Churn Rate Analysis

Namrah and Dutt

2023-04-29

#1.Project Objective

ABC wireless INC is a telecom provider. The purpose of this project is to help address their customer churn rate issue. With the help of the company's historical data, we aim to predict or identify customers who are likely to churn. Churn is basically the loss of customers to the competitor. This is a serious issue for telecom companies where the competition is cut-throat. Retaining a customer is less expensive than acquiring a new one. The task of our team is to apply analytics and help the management take appropriate decisions to reduce their churn rate and increase client retention.

#2.Packages required for current project.

```
library("dplyr")
library("magrittr")
library("randomForestExplainer")
library("ggplot2")
library("tidyverse")
library("randomForest")
library("usmap")
library("ggplot2")
library("ggcorrplot")
library("dlookr")
library("corrplot")
library("caret")
```

#3.Importing the dataset.

```
churndata_df <- read.csv("/Users/duttthakkar/Desktop/Churn_Train.csv")
summary(churndata_df)
```

```
##    state           account_length     area_code         international_plan
##  Length:3333        Min.   :-209.00   Length:3333        Length:3333
##  Class :character   1st Qu.:  72.00   Class :character   Class :character
##  Mode  :character   Median : 100.00   Mode  :character   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
##
##
##
##
```

#From above we can observe that, The following observations show significant NA values: #account_length

#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_charge #total_intl_minutes #total_intl_calls #total_intl_charge #number_customer_service_calls

#5.Negative value observation

```
churndata_df %>%
  select(account_length, number_vmail_messages) %>%
  summary()
```

```
##  account_length    number_vmail_messages
##  Min.   :-209.00   Min.   :-10.000
##  1st Qu.:  72.00   1st Qu.:  0.000
##  Median : 100.00   Median :  0.000
##  Mean   :  97.32   Mean   :  7.333
##  3rd Qu.: 127.00   3rd Qu.: 16.000
##  Max.   : 243.00   Max.   : 51.000
##  NA's   :501       NA's   :200
```

#account_length has a range of values between -209 and 243. In this data set's domain, 'account_length' denotes the number of months a customer has had an account assuming the account length is in months. As a result, any negative values in 'account_length' should be avoided.

#number_vmail_messages shows the number of voice mail messages a customer has had, this number can not be in the negative. In the dataset 'number_vmail_messages' has values ranging from -10 to 51. Thus any negative value should be avoided.

Other Missing Values in data

#NA refers to missing values. 16 out of of the 20 variables (columns) have NA values. It can be observed that 13 variables have about 200 'NA' values while 2 have 301 and 1 has 501.

#6.filtering and subsetting to compute the percentage of NAs accross all columns.

```
na_percent <- function(df, fmt = F) {
  return (df %>%
            is.na() %>%
            colMeans() %>%
            sapply(function(x) {
              if (fmt) {
                return(sprintf("%.5f%%", x * 100))
              }
              return (x)
            })
          )
}
na_percent_df <- na_percent(churndata_df) %>%
  data_frame(Columns = names(.), `NA %` = .) %>%
  mutate_at(
    vars(`NA %`),
    funs(round(. * 100, 2))
  ) %>%
  mutate(label = sprintf("%g%%", `NA %`)) %>%
  arrange(desc(`NA %`))
na_percent_df %>% select(-label)
```

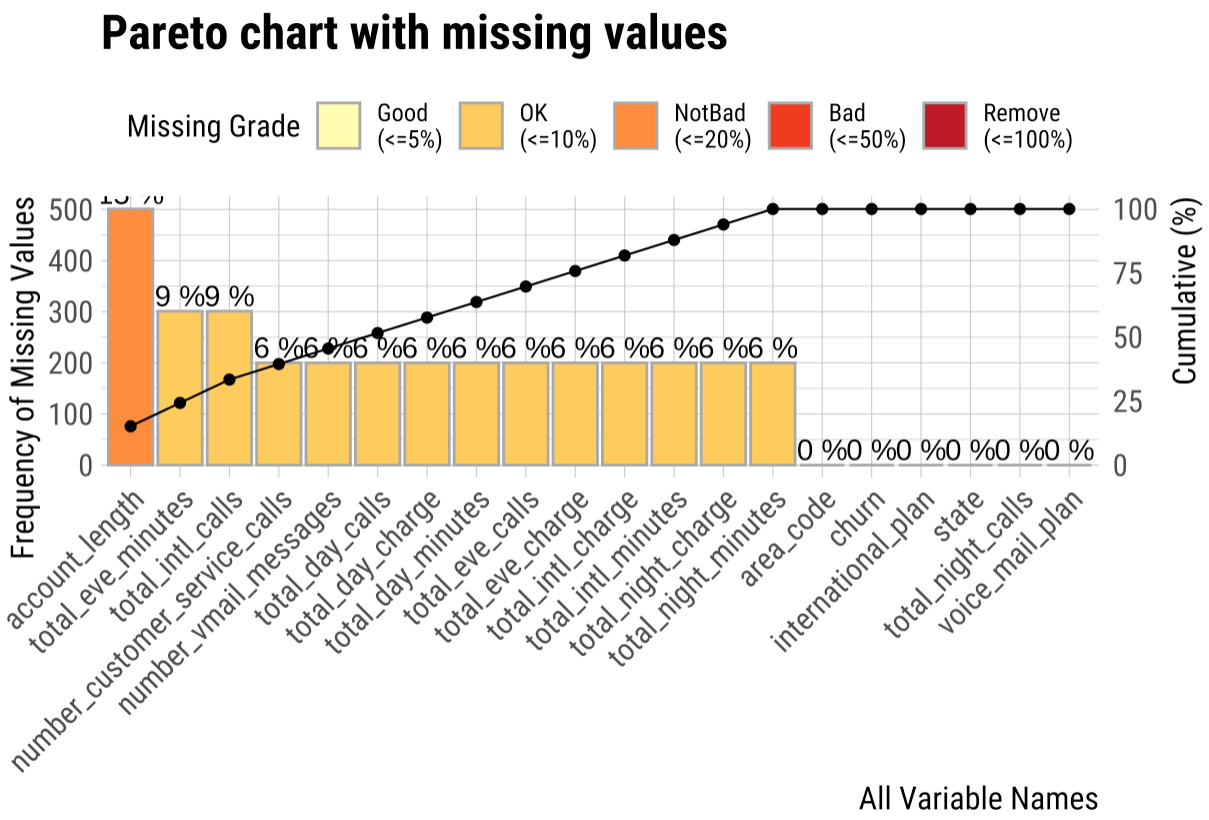
```
## # A tibble: 20 × 2
##    Columns                       `NA %`
##    <chr>                          <dbl>
##  1 account_length                 15.0
##  2 total_eve_minutes               9.03
##  3 total_intl_calls                9.03
##  4 number_vmail_messages           6
##  5 total_day_minutes               6
##  6 total_day_calls                 6
##  7 total_day_charge                6
##  8 total_eve_calls                 6
##  9 total_eve_charge                6
## 10 total_night_minutes             6
## 11 total_night_charge              6
## 12 total_intl_minutes              6
## 13 total_intl_charge               6
## 14 number_customer_service_calls   6
## 15 state                           0
## 16 area_code                       0
## 17 international_plan              0
## 18 voice_mail_plan                 0
## 19 total_night_calls               0
## 20 churn                           0
```

#The categorical variables such as state, area code, international plan, and total night calls (numerical variable) have no ‘NA’ values in the table above.

#The below bar chart gives a good graphical representation of the same.The major contribution of ‘NA’ comes

from 'account_length`, total_intl_calls and total_intl_minutes.

```
plot_na_pareto(churndata_df)
```



we further observe that 11 variables have an NA percentage of 6%. Below table shows only the variables that have NA in them. The code chunk removes columns that have an NA percentage of 0% and then only shows rows that have at least 1 NA value in them.

#7.Data Cleaning: #Turning Negatives into Positives.To deal with those variables that have negative values in them, we use abs function.

```
churndata_df <- churndata_df %>%
  mutate_at(.vars = vars(number_vmail_messages), .funs = funs(abs))
summary(churndata_df)
```

```
##    state           account_length     area_code         international_plan
##  Length:3333        Min.   :-209.00   Length:3333        Length:3333
##  Class :character   1st Qu.:  72.00   Class :character   Class :character
##  Mode  :character   Median : 100.00   Mode  :character   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.   : 0.000        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   : 8.056        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
##
##
##
##
```

#From the summary table, we can see that all our variables are positive except account_length.

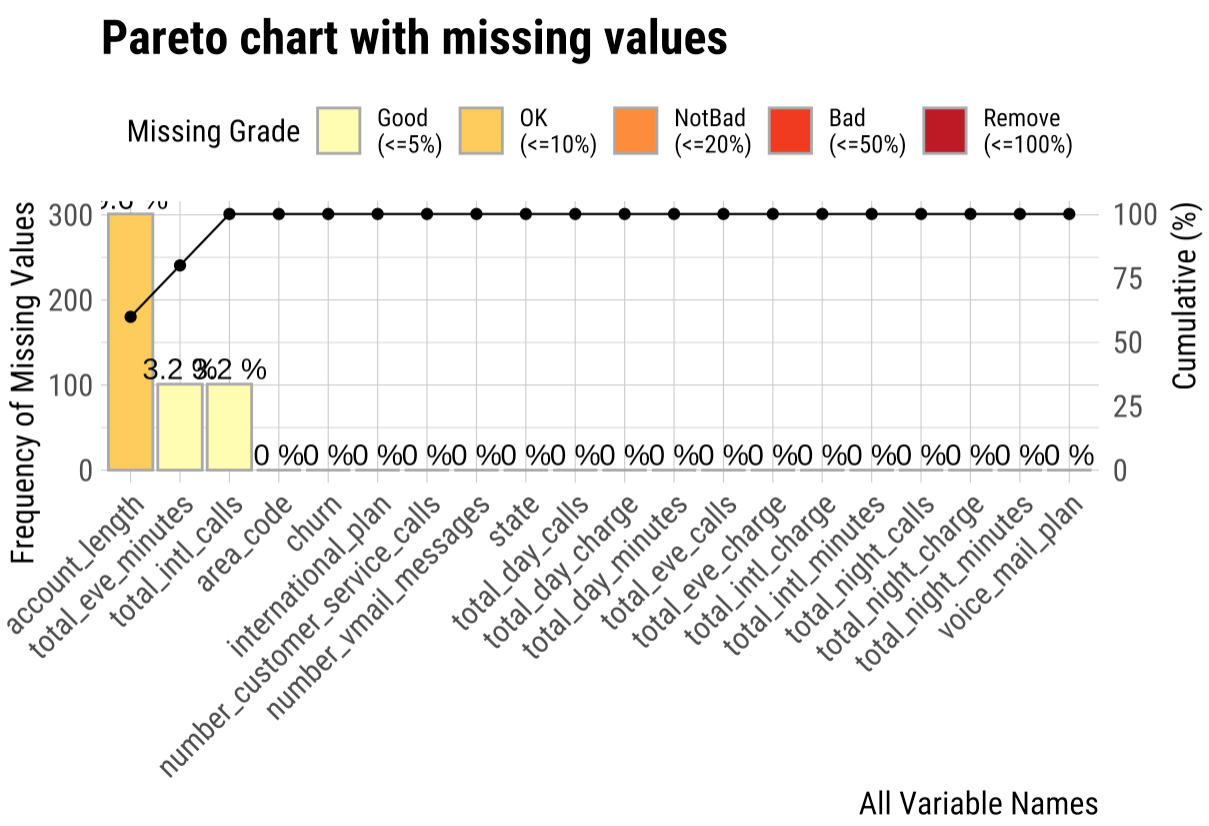
#Any NA in the dataset are always problematic to any machine learning momdel. These values either have to be imputed or removed completely. If the rows have 100% NA's, these would have no predictive power in them. Therefore it is better to remove these rows. #we have eliminated rows that have more than 75% of their elements 'NA' thus removing the rows that are unimportant. This way we can concentrate on imputing rows with less missing NA's.

```
churndata_df_1 <- churndata_df[rowMeans(is.na(churndata_df)) <= 0.25,]
summary(churndata_df_1)
```

```
##     state           account_length     area_code         international_plan
##  Length:3133        Min.   :-209.00   Length:3133        Length:3133
##  Class :character   1st Qu.:  72.00   Class :character   Class :character
##  Mode  :character   Median : 100.00   Mode  :character   Mode  :character
##                     Mean   :  97.32
##                     3rd Qu.: 127.00
##                     Max.   : 243.00
##                     NA's   :301
##  voice_mail_plan    number_vmail_messages total_day_minutes total_day_calls
##  Length:3133        Min.   : 0.000        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   : 8.056        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
##
##  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   :101
##  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.:114.0     3rd Qu.:10.590     3rd Qu.:12.10
##  Max.   :395.0       Max.   :175.0     Max.   :17.770     Max.   :20.00
##
##  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   :101
##     churn
##  Length:3133
##  Class :character
##  Mode  :character
##
##
##
##
```

#Visual presentation of how the NA values have changed:

```
plot_na_pareto(churndata_df_1)
```



#From above we can observe that account length has 9.61 percent of the total, while ‘total eve minutes’ and ‘total intl class’ each have 3.22 percent.

#We will be omitting account_length and state from our data set as they are categorical varibales and to change it to factors it wont give a accurate method.

```
churndata_df_2 <- churndata_df_1 %>% select(-account_length)
summary(churndata_df_2)
```

```
##     state            area_code         international_plan voice_mail_plan
##  Length:3133        Length:3133        Length:3133        Length:3133
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  number_vmail_messages total_day_minutes total_day_calls total_day_charge
##  Min.   : 0.000        Min.   :   0.0    Min.   :  0.0   Min.   : 0.00
##  1st Qu.: 0.000        1st Qu.: 149.3    1st Qu.: 87.0   1st Qu.:24.45
##  Median : 0.000        Median : 190.5    Median :101.0   Median :30.65
##  Mean   : 8.056        Mean   : 418.9    Mean   :100.3   Mean   :30.63
##  3rd Qu.:16.000        3rd Qu.: 237.8    3rd Qu.:114.0   3rd Qu.:36.84
##  Max.   :51.000        Max.   :2185.1    Max.   :165.0   Max.   :59.64
##
##  total_eve_minutes total_eve_calls total_eve_charge total_night_minutes
##  Min.   :   0.0    Min.   :  0.0   Min.   : 0.00    Min.   : 23.2
##  1st Qu.: 170.5    1st Qu.: 87.0   1st Qu.:14.14    1st Qu.:167.3
##  Median : 209.9    Median :100.0   Median :17.09    Median :201.4
##  Mean   : 324.3    Mean   :100.1   Mean   :17.08    Mean   :201.2
##  3rd Qu.: 257.6    3rd Qu.:114.0   3rd Qu.:20.00    3rd Qu.:235.3
##  Max.   :1244.2    Max.   :170.0   Max.   :30.91    Max.   :395.0
##  NA's   :101
##  total_night_calls total_night_charge total_intl_minutes total_intl_calls
##  Min.   : 33.0     Min.   : 1.040     Min.   : 0.00      Min.   : 0.00
##  1st Qu.: 87.0     1st Qu.: 7.530     1st Qu.: 8.50      1st Qu.: 3.00
##  Median :100.0     Median : 9.060     Median :10.30      Median : 4.00
##  Mean   :100.1     Mean   : 9.054     Mean   :10.23      Mean   : 4.47
##  3rd Qu.:114.0     3rd Qu.:10.590     3rd Qu.:12.10      3rd Qu.: 6.00
##  Max.   :175.0     Max.   :17.770     Max.   :20.00      Max.   :20.00
##                                                          NA's   :101
##  total_intl_charge number_customer_service_calls    churn
##  Min.   :0.000     Min.   :0.000                 Length:3133
##  1st Qu.:2.300     1st Qu.:1.000                 Class :character
##  Median :2.780     Median :1.000                 Mode  :character
##  Mean   :2.762     Mean   :1.561
##  3rd Qu.:3.270     3rd Qu.:2.000
##  Max.   :5.400     Max.   :9.000
##
```

#From above the summary shows, the remaining NA values that needs to be imputed for further analysis.

#8.Data Preparation:

```
library("randomForest")
```

#Data Imputation:

#Data Imputation using RandomForest

#The NAs' imputation is updated using the proximity matrix from the randomForest. The imputed value for continuous predictors is the weighted average of the non-missing observations, with the weights being the proximities. For categorical predictors, the imputed value is the category with the largest average proximity. This process is iterated n times.

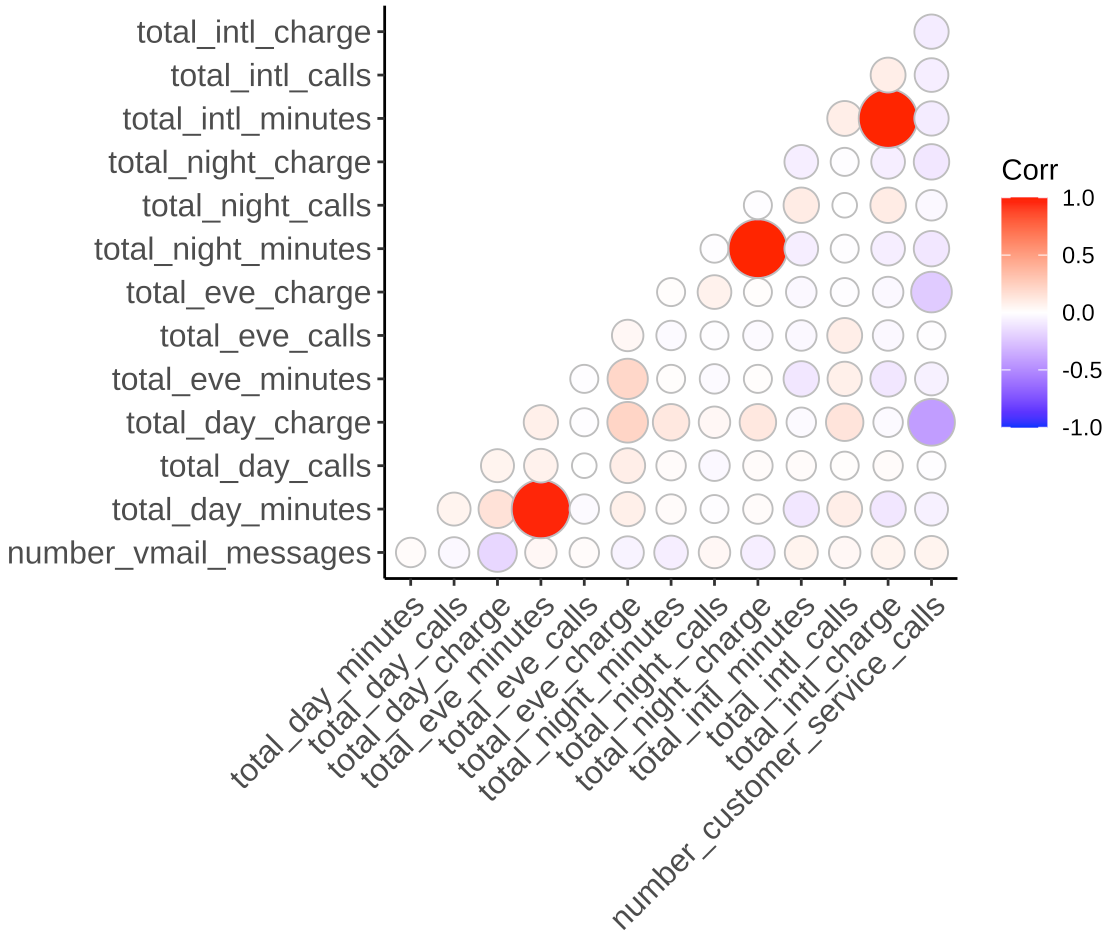
```
str(churndata_df_2)
churndata_df_2$churn =as.factor(churndata_df_2$churn)
churndata_df_2$state = as.factor(churndata_df_2$state)
churndata_df_2$international_plan = as.factor(churndata_df_2$international_plan)
churndata_df_2$voice_mail_plan =as.factor(churndata_df_2$voice_mail_plan)
churndata_df_2$area_code = as.factor(churndata_df_2$area_code)
churndata_df_2 = select(churndata_df_2,-c(area_code))
rf_imputed <- rfImpute(churn ~ ., data = churndata_df_2)
```

#Checking the correlation of the imputed values to understand which model to apply:

```
str(rf_imputed)
```

```
## 'data.frame':    3133 obs. of  18 variables:
##  $ churn                       : Factor w/ 2 levels "no","yes": 1 2 2 1 2 1 1 1 1
2 ...
##  $ state                       : Factor w/ 51 levels "AK","AL","AR",..: 34 12 8 1
2 36 25 28 39 13 16 ...
##  $ international_plan          : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1
1 ...
##  $ voice_mail_plan             : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1
1 ...
##  $ number_vmail_messages       : int  0 0 0 30 0 0 0 0 0 0 ...
##  $ total_day_minutes           : num  2013 292 300 110 337 ...
##  $ total_day_calls             : int  99 99 109 71 120 81 81 87 115 137 ...
##  $ total_day_charge            : num  28.7 49.6 51 18.8 57.4 ...
##  $ total_eve_minutes           : num  1108 221 181 182 227 ...
##  $ total_eve_calls             : int  107 93 100 108 116 74 114 92 112 83 ...
##  $ total_eve_charge            : num  14.9 18.8 15.4 15.5 19.3 ...
##  $ total_night_minutes         : num  243 229 270 184 154 ...
##  $ total_night_calls           : int  92 110 73 88 114 120 82 112 95 111 ...
##  $ total_night_charge          : num  10.95 10.31 12.15 8.27 6.93 ...
##  $ total_intl_minutes          : num  10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7
...
##  $ total_intl_calls            : 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 2.6
5 3.43 ...
##  $ number_customer_service_calls: int  0 2 0 2 0 1 1 3 2 4 ...
```

```
churn_yes<-rf_imputed %>% filter(churn=='yes')
churn_cor<- cor(churn_yes[, 5:18])
ggcorrplot(churn_cor, method = "circle", type = "lower", ggtheme = theme_classic)
```



#9.Data partition:

```
set.seed(123)
train_index<-createDataPartition(rf_imputed$churn, p=0.70, list = FALSE)
train_set<-rf_imputed[train_index,]
test_set<-rf_imputed[-train_index,]
```

#10.Model selection:

```
model_glm<-glm(churn~., data = train_set[,-c(2)], family = "binomial")
summary(model_glm)
```

```
##
## Call:
## glm(formula = churn ~ ., family = "binomial", data = train_set[,
##     -c(2)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7649  -0.5147  -0.3447  -0.2038   3.0901
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -8.214e+00  8.889e-01  -9.241  < 2e-16 ***
## international_planyes        2.048e+00  1.770e-01  11.568  < 2e-16 ***
## voice_mail_planyes          -6.469e-01  4.783e-01  -1.353  0.17621
## number_vmail_messages       -9.123e-03  1.645e-02  -0.555  0.57913
## total_day_minutes           -1.301e-04  1.107e-03  -0.118  0.90642
## total_day_calls              4.314e-03  3.431e-03   1.257  0.20860
## total_day_charge             7.176e-02  9.371e-03   7.658 1.89e-14 ***
## total_eve_minutes            1.069e-04  2.195e-03   0.049  0.96118
## total_eve_calls              5.241e-04  3.475e-03   0.151  0.88013
## total_eve_charge             9.436e-02  2.983e-02   3.163  0.00156 **
## total_night_minutes          6.032e-01  1.080e+00   0.558  0.57666
## total_night_calls           -2.110e-03  3.617e-03  -0.583  0.55962
## total_night_charge          -1.334e+01  2.401e+01  -0.556  0.57849
## total_intl_minutes          -1.226e+00  6.418e+00  -0.191  0.84856
## total_intl_calls            -6.713e-02  3.114e-02  -2.155  0.03113 *
## total_intl_charge            4.819e+00  2.377e+01   0.203  0.83932
## number_customer_service_calls  5.430e-01  4.805e-02  11.299  < 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: 1823.0  on 2193  degrees of freedom
## Residual deviance: 1432.9  on 2177  degrees of freedom
## AIC: 1466.9
##
## Number of Fisher Scoring iterations: 5
```

#11:Evaluating The Accuracy of the glm model:

```
library(pROC)
```

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

```
##
## Attaching package: 'pROC'
```

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

```
churn_rf <- predict(model_glm, newdata = test_set, type = "response")
roc_test <- roc(test_set$churn, churn_rf)
```

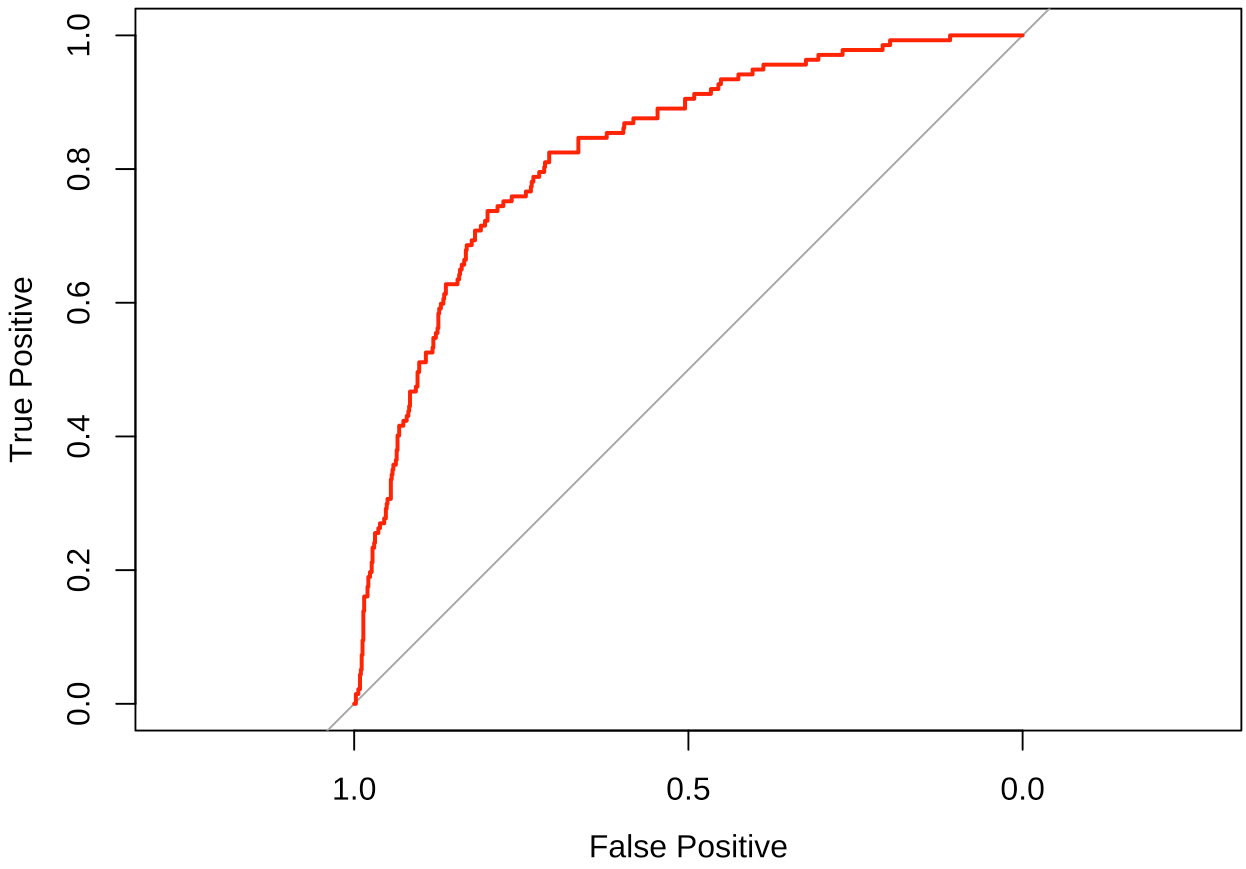
```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
roc_test
```

```
##
## Call:
## roc.default(response = test_set$churn, predictor = churn_rf)
##
## Data: churn_rf in 802 controls (test_set$churn no) < 137 cases (test_set$churn ye
s).
## Area under the curve: 0.8269
```

```
plot(roc_test, col = "red", xlab = "False Positive", ylab = "True Positive")
```



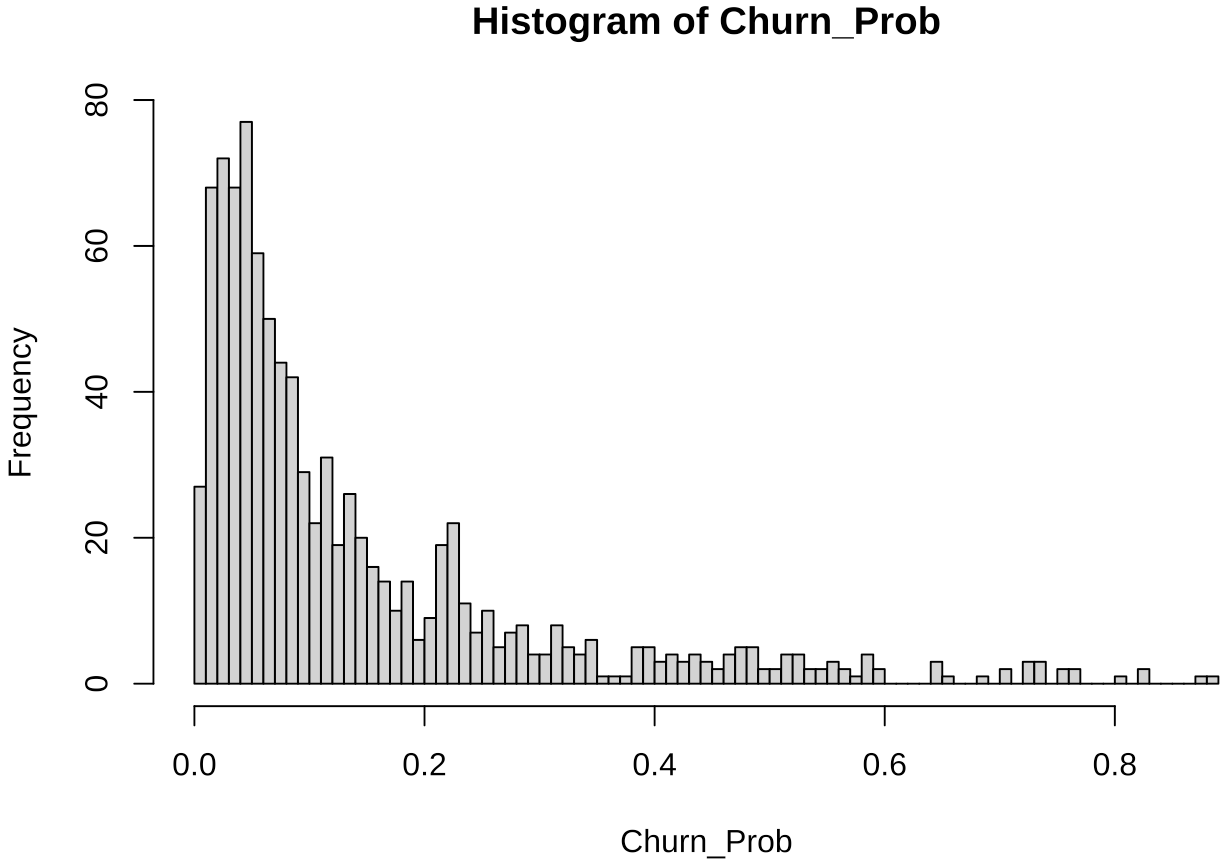
#11.Model Prediction:

#Plotting of glm model with customer to predict data to understand the cut-off value:

```
#load("Customers_To_Predict.RData")
```

#Making The Prediction

```
Churn_Prob <- predict(model_glm, newdata = test_set, type = "response")
hist(Churn_Prob, 100)
```



#Churn_Prob contains all the probabilities (from 0 to 1) that a customer from the test set that customers will churn or not. The histogram above reveals the distribution of the probabilities of churn. The histogram tells us that most customers stayed (i.e. they did not churn). Since the frequency of a customer not churning was higher between the probabilities of 0.0 to 0.3, with the larger subset between 0.0 to 0.2.

#We get the “yes” and “no” churn replies for the ‘Customers To Predict’ dataframe by using the 0.2 churning threshold (cutoff).

```
predicted_churn_status <- as.factor(churn_rf > 0.2)
levels(predicted_churn_status)  <- list(no = "FALSE", yes = "TRUE")
confusion_matrix <- table(predicted_churn_status, actual_churn_status = test_set$chur
n)
confusionMatrix(confusion_matrix, positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##                       actual_churn_status
## predicted_churn_status  no yes
##                     no  668  46
##                     yes 134  91
##
##                Accuracy : 0.8083
##                  95% CI : (0.7816, 0.833)
##     No Information Rate : 0.8541
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.3926
##
##  Mcnemar's Test P-Value : 8.897e-11
##
##             Sensitivity : 0.66423
##             Specificity : 0.83292
##          Pos Pred Value : 0.40444
##          Neg Pred Value : 0.93557
##              Prevalence : 0.14590
##          Detection Rate : 0.09691
##    Detection Prevalence : 0.23962
##       Balanced Accuracy : 0.74858
##
##        'Positive' Class : yes
##
```

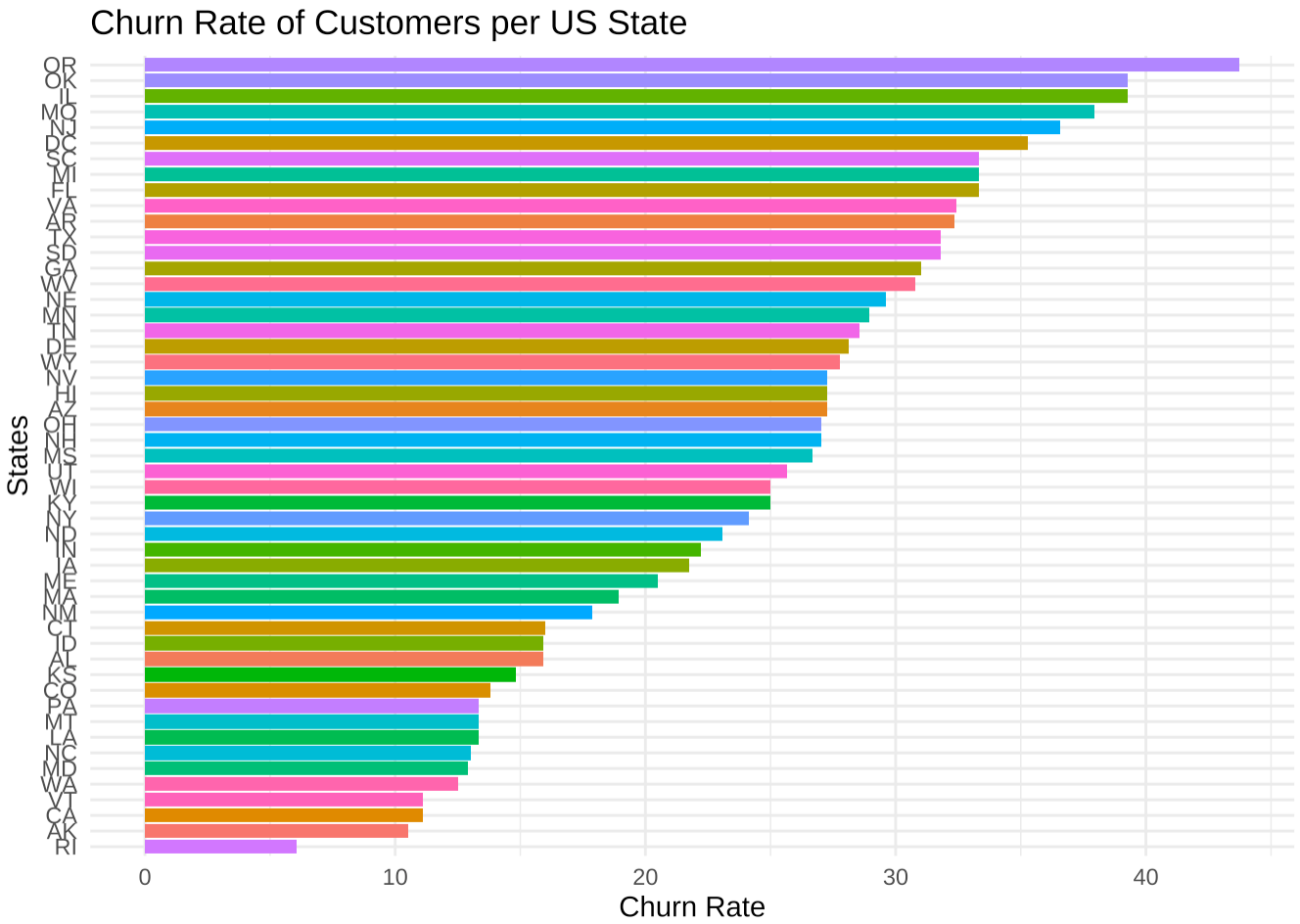
```
load("/Users/duttthakkar/Desktop/Customers_To_Predict.RData")
churn <- rep("no", nrow(Customers_To_Predict))
churn[Churn_Prob > 0.2] = "yes"
Customers_To_Predict$churn <- as.factor(churn)
```

```
calc.churn_rate <- function(churn) {
  count_churn <- function(value) {
    return (churn %>%
              subset(churn == value) %>%
              length())
  }
  num_yes <- count_churn("yes")
  return(num_yes/length(churn) * 100)
}
state_churn_rate <- Customers_To_Predict %>%
  select(state, churn) %>%
  group_by(state) %>%
  summarise(churn_rate = calc.churn_rate(churn))
head(state_churn_rate)
```

```
## # A tibble: 6 × 2
##   state churn_rate
##   <chr>      <dbl>
## 1 AK          10.5
## 2 AL          15.9
## 3 AR          32.4
## 4 AZ          27.3
## 5 CA          11.1
## 6 CO          13.8
```

#Visual presentation of states churn rates:

```
ggplot(state_churn_rate, aes(x = reorder(state, churn_rate),  y = churn_rate, fill =
state)) +
  geom_bar(stat = 'identity') +
  coord_flip() +
  theme_minimal() +
  guides(fill = F) +
  labs(x = "States", y = "Churn Rate", title = "Churn Rate of Customers per US Stat
e")
```



```
calc.churn_rate <- function(churn) {
  count_churn <- function(value) {
    return (churn %>%
              subset(churn == value) %>%
              length())
  }
  num_yes <- count_churn("yes")
  return(num_yes/length(churn) * 100)
}
number_customer_service_calls_churn_rate <- Customers_To_Predict %>%
  select(number_customer_service_calls, churn) %>%
  group_by(number_customer_service_calls) %>%
  summarise(number_customer_service_calls_churn_rate = calc.churn_rate(churn)) %>% ar
range(desc(number_customer_service_calls_churn_rate))
head(number_customer_service_calls_churn_rate)
```

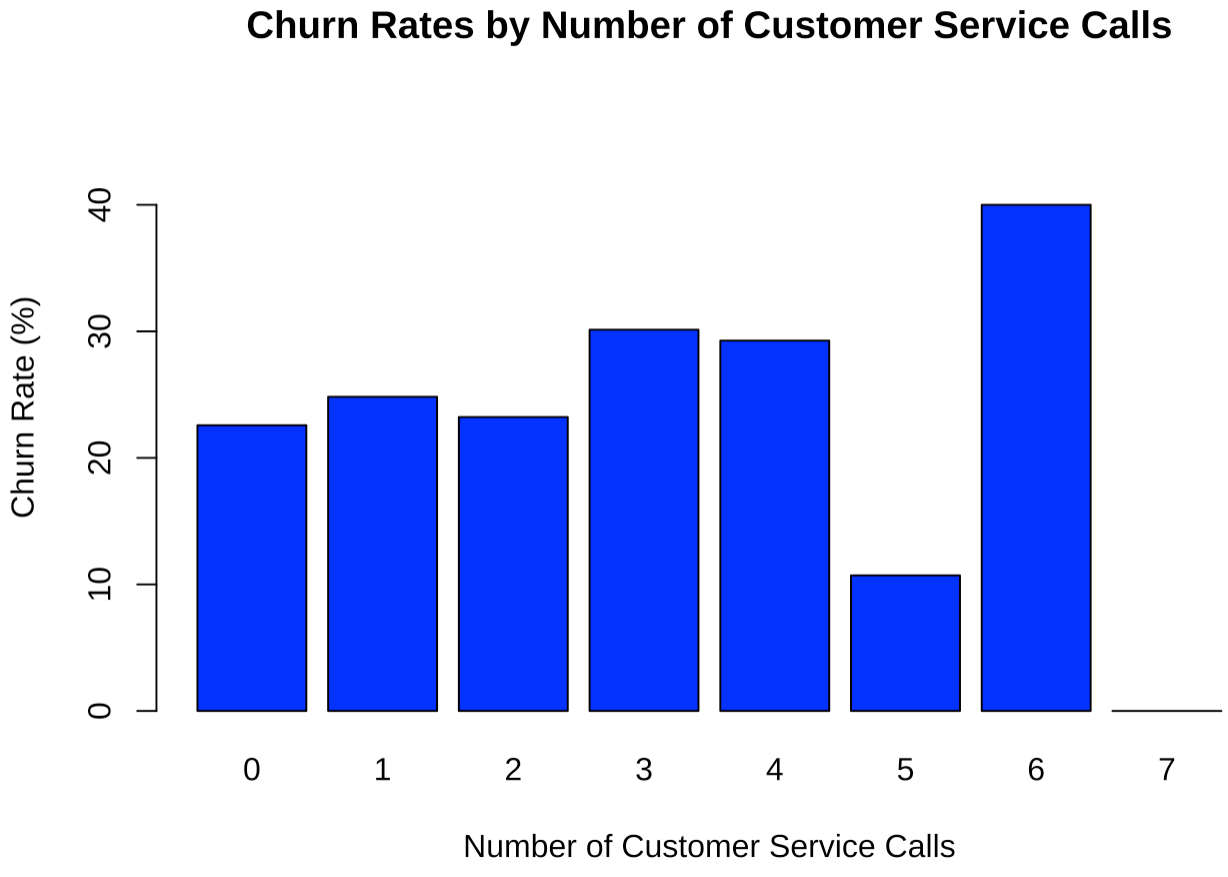
```
## # A tibble: 6 × 2
##   number_customer_service_calls number_customer_service_calls_churn_rate
##                           <dbl>                                    <dbl>
## 1                             6                                     40
## 2                             3                                     30.1
## 3                             4                                     29.3
## 4                             1                                     24.8
## 5                             2                                     23.2
## 6                             0                                     22.6
```

#Graphical representation.

```
calc.churn_rate <- function(churn) {
  count_churn <- function(value) {
    return(churn %>%
             subset(churn == value) %>%
             length())
  }
  num_yes <- count_churn("yes")
  return(num_yes/length(churn) * 100)
}

churn_rates <- Customers_To_Predict %>%
  select(number_customer_service_calls, churn) %>%
  group_by(number_customer_service_calls) %>%
  summarise(number_customer_service_calls_churn_rate = calc.churn_rate(churn)) %>%
  arrange(number_customer_service_calls)

barplot(churn_rates$number_customer_service_calls_churn_rate,
        names.arg = churn_rates$number_customer_service_calls,
        xlab = "Number of Customer Service Calls",
        ylab = "Churn Rate (%)",
        main = "Churn Rates by Number of Customer Service Calls",
        col = "blue",
        ylim = c(0, max(churn_rates$number_customer_service_calls_churn_rate) * 1.2))
```



```
calc.churn_rate <- function(churn) {
  count_churn <- function(value) {
    return (churn %>%
              subset(churn == value) %>%
              length())
  }
  num_yes <- count_churn("yes")
  return(num_yes/length(churn) * 100)
}
total_intl_calls_churn_rate <- Customers_To_Predict %>%
  select(total_intl_calls , churn) %>%
  group_by(total_intl_calls) %>%
  summarise(total_intl_calls_churn_rate = calc.churn_rate(churn)) %>% arrange(desc(to
tal_intl_calls_churn_rate))
head(total_intl_calls_churn_rate)
```

```
## # A tibble: 6 × 2
##   total_intl_calls total_intl_calls_churn_rate
##              <dbl>                       <dbl>
## 1               18                       100
## 2                0                        50
## 3               15                        50
## 4               11                        43.8
## 5                3                        29.6
## 6                7                        28.7
```

```
calc.churn_rate <- function(churn) {
  count_churn <- function(value) {
    return (churn %>%
              subset(churn == value) %>%
              length())
  }
  num_yes <- count_churn("yes")
  return(num_yes/length(churn) * 100)
}
total_day_calls_churn_rate <- Customers_To_Predict %>%
  select(total_day_calls , churn) %>%
  group_by(total_day_calls) %>%
  summarise(total_day_calls_churn_rate = calc.churn_rate(churn)) %>% arrange(desc(tot
al_day_calls_churn_rate))
head(total_day_calls_churn_rate)
```

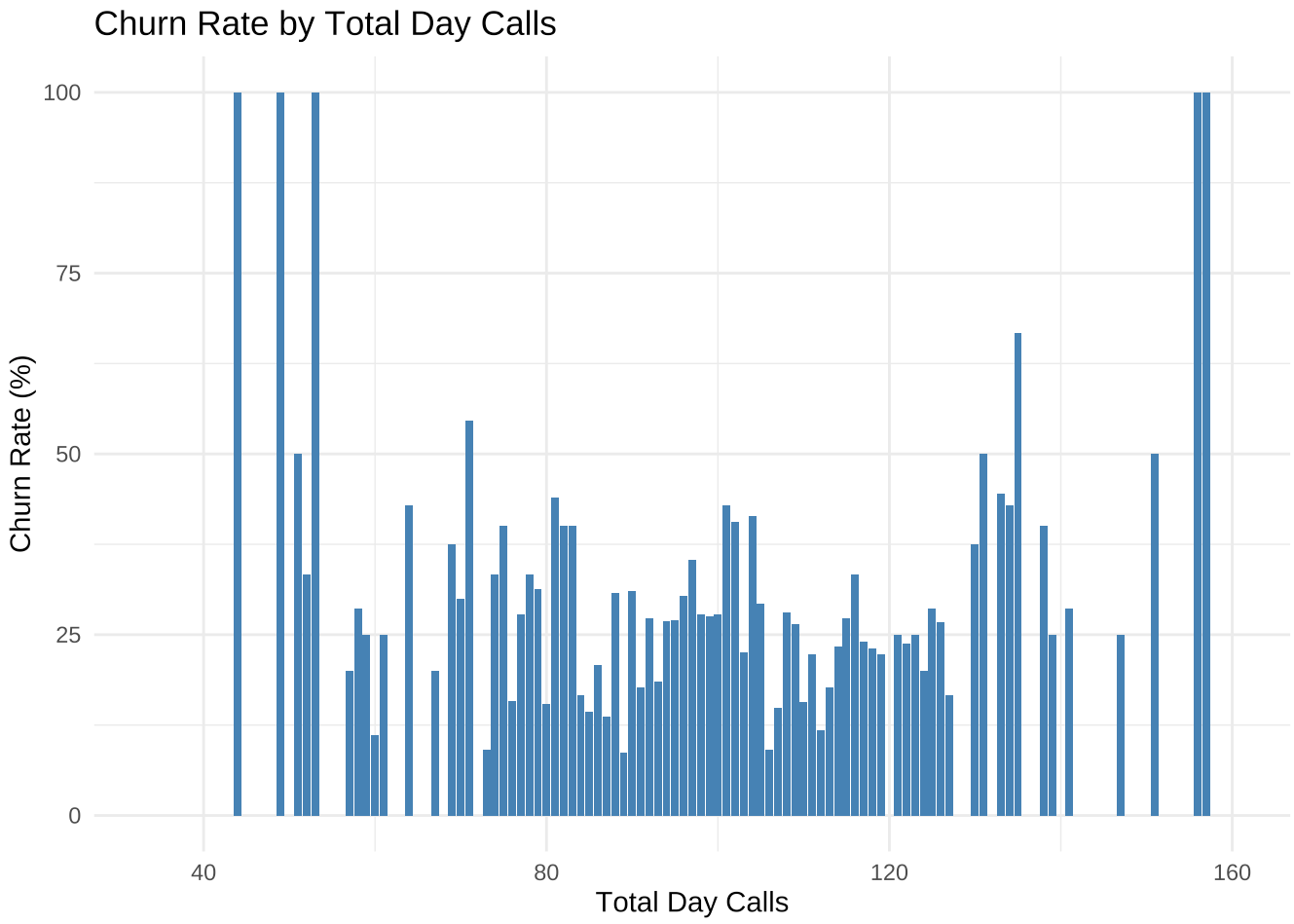
```
## # A tibble: 6 × 2
##   total_day_calls total_day_calls_churn_rate
##             <dbl>                      <dbl>
## 1              44                      100
## 2              49                      100
## 3              53                      100
## 4             156                      100
## 5             157                      100
## 6             135                       66.7
```

Graphical presentation of total_day_calls vs. total_day_calls_churn_rate

```
calc_churn_rate <- function(churn) {
  count_churn <- function(value) {
    return(churn[churn == value] %>% length())
  }
  num_churned <- count_churn("yes")
  return(num_churned / length(churn) * 100)
}


total_day_calls_churn_rate <- Customers_To_Predict %>%
  select(total_day_calls, churn) %>%
  group_by(total_day_calls) %>%
  summarise(churn_rate = calc_churn_rate(churn)) %>%
  arrange(total_day_calls)


ggplot(total_day_calls_churn_rate, aes(x = total_day_calls, y = churn_rate)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Churn Rate by Total Day Calls",
       x = "Total Day Calls",
       y = "Churn Rate (%)") +
  theme_minimal()
```




```
write.csv(Customers_To_Predict, file = "churned_data.csv")
```