FINAL PROJECT

Group-3

2022-12-10

#Business problem Most telecom companies suffer from voluntary churn. Churn rate has strong impact on the life time value of the customer because it affects the length of service and the future revenue of the company. For example if a company has 25% churn rate then the average customer lifetime is 4 years; similarly a company with a churn rate of 50%, has an average customer lifetime of 2 years. It is estimated that 75 percent of the 17 to 20 million subscribers signing up with a new wireless carrier every year are coming from another wireless provider, which means they are churners. Telecom companies spend hundreds of dollars to acquire a new customer and when that customer leaves, the company not only loses the future revenue from that customer but also the resources spend to acquire that customer. Churn erodes profitability.

#Approaches adapted by telecom companies to address churn. Untargeted and targeted aproach.

In this project, we will be working as a part of a team to use historical data from ACB Wireless Inc. to build a model that can predict/identify their customers who are likely to churn.

```
churn.train_data <- read.csv("/Users/ELMYLUKA/Desktop/MS BA/Business Analytics/Assignment-4/Churn_Train

#Analysing the data

str(churn.train_data)
```

Importing the dataset

```
## ’data.frame':    3333 obs. of  20 variables:
##  $ state                        : chr  "NV" "HI" "DC" "HI" ...
##  $ account_length               : int  125 108 82 NA 83 89 135 28 86 65 ...
##  $ area_code                    : chr  "area_code_510" "area_code_415" "area_code_415" "area_code_408
##  $ international_plan           : chr  "no" "no" "no" "no" ...
##  $ voice_mail_plan              : chr  "no" "no" "no" "yes" ...
##  $ number_vmail_messages        : int  0 0 0 30 0 0 0 0 0 0 ...
##  $ total_day_minutes            : num  2013 292 300 110 337 ...
##  $ total_day_calls              : int  99 99 109 71 120 81 81 87 115 137 ...
##  $ total_day_charge             : num  28.7 49.6 51 18.8 57.4 ...
##  $ total_eve_minutes            : num  1108 221 181 182 227 ...
##  $ total_eve_calls              : int  107 93 100 108 116 74 114 92 112 83 ...
##  $ total_eve_charge             : num  14.9 18.8 15.4 15.5 19.3 ...
##  $ total_night_minutes          : num  243 229 270 184 154 ...
##  $ total_night_calls            : int  92 110 73 88 114 120 82 112 95 111 ...
##  $ total_night_charge           : num  10.95 10.31 12.15 8.27 6.93 ...
##  $ total_intl_minutes           : num  10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...
##  $ total_intl_calls             : int  7 9 4 8 7 4 6 3 7 6 ...
##  $ total_intl_charge            : num  2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...
##  $ number_customer_service_calls: int  0 2 0 2 0 1 1 3 2 4 ...
##  $ churn                        : chr  "no" "yes" "yes" "no" ...
```

```
glimpse(churn.train_data)
```

```
## Rows: 3,333
## Columns: 20
## $ state                         <chr> "NV", "HI", "DC", "HI", "OH", "MO", "NC"~
## $ account_length                <int> 125, 108, 82, NA, 83, 89, 135, 28, 86, 6~
## $ area_code                     <chr> "area_code_510", "area_code_415", "area_~
## $ international_plan            <chr> "no", "no", "no", "no", "no", "no", "no"~
## $ voice_mail_plan               <chr> "no", "no", "no", "yes", "no", "no", "no~
## $ number_vmail_messages         <int> 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, NA, 32~
## $ total_day_minutes             <dbl> 2013.4, 291.6, 300.3, 110.3, 337.4, 178.~
## $ total_day_calls               <int> 99, 99, 109, 71, 120, 81, 81, 87, 115, 1~
## $ total_day_charge              <dbl> 28.66, 49.57, 51.05, 18.75, 57.36, 30.38~
## $ total_eve_minutes             <dbl> 1107.6, 221.1, 181.0, 182.4, 227.4, NA, ~
## $ total_eve_calls               <int> 107, 93, 100, 108, 116, 74, 114, 92, 112~
## $ total_eve_charge              <dbl> 14.93, 18.79, 15.39, 15.50, 19.33, 19.86~
## $ total_night_minutes           <dbl> 243.3, 229.2, 270.1, 183.8, 153.9, 131.9~
## $ total_night_calls             <int> 92, 110, 73, 88, 114, 120, 82, 112, 95, ~
## $ total_night_charge            <dbl> 10.95, 10.31, 12.15, 8.27, 6.93, 5.94, 9~
## $ total_intl_minutes            <dbl> 10.9, 14.0, 11.7, 11.0, 15.8, 9.1, 10.3,~
## $ total_intl_calls              <int> 7, 9, 4, 8, 7, 4, 6, 3, 7, 6, 7, NA, 4, ~
## $ total_intl_charge             <dbl> 2.94, 3.78, 3.16, 2.97, 4.27, 2.46, 2.78~
## $ number_customer_service_calls <int> 0, 2, 0, 2, 0, 1, 1, 3, 2, 4, 1, NA, 3, ~
## $ churn                         <chr> "no", "yes", "yes", "no", "yes", "no", "~
```

```
#Summary of the dataset
summary(churn.train_data)
```

```
##     state           account_length     area_code         international_plan
##  Length:3333        Min.   :-209.00   Length:3333        Length:3333
##  Class :character   1st Qu.:  72.00   Class :character   Class :character
##  Mode  :character   Median : 100.00   Mode  :character   Mode  :character
##                     Mean   :  97.32
##                     3rd Qu.: 127.00
##                     Max.   : 243.00
##                     NA’s   :501
##  voice_mail_plan    number_vmail_messages total_day_minutes total_day_calls
##  Length:3333        Min.   :-10.000       Min.   :   0.0    Min.   :  0.0
##  Class :character   1st Qu.:  0.000       1st Qu.: 149.3    1st Qu.: 87.0
##  Mode  :character   Median :  0.000       Median : 190.5    Median :101.0
##                     Mean   :  7.333       Mean   : 418.9    Mean   :100.3
##                     3rd Qu.: 16.000       3rd Qu.: 237.8    3rd Qu.:114.0
##                     Max.   : 51.000       Max.   :2185.1    Max.   :165.0
##                     NA’s   :200           NA’s   :200       NA’s   :200
##  total_day_charge total_eve_minutes total_eve_calls total_eve_charge
##  Min.   : 0.00    Min.   :   0.0    Min.   :  0.0   Min.   : 0.00
##  1st Qu.:24.45    1st Qu.: 170.5    1st Qu.: 87.0   1st Qu.:14.14
##  Median :30.65    Median : 209.9    Median :100.0   Median :17.09
##  Mean   :30.63    Mean   : 324.3    Mean   :100.1   Mean   :17.08
##  3rd Qu.:36.84    3rd Qu.: 257.6    3rd Qu.:114.0   3rd Qu.:20.00
##  Max.   :59.64    Max.   :1244.2    Max.   :170.0   Max.   :30.91
##  NA’s   :200      NA’s   :301       NA’s   :200     NA’s   :200
##  total_night_minutes total_night_calls total_night_charge total_intl_minutes
##  Min.   : 23.2       Min.   : 33.0     Min.   : 1.040     Min.   : 0.00
##  1st Qu.:167.3       1st Qu.: 87.0     1st Qu.: 7.530     1st Qu.: 8.50
##  Median :201.4       Median :100.0     Median : 9.060     Median :10.30
##  Mean   :201.2       Mean   :100.1     Mean   : 9.054     Mean   :10.23
##  3rd Qu.:235.3       3rd Qu.:113.0     3rd Qu.:10.590     3rd Qu.:12.10
##  Max.   :395.0       Max.   :175.0     Max.   :17.770     Max.   :20.00
##  NA’s   :200                           NA’s   :200        NA’s   :200
##  total_intl_calls total_intl_charge number_customer_service_calls
##  Min.   : 0.00    Min.   :0.000     Min.   :0.000
##  1st Qu.: 3.00    1st Qu.:2.300     1st Qu.:1.000
##  Median : 4.00    Median :2.780     Median :1.000
##  Mean   : 4.47    Mean   :2.762     Mean   :1.561
##  3rd Qu.: 6.00    3rd Qu.:3.270     3rd Qu.:2.000
##  Max.   :20.00    Max.   :5.400     Max.   :9.000
##  NA’s   :301      NA’s   :200       NA’s   :200
##     churn
##  Length:3333
##  Class :character
##  Mode  :character
##
##
##
##
```

##transforming categorical variables to numeric.

```r
churn.train_data$state <- as.factor(churn.train_data$state)
churn.train_data$area_code <- as.factor(churn.train_data$area_code)
churn.train_data$international_plan <- as.factor(churn.train_data$international_plan)
churn.train_data$voice_mail_plan <- as.factor(churn.train_data$voice_mail_plan)
churn.train_data$churn <- as.factor(churn.train_data$churn)
churn_true  <- subset(churn.train_data, churn.train_data$churn == "yes")
churn_false <- subset(churn.train_data, churn.train_data$churn == "no")
```

#churn count number of yes/no

```r
churn_count_number<-table(churn.train_data$churn)
churn_count_number
```

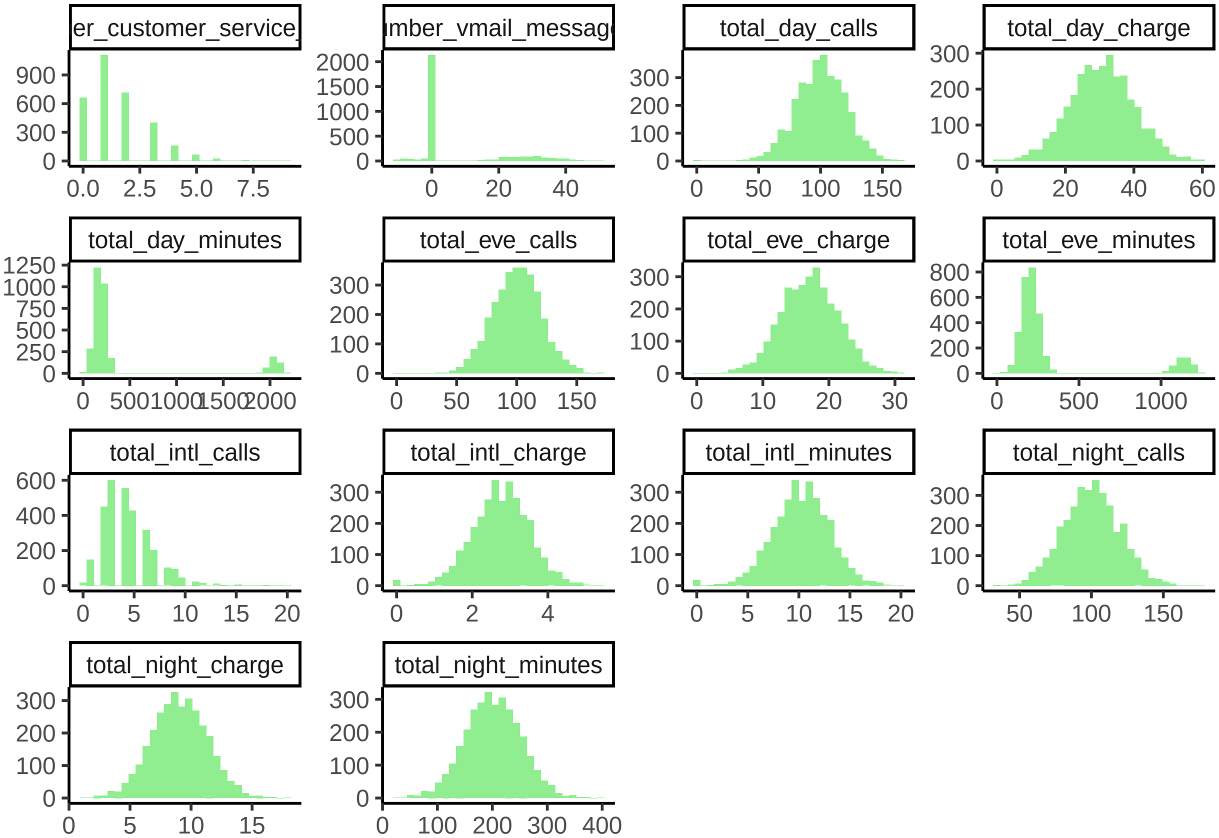
```
##
##   no  yes
## 2850  483
```

#examining the skewness and distribution of each variable in the dataset.

```r
churn.train_data[, 6:19] %>%
  gather(key = Variable, value = Value) %>%
  ggplot() +
  geom_histogram(aes(x = Value), fill = "light green") +
  facet_wrap(~Variable, scales='free') +
  theme_classic() +
  theme(aspect.ratio = 0.5, axis.title = element_blank(), panel.grid = element_blank())
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2802 rows containing non-finite values (stat_bin).
```



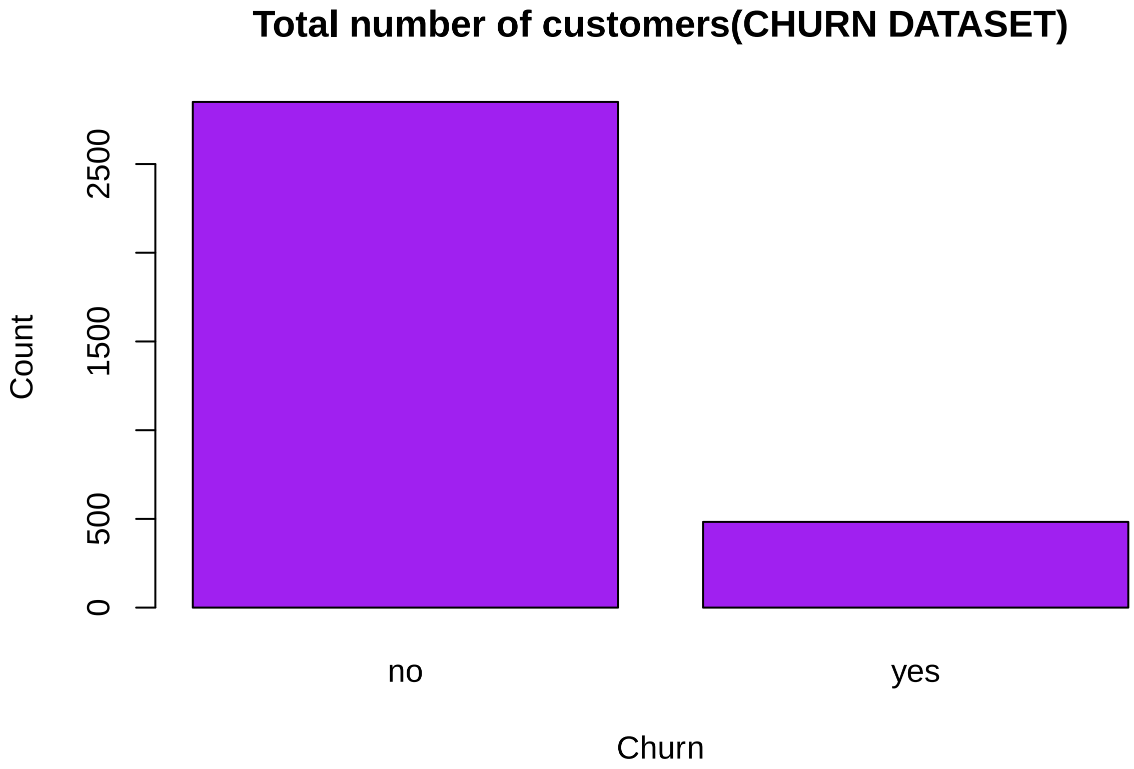
We can determine from the output that there is a bell curve distribution of data or variables for the majority of the data.It is also an observation that “total day minutes” and total evening minutes” have a tiny percentage or sizeable quantity of outliers. An other observation determined is that “Customer_Service_calls” has an irregular skewness.

#Determining the number of customers from the dataset.

```
churn_count_number
```

```
##
##   no  yes
## 2850  483
```

```
barplot(churn_count_number,xlab ="Churn",ylab="Count" ,col = "purple" ,main = "Total number of customer
```



It can be determined from the above graph that among the customers, 483 customers have switched to other providers while the remaining 2850 of them have decided to stay.

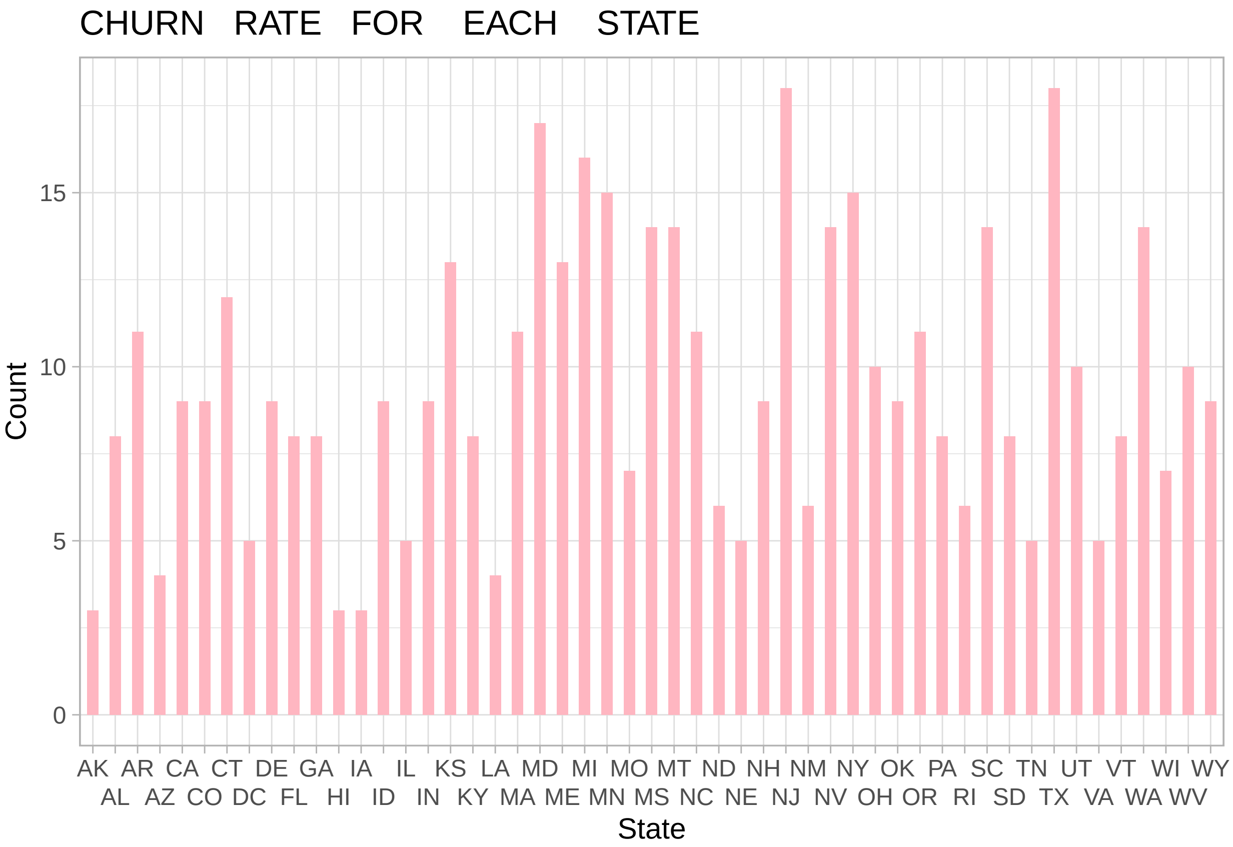
#Determing the number of customers as per the States

```
count_state<-churn_true %>% group_by(state) %>% summarise(count_churn_state=n())
churn_state <- churn.train_data %>%group_by(churn.train_data$state, churn.train_data$churn) %>% summaris
```

```
## ‘summarise()‘ has grouped output by ’churn.train_data$state’. You can override
## using the ‘.groups‘ argument.
```

“summarise()” function has grouped output by ’Churn_Data$state’. Therefore we can override using the ’.groups‘ argument.

```
ggplot(count_state) +
  aes(x = state, weight = count_churn_state) +
  geom_bar(width=0.5, position = position_dodge(width=0.5), fill = "#FFB6C1") + scale_x_discrete(guide
 labs(x = "State", y = "Count", title = "CHURN   RATE   FOR    EACH    STATE")+theme_light()
```

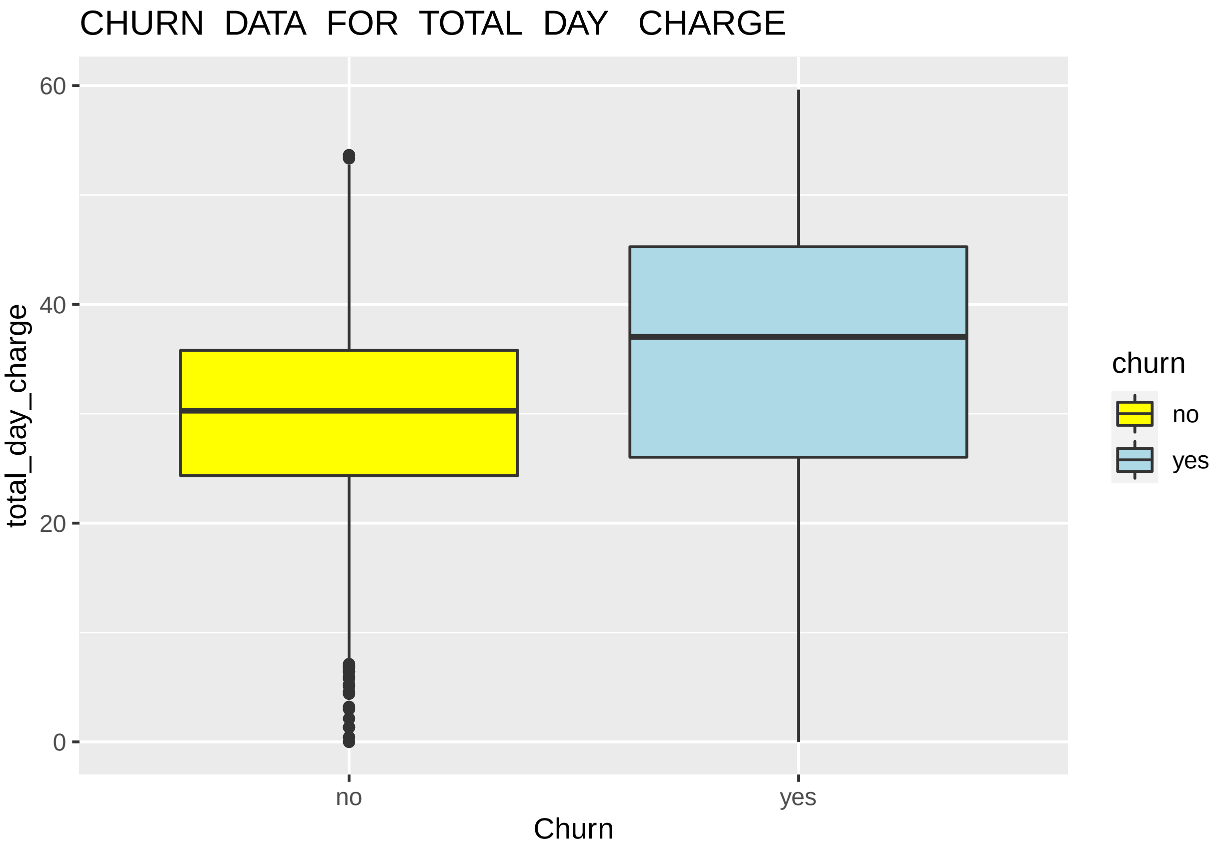


It is determined from the graph that Maryland, New Jersey, Michigan and Texas are the states with high churn rates.

#Distributing the dataset by the Total day charges.

```
ggplot(churn.train_data) +
  aes(x = churn, y = total_day_charge, fill = churn) +
  geom_boxplot(shape = "square") +
  scale_fill_manual(breaks = churn.train_data$churn,
                    values = c("yellow", "light blue"))+ labs(x = "Churn", y = "total_day_charge",title
```

```
## Warning: Removed 200 rows containing non-finite values (stat_boxplot).
```



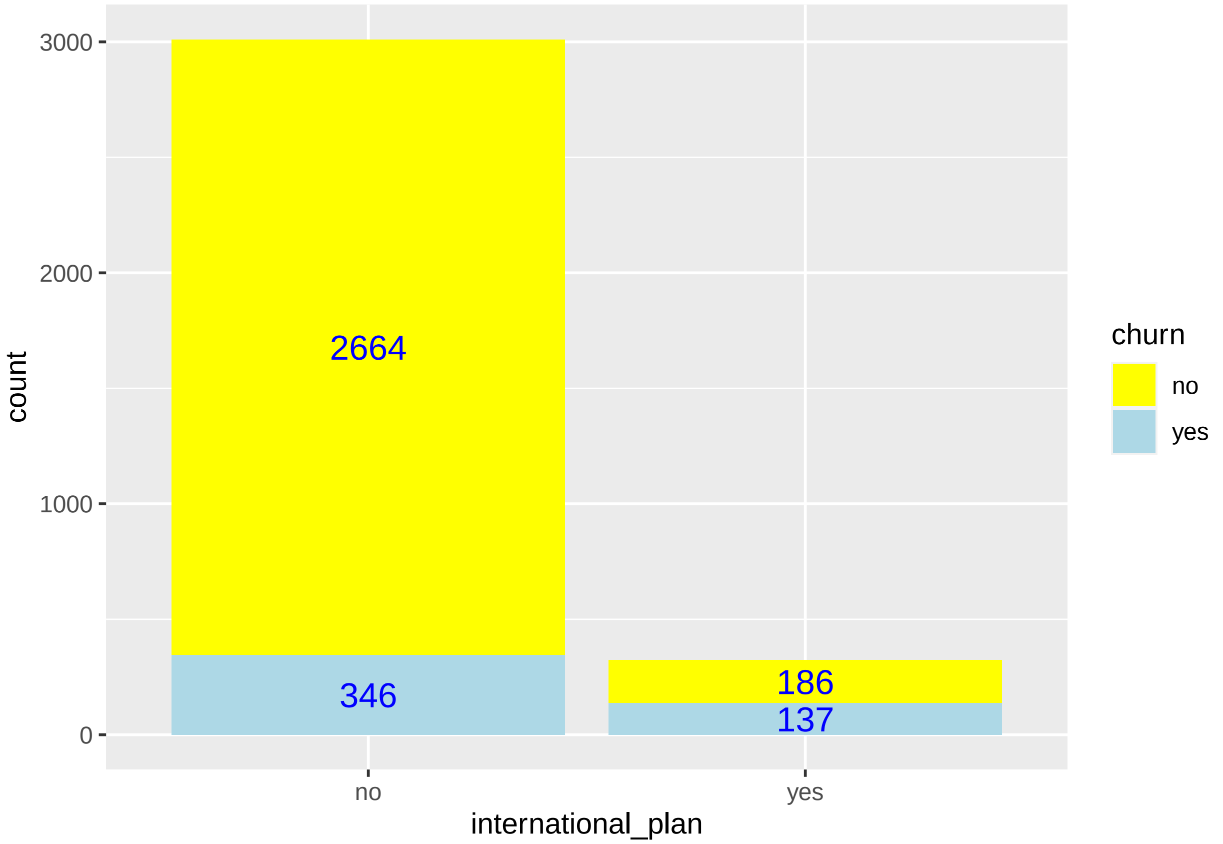
```
theme_minimal()+
theme(plot.title = element_text(size = 16L,
                                face = "bold", hjust = 0.5))
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : ’margin’ num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : ’simpleUnit’ num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : ’simpleUnit’ num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : ’margin’ num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : ’simpleUnit’ num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : chr "bold"
##   ..$ colour       : NULL
##   ..$ size         : int 16
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : ’rel’ num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : ’margin’ num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : ’rel’ num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : ’margin’ num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

It is observed from the box plot graph that customers having the day charge between 30-40 are more inclined towards cancelling their services with the current providers and shift to a different provider.

#Determing the customers who had the international package and shifted to another provider based on the dataset.

```
ggplot(data = churn.train_data, aes(x = international_plan, y = ..count.., fill = churn)) +
  scale_fill_manual(breaks = churn.train_data$churn,
                    values = c("yellow", "light blue"))+
  geom_bar(stat = "count") +
stat_count(geom = "text", colour = "blue", size = 4.5,
aes(label = ..count..),position=position_stack(vjust=0.5))
```



```
churn_true %>%
  group_by(international_plan) %>%
  select(international_plan) %>%
  dplyr:: summarise("Churn Count" =n(), "Percent" = n()/483)
```

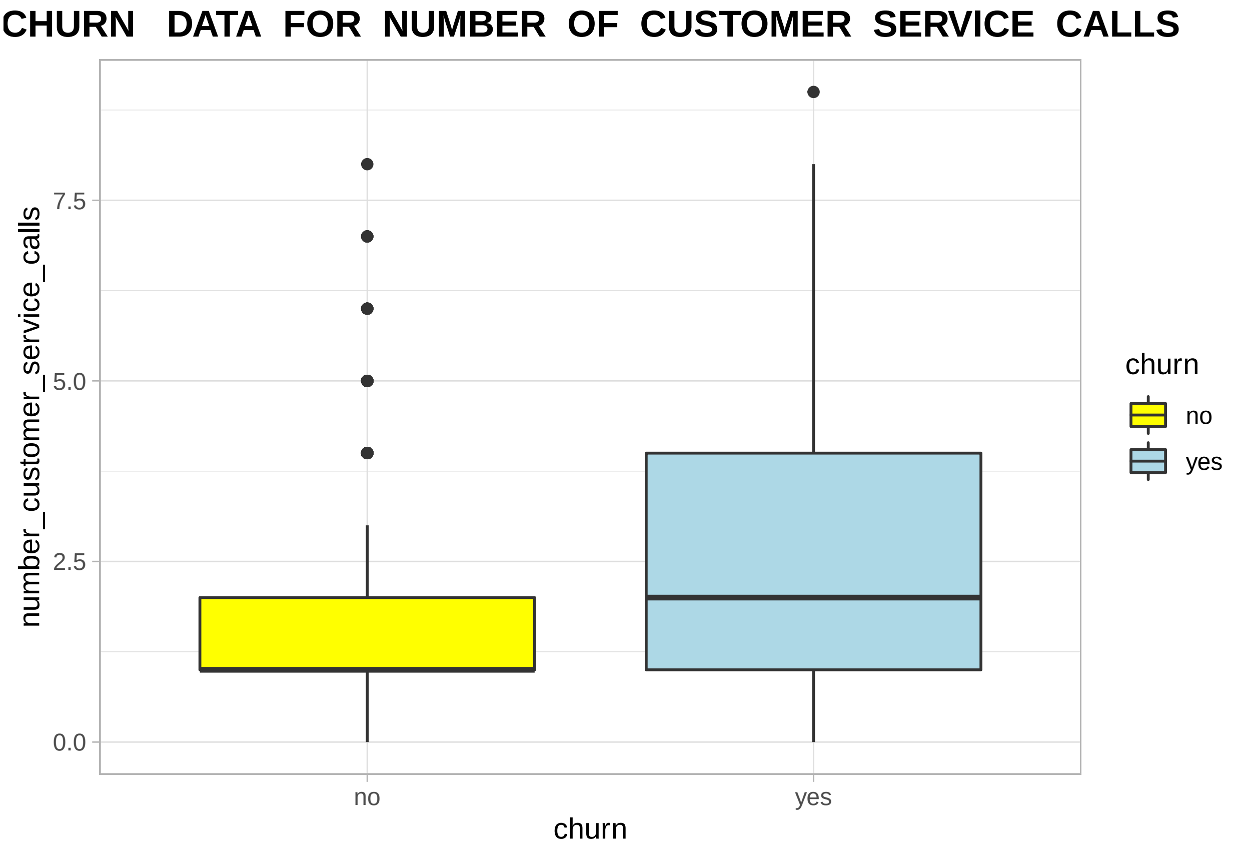
```
## # A tibble: 2 x 3
##   international_plan ‘Churn Count‘ Percent
##   <fct>                      <int>   <dbl>
## 1 no                           346   0.716
## 2 yes                          137   0.284
```

The results depict the percentage of customers who are a part of the international plan and have moved to another provider i.e. 28% of the customers are likely to churn.

#Determining the customers who churned based on the number of customer service calls.

```
ggplot(churn.train_data) +
  aes(x = churn, y = number_customer_service_calls, fill = churn) +
  geom_boxplot(shape = "circle") +
    scale_fill_manual(breaks = churn.train_data$churn,
                    values = c("yellow", "light blue"))+
  labs(title = "CHURN   DATA  FOR  NUMBER  OF  CUSTOMER  SERVICE  CALLS") +
  theme_light() +
  theme(plot.title = element_text(size = 14L, face = "bold", hjust = 0.5))
```

```
## Warning: Removed 200 rows containing non-finite values (stat_boxplot).
```



```
churn_true %>%filter(number_customer_service_calls >= 1 & number_customer_service_calls <= 4) %>%tally
```

```
##           n
## 1 0.6397516
```

The box plot above depicts that the customers who have reached out to the customer services more than 2-4 times are likely to move to other providers. We can interpret that the customers who have churned are approximately 64% and the reason being, them reaching out to the customer service 1-4 times.

#Data Cleaning

```
#Sorting and imputing the missing values using mice package.
set.seed(111)
#As per mice, total_night_charge and total_intl_charge are multi-collinear variables.
#Therefore mice will not impute missing values for these columns.
churn.train_data$total_night_charge[1] <- 2
churn.train_data$total_intl_charge[1] <- 0.5
mice_model <- mice(churn.train_data[, -20], method="rf")
```

```
##
##  iter imp variable
##   1   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   1   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   1   3  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   1   4  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   1   5  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   3  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   4  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   5  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   3  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   4  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   5  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   3  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   4  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   5  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   3  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   4  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   5  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
```

```
## Warning: Number of logged events: 350
```

```r
#mice imputation using random forests.
mice_output <- complete(mice_model)
# Generating the complete data.
anyNA(mice_output)
```

```
## [1] FALSE
```

```r
churn.train_data_imputed <- mutate(mice_output,churn=churn.train_data$churn)
summary(churn.train_data)
```

```
##      state      account_length            area_code    international_plan
##  WV     : 106   Min.   :-209.00   area_code_408: 838   no :3010
##  MN     :  84   1st Qu.:  72.00   area_code_415:1655   yes: 323
##  NY     :  83   Median : 100.00   area_code_510: 840
##  AL     :  80   Mean   :  97.32
##  OH     :  78   3rd Qu.: 127.00
##  OR     :  78   Max.   : 243.00
##  (Other):2824   NA's   :501
##  voice_mail_plan number_vmail_messages total_day_minutes total_day_calls
##  no :2411        Min.   :-10.000       Min.   :   0.0    Min.   :  0.0
##  yes: 922        1st Qu.:  0.000       1st Qu.: 149.3    1st Qu.: 87.0
##                  Median :  0.000       Median : 190.5    Median :101.0
##                  Mean   :  7.333       Mean   : 418.9    Mean   :100.3
##                  3rd Qu.: 16.000       3rd Qu.: 237.8    3rd Qu.:114.0
##                  Max.   : 51.000       Max.   :2185.1    Max.   :165.0
##                  NA's   :200           NA's   :200       NA's   :200
##  total_day_charge total_eve_minutes total_eve_calls total_eve_charge
##  Min.   : 0.00    Min.   :   0.0    Min.   :  0.0   Min.   : 0.00
##  1st Qu.:24.45    1st Qu.: 170.5    1st Qu.: 87.0   1st Qu.:14.14
##  Median :30.65    Median : 209.9    Median :100.0   Median :17.09
##  Mean   :30.63    Mean   : 324.3    Mean   :100.1   Mean   :17.08
##  3rd Qu.:36.84    3rd Qu.: 257.6    3rd Qu.:114.0   3rd Qu.:20.00
##  Max.   :59.64    Max.   :1244.2    Max.   :170.0   Max.   :30.91
##  NA’s   :200      NA’s   :301       NA’s   :200     NA’s   :200
##  total_night_minutes total_night_calls total_night_charge total_intl_minutes
##  Min.   : 23.2       Min.   : 33.0     Min.   : 1.040     Min.   : 0.00
##  1st Qu.:167.3       1st Qu.: 87.0     1st Qu.: 7.530     1st Qu.: 8.50
##  Median :201.4       Median :100.0     Median : 9.060     Median :10.30
##  Mean   :201.2       Mean   :100.1     Mean   : 9.051     Mean   :10.23
##  3rd Qu.:235.3       3rd Qu.:113.0     3rd Qu.:10.590     3rd Qu.:12.10
##  Max.   :395.0       Max.   :175.0     Max.   :17.770     Max.   :20.00
##  NA’s   :200                           NA’s   :200        NA’s   :200
##  total_intl_calls total_intl_charge number_customer_service_calls churn
##  Min.   : 0.00    Min.   :0.000     Min.   :0.000                 no :2850
##  1st Qu.: 3.00    1st Qu.:2.300     1st Qu.:1.000                 yes: 483
##  Median : 4.00    Median :2.780     Median :1.000
##  Mean   : 4.47    Mean   :2.761     Mean   :1.561
##  3rd Qu.: 6.00    3rd Qu.:3.270     3rd Qu.:2.000
##  Max.   :20.00    Max.   :5.400     Max.   :9.000
##  NA’s   :301      NA’s   :200       NA’s   :200
```

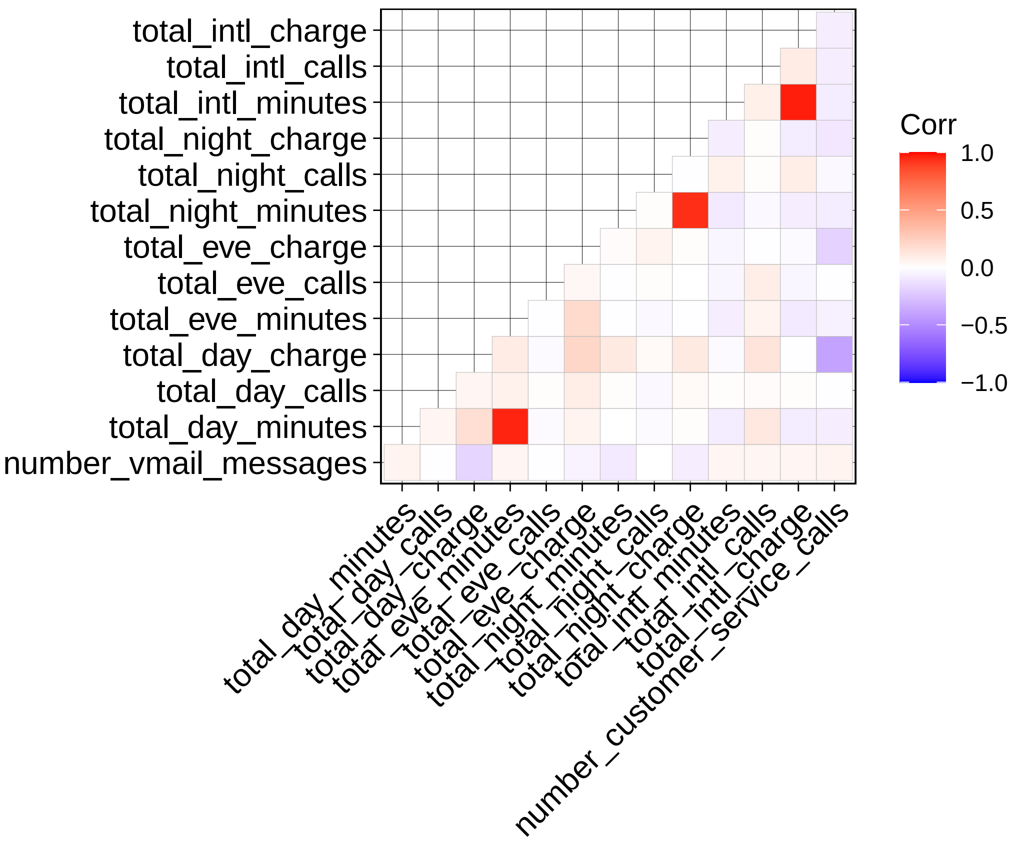
```
str(churn.train_data)
```

```
## ’data.frame’:    3333 obs. of  20 variables:
##  $ state                        : Factor w/ 51 levels "AK","AL","AR",..: 34 12 8 12 36 25 28 39 13 16
##  $ account_length               : int  125 108 82 NA 83 89 135 28 86 65 ...
##  $ area_code                    : Factor w/ 3 levels "area_code_408",..: 3 2 2 1 2 2 2 2 1 2 ...
##  $ international_plan           : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ voice_mail_plan              : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ number_vmail_messages        : int  0 0 0 30 0 0 0 0 0 0 ...
##  $ total_day_minutes            : num  2013 292 300 110 337 ...
##  $ total_day_calls              : int  99 99 109 71 120 81 81 87 115 137 ...
##  $ total_day_charge             : num  28.7 49.6 51 18.8 57.4 ...
##  $ total_eve_minutes            : num  1108 221 181 182 227 ...
##  $ total_eve_calls              : int  107 93 100 108 116 74 114 92 112 83 ...
##  $ total_eve_charge             : num  14.9 18.8 15.4 15.5 19.3 ...
##  $ total_night_minutes          : num  243 229 270 184 154 ...
##  $ total_night_calls            : int  92 110 73 88 114 120 82 112 95 111 ...
##  $ total_night_charge           : num  2 10.31 12.15 8.27 6.93 ...
##  $ total_intl_minutes           : num  10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...
##  $ total_intl_calls             : int  7 9 4 8 7 4 6 3 7 6 ...
##  $ total_intl_charge            : num  0.5 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...
##  $ number_customer_service_calls: int  0 2 0 2 0 1 1 3 2 4 ...
##  $ churn                        : Factor w/ 2 levels "no","yes": 1 2 2 1 2 1 1 1 1 2 ...
```

```
churn_yes<-churn.train_data_imputed %>% filter(churn=='yes')
correlation_churn_cust<- cor(churn_yes[, 6:19])
```

#We will be using ggplot to represent the correlation between the variables where churn is equal to yes.

```
ggcorrplot(correlation_churn_cust, method = "square", type = "lower", ggtheme = theme_linedraw)
```



As per the ggplot, it can be depicted that for the people who have churned, there lies a significant negative correlation between total__day__charge and the number of customer__ service__calls and also total__international__charges and total__evening__charges. The statistics show that customer service calls have a greater churn rate than other calls since the charges are higher.

#Prediction Model Selection Using a predictive model based on regression and decision tree models. It is possible to demonstrate the influence of various variables and the importance of each in foreseeing the outcome of the dependent variable.

A logistic regression model is preferred to others since the dependent variable (target variable) in this data is categorical and also classification being our prime objective . While in a linear regression model, performance probability may be negative or more than 1, making it ineffective for predicting a binomial feature. The best result for this model is a likelihood of possibilities that falls between 0 and 1 i.e. logistic regression.

For our analysis we will be using both the models and select the best among the two to be the final model. Using Logistic Regression and Decision Tree Models to determine Predictive Ability: Before choosing a model, the following procedures were followed: - The dataset has been divided into training and validation sets to prevent overfitting the model. -Constructing a logistic regression model and forecasting the outcomes from the validation set. -Using a confusion matrix to confirm the validity of the model. -Making a decision tree model and predict the results of the validation set. -Validating the model's performance with a confusion matrix. -Considering the results of both models and selecting the best one.

#Data Partitioning

```
set.seed(111)
index<- createDataPartition(churn.train_data_imputed $churn,p=0.8,list=FALSE)
```

```
train_data<-churn.train_data_imputed [index,]
valid_data <- churn.train_data_imputed [-index,]
```

#Building a Logistic Regression model:- Logistic regression is a statistical analytic approach for predicting a binary outcome, such as yes or no.

```
set.seed(222)
log_model <- glm(churn~.,data=train_data ,family = "binomial" ) #summary(Logistic Model)
predict_valid<-predict(log_model,valid_data,type="response")
head(predict_valid)
```

```
##          10          21          28          34          39          40
## 0.16572725 0.07577522 0.05251576 0.02260023 0.22631119 0.02431119
```

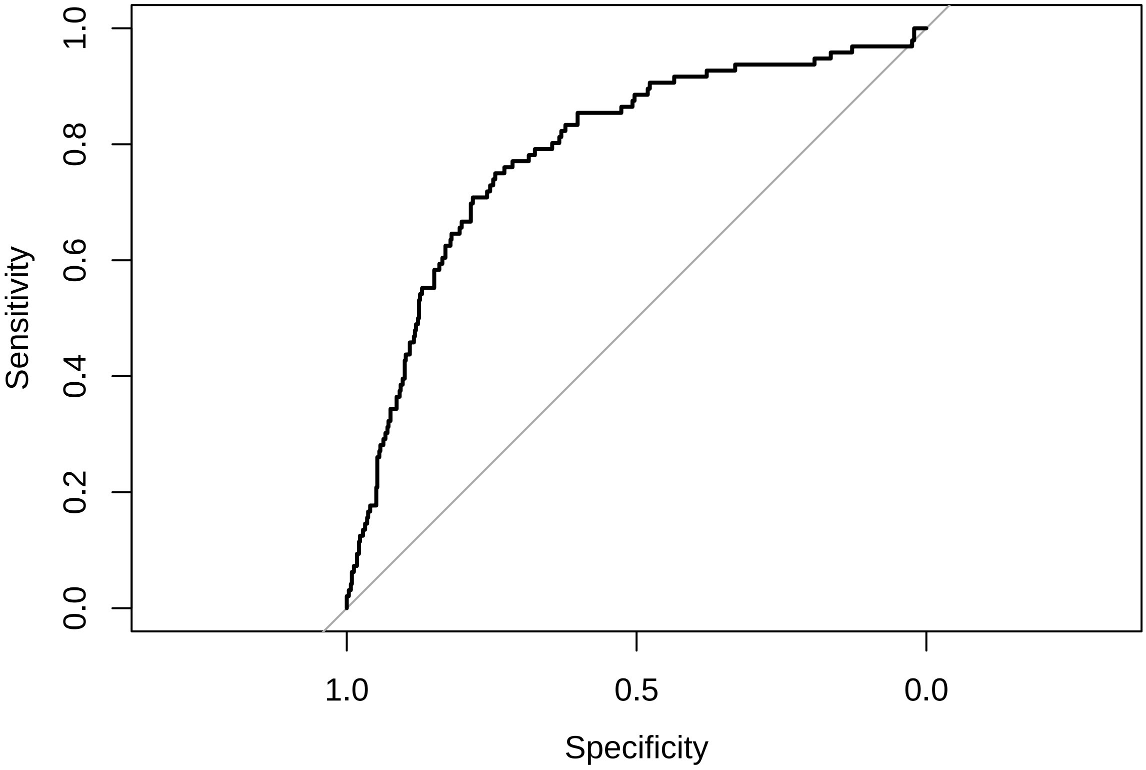
```
result_check<-ifelse(predict_valid > 0.5,'yes','no')
#Accuracy Check
error<-mean(result_check!=valid_data$churn)
accuracy <-1- error
print(accuracy)
```

```
## [1] 0.8468468
```

```
plot.roc(valid_data$churn,predict_valid)
```

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

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



#Using confusion matrix for the logistic regression model.

```
set.seed(333)
log_confusion_matrix <- confusionMatrix(as.factor(result_check),as.factor(valid_data$churn))
log_confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  549  81
##        yes  21  15
##
##                Accuracy : 0.8468
##                  95% CI : (0.8172, 0.8734)
##     No Information Rate : 0.8559
##     P-Value [Acc > NIR] : 0.7653
##
##                   Kappa : 0.1613
##
##  Mcnemar’s Test P-Value : 5.162e-09
##
##             Sensitivity : 0.9632
##             Specificity : 0.1562
##          Pos Pred Value : 0.8714
##          Neg Pred Value : 0.4167
##              Prevalence : 0.8559
##          Detection Rate : 0.8243
##    Detection Prevalence : 0.9459
##       Balanced Accuracy : 0.5597
##
##        ’Positive’ Class : no
##
```

Results produced from the confusion matrix :- #1.Accuracy :- 84.68% #2. Sensitivity :- 96.32% #3. Specificity:- 15.62%

#Building a Decision Tree Model Decision tree analysis is basically producing a tree-shaped diagram to chart out a course of action or a statistical probability analysis.

```
set.seed(444)
decisiontree_model<- rpart(churn ~ .,data=train_data,method = 'class')
# Show the variable importance
#DT_model$variable.importance
# Show the split for variable
head(decisiontree_model$splits)
```

```
##                               count ncat  improve   index adj
## total_day_charge               2667   -1 78.51181  44.975   0
## number_customer_service_calls  2667   -1 57.34523   3.500   0
## international_plan             2667    2 37.82693   1.000   0
## total_day_minutes              2667   -1 22.37794 263.600   0
## state                          2667   51 15.22592   2.000   0
## number_customer_service_calls  2485   -1 59.34436   3.500   0
```

```
#Predicting the probability
prob_decisiontree <- predict(decisiontree_model, newdata = valid_data, type = "prob")
#determining AUC Value
roc(valid_data$churn,prob_decisiontree[,2])
```

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

## Setting direction: controls < cases

##
## Call:
## roc.default(response = valid_data$churn, predictor = prob_decisiontree[,     2])
##
## Data: prob_decisiontree[, 2] in 570 controls (valid_data$churn no) < 96 cases (valid_data$churn yes)
## Area under the curve: 0.8234
```

Using a Confusion Matrix for the Decision Tree Model.

```
set.seed(555)
decisiontree_class<- predict(decisiontree_model, newdata = valid_data, type = "class")
confusionMatrix(as.factor(decisiontree_class),as.factor(valid_data$churn))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  555  42
##        yes  15  54
##
##                Accuracy : 0.9144
##                  95% CI : (0.8905, 0.9345)
##     No Information Rate : 0.8559
##     P-Value [Acc > NIR] : 3.035e-06
##
##                   Kappa : 0.6072
##
##  Mcnemar's Test P-Value : 0.0005736
##
##             Sensitivity : 0.9737
##             Specificity : 0.5625
##          Pos Pred Value : 0.9296
##          Neg Pred Value : 0.7826
##              Prevalence : 0.8559
##          Detection Rate : 0.8333
##    Detection Prevalence : 0.8964
##       Balanced Accuracy : 0.7681
##
##        'Positive' Class : no
##
```

From the Confusion Matrix, the following conclusions have been made :- #1. Accuracy :- 91.44% #2. Sensitivity :- 97.37% #3. Specificity:- 56.25%

#Choosing the optimal model

On the comparison of the two models, Decision Tree Model is interpreted the best model to put in use as it has higher accuracy than the logistical regression model.

Though the Sensitivities of both the models are almost equal, Decision Tree has a higher specificity. Therefore, Decision Tree Model is the right and optimal model to use.

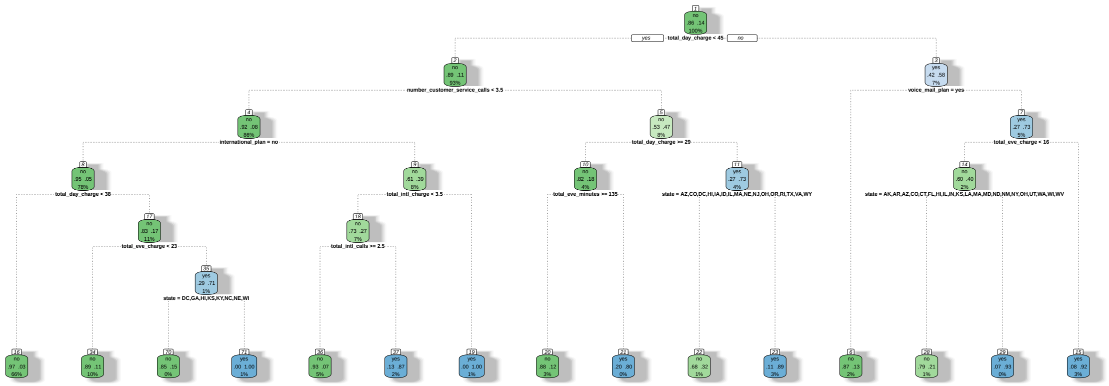
#Predicting the churn using the test data and the decision tree algorithm for the final model analysis.

```
# After the accuracy has been tested for the validation and training data we can use the entire data to
set.seed(666)
ABC_model<- rpart(churn ~ .,data= churn.train_data_imputed,method = 'class')
```

```
#Model Splits.
head(ABC_model$splits)
```

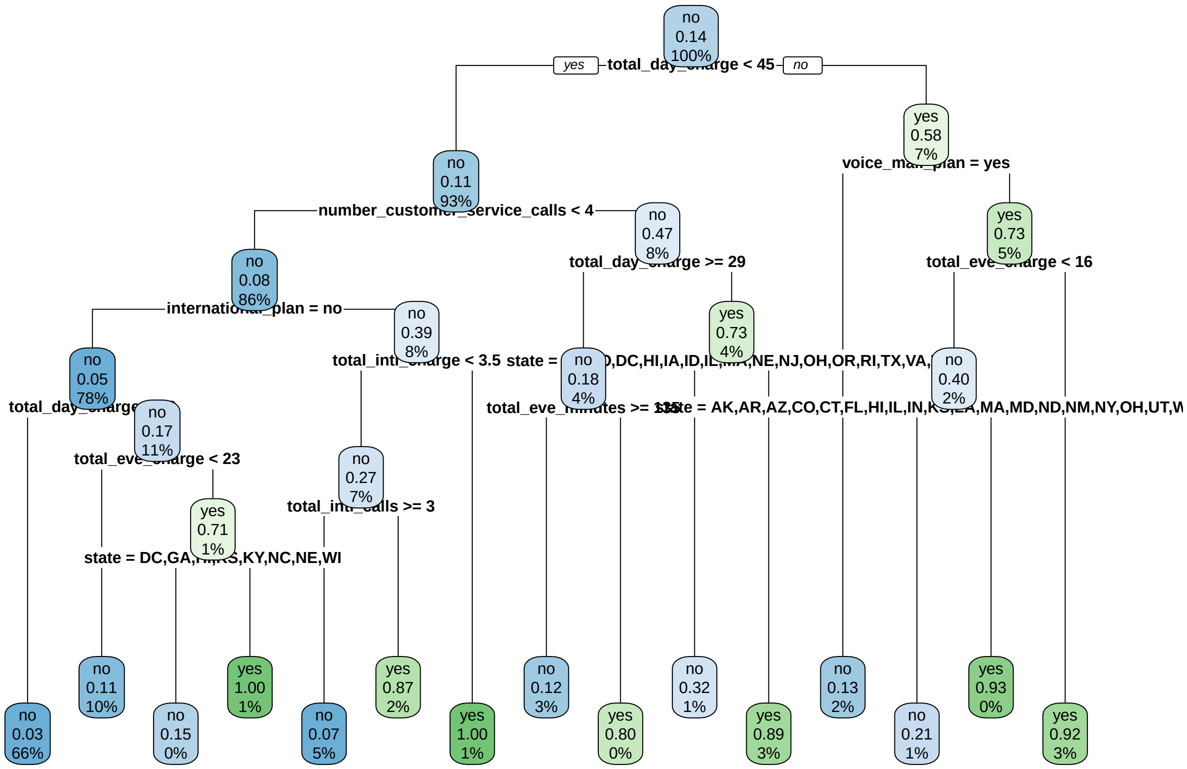
```
##                              count ncat  improve   index adj
## total_day_charge              3333   -1 88.13813  44.975   0
## number_customer_service_calls 3333   -1 68.18448   3.500   0
## international_plan            3333    2 55.77483   1.000   0
## total_day_minutes             3333   -1 26.21947 223.250   0
## state                         3333   51 14.95004   2.000   0
## number_customer_service_calls 3116   -1 71.09667   3.500   0
```

```
#Plotting Decision Tree
fancyRpartPlot(ABC_model)
```



Rattle 2022−Dec−10 11:03:23 ELMYLUKA

```
rpart.plot(ABC_model, cex=0.5)
```



```
#Probability Prediction(decision tree)
decisiontree_prob <- predict(ABC_model, newdata = churn.train_data_imputed, type = "prob")
#Determining the AUC Value
roc(churn.train_data_imputed$churn,decisiontree_prob[,2])
```

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

## Setting direction: controls < cases

##
## Call:
## roc.default(response = churn.train_data_imputed$churn, predictor = decisiontree_prob[,     2])
##
## Data: decisiontree_prob[, 2] in 2850 controls (churn.train_data_imputed$churn no) < 483 cases (churn
## Area under the curve: 0.8879
```

#Prediction of the Test Data

```
set.seed(777)
load("~/Desktop/MS BA/Business Analytics/Assignment-4/Customers_To_Predict.RData")

count(Customers_To_Predict)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1600
```

```r
summary(Customers_To_Predict)
```

```
##     state           account_length    area_code         international_plan
##  Length:1600        Min.   :  1.00   Length:1600        Length:1600
##  Class  :character  1st Qu.: 71.00   Class  :character  Class  :character
##  Mode   :character  Median : 98.00   Mode   :character  Mode   :character
##                     Mean   : 98.52
##                     3rd Qu.:126.00
##                     Max.   :238.00
##  voice_mail_plan    number_vmail_messages total_day_minutes total_day_calls
##  Length:1600        Min.   : 0.000        Min.   :  6.6     Min.   : 34.00
##  Class  :character  1st Qu.: 0.000        1st Qu.:143.8     1st Qu.: 86.00
##  Mode   :character  Median : 0.000        Median :180.9     Median : 99.00
##                     Mean   : 7.043        Mean   :181.6     Mean   : 99.06
##                     3rd Qu.: 0.000        3rd Qu.:215.9     3rd Qu.:112.00
##                     Max.   :52.000        Max.   :351.5     Max.   :160.00
##  total_day_charge total_eve_minutes total_eve_calls total_eve_charge
##  Min.   : 1.12    Min.   : 22.3     Min.   : 38.0   Min.   : 1.90
##  1st Qu.:24.45    1st Qu.:165.8     1st Qu.: 88.0   1st Qu.:14.10
##  Median :30.76    Median :199.9     Median :101.0   Median :17.00
##  Mean   :30.87    Mean   :199.6     Mean   :100.6   Mean   :16.96
##  3rd Qu.:36.70    3rd Qu.:231.8     3rd Qu.:114.0   3rd Qu.:19.70
##  Max.   :59.76    Max.   :359.3     Max.   :169.0   Max.   :30.54
##  total_night_minutes total_night_calls total_night_charge total_intl_minutes
##  Min.   :  0.0       Min.   :  0.00    Min.   : 0.000     Min.   : 0.00
##  1st Qu.:166.6       1st Qu.: 86.00    1st Qu.: 7.500     1st Qu.: 8.60
##  Median :199.2       Median : 99.00    Median : 8.960     Median :10.40
##  Mean   :199.2       Mean   : 99.45    Mean   : 8.963     Mean   :10.32
##  3rd Qu.:232.4       3rd Qu.:113.00    3rd Qu.:10.463     3rd Qu.:12.00
##  Max.   :381.6       Max.   :170.00    Max.   :17.170     Max.   :19.70
##  total_intl_calls total_intl_charge number_customer_service_calls
##  Min.   : 0.000   Min.   :0.000     Min.   :0.000
##  1st Qu.: 3.000   1st Qu.:2.320     1st Qu.:1.000
##  Median : 4.000   Median :2.810     Median :1.000
##  Mean   : 4.356   Mean   :2.786     Mean   :1.583
##  3rd Qu.: 5.000   3rd Qu.:3.240     3rd Qu.:2.000
##  Max.   :19.000   Max.   :5.320     Max.   :7.000
```

```r
#Checking NA Values
colMeans(is.na(Customers_To_Predict))
```

```
##                          state                 account_length
##                              0                              0
##                      area_code             international_plan
##                              0                              0
##                voice_mail_plan          number_vmail_messages
##                              0                              0
##              total_day_minutes                total_day_calls
##                              0                              0
##               total_day_charge              total_eve_minutes
##                              0                              0
##                total_eve_calls               total_eve_charge
##                              0                              0
##           total_night_minutes             total_night_calls
##                             0                             0
##            total_night_charge            total_intl_minutes
##                             0                             0
##              total_intl_calls             total_intl_charge
##                             0                             0
## number_customer_service_calls
##                             0
```

```
prob_churn <- predict(ABC_model,Customers_To_Predict,type = "prob")
head(prob_churn)
```

```
##          no        yes
## 1 0.9683973 0.03160271
## 2 0.9683973 0.03160271
## 3 0.9683973 0.03160271
## 4 0.9289941 0.07100592
## 5 0.9683973 0.03160271
## 6 0.6756757 0.32432432
```

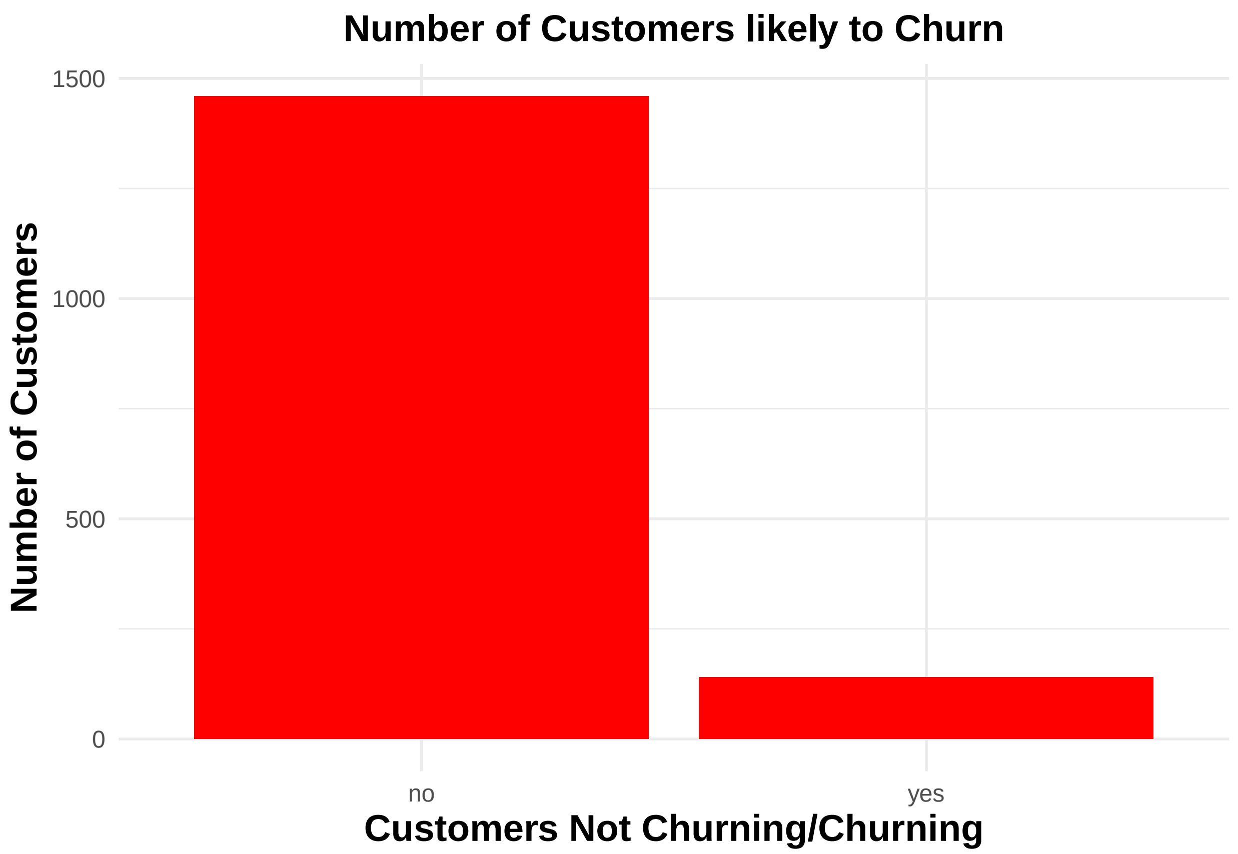
```
predict_churn <- predict(ABC_model,Customers_To_Predict,type = "class")
head(predict_churn)
```

```
##  1  2  3  4  5  6
## no no no no no no
## Levels: no yes
```

```
 predict_churn<- as.data.frame(predict_churn)
summary(predict_churn)
```

```
##  predict_churn
##  no :1460
##  yes: 140
```

```
ggplot(predict_churn) +
 aes(x = predict_churn) +
 geom_bar(fill = "red")+
 labs(x = "Customers Not Churning/Churning",
 y = "Number of Customers", title = "Number of Customers likely to Churn") +
 theme_minimal() +
 theme(plot.title = element_text(size = 14L,
 face = "bold", hjust = 0.5), axis.title.y = element_text(size = 14L, face = "bold"), axis.title.x = el
```



#From the above graph the following has been depicted.

Predict_Churn :- No :- 1460 Yes :- 140

#From the analysis of the data,the following are the conclusions:- #- Customers are more inclined to switch to another provider if they have paid more than $30 in daily fees. #- Customers will undoubtedly go to another supplier if they have to pay international day charges. This is evident from the data above, which indicates that about 28% of clients left the company. #-The results show that the company has dissatisfactory customers, and it is because of these results that we have concluded that customers who have called customer service 2-4 times have left the company. #-States with a higher rate of churn include Maryland, New Jersey, Michigan, and Texas.

#Recommendations to reduce customer Churn rate :- #- Enhancing client satisfaction through action. #- Using a competitive pricing strategy. #- In the states with a higher churn rate, conducting a thorough market analysis.