Model 2: churn ~ account\_length + area\_code + international\_plan + voice\_mail\_plan +   
## number\_vmail\_messages + total\_day\_minutes + total\_day\_calls +   
## total\_day\_charge + total\_eve\_minutes + total\_eve\_charge +   
## total\_night\_minutes + total\_night\_charge + total\_intl\_minutes +   
## total\_intl\_calls + number\_customer\_service\_calls + total\_day\_charge:number\_customer\_service\_calls +   
## total\_day\_charge:total\_eve\_charge + voice\_mail\_plan:total\_day\_charge +   
## international\_plan:total\_intl\_minutes + international\_plan:number\_customer\_service\_calls +   
## total\_eve\_charge:number\_customer\_service\_calls + total\_day\_charge:total\_night\_charge +   
## international\_plan:total\_intl\_calls + area\_code:number\_vmail\_messages +   
## voice\_mail\_plan:total\_intl\_calls + total\_intl\_calls:number\_customer\_service\_calls +   
## total\_day\_calls:total\_eve\_charge + number\_vmail\_messages:total\_intl\_calls +   
## international\_plan:total\_day\_calls + voice\_mail\_plan:total\_night\_charge +   
## total\_night\_minutes:number\_customer\_service\_calls + total\_eve\_charge:total\_intl\_calls +   
## voice\_mail\_plan:total\_eve\_charge + total\_eve\_charge:total\_night\_minutes +   
## total\_day\_charge:total\_intl\_calls + area\_code:total\_day\_minutes +   
## international\_plan:total\_eve\_minutes + international\_plan:total\_day\_minutes +   
## international\_plan:total\_eve\_charge + total\_night\_minutes:total\_night\_charge  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 1980 1334.4   
## 2 1956 1055.7 24 278.66 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Observation:

1. Based on Anova model comparison and Confusion matrix results we can say that the performance has improved significantly by the improvised model.The Accuracy improved by 5 percent. Therefore, we will consider the improvised logistic regression model as the best logistic model with an accuracy of 90%
2. The improved Accuracy is good for the logistic model however , we are getting slightly better accuracy from the decision tree model(Accuracy 91%) when comparing the results.

Model Comparison result:-

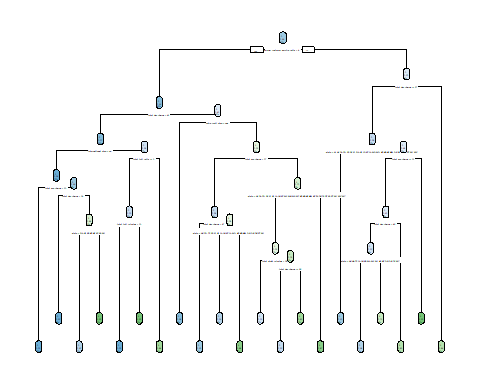
As per the targeted approach the company will be trying to identify in advance customers who are likely to churn. The company then targets those customers with special programs or incentives. Therefore Sensitivity is the top criteria for the model selection. Therefore, we would like to choose Decision trees as the best model to predict the customers who are likely to churn.The Decision Tree did a better job of predicting those who would churn (Specificity: 67%) and the Improvised Logistic Model had a specificity of 44%. Decision tree predicted 45 more people who churned than the improvised logistic model.

#### 9.Implementing Decision Tree model based on the above result:

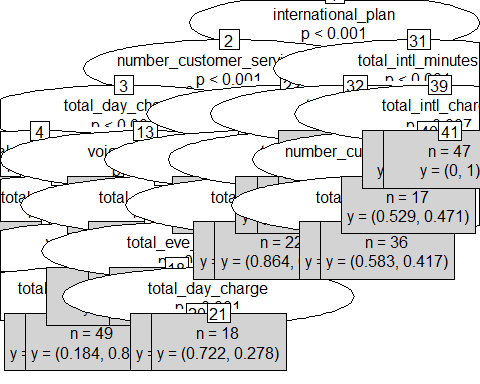
#Converting the data type Churn\_Train according to the Customers\_To\_Predict  
Churn\_Train[, c(2,6,8,11,14,17,19)] <- as.integer(unlist(Churn\_Train[, c(2,6,8,11,14,17,19)]))  
#Building the model on the Churn\_Train dataset using ctree()  
Model\_ABC\_Wireless <- ctree(Churn\_Train$churn~ ., Churn\_Train[,-1])  
## Predicting churn results based on the Decision Tree Model  
predict\_validation <- predict(Model\_ABC\_Wireless, newdata = Customers\_To\_Predict, type='response')  
table(predict\_validation)

## predict\_validation  
## 0 1   
## 920 80

predict\_validation <- as.data.frame(predict\_validation)  
#Plotting Decision Tree  
dcplot<-rpart(Churn\_Train$churn ~.,data=Churn\_Train,method='class')  
rpart.plot(dcplot,extra=106)



plot(Model\_ABC\_Wireless, type='simple')



#plotting the prediction results :  
predict\_validation %>%   
 ggplot(aes(x = `predict\_validation`)) +  
 geom\_histogram(stat = "count", fill = "orange") +  
 labs(x = "Customer Churn Or Not", y = "# of Customers")+  
 ggtitle(" Number of Customers likely to Churn") +  
 theme(plot.title = element\_text(hjust =.5, size = 16, face = c("bold", "italic")))

