Business Analytics-Group Project Group1

Khushboo Yadav,Mark Burner,Rakhee Moolchandani,Mayank Pugalia,Tanmoy Kanti Kumar

11/24/2020

```
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
```

1.Importing Libraries

```
#setwd("~/Documents/BusinessAnalyticsGroupProject/R code and Script")
#Loading the training data to analyze and build model
Churn_Train <- read_csv("Churn_Train(1).csv")
```

2. Loading and Reading the Dataset

```
## Parsed with 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()
## )
```

```
## See spec(...) for full column specifications.
```

```
#Loading the file containing the list of consumers that we need to predict their future churn
load("Customers_To_Predict(1).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))
```

```
summary(Churn_Train)
```

3. Analysing Count of the NA Values in the Dataset

```
##     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
```

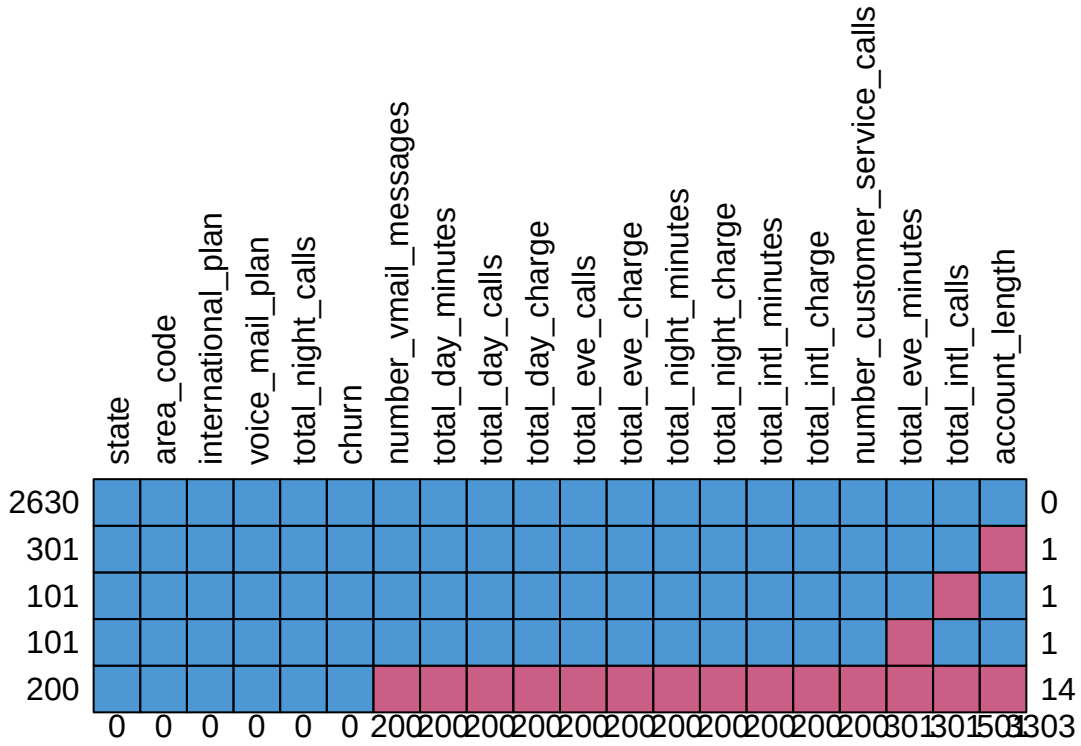
```
##                         state                account_length
##                             0                           501
```

```
##                     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
```

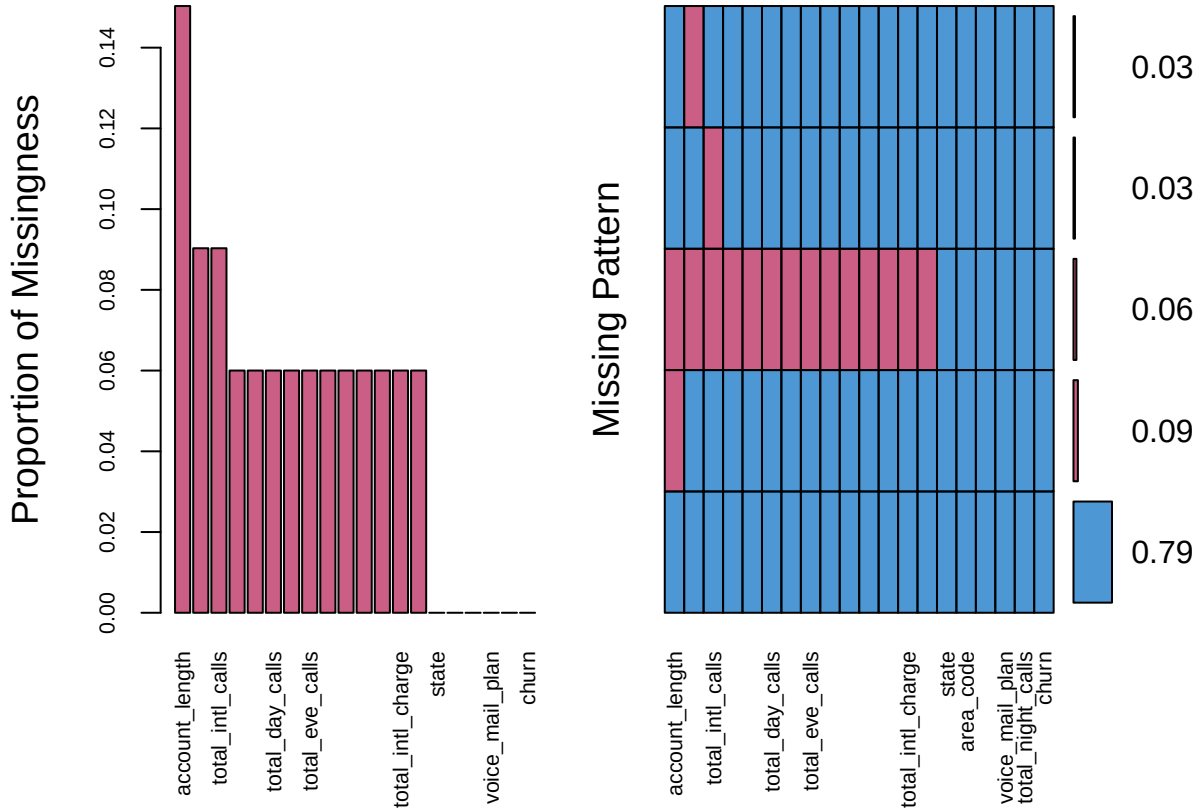
```
md.pattern(Churn_Train,rotate.names = TRUE) # See the missing value pattern
```




```
##      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
```




```
##
##  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
```

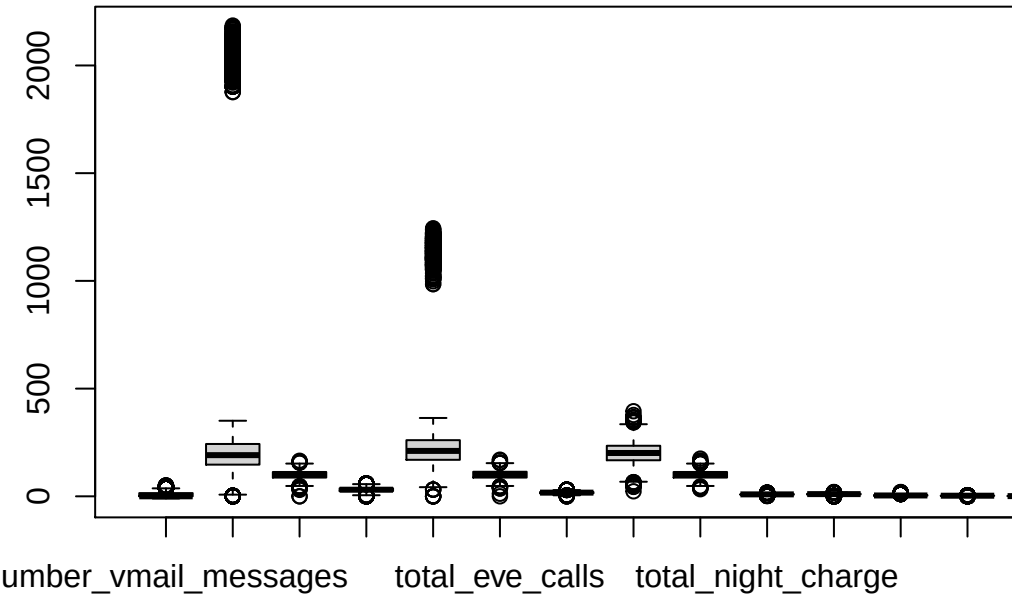
As we can see that the current dataset Churn_Train has lots of NA values .Removing them from the dataset will be lead to the loss of useful information and can imapct data analysis and model predictions.

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
```

```
boxplot(Churn_Train[, 6:19])
```



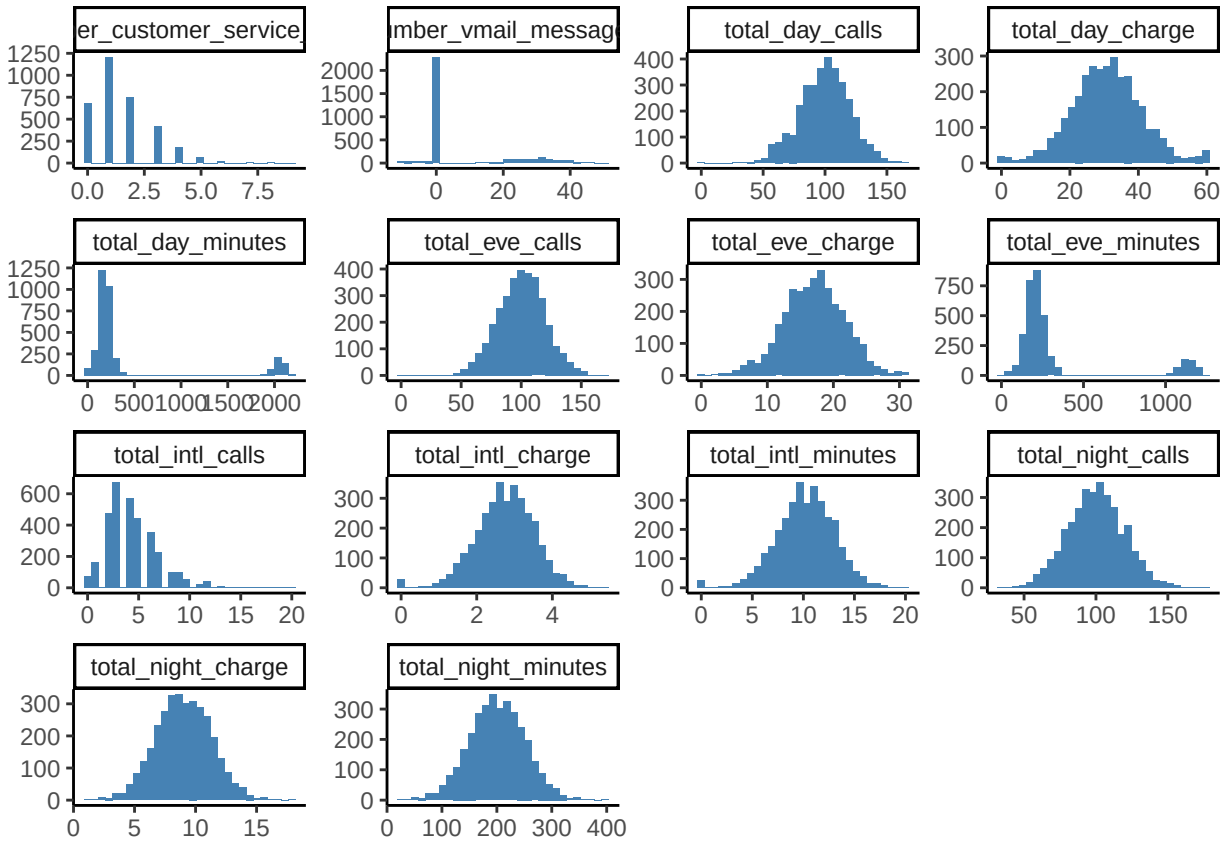
5. Exploratory Data Analysis

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 .

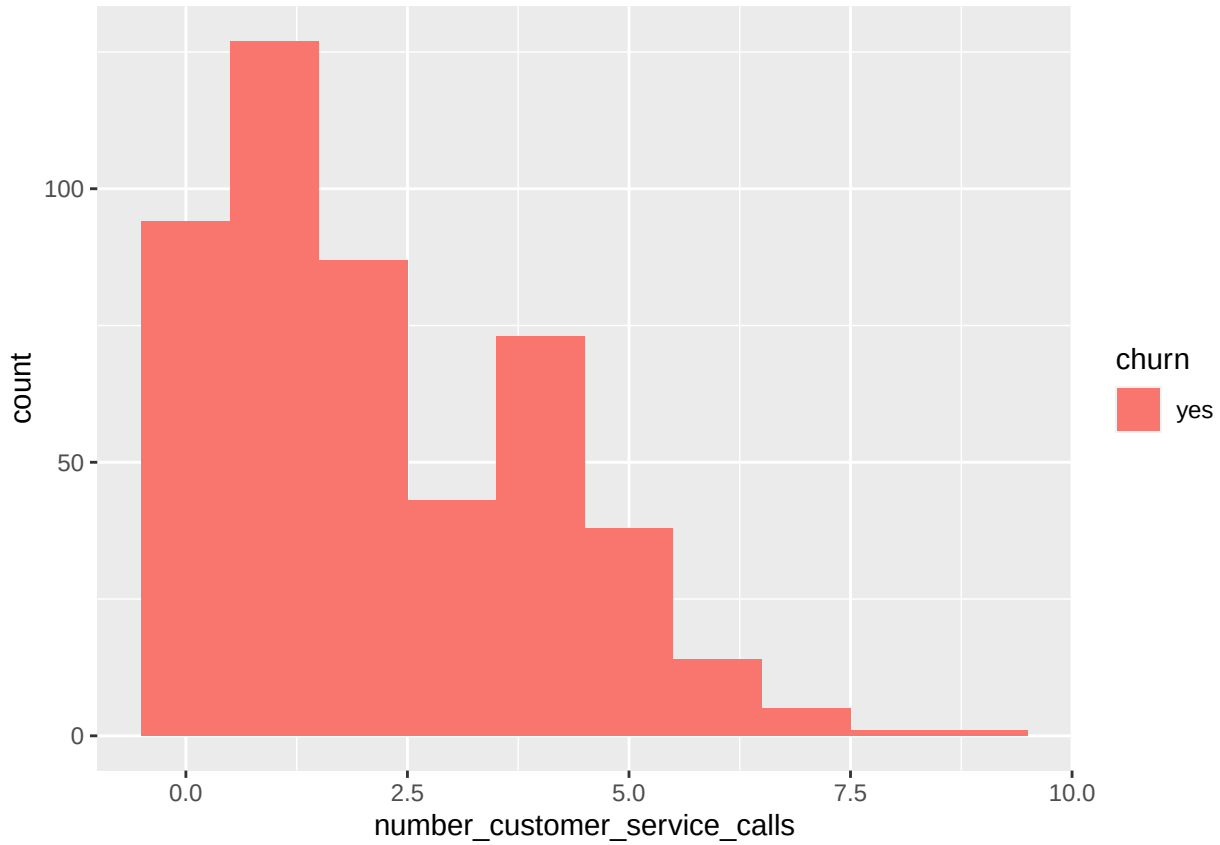
```
Churn_Train[, 6:19] %>%
  gather(key = Variable, value = Value) %>%
  ggplot() +
    geom_histogram(aes(x = Value), fill = "steelblue") +
    facet_wrap(~Variable, scales='free') +
    theme_classic() +
    theme(aspect.ratio = 0.5, axis.title = element_blank(), panel.grid = element_blank())
```



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 ch
```




```
Churn_Train %>%
  group_by(churn) %>%
  tally(churn == "yes") # total churned in data set.
```

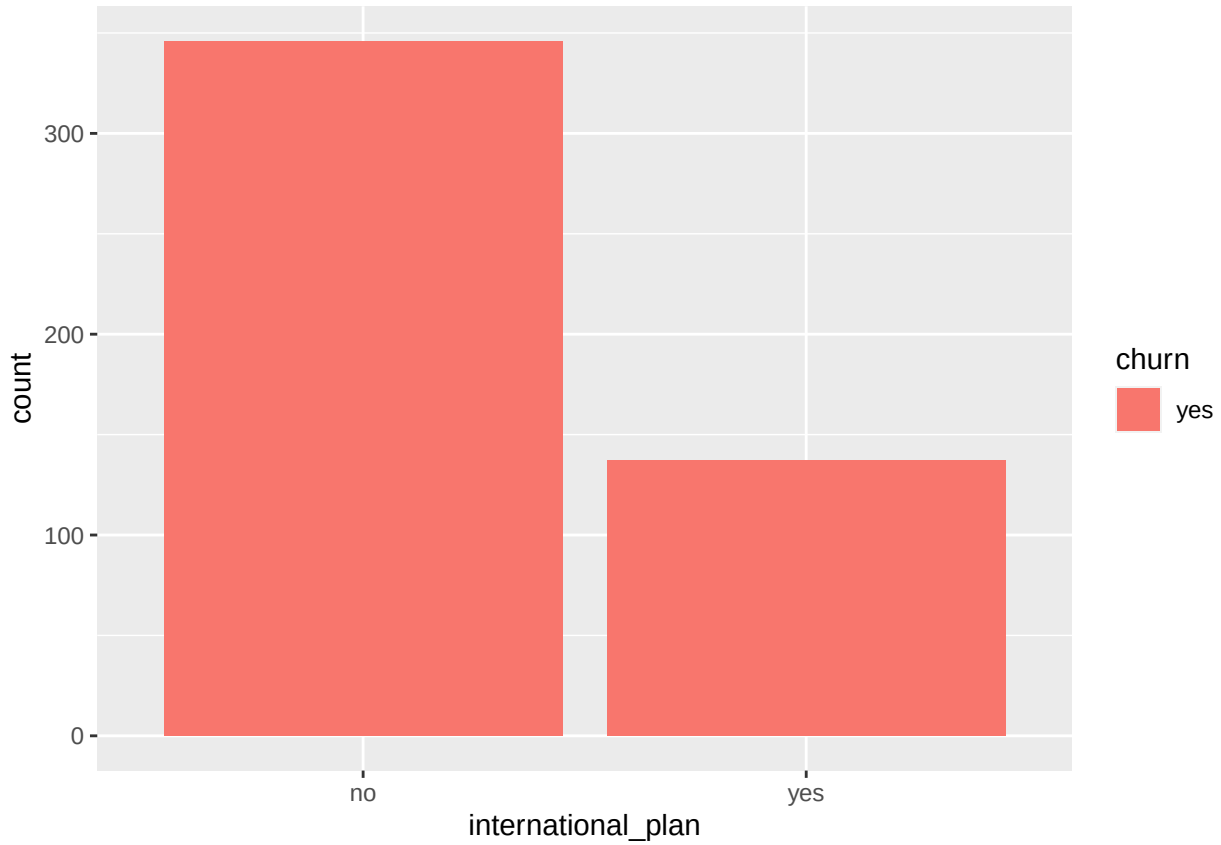
```
## # A tibble: 2 x 2
##   churn     n
##   <chr> <int>
## 1 no        0
## 2 yes     483
```

```
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 0.6832298
```

```
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
```

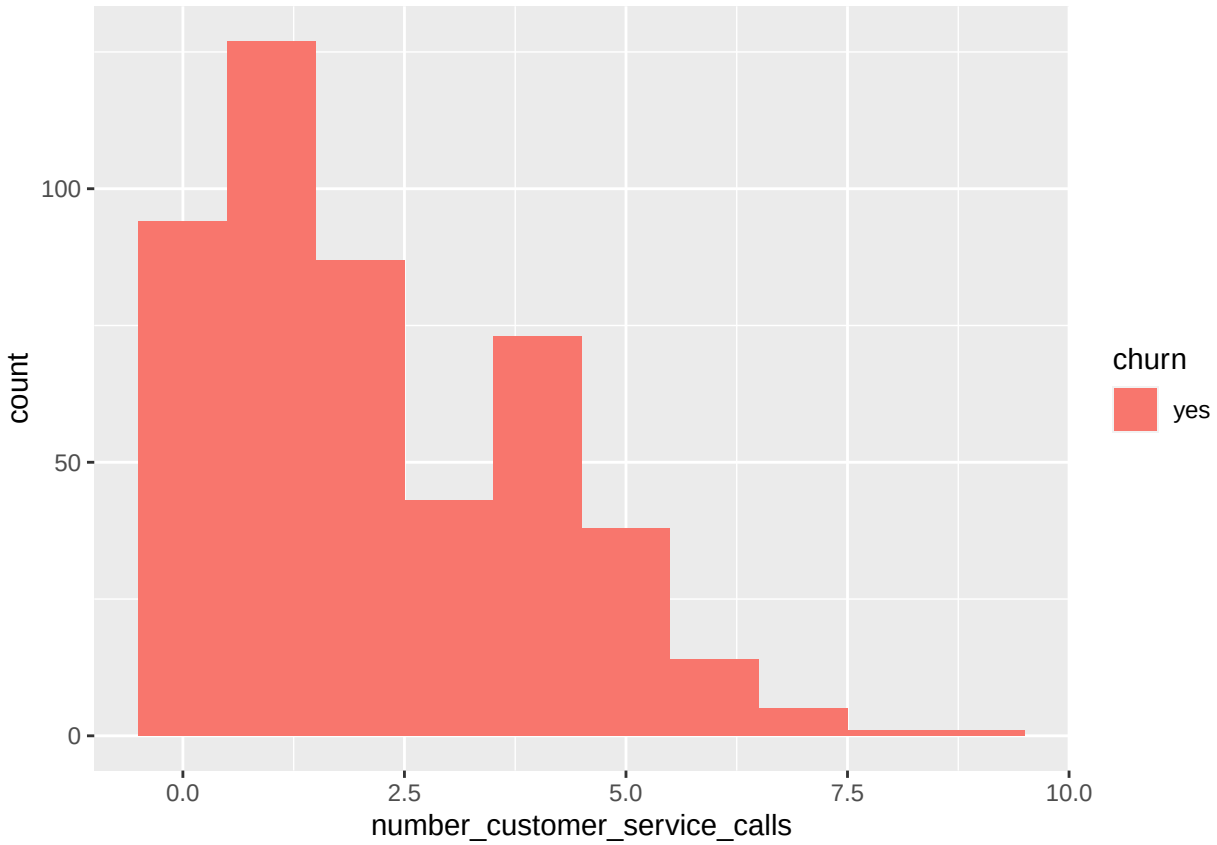



```
Churn_Train %>%
  group_by(international_plan) %>%
  filter(churn == "yes") %>%
  select(international_plan) %>%
  dplyr:: summarise("Churn Count" =n(), "Percent" = n()/483)
```

```
## `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)
```

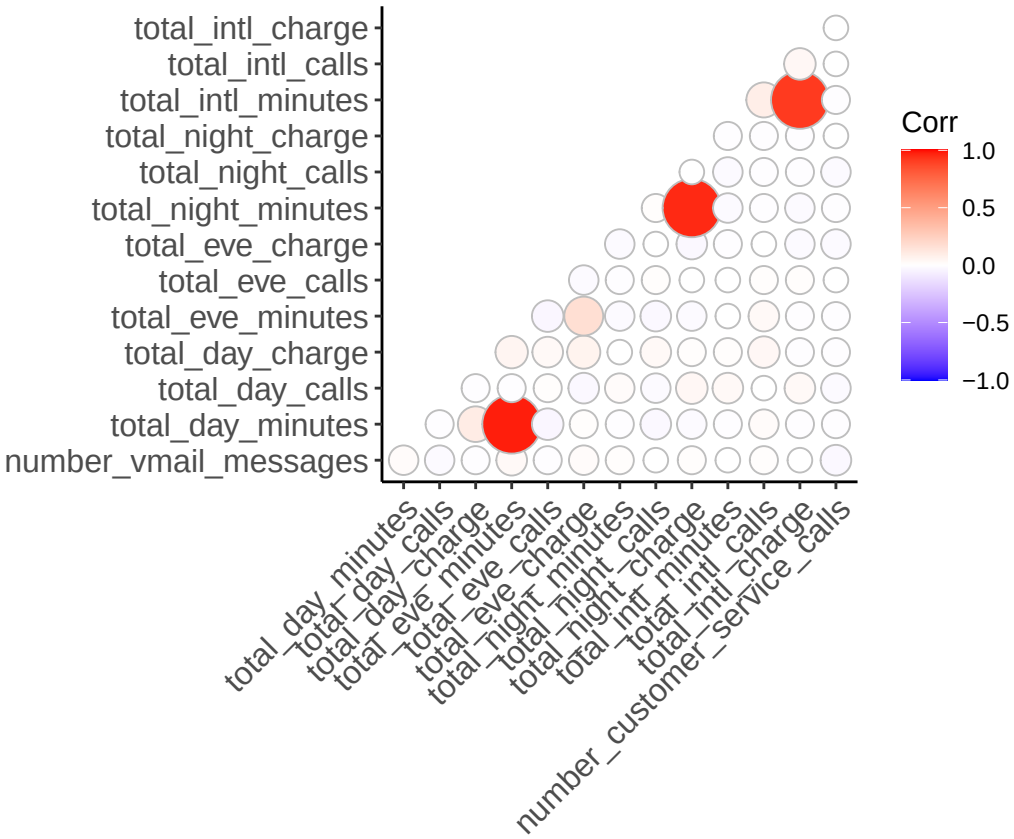


5.2 Correlation between variables of Train_Churn and dataset.

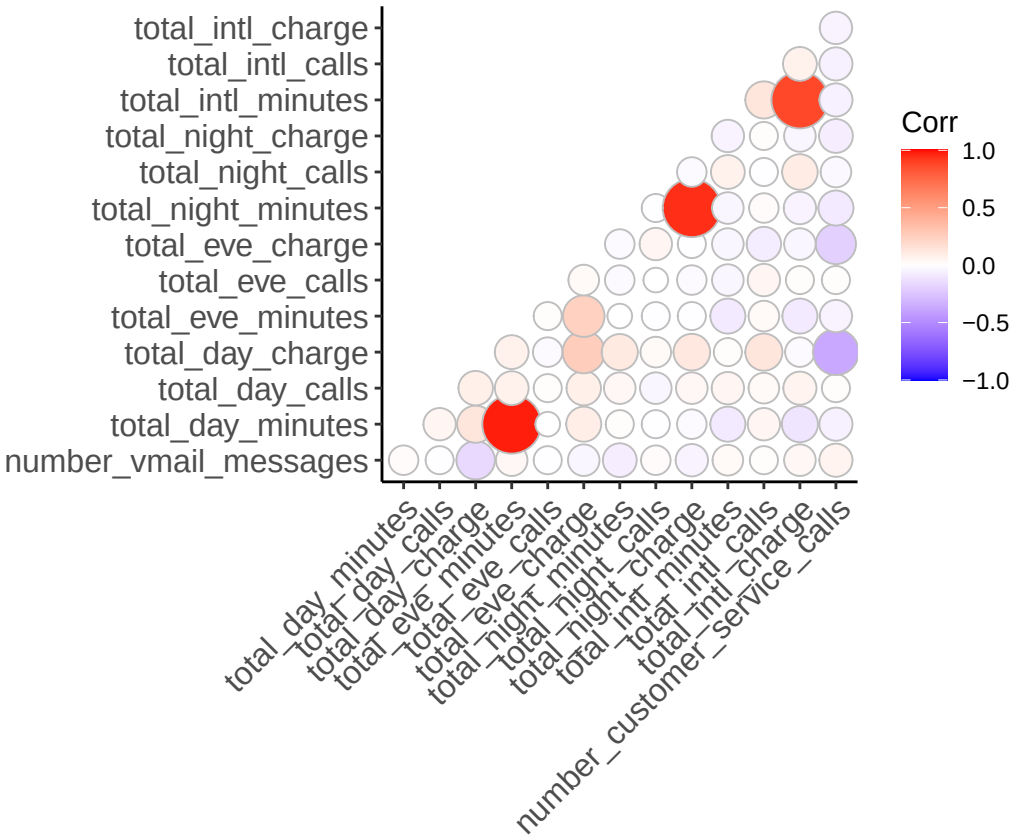
Here, we will analyze the correlation of variables with the following condition:

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

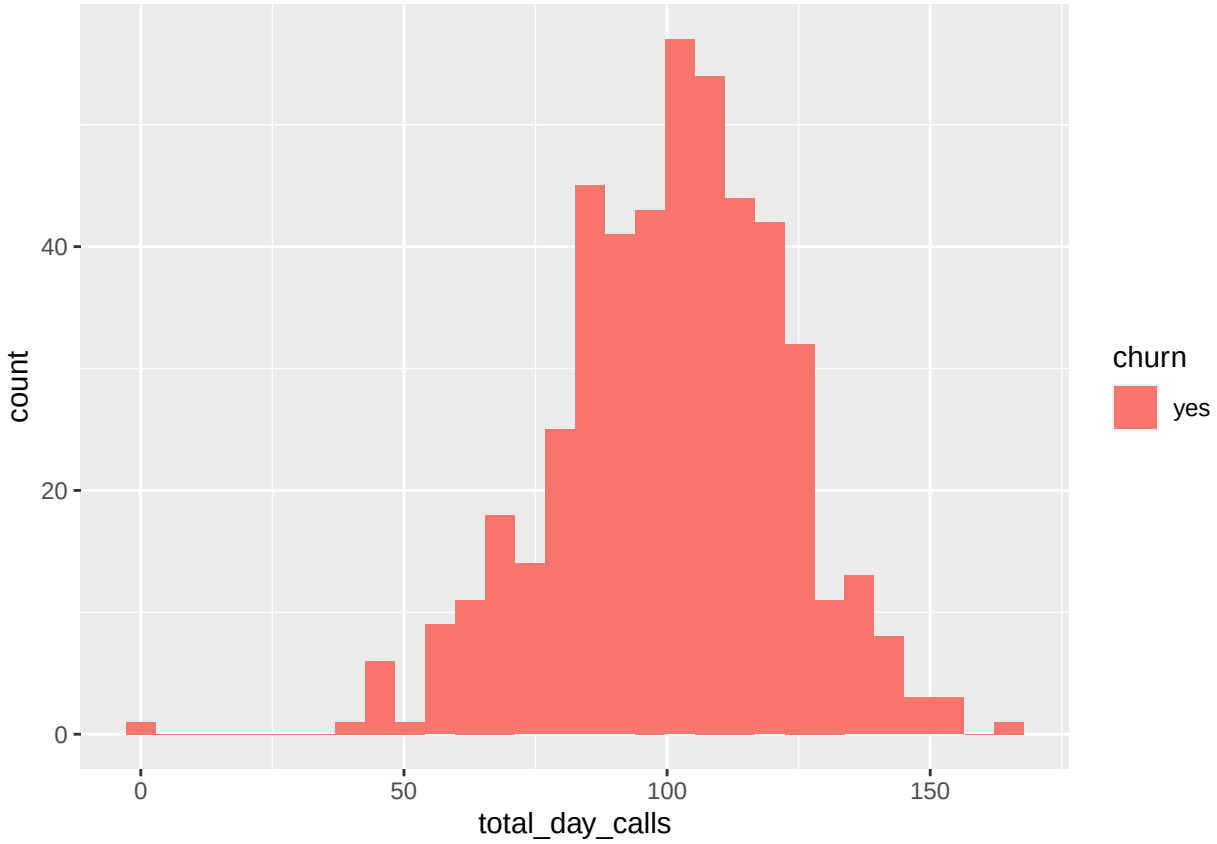
Strong negative correlation between total day minutes and total evening minutes. Meaning that the as the evening minutes increase the total day minutes decrease. Also, a slight negative correlation between the total evening minutes and the total evening charges.

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.

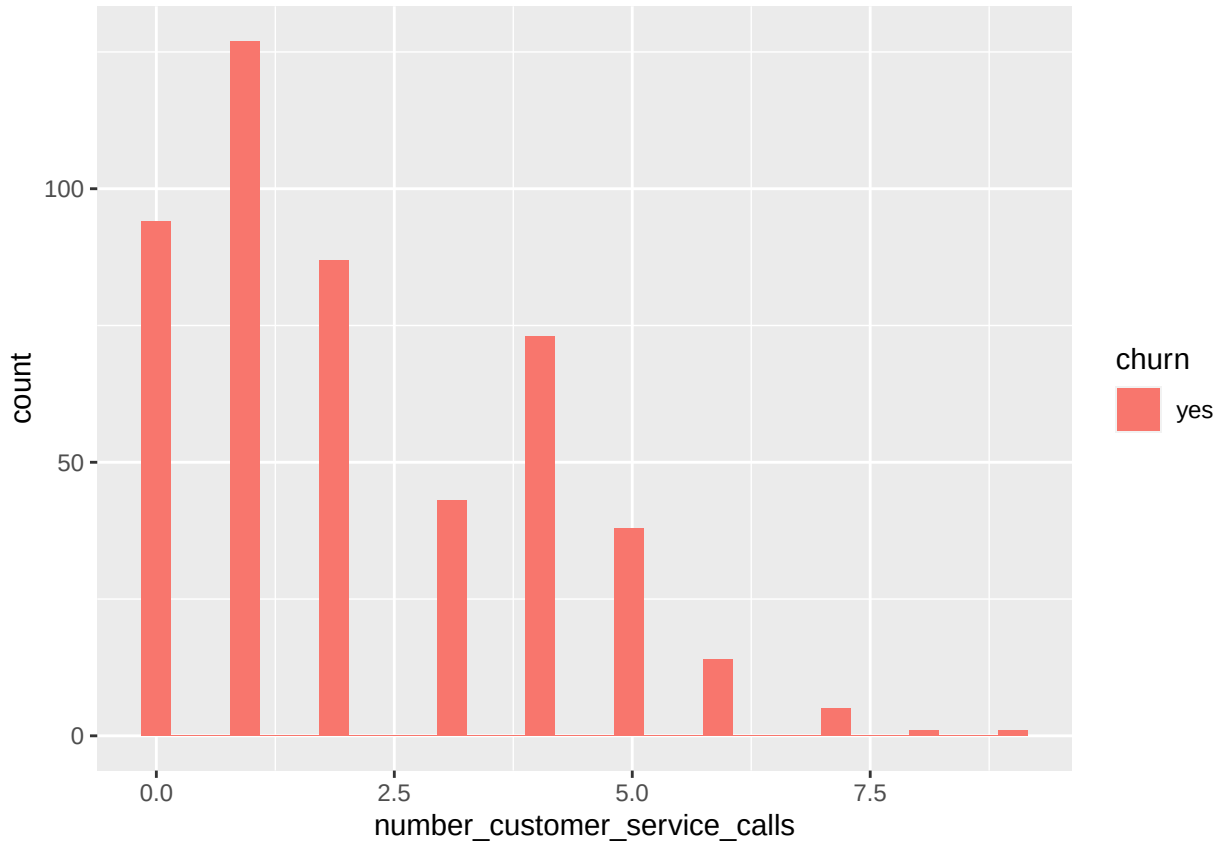
```
Churn_Train %>%
  filter(churn=="yes") %>%
  ggplot(mapping = aes(x = total_day_calls)) +
  geom_histogram(aes(fill = churn))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



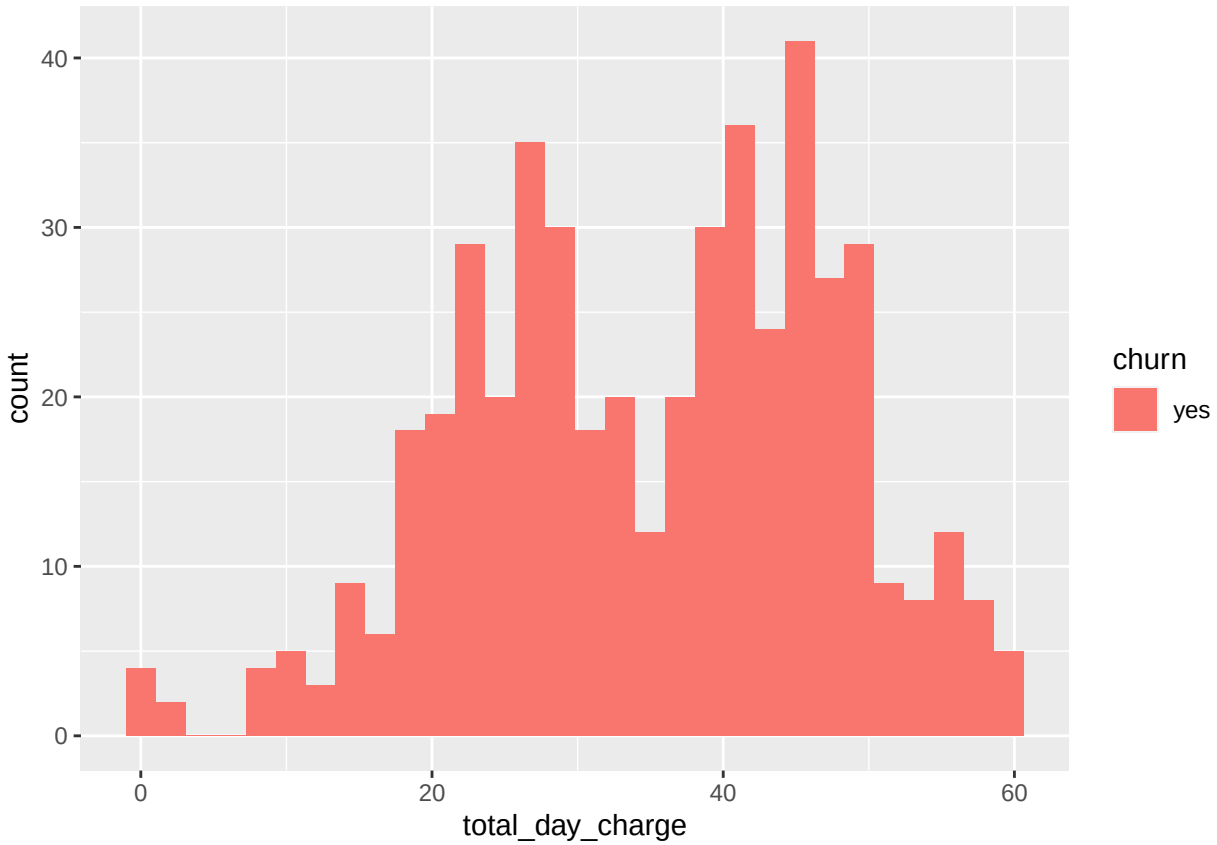

```
Churn_Train %>%
  filter(churn=="yes") %>%
  ggplot(mapping = aes(x = number_customer_service_calls)) +
  geom_histogram(aes(fill = churn))
```

```
## `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 1
##  $ account_length                : num  125 108 82 13 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.

```
set.seed(2020)
partition<- createDataPartition(Churn_Train$churn,p=0.6,list=FALSE)
train_data<- Churn_Train[partition,]
validation_data<- Churn_Train[-partition,]
```

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

```
# Decision Tree
DecisionTree_model <- ctree(churn~ ., train_data[,-1]) #not including state column
pred_tree <- predict(DecisionTree_model, validation_data)
#table
table(pred_tree)
```

8.2 Building Decision tree model:

```
## pred_tree
##    0    1
## 1168  165
```

```
#confusion matrix
confusionMatrix(pred_tree,validation_data$churn) ## without states
```

8.3 Confusion matrix for decision trees

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1095   73
##          1   45  120
##
##                Accuracy : 0.9115
##                  95% CI : (0.8949, 0.9262)
##     No Information Rate : 0.8552
##     P-Value [Acc > NIR] : 3.472e-10
##
##                   Kappa : 0.6196
##
##  Mcnemar's Test P-Value : 0.01294
##
##             Sensitivity : 0.9605
##             Specificity : 0.6218
##          Pos Pred Value : 0.9375
##          Neg Pred Value : 0.7273
##              Prevalence : 0.8552
##          Detection Rate : 0.8215
##    Detection Prevalence : 0.8762
##       Balanced Accuracy : 0.7911
```

```
##
##        'Positive' Class : 0
##
```

```
#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)
```

8.4 Building Logistic Regression Model:

```
##
## Call:
## glm(formula = churn ~ ., family = binomial(link = "logit"), data = train_data[,
##     -1])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9794  -0.5155  -0.3507  -0.2150   3.0227
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -7.2525012  0.9136157  -7.938 2.05e-15 ***
## account_length               -0.0008949  0.0014701  -0.609   0.5427
## area_code415                 -0.0517050  0.1739901  -0.297   0.7663
## area_code510                 -0.1668663  0.2016928  -0.827   0.4081
## international_planyes         1.9966920  0.1871902  10.667  < 2e-16 ***
## voice_mail_planyes           -0.8586488  0.3420295  -2.510   0.0121 *
## number_vmail_messages        -0.0077598  0.0113273  -0.685   0.4933
## total_day_minutes             0.0007958  0.0007295   1.091   0.2753
## total_day_calls               0.0029570  0.0035693   0.828   0.4074
## total_day_charge              0.0579611  0.0079365   7.303 2.81e-13 ***
## total_eve_minutes            -0.0018481  0.0014618  -1.264   0.2061
## total_eve_calls              -0.0011089  0.0035549  -0.312   0.7551
## total_eve_charge              0.0956540  0.0230616   4.148 3.36e-05 ***
## total_night_minutes           0.0058491  0.0041110   1.423   0.1548
## total_night_calls            -0.0029798  0.0036622  -0.814   0.4158
## total_night_charge           -0.0478802  0.0914668  -0.523   0.6006
## total_intl_minutes            0.0602480  0.0622979   0.967   0.3335
## total_intl_calls             -0.0629020  0.0296312  -2.123   0.0338 *
## total_intl_charge             0.1476630  0.2253042   0.655   0.5122
## number_customer_service_calls  0.4561793  0.0503195   9.066  < 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: 1317.8  on 1980  degrees of freedom
## AIC: 1357.8
##
## 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)
```

```
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)
```

8.5 Building Improvised Logistic Regression model

```
##
## 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
```

```
## -2.5539  -0.4294  -0.2369  -0.1136   3.5846
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                      -2.0976689  3.2567305
## account_length                                    0.0007098  0.0017248
## area_code415                                      0.0968256  0.2510940
## area_code510                                      0.1498897  0.2933005
## international_planyes                             4.8905986  1.9352545
## voice_mail_planyes                                5.9088539  1.4684425
## number_vmail_messages                            -0.0025109  0.0263911
## total_day_minutes                                 0.0018420  0.0009035
## total_day_calls                                  -0.0443088  0.0170930
## total_day_charge                                 -0.0358926  0.0505723
## total_eve_minutes                                -0.0037846  0.0017963
## total_eve_charge                                 -0.4669646  0.1547612
## total_night_minutes                              -0.0038397  0.0104344
## total_night_charge                               -0.3473672  0.2063488
## total_intl_minutes                                0.0432297  0.0321745
## total_intl_calls                                  0.2403886  0.1670354
## number_customer_service_calls                     3.5363914  0.4088224
## total_day_charge:number_customer_service_calls   -0.0533452  0.0060323
## total_day_charge:total_eve_charge                 0.0088934  0.0017689
## voice_mail_planyes:total_day_charge              -0.1318402  0.0232540
## international_planyes:total_intl_minutes          0.4970203  0.1043479
## international_planyes:number_customer_service_calls -0.3483093  0.1599609
## total_eve_charge:number_customer_service_calls   -0.0352054  0.0139725
## total_day_charge:total_night_charge               0.0089131  0.0039260
## international_planyes:total_intl_calls           -0.4121895  0.0997139
## area_code415:number_vmail_messages               -0.0122988  0.0175401
## area_code510:number_vmail_messages               -0.0011834  0.0200024
## voice_mail_planyes:total_intl_calls              -0.0054272  0.1333141
## total_intl_calls:number_customer_service_calls   -0.0741319  0.0239401
## total_day_calls:total_eve_charge                  0.0030594  0.0009369
## number_vmail_messages:total_intl_calls            0.0007663  0.0044330
## international_planyes:total_day_calls            -0.0279604  0.0113750
## voice_mail_planyes:total_night_charge            -0.1144084  0.0943840
## total_night_minutes:number_customer_service_calls -0.0012977  0.0011752
## total_eve_charge:total_intl_calls                -0.0087705  0.0068818
## voice_mail_planyes:total_eve_charge              -0.0910944  0.0534331
## total_eve_charge:total_night_minutes              0.0006662  0.0004018
## total_day_charge:total_intl_calls                 0.0018077  0.0031946
## area_code415:total_day_minutes                   -0.0001422  0.0003321
## area_code510:total_day_minutes                   -0.0007052  0.0004216
## international_planyes:total_eve_minutes           0.0271976  0.0054349
## international_planyes:total_day_minutes          -0.0129007  0.0027050
## international_planyes:total_eve_charge           -0.3342971  0.0734806
## total_night_minutes:total_night_charge            0.0001404  0.0005278
##                                                  z value Pr(>|z|)
## (Intercept)                                       -0.644  0.51951
## account_length                                     0.412  0.68067
## area_code415                                       0.386  0.69978
## area_code510                                       0.511  0.60932
## international_planyes                              2.527  0.01150 *
```

```
## voice_mail_planyes                                      4.024 5.72e-05 ***
## number_vmail_messages                                  -0.095  0.92420
## total_day_minutes                                       2.039  0.04149 *
## total_day_calls                                        -2.592  0.00954 **
## total_day_charge                                       -0.710  0.47787
## total_eve_minutes                                      -2.107  0.03513 *
## total_eve_charge                                       -3.017  0.00255 **
## total_night_minutes                                    -0.368  0.71288
## total_night_charge                                     -1.683  0.09230 .
## total_intl_minutes                                      1.344  0.17908
## total_intl_calls                                        1.439  0.15011
## number_customer_service_calls                           8.650  < 2e-16 ***
## total_day_charge:number_customer_service_calls         -8.843  < 2e-16 ***
## total_day_charge:total_eve_charge                       5.028 4.96e-07 ***
## voice_mail_planyes:total_day_charge                    -5.670 1.43e-08 ***
## international_planyes:total_intl_minutes                4.763 1.91e-06 ***
## international_planyes:number_customer_service_calls    -2.177  0.02945 *
## total_eve_charge:number_customer_service_calls         -2.520  0.01175 *
## total_day_charge:total_night_charge                     2.270  0.02319 *
## international_planyes:total_intl_calls                 -4.134 3.57e-05 ***
## area_code415:number_vmail_messages                     -0.701  0.48319
## area_code510:number_vmail_messages                     -0.059  0.95282
## voice_mail_planyes:total_intl_calls                    -0.041  0.96753
## total_intl_calls:number_customer_service_calls         -3.097  0.00196 **
## total_day_calls:total_eve_charge                        3.266  0.00109 **
## number_vmail_messages:total_intl_calls                  0.173  0.86276
## international_planyes:total_day_calls                  -2.458  0.01397 *
## voice_mail_planyes:total_night_charge                  -1.212  0.22545
## total_night_minutes:number_customer_service_calls      -1.104  0.26951
## total_eve_charge:total_intl_calls                      -1.274  0.20250
## voice_mail_planyes:total_eve_charge                    -1.705  0.08823 .
## total_eve_charge:total_night_minutes                    1.658  0.09731 .
## total_day_charge:total_intl_calls                       0.566  0.57148
## area_code415:total_day_minutes                         -0.428  0.66855
## area_code510:total_day_minutes                         -1.673  0.09439 .
## international_planyes:total_eve_minutes                 5.004 5.61e-07 ***
## international_planyes:total_day_minutes                -4.769 1.85e-06 ***
## international_planyes:total_eve_charge                 -4.549 5.38e-06 ***
## total_night_minutes:total_night_charge                  0.266  0.79021
## ---
## 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: 1036.1  on 1956  degrees of freedom
## AIC: 1124.1
##
## Number of Fisher Scoring iterations: 6
```

```r
#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)
```

```
##Logistic method
error<-mean(resultcheck!=validation_data$churn)
accuracy<-1-error
print(accuracy)
```

8.7 Accuracy check for both logistic regression models

```
## [1] 0.8514629
```

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

```
## [1] 0.9002251
```

Result: Accuracy of the improvised model using the step() function has better results with Accuracy = 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)
```

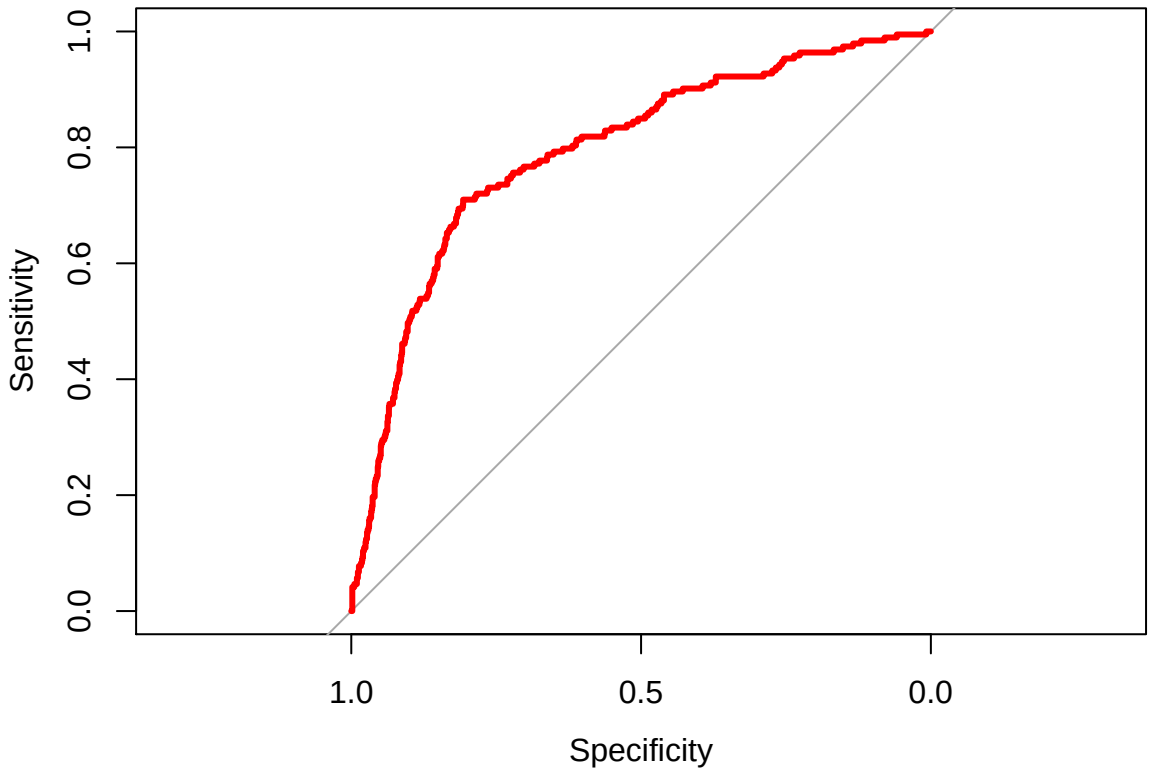
```
## Setting levels: control = 0, case = 1
```

```
## 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$chu
## Area under the curve: 0.7927
```

```
plot.roc(validation_data$churn,predict_validation,col = "red", lwd = 3)
```

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



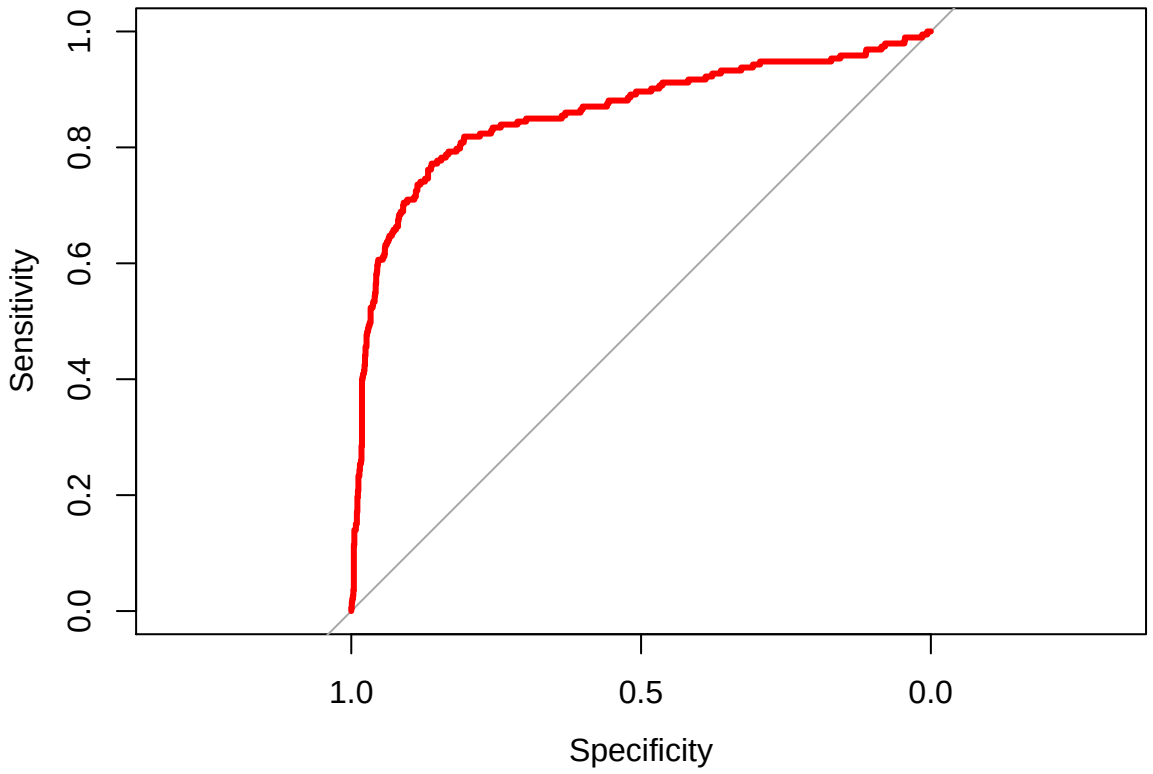

```
#ROC Curve for validation Data set with Improvised Logistic Model
roc(validation_data$churn, predict_validation2)
```

```
## 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$chu
## Area under the curve: 0.8579
```

```
plot.roc(validation_data$churn,predict_validation2,col = "red", lwd = 3)
```

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




```
# Logistic Regression Confusion Matrix
resultcheck<- as.factor(resultcheck)
confusionMatrix(resultcheck,validation_data$churn)
```

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1105  163
##          1   35   30
##
##                Accuracy : 0.8515
##                  95% CI : (0.8312, 0.8701)
##     No Information Rate : 0.8552
##     P-Value [Acc > NIR] : 0.6684
##
##                   Kappa : 0.1722
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9693
##             Specificity : 0.1554
##          Pos Pred Value : 0.8715
##          Neg Pred Value : 0.4615
##              Prevalence : 0.8552
##          Detection Rate : 0.8290
```

```
##    Detection Prevalence : 0.9512
##       Balanced Accuracy : 0.5624
##
##        '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  103
##          1   30   90
##
##                Accuracy : 0.9002
##                  95% CI : (0.8829, 0.9158)
##     No Information Rate : 0.8552
##     P-Value [Acc > NIR] : 6.110e-07
##
##                   Kappa : 0.522
##
##  Mcnemar's Test P-Value : 4.287e-10
##
##             Sensitivity : 0.9737
##             Specificity : 0.4663
##          Pos Pred Value : 0.9151
##          Neg Pred Value : 0.7500
##              Prevalence : 0.8552
##          Detection Rate : 0.8327
##    Detection Prevalence : 0.9100
##       Balanced Accuracy : 0.7200
##
##        '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     1317.8
## 2      1956     1036.1 24   281.77 < 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:-

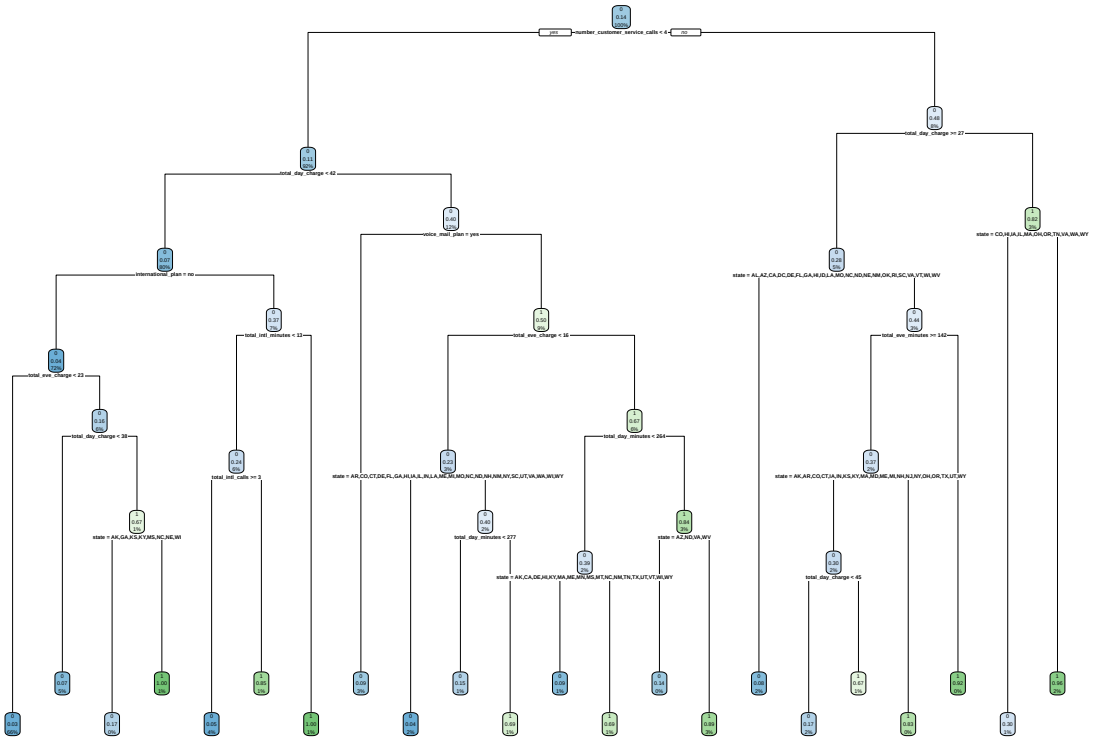
As per the targeted approach the company will be trying to identify in advance customers who are likely to churn. The company then targets those customers with special programs or incentives. Therefore Sensitivity is the top criteria for the model selection. Therefore, we would like to choose Decision 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.

```
#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)
```

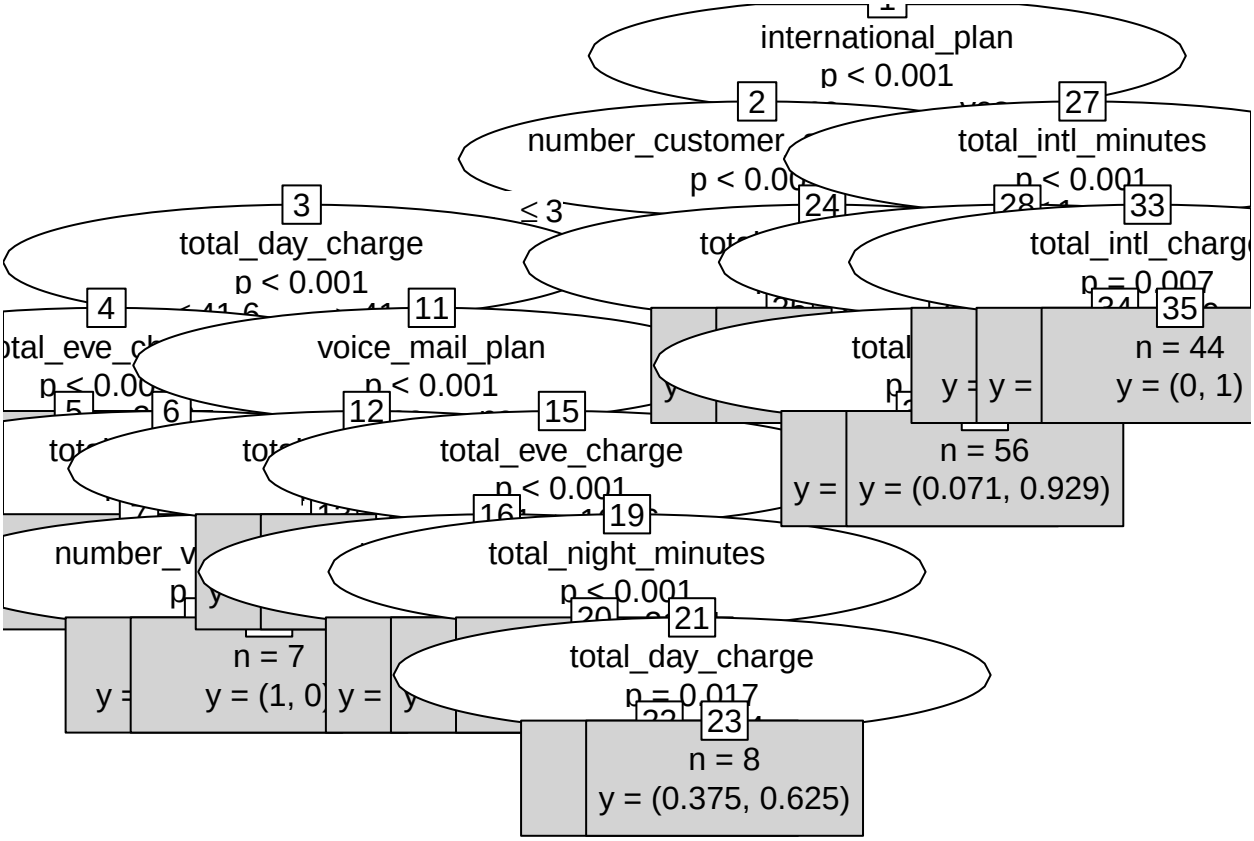
9.Implementing Decision Tree model based on the above result:

```
## predict_validation
##   0   1
## 923  77
```

```
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")))
```