Author: Fangjung (Kristy) Lin

Purpose: Application of Tree methods and Random forests in Telecom Industry (Case 4)

Introduction and overview

- Using Tree methods and Random forests to predict customer churn in telecom industry

  Goal: To know more about what make the customer churn and predicting if he/she will churn or not.

  Data set: The data was downloaded from IBM Sample Data Sets for customer retention programs. You can download it from here: https://www.kaggle.com/blastchar/telco-customer-churn/download

  The data set includes information about:

  - Customers who left within the last month – the column is called Churn
  - Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
  - Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
  - Demographic info about customers – gender, age range, and if they have partners and dependents

Content of the analysis

Import necessary packages

Part1. Data Acquisition and Data Cleaning

Part2. Exploratory Data Analysis

Part3. Decision Tree

Part4. Random Forest

Part5. Conclusion

Import necessary packages

```
install.packages("tidyverse")
library(tidyverse)
install.packages("recipes")
library(recipes)
library(readr)
library(tidyverse)
library(modelr)
library(ggplot2)
install.packages("GGally")
library(GGally)
install.packages("corrplot")
library(corrplot)
library(gridExtra)
library(caret)
install.packages("e1071")
library("e1071")
install.packages("gplots")
install.packages("ROCR")
library(ROCR)
library(rpart)
install.packages('rpart.plot')
library(rpart.plot)
library(randomForest)
library(gplots)
install.packages('pROC')
library(pROC)
```

Part1. Data Acquisition and Data Cleaning

Data Acquisition

Walk through telecom data set

In [2]:
```
tele_churn <- read_csv("Desktop/churn.csv")
head(tele_churn)
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSuppor
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	N
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	N
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	N
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Ye
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	N
9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	Yes	N

Explore the structure of data

We can find that the data set contains 7043 observations and 21 variables

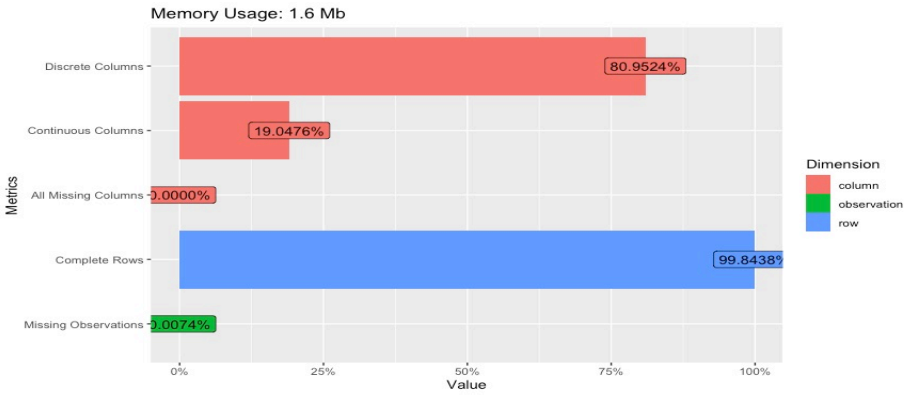
In [73]:
```
str(tele_churn)
```

```
tibble [7,043 × 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ customerID      : chr [1:7043] "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
 $ gender          : chr [1:7043] "Female" "Male" "Male" "Male" ...
 $ SeniorCitizen   : num [1:7043] 0 0 0 0 0 0 0 0 0 0 ...
 $ Partner         : chr [1:7043] "Yes" "No" "No" "No" ...
 $ Dependents      : chr [1:7043] "No" "No" "No" "No" ...
 $ tenure          : num [1:7043] 1 34 2 45 2 8 22 10 28 62 ...
 $ PhoneService    : chr [1:7043] "No" "Yes" "Yes" "No" ...
 $ MultipleLines   : chr [1:7043] "No phone service" "No" "No" "No phone service" ...
 $ InternetService : chr [1:7043] "DSL" "DSL" "DSL" "DSL" ...
 $ OnlineSecurity  : chr [1:7043] "No" "Yes" "Yes" "Yes" ...
 $ OnlineBackup    : chr [1:7043] "Yes" "No" "Yes" "No" ...
 $ DeviceProtection: chr [1:7043] "No" "Yes" "No" "Yes" ...
 $ TechSupport     : chr [1:7043] "No" "No" "No" "Yes" ...
 $ StreamingTV     : chr [1:7043] "No" "No" "No" "No" ...
 $ StreamingMovies : chr [1:7043] "No" "No" "No" "No" ...
 $ Contract        : chr [1:7043] "Month-to-month" "One year" "Month-to-month" "One year" ...
 $ PaperlessBilling: chr [1:7043] "Yes" "No" "Yes" "No" ...
 $ PaymentMethod   : chr [1:7043] "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automatic)" ...
 $ MonthlyCharges  : num [1:7043] 29.9 57 53.9 42.3 70.7 ...
 $ TotalCharges    : num [1:7043] 29.9 1889.5 108.2 1840.8 151.7 ...
 $ Churn           : chr [1:7043] "No" "No" "Yes" "No" ...
```

Overview of the data

Let's take an overview on some aspects about the data. We can find that most of the data is discrete, there is not a lot of missing values, but we need to deal with them by removing or imputing and no columns missing completely.

```
In [6]:
1 install.packages("DataExplorer")
2 library(DataExplorer)
3 plot_intro(tele_churn)
```



Check the missing value

We are checking if this dataset has any missing values.

```
In [74]:
1 unlist(lapply(tele_churn,function(x) sum(is.na(x))))
```

customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	11
Churn	0

I use lapply to check the number if missing values in each column. We found that there are 11 missing values in “TotalCharges” columns. So, let’s remove all rows with missing values.

Missing value deletion

In this step, I deleted the rows which have missing values.

```
In [75]:
1 tele_churn <- na.omit(tele_churn)
```

Part2. Exploratory Data Analysis

Let's see the overall summarize of our target variable

Summary of the data frame

```
In [9]: 1 summary(tele_churn)
```

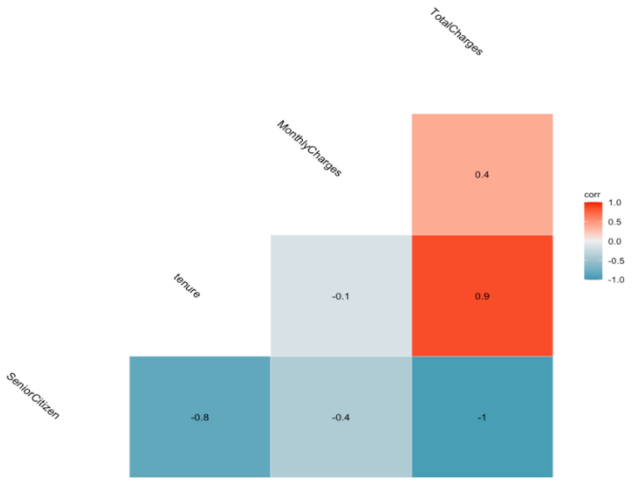
```
  customerID          gender          SeniorCitizen      Partner
 Length:7032        Length:7032        Min.   :0.0000   Length:7032
 Class :character   Class :character   1st Qu.:0.0000   Class :character
 Mode  :character   Mode  :character   Median :0.0000   Mode  :character
                                       Mean   :0.1624
                                       3rd Qu.:0.0000
                                       Max.   :1.0000
  Dependents            tenure      PhoneService       MultipleLines
 Length:7032        Min.   : 1.00   Length:7032        Length:7032
 Class :character   1st Qu.: 9.00   Class :character   Class :character
 Mode  :character   Median :29.00   Mode  :character   Mode  :character
                    Mean   :32.42
                    3rd Qu.:55.00
                    Max.   :72.00
 InternetService    OnlineSecurity     OnlineBackup       DeviceProtection
 Length:7032        Length:7032        Length:7032        Length:7032
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character


 TechSupport        StreamingTV        StreamingMovies      Contract
 Length:7032        Length:7032        Length:7032        Length:7032
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character


 PaperlessBilling   PaymentMethod      MonthlyCharges    TotalCharges
 Length:7032        Length:7032        Min.   : 18.25   Min.   :  18.8
 Class :character   Class :character   1st Qu.: 35.59   1st Qu.: 401.4
 Mode  :character   Mode  :character   Median : 70.35   Median :1397.5
                                       Mean   : 64.80   Mean   :2283.3
                                       3rd Qu.: 89.86   3rd Qu.:3794.7
                                       Max.   :118.75   Max.   :8684.8
    Churn
 Length:7032
 Class :character
 Mode  :character
```

From the summarize above, I find that the min/max difference between tenure is quite huge. The difference between MonthlyCharges and TotalCharges are quite huge. I need to have a deeper analysis in these variables to see if they can influence the customer churn.

Correlation among numeric variables

First, let's check the correlation between the numerical variables.

```
In [76]:
1 numeric.var <- sapply(tele_churn, is.numeric)
2 corr.matrix <- cor(tele_churn[,numeric.var])
3 options(repr.plot.width = 15, repr.plot.height = 8)
4 # corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", method = "number")
5
6 ggcorr(corr.matrix,name = "corr", label = TRUE, hjust = 1, label_size = 3.5,
7        angle = -45, size = 4)
```



- SeniorCitizen has a completely negative relation with TotalCharges. I assume that SeniorCitizen can be a factor that influence customer churn.
- tenure has a strong positive relation with totalcharges, which we need to take a deeper analysis to find if tenure has a positive relation with customer churn.

Creating new column tenure_bin and converting the tenure in years

In [77]:
```
#senior citizen is in integer form
tele_churn$SeniorCitizen <- as.factor(tele_churn$SeniorCitizen)

tele_churn <- mutate(tele_churn,tenure_bin=tenure)
tele_churn$tenure_bin[tele_churn$tenure_bin >= 0 & tele_churn$tenure_bin <= 12]   <- "0 - 1 years"
tele_churn$tenure_bin[tele_churn$tenure_bin >= 13 & tele_churn$tenure_bin <= 24]  <- "1 - 2 years"
tele_churn$tenure_bin[tele_churn$tenure_bin >= 25 & tele_churn$tenure_bin <= 36]  <- "2 - 3 years"
tele_churn$tenure_bin[tele_churn$tenure_bin >= 37 & tele_churn$tenure_bin <= 48]  <- "3 - 4 years"
tele_churn$tenure_bin[tele_churn$tenure_bin >= 49 & tele_churn$tenure_bin <= 60]  <- "4 - 5 years"
tele_churn$tenure_bin[tele_churn$tenure_bin >= 61 & tele_churn$tenure_bin <= 72]  <- "5 - 6 years"

#converting the newly created column into factor tenure_bin.
tele_churn$tenure_bin = as.factor(tele_churn$tenure_bin)
```

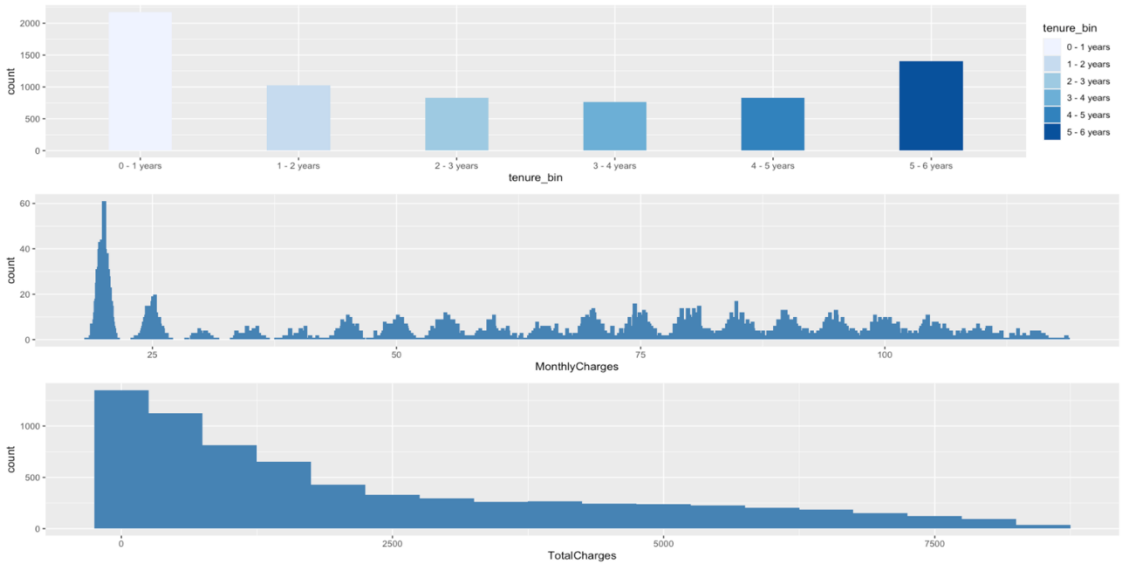
In the column MultipleLines modifying the No phone service -> No

In [78]:
```
tele_churn$MultipleLines[which(tele_churn$MultipleLines=="No phone service")]<-"No"
```

Graphical Representation of the Numerical Variables histograms

In [79]:
```
plot_A <- ggplot(tele_churn,aes(x = tenure_bin,fill=tenure_bin))+ geom_bar(width=0.4)+
                scale_fill_brewer(palette="Blues")
plot_B <- ggplot(tele_churn,aes(x = MonthlyCharges))+ geom_bar(fill="steelblue",width=0.4)
plot_C <- ggplot(tele_churn,aes(x = TotalCharges))+ geom_histogram(fill="steelblue",binwidth =500)

options(repr.plot.width = 15, repr.plot.height = 8)
grid.arrange(plot_A,plot_B,plot_C)
```

```
Warning message:
"position_stack requires non-overlapping x intervals"
```



From the tenure bin, we can find:

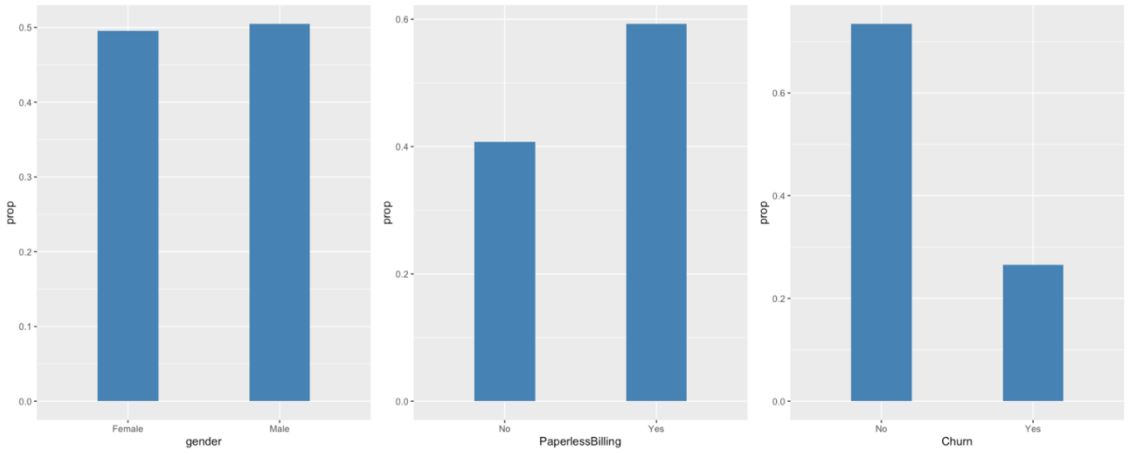
- Most people tenure year is between 0-1 years
- The trend of monthly charges and total charges is the similar.

Visualize the distribution of each variable

In [81]:
```
plot4 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = PaymentMethod,y=..prop..,group=2)
                                            ,fill="Steelblue",stat='count',width=0.4)
plot5 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = gender,y=..prop..,group=2)
                                            ,fill="Steelblue",stat='count',width=0.4)
plot6 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = SeniorCitizen,y=..prop..,group=2)
                                            ,fill="Steelblue",stat='count',width=0.4)
plot7 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = Partner,y=..prop..,group=2)
                                            ,fill="Steelblue",width=0.4,stat='count')
plot8 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = Dependents,y=..prop..,group=2)
                                            ,fill="Steelblue",width=0.4,stat='count')
plot9 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = PhoneService,y=..prop..,group=2),
                                           fill="Steelblue",width=0.4,stat='count')
plot10 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = MultipleLines,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot11 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = InternetService,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot12 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = OnlineSecurity,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot13 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = OnlineBackup,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot14 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = DeviceProtection,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot15 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = TechSupport,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot16 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = StreamingTV,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot17 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = StreamingMovies,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot18 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = Contract,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot19 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = PaperlessBilling,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
plot20 <- ggplot(data=tele_churn) + geom_bar(mapping = aes(x = Churn,y=..prop..,group=2),
                                             fill="Steelblue",width=0.4,stat='count')
```

Gender/PaperlessBilling/Churn

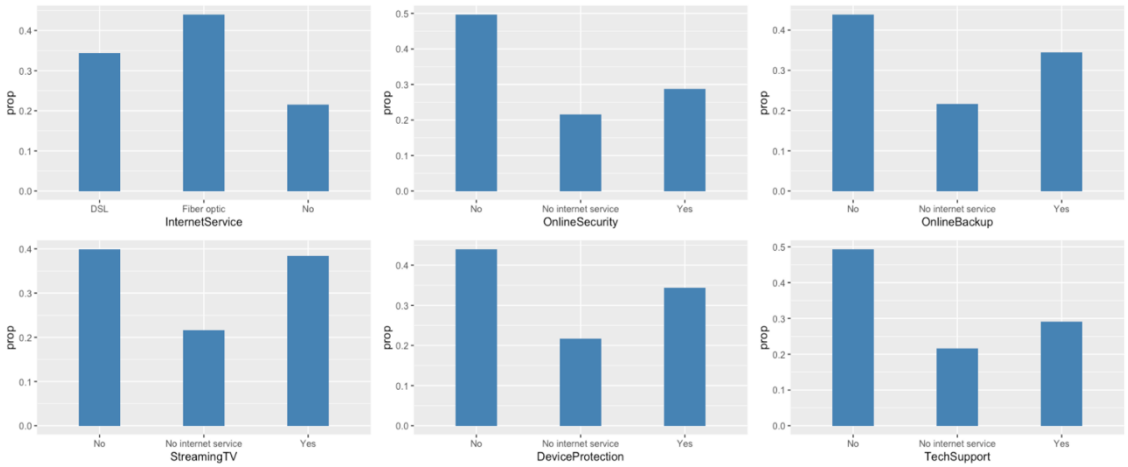
In [82]:
```
options(repr.plot.width = 15, repr.plot.height = 6)
grid.arrange(plot5,plot19,plot20,ncol=3)
```



- The data includes almost equal proportion of males and females.
- Almost 58% customers are on paperless billing.
- 26% of the customers have churned from the platform. From the distribution of customer churn, we can find that it's an imbalance data set.

InternetService/ OnlineSecurity/ OnlineBackup/ StreamingTV/ DeviceProtection/ TechSupport

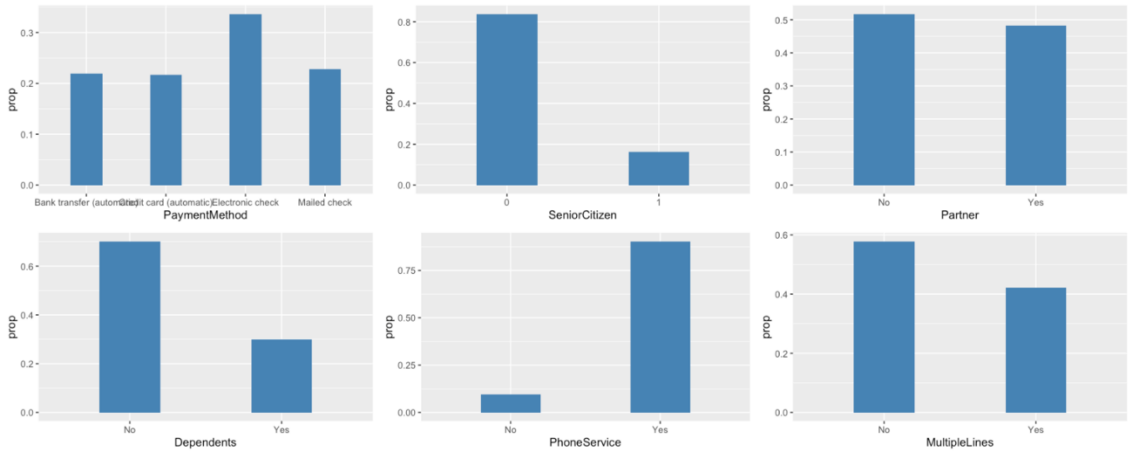
```
In [16]: 1 grid.arrange(plot11,plot12,plot13,plot16,plot14,plot15,nrow=2, ncol=3)
```



- Almost 40% of the customers have subscribed for the Fibre optic internet service.
- Almost 50% of the customers have no online security and almost 45% customers have no online backup.
- Almost 50% customers have no techsupport access and 40% have no streamingtv as a service.
- 45% of the customers have no service of device protection.

PaymentMethod/ SeniorCitizen/ Partner/ Dependents/ PhoneService/ MultipleLines

```
In [83]: 1 grid.arrange(plot4,plot6,plot7,plot8,plot9,plot10,nrow=2,ncol=3)
```



- Maximum number of customers have subscribered for electronic check for their payments.
- Very less i.e approx 20% of the customers are senior citizens.
- Equal number of customers with and without partners.
- 65% of the customers have no dependents.
- Almost 87% of the customers are with the phoneservice.

Summaries of Variables

In [84]:

```
#Gender
cat("Gender")
type_counts1 <- table(tele_churn$gender)
type_counts1 / sum(type_counts1) * 100

#SeniorCitizen
cat("SeniorCitizen")
type_counts2 <- table(tele_churn$SeniorCitizen)
type_counts2 / sum(type_counts2) * 100

#Partner
cat("Partner")
type_counts3 <- table(tele_churn$Partner)
type_counts3 / sum(type_counts3) * 100

#Dependents
cat("SeniorCitizen")
type_counts4 <- table(tele_churn$Dependents)
type_counts4 / sum(type_counts4) * 100

#PhoneService
cat("Dependents")
type_counts5 <- table(tele_churn$PhoneService)
type_counts5 / sum(type_counts5) * 100
```

```
Gender

  Female     Male
49.53072 50.46928

SeniorCitizen

       0        1
83.75995 16.24005

Partner

      No      Yes
51.74915 48.25085

SeniorCitizen

      No      Yes
70.15074 29.84926

Dependents

      No      Yes
 9.67008 90.32992
```

In [85]:

```
#MultipleLines
cat("MultipleLines")
type_counts6 <- table(tele_churn$MultipleLines)
type_counts6 / sum(type_counts6) * 100

#InternetService
cat("InternetService")
type_counts7 <- table(tele_churn$InternetService)
type_counts7 / sum(type_counts7) * 100

#OnlineSecurity
cat("OnlineSecurity")
type_counts8 <- table(tele_churn$OnlineSecurity)
type_counts8 / sum(type_counts8) * 100

#OnlineBackup
cat("OnlineBackup")
type_counts9 <- table(tele_churn$OnlineBackup)
type_counts9 / sum(type_counts9) * 100

#DeviceProtection
cat("DeviceProtection")
type_counts10 <- table(tele_churn$DeviceProtection)
type_counts10 / sum(type_counts10) * 100

#TechSupport
cat("TechSupport")
type_counts11 <- table(tele_churn$TechSupport)
type_counts11 / sum(type_counts11) * 100
```

```
MultipleLines

      No      Yes
57.80717 42.19283

InternetService

       DSL Fiber optic          No
   34.35722    44.02730    21.61547

OnlineSecurity

                  No No internet service                 Yes
            49.72981            21.61547            28.65472

OnlineBackup

                  No No internet service                 Yes
            43.89932            21.61547            34.48521

DeviceProtection

                  No No internet service                 Yes
            43.99886            21.61547            34.38567

TechSupport

                  No No internet service                 Yes
            49.37429            21.61547            29.01024
```

In [86]:

```
#StreamingTV
cat("StreamingTV")
type_counts12 <- table(tele_churn$StreamingTV)
type_counts12 / sum(type_counts12) * 100

#STreaming Movies
cat("STreaming Movies")
type_counts13 <- table(tele_churn$StreamingMovies)
type_counts13 / sum(type_counts13) * 100

#Contract
cat("Contract")
type_counts14 <- table(tele_churn$Contract)
type_counts14 / sum(type_counts14) * 100

#PaperlessBilling
cat("PaperlessBilling")
type_counts15 <- table(tele_churn$PaperlessBilling)
type_counts15 / sum(type_counts15) * 100

#PaymentMethod
cat("PaymentMethod")
type_counts16 <- table(tele_churn$PaymentMethod)
type_counts16 / sum(type_counts16) * 100

#Churn
cat("Churn")
type_counts17 <- table(tele_churn$Churn)
type_counts17 / sum(type_counts17) * 100
```

```
StreamingTV

                  No No internet service                 Yes
           39.94596            21.61547            38.43857

STreaming Movies

                  No No internet service                 Yes
           39.54778            21.61547            38.83675

Contract

Month-to-month       One year       Two year
      55.10523       20.93288       23.96189

PaperlessBilling

     No     Yes
40.7281 59.2719

PaymentMethod

Bank transfer (automatic)   Credit card (automatic)         Electronic check
                 21.92833                  21.62969                  33.63197
             Mailed check
                 22.81001

Churn

     No     Yes
73.4215 26.5785
```

Dependent variable Churn vs Tenure_bin

In [87]:
```
options(repr.plot.width = 15, repr.plot.height = 6)
ggplot(tele_churn, aes(x = tenure_bin , y = MonthlyCharges)) + geom_point(aes(colour=factor(Churn)))
ggplot(tele_churn, aes(x = MonthlyCharges , y = TotalCharges)) + geom_point(aes(colour=factor(Churn)))
```

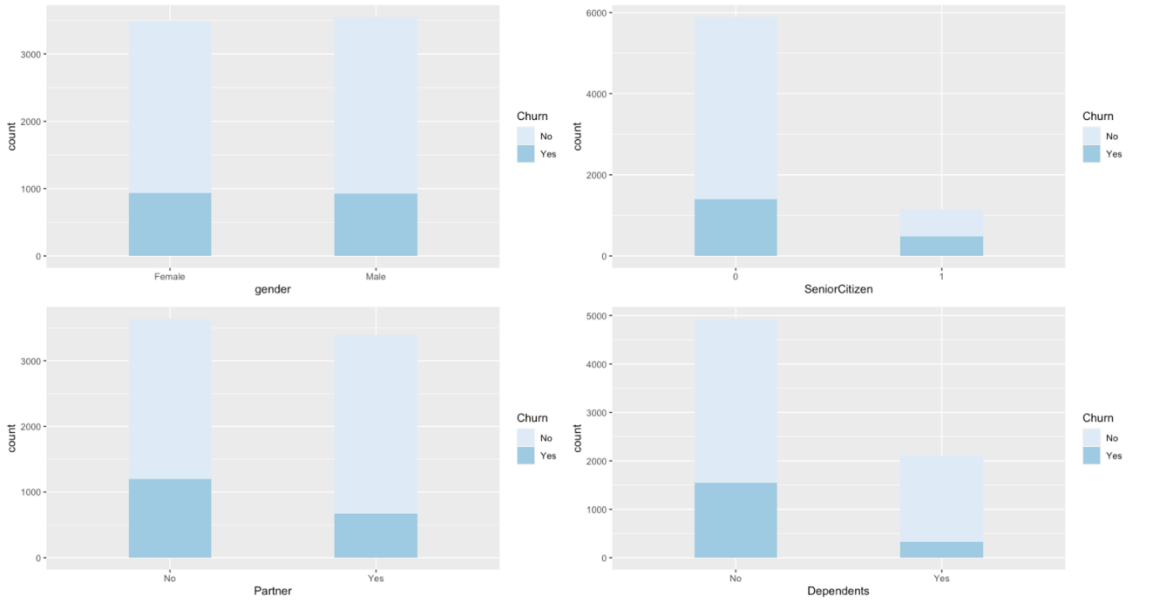
- Maximum Customers churned from the platform are the one having a tenure of 0-1 years.
- Maximum Churned customers have a Monthly charge more than $65.

Relationship between the variables (dependent variable Churn vs Categorical Variables)

In [88]:
```
#gender vs churn
plot_relation_1 =ggplot(tele_churn, aes(x = gender,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#SeniorCitizen vs Churn
plot_relation_2 =ggplot(tele_churn, aes(x = SeniorCitizen,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#partner vs Churn
plot_relation_3 =ggplot(tele_churn, aes(x = Partner,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#Dependents vs Churn
plot_relation_4 =ggplot(tele_churn, aes(x = Dependents,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#phoneservices vs Churn
plot_relation_5 =ggplot(tele_churn, aes(x = PhoneService,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#MultipleLines vs Churn
plot_relation_6 =ggplot(tele_churn, aes(x = MultipleLines,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#InternetServices vs Churn
plot_relation_7 =ggplot(tele_churn, aes(x = InternetService,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#OnlineSecurity vs Churn
plot_relation_8 =ggplot(tele_churn, aes(x = OnlineSecurity,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#Onlinebackup vs Churn
plot_relation_9 =ggplot(tele_churn, aes(x = OnlineBackup,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#DeviceProtection vs Churn
plot_relation_10 =ggplot(tele_churn, aes(x = DeviceProtection,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#TechSupport vs Churn
plot_relation_11 =ggplot(tele_churn, aes(x = TechSupport,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#StreamingTV vs Churn
plot_relation_12 =ggplot(tele_churn, aes(x = StreamingTV,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#StreamingMovies vs Churn
plot_relation_13 =ggplot(tele_churn, aes(x = StreamingMovies,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#Contract vs Churn
plot_relation_14 =ggplot(tele_churn, aes(x = Contract,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#PaperlessBilling vs Churn
plot_relation_15 =ggplot(tele_churn, aes(x = PaperlessBilling,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
#PaymentMethod vs Churn
plot_relation_16 =ggplot(tele_churn, aes(x = PaymentMethod,fill=Churn)) + geom_bar(width = 0.4)
+scale_fill_brewer(palette="Blues")
```

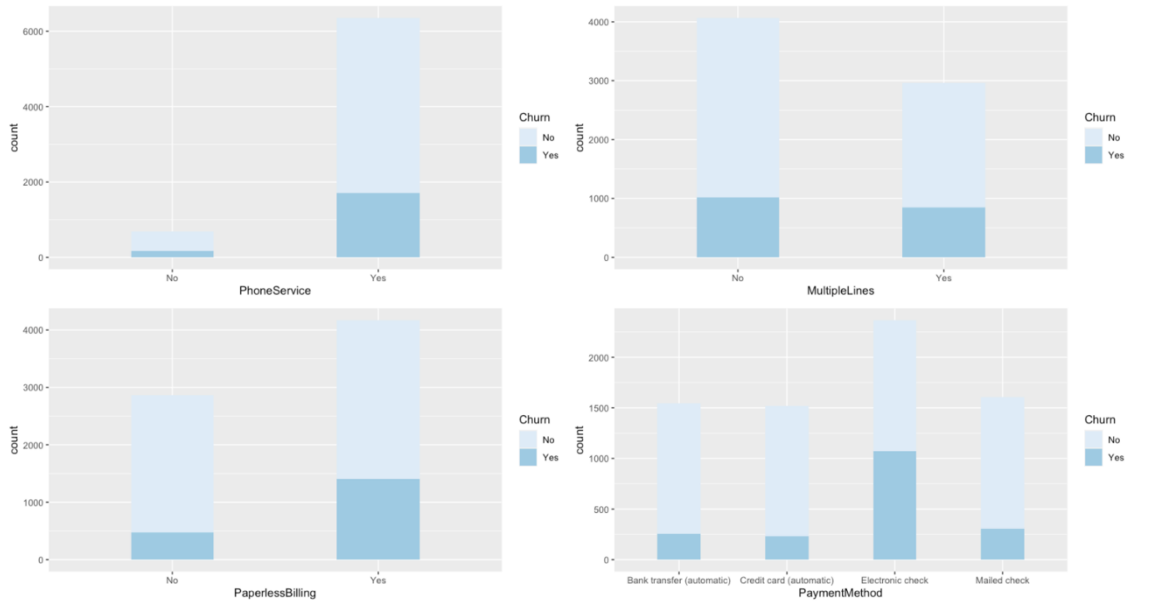
Visualize the result

In [89]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
grid.arrange(plot_relation_1,plot_relation_2,plot_relation_3,plot_relation_4)
```



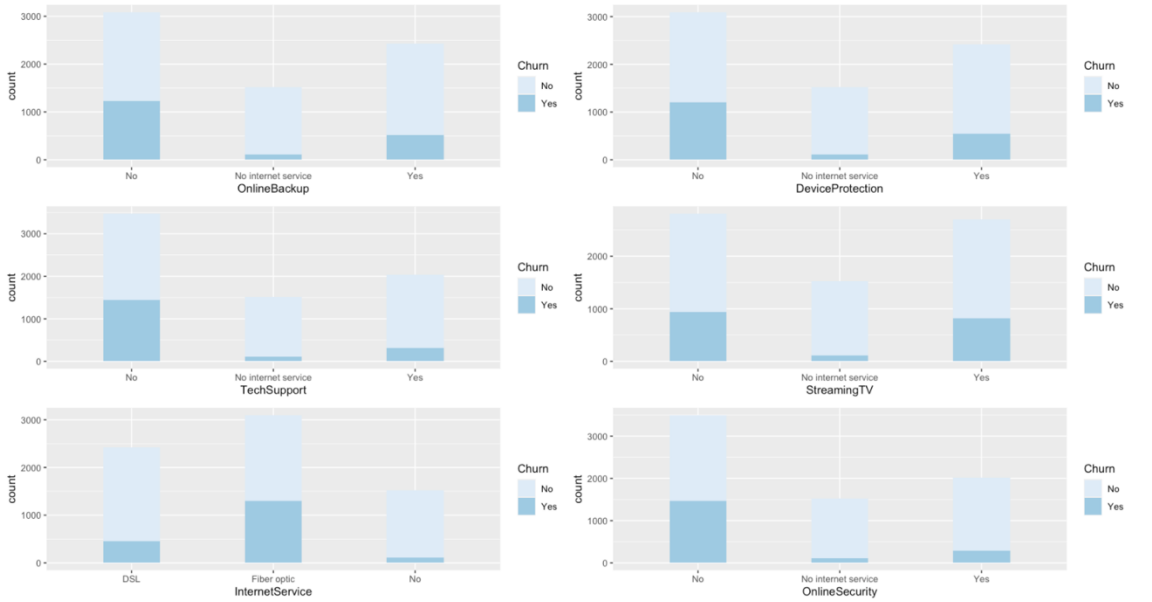
- We can see clearly that there is no specific gender correlated with Churned or not churned both genders have the same distribution for both 0 and 1
- The fraction of churned out of all SeniorCitizen is much bigger that the fraction of churned out of all non SeniorCitizen so being SeniorCitizen is indeed an important factor of churning
- Half the people have Partner and they have less probability of Churning than people who don't have as we can see the fraction of churned in Partner group is less than no Partner group
- Churn rate is more among the customers with no dependents. The ratio of churned people who don't have Dependents is 31%, in the people who have dependent is 15% only which is half the ratio which means that not having dependent increase the probability of churn

In [90]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
grid.arrange(plot_relation_5,plot_relation_6,plot_relation_15,plot_relation_16)
```



- Churn rate is more with customers having phone service.
- Churn rate is more with customers having paperless billing.
- Churn rate is more with customers having electronic check as the payment mode.

In [91]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
grid.arrange(plot_relation_9,plot_relation_10,plot_relation_11,plot_relation_12,plot_relation_7,plot_relation_8)
```



- Churn rate is more with customers having month to month contract.
- Churn rate is more with customers having no online security and techsupport.
- Churn rate is almost equal among the subscribers with or without the streamingtv.

Part3. Decision Tree

A decision tree can give a nice and easy visualization of the prediction rules. More specifically, rpart classification tree will be grown. No tuning, just a simple tree.

Data frame manipulation

In [92]:
```
tele_churn$churn_number <- 0
tele_churn$churn_number[tele_churn$Churn == 'Yes'] <- 1
tele_churn$Churn <- NULL
```

Convert categorical variables to factor

In [93]:
```
tele_churn$gender <- as.factor(tele_churn$gender)
tele_churn$SeniorCitizen <- as.factor(tele_churn$SeniorCitizen)
tele_churn$Partner <- as.factor(tele_churn$Partner)
tele_churn$Dependents <- as.factor(tele_churn$Dependents)
tele_churn$PhoneService <- as.factor(tele_churn$PhoneService)
tele_churn$MultipleLines <- as.factor(tele_churn$MultipleLines)
tele_churn$InternetService <- as.factor(tele_churn$InternetService)
tele_churn$OnlineSecurity <- as.factor(tele_churn$OnlineSecurity)
tele_churn$OnlineBackup <- as.factor(tele_churn$OnlineBackup)
tele_churn$DeviceProtection <- as.factor(tele_churn$DeviceProtection)
tele_churn$TechSupport <- as.factor(tele_churn$TechSupport)
tele_churn$StreamingTV <- as.factor(tele_churn$StreamingTV)
tele_churn$StreamingMovies <- as.factor(tele_churn$StreamingMovies)
tele_churn$Contract <- as.factor(tele_churn$Contract)
tele_churn$PaperlessBilling <- as.factor(tele_churn$PaperlessBilling)
tele_churn$PaymentMethod <- as.factor(tele_churn$PaymentMethod)
```

Remove unnecessary columns for modeling

We don't need the customerID and tenure (tenure from the dataset has already transformed the tenure values in tenure_bin).

In [94]:
```
tele_churn$customerID <- NULL
tele_churn$tenure <- NULL
```

Creating dummy variables

In [95]:
```
trainDummy <- dummyVars('~.', data = tele_churn, fullRank = F)
```

In [96]:
```
train <- as.data.frame(predict(trainDummy, tele_churn))
colnames(train)
```

'gender.Female' 'gender.Male' 'SeniorCitizen.0' 'SeniorCitizen.1' 'Partner.No' 'Partner.Yes' 'Dependents.No' 'Dependents.Yes' 'PhoneService.No' 'PhoneService.Yes' 'MultipleLines.No' 'MultipleLines.Yes' 'InternetService.DSL' 'InternetService.Fiber optic' 'InternetService.No' 'OnlineSecurity.No' 'OnlineSecurity.No internet service' 'OnlineSecurity.Yes' 'OnlineBackup.No' 'OnlineBackup.No internet service' 'OnlineBackup.Yes' 'DeviceProtection.No' 'DeviceProtection.No internet service' 'DeviceProtection.Yes' 'TechSupport.No' 'TechSupport.No internet service' 'TechSupport.Yes' 'StreamingTV.No' 'StreamingTV.No internet service' 'StreamingTV.Yes' 'StreamingMovies.No' 'StreamingMovies.No internet service' 'StreamingMovies.Yes' 'Contract.Month-to-month' 'Contract.One year' 'Contract.Two year' 'PaperlessBilling.No' 'PaperlessBilling.Yes' 'PaymentMethod.Bank transfer (automatic)' 'PaymentMethod.Credit card (automatic)' 'PaymentMethod.Electronic check' 'PaymentMethod.Mailed check' 'MonthlyCharges' 'TotalCharges' 'tenure_bin.0 - 1 years' 'tenure_bin.1 - 2 years' 'tenure_bin.2 - 3 years' 'tenure_bin.3 - 4 years' 'tenure_bin.4 - 5 years' 'tenure_bin.5 - 6 years' 'churn_number'

In [97]:
```
train$churn_number <- as.factor(ifelse(train$churn_number == 1, 'yes', 'no'))
```

Make sure the structure of data

In [99]:
```
str(tele_churn)
```

```
tibble [7,032 × 20] (S3: tbl_df/tbl/data.frame)
 $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
 $ SeniorCitizen   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
 $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
 $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
 $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1 ...
 $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
 $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
 $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
 $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
 $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
 $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
 $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
 $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
 $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
 $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
 $ MonthlyCharges  : num [1:7032] 29.9 57 53.9 42.3 70.7 ...
 $ TotalCharges    : num [1:7032] 29.9 1889.5 108.2 1840.8 151.7 ...
 $ tenure_bin      : Factor w/ 6 levels "0 - 1 years",..: 1 3 1 4 1 1 2 1 3 6 ...
 $ churn_number    : num [1:7032] 0 0 1 0 1 1 0 0 1 0 ...
 - attr(*, "na.action")= 'omit' Named int [1:11] 489 754 937 1083 1341 3332 3827 4381 5219 6671 ...
  ..- attr(*, "names")= chr [1:11] "489" "754" "937" "1083" ...
```

Splitting Training and Test Data

Data splitting into test and train into 70/30 ratio.

In [100]:
```
set.seed(123)
split <- sample(2,nrow(train),replace = T,prob = c(0.7,0.3))
traindf <- train[split == 1,]
testdf <- train[split == 2,]

dim(traindf)
dim(testdf)
```

```
4943  51
```

```
2089  51
```

Check whether any NA value exists or not

In [102]:
```
anyNA(tele_churn)
anyNA(traindf)
anyNA(testdf)
```

```
FALSE
```

```
FALSE
```

```
FALSE
```

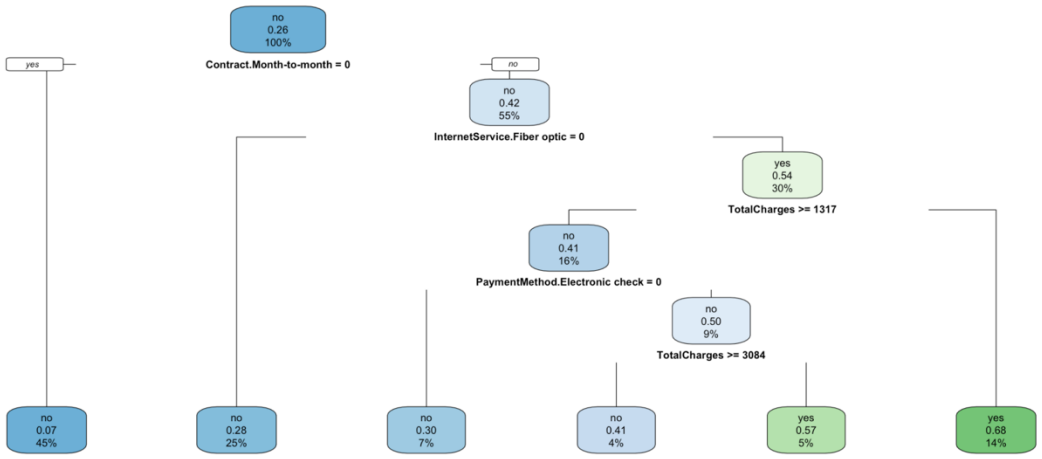
Train the model

I set up the method = class to make it a binary classification.

In [33]:
```
set.seed(123)
tree_df <- rpart(churn_number~., data = traindf, method = "class", parms = list(split = "gini"))
```

Visualize the model

In [34]:
```
options(repr.plot.width = 18, repr.plot.height = 8)
rpart.plot(tree_df)
```



I noticed here that the model didn't use a lot of the features, so this is an indicator that feature engineering is important in the problem

Predict the value

In [35]:
```
predict(tree_df, data = traindf, type = "class") -> traintree_pred1
predict(tree_df, data = traindf, type = "prob") -> traintree_prob1
predict(tree_df, newdata= testdf, type = "class") -> testtree_pred1
predict(tree_df, newdata = testdf, type = "prob") -> testtree_prob1
```

Confusion Matrix of Training Set

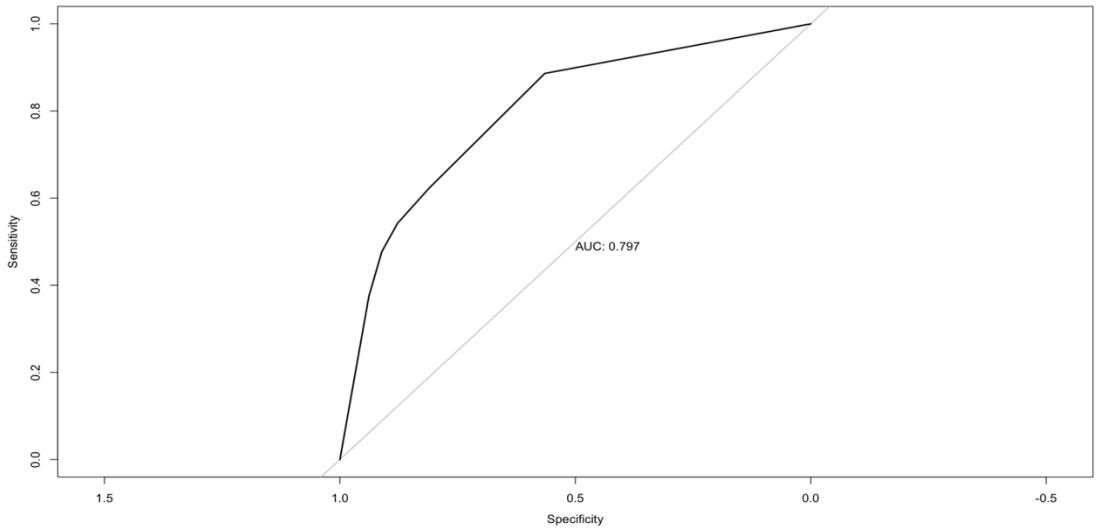
In [36]:
```
confusionMatrix(data = traintree_pred1, reference = traindf$churn_number)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3319  680
       yes  324  620

               Accuracy : 0.7969
                 95% CI : (0.7854, 0.808)
    No Information Rate : 0.737
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4255

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9111
            Specificity : 0.4769
         Pos Pred Value : 0.8300
         Neg Pred Value : 0.6568
             Prevalence : 0.7370
         Detection Rate : 0.6715
   Detection Prevalence : 0.8090
      Balanced Accuracy : 0.6940

       'Positive' Class : no
```

AUC-ROC Curve of Training Set

In [37]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
traintree_actual <- ifelse(traindf$churn_number == "yes", 1,0)
roc <- roc(traintree_actual, traintree_prob1[,2], plot= TRUE, print.auc=TRUE)
```

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



Confusing Matrix of Testing Set

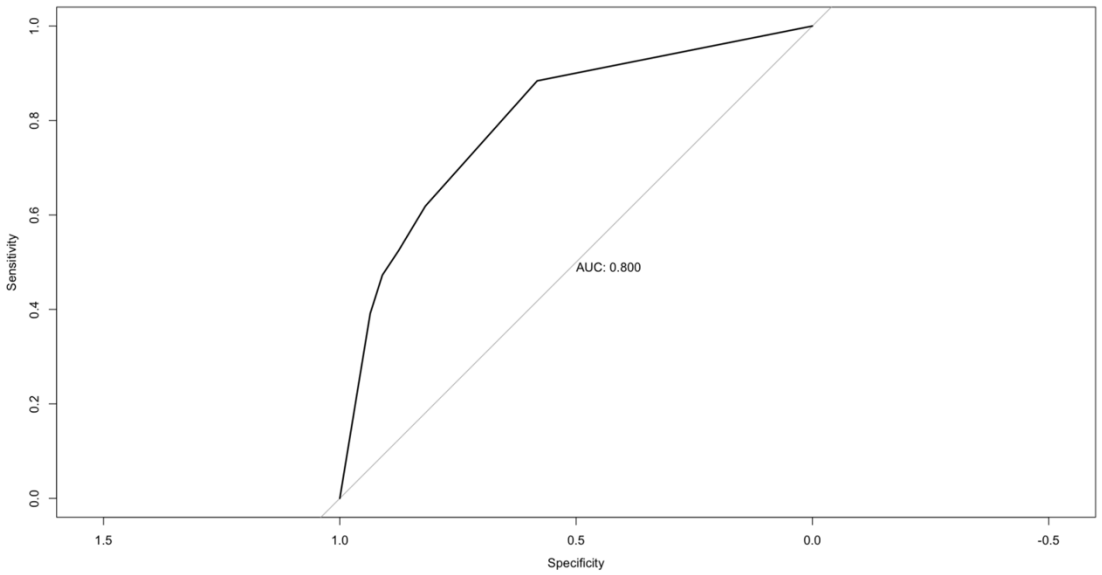
In [38]:
```
confusionMatrix(data = testtree_pred1, reference = testdf$churn_number)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1383  300
       yes  137  269

               Accuracy : 0.7908
                 95% CI : (0.7727, 0.8081)
    No Information Rate : 0.7276
    P-Value [Acc > NIR] : 1.597e-11

                  Kappa : 0.4203

 Mcnemar's Test P-Value : 9.225e-15

            Sensitivity : 0.9099
            Specificity : 0.4728
         Pos Pred Value : 0.8217
         Neg Pred Value : 0.6626
             Prevalence : 0.7276
         Detection Rate : 0.6620
   Detection Prevalence : 0.8056
      Balanced Accuracy : 0.6913

       'Positive' Class : no
```

AUC-ROC Curve of Testing Set

In [39]:
```
testtree_actual <- ifelse(testdf$churn_number == "yes", 1,0)
roc <- roc(testtree_actual, testtree_prob1[,2], plot = TRUE, print.auc = TRUE)
```

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



For the training set, the Accuracy is 0.7969 and the AUC is 0.797. For the testing set, the Accuracy is 0.7908 and the AUC is 0.80.

Part4. Random Forest

Data Preparation - In this part, I use the same data prepared for Classification Tree models to see if I can get a higher accuracy.

Train the model

In [40]:
```
names(traindf) <- make.names(names(traindf))
names(testdf) <- make.names(names(testdf))
rf_df <- randomForest(as.factor(churn_number) ~ ., data=traindf, importance = TRUE, ntree=500, do.trace=FALSE )
```

Predict the value

In [41]:
```
predict(rf_df, data = traindf, type = "class") -> trainrf_pred1
predict(rf_df, data = traindf, type = "prob") -> trainrf_prob1
predict(rf_df, newdata= testdf, type = "class") -> testrf_pred1
predict(rf_df, newdata = testdf, type = "prob") -> testrf_prob1
```

Confusion Matrix of training set

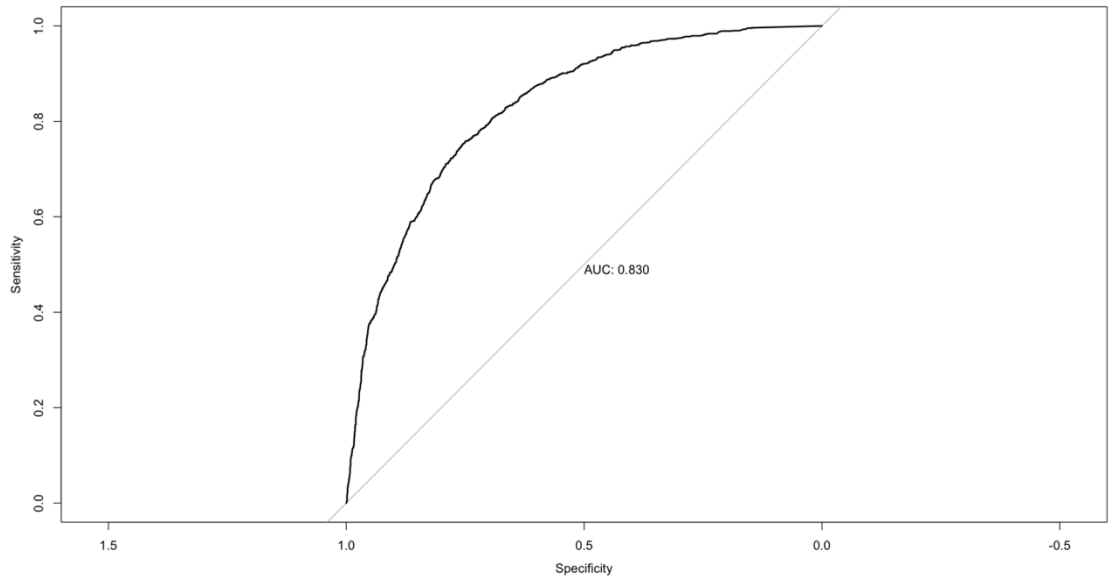
In [42]:
```
confusionMatrix(data = trainrf_pred1, reference = traindf$churn_number)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3278  654
       yes  365  646

               Accuracy : 0.7938
                 95% CI : (0.7823, 0.8051)
    No Information Rate : 0.737
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4273

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8998
            Specificity : 0.4969
         Pos Pred Value : 0.8337
         Neg Pred Value : 0.6390
             Prevalence : 0.7370
         Detection Rate : 0.6632
   Detection Prevalence : 0.7955
      Balanced Accuracy : 0.6984

       'Positive' Class : no
```

AUC-ROC Curve of training set

In [43]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
rf_df_actual <- ifelse(traindf$churn_number == "yes", 1,0)
roc <- roc(rf_df_actual, trainrf_prob1[,2], plot= TRUE, print.auc=TRUE)
```

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



Confusion Matrix of testing set

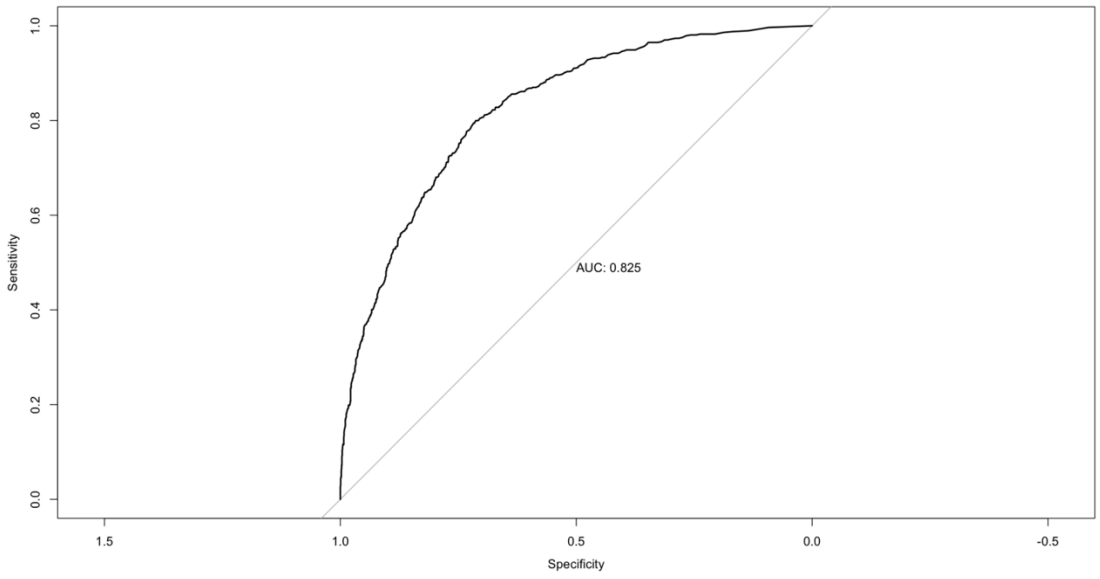
In [44]:
```
1 confusionMatrix(data = testrf_pred1, reference = testdf$churn_number)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1370  291
       yes  150  278

               Accuracy : 0.7889
                 95% CI : (0.7708, 0.8062)
    No Information Rate : 0.7276
    P-Value [Acc > NIR] : 6.410e-11

                  Kappa : 0.4227

 Mcnemar's Test P-Value : 2.617e-11

            Sensitivity : 0.9013
            Specificity : 0.4886
         Pos Pred Value : 0.8248
         Neg Pred Value : 0.6495
             Prevalence : 0.7276
         Detection Rate : 0.6558
   Detection Prevalence : 0.7951
      Balanced Accuracy : 0.6949

       'Positive' Class : no
```

AUC-ROC Curve of testing set

In [45]:
```
1 options(repr.plot.width = 15, repr.plot.height = 8)
2 rf_df_actual <- ifelse(testdf$churn_number == "yes", 1,0)
3 roc <- roc(rf_df_actual, testrf_prob1[,2], plot= TRUE, print.auc=TRUE)
```

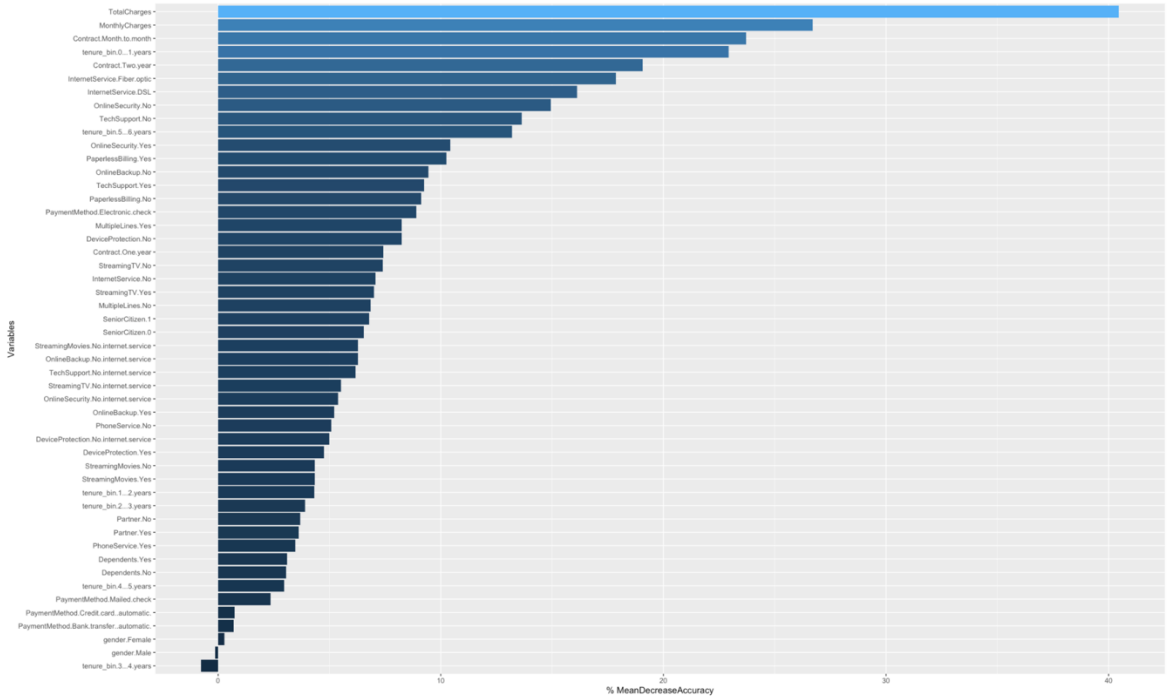
```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```



The training random forest model has the accuracy of 0.7957 and AUC of 0.829 for the training set. The testing random forest model has the Accuracy of 0.7927 and AUC of 0.828 for the testing set.

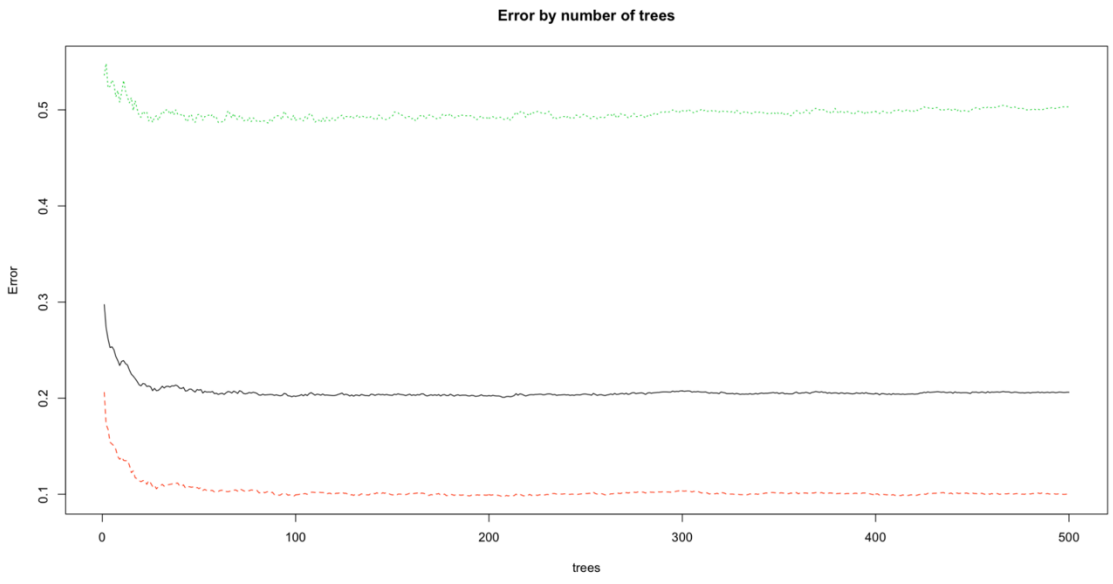
Plot the feature importance

In [46]:
```
options(repr.plot.width = 20, repr.plot.height = 12)
imp_rf_df <- importance(rf_df)
imp_DF <- data.frame(Variables = row.names(imp_rf_df), MeanDecreaseAccuracy = imp_rf_df[,3])
imp_DF <- imp_DF[order(imp_DF$MeanDecreaseAccuracy, decreasing = TRUE),]

ggplot(imp_DF, aes(x=reorder(Variables, MeanDecreaseAccuracy), y=MeanDecreaseAccuracy, fill=MeanDecreaseAccuracy))
```



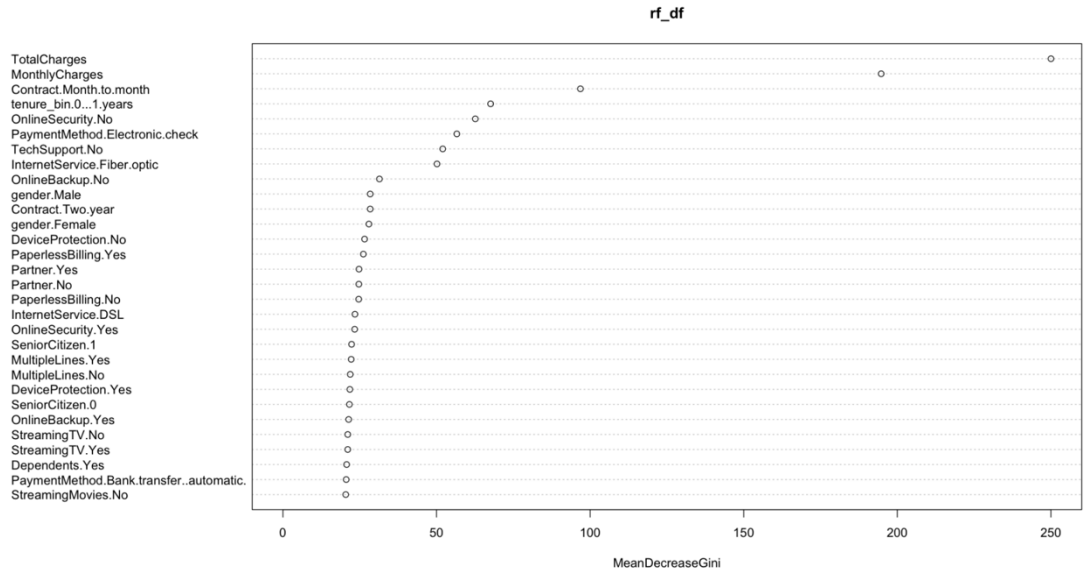
According to the Variable Importance plot, TotalCharges, MonthlyCharges, Tenure_year and Contract are the top 4 most important variables to predict churn. The PhoneSerivce, Gender, SeniorCitizen, Dependents, Partner, MultipleLines, PaperlessBilling, StreamingTV, Movies, DeviceProtection and OnlineBackup have comparatively less effect on Churn.

In [47]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
plot(rf_df , main = "Error by number of trees")
```



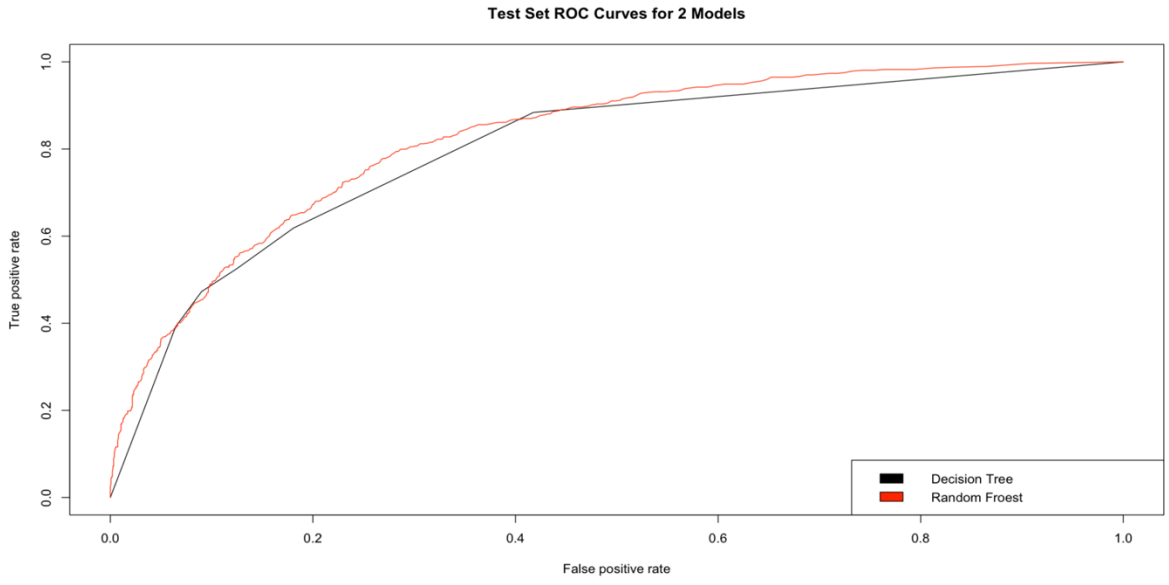
Variable Importance

In [48]:
```
options(repr.plot.width = 15, repr.plot.height = 8)
varImpPlot(rf_df,type=2)
```



Comparison of ROC and AUC for Decision Tree and Random Forest models

In [50]:
```
preds_list <- list(testtree_prob1[,2], testrf_prob1[,2])
m <- length(preds_list)
actuals_list <- rep(list(testdf$churn_number), m)

pred <- prediction(preds_list, actuals_list)
rocs <- performance(pred, "tpr", "fpr")
options(repr.plot.width = 15, repr.plot.height = 8)
plot(rocs, col = as.list(1:m), main = "Test Set ROC Curves for 2 Models")
legend(x = "bottomright",
       legend = c("Decision Tree", "Random Froest"),
       fill = 1:m)
```



The random forest model works better than the decision tree model. The accuracy is 0.7908 for Decision Tree and 0.7927 for Random Forest. The AUC score of decision tree is 0.79 and 0.828 for random forest.

Part5. Conclusion

- What makes the problem interesting from the viewpoint of analytics?

For me, I think the most interesting part is the difference between the decision tree and random forest. While random forest is a collection of decision trees, there are some differences. However, both of them can be used in different scenario and return the result we want!

If you input a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions. If you put the features and labels into a decision tree, it will generate some rules that help predict customer churn. In comparison, the random forest algorithm randomly selects observations and features to build several decision trees and then averages the results. Another difference is "deep" decision trees might suffer from overfitting. Most of the time, random forest prevents this by creating random subsets of the features and building smaller trees using those subsets. Afterwards, it combines the subtrees.

- How did the chosen technique help to illuminate the, or solve the problem?

Tree-based models are very popular in machine learning. The decision tree, the foundation of tree-based models, is quite straightforward to interpret, but generally a weak predictor. Ensemble models can be used to generate stronger predictions from many trees, with random forest or gradient boosting as two of the most popular. All tree-based models can be used for regression or classification and can handle non-linear relationships quite well. It's awesome that we can use these models to predict the outcome and make the decision based on the result.

- What analysis do you think should be conducted next?

For customer churn prediction, the next analysis should be customer analysis. The company should focus on the customers who show non churn and find out the reason. Therefore, company can find out a better approach to retain the customers. The company should also focus on customers who show satisfaction. Company can propose some rewards for them to make sure that customers keep high satisfaction.

- Overall Summary

From the above analysis, we can get the result below.

<u>Decision Tree</u>

```
Confusion Matrix and Statistics

          Reference
Prediction no yes no 1383 300 yes 137 269

               Accuracy : 0.7908
                 95% CI : (0.7727, 0.8081)
    No Information Rate : 0.7276
    P-Value [Acc > NIR] : 1.597e-11

                  Kappa : 0.4203

 Mcnemar's Test P-Value : 9.225e-15

            Sensitivity : 0.9099
            Specificity : 0.4728
         Pos Pred Value : 0.8217
         Neg Pred Value : 0.6626
             Prevalence : 0.7276
         Detection Rate : 0.6620

 Detection Prevalence : 0.8056
 Balanced Accuracy : 0.6913
```

Random Forest

```
    Confusion Matrix and Statistics

          Reference
```

Prediction no yes no 1375 288 yes 145 281

```
               Accuracy : 0.7927
                 95% CI : (0.7747, 0.8099)
    No Information Rate : 0.7276
    P-Value [Acc > NIR] : 3.803e-12

                  Kappa : 0.4325
```

Mcnemar's Test P-Value : 8.849e-12

```
              Sensitivity : 0.9046
              Specificity : 0.4938
           Pos Pred Value : 0.8268
           Neg Pred Value : 0.6596
               Prevalence : 0.7276
           Detection Rate : 0.6582
```

Detection Prevalence : 0.7961
Balanced Accuracy : 0.6992

The AUC score of Decision Tree is 0.8, and the AUC score of Random Forest is 0.828. We can conclude that the performance of random forest is better than decision tree.