

Mini Project 5

Customer Churn



May 16, 2019

Saurabh MUDGal

Table of Contents

[Business Case 2](#_Toc9976720)

[Data to be analysed 2](#_Toc9976721)

[Library Used 2](#_Toc9976722)

[Exploratory Data Analysis and Descriptive Analysis 3](#_Toc9976723)

[**Environment Set up and Data Import** 3](#_Toc9976724)

[**Variable Identification** 3](#_Toc9976725)

[**Univariate Analysis** 4](#_Toc9976726)

[Bi-Variate/Multivariate Analysis 10](#_Toc9976727)

[Model Creation 15](#_Toc9976728)

[Model 1 16](#_Toc9976729)

[Model 2 18](#_Toc9976730)

[Model 3 20](#_Toc9976731)

[Model 4 22](#_Toc9976732)

[Model 5 24](#_Toc9976733)

[Model Comparison 26](#_Toc9976734)

[**Appendix** 27](#_Toc9976735)

[R Markdown 27](#_Toc9976736)

[Including Plots 68](#_Toc9976737)

# Business Case

Objective is reduce customer churn for Telecom service provider by identifying the potential churn candidates beforehand and take proactive actions to make them stay.

Customer churn is subscriber’s cancelling their mobile or fixed line services or porting out to another service provider. Customer churn is loss to service provider as its really difficult to win back the lost customer and it’s bad marketing if he/she is leaving due to bad customer or user experience.

There can be any reason for customer churn

* Network issues
* Billing issues
* Competitive offers from another Telecom Service Provider
* Financial hardship
* Bad user experience
* High roaming charges
* Overage charge

We will try to build model as per existing data for the customer who has already churned and their behaviour before the churned happened. This will help to identify potential customer who may get churn in future and service provider can take necessary steps to stop same .

# Data to be analysed

Attached data will be used to build the model.

# Library Used

library(knitr)

library(ggplot2)

library(dplyr)

library(car)

library(caret)

library (ROCR)

library(pROC)

# Exploratory Data Analysis and Descriptive Analysis

Typical Data exploration activity consists of the following steps:

## **Environment Set up and Data Import**

Please refer Appendix section

## **Variable Identification**

- 11 variables (features) and 3333 rows

|  |  |  |
| --- | --- | --- |
| 1 | Churn | 1 if customer cancelled service, 0 if not |
| 2 | AccountWeeks | number of weeks customer has had active account |
| 3 | ContractRenewal | 1 if customer recently renewed contract, 0 if not |
| 4 | DataPlan | 1 if customer has data plan, 0 if not |
| 5 | DataUsage | gigabytes of monthly data usage |
| 6 | CustServCalls | number of calls into customer service |
| 7 | DayMins | average daytime minutes per month |
| 8 | DayCalls | average number of daytime calls |
| 9 | MonthlyCharge | average monthly bill |
| 10 | OverageFee | largest overage fee in last 12 months |
| 11 | RoamMins | average number of roaming minutes |

- All variables of type number

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 3333 obs. of 11 variables:

$ Churn : num 0 0 0 0 0 0 0 0 0 0 ...

$ AccountWeeks : num 128 107 137 84 75 118 121 147 117 141 ...

$ ContractRenewal: num 1 1 1 0 0 0 1 0 1 0 ...

$ DataPlan : num 1 1 0 0 0 0 1 0 0 1 ...

$ DataUsage : num 2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...

$ CustServCalls : num 1 1 0 2 3 0 3 0 1 0 ...

$ DayMins : num 265 162 243 299 167 ...

$ DayCalls : num 110 123 114 71 113 98 88 79 97 84 ...

$ MonthlyCharge : num 89 82 52 57 41 57 87.3 36 63.9 93.2 ...

$ OverageFee : num 9.87 9.78 6.06 3.1 7.42 ...

$ RoamMins : num 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...

- No missing value identified in data set

> colSums(is.na(CustChurn))

Churn AccountWeeks ContractRenewal DataPlan DataUsage

0 0 0 0 0

CustServCalls DayMins DayCalls MonthlyCharge OverageFee

0 0 0 0 0

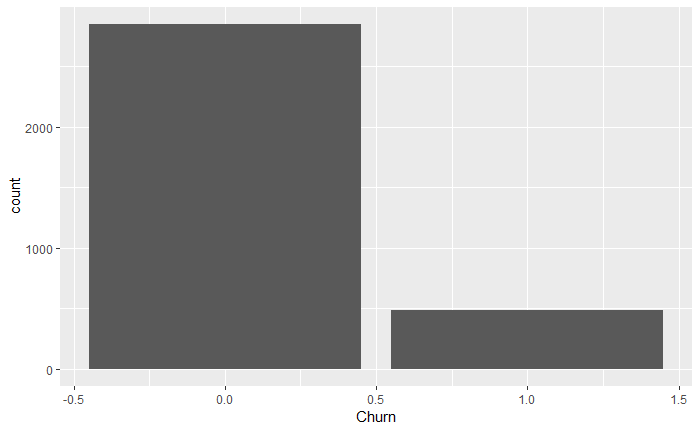
RoamMins

0

## **Univariate Analysis**

1. **Churn**

* Churn has 2 values 0 and 1
* Out of 3333 customers , 2850 customer have this value as 0(Not Churned ) and 483 have value as 1(Churned ).
* 14% of the customer is churned .



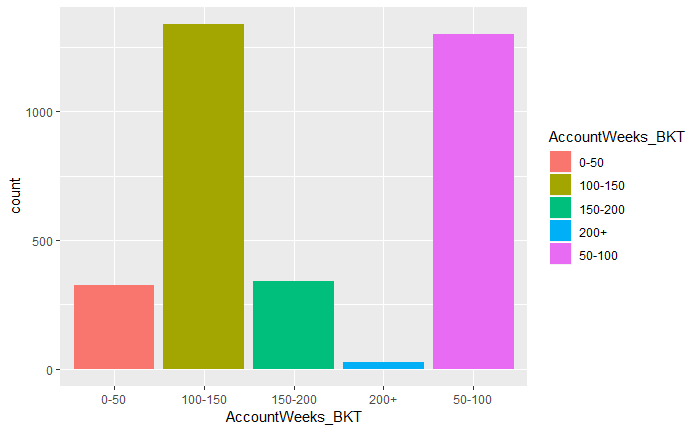
1. **AccountWeeks**

* Is numeric variable
* Average customers have active service from 100+ weeks

Min. 1st Qu. Median Mean 3rd Qu. Max.

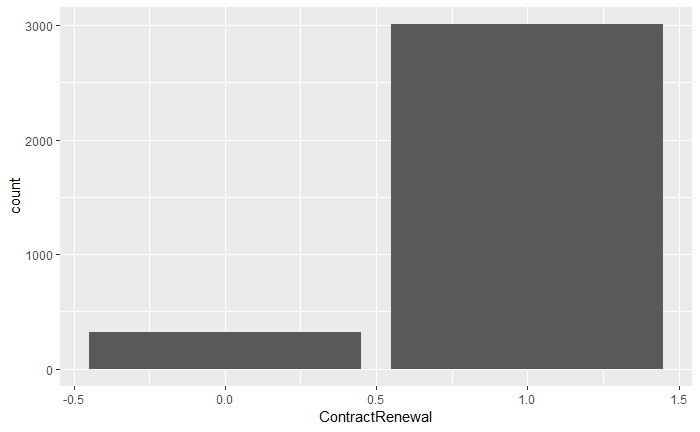
1.0 74.0 101.0 101.1 127.0 243.0

* Most of the customers in the data set have active accounts from 50 to 150 weeks



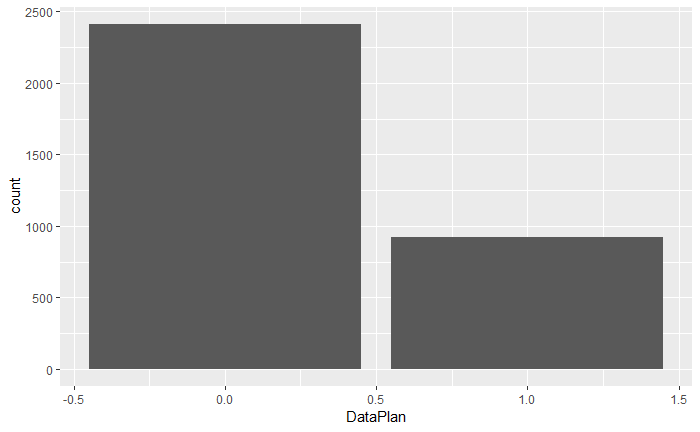
1. **ContractRenewal**

* Contract Renewal is Numeric variable
* Customer Renewal has 2 values 0 (not renewed subscription )and 1 (Renewed subscription )
* 90% of customers has done the contract renewal



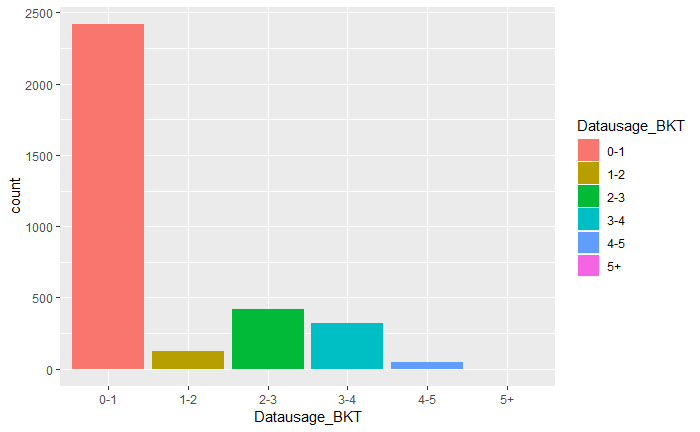
1. **DataPlan**

* Dataplan is numeric variable
* Around 30% of customers only have data plan
* DataPlan
* 0 1
* 2411 922



1. **Data Usage**

* Data usage is distributed from 0GB to 5.4 GB per month across customers
* Most of the customers usage is less than 1 GB



1. **Customer Service calls**

- Customer Service calls is numeric variable

unique(CustServCalls)

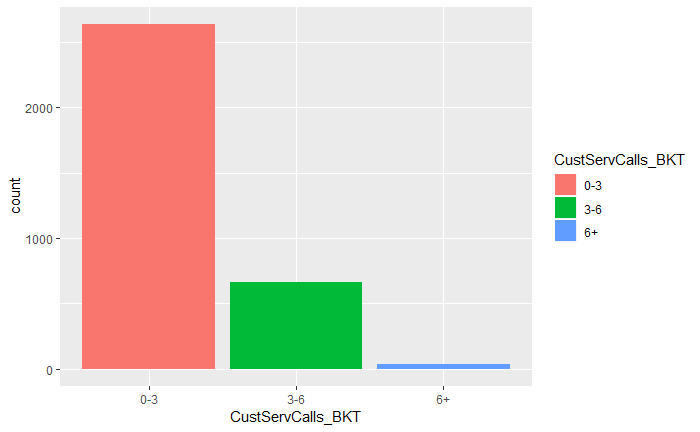
[1] 1 0 2 3 4 5 7 9 6 8

CustServCalls

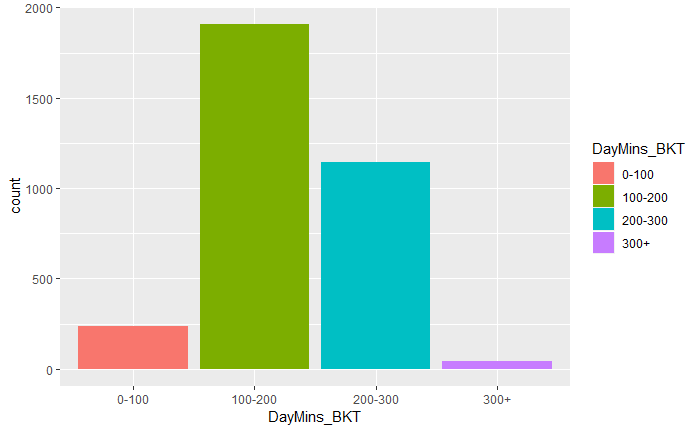
0 1 2 3 4 5 6 7 8 9

697 1181 759 429 166 66 22 9 2 2

- Most of the customers have not called or called only 3 times to customer care



1. DayMins

* Shows number of minutes in day customer was on call
* Majority of customers use between 100-300minutes in a day
* 

1. **DayCalls**

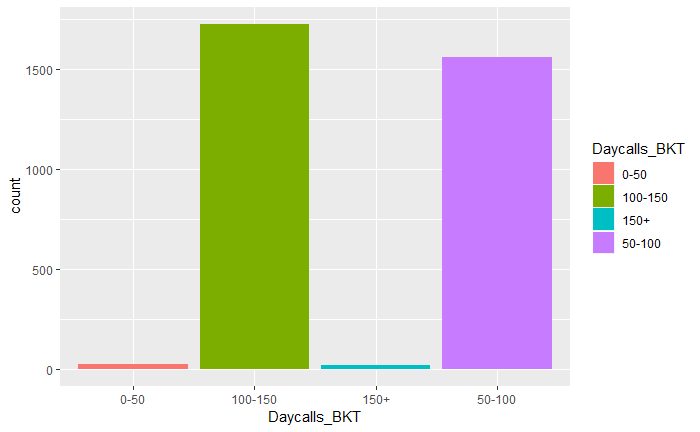
* Mean of minute in a day is 180

summary(DayMins)

Min. 1st Qu. Median Mean 3rd Qu. Max.

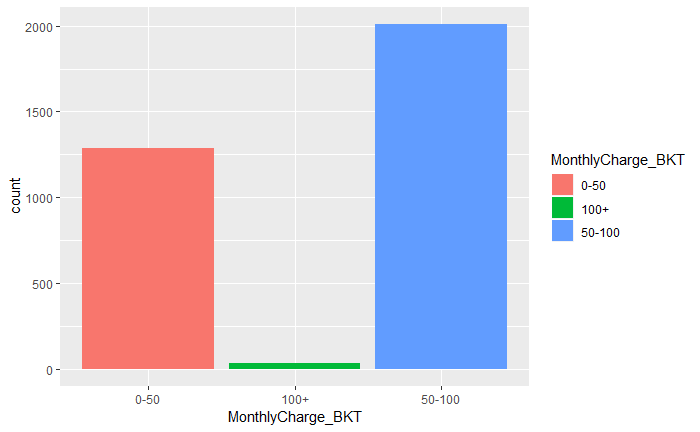
0.0 143.7 179.4 179.8 216.4 350.8

* Most of the customers have calls between 50-150 In a day



1. **MonthlyCharge**

* Mean monthly charge is around $56
* summary(MonthlyCharge)
* Min. 1st Qu. Median Mean 3rd Qu. Max.
* 14.00 45.00 53.50 56.31 66.20 111.30
* Most of the customers have monthly charge between 0 to 100 .



1. **OverageFee**

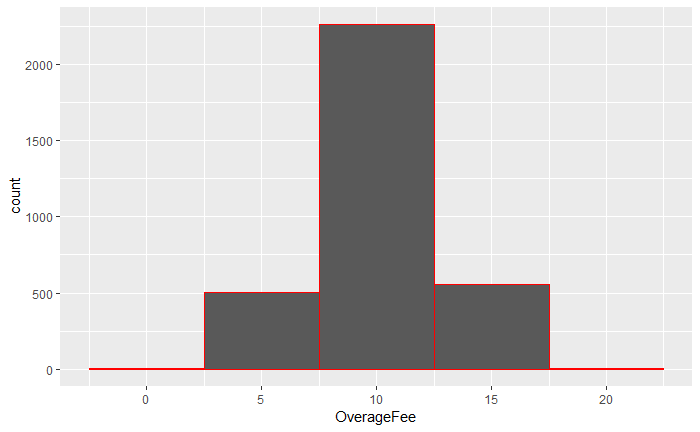
* Mean overage charge is $10

summary(OverageFee)

Min. 1st Qu. Median Mean 3rd Qu. Max.

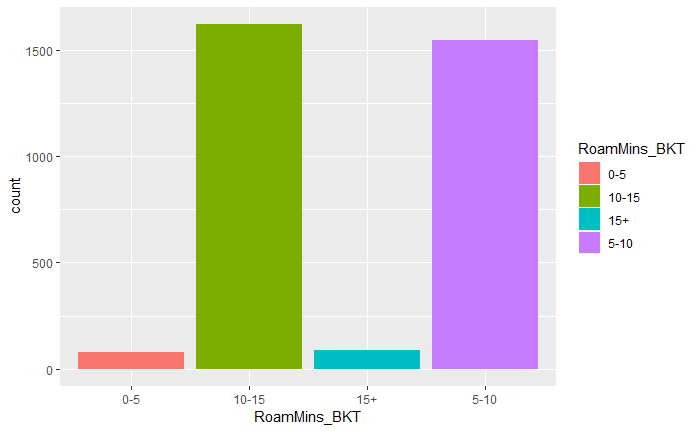
0.00 8.33 10.07 10.05 11.77 18.19

* Most of the customers are paying overage fees in bracket 6-12 per month



1. **Roaming** Mins

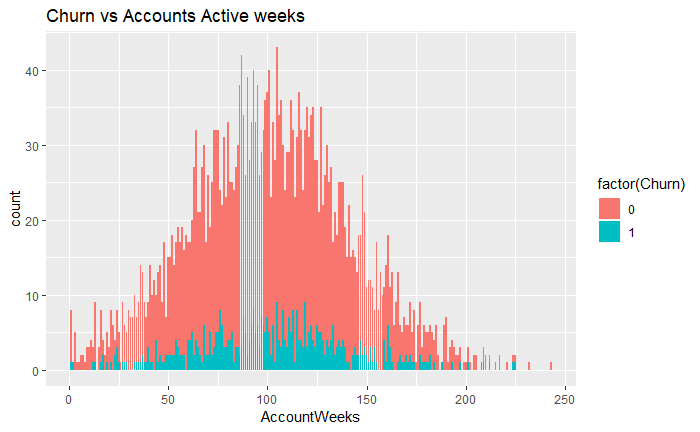
* 90% of customers have roaming mins between 5 to 15min.



## Bi-Variate/Multivariate Analysis

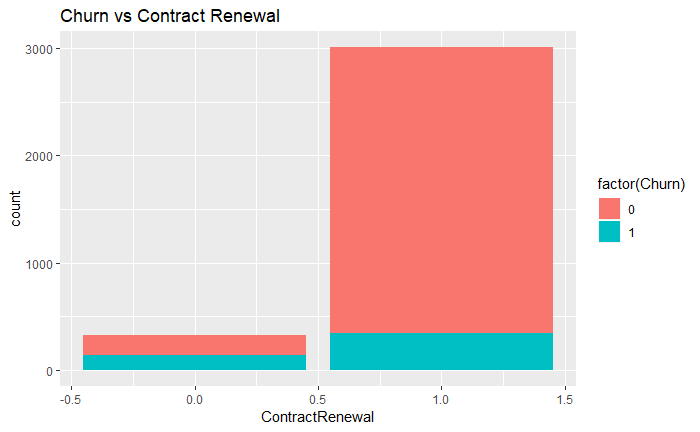
1. Churn and Accounts week

* Churn is spread across all weeks and does not dependent on number of weeks subscription is active .



1. Churn and Contract renewal.

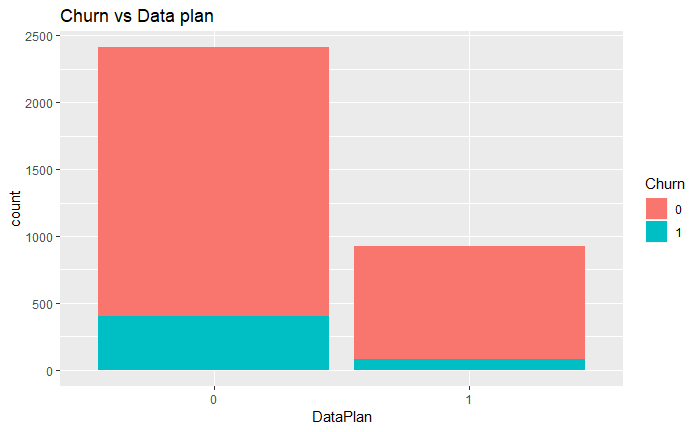
* As per data the customers who have done contract renewal are less probable for churn .



1. Churn and data plan

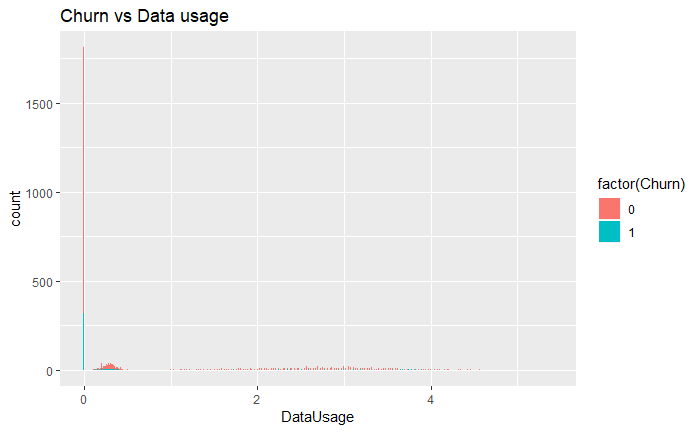
Customer churn with data plan is double then customer who don’t have data plan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DataPlan** | **Churn** | **n** | **% Churn** |  |
| <fctr> | <fctr> | <int> |
| 0 | 0 | 2008 | 0.1672 |  |
| 0 | 1 | 403 |  |  |
| 1 | 0 | 842 | 0.0868 |  |
| 1 | 1 | 80 |  |  |



1. **Churn and Data usage**

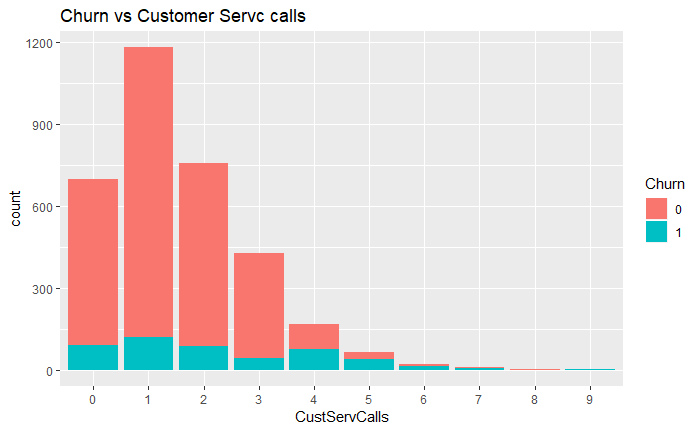
There is no definite relationship between churn and data usage but seems customer with high data usage are seems to be less probable to churn



1. Churn and Customer Service calls

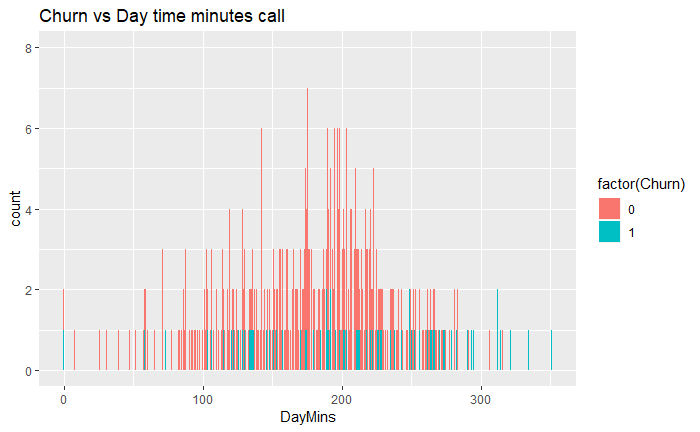
As the number of calls to customer Service are increasing the percentage of customer getting churned is increasing .

|  |  |  |  |
| --- | --- | --- | --- |
| **CustServCalls** | **Churn** | **n** | % churn |
| <fctr> | <fctr> | <int> |  |
| 0 | 0 | 605 |  |
| 0 | 1 | 92 | 0.131994 |
| 1 | 0 | 1059 |  |
| 1 | 1 | 122 | 0.103302 |
| 2 | 0 | 672 |  |
| 2 | 1 | 87 | 0.114625 |
| 3 | 0 | 385 |  |
| 3 | 1 | 44 | 0.102564 |
| 4 | 0 | 90 |  |
| 4 | 1 | 76 | 0.457831 |
| 5 | 0 | 26 |  |
| 5 | 1 | 40 | 0.606061 |
| 6 | 0 | 8 |  |
| 6 | 1 | 14 | 0.636364 |
| 7 | 0 | 4 |  |
| 7 | 1 | 5 | 0.555556 |
| 8 | 0 | 1 |  |
| 8 | 1 | 1 | 0.5 |
| 9 | 1 | 2 | 1 |



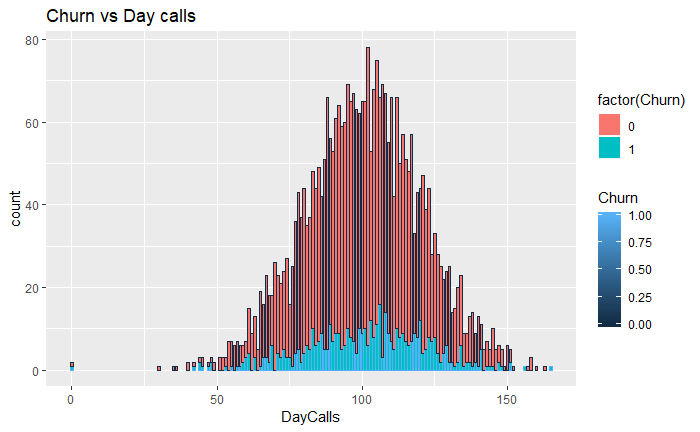
1. Churn and Day time call

Customer churn is more in high usage day time minutes



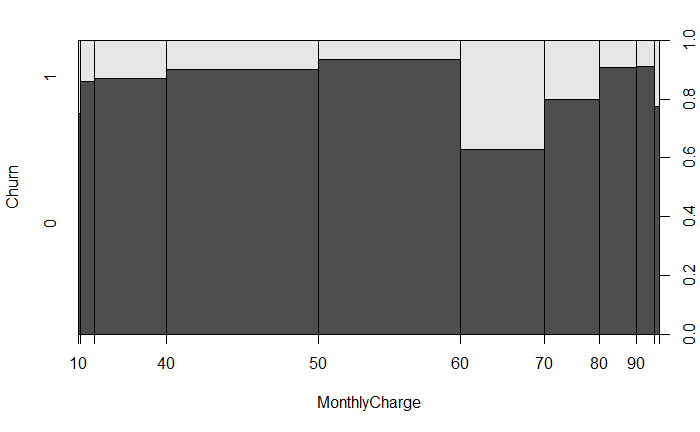
1. Churn and Day calls

Churn is distributed across calls and have similar trend across number of calls



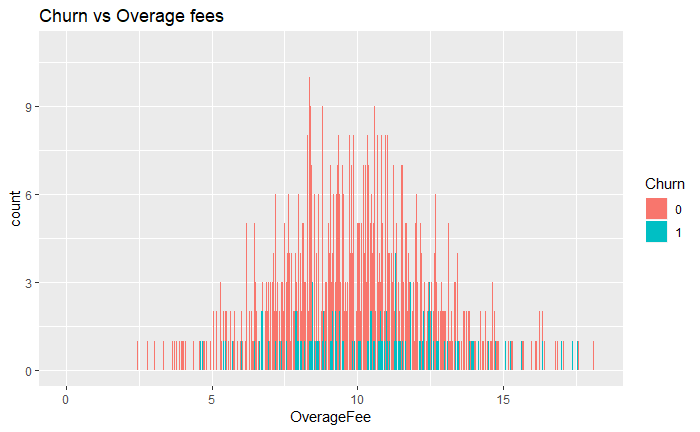
1. Churn and Monthly charge<pending >

There are more churns in the range of 60-70 monthly charge .



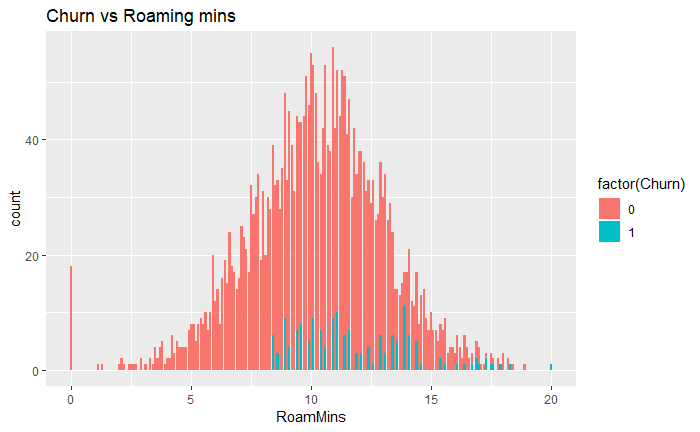
1. Churn and overage fees

There is slight increase in Churn percentage as overage fees keeps on increasing



1. Churn and Roaming min

There is slight increase in churn as roaming minutes keep on increasing .



# Model Creation

As dependent variable Churn is categorical variable with 2 classes, hence its classic case of binomial classification problem and logistic regression can be used to predict the model.

Fist divide the data into training and test sample .

Train sample will have 70% of churn data and test sample will have 30% of data .

> table(CustChurn$Churn )

0 1

2850 483

> table(dev\_sample$Churn)

0 1

1994 338

> table(test\_sample$Churn)

0 1

856 145

For code refer appendix section

## **Model 1**

Now running logistic regression model-**Model 1** on train sample with all the predictive variable

model1 = glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )

>

> summary(model1)

Call:

glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +

DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

OverageFee + RoamMins, family = "binomial", data = dev\_sample)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9668 -0.5202 -0.3512 -0.2116 2.9745

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.895755 0.653676 -9.019 < 2e-16 \*\*\*

AccountWeeks 0.001117 0.001647 0.678 0.49760

ContractRenewal -1.935344 0.174340 -11.101 < 2e-16 \*\*\*

DataPlan -1.100625 0.646462 -1.703 0.08866 .

DataUsage 1.913574 2.306805 0.830 0.40680

CustServCalls 0.488415 0.046601 10.481 < 2e-16 \*\*\*

DayMins 0.045392 0.038951 1.165 0.24387

DayCalls 0.003432 0.003278 1.047 0.29514

MonthlyCharge -0.187026 0.228794 -0.817 0.41368

OverageFee 0.453788 0.389956 1.164 0.24455

RoamMins 0.066855 0.025675 2.604 0.00922 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1930.1 on 2331 degrees of freedom

Residual deviance: 1537.3 on 2321 degrees of freedom

AIC: 1559.3

Number of Fisher Scoring iterations: 5

**Checking multicollinearity with VIF**

> vif(model1)

AccountWeeks ContractRenewal DataPlan DataUsage

1.003991 1.056485 13.900207 1582.255665

CustServCalls DayMins DayCalls MonthlyCharge

1.085483 945.525019 1.009814 2824.386789

OverageFee RoamMins

206.703087 1.185983

Any value greater then 10 is not good for our model .

**Creating Confusion matrix**

confusionMatrix(churn\_pred,as.factor(churn\_act))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 835 119

1 21 26

Accuracy : 0.8601

95% CI : (0.8371, 0.881)

No Information Rate : 0.8551

P-Value [Acc > NIR] : 0.3464

Kappa : 0.2152

Mcnemar's Test P-Value : 2.444e-16

Sensitivity : 0.9755

Specificity : 0.1793

Pos Pred Value : 0.8753

Neg Pred Value : 0.5532

Prevalence : 0.8551

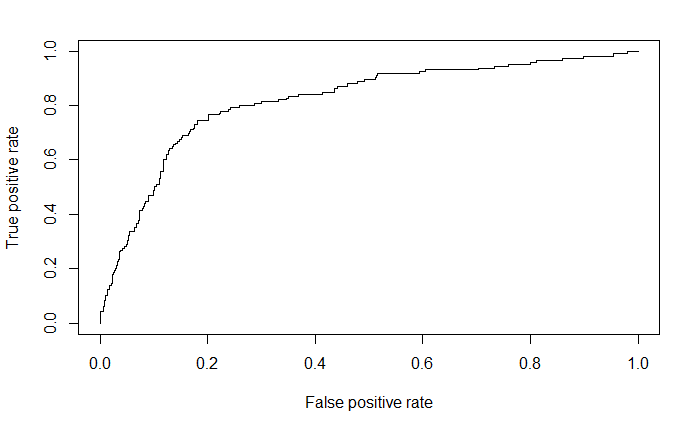
Detection Rate : 0.8342

Detection Prevalence : 0.9530

Balanced Accuracy : 0.5774

'Positive' Class : 0

Plot ROC to get the model’s accuracy



auc(test\_sample$Churn,pred1)

Area under the curve: 0.8176

## **Model 2**

Running logistic regression – **Model 2** by removing accountsweek as its not significant variable

model2 = glm(formula = Churn ~ ContractRenewal + DataPlan + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )

>

> summary(model2)

Call:

glm(formula = Churn ~ ContractRenewal + DataPlan + DataUsage +

CustServCalls + DayMins + DayCalls + MonthlyCharge + OverageFee +

RoamMins, family = "binomial", data = dev\_sample)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9696 -0.5152 -0.3541 -0.2108 2.9735

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.781490 0.630878 -9.164 <2e-16 \*\*\*

ContractRenewal -1.937053 0.174299 -11.113 <2e-16 \*\*\*

DataPlan -1.106444 0.646227 -1.712 0.0869 .

DataUsage 1.955342 2.305215 0.848 0.3963

CustServCalls 0.489284 0.046592 10.501 <2e-16 \*\*\*

DayMins 0.046077 0.038926 1.184 0.2365

DayCalls 0.003511 0.003274 1.073 0.2834

MonthlyCharge -0.190995 0.228641 -0.835 0.4035

OverageFee 0.460001 0.389721 1.180 0.2379

RoamMins 0.066629 0.025655 2.597 0.0094 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1930.1 on 2331 degrees of freedom

Residual deviance: 1537.8 on 2322 degrees of freedom

AIC: 1557.8

Number of Fisher Scoring iterations: 5

Checking multicollinearity for new model 2

vif(model2)

ContractRenewal DataPlan DataUsage CustServCalls

1.056622 13.885780 1579.677485 1.085563

DayMins DayCalls MonthlyCharge OverageFee

945.340012 1.008396 2819.916729 206.379352

RoamMins

1.185670

> confusionMatrix(churn\_pred,as.factor(churn\_act))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 834 120

1 22 25

Accuracy : 0.8581

95% CI : (0.835, 0.8792)

No Information Rate : 0.8551

P-Value [Acc > NIR] : 0.4151

Kappa : 0.204

Mcnemar's Test P-Value : 3.951e-16

Sensitivity : 0.9743

Specificity : 0.1724

Pos Pred Value : 0.8742

Neg Pred Value : 0.5319

Prevalence : 0.8551

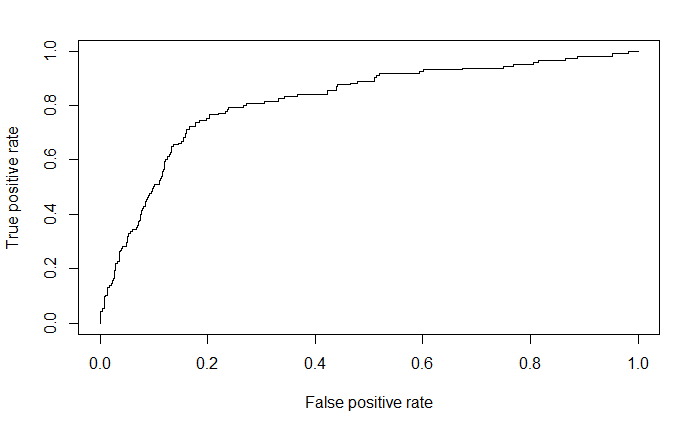
Detection Rate : 0.8332

Detection Prevalence : 0.9530

Balanced Accuracy : 0.5734

'Positive' Class : 0

Plot ROC to get the model’s accuracy



auc(test\_sample$Churn,pred2)

Area under the curve: 0.8176

## **Model 3**

Running logistic regression – **Model 3** by removing Dataplan as its not significant variable

model3 = glm(formula = Churn ~ ContractRenewal + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )

>

> summary(model3)

Call:

glm(formula = Churn ~ ContractRenewal + DataUsage + CustServCalls +

DayMins + DayCalls + MonthlyCharge + OverageFee + RoamMins,

family = "binomial", data = dev\_sample)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9852 -0.5191 -0.3524 -0.2157 2.8992

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.894484 0.627488 -9.394 < 2e-16 \*\*\*

ContractRenewal -1.948386 0.174204 -11.184 < 2e-16 \*\*\*

DataUsage 1.444688 2.286187 0.632 0.527439

CustServCalls 0.486528 0.046556 10.450 < 2e-16 \*\*\*

DayMins 0.043517 0.038897 1.119 0.263242

DayCalls 0.003329 0.003266 1.019 0.308148

MonthlyCharge -0.176478 0.228507 -0.772 0.439933

OverageFee 0.434748 0.389439 1.116 0.264275

RoamMins 0.083277 0.023971 3.474 0.000513 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1930.1 on 2331 degrees of freedom

Residual deviance: 1540.8 on 2323 degrees of freedom

AIC: 1558.8

Number of Fisher Scoring iterations: 5

Checking multicollinearity for new model 3

vif(model3)

ContractRenewal DataUsage CustServCalls DayMins

1.055011 1506.898522 1.082501 948.021057

DayCalls MonthlyCharge OverageFee RoamMins

1.007522 2775.111344 206.682070 1.022187

confusionMatrix(churn\_pred,as.factor(churn\_act))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 834 119

1 22 26

Accuracy : 0.8591

95% CI : (0.836, 0.8801)

No Information Rate : 0.8551

P-Value [Acc > NIR] : 0.3803

Kappa : 0.2127

Mcnemar's Test P-Value : 6.234e-16

Sensitivity : 0.9743

Specificity : 0.1793

Pos Pred Value : 0.8751

Neg Pred Value : 0.5417

Prevalence : 0.8551

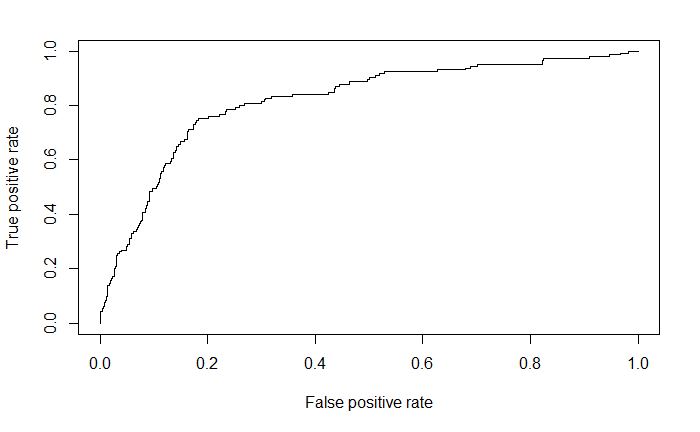
Detection Rate : 0.8332

Detection Prevalence : 0.9520

Balanced Accuracy : 0.5768

'Positive' Class : 0

Plot ROC to get the model’s accuracy



auc(test\_sample$Churn,pred3)

Area under the curve: 0.8177

>

## **Model 4**

Running Logistic Regression model – **Model 4 by removing data usage as its not significant variable**

model4 = glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )

>

> summary(model4)

Call:

glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins +

DayCalls + MonthlyCharge + OverageFee + RoamMins, family = "binomial",

data = dev\_sample)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9892 -0.5184 -0.3521 -0.2137 2.9093

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.959657 0.619501 -9.620 < 2e-16 \*\*\*

ContractRenewal -1.952574 0.174128 -11.213 < 2e-16 \*\*\*

CustServCalls 0.486941 0.046539 10.463 < 2e-16 \*\*\*

DayMins 0.018965 0.001687 11.240 < 2e-16 \*\*\*

DayCalls 0.003305 0.003264 1.013 0.311220

MonthlyCharge -0.032136 0.005971 -5.382 7.37e-08 \*\*\*

OverageFee 0.189421 0.029905 6.334 2.39e-10 \*\*\*

RoamMins 0.083632 0.023973 3.489 0.000486 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1930.1 on 2331 degrees of freedom

Residual deviance: 1541.2 on 2324 degrees of freedom

AIC: 1557.2

Number of Fisher Scoring iterations: 5

Checking multicollinearity for new model 4

> vif(model4)

ContractRenewal CustServCalls DayMins DayCalls

1.054128 1.082623 1.784465 1.007298

MonthlyCharge OverageFee RoamMins

1.894342 1.219067 1.021895

Creating Confusion matrix

> confusionMatrix(churn\_pred,as.factor(churn\_act))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 834 118

1 22 27

Accuracy : 0.8601

95% CI : (0.8371, 0.881)

No Information Rate : 0.8551

P-Value [Acc > NIR] : 0.3464

Kappa : 0.2214

Mcnemar's Test P-Value : 9.83e-16

Sensitivity : 0.9743

Specificity : 0.1862

Pos Pred Value : 0.8761

Neg Pred Value : 0.5510

Prevalence : 0.8551

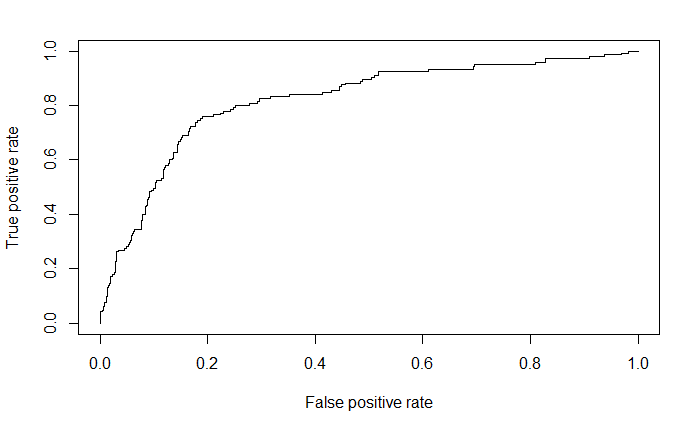
Detection Rate : 0.8332

Detection Prevalence : 0.9510

Balanced Accuracy : 0.5803

'Positive' Class : 0

Plot ROC to get the model’s accuracy



auc(test\_sample$Churn,pred4)

Area under the curve: 0.8185

## **Model 5**

Running Logistic Regression model – **Model 5 by removing day calls as its not significant variable**

> model5 = glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )

>

> summary(model5)

Call:

glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins +

MonthlyCharge + OverageFee + RoamMins, family = "binomial",

data = dev\_sample)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9723 -0.5213 -0.3538 -0.2161 2.9359

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -5.617183 0.516196 -10.882 < 2e-16 \*\*\*

ContractRenewal -1.952575 0.174162 -11.211 < 2e-16 \*\*\*

CustServCalls 0.485537 0.046462 10.450 < 2e-16 \*\*\*

DayMins 0.019000 0.001687 11.263 < 2e-16 \*\*\*

MonthlyCharge -0.032181 0.005968 -5.392 6.95e-08 \*\*\*

OverageFee 0.187678 0.029852 6.287 3.24e-10 \*\*\*

RoamMins 0.084364 0.023968 3.520 0.000432 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1930.1 on 2331 degrees of freedom

Residual deviance: 1542.2 on 2325 degrees of freedom

AIC: 1556.2

Number of Fisher Scoring iterations: 5

Checking multicollinearity for new model 5

> vif(model5)

ContractRenewal CustServCalls DayMins MonthlyCharge

1.054101 1.080490 1.782955 1.893563

OverageFee RoamMins

1.213727 1.021074

Model 5 has all the predictor variable as significant and VIF value is also moderately low.

Predict the test sample data on the above model and creating confusion matrix between predicted and actual value in Churn variable to understand the model accuracy

> confusionMatrix(churn\_pred,as.factor(churn\_act))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 835 119

1 21 26

Accuracy : 0.8601

95% CI : (0.8371, 0.881)

No Information Rate : 0.8551

P-Value [Acc > NIR] : 0.3464

Kappa : 0.2152

Mcnemar's Test P-Value : 2.444e-16

Sensitivity : 0.9755

Specificity : 0.1793

Pos Pred Value : 0.8753

Neg Pred Value : 0.5532

Prevalence : 0.8551

Detection Rate : 0.8342

Detection Prevalence : 0.9530

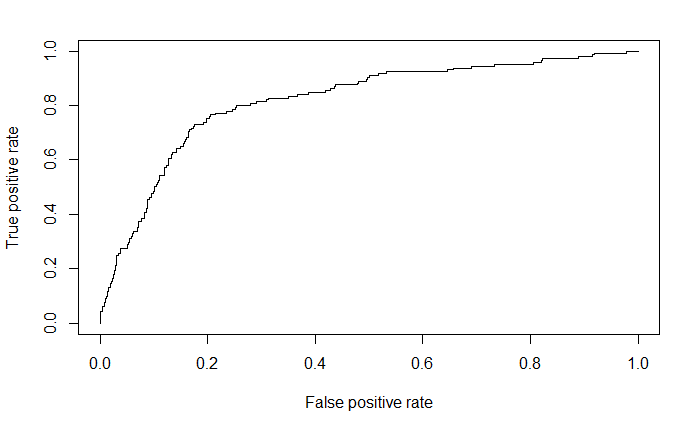
Balanced Accuracy : 0.5774

'Positive' Class : 0

Model 5 is able to predict 86% of data accurately .

AIC(Akaike information Criteria ) of model 5 suggest that its quality is better then other 4 models .

Plot ROC to get the model’s accuracy



> auc(test\_sample$Churn,pred5)

Area under the curve: 0.8178

# **Model Comparison**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Models | Model 1 | Model2 | Model3 | Model 4 | Model 5 |
| AIC | 1559.3 | 1557.8 | 1558.8 | 1557.2 | 1556.2 |
| Confusion  Matrix -accuracy | 86.01% | 85.81% | 85.91% | 86.01% | 86.01% |
| ROC – AUC | 81.76% | 81.76% | 81.77% | 81.85% | 81.78% |

**Conclusion**

Model 5 is the best model as compare to others it has highest area under the curve and prediction rate of 86.01% and with lowest AIC values.

# **Appendix**

Customer Churn

Saurabh Mudgal

16 May 2019

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#Set working directory  
  
setwd("C:\\BACP\\Module 5 - Predictive Modelling\\customer Churn")  
getwd()

## [1] "C:/BACP/Module 5 - Predictive Modelling/customer Churn"

#libraries ######  
library(knitr)  
library(rmarkdown)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.3

library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

library("car")

## Warning: package 'car' was built under R version 3.5.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.5.2

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

library("caret")

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.5.3

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.5.2

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

#install.packages("pROC")  
library("pROC")

## Warning: package 'pROC' was built under R version 3.5.3

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

##   
## Attaching package: 'pROC'

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

#read data from excel  
CustChurn = read.csv("Cellphone.csv",2)  
attach(CustChurn)  
  
##Exploratory data analysis   
  
  
#Names of the columns  
names(CustChurn)

## [1] "Churn" "AccountWeeks" "ContractRenewal"  
## [4] "DataPlan" "DataUsage" "CustServCalls"   
## [7] "DayMins" "DayCalls" "MonthlyCharge"   
## [10] "OverageFee" "RoamMins"

#total number of rows and columns  
dim(CustChurn)

## [1] 3333 11

#class of each feature   
str(CustChurn)

## 'data.frame': 3333 obs. of 11 variables:  
## $ Churn : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ AccountWeeks : int 128 107 137 84 75 118 121 147 117 141 ...  
## $ ContractRenewal: int 1 1 1 0 0 0 1 0 1 0 ...  
## $ DataPlan : int 1 1 0 0 0 0 1 0 0 1 ...  
## $ DataUsage : num 2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...  
## $ CustServCalls : int 1 1 0 2 3 0 3 0 1 0 ...  
## $ DayMins : num 265 162 243 299 167 ...  
## $ DayCalls : int 110 123 114 71 113 98 88 79 97 84 ...  
## $ MonthlyCharge : num 89 82 52 57 41 57 87.3 36 63.9 93.2 ...  
## $ OverageFee : num 9.87 9.78 6.06 3.1 7.42 ...  
## $ RoamMins : num 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...

#top 6 rows   
head(CustChurn)

## Churn AccountWeeks ContractRenewal DataPlan DataUsage CustServCalls  
## 1 0 128 1 1 2.7 1  
## 2 0 107 1 1 3.7 1  
## 3 0 137 1 0 0.0 0  
## 4 0 84 0 0 0.0 2  
## 5 0 75 0 0 0.0 3  
## 6 0 118 0 0 0.0 0  
## DayMins DayCalls MonthlyCharge OverageFee RoamMins  
## 1 265.1 110 89 9.87 10.0  
## 2 161.6 123 82 9.78 13.7  
## 3 243.4 114 52 6.06 12.2  
## 4 299.4 71 57 3.10 6.6  
## 5 166.7 113 41 7.42 10.1  
## 6 223.4 98 57 11.03 6.3

#missing value  
colSums(is.na(CustChurn))

## Churn AccountWeeks ContractRenewal DataPlan   
## 0 0 0 0   
## DataUsage CustServCalls DayMins DayCalls   
## 0 0 0 0   
## MonthlyCharge OverageFee RoamMins   
## 0 0 0

#univariate analysis   
  
#Churn  
unique(Churn)

## [1] 0 1

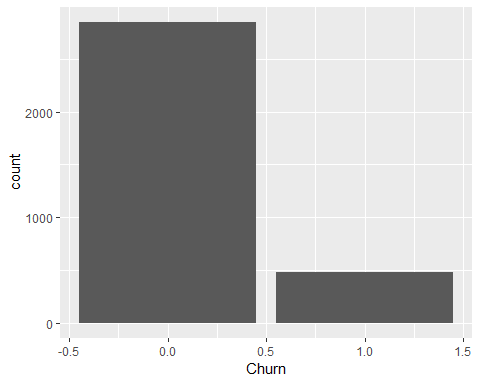
lables=c("not churn", "churn")  
table(Churn)

## Churn  
## 0 1   
## 2850 483

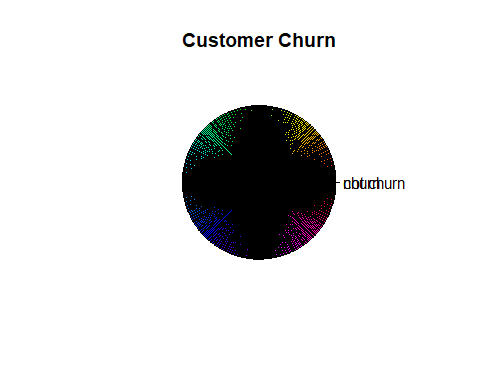
summary(Churn)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.1449 0.0000 1.0000

CustChurn %>% ggplot(aes(x=Churn))+  
 geom\_bar(aes(fill=Churn))



pie(as.numeric(Churn),lables,col = rainbow(length(Churn)),main="Customer Churn" )



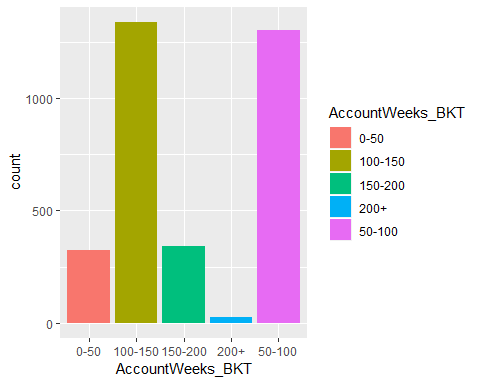
#Accounts active timeperiod  
  
summary(AccountWeeks)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 74.0 101.0 101.1 127.0 243.0

AccountWeeks\_BKT = ifelse(AccountWeeks < 50, "0-50",  
 ifelse(AccountWeeks < 100 , "50-100",  
 ifelse(AccountWeeks < 150, "100-150",  
 ifelse(AccountWeeks<200,"150-200","200+"  
 ))))  
summary(AccountWeeks\_BKT)

## Length Class Mode   
## 3333 character character

CustChurn %>% ggplot(aes(x=AccountWeeks\_BKT))+  
 geom\_bar(aes(fill=AccountWeeks\_BKT))



#Contracts renewal  
summary(ContractRenewal)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 1.0000 1.0000 0.9031 1.0000 1.0000

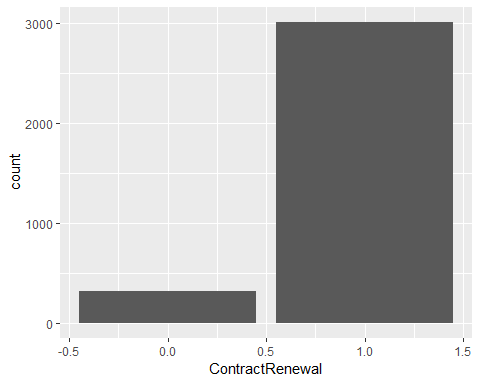
unique(ContractRenewal)

## [1] 1 0

table(ContractRenewal)

## ContractRenewal  
## 0 1   
## 323 3010

#converting to factor variable as contractrenewal has 2 values  
#CustChurn$ContractRenewal = as.factor(ContractRenewal)  
  
  
CustChurn %>% ggplot(aes(x=ContractRenewal))+  
 geom\_bar(aes(fill=ContractRenewal))



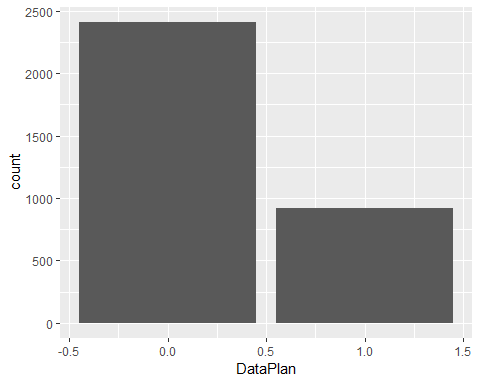
##Data plan  
  
unique(DataPlan)

## [1] 1 0

table(DataPlan)

## DataPlan  
## 0 1   
## 2411 922

#converting to factor variable as contractrenewal has 2 values  
#CustChurn$DataPlan = as.factor(DataPlan)  
  
CustChurn %>% ggplot(aes(x=DataPlan))+  
 geom\_bar(aes(fill=DataPlan))



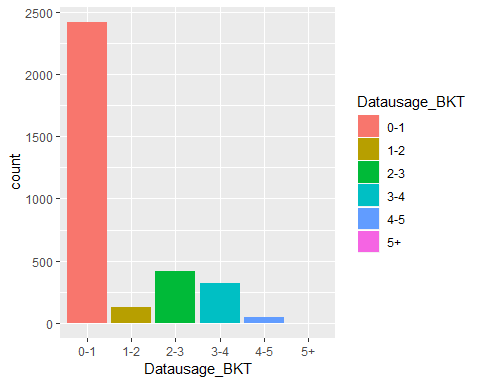
##Data usage  
summary(DataUsage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.8165 1.7800 5.4000

unique(DataUsage)

## [1] 2.70 3.70 0.00 2.03 0.19 3.02 0.29 0.34 0.44 3.73 0.31 0.39 2.57 0.32  
## [15] 0.21 2.24 3.97 3.92 2.84 0.30 3.94 0.38 0.24 2.30 0.35 2.97 0.11 1.57  
## [29] 3.13 3.40 1.67 3.27 0.26 2.65 2.32 2.94 0.17 1.19 2.43 2.73 5.40 1.94  
## [43] 0.40 1.73 1.43 2.75 1.84 3.00 2.59 3.46 2.86 0.14 0.23 2.67 0.27 0.22  
## [57] 0.33 3.54 2.89 3.29 3.59 3.21 0.25 3.19 3.75 0.28 3.81 1.81 0.51 3.08  
## [71] 2.27 1.22 2.16 1.76 0.42 0.13 4.21 2.78 0.20 2.62 2.08 2.05 1.35 2.54  
## [85] 3.48 1.92 3.78 0.18 2.48 3.24 0.43 1.00 0.45 0.36 4.32 2.51 3.05 3.32  
## [99] 1.70 3.86 0.16 1.62 4.64 4.00 1.59 1.78 4.73 2.11 1.46 3.83 2.46 4.16  
## [113] 4.46 3.56 1.97 2.38 1.11 1.30 0.37 4.19 4.40 0.15 2.81 2.00 4.43 3.43  
## [127] 3.62 2.21 4.05 4.10 3.67 4.29 3.35 2.19 4.35 3.16 2.40 2.35 1.49 4.56  
## [141] 1.38 0.47 2.92 4.24 3.65 1.86 3.51 1.05 2.13 0.46 0.41 3.89 3.38 3.11  
## [155] 1.51 1.54 1.89 1.13 0.12 4.08 1.03 1.65 4.75 1.32 4.13 1.27 1.16 4.59  
## [169] 4.02 4.48 0.78 1.40 0.65 0.68

Datausage\_BKT = ifelse(DataUsage < 1, "0-1",  
 ifelse(DataUsage < 2 , "1-2",  
 ifelse(DataUsage <3, "2-3",  
 ifelse(DataUsage < 4,"3-4",  
 ifelse(DataUsage < 5,"4-5",   
 "5+"  
 )))))  
  
CustChurn %>% ggplot(aes(x=Datausage\_BKT))+  
 geom\_bar(aes(fill=Datausage\_BKT))



##Cust service calls   
summary(CustServCalls)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 1.000 1.000 1.563 2.000 9.000

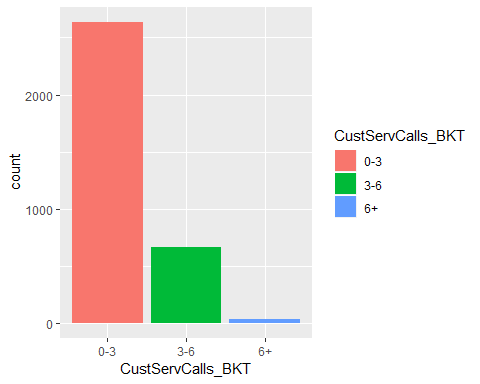
unique(CustServCalls)

## [1] 1 0 2 3 4 5 7 9 6 8

table(CustServCalls)

## CustServCalls  
## 0 1 2 3 4 5 6 7 8 9   
## 697 1181 759 429 166 66 22 9 2 2

#CustChurn$CustServCalls = as.factor(CustServCalls)  
  
CustServCalls\_BKT = ifelse( CustServCalls < 3, "0-3",  
 ifelse(CustServCalls < 6 , "3-6",  
 "6+"  
 ))  
  
CustChurn %>% ggplot(aes(x=CustServCalls\_BKT))+  
 geom\_bar(aes(fill=CustServCalls\_BKT))



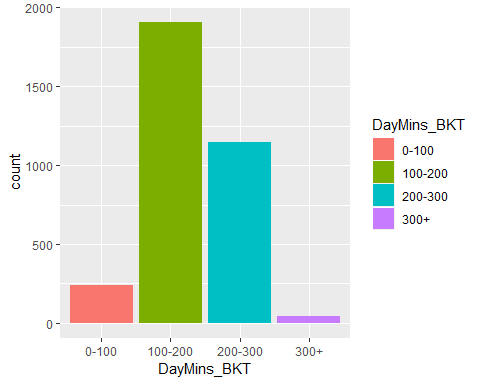
##Average mins per day calls  
summary(DayMins)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 143.7 179.4 179.8 216.4 350.8

unique(DayMins)

## [1] 265.1 161.6 243.4 299.4 166.7 223.4 218.2 157.0 184.5 258.6 129.1  
## [12] 187.7 128.8 156.6 120.7 332.9 196.4 190.7 189.7 224.4 155.1 62.4  
## [23] 183.0 110.4 81.1 124.3 213.0 134.3 190.0 119.3 84.8 226.1 212.0  
## [34] 249.6 176.8 220.0 146.3 130.8 203.9 140.4 126.3 173.1 124.8 85.8  
## [45] 154.0 120.9 211.3 187.0 159.1 133.2 191.9 220.6 186.1 160.2 151.0  
## [56] 175.5 126.9 198.4 148.8 229.3 192.1 268.6 193.7 180.7 131.2 148.1  
## [67] 251.5 125.2 211.6 178.9 241.8 224.9 248.6 203.4 235.8 157.1 300.3  
## [78] 61.6 214.1 170.2 201.1 215.4 165.6 249.5 210.6 179.3 157.9 214.3  
## [89] 154.1 237.9 143.9 252.9 179.1 278.4 160.1 198.2 212.1 251.8 161.2  
## [100] 178.3 151.7 135.0 170.5 238.1 281.4 117.9 148.6 229.8 165.0 185.0  
## [111] 161.0 126.7 58.9 196.8 162.6 282.5 113.7 239.8 210.2 213.8 170.9  
## [122] 154.2 201.4 70.7 187.5 91.7 214.2 145.5 166.3 231.0 200.3 197.0  
## [133] 129.9 175.8 203.1 183.2 205.0 148.5 192.6 246.5 167.1 231.9 146.7  
## [144] 271.5 181.5 257.7 193.8 102.8 187.9 226.0 260.4 178.7 337.4 157.6  
## [155] 183.6 142.1 136.3 217.1 98.9 206.3 243.1 189.8 202.0 170.1 230.9  
## [166] 237.1 182.1 116.8 219.2 252.6 147.1 202.1 173.5 232.1 197.1 58.2  
## [177] 115.6 259.9 158.7 271.6 160.6 232.4 133.8 176.9 209.9 137.5 289.5  
## [188] 198.1 149.7 326.5 292.9 83.0 145.7 182.3 218.0 140.6 152.7 106.7  
## [199] 243.8 194.4 213.9 217.2 241.1 203.5 155.2 167.6 226.7 151.4 180.0  
## [210] 250.2 223.0 166.0 136.1 149.3 65.4 213.4 206.9 186.2 280.2 196.6  
## [221] 312.0 199.0 168.8 134.4 202.6 74.5 83.6 192.2 220.2 135.1 253.4  
## [232] 225.0 198.5 110.3 60.0 214.8 181.8 157.4 207.9 207.0 119.0 143.7  
## [243] 165.9 138.6 84.7 62.6 164.9 134.5 143.3 168.3 262.4 206.2 225.8  
## [254] 138.3 94.4 160.0 206.6 134.7 214.4 192.8 151.1 221.4 218.9 192.7  
## [265] 204.4 172.3 211.7 221.6 197.9 147.5 206.4 205.9 207.6 303.9 230.6  
## [276] 99.5 177.1 172.7 204.2 85.7 215.5 171.7 266.6 170.4 158.0 92.0  
## [287] 234.0 272.1 296.4 227.2 248.7 236.3 205.6 94.1 60.4 121.0 117.8  
## [298] 223.5 176.3 138.7 86.3 58.8 68.7 239.2 198.3 205.2 272.6 128.3  
## [309] 169.6 201.3 214.7 169.2 194.1 233.8 225.1 183.9 221.8 64.6 154.6  
## [320] 260.2 155.9 107.0 182.5 220.1 152.2 236.2 166.1 244.6 134.2 150.1  
## [331] 257.1 124.4 141.7 230.0 162.3 350.8 193.3 78.2 83.4 195.6 201.8  
## [342] 164.8 179.2 214.0 205.7 165.5 221.0 242.1 151.6 176.2 196.0 159.5  
## [353] 230.2 210.5 102.0 126.0 168.4 105.6 206.5 229.6 278.3 234.4 167.3  
## [364] 221.1 145.8 222.8 183.4 264.3 146.0 127.3 178.8 97.2 259.8 256.5  
## [375] 169.5 239.7 171.5 239.9 142.3 184.1 203.8 248.8 192.9 122.4 104.9  
## [386] 173.2 119.4 250.3 155.0 288.7 240.4 190.3 278.0 153.5 273.4 155.3  
## [397] 133.1 246.8 165.4 59.5 286.7 117.3 127.9 225.5 149.0 198.9 256.4  
## [408] 264.8 98.2 159.8 190.6 184.0 261.8 147.9 106.4 133.7 193.5 178.2  
## [419] 226.2 70.9 240.3 75.0 69.1 96.6 214.6 258.1 149.8 190.4 181.4  
## [430] 155.7 149.9 222.3 149.4 242.9 150.4 208.9 130.7 119.6 273.6 156.1  
## [441] 177.5 175.2 114.3 251.4 216.9 159.3 143.1 186.6 170.8 124.0 172.8  
## [452] 217.4 265.9 93.6 168.2 202.9 261.4 73.3 253.7 45.0 231.3 47.4  
## [463] 227.4 40.9 68.5 163.5 163.0 213.7 310.4 48.4 171.2 166.5 216.6  
## [474] 107.8 141.3 237.5 234.5 103.1 129.5 279.8 136.8 100.1 224.5 288.1  
## [485] 148.7 194.6 194.5 174.1 131.8 146.8 200.7 145.6 229.4 211.0 121.5  
## [496] 216.0 293.0 74.3 62.3 228.6 228.1 309.9 201.9 183.8 186.7 209.4  
## [507] 223.2 164.2 150.5 234.2 55.3 89.7 80.2 125.7 207.2 157.5 160.4  
## [518] 159.0 102.6 159.7 202.8 57.5 169.9 335.5 139.5 187.8 146.2 231.8  
## [529] 156.4 220.7 172.0 128.2 130.2 195.4 293.3 191.3 209.6 215.7 161.4  
## [540] 144.2 256.2 112.7 299.5 194.8 100.8 82.5 146.4 177.9 150.7 180.1  
## [551] 265.3 128.6 161.5 165.3 195.0 205.5 235.6 192.0 261.7 235.5 263.8  
## [562] 175.6 242.5 138.1 264.7 282.3 211.2 205.3 252.0 231.2 200.1 266.7  
## [573] 118.1 175.3 125.1 241.9 241.2 222.4 189.5 123.1 256.7 30.9 187.4  
## [584] 315.6 277.5 147.2 185.8 155.4 97.6 206.0 216.8 103.3 139.4 191.2  
## [595] 221.7 62.9 215.6 94.7 203.2 195.3 114.4 175.9 249.9 210.7 87.2  
## [606] 137.4 224.8 261.2 196.5 271.2 300.4 57.1 162.1 145.0 34.0 193.4  
## [617] 191.7 161.3 150.6 184.6 121.1 109.6 167.5 115.8 276.6 179.4 187.3  
## [628] 201.2 189.6 186.8 187.6 244.9 187.1 170.7 161.1 169.4 254.4 127.7  
## [639] 219.1 273.5 161.9 241.7 62.8 281.1 228.2 209.8 265.6 214.9 110.5  
## [650] 137.8 112.8 180.4 153.7 261.3 246.2 191.0 208.3 253.0 202.3 174.4  
## [661] 127.1 143.5 186.9 194.0 234.8 123.7 173.9 130.9 314.6 227.9 95.5  
## [672] 185.3 105.8 178.0 172.1 169.3 119.1 194.2 198.8 167.7 202.2 322.5  
## [683] 216.2 76.4 72.7 210.4 127.2 219.5 99.3 224.7 176.6 283.9 180.6  
## [694] 125.9 237.6 274.3 199.6 217.7 212.7 256.3 267.9 163.6 180.9 105.0  
## [705] 271.4 206.7 166.8 204.9 127.0 267.4 281.0 270.8 124.1 162.8 254.8  
## [716] 254.9 107.7 158.8 182.9 178.4 110.9 166.9 244.8 120.8 215.9 140.1  
## [727] 139.8 321.6 166.6 260.0 190.2 82.2 163.8 267.8 287.3 101.2 109.1  
## [738] 110.1 111.0 144.8 135.4 84.2 209.1 130.1 136.7 67.7 200.4 125.8  
## [749] 226.3 120.5 91.1 167.9 257.4 237.2 103.0 153.8 205.1 175.7 154.4  
## [760] 209.7 150.0 199.2 217.6 175.4 152.0 174.9 176.4 160.9 228.7 144.0  
## [771] 135.9 334.3 130.5 105.4 188.9 111.8 212.4 346.8 113.9 171.4 275.4  
## [782] 197.2 116.1 217.3 207.7 277.3 125.3 216.7 97.4 246.4 143.4 156.2  
## [793] 114.8 232.5 143.6 176.7 263.4 167.8 142.5 133.0 95.0 198.6 142.6  
## [804] 111.9 122.8 189.3 93.5 158.6 243.2 220.9 144.4 212.3 147.0 96.2  
## [815] 12.5 178.1 123.0 208.0 193.0 174.5 116.7 93.8 239.5 167.4 143.2  
## [826] 232.8 162.0 25.9 322.3 191.5 291.1 208.8 255.9 252.7 132.1 217.0  
## [837] 101.9 211.5 153.4 185.2 104.6 245.2 274.4 98.4 279.9 187.2 276.2  
## [848] 217.8 190.5 179.9 235.9 144.6 189.0 101.0 165.1 189.1 131.5 166.4  
## [859] 87.7 35.1 246.6 78.5 251.6 270.3 177.3 262.2 173.6 106.6 209.5  
## [870] 95.4 131.6 112.2 172.5 194.3 307.1 118.2 155.5 125.6 199.3 222.2  
## [881] 92.8 193.2 113.2 166.2 207.8 245.4 287.1 192.3 141.9 220.5 156.0  
## [892] 235.1 188.4 247.8 221.2 118.5 83.5 183.3 236.8 134.0 191.4 174.8  
## [903] 275.2 174.0 107.9 221.3 141.1 178.6 139.0 181.6 84.9 217.9 270.9  
## [914] 243.0 150.9 219.9 168.0 256.8 182.8 117.6 145.4 169.1 186.4 76.1  
## [925] 260.8 211.8 162.7 121.7 67.4 229.7 176.0 247.7 115.4 139.6 217.5  
## [936] 196.3 253.2 98.0 249.4 129.6 87.6 203.6 213.6 266.3 115.0 270.5  
## [947] 61.9 189.2 171.6 78.6 200.9 185.1 254.3 185.4 197.8 153.1 96.8  
## [958] 247.0 321.3 243.7 236.9 148.2 254.7 284.4 0.0 151.8 141.4 285.7  
## [969] 58.4 90.4 147.7 302.7 169.7 124.2 132.9 245.0 89.5 186.0 223.9  
## [980] 179.5 112.0 245.7 142.8 202.4 236.1 51.9 81.3 115.7 157.2 269.7  
## [991] 132.0 82.6 125.5 82.3 183.1 165.7 176.1 177.6 83.2 235.0 105.7  
## [1002] 160.3 95.9 140.7 119.7 99.9 250.9 200.6 198.0 164.5 112.6 226.4  
## [1013] 152.6 274.9 195.7 204.3 222.5 174.3 219.7 144.9 236.5 109.5 81.6  
## [1024] 133.4 137.1 39.5 199.5 156.8 132.4 63.2 181.1 117.5 218.7 207.3  
## [1035] 273.3 266.1 104.7 168.6 174.7 87.0 204.5 226.5 234.1 133.3 133.9  
## [1046] 72.8 196.1 219.6 222.9 115.9 154.7 136.4 272.4 210.0 153.9 252.4  
## [1057] 152.4 237.4 197.3 199.1 233.7 207.1 139.7 177.2 169.8 159.9 115.1  
## [1068] 92.2 243.9 117.1 223.3 154.8 46.5 242.2 259.4 69.4 156.5 61.2  
## [1079] 102.3 219.4 137.2 271.1 103.4 52.2 235.7 103.7 148.4 191.1 218.5  
## [1090] 97.5 128.7 236.6 85.9 141.2 216.4 118.7 209.2 244.3 197.4 185.6  
## [1101] 137.6 273.9 242.3 254.1 115.5 132.7 236.4 179.8 119.2 224.0 19.5  
## [1112] 184.8 122.5 226.9 204.8 158.1 225.2 159.4 54.8 283.1 291.8 222.7  
## [1123] 68.4 273.0 225.3 283.2 131.4 105.9 247.3 193.1 241.0 134.8 163.1  
## [1134] 329.8 131.9 99.7 37.8 94.8 269.0 268.3 198.7 201.7 293.7 120.3  
## [1145] 278.5 273.8 131.1 197.7 129.7 200.0 152.8 268.4 188.5 170.6 75.3  
## [1156] 131.7 101.4 107.5 157.7 286.4 173.0 268.7 255.3 41.9 239.4 113.1  
## [1167] 182.7 202.7 190.8 132.6 122.0 91.5 153.6 244.7 172.4 157.3 135.8  
## [1178] 160.7 202.5 191.6 138.9 286.6 164.6 141.6 163.2 254.2 109.7 277.0  
## [1189] 139.2 121.6 270.4 203.3 215.1 301.7 152.3 208.7 190.1 54.2 230.3  
## [1200] 240.8 276.1 179.7 165.8 144.1 199.8 171.8 245.8 195.1 81.9 239.0  
## [1211] 152.5 295.4 272.7 236.7 111.4 153.0 218.8 205.4 225.4 275.8 142.9  
## [1222] 210.3 225.7 80.3 231.7 188.8 163.7 70.8 101.7 258.4 242.4 188.0  
## [1233] 250.6 260.1 281.3 128.5 172.2 230.4 129.3 238.0 211.1 212.8 191.8  
## [1244] 108.0 174.6 150.3 157.8 237.8 204.0 118.0 272.5 118.9 7.9 150.2  
## [1255] 144.5 220.8 216.3 266.0 96.7 82.7 204.6 253.1 130.0 261.9 240.2  
## [1266] 154.5 328.1 145.9 139.1 240.1 83.8 269.8 88.1 184.2 149.2 204.7  
## [1277] 213.2 269.6 140.2 136.2 88.5 215.3 269.2 160.5 228.9 224.2 148.3  
## [1288] 138.5 109.0 210.8 142.4 193.6 192.4 233.2 126.1 290.4 60.6 298.1  
## [1299] 222.1 136.6 289.8 260.9 196.2 172.9 249.8 268.8 106.1 27.0 196.7  
## [1310] 149.6 206.1 199.9 213.1 252.3 111.1 96.5 156.9 123.3 197.6 270.0  
## [1321] 47.7 182.6 231.5 251.3 177.4 47.8 121.8 101.1 92.3 168.9 146.5  
## [1332] 190.9 123.8 96.0 93.4 90.6 152.9 257.9 85.2 152.1 278.9 303.2  
## [1343] 238.9 92.7 137.3 203.7 88.8 137.9 224.3 268.1 267.1 37.7 239.3  
## [1354] 134.9 239.1 92.6 17.6 276.5 313.8 288.5 210.9 64.9 243.5 313.2  
## [1365] 75.8 195.9 171.9 228.4 227.1 271.7 245.5 245.3 134.1 92.4 159.6  
## [1376] 49.9 116.9 270.7 145.3 230.7 151.5 146.1 256.0 200.2 212.9 230.5  
## [1387] 253.5 98.8 129.0 86.0 193.9 109.4 195.5 264.4 141.5 133.5 273.2  
## [1398] 224.6 104.0 116.4 129.4 161.8 147.8 262.3 259.7 51.8 164.3 154.3  
## [1409] 122.9 297.9 125.4 247.5 294.9 126.8 143.0 91.9 86.5 109.9 326.3  
## [1420] 110.0 105.3 111.7 81.7 128.1 55.6 232.6 102.7 263.1 201.5 251.0  
## [1431] 109.8 279.5 173.4 96.1 109.2 183.5 227.8 305.2 197.5 181.3 262.9  
## [1442] 160.8 141.8 50.6 252.1 230.1 73.8 145.1 135.2 124.7 144.3 235.2  
## [1453] 218.6 265.2 184.7 115.3 113.0 294.2 71.2 94.9 215.8 139.9 153.2  
## [1464] 103.5 182.2 185.7 222.6 108.3 247.2 118.4 317.8 123.2 164.1 345.3  
## [1475] 264.9 208.4 162.4 281.2 232.9 234.7 240.0 298.4 111.6 207.5 258.8  
## [1486] 156.7 150.8 322.4 180.5 274.7 142.0 232.7 288.0 216.1 227.0 245.9  
## [1497] 257.3 301.5 233.9 99.6 116.2 140.5 277.9 308.6 242.6 213.5 240.7  
## [1508] 314.1 255.1 49.2 40.4 291.2 186.5 163.3 295.3 114.1 279.1 255.8  
## [1519] 169.0 2.6 181.9 274.0 229.9 139.3 7.8 140.0 237.3 51.5 116.0  
## [1530] 106.5 218.4 227.5 185.9 111.2 244.1 127.4 54.7 283.4 258.0 90.5  
## [1541] 54.0 264.0 220.4 221.9 263.7 61.3 324.7 129.2 257.2 124.6 175.1  
## [1552] 78.7 211.9 63.7 237.7 225.9 137.0 220.3 308.0 58.0 90.0 86.1  
## [1563] 296.0 146.6 260.5 124.5 77.6 138.8 164.0 90.7 108.6 89.8 279.3  
## [1574] 210.1 113.6 127.8 99.4 276.9 163.4 287.4 158.4 288.8 102.1 177.7  
## [1585] 250.8 233.3 180.2 234.9 233.5 135.7 122.2 153.3 122.3 259.3 194.9  
## [1596] 44.9 262.8 171.1 178.5 203.0 242.8 182.0 118.6 161.7 248.9 158.9  
## [1607] 93.3 18.9 130.6 94.2 189.4 222.0 271.8 100.0 305.1 72.5 105.2  
## [1618] 86.8 78.3 97.1 291.6 247.6 113.3 146.9 96.3 280.4 173.7 113.8  
## [1629] 184.4 223.8 143.8 29.9 276.7 181.2 247.4 107.2 294.7 306.2 238.8  
## [1640] 251.9 264.5 141.0 140.8 125.0 103.2 138.4 274.6 286.2 268.0 142.2  
## [1651] 97.8 266.9 289.1 180.3 295.0 240.9 107.3 238.4 51.1 227.7 78.1  
## [1662] 280.0 224.1 124.9 321.1 231.1 180.8

DayMins\_BKT = ifelse(DayMins < 100, "0-100",  
 ifelse(DayMins < 200 , "100-200",  
 ifelse(DayMins <300, "200-300",  
 "300+"  
 )))  
  
CustChurn %>% ggplot(aes(x=DayMins\_BKT))+  
 geom\_bar(aes(fill=DayMins\_BKT))



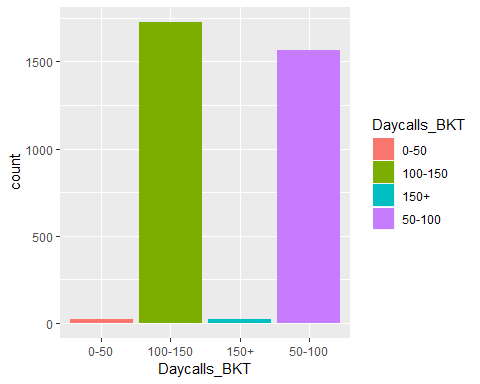
## number of calls in a day   
  
summary(DayCalls)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 87.0 101.0 100.4 114.0 165.0

unique(DayCalls)

## [1] 110 123 114 71 113 98 88 79 97 84 137 127 96 70 67 139 66  
## [18] 90 117 89 112 103 86 76 115 73 109 95 105 121 118 94 80 128  
## [35] 64 106 102 85 82 77 120 133 135 108 57 83 129 91 92 74 93  
## [52] 101 146 72 99 104 125 61 100 87 131 65 124 119 52 68 107 47  
## [69] 116 151 126 122 111 145 78 136 140 148 81 55 69 158 134 130 63  
## [86] 53 75 141 163 59 132 138 54 58 62 144 143 147 36 40 150 56  
## [103] 51 165 30 48 60 42 0 45 160 149 152 142 156 35 49 157 44

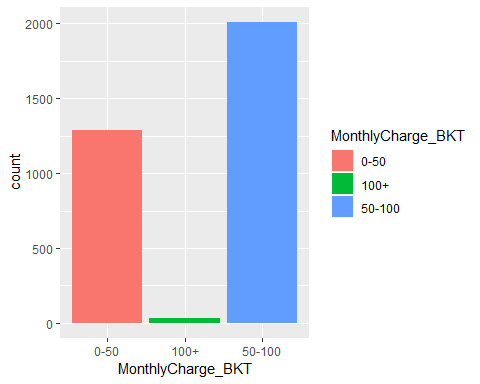
Daycalls\_BKT = ifelse(DayCalls < 50, "0-50",  
 ifelse(DayCalls < 100 , "50-100",  
 ifelse(DayCalls <150, "100-150",  
 "150+"  
 )))  
  
  
CustChurn %>% ggplot(aes(x=Daycalls\_BKT))+  
 geom\_bar(aes(fill=Daycalls\_BKT))



## monthly bill  
summary(MonthlyCharge)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 14.00 45.00 53.50 56.31 66.20 111.30

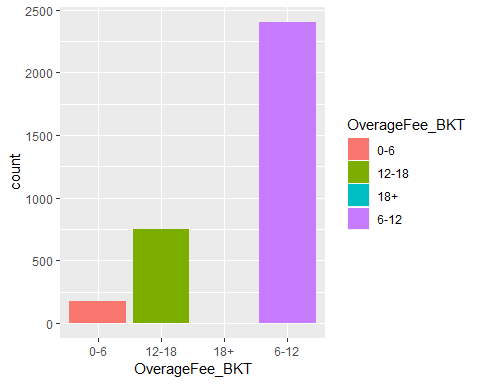
MonthlyCharge\_BKT = ifelse(MonthlyCharge < 50, "0-50",  
 ifelse(MonthlyCharge < 100 , "50-100",  
 "100+"  
 ))  
CustChurn %>% ggplot(aes(x=MonthlyCharge\_BKT))+  
 geom\_bar(aes(fill=MonthlyCharge\_BKT))



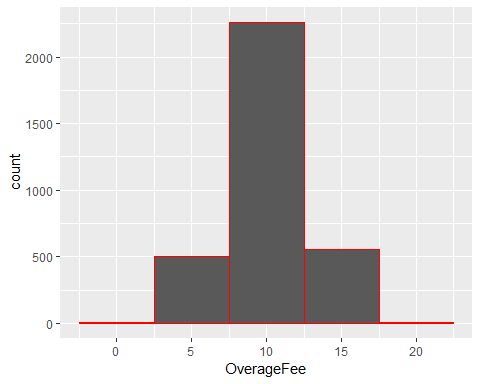
## overage charges  
  
summary(OverageFee)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 8.33 10.07 10.05 11.77 18.19

OverageFee\_BKT = ifelse(OverageFee < 6, "0-6",  
 ifelse(OverageFee < 12 , "6-12",  
 ifelse(OverageFee < 18 , "12-18",  
 "18+"  
 )))  
   
CustChurn %>% ggplot(aes(x=OverageFee\_BKT))+  
 geom\_bar(aes(fill=OverageFee\_BKT))



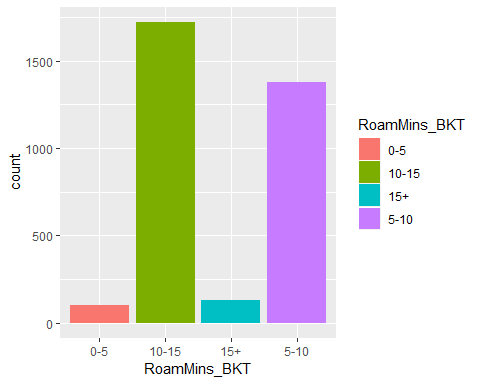
ggplot(CustChurn, aes(x=OverageFee)) +  
 geom\_histogram(binwidth = 5, colour='red' )



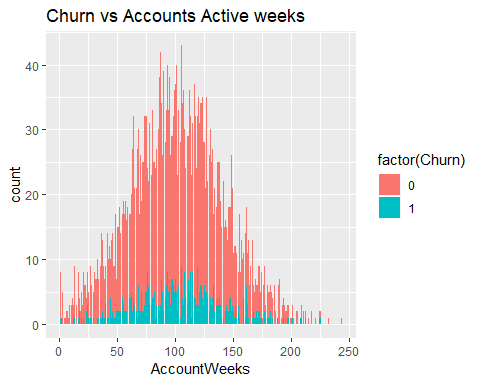
## Roaming mins   
  
  
summary(RoamMins)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 8.50 10.30 10.24 12.10 20.00

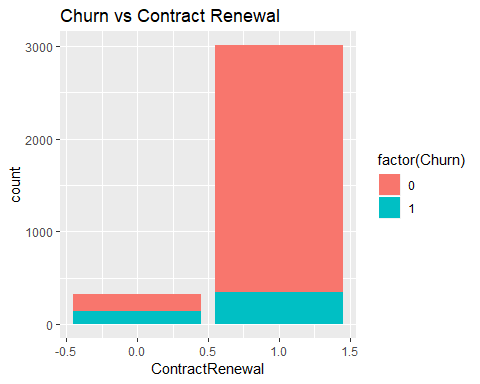
RoamMins\_BKT = ifelse(RoamMins < 5,"0-5",  
 ifelse(RoamMins < 10 , "5-10",  
 ifelse(RoamMins < 15 , "10-15",  
 "15+"  
 )))  
   
CustChurn %>% ggplot(aes(x=RoamMins\_BKT))+  
 geom\_bar(aes(fill=RoamMins\_BKT))



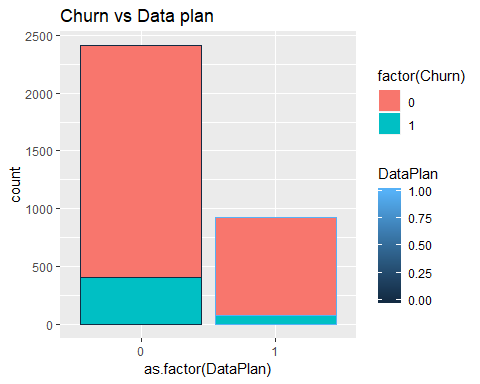
#ggplot(CustChurn, aes(x=RoamMins)) +  
 # geom\_histogram(binwidth = 4, colour='red' )  
  
  
### Bi-variate analysis   
  
##Churn and Accounts week  
#ggplot(CustChurn,aes(x=AccountWeeks, fill=Churn)) + geom\_bar() +  
 # labs(title = "Churn vs ActiveAcconts")  
  
ggplot(CustChurn,aes(x=AccountWeeks, fill=factor(Churn),colour=AccountWeeks)) + geom\_bar() +labs(title = "Churn vs Accounts Active weeks ")



#plot(Churn~AccountWeeks)  
  
## Churn and COntract renewal  
  
ggplot(CustChurn,aes(x=ContractRenewal, fill=factor(Churn),colour=ContractRenewal)) + geom\_bar() +labs(title = "Churn vs Contract Renewal")



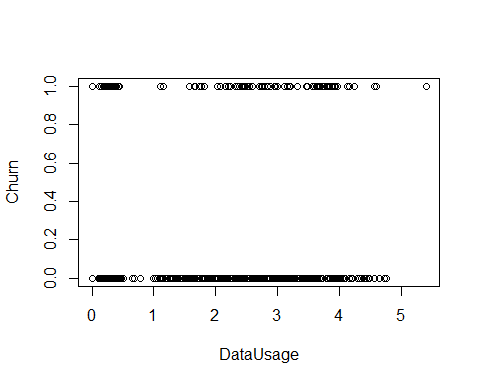
# (Churn~ContractRenewal)  
## Churn and data plan  
  
ggplot(CustChurn,aes(x=as.factor(DataPlan) , fill=factor(Churn),colour=DataPlan)) + geom\_bar() +labs(title = "Churn vs Data plan")



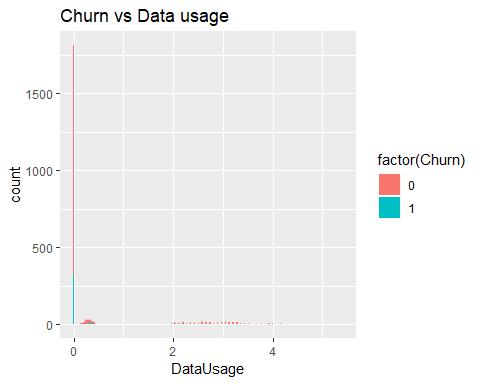
CustChurn %>%  
 group\_by(DataPlan,Churn) %>% tally()

## # A tibble: 4 x 3  
## # Groups: DataPlan [?]  
## DataPlan Churn n  
## <int> <int> <int>  
## 1 0 0 2008  
## 2 0 1 403  
## 3 1 0 842  
## 4 1 1 80

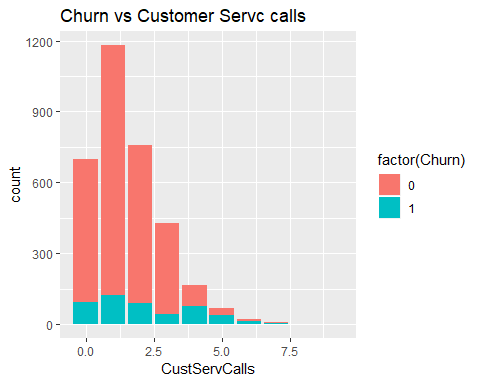
## Churn and data usage  
  
plot(Churn ~ DataUsage,CustChurn)



ggplot(CustChurn,aes(x=DataUsage , fill=factor(Churn),colour=DataUsage)) + geom\_bar() +labs(title = "Churn vs Data usage")



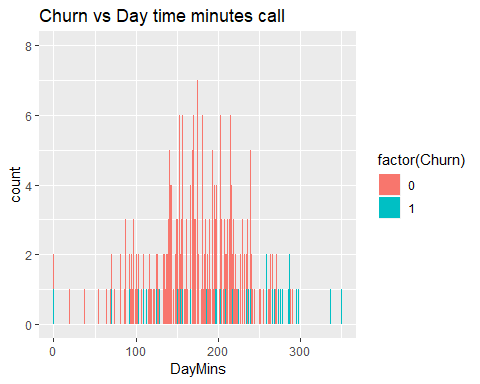
#CustChurn %>%  
 # group\_by(Churn,DataUsage) %>% tally()  
  
  
## Churn and cust service calls   
  
ggplot(CustChurn,aes(x=CustServCalls , fill=factor(Churn),colur=CustServCalls)) + geom\_bar() +labs(title = "Churn vs Customer Servc calls")



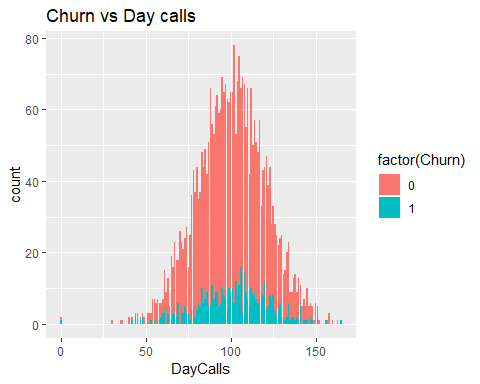
CustChurn %>%  
 group\_by(CustServCalls,Churn) %>% tally()

## # A tibble: 19 x 3  
## # Groups: CustServCalls [?]  
## CustServCalls Churn n  
## <int> <int> <int>  
## 1 0 0 605  
## 2 0 1 92  
## 3 1 0 1059  
## 4 1 1 122  
## 5 2 0 672  
## 6 2 1 87  
## 7 3 0 385  
## 8 3 1 44  
## 9 4 0 90  
## 10 4 1 76  
## 11 5 0 26  
## 12 5 1 40  
## 13 6 0 8  
## 14 6 1 14  
## 15 7 0 4  
## 16 7 1 5  
## 17 8 0 1  
## 18 8 1 1  
## 19 9 1 2

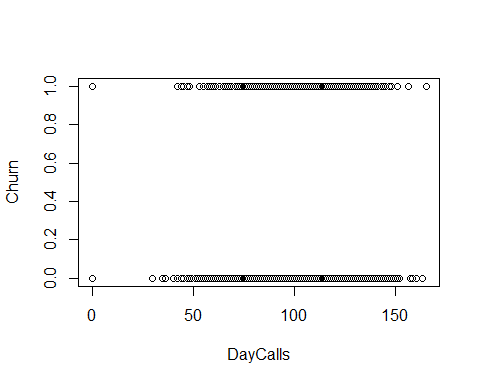
## Churn and dat time minutes calls   
  
ggplot(CustChurn,aes(x=DayMins , fill=factor(Churn),colour=DayMins)) + geom\_bar() +  
labs(title = "Churn vs Day time minutes call")



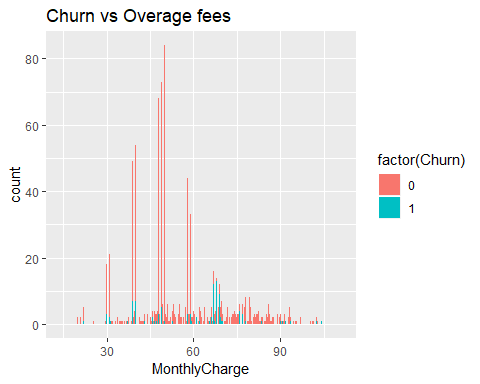
#plot(Churn~DayMins,CustChurn)  
  
#CustChurn %>%  
# group\_by(DayMins,Churn) %>% tally()  
## Churn and day calls   
  
ggplot(CustChurn,aes(x=DayCalls , fill=factor(Churn),colour=DayCalls)) + geom\_bar() +labs(title = "Churn vs Day calls")



plot(Churn~DayCalls,CustChurn)

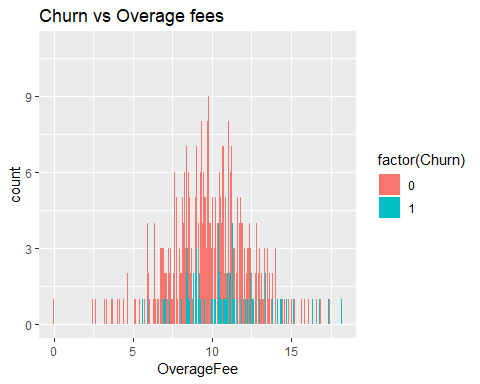


## Churn and monthcharge   
  
ggplot(CustChurn,aes(x=MonthlyCharge, fill=factor(Churn),colour=MonthlyCharge)) + geom\_bar() +labs(title = "Churn vs Overage fees")



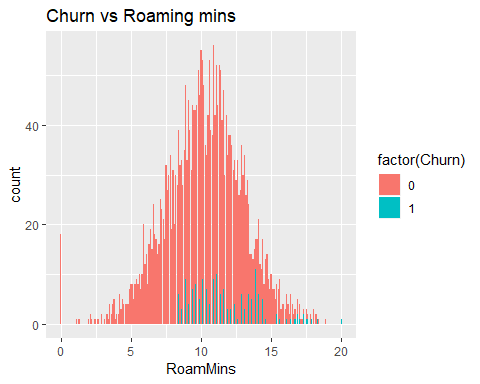
#plot(Churn~MonthlyCharge,CustChurn)  
  
  
## Churn and overage fees   
  
ggplot(CustChurn,aes(x=OverageFee , fill=factor(Churn),colour=OverageFee)) + geom\_bar() +labs(title = "Churn vs Overage fees")

## Warning: position\_stack requires non-overlapping x intervals

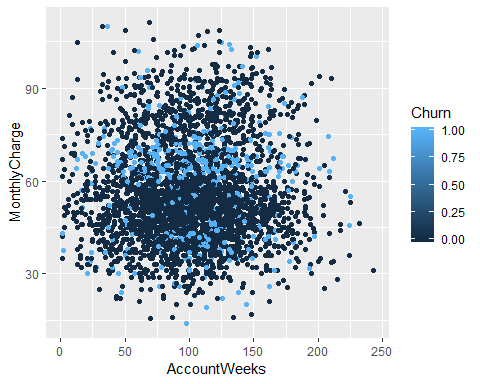


#plot(Churn~OverageFee,CustChurn)  
  
  
## Churn and roaming mins  
  
ggplot(CustChurn,aes(x=RoamMins , fill=factor(Churn),colour=RoamMins)) + geom\_bar() + labs(title = "Churn vs Roaming mins")

## Warning: position\_stack requires non-overlapping x intervals



## plot for Overage fees and Month charge and Churm  
  
qplot(AccountWeeks,MonthlyCharge,colour=Churn ,data=CustChurn)



#plot(Churn~RoamMins,CustChurn)  
  
## Dividind the data into test and training data set   
  
set.seed(123)  
  
 ind = sample(2,nrow(CustChurn),replace = TRUE,prob = c(.7,.3))  
 dev\_sample = CustChurn[ind==1,]   
 test\_sample = CustChurn[ind==2,]  
 table(CustChurn$Churn )

##   
## 0 1   
## 2850 483

table(dev\_sample$Churn)

##   
## 0 1   
## 1994 338

table(test\_sample$Churn)

##   
## 0 1   
## 856 145

# Run the logistic model   
   
   
   
 model1 = glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )  
  
 summary(model1)

##   
## Call:  
## glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +   
## DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +   
## OverageFee + RoamMins, family = "binomial", data = dev\_sample)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9668 -0.5202 -0.3512 -0.2116 2.9745   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.895755 0.653676 -9.019 < 2e-16 \*\*\*  
## AccountWeeks 0.001117 0.001647 0.678 0.49760   
## ContractRenewal -1.935344 0.174340 -11.101 < 2e-16 \*\*\*  
## DataPlan -1.100625 0.646462 -1.703 0.08866 .   
## DataUsage 1.913574 2.306805 0.830 0.40680   
## CustServCalls 0.488415 0.046601 10.481 < 2e-16 \*\*\*  
## DayMins 0.045392 0.038951 1.165 0.24387   
## DayCalls 0.003432 0.003278 1.047 0.29514   
## MonthlyCharge -0.187026 0.228794 -0.817 0.41368   
## OverageFee 0.453788 0.389956 1.164 0.24455   
## RoamMins 0.066855 0.025675 2.604 0.00922 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1930.1 on 2331 degrees of freedom  
## Residual deviance: 1537.3 on 2321 degrees of freedom  
## AIC: 1559.3  
##   
## Number of Fisher Scoring iterations: 5

vif(model1)

## AccountWeeks ContractRenewal DataPlan DataUsage   
## 1.003991 1.056485 13.900207 1582.255665   
## CustServCalls DayMins DayCalls MonthlyCharge   
## 1.085483 945.525019 1.009814 2824.386789   
## OverageFee RoamMins   
## 206.703087 1.185983

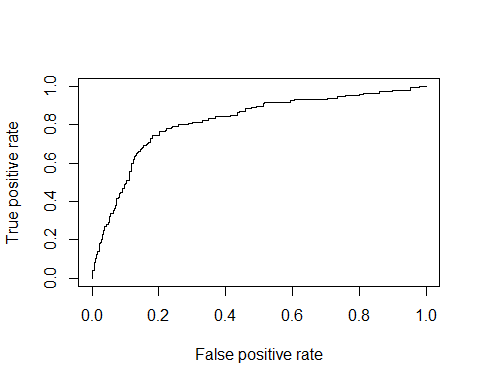
#Predict the test sample   
   
   
 pred1 = predict(model1,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred1 >.5,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
   
   
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 835 119  
## 1 21 26  
##   
## Accuracy : 0.8601   
## 95% CI : (0.8371, 0.881)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.3464   
##   
## Kappa : 0.2152   
##   
## Mcnemar's Test P-Value : 2.444e-16   
##   
## Sensitivity : 0.9755   
## Specificity : 0.1793   
## Pos Pred Value : 0.8753   
## Neg Pred Value : 0.5532   
## Prevalence : 0.8551   
## Detection Rate : 0.8342   
## Detection Prevalence : 0.9530   
## Balanced Accuracy : 0.5774   
##   
## 'Positive' Class : 0   
##

#CF$table[1,2]  
   
 mean(churn\_pred == churn\_act) ##accuracy of 86%%

## [1] 0.8601399

#check for ROC  
 pr = prediction(pred1,test\_sample$Churn)  
 perf = performance(pr,measure = "tpr",x.measure = "fpr")  
 plot(perf)



auc(test\_sample$Churn,pred1)

## Area under the curve: 0.8176

#pred2 = predict(model1,newdata = dev\_sample, type = "response")  
 #Churn\_pred\_num1 = ifelse(pred2 >.5,1,0)  
 #churn\_pred1 = factor(Churn\_pred\_num1,levels = c(0,1))   
 #churn\_act1 = dev\_sample$Churn  
 #confusionMatrix(churn\_pred1,churn\_act1)  
   
 # Run the logistic model by removing Accountweek as its insignificant   
 model2 = glm(formula = Churn ~ ContractRenewal + DataPlan + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )  
  
 summary(model2)

##   
## Call:  
## glm(formula = Churn ~ ContractRenewal + DataPlan + DataUsage +   
## CustServCalls + DayMins + DayCalls + MonthlyCharge + OverageFee +   
## RoamMins, family = "binomial", data = dev\_sample)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9696 -0.5152 -0.3541 -0.2108 2.9735   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.781490 0.630878 -9.164 <2e-16 \*\*\*  
## ContractRenewal -1.937053 0.174299 -11.113 <2e-16 \*\*\*  
## DataPlan -1.106444 0.646227 -1.712 0.0869 .   
## DataUsage 1.955342 2.305215 0.848 0.3963   
## CustServCalls 0.489284 0.046592 10.501 <2e-16 \*\*\*  
## DayMins 0.046077 0.038926 1.184 0.2365   
## DayCalls 0.003511 0.003274 1.073 0.2834   
## MonthlyCharge -0.190995 0.228641 -0.835 0.4035   
## OverageFee 0.460001 0.389721 1.180 0.2379   
## RoamMins 0.066629 0.025655 2.597 0.0094 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1930.1 on 2331 degrees of freedom  
## Residual deviance: 1537.8 on 2322 degrees of freedom  
## AIC: 1557.8  
##   
## Number of Fisher Scoring iterations: 5

vif(model2)

## ContractRenewal DataPlan DataUsage CustServCalls   
## 1.056622 13.885780 1579.677485 1.085563   
## DayMins DayCalls MonthlyCharge OverageFee   
## 945.340012 1.008396 2819.916729 206.379352   
## RoamMins   
## 1.185670

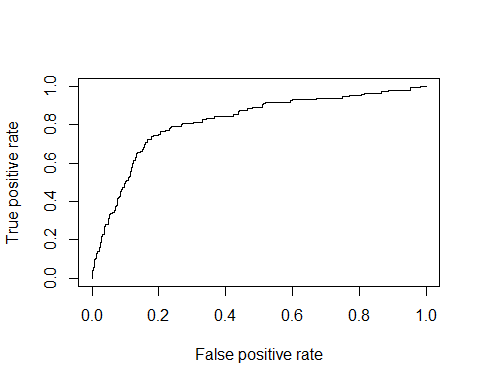
pred2 = predict(model2,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred2 >.5,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
   
   
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 834 120  
## 1 22 25  
##   
## Accuracy : 0.8581   
## 95% CI : (0.835, 0.8792)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.4151   
##   
## Kappa : 0.204   
##   
## Mcnemar's Test P-Value : 3.951e-16   
##   
## Sensitivity : 0.9743   
## Specificity : 0.1724   
## Pos Pred Value : 0.8742   
## Neg Pred Value : 0.5319   
## Prevalence : 0.8551   
## Detection Rate : 0.8332   
## Detection Prevalence : 0.9530   
## Balanced Accuracy : 0.5734   
##   
## 'Positive' Class : 0   
##

#CF$table[1,2]  
   
 mean(churn\_pred == churn\_act) ##accuracy of 86%%

## [1] 0.8581419

# checking ROC  
 pr = prediction(pred2,test\_sample$Churn)  
 perf = performance(pr,measure = "tpr",x.measure = "fpr")  
 plot(perf)



auc(test\_sample$Churn,pred2)

## Area under the curve: 0.8176

# Run the logistic model by removing Dataplan as its insignific  
 model3 = glm(formula = Churn ~ ContractRenewal + DataUsage + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )  
  
 summary(model3)

##   
## Call:  
## glm(formula = Churn ~ ContractRenewal + DataUsage + CustServCalls +   
## DayMins + DayCalls + MonthlyCharge + OverageFee + RoamMins,   
## family = "binomial", data = dev\_sample)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9852 -0.5191 -0.3524 -0.2157 2.8992   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.894484 0.627488 -9.394 < 2e-16 \*\*\*  
## ContractRenewal -1.948386 0.174204 -11.184 < 2e-16 \*\*\*  
## DataUsage 1.444688 2.286187 0.632 0.527439   
## CustServCalls 0.486528 0.046556 10.450 < 2e-16 \*\*\*  
## DayMins 0.043517 0.038897 1.119 0.263242   
## DayCalls 0.003329 0.003266 1.019 0.308148   
## MonthlyCharge -0.176478 0.228507 -0.772 0.439933   
## OverageFee 0.434748 0.389439 1.116 0.264275   
## RoamMins 0.083277 0.023971 3.474 0.000513 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1930.1 on 2331 degrees of freedom  
## Residual deviance: 1540.8 on 2323 degrees of freedom  
## AIC: 1558.8  
##   
## Number of Fisher Scoring iterations: 5

vif(model3)

## ContractRenewal DataUsage CustServCalls DayMins   
## 1.055011 1506.898522 1.082501 948.021057   
## DayCalls MonthlyCharge OverageFee RoamMins   
## 1.007522 2775.111344 206.682070 1.022187

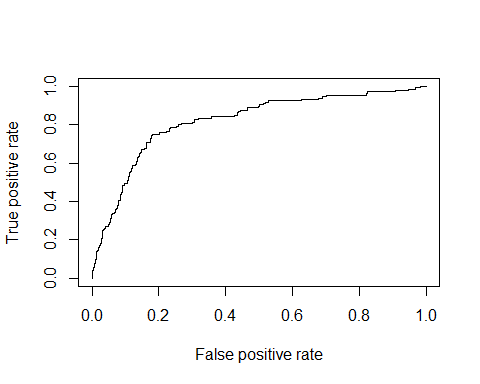
pred3 = predict(model3,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred3>.5,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
   
   
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 834 119  
## 1 22 26  
##   
## Accuracy : 0.8591   
## 95% CI : (0.836, 0.8801)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.3803   
##   
## Kappa : 0.2127   
##   
## Mcnemar's Test P-Value : 6.234e-16   
##   
## Sensitivity : 0.9743   
## Specificity : 0.1793   
## Pos Pred Value : 0.8751   
## Neg Pred Value : 0.5417   
## Prevalence : 0.8551   
## Detection Rate : 0.8332   
## Detection Prevalence : 0.9520   
## Balanced Accuracy : 0.5768   
##   
## 'Positive' Class : 0   
##

#CF$table[1,2]  
   
 mean(churn\_pred == churn\_act) ##accuracy of 86%%

## [1] 0.8591409

# checking ROC  
 pr = prediction(pred3,test\_sample$Churn)  
 perf = performance(pr,measure = "tpr",x.measure = "fpr")  
 plot(perf)



auc(test\_sample$Churn,pred3)

## Area under the curve: 0.8177

# Run the logistic model by removing Data usage as its insignific  
 model4 = glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins+DayCalls + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )  
  
 summary(model4)

##   
## Call:  
## glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins +   
## DayCalls + MonthlyCharge + OverageFee + RoamMins, family = "binomial",   
## data = dev\_sample)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9892 -0.5184 -0.3521 -0.2137 2.9093   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.959657 0.619501 -9.620 < 2e-16 \*\*\*  
## ContractRenewal -1.952574 0.174128 -11.213 < 2e-16 \*\*\*  
## CustServCalls 0.486941 0.046539 10.463 < 2e-16 \*\*\*  
## DayMins 0.018965 0.001687 11.240 < 2e-16 \*\*\*  
## DayCalls 0.003305 0.003264 1.013 0.311220   
## MonthlyCharge -0.032136 0.005971 -5.382 7.37e-08 \*\*\*  
## OverageFee 0.189421 0.029905 6.334 2.39e-10 \*\*\*  
## RoamMins 0.083632 0.023973 3.489 0.000486 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1930.1 on 2331 degrees of freedom  
## Residual deviance: 1541.2 on 2324 degrees of freedom  
## AIC: 1557.2  
##   
## Number of Fisher Scoring iterations: 5

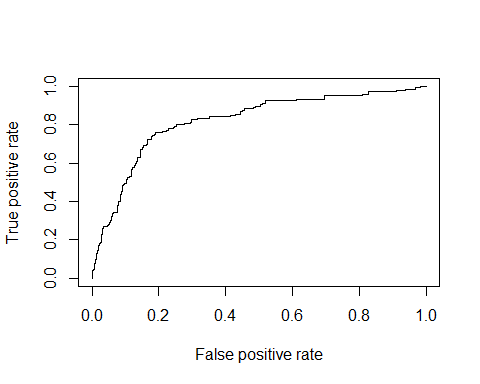
##calculate VIF for multicolinearity   
 vif(model4)

## ContractRenewal CustServCalls DayMins DayCalls   
## 1.054128 1.082623 1.784465 1.007298   
## MonthlyCharge OverageFee RoamMins   
## 1.894342 1.219067 1.021895

pred4 = predict(model4,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred4>.5,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
   
   
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 834 118  
## 1 22 27  
##   
## Accuracy : 0.8601   
## 95% CI : (0.8371, 0.881)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.3464   
##   
## Kappa : 0.2214   
##   
## Mcnemar's Test P-Value : 9.83e-16   
##   
## Sensitivity : 0.9743   
## Specificity : 0.1862   
## Pos Pred Value : 0.8761   
## Neg Pred Value : 0.5510   
## Prevalence : 0.8551   
## Detection Rate : 0.8332   
## Detection Prevalence : 0.9510   
## Balanced Accuracy : 0.5803   
##   
## 'Positive' Class : 0   
##

# checking ROC  
 pr = prediction(pred4,test\_sample$Churn)  
 perf = performance(pr,measure = "tpr",x.measure = "fpr")  
 plot(perf)



auc(test\_sample$Churn,pred4)

## Area under the curve: 0.8185

# Run the logistic model by removing day calls as its insignific  
 model5 = glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins + MonthlyCharge + OverageFee + RoamMins,family = "binomial" ,data=dev\_sample )  
  
 summary(model5)

##   
## Call:  
## glm(formula = Churn ~ ContractRenewal + CustServCalls + DayMins +   
## MonthlyCharge + OverageFee + RoamMins, family = "binomial",   
## data = dev\_sample)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9723 -0.5213 -0.3538 -0.2161 2.9359   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.617183 0.516196 -10.882 < 2e-16 \*\*\*  
## ContractRenewal -1.952575 0.174162 -11.211 < 2e-16 \*\*\*  
## CustServCalls 0.485537 0.046462 10.450 < 2e-16 \*\*\*  
## DayMins 0.019000 0.001687 11.263 < 2e-16 \*\*\*  
## MonthlyCharge -0.032181 0.005968 -5.392 6.95e-08 \*\*\*  
## OverageFee 0.187678 0.029852 6.287 3.24e-10 \*\*\*  
## RoamMins 0.084364 0.023968 3.520 0.000432 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1930.1 on 2331 degrees of freedom  
## Residual deviance: 1542.2 on 2325 degrees of freedom  
## AIC: 1556.2  
##   
## Number of Fisher Scoring iterations: 5

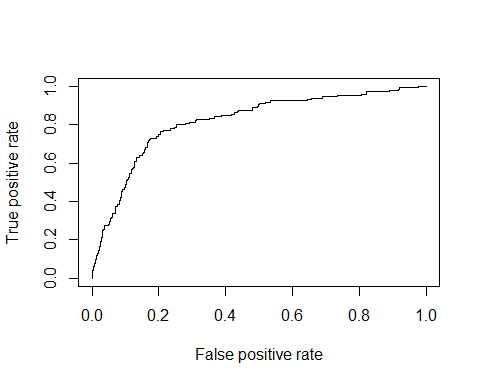
##calculate VIF for multicolinearity   
   
 vif(model5)

## ContractRenewal CustServCalls DayMins MonthlyCharge   
## 1.054101 1.080490 1.782955 1.893563   
## OverageFee RoamMins   
## 1.213727 1.021074

#checking with proabability of .6   
 pred5 = predict(model5,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred5 >.6,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 848 133  
## 1 8 12  
##   
## Accuracy : 0.8591   
## 95% CI : (0.836, 0.8801)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.3803   
##   
## Kappa : 0.1144   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.99065   
## Specificity : 0.08276   
## Pos Pred Value : 0.86442   
## Neg Pred Value : 0.60000   
## Prevalence : 0.85514   
## Detection Rate : 0.84715   
## Detection Prevalence : 0.98002   
## Balanced Accuracy : 0.53671   
##   
## 'Positive' Class : 0   
##

#check for ROC for model fit   
   
   
 pr = prediction(pred5,test\_sample$Churn)  
 perf = performance(pr,measure = "tpr",x.measure = "fpr")  
 plot(perf)



auc(test\_sample$Churn,pred5)

## Area under the curve: 0.8178

pred5 = predict(model5,newdata = test\_sample, type = "response")  
   
 Churn\_pred\_num = ifelse(pred1 > .65,1,0)  
 churn\_pred = factor(Churn\_pred\_num,levels = c(0,1))   
 churn\_act = test\_sample$Churn  
 confusionMatrix(churn\_pred,as.factor(churn\_act))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 849 134  
## 1 7 11  
##   
## Accuracy : 0.8591   
## 95% CI : (0.836, 0.8801)  
## No Information Rate : 0.8551   
## P-Value [Acc > NIR] : 0.3803   
##   
## Kappa : 0.1064   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.99182   
## Specificity : 0.07586   
## Pos Pred Value : 0.86368   
## Neg Pred Value : 0.61111   
## Prevalence : 0.85514   
## Detection Rate : 0.84815   
## Detection Prevalence : 0.98202   
## Balanced Accuracy : 0.53384   
##   
## 'Positive' Class : 0   
##

#CF$table[1,2]  
   
 mean(churn\_pred == churn\_act) ##accuracy of 86%%

## [1] 0.8591409

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.