churn-analysis-in-telecom-industry

May 2, 2024

1 Project:4 Churn Analysis in Telecom Industry

2 Problem Statement

Customer churn occurs when customers or subscribers stop doing business with a company or service, also known as customer attrition. It is also referred as loss of clients or customers. One industry in which churn rates are particularly useful is the telecommunications industry, because most customers have multiple options from which to choose within a geographic location.

We are going to build classification models using decision tree algorithm to predict whether the customer be churned or not on the basis of its billing information and customer demographics.

3 Data Description

[1]:
```
#Input Variables

#Gender: This column describes about customers gender details. Value of this␣
↪column is "Male" or "Female"

#SeniorCitizen: This column describes whether a customer is a senior citizen or␣
↪not. Value of this column is "Yes" or "No"

#Partner: This column describes whether a customer has a partner or not. Value␣
↪of this column is "Yes" or "No"

#Dependents: It describes whether a customer dependents or not. Value of this␣
↪column is "Yes" or "No"

#Tenure: This column describes how many months a customer got a telephone␣
↪connection. Value of this column is integer

#Call Service: This column describes whether a customer has a phone service␣
↪with their connection or not. Value of this column is "Yes" or "No"

#Multiple Connections: This column describes whether a customer has single or␣
↪multiple connection. Value of this column is "Yes" or "No"
```

```
#Internet Connection: This column describes whether a customer has an internet connection with their telephone connection or not.Value of this column is "Yes" or "No"

#Online Security: This column describes whether a customer has online security for internet connection or not. Value of this column is "Yes" or "No"

#Online Backup: Some customer wants to save their data in online storage. This column describes whether a customer has an online backup service or no. Value of this column is "Yes" or "No"

#Device Protection Service: Customer wants to protect their device from virus and other attack. So they might have device protection service which is got with telephone service. So this column describes whether a customer has device protection service or not.Value of this column is "Yes" or "No"

#Technical Help: If a customer wants technical help, they must have registered with their connection. So this column describes whether a customer has technical help service or not. Value of this column is "Yes" or "No"

#Online TV: Some customer might watch a TV program online. This column describes whether a customer is subscribed with online TV service or not. Value of this column is "Yes" or "No"

#Online Movies: Some customer might watch online movies. This column describes whether a customer is subscribed with online movies service or not. Value of this column is "Yes" or "No"

#Agreement: A customer might have a month to month agreement or one-year connection or two-year connection and so on. This column describes a customer agreement type

#Value of this column is "Month-to-Month", "One year" and " Two years".

#Billing Method: This column describes whether a customer wants paperless billing or not. Value of this column is "Yes" or "No"

#Payment Method: This column describes in which payment way a customer will pay like Bank transfer(automatic), Credit card(automatic), Electronic check and Mailed check

#Value of this column is Bank transfer(automatic), Credit card(automatic), Electronic check and Mailed check.Monthly 18 - Service Charge: This column describes how much will be charged from a customer for their connection. Value of this column is numeric
```

```
#Total Amount: This column describes how much a customer has to pay totally for␣
↪their connection. Value of this column is numeric

#Target Variable

#Churn: This column describes a customer is churn or not. Value of this column␣
↪is "Yes" or "No"
```

4 Model Selection

we select the best model. Model selection will be based on Accuracy, Sensitivity, Specificity, Positive predictive value, Negative predictive value, Prevalence, Detection rate, Detection prevalence and balanced accuracy.

5 Expected Outcome

Higher accuracy in predicting the outcome using test data.

6 Installing required packages

[2]:
```
#install.packages("plyr")

#install.packages("ggplot2")

#install.packages("caret")

#install.packages("MASS")

#install.packages("party")

#install.packages("RcolorBrewer")

#install.packages("ROCR")

#install.packages("rpart")

#install.packages("rattle")

#install.packages("rpart.plot")
```

7 Loading packages

[3]:
```
library(plyr)

library(ggplot2)

library(lattice)

library(caret)

library(MASS)

library(party)

library(RColorBrewer)

library(ROCR)

library(rpart)

library(rattle)

library(rpart.plot)
```

```
Attaching package: ‘caret’


The following object is masked from ‘package:httr’:

    progress


Loading required package: grid

Loading required package: mvtnorm

Loading required package: modeltools

Loading required package: stats4


Attaching package: ‘modeltools’


The following object is masked from ‘package:plyr’:
```

```
    empty


Loading required package: strucchange

Loading required package: zoo


Attaching package: ‘zoo’


The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric


Loading required package: sandwich

Loading required package: tibble

Loading required package: bitops

Rattle: A free graphical interface for data science with R.
Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
```

8 Loading Dataset

[4]:
```
df <- read.csv("../input/churn-analysis-in-telecom-industry-dataset/churn.csv")
```

[5]:
```
#Running head command to see first 6 rows

head(df)
```

A data.frame: 6 × 21

	customerID <fct>	gender <fct>	SeniorCitizen <dbl>	Partner <fct>	Dependents <fct>	tenure <dbl>	CallService <fct>
1	2907-ILJBN	Female	0	Yes	Yes	11	Yes
2	3896-RCYYE	Female	0	No	No	67	No
3	9764-REAFF	Female	0	Yes	No	59	Yes
4	6651-RLGGM	Male	0	Yes	Yes	67	Yes
5	5879-SESNB	Female	0	No	No	11	Yes
6	8670-MEFCP	Female	0	Yes	Yes	36	Yes

[6]:
```
#Running tail command to last 6 rows

tail(df)
```

A data.frame: 6 × 21

	customerID <fct>	gender <fct>	SeniorCitizen <dbl>	Partner <fct>	Dependents <fct>	tenure <dbl>	Call <fct
12330	6894-LFHLY	Female	0.07703193	Yes	No	7.460777	Yes
12331	6894-LFHLY	Male	0.35742330	No	Yes	2.927730	Yes
12332	0639-TSIQW	Female	0.00000000	No	No	63.430477	No
12333	0639-TSIQW	Male	0.00000000	No	Yes	49.677352	No
12334	0607-DAAHE	Male	0.66000182	Yes	Yes	20.659911	Yes
12335	1038-ZAGBI	Female	0.43385552	Yes	No	8.529156	Yes

[7]:
```r
#Running structure command

str(df)
```

```
'data.frame':	12335 obs. of  21 variables:
 $ customerID              : Factor w/ 5590 levels "0002-ORFBO","0004-TLHLJ",..:
1573 2155 5463 3755 3292 4852 5474 508 2771 4140 …
 $ gender                  : Factor w/ 2 levels "Female","Male": 1 1 1 2 1 1 1 1
2 2 …
 $ SeniorCitizen           : num  0 0 0 0 0 0 0 0 0 0 …
 $ Partner                 : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 2 1 1 1
…
 $ Dependents              : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 2 1 1 1
…
 $ tenure                  : num  11 67 59 67 11 36 49 54 26 19 …
 $ CallService             : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 2 2
…
 $ MultipleConnections     : Factor w/ 3 levels "No","No phone service",..: 1 2 1
3 3 3 1 1 3 1 …
 $ InternetConnection      : Factor w/ 3 levels "DSL","Fiber optic",..: 3 1 3 3 2
1 1 2 2 3 …
 $ OnlineSecurity          : Factor w/ 3 levels "No","No internet service",..: 2
1 2 2 1 3 3 3 1 2 …
 $ OnlineBackup            : Factor w/ 3 levels "No","No internet service",..: 2
1 2 2 1 3 1 3 1 2 …
 $ DeviceProtectionService: Factor w/ 3 levels "No","No internet service",..: 2
3 2 2 1 3 3 1 3 2 …
 $ TechnicalHelp           : Factor w/ 3 levels "No","No internet service",..: 2
3 2 2 1 3 3 1 3 2 …
 $ OnlineTV                : Factor w/ 3 levels "No","No internet service",..: 2
3 2 2 1 3 1 3 1 2 …
 $ OnlineMovies            : Factor w/ 3 levels "No","No internet service",..: 2
3 2 2 1 3 1 3 1 2 …
 $ Agreement               : Factor w/ 3 levels "Month-to-month",..: 2 1 3 3 1 3
1 2 1 2 …
 $ BillingMethod           : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 2 1
…
 $ PaymentMethod           : Factor w/ 4 levels "Bank transfer (automatic)",..: 4
2 1 4 3 2 1 2 4 4 …
```

```
 $ MonthlyServiceCharges  : num  20.6 53.4 18.4 26.3 75.2 …
 $ TotalAmount            : num  234 3579 1058 1689 889 …
 $ Churn                  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
…
```

Total 12335 observations with 21 variables present in our churn.csv dataset.Maximum we have factor variables with 2,3 and 4 levels.

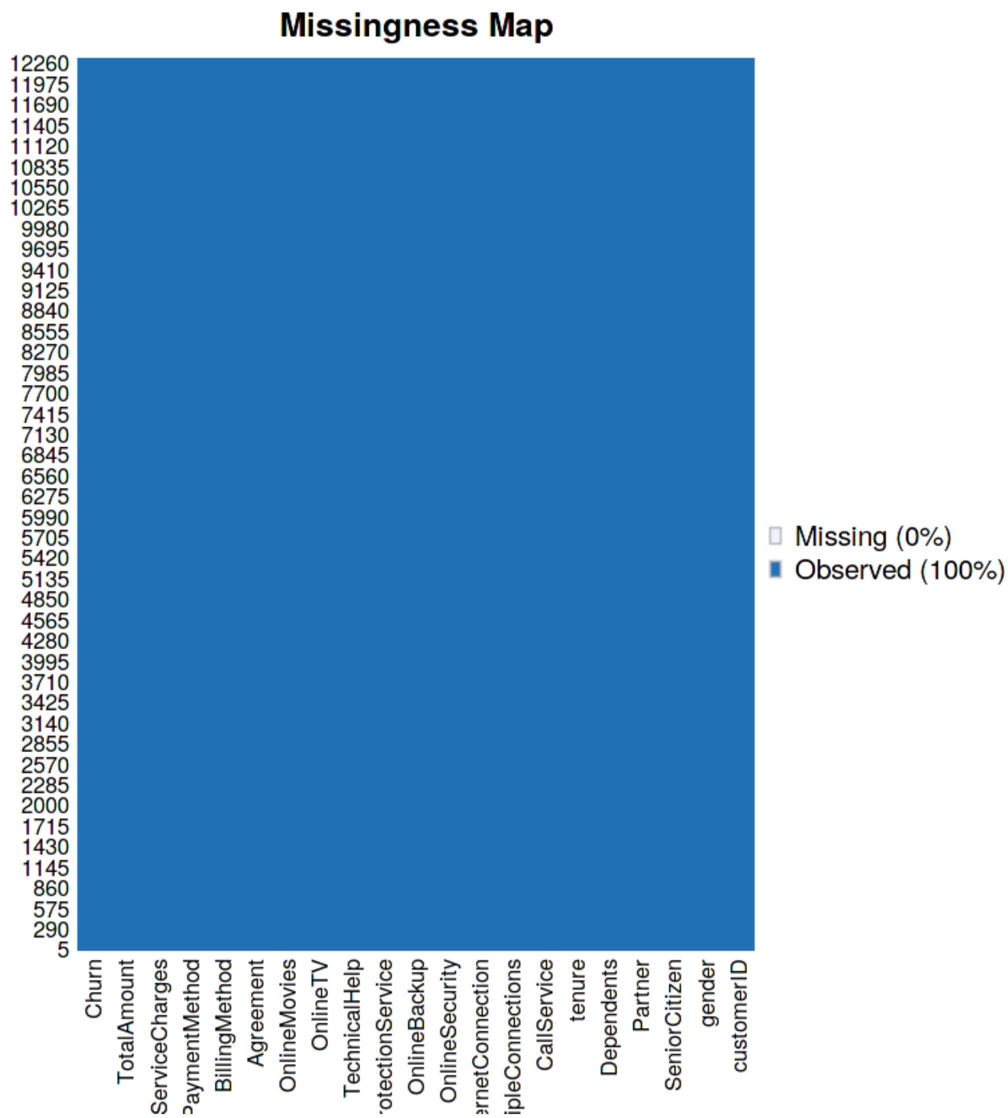
9 Checking for missing values

Loading Amelia package for missmap

[8]:
```r
install.packages("Amelia")
library(Amelia)
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Warning message:
"unable to access index for repository http://cran.rstudio.com/src/contrib:
  cannot open URL 'http://cran.rstudio.com/src/contrib/PACKAGES'"
Warning message:
"package 'Amelia' is not available (for R version 3.6.3)"
Loading required package: Rcpp

##
## Amelia II: Multiple Imputation
## (Version 1.7.6, built: 2019-11-24)
## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## Refer to http://gking.harvard.edu/amelia/ for more information
##
```

[9]:
```r
missmap(df)
```



No missing value present in our dataset.

[10]:
```
colSums(is.na(df))
```

customerID 0 **gender** 0 **SeniorCitizen** 0 **Partner** 0 **Dependents** 0 **tenure** 0 **CallService** 0 **MultipleConnections** 0 **InternetConnection** 0 **OnlineSecurity** 0 **OnlineBackup** 0 **DeviceProtectionService** 0 **TechnicalHelp** 0 **OnlineTV** 0 **OnlineMovies** 0 **Agreement** 0 **BillingMethod** 0 **PaymentMethod** 0 **MonthlyServiceCharges** 0 **TotalAmount** 0 **Churn** 0

This also confirms that our dataset contain no NA values.

[11]:
```
#Total number of customers

length(df$customerID)
```

12335

We have total 12335 customers present in our dataset.

10 Visualisation

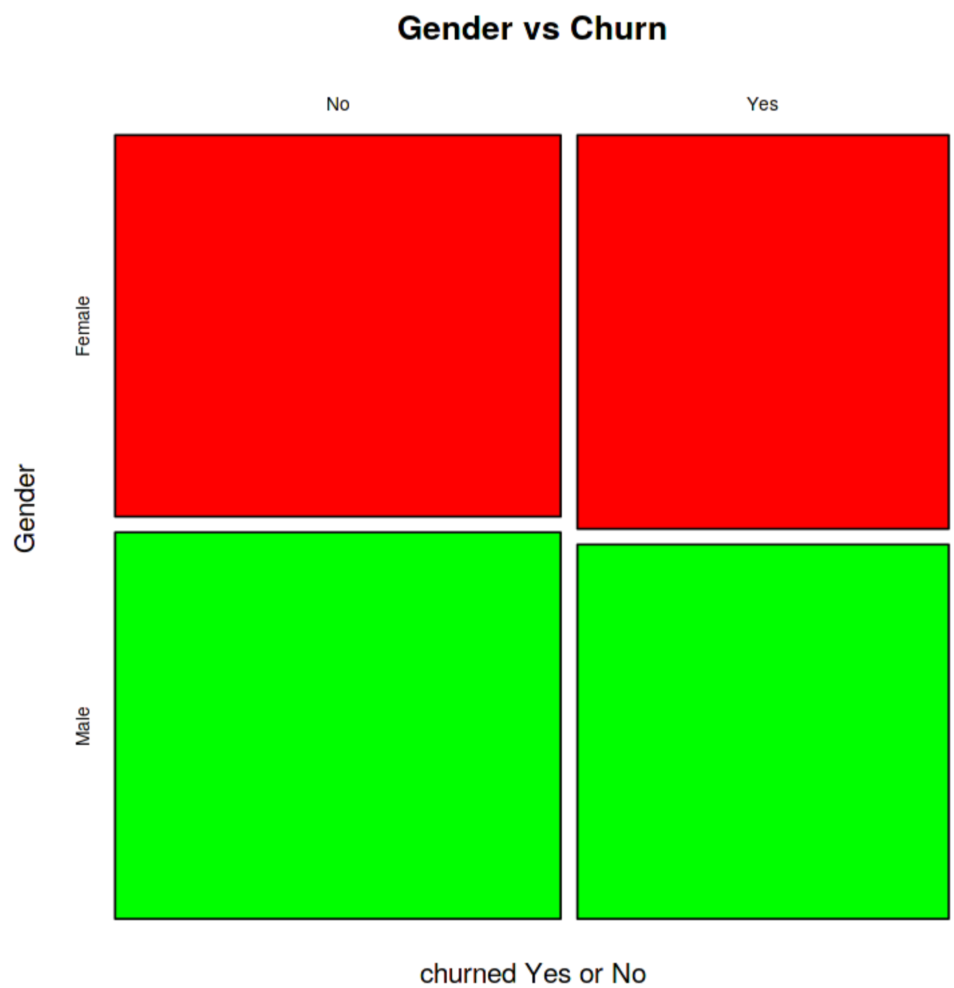
[12]:
```
#Number of Female and Male
count(df$gender)
```

A data.frame: 2 × 2

x	freq
<fct>	<int>
Female	6216
Male	6119

We have 6216 female and 6119 male present in dataset.

[13]:
```
#Gender vs Churn

plot(table(df$Churn, df$gender), col=c("red","green"),
      xlab = "churned Yes or No", ylab = "Gender", main = "Gender vs Churn")
```



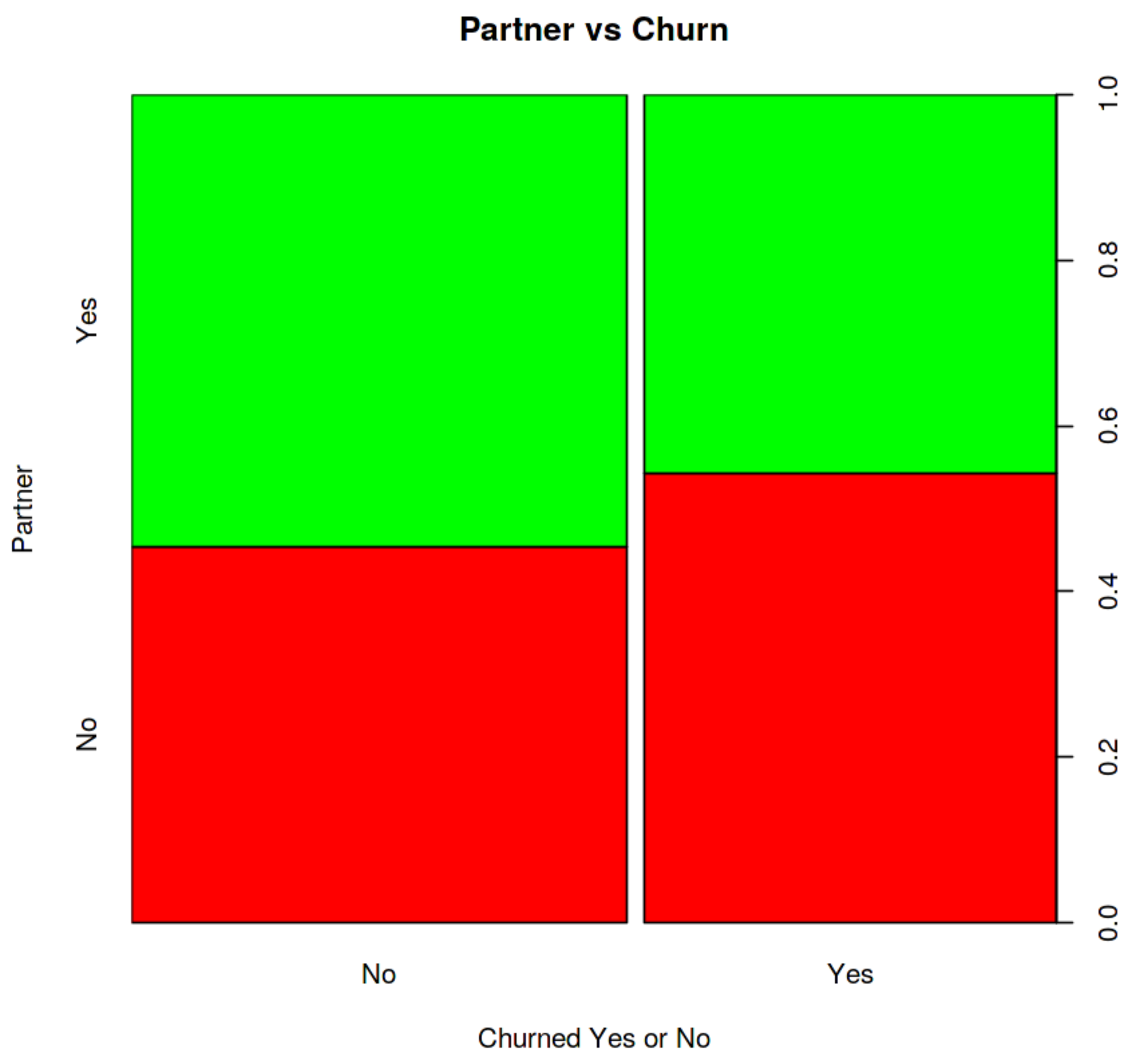
Churn is almost same in case of female and male, So gender is not very much influencing variable to churn.

[14]:
```
#Number of customers having partner
table(df$Partner)
```

```
  No  Yes
6105 6230
```

[15]:
```
#Partner vs Churn
```

```
plot(df$Churn, df$Partner, col=c("red","green"),
     xlab = "Churned Yes or No", ylab = "Partner", main = "Partner vs Churn")
```



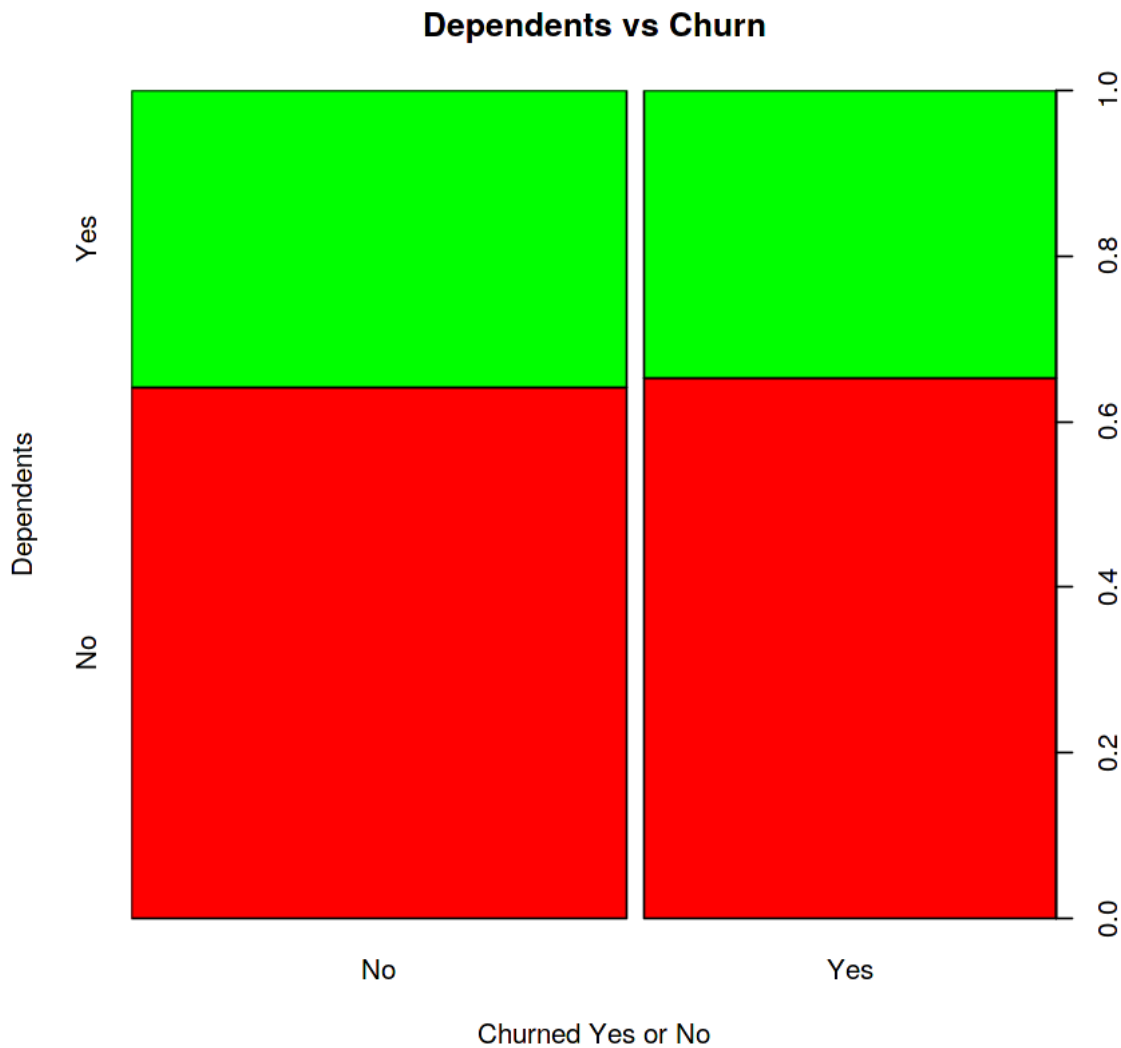
From above plot we can see that customers having partner has churned less than those who do not have partner.

[16]:
```
#Customers having dependents
table(df$Dependents)
```

```
  No  Yes
7974 4361
```

[17]:

```
#Dependents vs Churn

plot(df$Churn, df$Dependents, col=c("red","green"),
     xlab = "Churned Yes or No", ylab = "Dependents", main = "Dependents vs␣
 ↪Churn")
```

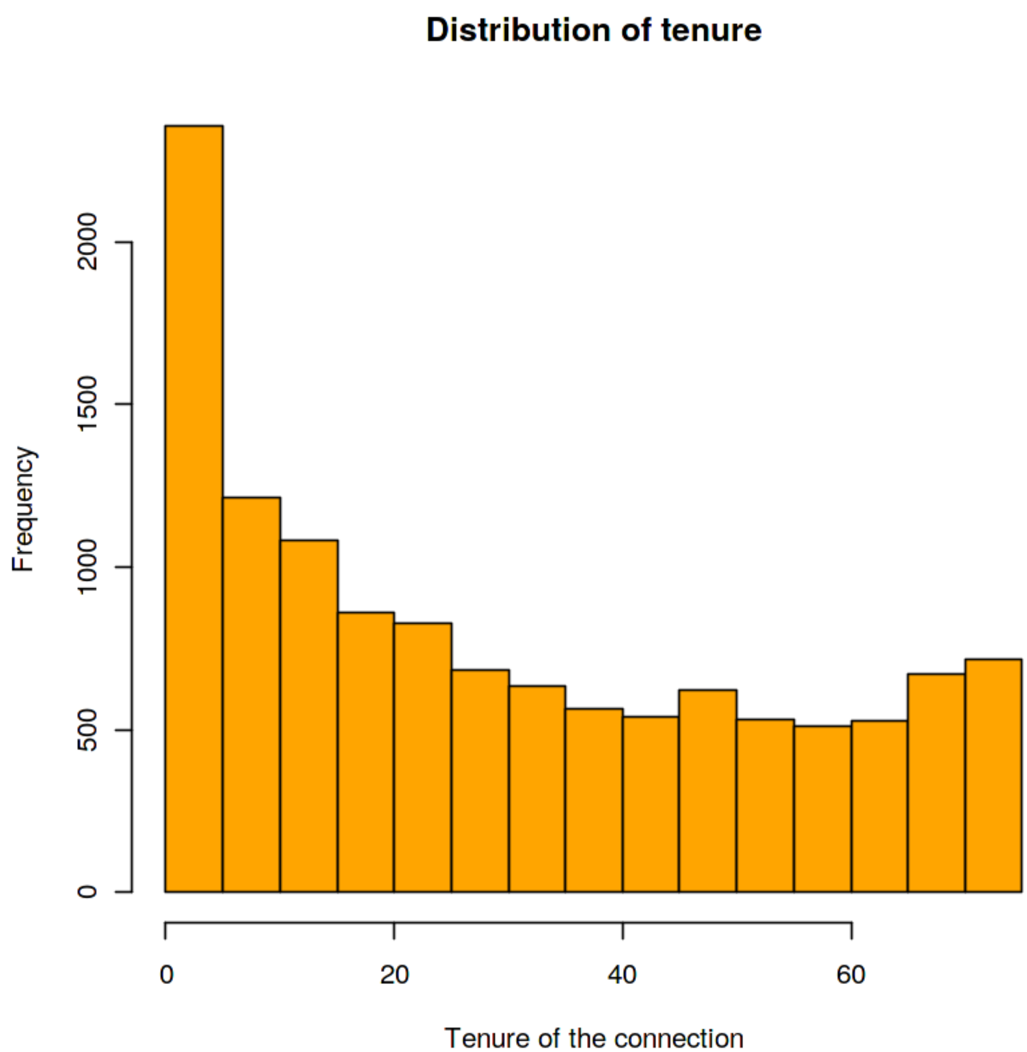


[18]:

```
#Summary of tenure
summary(df$tenure)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    8.00   24.00   29.53   49.00   72.00
```

[19]:
```
#Histogram of tenure
hist(df$tenure, col = "orange", xlab = "Tenure of the connection", main =␣
 ↪"Distribution of tenure")
```

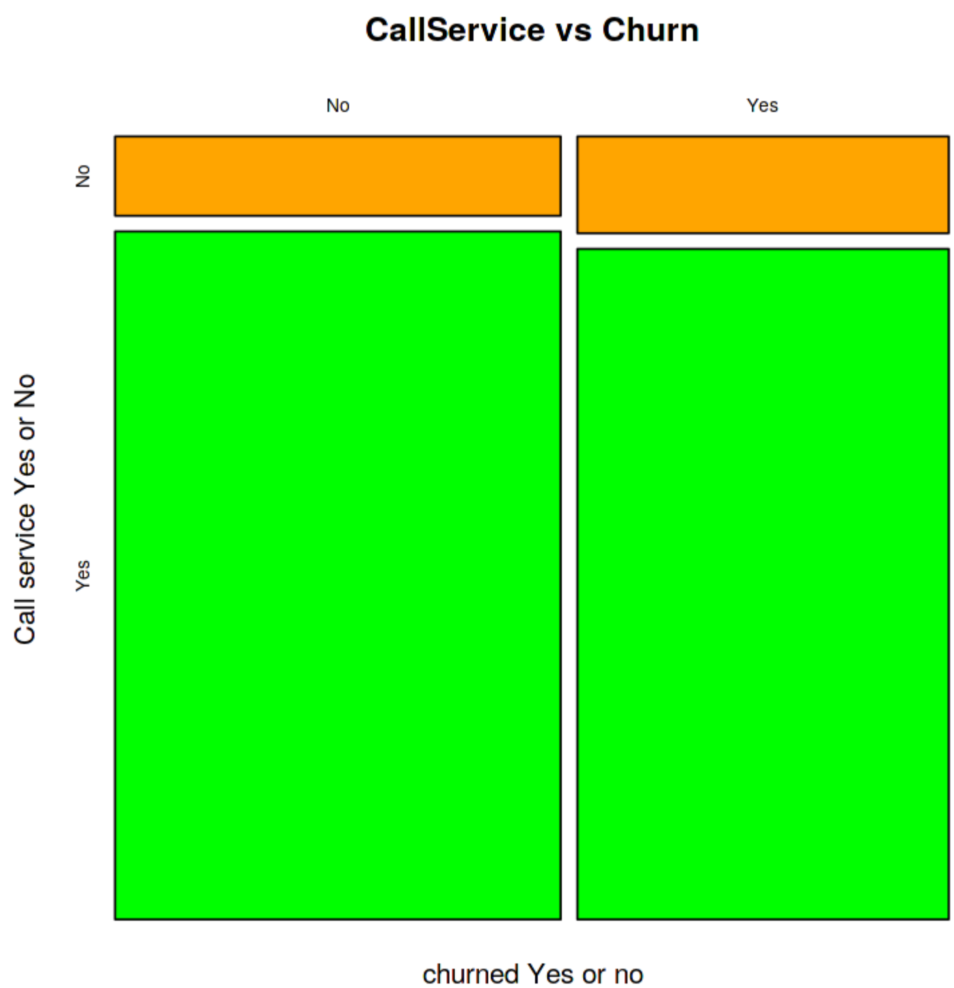


Maximum tenure of connection is between 0 to 10 months.

[20]:
```
#Number of customers subscribed to call service
table(df$CallService)
```

```
  No   Yes
1402 10933
```

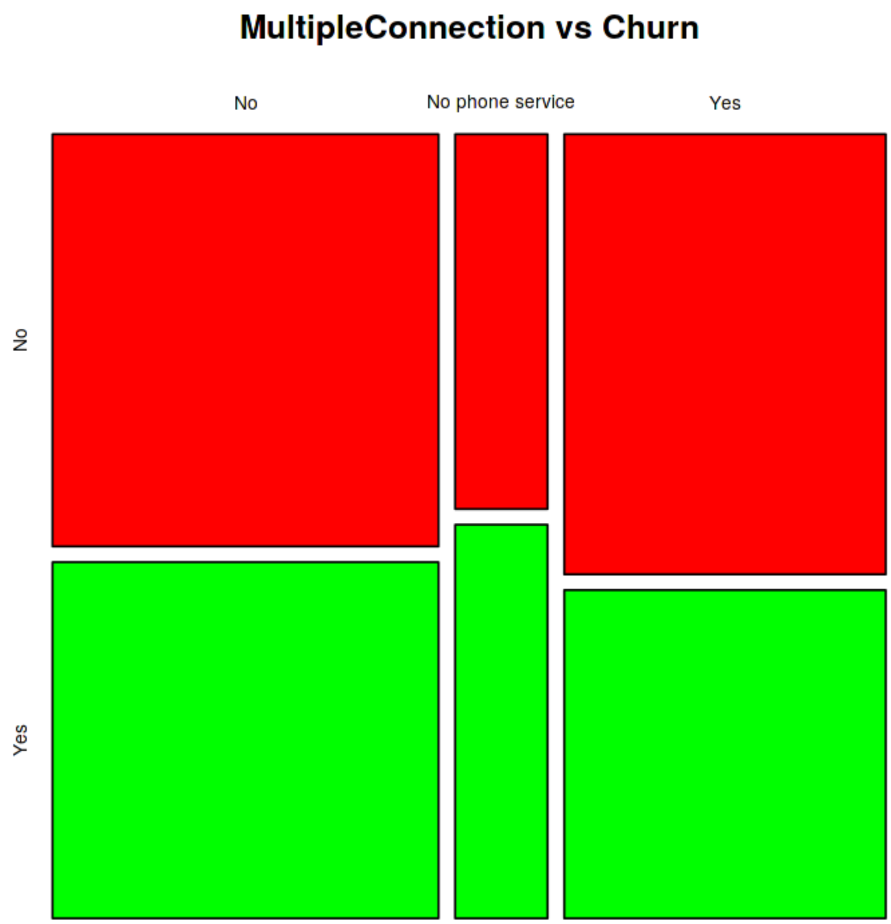
[21]:
```
plot(table(df$Churn, df$CallService), xlab= "churned Yes or no",
     ylab = "Call service Yes or No", col = c("orange", "green"), main =␣
 ↪"CallService vs Churn")
```



[22]:
```
#Multipleconnection

table(df$MultipleConnections)
```

```
   No No phone service              Yes
 5952             1425             4958
```

[23]:
```
#MultipleConnection vs Churn

plot(table(df$MultipleConnections, df$Churn), col = c("red", "green"), main =
 ↪"MultipleConnection vs Churn")
```

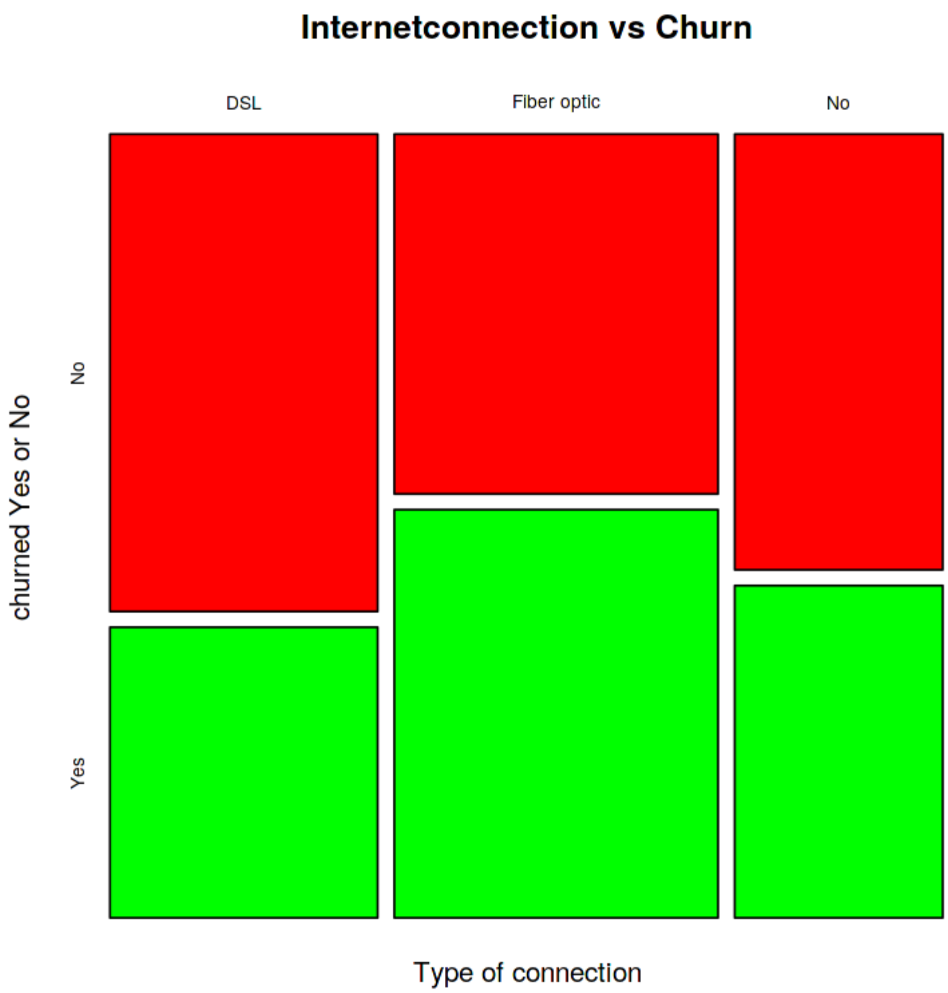


[24]:
```
#Type of InternetConnection

table(df$InternetConnection)
```

```
 DSL Fiber optic          No
4130        4995        3210
```

[25]:

```
#InternetConnection vs Churn
plot(table(df$InternetConnection, df$Churn),
     xlab = "Type of connection", ylab = "churned Yes or No",
     main = "Internetconnection vs Churn", col = c("red", "green"))
```



People having fiber optic internetconnection has churned more.

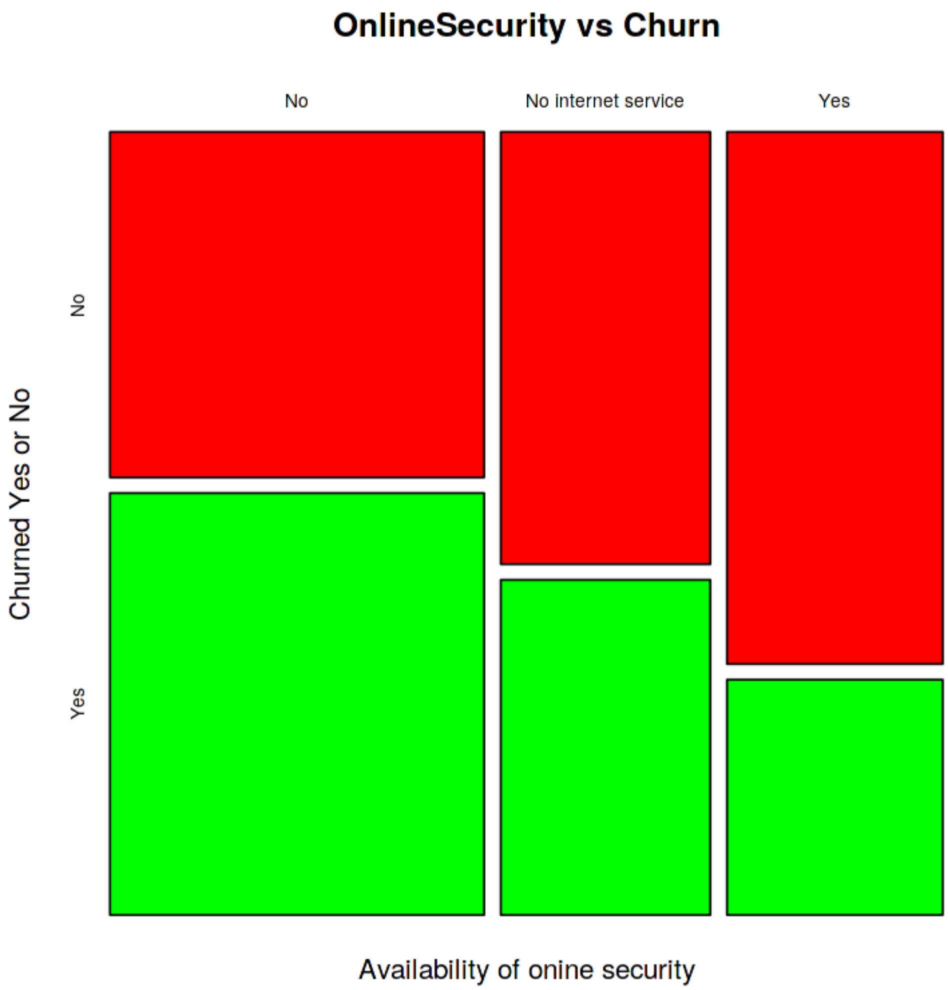
[26]:

```
#OnlineSecurity

table(df$OnlineSecurity)
```

```
  No No internet service                 Yes
5773                3233                3329
```

[27]:

```
#OnlineSecurity vs Churn

plot(table(df$OnlineSecurity, df$Churn),
     xlab = "Availability of onine security",ylab = "Churned Yes or No",
     col = c("red","green"), main = "OnlineSecurity vs Churn")
```



Customers having no online security has churned more.

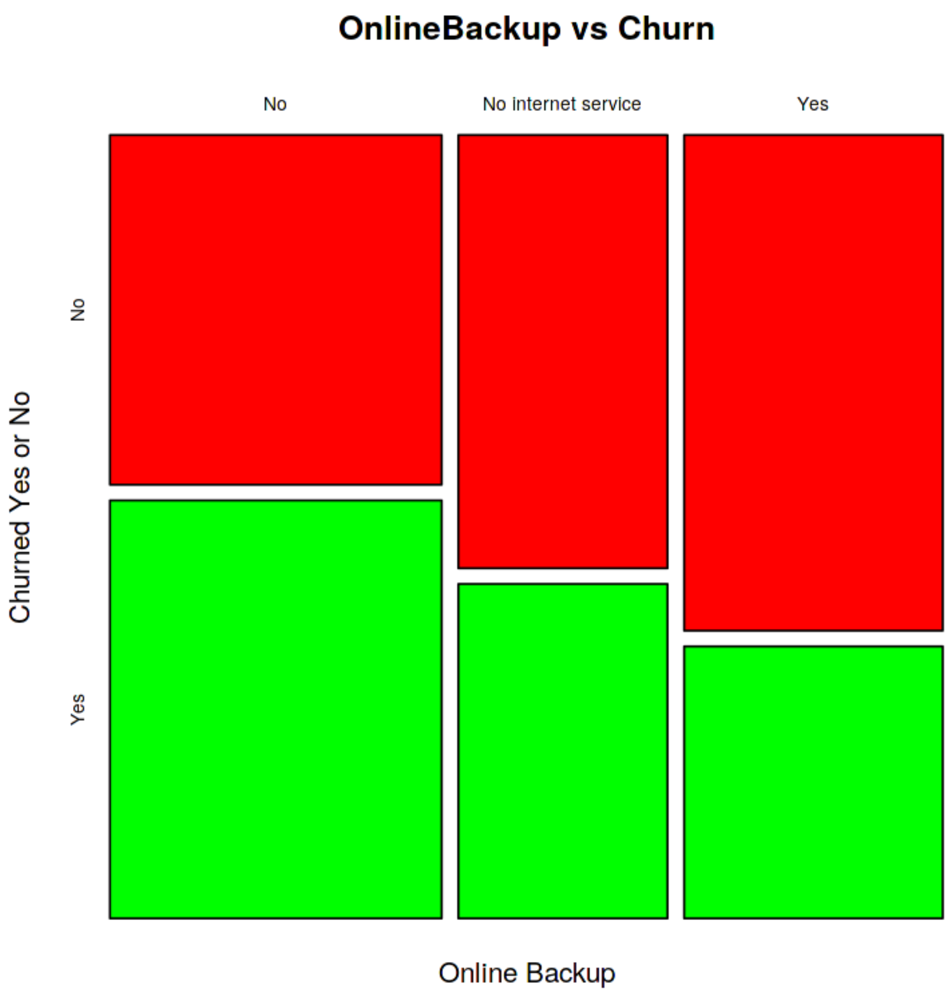
[28]:

```
#OnlineBackup

table(df$OnlineBackup)
```

```
             No No internet service                 Yes
```

5118 3227 3990

[29]:
```
#OnlineBackup vs Cjurn

plot(table(df$OnlineBackup, df$Churn),
     xlab = "Online Backup",ylab = "Churned Yes or No",
     col = c("red","green"), main = "OnlineBackup vs Churn")
```

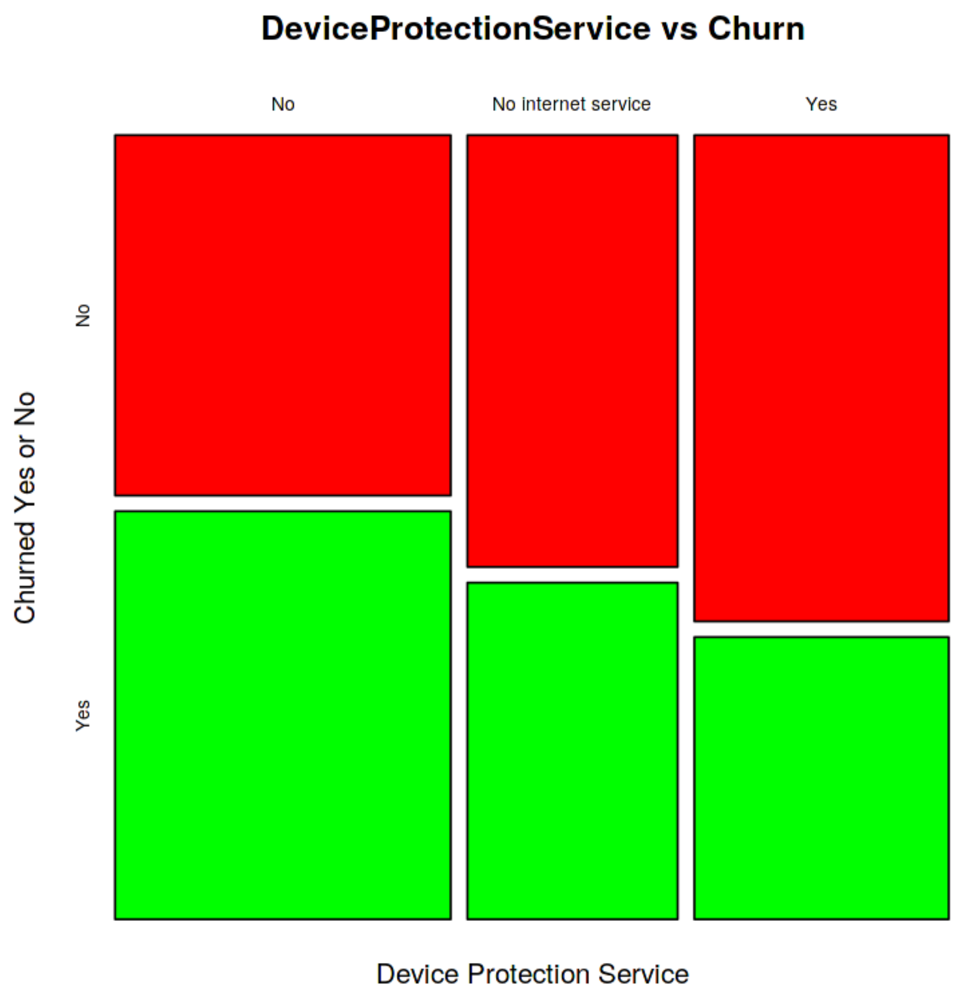


[30]:
```
#DeviceProtectionservice

table(df$DeviceProtectionService)
```

No No internet service Yes

```
5173                 3241                 3921
```

[31]:
```
#DeviceProtectionService vs Churn

plot(table(df$DeviceProtectionService, df$Churn),
     xlab = "Device Protection Service",ylab = "Churned Yes or No",
     col = c("red","green"), main = "DeviceProtectionService vs Churn")
```

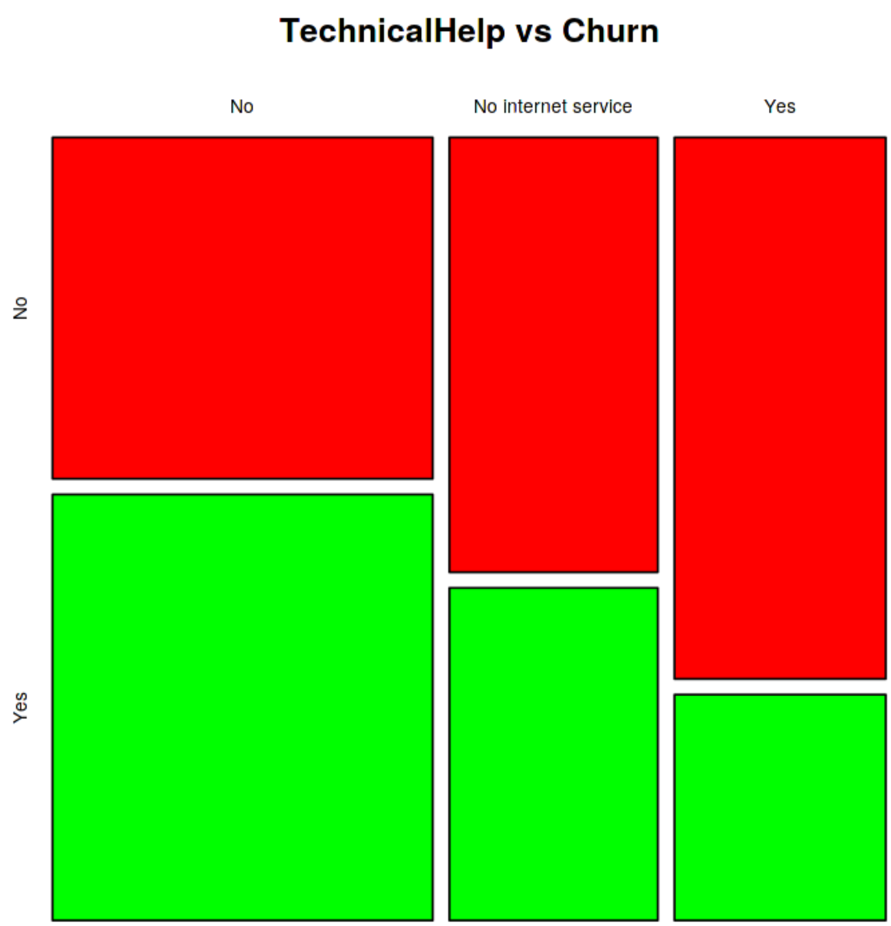


[32]:
```
#TechnicalHelp

table(df$TechnicalHelp)
```

```
   No No internet service                 Yes
```

```
5863                3214                3258
```

[33]:
```
#TechnicalHelp vs Churn

plot(table(df$TechnicalHelp, df$Churn),col = c("red","green"), main =␣
↪"TechnicalHelp vs Churn")
```

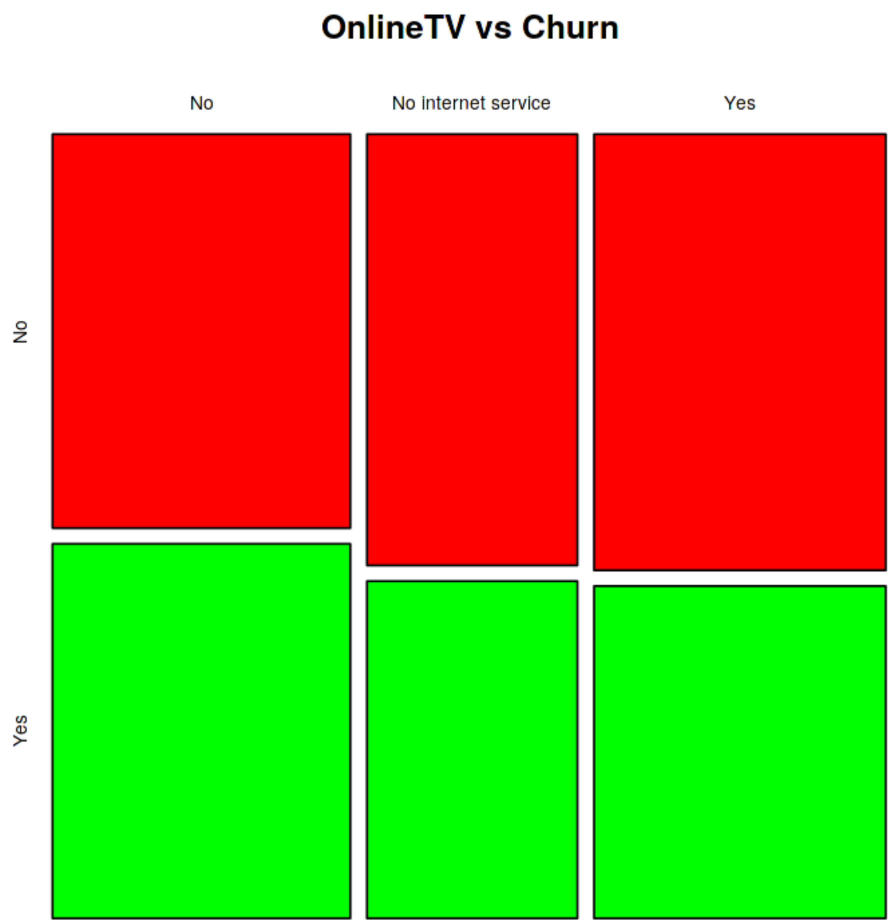


[34]:
```
#OnlineTv

table(df$OnlineTV)
```

```
  No No internet service                 Yes
4592                3245                4498
```

[35]:

```
#OnlineTv vs Churn

plot(table(df$OnlineTV, df$Churn),col = c("red","green"), main = "OnlineTV vs Churn")
```



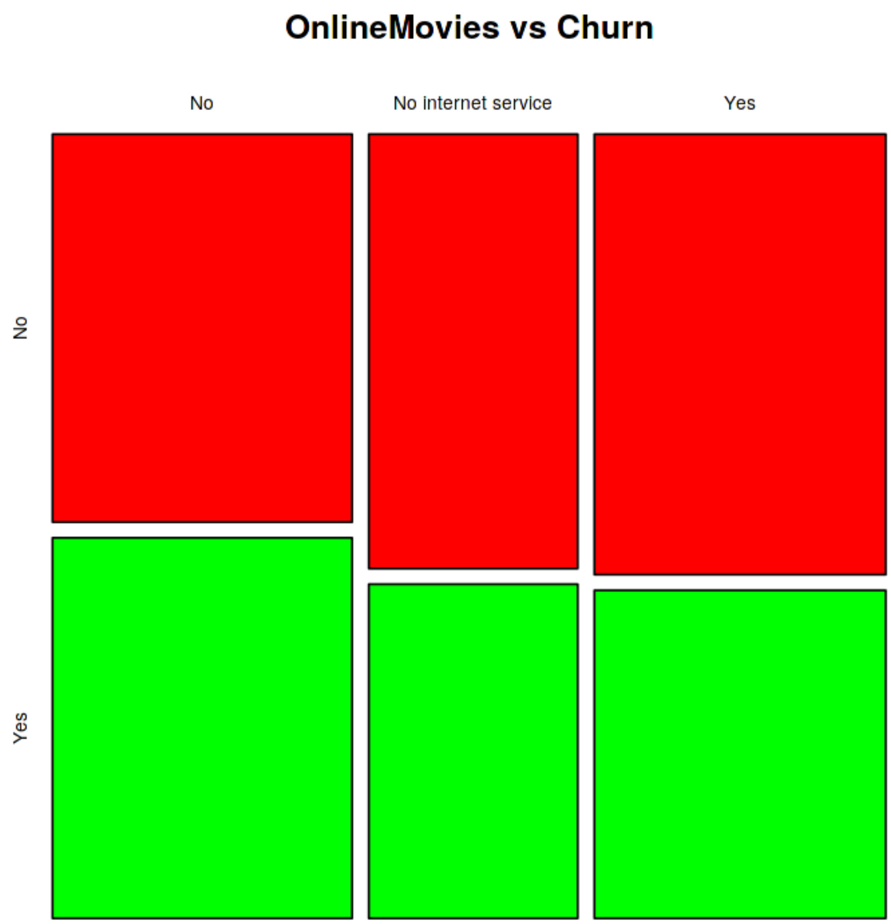
[36]:

```
#OnlineMovies

table(df$OnlineMovies)
```

```
  No No internet service                 Yes
4620                3222                4493
```

[37]:

```
#OnlineMovies vs Churn

plot(table(df$OnlineMovies, df$Churn),col = c("red","green"), main =␣
 ↪"OnlineMovies vs Churn")
```

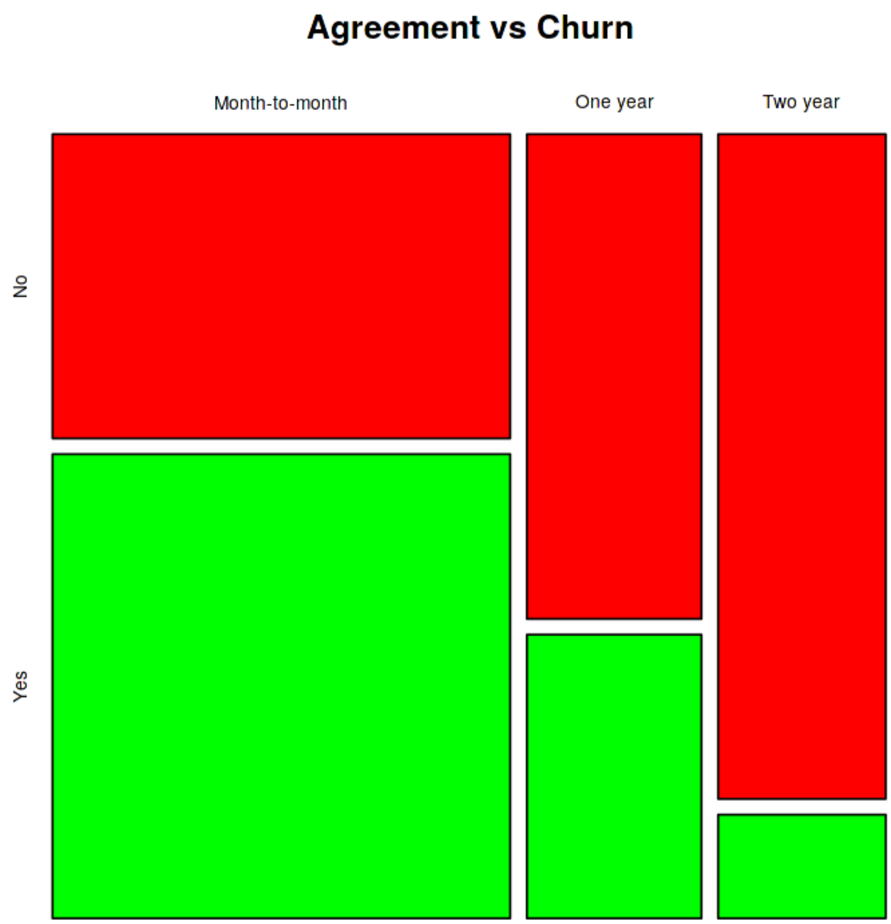


[38]:

```
#Agreement

table(df$Agreement)
```

```
Month-to-month       One year       Two year
          7058           2692           2585
```

[39]:
```
#Agreement vs Churn

plot(table(df$Agreement, df$Churn),col = c("red","green"), main ="Agreement vs␣
↪Churn")
```



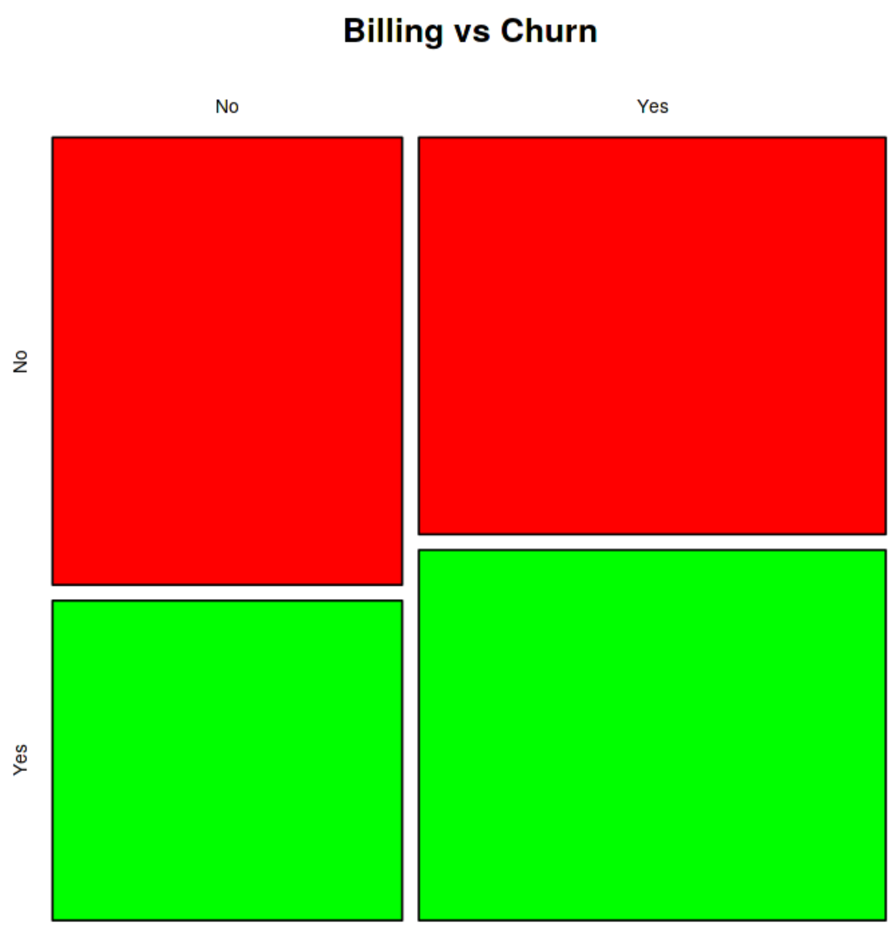
Customers subscribing two year plan has churned very less in comparison to month-to-month and one year.

[40]:
```
#BillingMethod

table(df$BillingMethod)
```

```
No  Yes
```

```
5282 7053
```

[41]:
```
#Billing vs Churn

plot(table(df$BillingMethod, df$Churn),col = c("red","green"), main = "Billing␣
↪vs Churn")
```

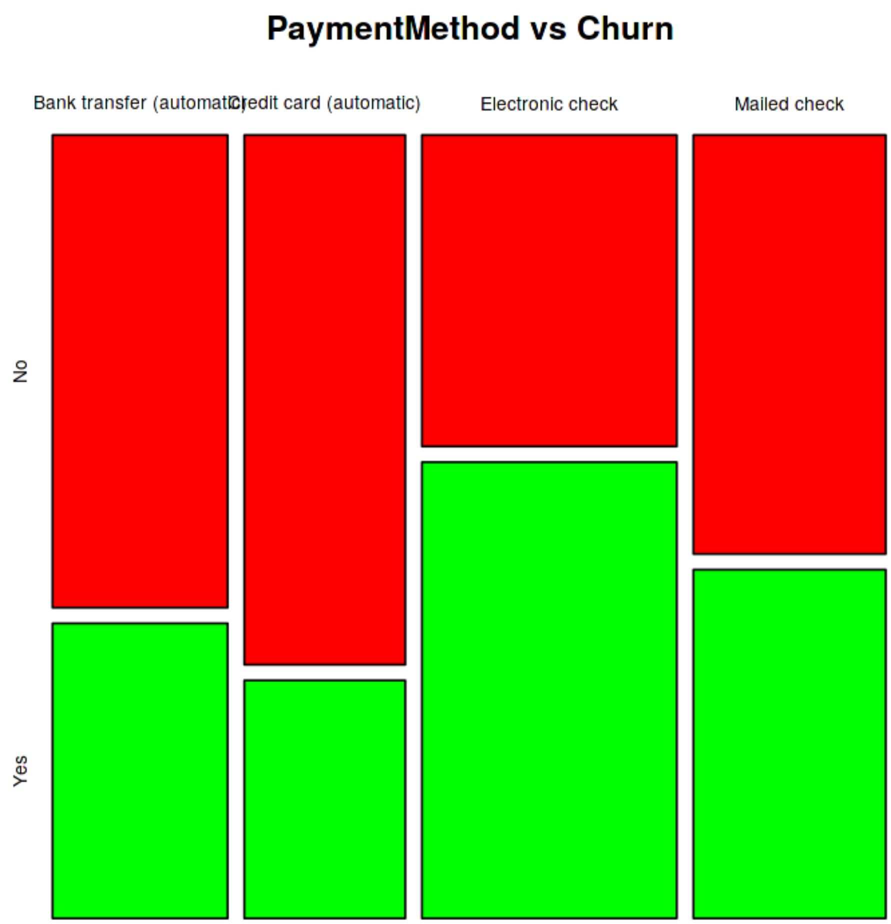


[42]:
```
#PaymentMethod

table(df$PaymentMethod)
```

```
Bank transfer (automatic)    Credit card (automatic)          Electronic check
                     2758                       2535                      4013
```

```
       Mailed check
               3029
```

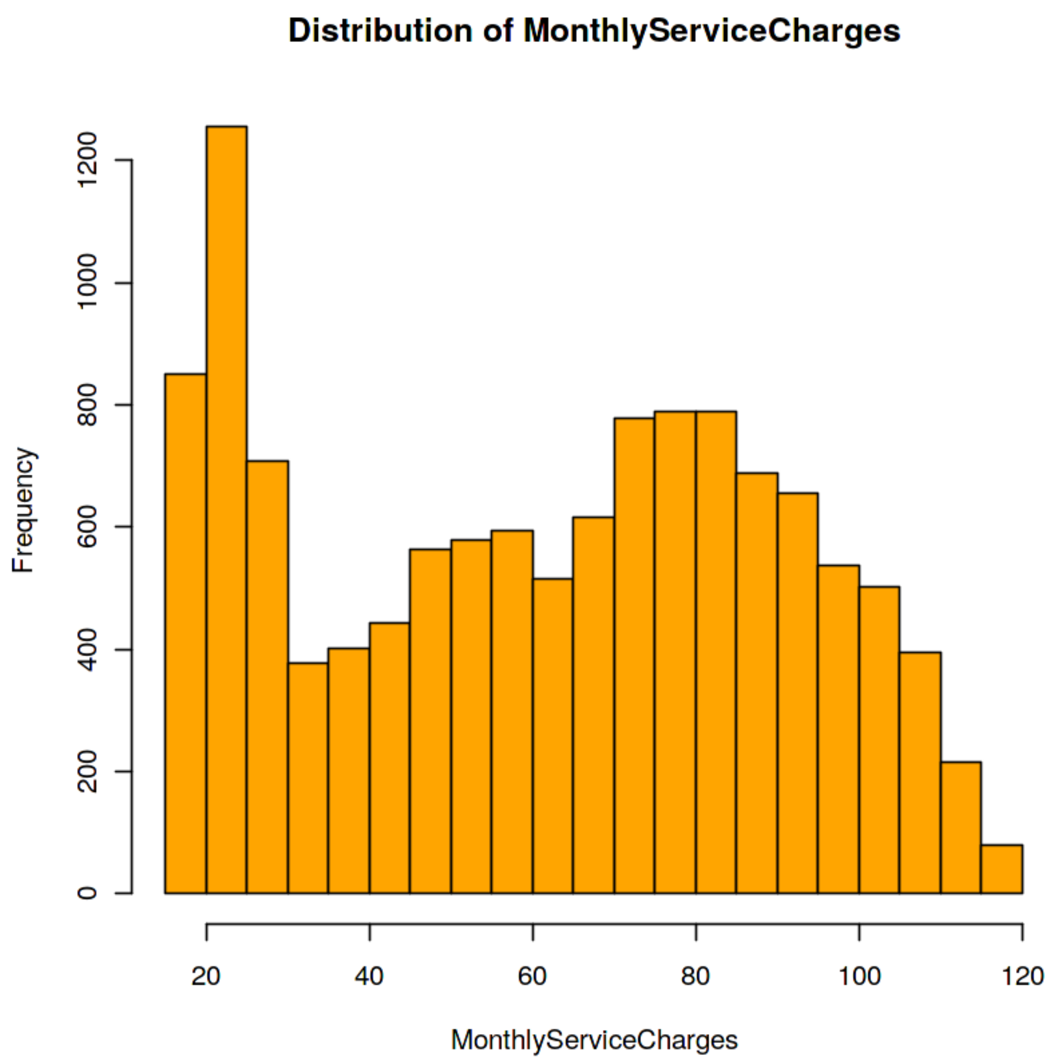
[43]:
```r
#PaymentMethod vs Churn

plot(table(df$PaymentMethod, df$Churn),col = c("red","green"), main =
 "PaymentMethod vs Churn")
```



[44]:
```r
#MonthlyServiceCharges

str(df$MonthlyServiceCharges)

summary(df$MonthlyServiceCharges)
```

```
num [1:12335] 20.6 53.4 18.4 26.3 75.2 …

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 18.25   33.89   64.36   61.50   85.00  118.75
```

[45]:
```
#Histogram of MonthlyServiceCharges

hist(df$MonthlyServiceCharges,col = "orange", main = "Distribution of␣
↪MonthlyServiceCharges",
    xlab="MonthlyServiceCharges")
```

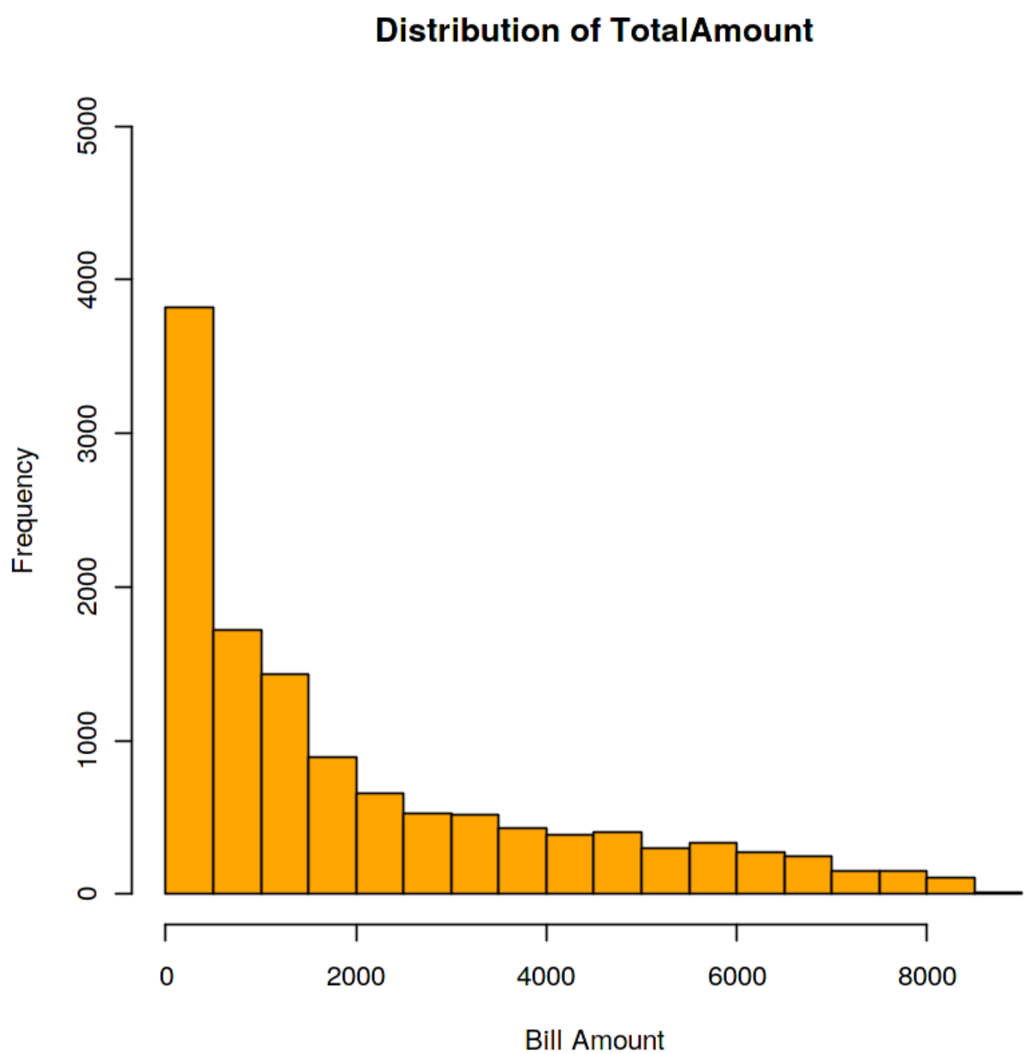


[46]:
```
#TotalAmount

str(df$TotalAmount)
```

```
summary(df$TotalAmount)
```

```
num [1:12335] 234 3579 1058 1689 889 …

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   18.8   352.7  1209.2  2032.3  3168.8  8684.8
```

[47]:
```
#Histogram of TotalAmount

hist(df$TotalAmount,col = "orange", xlab = "Bill Amount", ylim = c(0,5000),
 ↪main = "Distribution of TotalAmount")
```



[48]:
```
#customers who churned

table(df$Churn)
```

```
  No  Yes 
6728 5607 
```

Based on the result of the count each column change no internet service to "No" for six columns

They are: "OnlineSecrity", "OnlineBackup","Device Protection", technical-Help","Online TV","OnlineMovie"

[49]:
```
#Making a list of those columns.

cols_name <- c(10:15)

#Changing "No internet Srvice" to "No" for those columns

 for (i in 1:ncol(df[,cols_name])){
   df[,cols_name][,i]<- as.factor(mapvalues
                                                    (df[,cols_name][,i],from =␣
 ↪c("No internet service"), to = c("No")))}
```

[50]:
```
str(df)
```

```
'data.frame':   12335 obs. of  21 variables:
 $ customerID              : Factor w/ 5590 levels "0002-ORFBO","0004-TLHLJ",..:
1573 2155 5463 3755 3292 4852 5474 508 2771 4140 …
 $ gender                  : Factor w/ 2 levels "Female","Male": 1 1 1 2 1 1 1 1
2 2 …
 $ SeniorCitizen           : num  0 0 0 0 0 0 0 0 0 0 …
 $ Partner                 : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 2 1 1 1
…
 $ Dependents              : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 2 1 1 1
…
 $ tenure                  : num  11 67 59 67 11 36 49 54 26 19 …
 $ CallService             : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 2 2
…
 $ MultipleConnections     : Factor w/ 3 levels "No","No phone service",..: 1 2 1
3 3 3 1 1 3 1 …
 $ InternetConnection      : Factor w/ 3 levels "DSL","Fiber optic",..: 3 1 3 3 2
1 1 2 2 3 …
 $ OnlineSecurity          : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 1 1
…
 $ OnlineBackup            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 2 1 1
…
 $ DeviceProtectionService: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
```

```
…
 $ TechnicalHelp          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
…
 $ OnlineTV               : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ OnlineMovies           : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ Agreement              : Factor w/ 3 levels "Month-to-month",..: 2 1 3 3 1 3
1 2 1 2 …
 $ BillingMethod          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 2 1
…
 $ PaymentMethod          : Factor w/ 4 levels "Bank transfer (automatic)",..: 4
2 1 4 3 2 1 2 4 4 …
 $ MonthlyServiceCharges  : num  20.6 53.4 18.4 26.3 75.2 …
 $ TotalAmount            : num  234 3579 1058 1689 889 …
 $ Churn                  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
…
```

[51]:
```
table(df$OnlineBackup)
```

```
  No  Yes
8345 3990
```

[52]:
```
#Changing "No phone service" to "No" in MultipleConnections Column
df$MultipleConnections <- as.factor(mapvalues(df$MultipleConnections,from=c("No␣
 ↪phone service"),to = c("No")))
```

[53]:
```
str(df)
```

```
'data.frame':	12335 obs. of  21 variables:
 $ customerID             : Factor w/ 5590 levels "0002-ORFBO","0004-TLHLJ",..:
1573 2155 5463 3755 3292 4852 5474 508 2771 4140 …
 $ gender                 : Factor w/ 2 levels "Female","Male": 1 1 1 2 1 1 1 1
2 2 …
 $ SeniorCitizen          : num  0 0 0 0 0 0 0 0 0 0 …
 $ Partner                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 2 1 1 1
…
 $ Dependents             : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 2 1 1 1
…
 $ tenure                 : num  11 67 59 67 11 36 49 54 26 19 …
 $ CallService            : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 2 2
…
 $ MultipleConnections    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 1 2 1
…
 $ InternetConnection     : Factor w/ 3 levels "DSL","Fiber optic",..: 3 1 3 3 2
1 1 2 2 3 …
 $ OnlineSecurity         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 1 1
```

```
…
 $ OnlineBackup           : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 2 1 1
…
 $ DeviceProtectionService: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
…
 $ TechnicalHelp          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
…
 $ OnlineTV               : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ OnlineMovies           : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ Agreement              : Factor w/ 3 levels "Month-to-month",..: 2 1 3 3 1 3
1 2 1 2 …
 $ BillingMethod          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 2 1
…
 $ PaymentMethod          : Factor w/ 4 levels "Bank transfer (automatic)",..: 4
2 1 4 3 2 1 2 4 4 …
 $ MonthlyServiceCharges  : num  20.6 53.4 18.4 26.3 75.2 …
 $ TotalAmount            : num  234 3579 1058 1689 889 …
 $ Churn                  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
…
```

Now we succesfully changed “No internet service” and “No phone service” to No.

[54]:
```
table(df$SeniorCitizen)
```

```
          0 0.003857827  0.00465552 0.006524713 0.008518171 0.008563273
      10016           1           1           1           1           1
0.008908875 0.010998659 0.011698099 0.012328669 0.013113638 0.013501396
          1           1           1           1           1           1
0.013558515 0.014229723  0.01556298 0.017082523  0.02047418  0.02059616
          1           1           1           1           1           1
0.021272866 0.023845578 0.025250689 0.025826494 0.026318367 0.028499586
          1           1           1           1           1           1
0.028762124 0.029793944 0.031514167 0.031709892  0.03204944 0.032241194
          1           1           1           1           1           1
0.032586338 0.033442283 0.034359743 0.034574792 0.034883993 0.038973481
          1           1           1           1           1           1
0.042720959 0.051435794 0.051815482 0.054826488 0.055136558 0.055211132
          1           1           1           1           1           1
 0.05657654 0.056714567 0.056996101 0.058016615 0.061333964 0.061677696
          1           1           1           1           1           1
0.061706078 0.063556034 0.064050917 0.064165563 0.066563621 0.066646616
          1           1           1           1           1           1
0.067525074 0.069547823 0.070634143 0.072058338 0.074917246 0.075545687
          1           1           1           1           1           1
 0.07618527 0.077031926 0.077305482 0.077904413 0.078885755 0.082340542
```

1	1	1	1	1	1
0.083357555	0.08547894	0.086104149	0.087771574	0.08942089	0.090776469
1	1	1	1	1	1
0.091732421	0.093429101	0.093930092	0.097539386	0.097833508	0.097993799
1	1	1	1	1	1
0.098317471	0.0983902	0.099159396	0.10372812	0.104066707	0.109951496
1	1	1	1	1	1
0.11108816	0.111103816	0.111568782	0.11258284	0.112709404	0.112713476
1	1	1	1	1	1
0.113082277	0.114467089	0.115649531	0.117000961	0.118861037	0.121528436
1	1	1	1	1	1
0.122807605	0.123592348	0.124400882	0.128056267	0.129521861	0.129778767
1	1	1	1	1	1
0.130611929	0.1344534	0.134527015	0.137812692	0.137830729	0.141519899
1	1	1	1	1	1
0.14173339	0.142143402	0.143440839	0.14792925	0.14882965	0.150739137
1	1	1	1	1	1
0.15170098	0.152114472	0.153039768	0.153468119	0.154605203	0.155298075
1	1	1	1	1	1
0.165195204	0.165888258	0.16605313	0.166555315	0.168829921	0.171720638
1	1	1	1	1	1
0.172212287	0.173604144	0.175727479	0.178990989	0.179067532	0.180082669
1	1	1	1	1	1
0.181172687	0.182152024	0.182590453	0.182751149	0.185229733	0.185234194
1	1	1	1	1	1
0.188359586	0.188543621	0.189940861	0.191308704	0.191822262	0.191883625
1	1	1	1	1	1
0.193573121	0.193859956	0.194218781	0.195787759	0.197982981	0.198390421
1	1	1	1	1	1
0.200289505	0.201062074	0.201693825	0.204244183	0.205623435	0.206866998
1	1	1	1	1	1
0.207995106	0.208767761	0.209907392	0.209978155	0.210632709	0.21144637
1	1	1	1	1	1
0.211703739	0.211715281	0.212940688	0.212996645	0.213310258	0.214276642
1	1	1	1	1	1
0.21432071	0.215092062	0.216252554	0.217080126	0.218250421	0.223919433
1	1	1	1	1	1
0.224610123	0.224775624	0.225505869	0.226683937	0.229928648	0.230070701
1	1	1	1	1	1
0.233113155	0.233923754	0.234236486	0.235216907	0.236509481	0.237349355
1	1	1	1	1	1
0.239538674	0.240740984	0.240977415	0.241293875	0.241353143	0.244312502
1	1	1	1	1	1
0.246443869	0.24843664	0.250909456	0.251488884	0.25296413	0.257210659
1	1	1	1	1	1
0.258519904	0.259153702	0.259712705	0.261161787	0.261331137	0.26204036
1	1	1	1	1	1
0.262433544	0.265930452	0.266156174	0.267948997	0.270762394	0.271026762

1	1	1	1	1	1
0.271325728	0.273585995	0.276059793	0.277741147	0.278098355	0.278969076
1	1	1	1	1	1
0.279881425	0.280215198	0.281119173	0.281640721	0.281993467	0.283811913
1	1	1	1	1	1
0.283930874	0.284235164	0.28452982	0.285340468	0.286827425	0.287868874
1	1	1	1	1	1
0.289138849	0.289625767	0.290015215	0.291733192	0.292679736	0.293193139
1	1	1	1	1	1
0.294492116	0.294607221	0.297570083	0.29843659	0.302669067	0.304237471
1	1	1	1	1	1
0.305683867	0.306874979	0.311129158	0.314945232	0.315073581	0.316002145
1	1	1	1	1	1
0.319287835	0.320773681	0.322724345	0.323418986	0.323949499	0.325244392
1	1	1	1	1	1
0.325272369	0.32535024	0.325889745	0.326708607	0.327250977	0.327845486
1	1	1	1	1	1
0.328259317	0.328672019	0.329219429	0.3318019	0.334009602	0.334109984
1	1	1	1	1	1
0.334683667	0.338797132	0.341520198	0.342845518	0.345465865	0.3471547
1	1	1	1	1	1
0.34981906	0.352215733	0.352672822	0.35311647	0.356052797	0.357245691
1	1	1	1	1	1
0.357423297	0.359130135	0.359658089	0.361070342	0.363789614	0.364861082
1	1	1	1	1	1
0.365561157	0.365637372	0.36650935	0.367292218	0.368976904	0.369045646
1	1	1	1	1	1
0.371429934	0.374944251	0.376196469	0.377678059	0.378760106	0.382167445
1	1	1	1	1	1
0.382496186	0.382835615	0.383601692	0.384269835	0.387478333	0.388082126
1	1	1	1	1	1
0.392519722	0.398613939	0.400875106	0.403935296	0.404647781	0.405314444
1	1	1	1	1	1
0.405664639	0.407347098	0.408460309	0.412508713	0.414706241	0.415236023
1	1	1	1	1	1
0.416644795	0.419396054	0.423269237	0.423624149	0.423915209	0.427306964
1	1	1	1	1	1
0.427635222	0.428588828	0.429375762	0.429650449	0.431213823	0.43166743
1	1	1	1	1	1
0.433179443	0.433591105	0.433855521	0.433949603	0.436161347	0.43637575
1	1	1	1	1	1
0.436587603	0.437305236	0.437520759	0.438757583	0.440469432	0.442306299
1	1	1	1	1	1
0.442874096	0.445910264	0.447308016	0.449266237	0.449427263	0.450011794
1	1	1	1	1	1
0.450519222	0.450993647	0.45104372	0.451255336	0.456053096	0.456357463
1	1	1	1	1	1
0.456617411	0.457843551	0.460360525	0.463822236	0.466210164	0.468691698

1	1	1	1	1	1
0.468761807	0.470763215	0.471034163	0.472404078	0.472799282	0.472813102
1	1	1	1	1	1
0.472867324	0.473400935	0.474111528	0.474576837	0.474879745	0.475886984
1	1	1	1	1	1
0.476285657	0.480415298	0.482968479	0.483360448	0.483410483	0.483792907
1	1	1	1	1	1
0.484977509	0.488732842	0.48933198	0.493139159	0.495390161	0.497156918
1	1	1	1	1	1
0.499477753	0.500642476	0.500818173	0.501247947	0.504409113	0.505160275
1	1	1	1	1	1
0.505249158	0.514482028	0.515318312	0.515346002	0.517356654	0.517555528
1	1	1	1	1	1
0.517782991	0.51916053	0.519839242	0.520026757	0.521436196	0.525551035
1	1	1	1	1	1
0.526494311	0.526577041	0.526930159	0.528348784	0.530789198	0.532520314
1	1	1	1	1	1
0.533613873	0.534104352	0.534861278	0.535934892	0.536019154	0.537862896
1	1	1	1	1	1
0.539528827	0.539980022	0.540560799	0.544876271	0.54540415	0.546476759
1	1	1	1	1	1
0.551250627	0.551492745	0.552631752	0.552959164	0.553650316	0.554083384
1	1	1	1	1	1
0.558028638	0.558254468	0.558988263	0.559361918	0.559458549	0.559463662
1	1	1	1	1	1
0.559689906	0.561573906	0.561821516	0.561905891	0.562129573	0.562740079
1	1	1	1	1	1
0.563226804	0.571219794	0.571341003	0.571596794	0.573709323	0.573717076
1	1	1	1	1	1
0.57466145	0.575298025	0.575482619	0.577043216	0.579402945	0.579900529
1	1	1	1	1	1
0.580020512	0.580373745	0.58150333	0.581771042	0.582403217	0.583119486
1	1	1	1	1	1
0.583469544	0.583521586	0.584007717	0.586101135	0.587296052	0.587771692
1	1	1	1	1	1
0.587851118	0.589052522	0.590525065	0.593170459	0.59327852	0.593505125
1	1	1	1	1	1
0.594108832	0.594788386	0.595197104	0.598826872	0.600333617	0.600503854
1	1	1	1	1	1
0.604145802	0.604639039	0.604819569	0.605935659	0.607499203	0.608540297
1	1	1	1	1	1
0.60998252	0.611871842	0.615663716	0.615712199	0.616437795	0.616736584
1	1	1	1	1	1
0.616924991	0.618747628	0.621598859	0.623770492	0.623787284	0.624691728
1	1	1	1	1	1
0.625414841	0.626183292	0.627034717	0.630195943	0.630274197	0.631767487
1	1	1	1	1	1
0.633000272	0.638407124	0.638730871	0.642656836	0.643121044	0.644504066

1	1	1	1	1	1
0.645103385	0.645278762	0.645730803	0.646342094	0.650140136	0.650803014
1	1	1	1	1	1
0.652664539	0.653311316	0.65703394	0.65753467	0.659142214	0.65985756
1	1	1	1	1	1
0.660001819	0.660614163	0.663148071	0.664287903	0.665819884	0.666230617
1	1	1	1	1	1
0.670454311	0.670481909	0.671606591	0.672038883	0.673260624	0.673672702
1	1	1	1	1	1
0.677793182	0.678111682	0.681270725	0.68221243	0.685088699	0.690445404
1	1	1	1	1	1
0.692140001	0.693021697	0.695775508	0.696629466	0.697479531	0.699465891
1	1	1	1	1	1
0.699986783	0.700342053	0.700739367	0.701501259	0.702012106	0.702573634
1	1	1	1	1	1
0.702805652	0.704295477	0.704439314	0.710424562	0.712851638	0.713902897
1	1	1	1	1	1
0.714355498	0.715589884	0.715610342	0.717176691	0.717250877	0.718534134
1	1	1	1	1	1
0.719649022	0.720085149	0.720576632	0.721636763	0.721658012	0.721979128
1	1	1	1	1	1
0.722382552	0.724283206	0.724908718	0.725094253	0.726407701	0.727298584
1	1	1	1	1	1
0.734055876	0.735255474	0.739842833	0.746077573	0.748675397	0.749083378
1	1	1	1	1	1
0.749504747	0.750185426	0.751643758	0.752007675	0.756618423	0.761411839
1	1	1	1	1	1
0.761931067	0.764605652	0.76538284	0.765763372	0.76698912	0.767728461
1	1	1	1	1	1
0.770774351	0.771164154	0.771431789	0.771531713	0.773512039	0.774326474
1	1	1	1	1	1
0.776809816	0.778340481	0.779016653	0.780213135	0.781305639	0.781751369
1	1	1	1	1	1
0.782739043	0.785069242	0.786355085	0.790299343	0.791987673	0.792188522
1	1	1	1	1	1
0.794953377	0.795233545	0.796965073	0.798557793	0.802464193	0.802793813
1	1	1	1	1	1
0.803996712	0.804498777	0.804784045	0.804793123	0.804861712	0.807587147
1	1	1	1	1	1
0.809618691	0.809881807	0.811592543	0.813177514	0.813338169	0.815531802
1	1	1	1	1	1
0.817900462	0.819614359	0.821066113	0.821354871	0.822778819	0.822988433
1	1	1	1	1	1
0.823744255	0.823930965	0.824063394	0.829612598	0.829772254	0.833134274
1	1	1	1	1	1
0.834137316	0.834666324	0.836104982	0.836977701	0.837164746	0.837179009
1	1	1	1	1	1
0.838187587	0.840392685	0.841529555	0.842735098	0.845699562	0.848292165

```
          1           1           1           1           1           1
0.852881742 0.855754117 0.856416991 0.857383418 0.858183591 0.867908356
          1           1           1           1           1           1
0.869242856 0.872091208 0.873107631 0.873187642 0.873892559 0.874403952
          1           1           1           1           1           1
0.876399825 0.878696314 0.879815674 0.880191055  0.88082031 0.882273618
          1           1           1           1           1           1
0.883286237 0.883639475 0.884045275 0.884739611 0.887830669 0.889594279
          1           1           1           1           1           1
0.890754571 0.891344635 0.891403701 0.894910851 0.896298997  0.89752094
          1           1           1           1           1           1
0.897644202  0.90210309 0.904506477 0.904726576 0.905823029 0.909390314
          1           1           1           1           1           1
0.909494864 0.911922923 0.914337188 0.914487042 0.915584998 0.916970611
          1           1           1           1           1           1
0.917416397 0.918041798 0.918287711 0.919360165 0.928599135  0.92874671
          1           1           1           1           1           1
 0.92885357  0.93012818  0.93036453 0.934015729 0.934973012 0.935415347
          1           1           1           1           1           1
 0.93581511 0.936273263 0.937790202 0.939338444  0.94044364 0.940514781
          1           1           1           1           1           1
0.942736097 0.942757396 0.943145868 0.946500182 0.946626705 0.949943666
          1           1           1           1           1           1
0.950957485  0.95099927 0.952268789 0.952712607 0.952793419 0.958821871
          1           1           1           1           1           1
0.959224348 0.959872886 0.961506301 0.963918892 0.964903708 0.965852818
          1           1           1           1           1           1
0.966273368 0.966455176  0.96646004 0.966657008 0.968218824 0.971854033
          1           1           1           1           1           1
0.973772097 0.975392685 0.975583901 0.979844439 0.980566715 0.981017978
          1           1           1           1           1           1
0.981326094 0.981566266 0.981587364  0.98212074 0.983134119 0.984139524
          1           1           1           1           1           1
0.984876699 0.985146298 0.987640246 0.987661698 0.988914535 0.989480832
          1           1           1           1           1           1
0.989636509 0.989826966 0.991243852 0.991494119 0.993087757 0.995342379
          1           1           1           1           1           1
0.995614282 0.995618216 0.998687699 0.998799711           1
          1           1           1           1        1566
```

[55]:

```
#Removing columns which are not very significant for our analysis

df$SeniorCitizen <- NULL

df$customerID <- NULL

df$gender <- NULL
```

[56]:
```
str(df)
```

```
'data.frame':	12335 obs. of  18 variables:
 $ Partner                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 2 1 1 1
…
 $ Dependents             : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 2 1 1 1
…
 $ tenure                 : num  11 67 59 67 11 36 49 54 26 19 …
 $ CallService            : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 2 2
…
 $ MultipleConnections    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 1 2 1
…
 $ InternetConnection     : Factor w/ 3 levels "DSL","Fiber optic",..: 3 1 3 3 2
1 1 2 2 3 …
 $ OnlineSecurity         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 1 1
…
 $ OnlineBackup           : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 2 1 1
…
 $ DeviceProtectionService: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
…
 $ TechnicalHelp          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 2 1 2 1
…
 $ OnlineTV               : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ OnlineMovies           : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1
…
 $ Agreement              : Factor w/ 3 levels "Month-to-month",..: 2 1 3 3 1 3
1 2 1 2 …
 $ BillingMethod          : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 2 1
…
 $ PaymentMethod          : Factor w/ 4 levels "Bank transfer (automatic)",..: 4
2 1 4 3 2 1 2 4 4 …
 $ MonthlyServiceCharges  : num  20.6 53.4 18.4 26.3 75.2 …
 $ TotalAmount            : num  234 3579 1058 1689 889 …
 $ Churn                  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
…
```

SeniorCitizen, CustomerID and Gender columns are removed from dataset.

11 Model Development

Splitting the data into training and testing sets

[57]:
```
library(caTools)
```

[58]:
```
spl <- sample.split(df$Churn, SplitRatio = .70)
```

[59]:
```
training <- df[spl==T,]

testing <- df[spl==F,]
```

[60]:
```
#Dimension of training and testing datasets.

noquote("Dimension of training dataset:"); dim(training)

noquote("Dimension of testing dataset:"); dim(testing)
```

```
[1] Dimension of training dataset:
```

1. 8635 2. 18

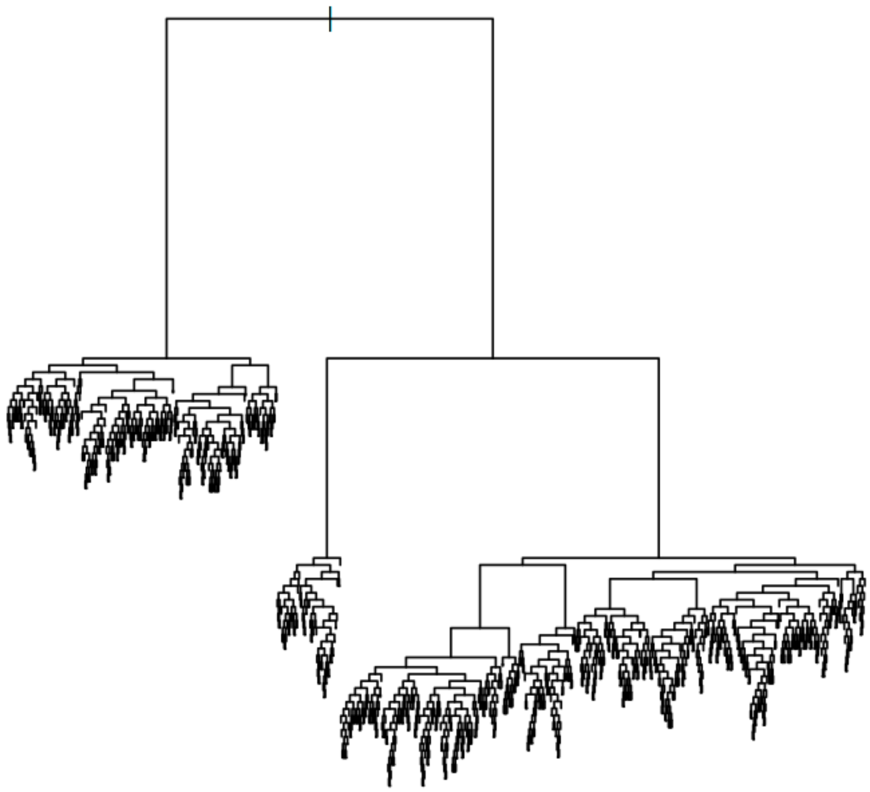
```
[1] Dimension of testing dataset:
```

1. 3700 2. 18

12 Full grown tree model concept

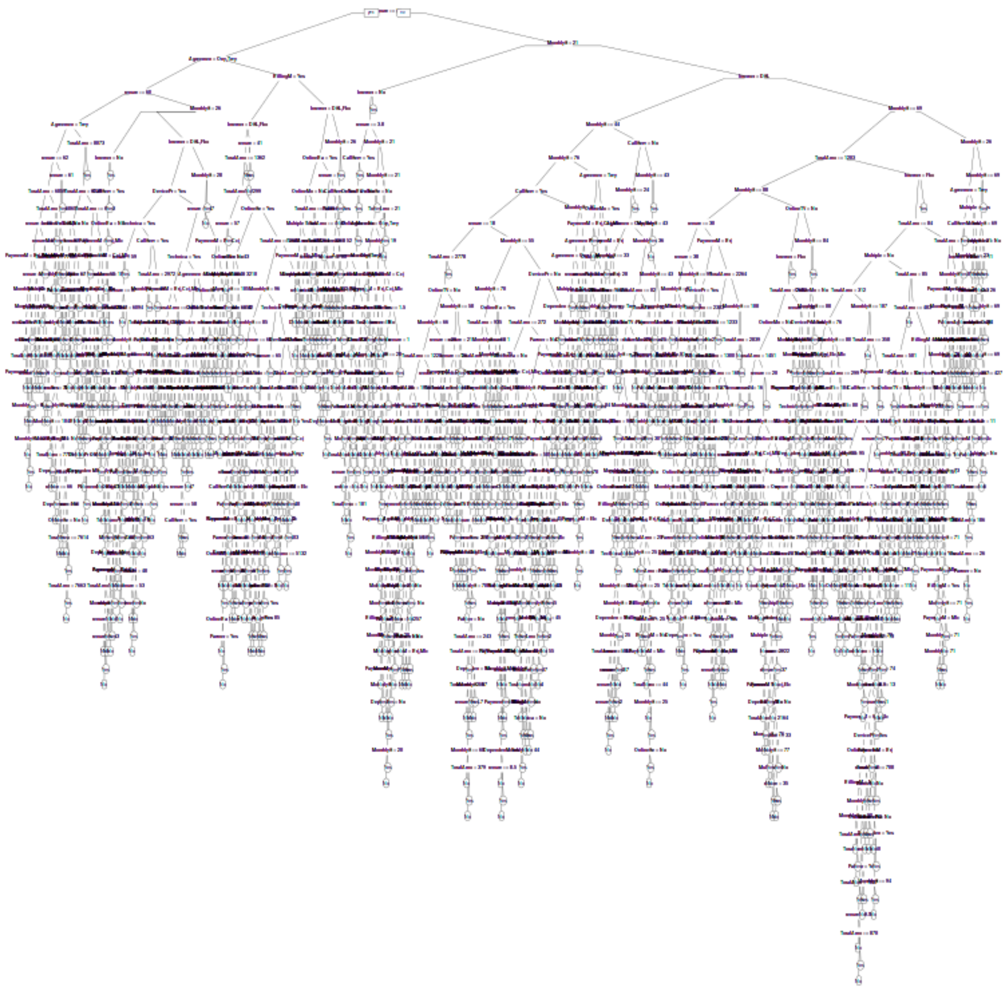
[61]:
```
#full grown tree model

full_model_tree <- rpart(Churn~., training, method = "class",minsplit=0,cp=0)
```

[62]:
```
#plotting model

plot(full_model_tree)
```



[63]:
```
prp(full_model_tree)
```

```
Warning message:
"labs do not fit even at cex 0.15, there may be some overplotting"
```



13 Prediction using the fully grown tree

[64]:
```
#Prediction using testing dataset

full_model_pred <- predict(full_model_tree, newdata = testing, type = "class")
```

[65]:
```
#Predictions

full_model_pred
```

1 No **5** No **25** No **29** No **30** Yes **51** No **55** No **56** No **58** No **62** No **67** No **68** No **71** No **76** No **78** No **80** No **81** No **83** No **84** No **86** No **87** Yes **96** No **97** Yes **98** No **100** Yes **102** Yes **110** No **111**

No **115** No **116** No **120** Yes **124** No **125** No **126** No **127** No **128** No **129** Yes **132** No **134** No **143** No **147** No **149** No **150** No **153** No **155** No **157** No **161** No **162** No **164** No **167** No **168** Yes **170** No **174** No **182** No **186** No **189** No **190** No **193** No **200** Yes **201** Yes **206** No **212** No **213** No **216** No **222** No **230** No **231** No **232** No **235** No **237** No **242** No **245** No **250** No **252** No **254** No **257** No **258** No **259** No **261** Yes **265** No **267** No **270** No **273** No **277** No **281** No **282** No **283** Yes **284** No **285** No **288** No **289** No **293** No **299** No **300** Yes **301** No **302** No **309** No **313** No **314** No **316** No **320** No **321** No **324** No **327** Yes **332** No **336** No **337** No **339** No **340** No **341** No **343** No **354** Yes **360** No **362** No **370** No **374** No **379** No **389** No **391** No **393** No **395** No **399** Yes **400** No **407** No **411** No **412** Yes **413** No **415** No **418** No **419** No **420** No **423** No **430** No **432** No **433** No **437** No **438** No **447** No **448** No **455** No **457** Yes **465** Yes **467** No **473** No **483** No **484** No **488** No **489** No **492** No **495** No **498** No **500** No **501** No **503** No **504** No **510** No **520** No **527** No **528** No **533** No **536** No **545** No **547** No **551** No **554** Yes **558** No **559** No **560** No **563** No **571** No **573** Yes **584** Yes **587** No **590** No **594** No **603** No **605** No **606** No **609** No **610** No **624** No **632** No **635** No **639** No **641** No **647** No **650** No **652** No **653** No **654** No **655** No **657** No **658** No **659** No **661** No **662** No **672** No **673** No **674** Yes **675** No **687** **689** Yes **690** Yes **691** Yes **692** Yes **693** Yes **694** Yes **696** No **698** No **706** Yes **709** Yes **710** No **711** Yes **714** No **722** Yes **723** Yes **724** Yes **728** Yes **729** Yes **730** Yes **733** Yes **734** Yes **738** Yes **740** Yes **750** Yes **751** Yes **770** Yes **773** Yes **784** Yes **785** Yes **787** Yes **788** Yes **793** Yes **794** Yes **799** Yes **800** Yes **802** No **804** Yes **805** Yes **806** Yes **808** Yes **815** Yes **821** Yes **823** Yes **829** Yes **833** Yes **835** Yes **838** Yes **843** Yes **844** Yes **847** Yes **850** Yes **851** Yes **853** No **854** Yes **858** Yes **859** Yes **861** No **862** Yes **870** Yes **871** No **874** Yes **875** Yes **877** Yes **878** Yes **880** Yes **892** Yes **897** Yes **900** Yes **901** Yes **904** Yes **905** No **911** Yes **914** Yes **916** Yes **917** Yes **918** Yes **921** Yes **925** Yes **930** Yes **932** Yes **933** Yes **935** Yes **941** Yes **949** Yes **952** Yes **959** Yes **961** Yes **963** Yes **964** Yes **967** Yes **969** Yes **970** Yes **973** Yes **984** Yes **987** Yes **996** Yes **997** No **998** Yes **1000** Yes **1012** Yes **1014** Yes **1016** Yes **1019** No **1020** Yes **1021** Yes **1027** No **1037** Yes **1038** Yes **1040** Yes **1042** Yes **1046** Yes **1047** No **1050** Yes **1054** Yes **1057** Yes **1062** Yes **1064** No **1066** Yes **1067** Yes **1080** Yes **1086** Yes **1087** Yes **1088** Yes **1090** Yes **1091** Yes **1092** Yes **1102** Yes **1105** No **1111** Yes **1114** Yes **1118** Yes **1119** No **1120** Yes **1122** Yes **1125** Yes **1126** Yes **1128** Yes **1129** Yes **1141** Yes **1144** Yes **1149** Yes **1153** Yes **1155** Yes **1156** Yes **1160** No **1162** Yes **1165** No **1168** Yes **1170** Yes **1172** Yes **1176** Yes **1178** Yes **1181** Yes **1183** Yes **1186** Yes **1187** Yes **1193** Yes **1197** Yes **1199** Yes **1203** No **1207** Yes **1208** Yes **1209** Yes **1210** Yes **1211** No **1217** Yes **1221** Yes **1223** Yes **1227** Yes **1228** Yes **1230** Yes **1233** Yes **1238** Yes **1239** Yes **1244** Yes **1247** Yes **1248** Yes **1252** Yes **1253** Yes **1257** Yes **1260** Yes **1262** Yes **1268** Yes **1270** Yes **1271** No **1273** Yes **1274** Yes **1276** Yes **1277** Yes **1278** Yes **1285** Yes **1287** No **1290** Yes **1291** Yes **1293** Yes **1295** Yes **1299** Yes **1300** Yes **1303** Yes **1315** Yes

Levels: 1. 'No' 2. 'Yes'

14 Accuracy and Confusion Matrix

[66]:
```r
#confusion matrix

conf_matrix_full_model <- table(testing$Churn, full_model_pred)

conf_matrix_full_model
```

```
    full_model_pred
```

```
      No  Yes
No  1814  204
Yes  350 1332
```

[67]:
```
#Print out the accuracy

sum(diag(conf_matrix_full_model))/sum(conf_matrix_full_model)*100
```

85.027027027027

15 cp implementation

"cp" stand for complexity parametr is used to control the size of the decision tree and to select the optimal tree size.

Smaller cp value will lead to bigger trees/complexity increases

Higher cp values will lead to smaller trees

[68]:
```
library(caret)
library(e1071)
```

[69]:
```
numfolds <- trainControl(method = "cv", number = 100)

#trainControl control parametrs for train
```

[70]:
```
cpGrid <- expand.grid(cp=seq(0.01,.05,0.01))
```

[71]:
```
cpGrid
```

A data.frame: 5 × 1

cp <dbl>
0.01
0.02
0.03
0.04
0.05

16 Checking the cross validation accuracy for cp parameters.

[72]:
```
train(Churn~., data = training, method = "rpart", trControl = numfolds,␣
↪tuneGrid = cpGrid)
```

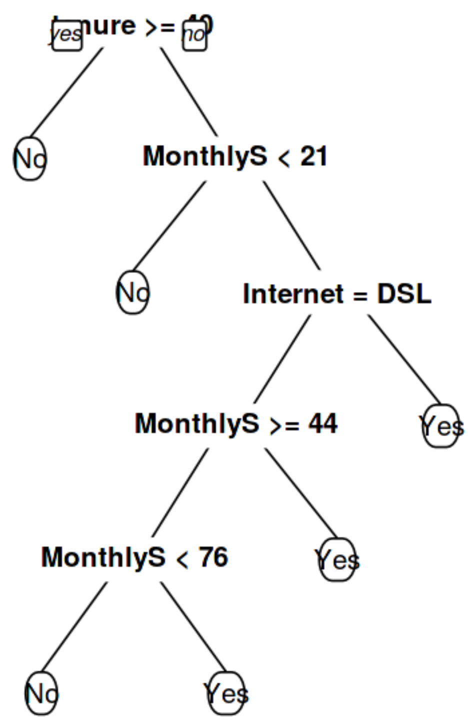
```
CART

8635 samples
  17 predictor
   2 classes: 'No', 'Yes'
```

```
No pre-processing
Resampling: Cross-Validated (100 fold)
Summary of sample sizes: 8549, 8549, 8548, 8549, 8549, 8549, …
Resampling results across tuning parameters:

  cp    Accuracy   Kappa
  0.01  0.7486109  0.4913721
  0.02  0.7209538  0.4482484
  0.03  0.7126031  0.4331189
  0.04  0.7116728  0.4317772
  0.05  0.7116728  0.4317772

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01.
```

17 Pruning the full_model_tree(fully grown model)

[73]:
```
pruned_tree <- prune(full_model_tree, cp = 0.01)
```

[74]:
```
prp(pruned_tree)
```



[75]:
```
printcp(pruned_tree)
```

```
Classification tree:
rpart(formula = Churn ~ ., data = training, method = "class",
    minsplit = 0, cp = 0)

Variables actually used in tree construction:
[1] InternetConnection    MonthlyServiceCharges tenure

Root node error: 3925/8635 = 0.45455

n= 8635
```

```
        CP nsplit rel error  xerror     xstd
1 0.238726      0   1.00000 1.00000 0.011789
2 0.140382      1   0.76127 0.80815 0.011413
3 0.022803      2   0.62089 0.63338 0.010720
4 0.020637      4   0.57529 0.58522 0.010461
5 0.010000      5   0.55465 0.55516 0.010283
```

18 Prediction using the pruned tree

[76]:
```r
pruned_model_pred <- predict(pruned_tree, newdata = testing, type = "class")
```

[77]:
```r
#predictions
pruned_model_pred
```

1 No 5 Yes 25 No 29 No 30 Yes 51 No 55 No 56 No 58 No 62 Yes 67 No 68 No 71 No 76 No 78 No 80 Yes 81 No 83 No 84 Yes 86 No 87 Yes 96 No 97 Yes 98 No 100 No 102 Yes 110 No 111 Yes 115 No 116 Yes 120 Yes 124 No 125 No 126 No 127 No 128 Yes 129 No 132 No 134 No 143 No 147 Yes 149 No 150 No 153 No 155 Yes 157 No 161 No 162 No 164 No 167 No 168 Yes 170 No 174 No 182 No 186 No 189 No 190 No 193 No 200 Yes 201 Yes 206 No 212 No 213 No 216 No 222 No 230 No 231 No 232 No 235 No 237 No 242 Yes 245 No 250 No 252 No 254 No 257 No 258 No 259 No 261 No 265 No 267 Yes 270 No 273 No 277 No 281 Yes 282 No 283 No 284 Yes 285 No 288 Yes 289 Yes 293 No 299 No 300 No 301 No 302 No 309 No 313 No 314 Yes 316 No 320 No 321 No 324 Yes 327 No 332 No 336 No 337 No 339 Yes 340 Yes 341 No 343 No 354 Yes 360 No 362 No 370 No 374 No 379 Yes 389 Yes 391 No 393 No 395 Yes 399 Yes 400 No 407 No 411 No 412 No 413 Yes 415 No 418 No 419 No 420 Yes 423 No 430 Yes 432 No 433 No 437 No 438 No 447 No 448 No 455 No 457 Yes 465 No 467 Yes 473 No 483 No 484 No 488 Yes 489 No 492 No 495 No 498 No 500 No 501 Yes 503 No 504 Yes 510 Yes 520 No 527 No 528 No 533 No 536 No 545 No 547 No 551 No 554 No 558 No 559 No 560 No 563 No 571 No 573 No 584 No 587 No 590 No 594 No 603 No 605 Yes 606 No 609 No 610 No 624 Yes 632 No 635 No 639 Yes 641 No 647 No 650 No 652 No 653 Yes 654 No 655 Yes 657 No 658 No 659 No 661 Yes 662 No 672 No 673 Yes 674 Yes 675 No 687 689 Yes 690 No 691 Yes 692 Yes 693 Yes 694 Yes 696 Yes 698 Yes 706 Yes 709 Yes 710 No 711 Yes 714 No 722 Yes 723 Yes 724 Yes 728 Yes 729 Yes 730 Yes 733 Yes 734 Yes 738 Yes 740 Yes 750 Yes 751 Yes 770 Yes 773 Yes 784 Yes 785 Yes 787 Yes 788 Yes 793 Yes 794 Yes 799 No 800 Yes 802 No 804 No 805 No 806 Yes 808 Yes 815 Yes 821 Yes 823 Yes 829 Yes 833 Yes 835 Yes 838 Yes 843 Yes 844 Yes 847 Yes 850 Yes 851 No 853 No 854 Yes 858 No 859 Yes 861 Yes 862 No 870 Yes 871 Yes 874 No 875 Yes 877 No 878 Yes 880 Yes 892 Yes 897 Yes 900 Yes 901 Yes 904 Yes 905 No 911 No 914 Yes 916 Yes 917 No 918 Yes 921 Yes 925 Yes 930 Yes 932 Yes 933 Yes 935 Yes 941 Yes 949 Yes 952 Yes 959 Yes 961 Yes 963 Yes 964 Yes 967 No 969 Yes 970 Yes 973 Yes 984 Yes 987 Yes 996 No 997 Yes 998 No 1000 No 1012 Yes 1014 Yes 1016 Yes 1019 Yes 1020 Yes 1021 Yes 1027 Yes 1037 Yes 1038 No 1040 Yes 1042 Yes 1046 Yes 1047 Yes 1050 Yes 1054 Yes 1057 Yes 1062 Yes 1064 No 1066 No 1067 Yes 1080 Yes 1086 Yes 1087 Yes 1088 Yes 1090 Yes 1091 No 1092 Yes 1102 Yes 1105 Yes 1111 Yes 1114 Yes 1118 Yes 1119 No 1120 Yes 1122 Yes 1125 Yes 1126 Yes 1128 Yes 1129 Yes 1141 No 1144 Yes 1149 Yes 1153 Yes 1155 No 1156 Yes 1160 No 1162 Yes 1165 Yes 1168 Yes 1170 Yes 1172 Yes 1176 Yes 1178 Yes 1181 No 1183 Yes 1186 Yes

1187 No **1193** No **1197** Yes **1199** Yes **1203** Yes **1207** Yes **1208** Yes **1209** Yes **1210** Yes **1211** No **1217** Yes **1221** Yes **1223** Yes **1227** Yes **1228** Yes **1230** No **1233** Yes **1238** Yes **1239** Yes **1244** No **1247** Yes **1248** No **1252** Yes **1253** Yes **1257** Yes **1260** Yes **1262** No **1268** Yes **1270** Yes **1271** No **1273** Yes **1274** Yes **1276** Yes **1277** Yes **1278** No **1285** Yes **1287** No **1290** Yes **1291** Yes **1293** Yes **1295** No **1299** Yes **1300** Yes **1303** Yes **1315** Yes

Levels: 1. 'No' 2. 'Yes'

19 Confusion Matrix and Accuracy

[78]:
```
#Confusion Matrix
conf_matrix_pruned_model <- table(testing$Churn,pruned_model_pred)
conf_matrix_pruned_model
```

```
     pruned_model_pred
        No  Yes
  No  1568  450
  Yes  456 1226
```

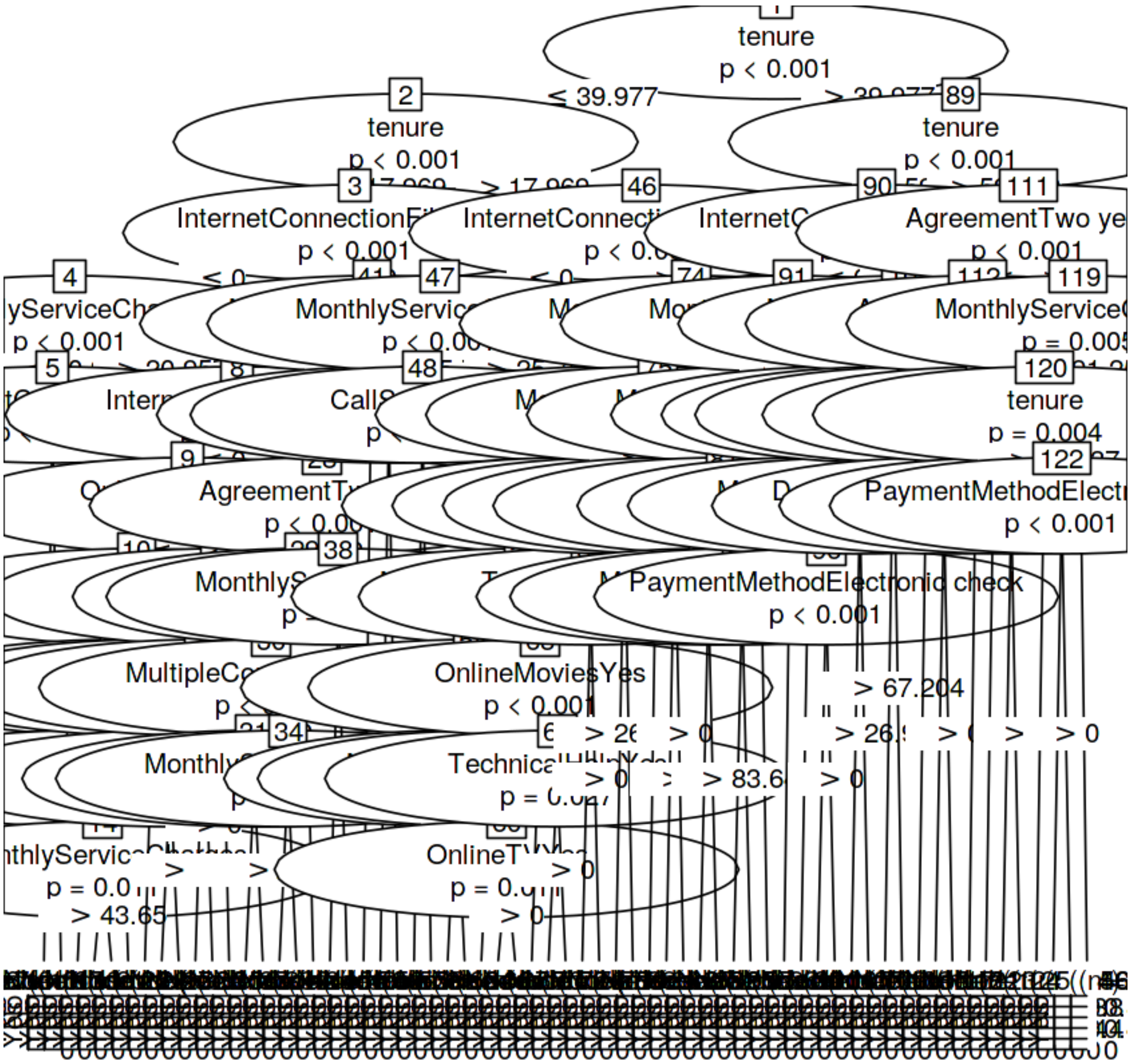
[79]:
```
#Print out the Accuracy
sum(diag(conf_matrix_pruned_model))/ sum(conf_matrix_pruned_model)
```

0.755135135135135

20 Another tree by ctree2 technique called classification tree

[80]:
```
model <- train(Churn~., data= training, method = "ctree2",
               trControl = trainControl("cv", number = 10),
               tuneGrid = expand.grid(maxdepth = 10, mincriterion = 0.95))
```

[81]:
```
plot(model$finalModel)
```



21 Prediction using the fully grown tree

[82]:
```
model_pred <- predict(model, newdata = testing, type = "raw")
#we can use "prob" also in type for probability
```

[83]:
```
#predictions
model_pred
```

1. No 2. Yes 3. No 4. No 5. Yes 6. No 7. No 8. No 9. No 10. No 11. No 12. No 13. No 14. No 15. No 16. No 17. No 18. No 19. No 20. No 21. Yes 22. No 23. Yes 24. No 25. No 26. Yes 27. No 28. Yes 29. No 30. No 31. No 32. No 33. No 34. No 35. No 36. No 37. No 38. No 39. No 40. No 41. Yes 42. No 43. No 44. No 45. No 46. No 47. No 48. No 49. No 50. No 51. Yes 52. No 53. No 54. No

55. No 56. No 57. No 58. No 59. Yes 60. Yes 61. No 62. No 63. No 64. No 65. No 66. Yes 67. No 68. No 69. No 70. No 71. Yes 72. No 73. No 74. No 75. No 76. No 77. No 78. No 79. No 80. No 81. No 82. No 83. No 84. No 85. Yes 86. Yes 87. No 88. No 89. No 90. No 91. Yes 92. No 93. No 94. No 95. No 96. No 97. No 98. No 99. No 100. No 101. No 102. No 103. No 104. No 105. No 106. No 107. No 108. No 109. No 110. No 111. No 112. No 113. No 114. No 115. No 116. No 117. Yes 118. No 119. No 120. No 121. Yes 122. No 123. No 124. No 125. No 126. No 127. No 128. No 129. No 130. No 131. No 132. No 133. Yes 134. No 135. No 136. No 137. No 138. No 139. Yes 140. No 141. Yes 142. No 143. No 144. No 145. No 146. No 147. No 148. No 149. No 150. No 151. No 152. No 153. No 154. No 155. No 156. Yes 157. No 158. No 159. No 160. No 161. No 162. No 163. No 164. No 165. Yes 166. No 167. No 168. No 169. No 170. No 171. No 172. No 173. No 174. No 175. No 176. No 177. No 178. No 179. No 180. No 181. No 182. No 183. No 184. No 185. No 186. No 187. No 188. No 189. Yes 190. No 191. No 192. No 193. No 194. No 195. No 196. No 197. No 198. No 199. No 200. No 201. 202. Yes 203. Yes 204. Yes 205. Yes 206. Yes 207. Yes 208. No 209. No 210. Yes 211. Yes 212. No 213. No 214. Yes 215. Yes 216. Yes 217. Yes 218. Yes 219. Yes 220. Yes 221. Yes 222. Yes 223. Yes 224. No 225. No 226. Yes 227. Yes 228. Yes 229. Yes 230. Yes 231. Yes 232. Yes 233. Yes 234. Yes 235. No 236. Yes 237. No 238. No 239. No 240. No 241. Yes 242. Yes 243. Yes 244. Yes 245. Yes 246. Yes 247. Yes 248. Yes 249. Yes 250. Yes 251. Yes 252. Yes 253. No 254. No 255. Yes 256. Yes 257. Yes 258. Yes 259. Yes 260. Yes 261. Yes 262. No 263. No 264. Yes 265. Yes 266. Yes 267. Yes 268. Yes 269. No 270. Yes 271. Yes 272. No 273. Yes 274. Yes 275. Yes 276. No 277. Yes 278. Yes 279. Yes 280. Yes 281. Yes 282. Yes 283. Yes 284. Yes 285. Yes 286. Yes 287. Yes 288. Yes 289. No 290. Yes 291. No 292. Yes 293. Yes 294. Yes 295. Yes 296. Yes 297. No 298. No 299. No 300. No 301. Yes 302. Yes 303. Yes 304. Yes 305. No 306. Yes 307. No 308. Yes 309. No 310. Yes 311. Yes 312. Yes 313. Yes 314. Yes 315. Yes 316. Yes 317. Yes 318. No 319. Yes 320. Yes 321. Yes 322. Yes 323. Yes 324. Yes 325. Yes 326. Yes 327. Yes 328. Yes 329. Yes 330. No 331. Yes 332. Yes 333. No 334. Yes 335. Yes 336. Yes 337. Yes 338. Yes 339. Yes 340. Yes 341. Yes 342. Yes 343. Yes 344. No 345. Yes 346. No 347. Yes 348. No 349. Yes 350. Yes 351. Yes 352. Yes 353. Yes 354. No 355. No 356. Yes 357. Yes 358. No 359. Yes 360. No 361. No 362. Yes 363. Yes 364. Yes 365. Yes 366. No 367. Yes 368. Yes 369. Yes 370. Yes 371. Yes 372. No 373. No 374. Yes 375. Yes 376. Yes 377. Yes 378. No 379. Yes 380. Yes 381. Yes 382. Yes 383. No 384. Yes 385. Yes 386. No 387. No 388. Yes 389. Yes 390. Yes 391. No 392. Yes 393. No 394. Yes 395. Yes 396. Yes 397. Yes 398. Yes 399. No 400. Yes 401. Yes

Levels: 1. 'No' 2. 'Yes'

22 Confusion matrix and Accuracy

[84]:
```
#Confusion Matrix
conf_matrix_model <- table(testing$Churn, model_pred)
conf_matrix_model
```

```
     model_pred
        No  Yes
  No  1801  217
  Yes  513 1169
```

[85]:
```
#Print out the accuracy
sum(diag(conf_matrix_model))/ sum(conf_matrix_model)
```

0.802702702702703

23 Random Forest

Installing package "randomForest" for Random Forest.

[86]:
```
#install.packages("randomForest")
library(randomForest)
```

```
randomForest 4.6-14

Type rfNews() to see new features/changes/bug fixes.


Attaching package: ‘randomForest’


The following object is masked from ‘package:rattle’:

    importance


The following object is masked from ‘package:ggplot2’:

    margin
```

[87]:
```
#Random Forest
rforest <- randomForest(Churn~., data = training, nodesize = 5, ntree = 200)
rforest
```

```

Call:
 randomForest(formula = Churn ~ ., data = training, nodesize = 5,      ntree = 
 ↪200)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 4

        OOB estimate of  error rate: 14.37%
Confusion matrix:
      No  Yes class.error
No  4211  499   0.1059448
Yes  742 3183   0.1890446
```

24 Prediction using Random forest

[88]:
```
rforest_pred <- predict(rforest, newdata = testing)
```

25 Confusion Matrix and Accuracy

[89]:
```
#Confusion Matrix
conf_matrix_RF <- table(testing$Churn, rforest_pred)

conf_matrix_RF
```

```
     rforest_pred
        No  Yes
  No  1830  188
  Yes  304 1378
```

[90]:
```
#print out the accuracy
sum(diag(conf_matrix_RF))/sum(conf_matrix_RF)*100
```

86.7027027027027

We get maximum accuracy in case of Random Forest. So our final model will be random forest with maximum accuracy than other created models.

[91]:
```
#                        ******This is the end of Project******
```