Customer Churn Prediction

Overview of Customer Churning:

Churn (loss of customers to competition) is a problem for telecom companies because it is more expensive to acquire a new customer than to keep your existing one from leaving. This project is about enabling churn reduction using analytics

Purpose of this Project:

In the targeted approach the company tries to identify in advance customers who are likely to churn.The company then targets those customers with special programs or incentives. This approach can bring in huge loss for a company, if churn predictions are inaccurate, because then firms are wasting incentive money on customers who would have stayed anyway

Problem Statement:

ABC Wireless Inc. has hired you to help them with the customers' churn issue. In this project, you will be working as a part of a team to use historical data from ACB Wireless Inc. to build a model that can predict/identify their customers who are likely to churn.

Loading library functions

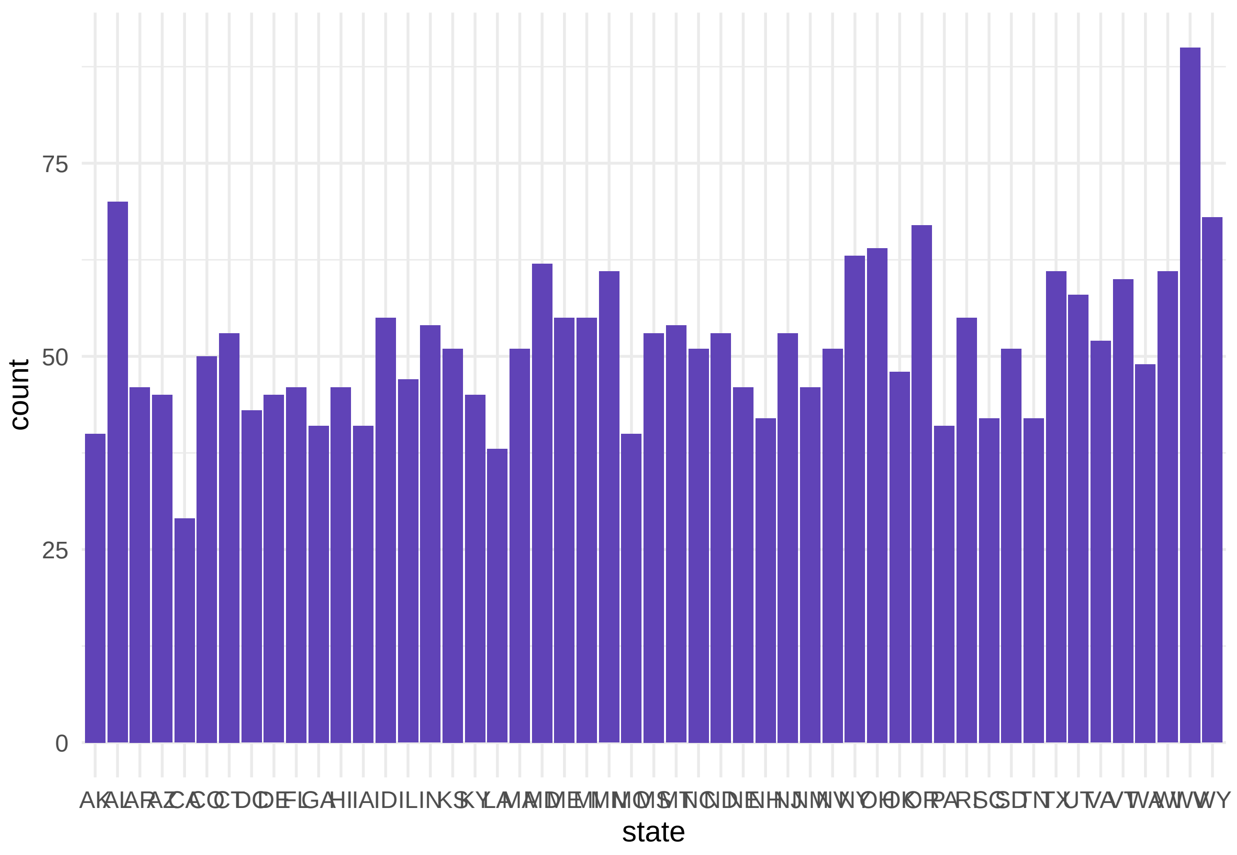
```r
library(dplyr)
library(caret)
library(ggplot2)
library(esquisse)
```

Importing Data and performing Data Cleaning

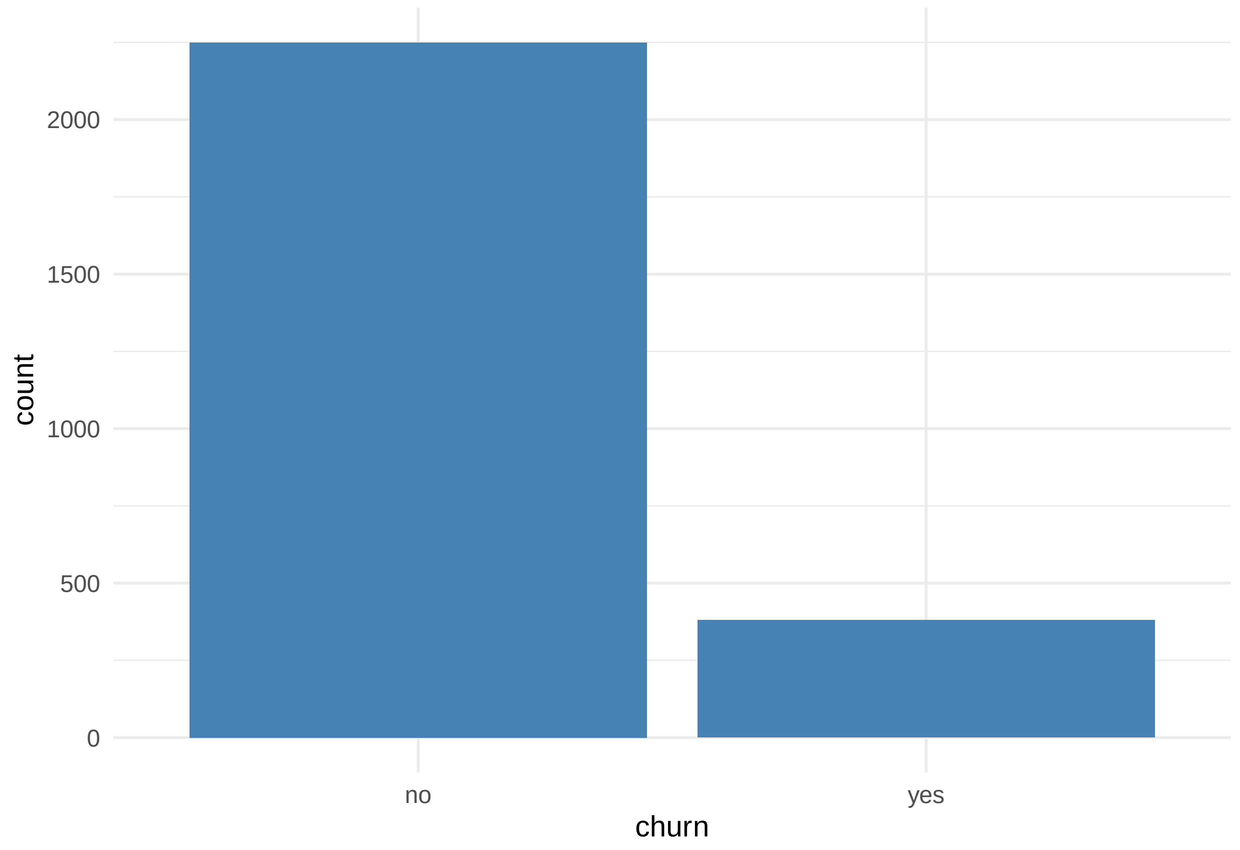
```r
data1<-read.csv("Churn_Train.csv")

View(data1)
load("Customers_To_Predict.RData")

#Check for average missing values in each column

perc<-data1%>%is.na()%>%colMeans()*100


#Data cleansing -removing all NA values

data2 <-data1%>%na.omit
```

Exploratory Data Analysis:

```r
#Plotting Customers in each state:
ggplot(data2) +
  aes(x = state) +
  geom_bar(fill = "#6043B7") +
  theme_minimal()
```



```
#Plotting to check the count of customers being churned in the past years:
ggplot(data2) +
  aes(x = churn) +
  geom_bar(fill = "#4682B4") +
  theme_minimal()
```



The above plot shows that the data is not balanced.

Data Transformation:

```
#Converting churn column to factor levels
data2$churn = as.factor(data2$churn)

head(data2$churn)
```

```
## [1] no  yes yes yes no  no
## Levels: no yes
```

Factor levels have not been interchanged as the second level of the factor is already "yes".

Creating Data Partition- Splitting data into 50:50 ratio as train and validation

```
set.seed(452)
Split_data<- createDataPartition(data2$churn,p=.5,list=FALSE,times=1)
train<-data2[Split_data,]
Validation<-data2[-Split_data,]
```

Developing Logistic regression model using train data

```
model<-glm(train$churn~. , data=train,family=binomial)

#interpretation of the model
summary(model)
```

```
##
## Call:
## glm(formula = train$churn ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.20728  -0.45784  -0.23907  -0.09355   3.10086
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.091e+01  1.742e+00  -6.263 3.77e-10 ***
## stateAL                      -1.423e+01  5.720e+02  -0.025  0.98015
## stateAR                       1.810e+00  1.215e+00   1.490  0.13631
## stateAZ                       1.149e+00  1.338e+00   0.859  0.39040
## stateCA                       2.347e+00  1.382e+00   1.698  0.08943 .
## stateCO                       8.982e-01  1.321e+00   0.680  0.49663
## stateCT                       4.469e-01  1.353e+00   0.330  0.74114
## stateDC                       1.710e+00  1.366e+00   1.251  0.21077
## stateDE                       1.655e+00  1.226e+00   1.350  0.17714
## stateFL                       1.141e+00  1.245e+00   0.917  0.35931
## stateGA                       2.600e-04  1.404e+00   0.000  0.99985
## stateHI                      -1.561e-01  1.399e+00  -0.112  0.91117
## stateIA                       3.812e-01  1.537e+00   0.248  0.80414
## stateID                       1.133e+00  1.321e+00   0.857  0.39137
## stateIL                      -5.466e-02  1.378e+00  -0.040  0.96836
## stateIN                       3.976e-01  1.326e+00   0.300  0.76432
## stateKS                       1.173e+00  1.230e+00   0.954  0.34001
## stateKY                       1.971e+00  1.303e+00   1.512  0.13050
## stateLA                       4.501e-01  1.547e+00   0.291  0.77104
## stateMA                       7.440e-01  1.358e+00   0.548  0.58372
## stateMD                       1.592e+00  1.209e+00   1.317  0.18788
## stateME                       2.509e+00  1.219e+00   2.059  0.03954 *
## stateMI                       2.037e+00  1.249e+00   1.631  0.10287
## stateMN                       8.657e-01  1.239e+00   0.699  0.48459
## stateMO                      -5.383e-01  1.571e+00  -0.343  0.73194
## stateMS                       1.994e+00  1.246e+00   1.600  0.10955
## stateMT                       2.579e+00  1.185e+00   2.175  0.02959 *
## stateNC                       7.296e-01  1.265e+00   0.577  0.56421
## stateND                       7.216e-01  1.358e+00   0.531  0.59511
## stateNE                       1.054e+00  1.369e+00   0.770  0.44137
## stateNH                       1.571e+00  1.303e+00   1.206  0.22800
## stateNJ                       1.908e+00  1.244e+00   1.534  0.12501
## stateNM                       7.653e-01  1.286e+00   0.595  0.55172
## stateNV                       1.906e+00  1.266e+00   1.506  0.13208
## stateNY                      -3.712e-02  1.362e+00  -0.027  0.97825
## stateOH                      -9.688e-01  1.559e+00  -0.622  0.53426
## stateOK                       1.314e+00  1.233e+00   1.066  0.28646
## stateOR                       1.202e+00  1.246e+00   0.964  0.33483
## statePA                       2.024e+00  1.240e+00   1.632  0.10276
## stateRI                      -4.174e-01  1.674e+00  -0.249  0.80305
## stateSC                       1.844e+00  1.265e+00   1.458  0.14495
## stateSD                      -4.379e-02  1.553e+00  -0.028  0.97750
## stateTN                       9.505e-01  1.290e+00   0.737  0.46114
## stateTX                       2.745e+00  1.195e+00   2.297  0.02160 *
## stateUT                         1.456e+00  1.254e+00   1.161  0.24560
## stateVA                         1.194e-01  1.430e+00   0.084  0.93342
## stateVT                        -1.197e+00  1.421e+00  -0.842  0.39957
## stateWA                         2.610e+00  1.238e+00   2.108  0.03502 *
## stateWI                         1.044e+00  1.325e+00   0.788  0.43063
## stateWV                         1.668e+00  1.187e+00   1.405  0.16001
## stateWY                         1.595e+00  1.221e+00   1.307  0.19132
## account_length                  7.543e-04  2.173e-03   0.347  0.72848
## area_codearea_code_415          2.037e-01  2.430e-01   0.838  0.40202
## area_codearea_code_510         -2.060e-02  2.878e-01  -0.072  0.94293
## international_planyes           2.589e+00  2.842e-01   9.110  < 2e-16 ***
## voice_mail_planyes             -1.752e+00  6.249e-01  -2.804  0.00505 **
## number_vmail_messages           1.521e-02  1.972e-02   0.771  0.44063
## total_day_minutes              -5.071e-03  3.428e-03  -1.479  0.13906
## total_day_calls                 2.056e-03  4.881e-03   0.421  0.67354
## total_day_charge                1.127e-01  2.137e-02   5.273 1.34e-07 ***
## total_eve_minutes               9.594e-03  6.759e-03   1.420  0.15575
## total_eve_calls                 8.712e-04  5.006e-03   0.174  0.86186
## total_eve_charge               -1.169e-02  8.227e-02  -0.142  0.88700
## total_night_minutes             1.542e+00  1.553e+00   0.993  0.32080
## total_night_calls               2.677e-03  5.119e-03   0.523  0.60101
## total_night_charge             -3.418e+01  3.451e+01  -0.990  0.32195
## total_intl_minutes              3.228e+00  9.435e+00   0.342  0.73227
## total_intl_calls               -7.306e-02  4.385e-02  -1.666  0.09569 .
## total_intl_charge              -1.168e+01  3.494e+01  -0.334  0.73822
## number_customer_service_calls  7.515e-01  7.562e-02   9.938  < 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: 1086.26  on 1314  degrees of freedom
## Residual deviance:  730.46  on 1245  degrees of freedom
## AIC: 870.46
##
## Number of Fisher Scoring iterations: 16
```

Eliminating all the columns with p value greater than 5%(NULL hypothesis).Independent variables with p value less than 5% has significance in predicting value for the predicted variable(churn).

```
data4<-train[,c(4,5,7,9,10,19,20)]
View(data4)
```

Developing new model with selected independent variables

```
#Developing a new model with selected dependent variables:
model2<-glm(data4$churn~.,data=data4,family=binomial)

summary(model2)
```

```
##
## Call:
## glm(formula = data4$churn ~ ., family = binomial, data = data4)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0664  -0.5111  -0.3194  -0.1848   3.1349
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -6.667882   0.572121 -11.655  < 2e-16 ***
## international_planyes          2.191671   0.239730   9.142  < 2e-16 ***
## voice_mail_planyes            -1.187697   0.241488  -4.918 8.73e-07 ***
## total_day_minutes             -0.003683   0.000919  -4.007 6.14e-05 ***
## total_day_charge               0.092512   0.011824   7.824 5.11e-15 ***
## total_eve_minutes              0.006694   0.001788   3.745  0.00018 ***
## number_customer_service_calls  0.643115   0.065022   9.891  < 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: 1086.26  on 1314  degrees of freedom
## Residual deviance:  833.41  on 1308  degrees of freedom
## AIC: 847.41
##
## Number of Fisher Scoring iterations: 6
```

```
#Calculating variable importance for objects produced by train data

varImp(model2, scale = FALSE)
```

```
##                                 Overall
## international_planyes         9.142262
## voice_mail_planyes            4.918249
## total_day_minutes             4.007233
## total_day_charge              7.824124
## total_eve_minutes             3.745186
## number_customer_service_calls 9.890739
```

As the p value for all independent variables is too small, it can be concluded that there is a significant relationship between each of these columns with the dependent variable.

Here,threshold for the VarImp =2,ie we consider only variables having Varimp>=2

Deploying the developed model on Validation data

```
#Filtering columns based on varImp to Validation data:

Predict_Validation<-Validation[c(4,5,7,9,10,14,19,20)]

Churn_Prob_Validation<-predict(model2, data = Predict_Validation, type = "response")
```

Threshold for the churn probability is chosen to be 30% as we want to predict all the customers who has the slightest chance of churning out.

```
class_prediction <- ifelse(Churn_Prob_Validation > 0.3, "yes", "no")


#Combining  the predictions value column  to validation data

com_data<-cbind(Predict_Validation,class_prediction)
```

Converting churn variable in validation set to factor levels

```
com_data$churn<-as.factor(com_data$churn)

com_data$class_prediction<-as.factor(com_data$class_prediction)
```

Developing Confusion Matrix with validation data

```
matrix=confusionMatrix(com_data$churn, com_data$class_prediction,positive="yes")

matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  962 163
##        yes 168  22
##
##                Accuracy : 0.7483
##                  95% CI : (0.7239, 0.7715)
##     No Information Rate : 0.8593
##     P-Value [Acc > NIR] : 1.000
##
##                   Kappa : -0.0294
##
##  Mcnemar's Test P-Value : 0.826
##
##             Sensitivity : 0.11892
##             Specificity : 0.85133
##          Pos Pred Value : 0.11579
##          Neg Pred Value : 0.85511
##              Prevalence : 0.14068
##          Detection Rate : 0.01673
##    Detection Prevalence : 0.14449
##       Balanced Accuracy : 0.48512
##
##        'Positive' Class : yes
##
```

Interpretation:

The accuracy of the model is 74.83%.

Based on the data distribution for churn variable shown below, we can say that the data is unbalanced. This factor has an impact on the accuracy of the model.

Another performance metrics to be considered is Specificity. We could see that the specificity is 85.13% which is the TRUE NEGATIVE RATE. Specificity, in general, indicates how well the model identifies the negative values out of all the negative values.

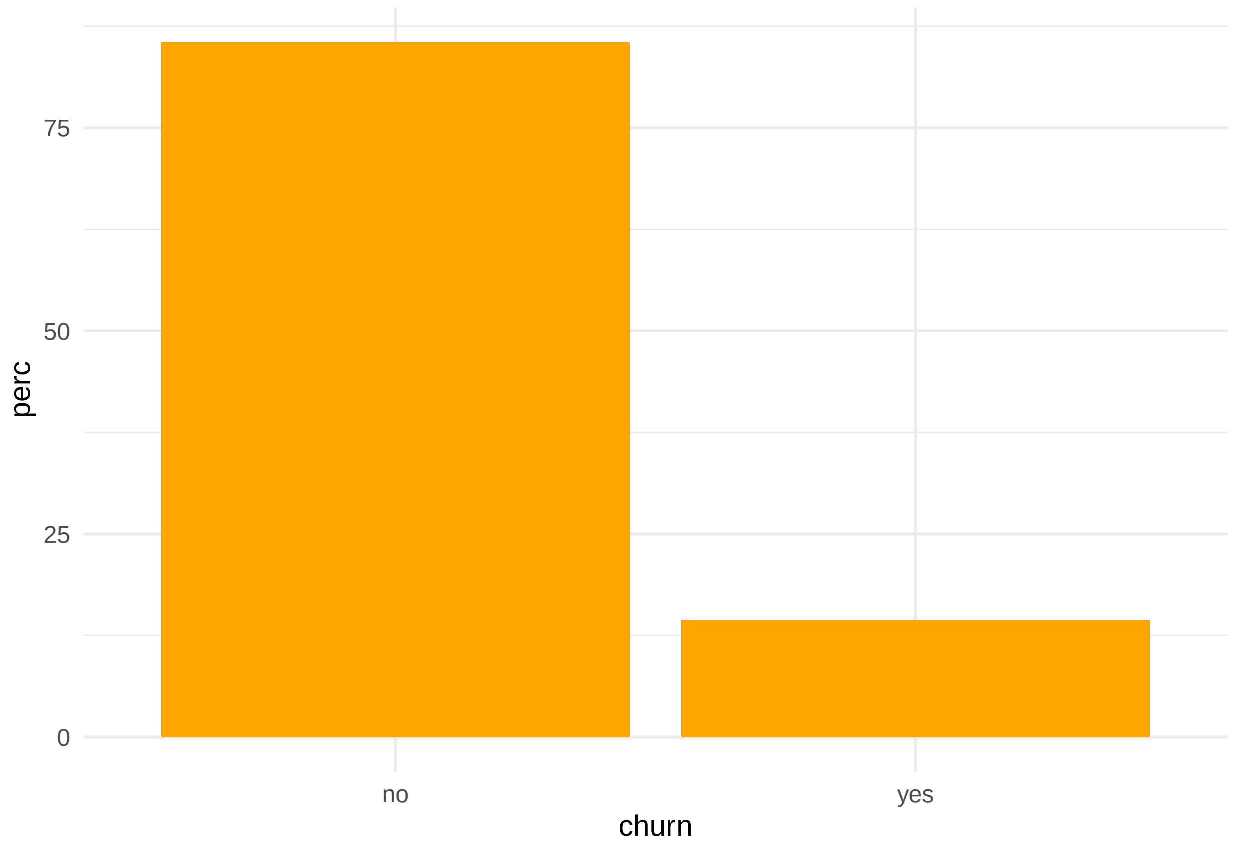
The reason for considering this performance metrics is that if we identify any customer as the one who is going to churn out wrongly, then the company will invest in strategies for that customer who was anyway going to stay. This will be an additional loss to the company as they are spending unnecessarily on such customers. Hence, it becomes important to us to not identify any customer to be churning out falsely.

Calculating data distribution for churn column

```
churn_perc<-data4 %>% group_by(churn) %>% summarise(cnt = n()) %>% mutate(perc =round((cnt/sum(cnt))*10
View(churn_perc)
```

Plotting the raw data churn column for better understanding/data distribution

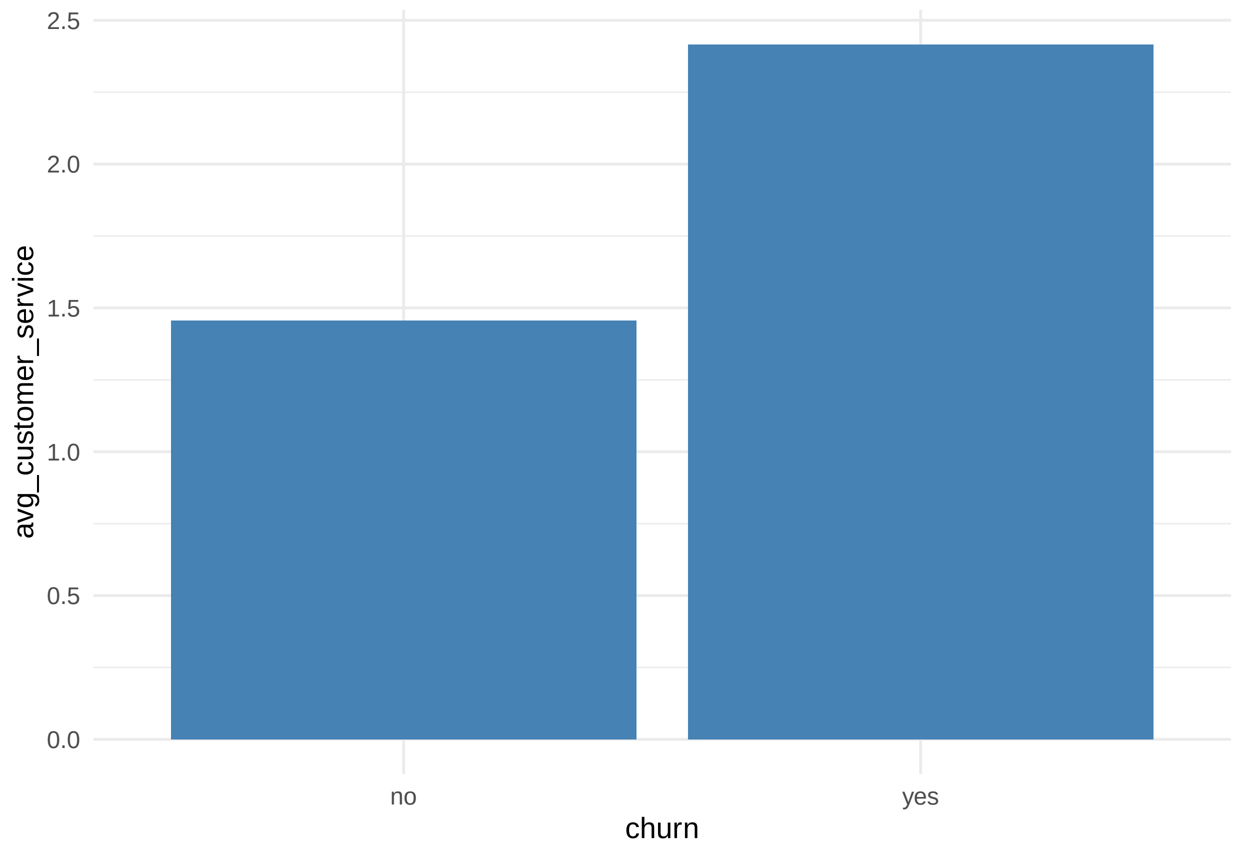
```
ggplot(churn_perc) +
 aes(x = churn, y = perc) +
 geom_col(fill = "#FFA500") +
 theme_minimal()
```



Average number of customer service calls made by customers with churn value yes and no

```
avg_cust_calls<-data4 %>% group_by(churn) %>% summarise(avg_customer_service=mean(number_customer_servi
```

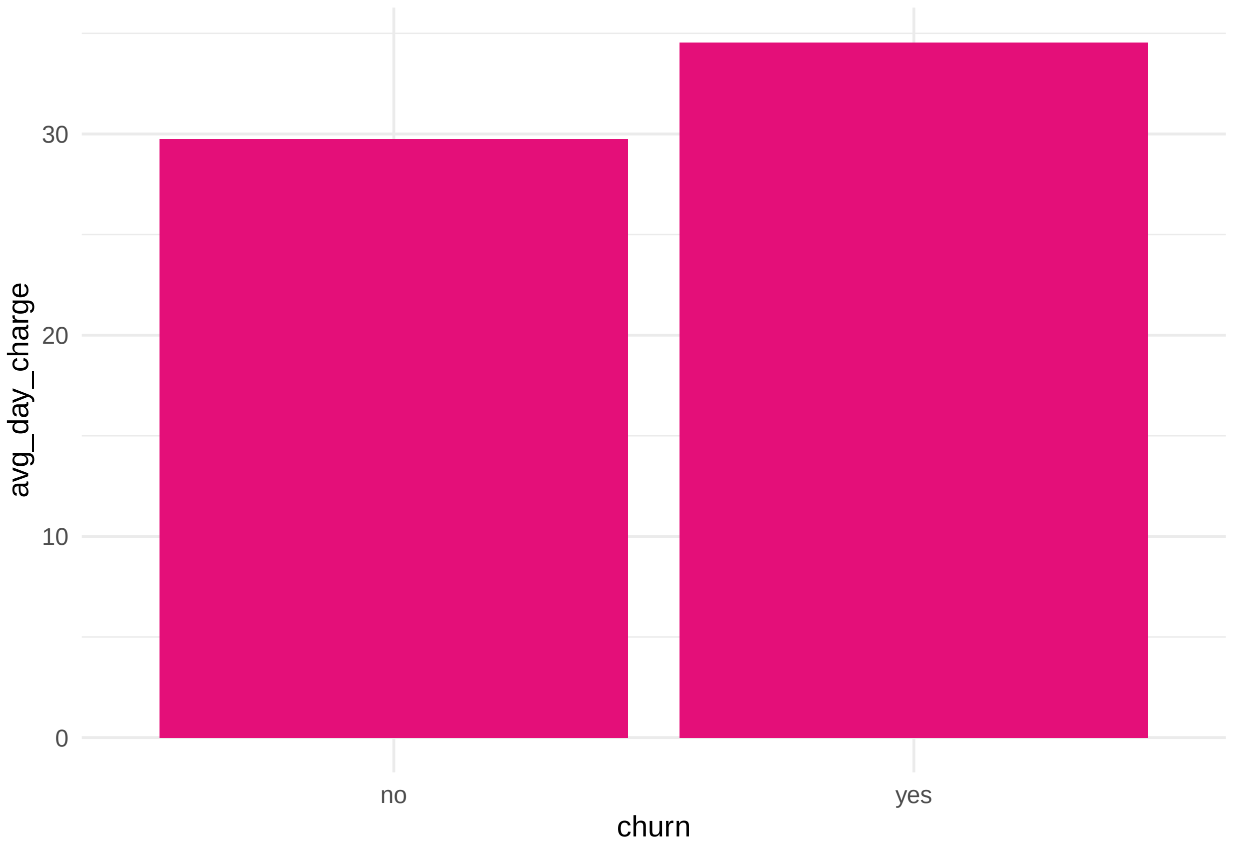
```r
#Plot

ggplot(avg_cust_calls) +
 aes(x = churn, y = avg_customer_service) +
 geom_col(fill = "#4682B4") +
 theme_minimal()
```



Based on the above plot, we can say that the customers who made the most number of customer service calls are to be the one's who are most likely to churn. These could be the customers who might have called the most times to raise their concerns. So, company should target customers who are calling them frequently and solve their problems on priority. This will also improve the customer satisfaction.

Avg number of total day charge for yes and no values

```r
avg_day_charge<-data4 %>% group_by(churn) %>% summarise(avg_day_charge=mean(total_day_charge))
View(avg_day_charge)

#Plot

ggplot(avg_day_charge) +
 aes(x = churn, y = avg_day_charge) +
 geom_col(fill = "#E40F79") +
 theme_minimal()
```



The plot indicates that the customers who are spending more on an average every day are the one's who might churn out. These customers could probably think that they are spending more on this network and if they are getting better deals at an other company, they might consider churn out. Therefore, company should target the customers who have taken the largest spending plan per day and offer discounts to them.

Predicting the probablity of each customer on Test Data

```
#Customers_To_Predict<-Customers_To_Predict[c(4,5,7,9,10,14,19)]

Churn_Prob<-predict(model2, data = Customers_To_Predict, type = "response")

#Threshold for churn probability= 30%.

Pred_data<-ifelse(Churn_Prob>0.3,"yes","no")
```