Group Project

Khushboo Yadav, Rahkee Moolchandani, Mayank Pugalia, Mark Bruner

11/16/2020

```r
rm(list=ls())
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## corrplot 0.84 loaded

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
# Creating new variable names for the datasets we will use for this assignment.
data(mlc_churn)
cust <- mlc_churn

load("Customers_To_Predict.RData")
```

```
## Warning: namespace 'pbdZMQ' is not available and has been replaced
## by .GlobalEnv when processing object '.pbd_env'
```

```r
test <- data.frame(Custmers_to_predict)
```

Making Life Easier with Variable Names

```r
# Renamed variables to make it easier to work with.
cust$area_code <- as.factor(sub("area_code_", "", cust$area_code))
cust %>%
  rename(
    acct_length = account_length,
    intl_plan = international_plan,
    vm_plan = voice_mail_plan,
    num_vm_mess = number_vmail_messages,
    tot_day_min = total_day_minutes,
    tot_day_calls = total_day_calls,
    tot_day_chg = total_day_charge,
    tot_eve_min = total_eve_minutes,
    tot_eve_calls = total_eve_calls,
    tot_night_min = total_night_minutes,
    tot_night_calls = total_night_calls,
    tot_night_chg = total_night_charge,
    tot_intl_min = total_intl_minutes,
    tot_intl_calls = total_intl_calls,
    tot_intl_chg = total_intl_charge,
    num_cust_serv_calls = number_customer_service_calls
    ) -> cust
cust
```

```
## # A tibble: 5,000 x 20
##    state acct_length area_code intl_plan vm_plan num_vm_mess tot_day_min
##    <fct>       <int> <fct>     <fct>     <fct>         <int>       <dbl>
##  1 KS            128 415       no        yes              25        265.
##  2 OH            107 415       no        yes              26        162.
##  3 NJ            137 415       no        no                0        243.
##  4 OH             84 408       yes       no                0        299.
##  5 OK             75 415       yes       no                0        167.
##  6 AL            118 510       yes       no                0        223.
##  7 MA            121 510       no        yes              24        218.
##  8 MO            147 415       yes       no                0        157
##  9 LA            117 408       no        no                0        184.
## 10 WV            141 415       yes       yes              37        259.
## # ... with 4,990 more rows, and 13 more variables: tot_day_calls <int>,
## #   tot_day_chg <dbl>, tot_eve_min <dbl>, tot_eve_calls <int>,
## #   total_eve_charge <dbl>, tot_night_min <dbl>, tot_night_calls <int>,
## #   tot_night_chg <dbl>, tot_intl_min <dbl>, tot_intl_calls <int>,
## #   tot_intl_chg <dbl>, num_cust_serv_calls <int>, churn <fct>
```

```r
# The types in the dataset all look good but will have to create some dummy variables. Will do that nex
str(cust)
```

```
## tibble [5,000 x 20] (S3: tbl_df/tbl/data.frame)
##  $ state              : Factor w/ 51 levels "AK","AL","AR",..: 17 36 32 36 37 2 20 25 19 50 ...
```

```
##  $ acct_length       : int [1:5000] 128 107 137 84 75 118 121 147 117 141 ...
##  $ area_code         : Factor w/ 3 levels "408","415","510": 2 2 2 1 2 3 3 2 1 2 ...
##  $ intl_plan         : Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 1 2 ...
##  $ vm_plan           : Factor w/ 2 levels "no","yes": 2 2 1 1 1 1 2 1 1 2 ...
##  $ num_vm_mess       : int [1:5000] 25 26 0 0 0 0 24 0 0 37 ...
##  $ tot_day_min       : num [1:5000] 265 162 243 299 167 ...
##  $ tot_day_calls     : int [1:5000] 110 123 114 71 113 98 88 79 97 84 ...
##  $ tot_day_chg       : num [1:5000] 45.1 27.5 41.4 50.9 28.3 ...
##  $ tot_eve_min       : num [1:5000] 197.4 195.5 121.2 61.9 148.3 ...
##  $ tot_eve_calls     : int [1:5000] 99 103 110 88 122 101 108 94 80 111 ...
##  $ total_eve_charge  : num [1:5000] 16.78 16.62 10.3 5.26 12.61 ...
##  $ tot_night_min     : num [1:5000] 245 254 163 197 187 ...
##  $ tot_night_calls   : int [1:5000] 91 103 104 89 121 118 118 96 90 97 ...
##  $ tot_night_chg     : num [1:5000] 11.01 11.45 7.32 8.86 8.41 ...
##  $ tot_intl_min      : num [1:5000] 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
##  $ tot_intl_calls    : int [1:5000] 3 3 5 7 3 6 7 6 4 5 ...
##  $ tot_intl_chg      : num [1:5000] 2.7 3.7 3.29 1.78 2.73 1.7 2.03 1.92 2.35 3.02 ...
##  $ num_cust_serv_calls: int [1:5000] 1 1 0 2 3 0 3 0 1 0 ...
##  $ churn             : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
tail(cust)
```

```
## # A tibble: 6 x 20
##   state acct_length area_code intl_plan vm_plan num_vm_mess tot_day_min
##   <fct>       <int> <fct>     <fct>     <fct>         <int>       <dbl>
## 1 NC             75 408       no        no                0        171.
## 2 HI             50 408       no        yes              40        236.
## 3 WV            152 415       no        no                0        184.
## 4 DC             61 415       no        no                0        141.
## 5 DC            109 510       no        no                0        189.
## 6 VT             86 415       no        yes              34        129.
## # ... with 13 more variables: tot_day_calls <int>, tot_day_chg <dbl>,
## #   tot_eve_min <dbl>, tot_eve_calls <int>, total_eve_charge <dbl>,
## #   tot_night_min <dbl>, tot_night_calls <int>, tot_night_chg <dbl>,
## #   tot_intl_min <dbl>, tot_intl_calls <int>, tot_intl_chg <dbl>,
## #   num_cust_serv_calls <int>, churn <fct>
```

```r
colMeans(is.na(cust))
```

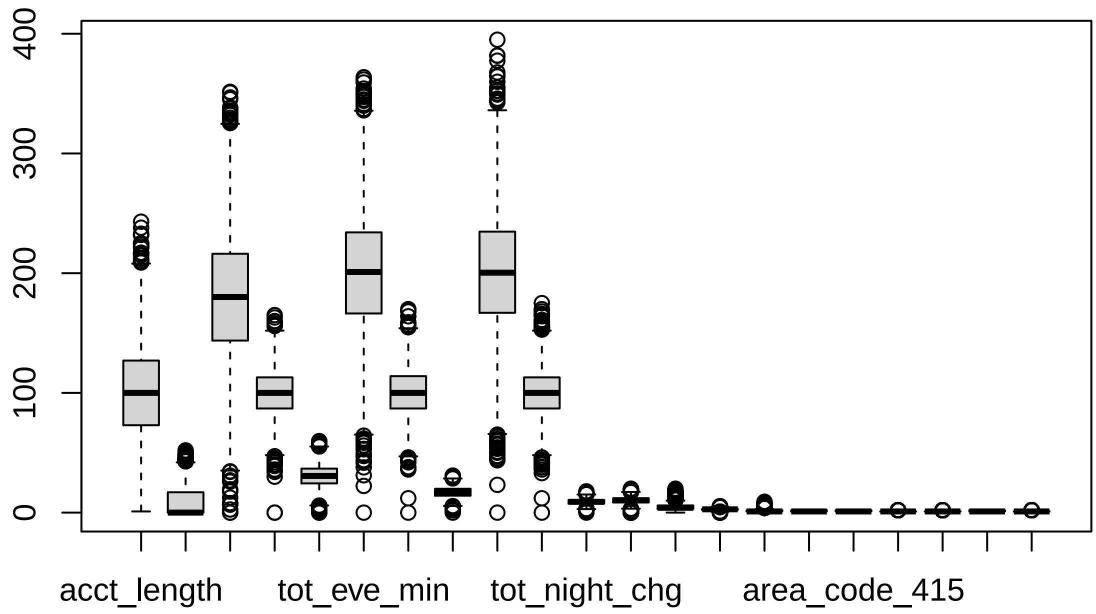
```
##               state         acct_length           area_code           intl_plan
##                   0                   0                   0                   0
##             vm_plan         num_vm_mess         tot_day_min       tot_day_calls
##                   0                   0                   0                   0
##         tot_day_chg         tot_eve_min       tot_eve_calls    total_eve_charge
##                   0                   0                   0                   0
##       tot_night_min     tot_night_calls       tot_night_chg        tot_intl_min
##                   0                   0                   0                   0
##      tot_intl_calls        tot_intl_chg num_cust_serv_calls               churn
##                   0                   0                   0                   0
```

```r
summary(cust)
```

```
##      state       acct_length    area_code  intl_plan  vm_plan
##  WV     : 158   Min.   :  1.0   408:1259   no :4527   no :3677
##  MN     : 125   1st Qu.: 73.0   415:2495   yes: 473   yes:1323
##  AL     : 124   Median :100.0   510:1246
##  ID     : 119   Mean   :100.3
##  VA     : 118   3rd Qu.:127.0
##  OH     : 116   Max.   :243.0
##  (Other):4240
##   num_vm_mess      tot_day_min    tot_day_calls  tot_day_chg     tot_eve_min
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0   Min.   : 0.00   Min.   :  0.0
##  1st Qu.: 0.000   1st Qu.:143.7   1st Qu.: 87   1st Qu.:24.43   1st Qu.:166.4
##  Median : 0.000   Median :180.1   Median :100   Median :30.62   Median :201.0
##  Mean   : 7.755   Mean   :180.3   Mean   :100   Mean   :30.65   Mean   :200.6
##  3rd Qu.:17.000   3rd Qu.:216.2   3rd Qu.:113   3rd Qu.:36.75   3rd Qu.:234.1
##  Max.   :52.000   Max.   :351.5   Max.   :165   Max.   :59.76   Max.   :363.7
##
##  tot_eve_calls   total_eve_charge tot_night_min   tot_night_calls
##  Min.   :  0.0   Min.   : 0.00    Min.   :  0.0   Min.   :  0.00
##  1st Qu.: 87.0   1st Qu.:14.14    1st Qu.:166.9   1st Qu.: 87.00
##  Median :100.0   Median :17.09    Median :200.4   Median :100.00
##  Mean   :100.2   Mean   :17.05    Mean   :200.4   Mean   : 99.92
##  3rd Qu.:114.0   3rd Qu.:19.90    3rd Qu.:234.7   3rd Qu.:113.00
##  Max.   :170.0   Max.   :30.91    Max.   :395.0   Max.   :175.00
##
##  tot_night_chg     tot_intl_min    tot_intl_calls    tot_intl_chg
##  Min.   : 0.000   Min.   : 0.00   Min.   : 0.000   Min.   :0.000
##  1st Qu.: 7.510   1st Qu.: 8.50   1st Qu.: 3.000   1st Qu.:2.300
##  Median : 9.020   Median :10.30   Median : 4.000   Median :2.780
##  Mean   : 9.018   Mean   :10.26   Mean   : 4.435   Mean   :2.771
##  3rd Qu.:10.560   3rd Qu.:12.00   3rd Qu.: 6.000   3rd Qu.:3.240
##  Max.   :17.770   Max.   :20.00   Max.   :20.000   Max.   :5.400
##
##  num_cust_serv_calls churn
##  Min.   :0.00        yes: 707
##  1st Qu.:1.00        no :4293
##  Median :1.00
##  Mean   :1.57
##  3rd Qu.:2.00
##  Max.   :9.00
##
```

```r
# Creating dummy variables for area_code, intl_plan, vm_plan, and churn to separate each of the factors
cust %>%
dummy_cols(c("area_code",
             "intl_plan",
             "vm_plan",
             "churn"),
           remove_selected_columns = TRUE) -> cust

cust <- cust[, c(-20, -22, -25)]

cust[, 17:22] <- lapply(cust[, 17:22], factor)
```

```r
# May be a few outliers but, to be honest, nothing seems so extreme that I think it is worth changing. 
boxplot(cust[, 2:22])
```

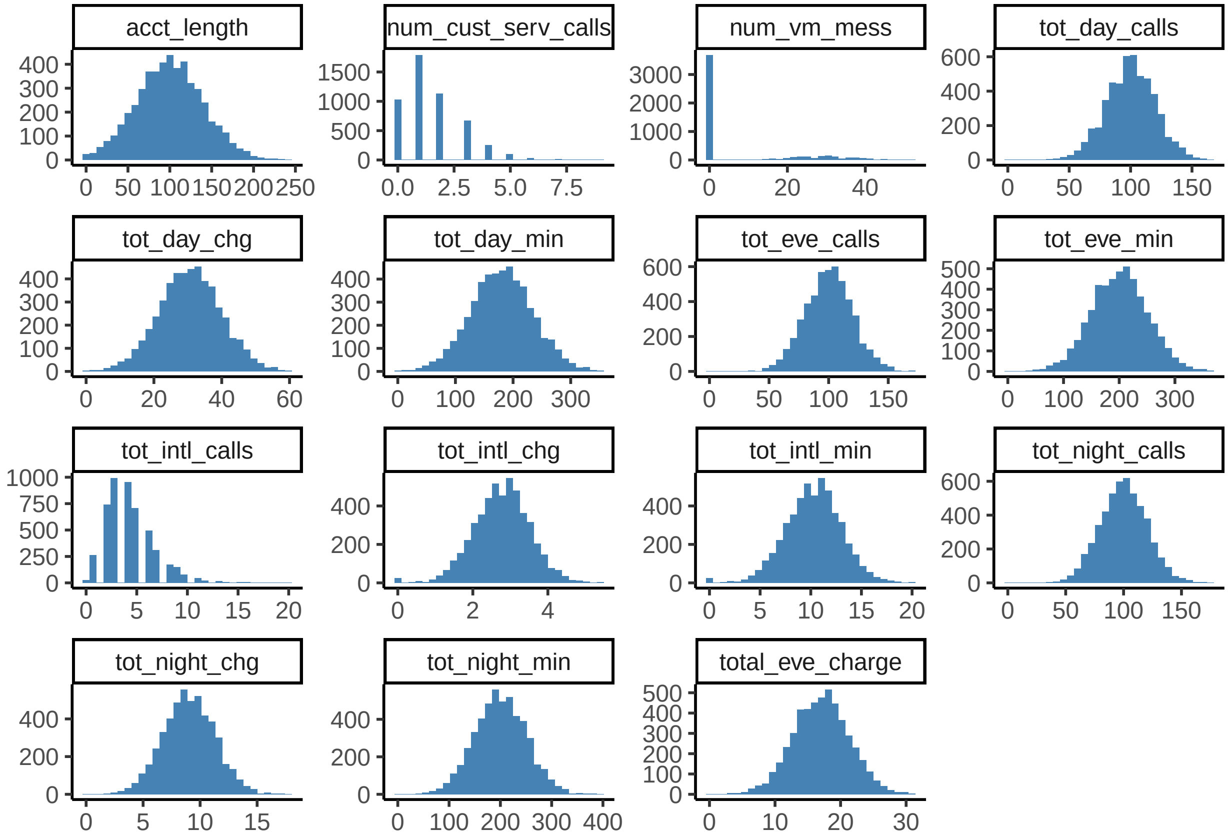



```r
cust %>%
  filter(tot_day_calls < 1)
```

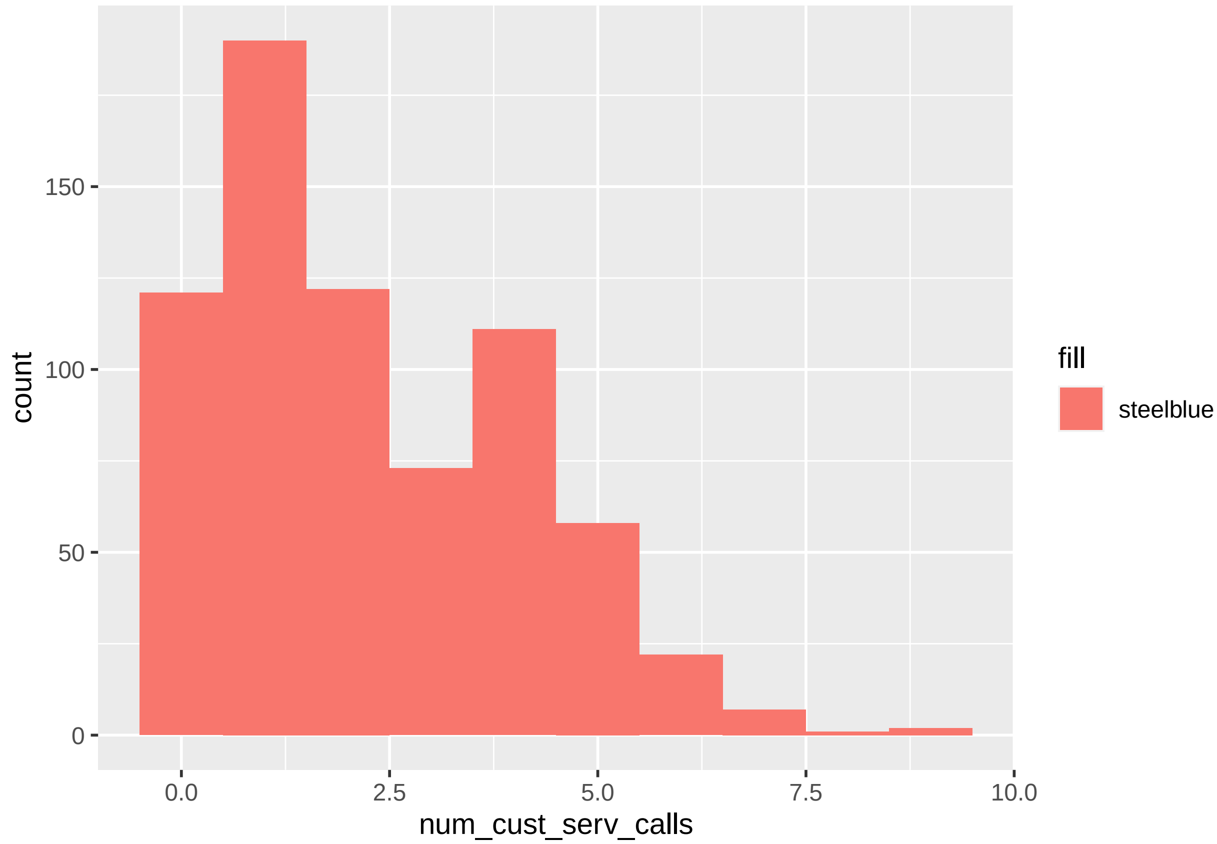
```
## # A tibble: 2 x 22
##   state acct_length num_vm_mess tot_day_min tot_day_calls tot_day_chg
##   <fct>       <int>       <int>       <dbl>         <int>       <dbl>
## 1 SD             98           0           0             0           0
## 2 VT            101           0           0             0           0
## # ... with 16 more variables: tot_eve_min <dbl>, tot_eve_calls <int>,
## #   total_eve_charge <dbl>, tot_night_min <dbl>, tot_night_calls <int>,
## #   tot_night_chg <dbl>, tot_intl_min <dbl>, tot_intl_calls <int>,
## #   tot_intl_chg <dbl>, num_cust_serv_calls <int>, area_code_408 <fct>,
## #   area_code_415 <fct>, area_code_510 <fct>, intl_plan_yes <fct>,
## #   vm_plan_yes <fct>, churn_yes <fct>
```

```r
cust[2:16] %>%
  gather(key = Variable, value = Value) %>%
  ggplot() +
    geom_histogram(aes(x = Value), fill = "steelblue") +
    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'.
```




```
cust %>%
  filter(churn_yes == 1) %>%
  ggplot(mapping = aes(x = num_cust_serv_calls)) +
  geom_histogram(aes(fill = "steelblue"), binwidth = 1)
```




```r
cust %>%
  count(churn_yes)
```

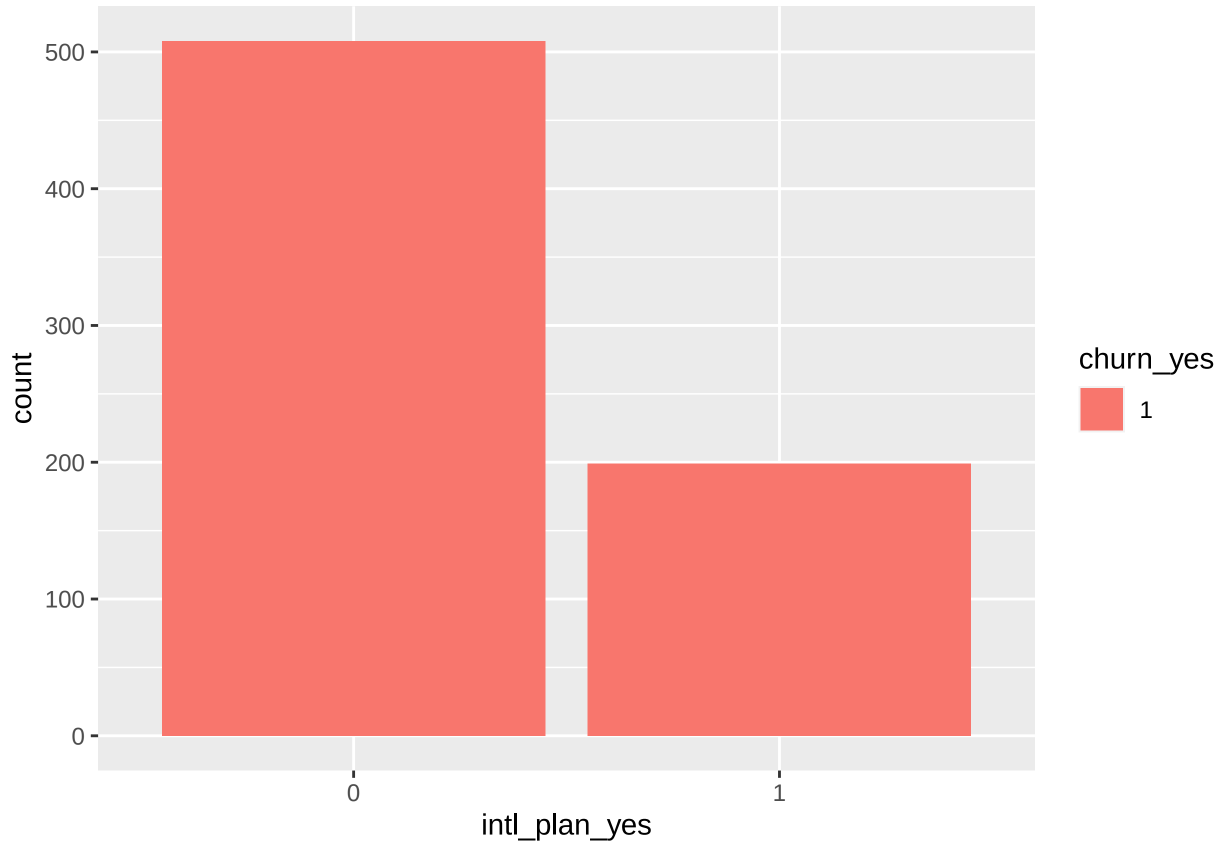
```
## # A tibble: 2 x 2
##   churn_yes     n
##   <fct>     <int>
## 1 0          4293
## 2 1           707
```

```r
cust %>%
 filter(churn_yes == 1 & num_cust_serv_calls >= 1 & num_cust_serv_calls <= 4)%>%
 summarise("% churned after making betw 1 and 4 cust service calls" = n()/707*100)
```

```
## # A tibble: 1 x 1
##   `% churned after making betw 1 and 4 cust service calls`
##                                                      <dbl>
## 1                                                     70.2
```

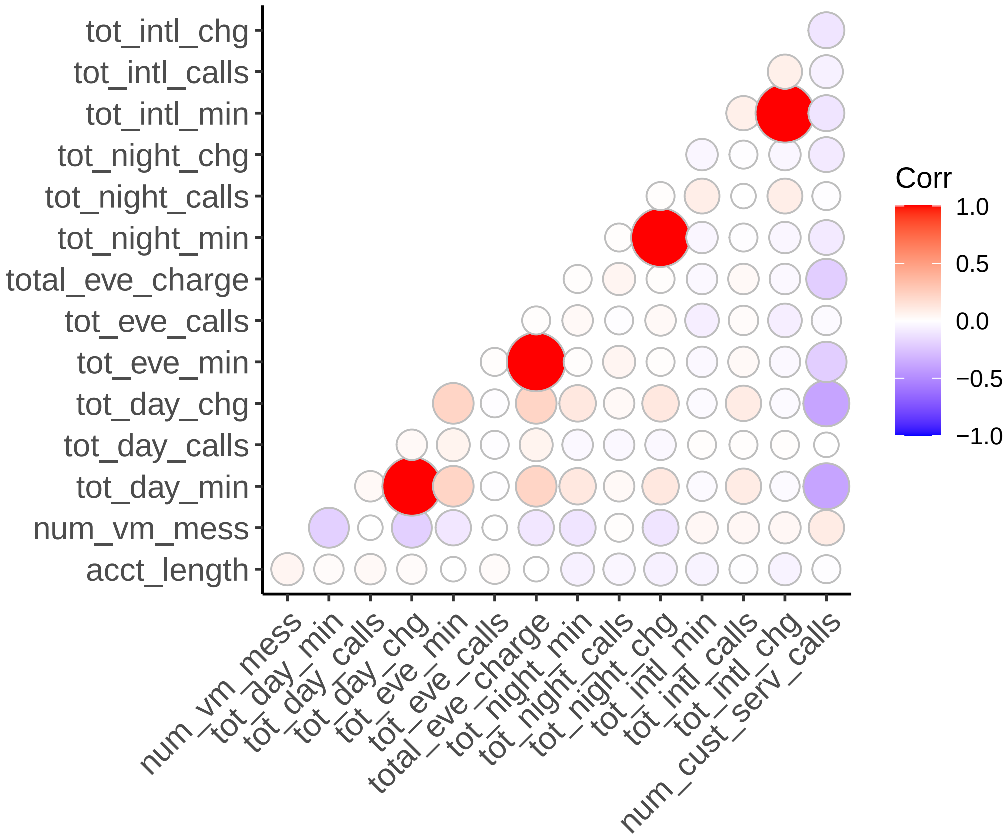
```r
cust %>%
 filter(churn_yes== 1) -> churn_cust

cust %>%
 filter(churn_yes== 1) %>%
 ggplot(mapping = aes(x = intl_plan_yes)) +
 geom_histogram(aes(fill = churn_yes), stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Most customers churn when they make 1 to 3 customer service calls. Also, about 30% of customers churn if they have a international plan.

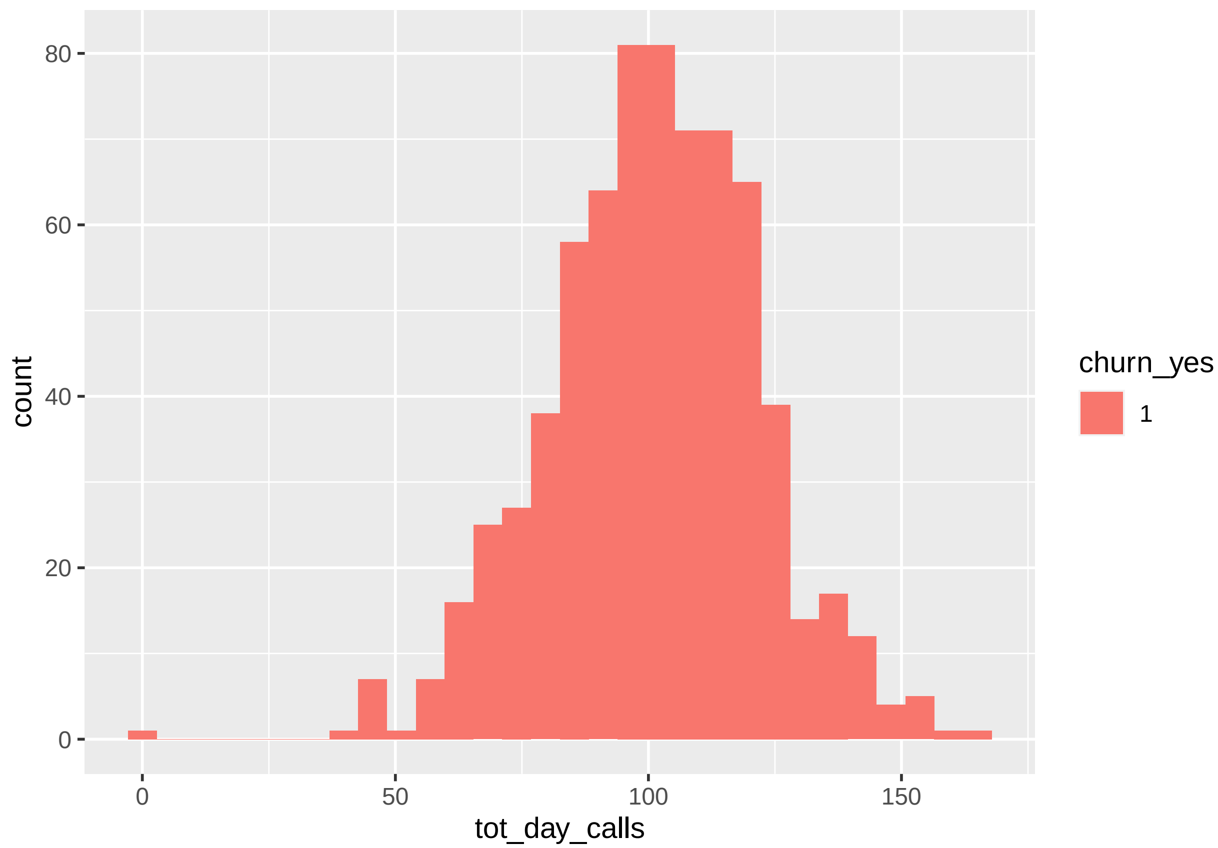
```
cor(churn_cust[, 2:16]) -> cc
ggcorrplot(cc, method = "circle", type = "lower", ggtheme = ggplot2::theme_classic)
```



Some positive correlation between number of customer service calls and total day charges also total day minutes. Actually, most of the variables have some positive correlation to customer service calls except total day calls, total evening calls, account length, and total night calls.

```
churn_cust %>%
  ggplot(mapping = aes(x = tot_day_calls)) +
  geom_histogram(aes(fill = churn_yes))
```

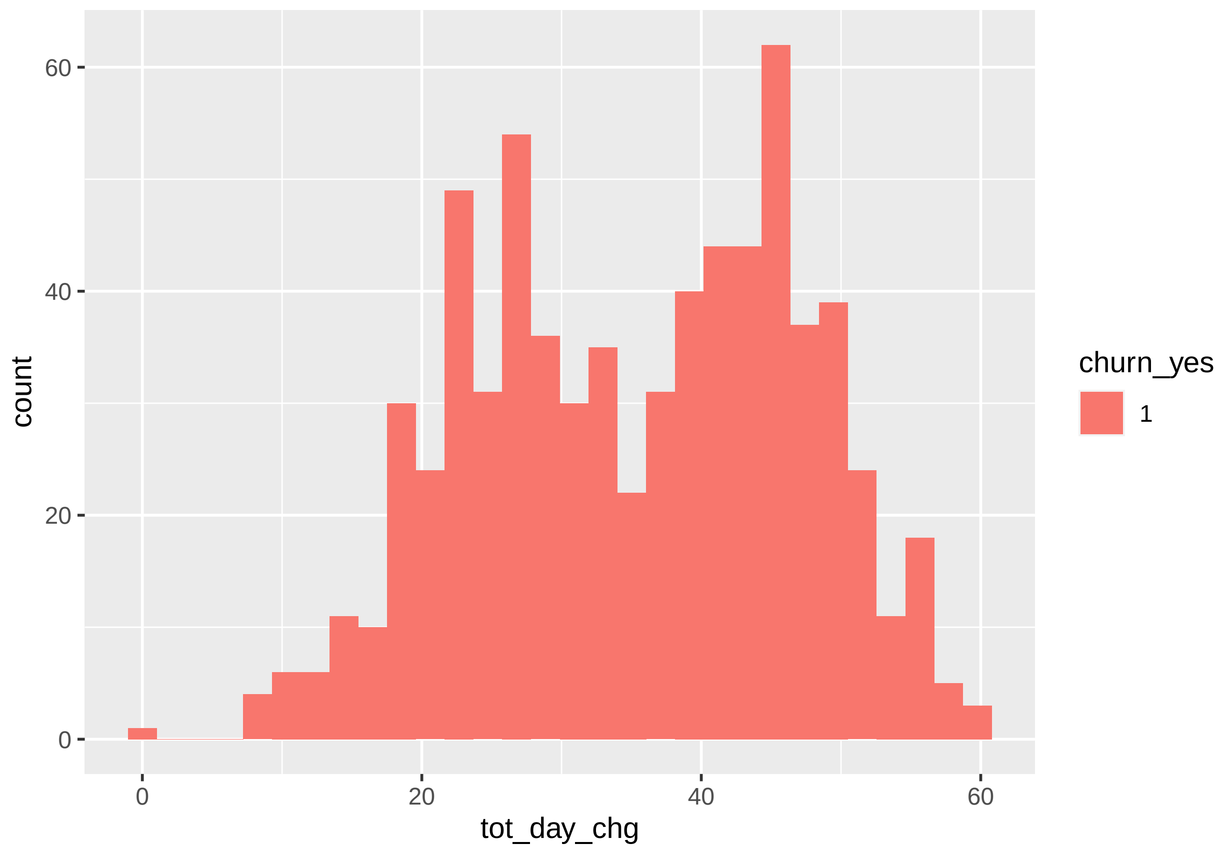
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Most of the people seem to churn between 75 to 125 calls per day.

```
churn_cust %>%
  ggplot(mapping = aes(x = tot_day_chg)) +
  geom_histogram(aes(fill = churn_yes))
```

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



Most people churn when charges are between 20 to 50 per day.

Based on the above, I might suggest that the reason people are churning is that the cost of daily phone call charages during the day are too much. FYI I think this data is really old as I remember when Cell Phone companies used to charge more for calls made during the day than the evening…

```
# Partioning the dataset into train and validation sets.
set.seed(15)
tra_val <- createDataPartition(cust$churn_yes, list = FALSE, p = .8)
train <- cust[tra_val, ]
```

```
## Warning: The 'i' argument of ''['()' can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_warnings()' to see where this warning was generated.
```

```
valid <- cust[-tra_val, ]
```

```
norm <- preProcess(train, method = c("scale", "center"))
train <- predict(norm, train)

summary(train)
```

```
##      state        acct_length       num_vm_mess       tot_day_min
##  WV     : 124   Min.   :-2.4987   Min.   :-0.5762   Min.   :-3.339290
```

```
##  MN     : 104   1st Qu.:-0.6923   1st Qu.:-0.5762   1st Qu.:-0.677937
##  AL     :  97   Median :-0.0149   Median :-0.5762   Median : 0.001754
##  OH     :  96   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.000000
##  ID     :  93   3rd Qu.: 0.6625   3rd Qu.: 0.6733   3rd Qu.: 0.670334
##  NJ     :  93   Max.   : 3.5728   Max.   : 3.2456   Max.   : 3.170560
##  (Other):3394
##  tot_day_calls       tot_day_chg         tot_eve_min       tot_eve_calls
##  Min.   :-5.075602   Min.   :-3.339354   Min.   :-3.53701   Min.   :-4.458807
##  1st Qu.:-0.661373   1st Qu.:-0.677890   1st Qu.:-0.67859   1st Qu.:-0.663570
##  Median :-0.001775   Median : 0.001911   Median : 0.01035   Median :-0.005729
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.000000
##  3rd Qu.: 0.657822   3rd Qu.: 0.670817   3rd Qu.: 0.67166   3rd Qu.: 0.702715
##  Max.   : 3.296212   Max.   : 3.171046   Max.   : 3.20239   Max.   : 3.536491
##
##  total_eve_charge  tot_night_min      tot_night_calls     tot_night_chg
##  Min.   :-3.5361   Min.   :-3.09433   Min.   :-4.387094   Min.   :-3.09284
##  1st Qu.:-0.6795   1st Qu.:-0.66675   1st Qu.:-0.650206   1st Qu.:-0.66505
##  Median : 0.0103   Median :-0.01005   Median :-0.002478   Median :-0.01209
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.00000
##  3rd Qu.: 0.6722   3rd Qu.: 0.67622   3rd Qu.: 0.695074   3rd Qu.: 0.67593
##  Max.   : 3.2013   Max.   : 3.57512   Max.   : 3.734411   Max.   : 3.57700
##
##   tot_intl_min      tot_intl_calls     tot_intl_chg       num_cust_serv_calls
##  Min.   :-3.68557   Min.   :-1.8180   Min.   :-3.686659   Min.   :-1.1938
##  1st Qu.:-0.63440   1st Qu.:-0.5874   1st Qu.:-0.628489   1st Qu.:-0.4310
##  Median : 0.01173   Median :-0.1773   Median : 0.009737   Median :-0.4310
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.0000
##  3rd Qu.: 0.62196   3rd Qu.: 0.6431   3rd Qu.: 0.621371   3rd Qu.: 0.3317
##  Max.   : 3.38596   Max.   : 6.3857   Max.   : 3.387020   Max.   : 5.6710
##
##  area_code_408 area_code_415 area_code_510 intl_plan_yes vm_plan_yes churn_yes
##  0:2995        0:1985        0:3022        0:3611        0:2932      0:3435
##  1:1006        1:2016        1: 979        1: 390        1:1069      1: 566
##
##
##
##
##
```

```r
# Creating a model for logistic regression based upon all the variables. I will create another logistic
model1 <- glm(churn_yes~., family = "binomial", data = train)
summary(model1)
```

```
##
## Call:
## glm(formula = churn_yes ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1239  -0.4847  -0.3016  -0.1628   3.2176
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -3.134e+00  6.580e-01  -4.763 1.91e-06 ***
```

```
## stateAL            5.381e-01  7.421e-01   0.725 0.468356    
## stateAR            9.838e-01  7.519e-01   1.308 0.190740    
## stateAZ            7.306e-01  7.745e-01   0.943 0.345549    
## stateCA            2.010e+00  7.516e-01   2.675 0.007483 ** 
## stateCO            6.791e-01  7.625e-01   0.891 0.373157    
## stateCT            1.226e+00  7.242e-01   1.693 0.090393 .  
## stateDC            7.744e-01  7.742e-01   1.000 0.317205    
## stateDE            4.898e-01  7.551e-01   0.649 0.516549    
## stateFL            6.173e-01  7.596e-01   0.813 0.416423    
## stateGA            8.287e-01  7.573e-01   1.094 0.273803    
## stateHI            2.882e-01  8.066e-01   0.357 0.720808    
## stateIA            9.718e-01  7.905e-01   1.229 0.218904    
## stateID            6.985e-01  7.411e-01   0.943 0.345934    
## stateIL            7.138e-02  8.042e-01   0.089 0.929274    
## stateIN            6.697e-01  7.472e-01   0.896 0.370096    
## stateKS            8.408e-01  7.412e-01   1.134 0.256606    
## stateKY            9.899e-01  7.392e-01   1.339 0.180531    
## stateLA            7.616e-01  8.043e-01   0.947 0.343660    
## stateMA            1.224e+00  7.280e-01   1.682 0.092645 .  
## stateMD            1.123e+00  7.135e-01   1.574 0.115391    
## stateME            1.465e+00  7.150e-01   2.048 0.040516 *  
## stateMI            1.484e+00  7.143e-01   2.078 0.037713 *  
## stateMN            1.158e+00  7.101e-01   1.631 0.102931    
## stateMO            7.419e-01  7.605e-01   0.975 0.329335    
## stateMS            1.094e+00  7.291e-01   1.500 0.133641    
## stateMT            2.055e+00  7.025e-01   2.925 0.003445 ** 
## stateNC            5.518e-01  7.545e-01   0.731 0.464593    
## stateND            2.730e-01  7.925e-01   0.344 0.730540    
## stateNE            3.153e-01  8.033e-01   0.392 0.694705    
## stateNH            9.399e-01  7.440e-01   1.263 0.206458    
## stateNJ            1.693e+00  6.965e-01   2.431 0.015070 *  
## stateNM            6.506e-01  7.641e-01   0.851 0.394494    
## stateNV            1.454e+00  7.186e-01   2.023 0.043095 *  
## stateNY            1.043e+00  7.233e-01   1.441 0.149460    
## stateOH            1.058e+00  7.213e-01   1.466 0.142528    
## stateOK            7.562e-01  7.503e-01   1.008 0.313511    
## stateOR            1.037e+00  7.173e-01   1.446 0.148198    
## statePA            2.942e-01  8.214e-01   0.358 0.720230    
## stateRI            6.277e-02  8.026e-01   0.078 0.937655    
## stateSC            1.393e+00  7.424e-01   1.877 0.060543 .  
## stateSD            6.270e-01  7.728e-01   0.811 0.417228    
## stateTN            1.081e+00  7.401e-01   1.460 0.144266    
## stateTX            1.500e+00  7.045e-01   2.129 0.033279 *  
## stateUT            1.442e+00  7.183e-01   2.007 0.044771 *  
## stateVA            8.492e-03  7.909e-01   0.011 0.991433    
## stateVT            3.919e-01  7.555e-01   0.519 0.603960    
## stateWA            2.046e+00  7.044e-01   2.905 0.003677 ** 
## stateWI            1.434e-01  8.005e-01   0.179 0.857845    
## stateWV            1.173e+00  6.977e-01   1.681 0.092835 .  
## stateWY            1.635e-01  7.573e-01   0.216 0.829065    
## acct_length        7.681e-02  5.268e-02   1.458 0.144851    
## num_vm_mess        3.424e-01  2.331e-01   1.469 0.141906    
## tot_day_min        1.869e+02  1.696e+02   1.102 0.270331    
## tot_day_calls      9.778e-03  5.284e-02   0.185 0.853208    
```

```
## tot_day_chg         -1.861e+02  1.695e+02  -1.098 0.272310
## tot_eve_min          1.387e+00  8.036e+01   0.017 0.986233
## tot_eve_calls       -2.516e-02  5.352e-02  -0.470 0.638184
## total_eve_charge    -9.732e-01  8.036e+01  -0.012 0.990338
## tot_night_min        6.899e+00  4.250e+01   0.162 0.871041
## tot_night_calls     -4.381e-02  5.308e-02  -0.825 0.409216
## tot_night_chg       -6.663e+00  4.250e+01  -0.157 0.875406
## tot_intl_min         6.686e+00  1.416e+01   0.472 0.636805
## tot_intl_calls      -1.746e-01  5.676e-02  -3.076 0.002097 **
## tot_intl_chg        -6.428e+00  1.416e+01  -0.454 0.649851
## num_cust_serv_calls  6.876e-01  4.975e-02  13.822  < 2e-16 ***
## area_code_4081       6.734e-02  1.520e-01   0.443 0.657757
## area_code_4151       2.134e-03  1.315e-01   0.016 0.987050
## area_code_5101              NA         NA      NA       NA
## intl_plan_yes1       2.328e+00  1.430e-01  16.282  < 2e-16 ***
## vm_plan_yes1        -1.854e+00  5.462e-01  -3.394 0.000689 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3261.7  on 4000  degrees of freedom
## Residual deviance: 2409.9  on 3931  degrees of freedom
## AIC: 2549.9
##
## Number of Fisher Scoring iterations: 6
```

```r
varImp(model1)
```

```
##                        Overall
## stateAL             0.72515618
## stateAR             1.30839248
## stateAZ             0.94325869
## stateCA             2.67456524
## stateCO             0.89057595
## stateCT             1.69333043
## stateDC             1.00021843
## stateDE             0.64867439
## stateFL             0.81264286
## stateGA             1.09434753
## stateHI             0.35737948
## stateIA             1.22944797
## stateID             0.94250571
## stateIL             0.08875791
## stateIN             0.89629437
## stateKS             1.13444916
## stateKY             1.33912093
## stateLA             0.94695782
## stateMA             1.68160723
## stateMD             1.57441732
## stateME             2.04844752
## stateMI             2.07796097
## stateMN             1.63080845
## stateMO             0.97545380
```

```
## stateMS              1.49989795
## stateMT              2.92493254
## stateNC              0.73130415
## stateND              0.34440742
## stateNE              0.39247764
## stateNH              1.26336646
## stateNJ              2.43069533
## stateNM              0.85149651
## stateNV              2.02278929
## stateNY              1.44144154
## stateOH              1.46644000
## stateOK              1.00788312
## stateOR              1.44592752
## statePA              0.35815114
## stateRI              0.07821706
## stateSC              1.87682117
## stateSD              0.81123889
## stateTN              1.46008886
## stateTX              2.12870520
## stateUT              2.00679581
## stateVA              0.01073699
## stateVT              0.51871441
## stateWA              2.90458280
## stateWI              0.17911758
## stateWV              1.68062764
## stateWY              0.21590068
## acct_length          1.45796170
## num_vm_mess          1.46873096
## tot_day_min          1.10230009
## tot_day_calls        0.18502634
## tot_day_chg          1.09775899
## tot_eve_min          0.01725559
## tot_eve_calls        0.47023905
## total_eve_charge     0.01210992
## tot_night_min        0.16233690
## tot_night_calls      0.82527450
## tot_night_chg        0.15679612
## tot_intl_min         0.47217001
## tot_intl_calls       3.07618309
## tot_intl_chg         0.45396852
## num_cust_serv_calls 13.82170417
## area_code_4081       0.44301226
## area_code_4151       0.01623066
## intl_plan_yes1      16.28176408
## vm_plan_yes1         3.39400328
```

```
# I ran a grid search algorithm and the best AIC model was the one below.
model3 <- glm(churn_yes ~ acct_length + num_vm_mess + tot_day_min + tot_day_calls +
    tot_day_chg + tot_eve_min + total_eve_charge + tot_night_min +
    tot_night_chg + tot_intl_min + tot_intl_calls + tot_intl_chg +
    num_cust_serv_calls + area_code_415 + intl_plan_yes + vm_plan_yes +
    tot_day_min:num_cust_serv_calls + tot_day_min:tot_day_chg +
    tot_intl_min:intl_plan_yes + tot_eve_min:num_cust_serv_calls +
    tot_day_min:vm_plan_yes + tot_day_min:tot_eve_min + tot_day_chg:tot_night_min +
```

```r
    tot_intl_calls:intl_plan_yes + tot_day_chg:intl_plan_yes +
    tot_eve_min:vm_plan_yes + num_cust_serv_calls:intl_plan_yes +
    tot_night_chg:vm_plan_yes + tot_night_min:num_cust_serv_calls +
    acct_length:num_vm_mess + total_eve_charge:tot_night_min +
    tot_intl_min:tot_intl_calls + num_cust_serv_calls:vm_plan_yes +
    tot_day_calls:total_eve_charge + tot_night_min:vm_plan_yes +
    total_eve_charge:num_cust_serv_calls + intl_plan_yes:vm_plan_yes +
    num_vm_mess:area_code_415 + tot_eve_min:total_eve_charge +
    tot_intl_chg:num_cust_serv_calls + tot_day_calls:num_cust_serv_calls +
    tot_day_chg:num_cust_serv_calls + tot_intl_calls:vm_plan_yes +
    tot_eve_min:tot_night_chg + tot_day_min:tot_night_chg + acct_length:tot_night_chg +
    acct_length:tot_night_min + tot_intl_calls:num_cust_serv_calls, family = "binomial", data = train)


model1 <- glm(churn_yes~., family = "binomial", data = train)
summary(model1)
```

```
##
## Call:
## glm(formula = churn_yes ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1239  -0.4847  -0.3016  -0.1628   3.2176
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.134e+00  6.580e-01  -4.763 1.91e-06 ***
## stateAL               5.381e-01  7.421e-01   0.725 0.468356
## stateAR               9.838e-01  7.519e-01   1.308 0.190740
## stateAZ               7.306e-01  7.745e-01   0.943 0.345549
## stateCA               2.010e+00  7.516e-01   2.675 0.007483 **
## stateCO               6.791e-01  7.625e-01   0.891 0.373157
## stateCT               1.226e+00  7.242e-01   1.693 0.090393 .
## stateDC               7.744e-01  7.742e-01   1.000 0.317205
## stateDE               4.898e-01  7.551e-01   0.649 0.516549
## stateFL               6.173e-01  7.596e-01   0.813 0.416423
## stateGA               8.287e-01  7.573e-01   1.094 0.273803
## stateHI               2.882e-01  8.066e-01   0.357 0.720808
## stateIA               9.718e-01  7.905e-01   1.229 0.218904
## stateID               6.985e-01  7.411e-01   0.943 0.345934
## stateIL               7.138e-02  8.042e-01   0.089 0.929274
## stateIN               6.697e-01  7.472e-01   0.896 0.370096
## stateKS               8.408e-01  7.412e-01   1.134 0.256606
## stateKY               9.899e-01  7.392e-01   1.339 0.180531
## stateLA               7.616e-01  8.043e-01   0.947 0.343660
## stateMA               1.224e+00  7.280e-01   1.682 0.092645 .
## stateMD               1.123e+00  7.135e-01   1.574 0.115391
## stateME               1.465e+00  7.150e-01   2.048 0.040516 *
## stateMI               1.484e+00  7.143e-01   2.078 0.037713 *
## stateMN               1.158e+00  7.101e-01   1.631 0.102931
## stateMO               7.419e-01  7.605e-01   0.975 0.329335
## stateMS               1.094e+00  7.291e-01   1.500 0.133641
```

```
## stateMT              2.055e+00  7.025e-01   2.925 0.003445 **
## stateNC              5.518e-01  7.545e-01   0.731 0.464593
## stateND              2.730e-01  7.925e-01   0.344 0.730540
## stateNE              3.153e-01  8.033e-01   0.392 0.694705
## stateNH              9.399e-01  7.440e-01   1.263 0.206458
## stateNJ              1.693e+00  6.965e-01   2.431 0.015070 *
## stateNM              6.506e-01  7.641e-01   0.851 0.394494
## stateNV              1.454e+00  7.186e-01   2.023 0.043095 *
## stateNY              1.043e+00  7.233e-01   1.441 0.149460
## stateOH              1.058e+00  7.213e-01   1.466 0.142528
## stateOK              7.562e-01  7.503e-01   1.008 0.313511
## stateOR              1.037e+00  7.173e-01   1.446 0.148198
## statePA              2.942e-01  8.214e-01   0.358 0.720230
## stateRI              6.277e-02  8.026e-01   0.078 0.937655
## stateSC              1.393e+00  7.424e-01   1.877 0.060543 .
## stateSD              6.270e-01  7.728e-01   0.811 0.417228
## stateTN              1.081e+00  7.401e-01   1.460 0.144266
## stateTX              1.500e+00  7.045e-01   2.129 0.033279 *
## stateUT              1.442e+00  7.183e-01   2.007 0.044771 *
## stateVA              8.492e-03  7.909e-01   0.011 0.991433
## stateVT              3.919e-01  7.555e-01   0.519 0.603960
## stateWA              2.046e+00  7.044e-01   2.905 0.003677 **
## stateWI              1.434e-01  8.005e-01   0.179 0.857845
## stateWV              1.173e+00  6.977e-01   1.681 0.092835 .
## stateWY              1.635e-01  7.573e-01   0.216 0.829065
## acct_length          7.681e-02  5.268e-02   1.458 0.144851
## num_vm_mess          3.424e-01  2.331e-01   1.469 0.141906
## tot_day_min          1.869e+02  1.696e+02   1.102 0.270331
## tot_day_calls        9.778e-03  5.284e-02   0.185 0.853208
## tot_day_chg         -1.861e+02  1.695e+02  -1.098 0.272310
## tot_eve_min          1.387e+00  8.036e+01   0.017 0.986233
## tot_eve_calls       -2.516e-02  5.352e-02  -0.470 0.638184
## total_eve_charge    -9.732e-01  8.036e+01  -0.012 0.990338
## tot_night_min        6.899e+00  4.250e+01   0.162 0.871041
## tot_night_calls     -4.381e-02  5.308e-02  -0.825 0.409216
## tot_night_chg       -6.663e+00  4.250e+01  -0.157 0.875406
## tot_intl_min         6.686e+00  1.416e+01   0.472 0.636805
## tot_intl_calls      -1.746e-01  5.676e-02  -3.076 0.002097 **
## tot_intl_chg        -6.428e+00  1.416e+01  -0.454 0.649851
## num_cust_serv_calls  6.876e-01  4.975e-02  13.822  < 2e-16 ***
## area_code_4081       6.734e-02  1.520e-01   0.443 0.657757
## area_code_4151       2.134e-03  1.315e-01   0.016 0.987050
## area_code_5101              NA         NA      NA       NA
## intl_plan_yes1       2.328e+00  1.430e-01  16.282  < 2e-16 ***
## vm_plan_yes1        -1.854e+00  5.462e-01  -3.394 0.000689 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3261.7  on 4000  degrees of freedom
## Residual deviance: 2409.9  on 3931  degrees of freedom
## AIC: 2549.9
##
```

```
## Number of Fisher Scoring iterations: 6
```

```r
summary(model3) # This model performs well as many of the variables are significant statistically. Also
```

```
##
## Call:
## glm(formula = churn_yes ~ acct_length + num_vm_mess + tot_day_min +
##     tot_day_calls + tot_day_chg + tot_eve_min + total_eve_charge +
##     tot_night_min + tot_night_chg + tot_intl_min + tot_intl_calls +
##     tot_intl_chg + num_cust_serv_calls + area_code_415 + intl_plan_yes +
##     vm_plan_yes + tot_day_min:num_cust_serv_calls + tot_day_min:tot_day_chg +
##     tot_intl_min:intl_plan_yes + tot_eve_min:num_cust_serv_calls +
##     tot_day_min:vm_plan_yes + tot_day_min:tot_eve_min + tot_day_chg:tot_night_min +
##     tot_intl_calls:intl_plan_yes + tot_day_chg:intl_plan_yes +
##     tot_eve_min:vm_plan_yes + num_cust_serv_calls:intl_plan_yes +
##     tot_night_chg:vm_plan_yes + tot_night_min:num_cust_serv_calls +
##     acct_length:num_vm_mess + total_eve_charge:tot_night_min +
##     tot_intl_min:tot_intl_calls + num_cust_serv_calls:vm_plan_yes +
##     tot_day_calls:total_eve_charge + tot_night_min:vm_plan_yes +
##     total_eve_charge:num_cust_serv_calls + intl_plan_yes:vm_plan_yes +
##     num_vm_mess:area_code_415 + tot_eve_min:total_eve_charge +
##     tot_intl_chg:num_cust_serv_calls + tot_day_calls:num_cust_serv_calls +
##     tot_day_chg:num_cust_serv_calls + tot_intl_calls:vm_plan_yes +
##     tot_eve_min:tot_night_chg + tot_day_min:tot_night_chg + acct_length:tot_night_chg +
##     acct_length:tot_night_min + tot_intl_calls:num_cust_serv_calls,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3815  -0.3639  -0.1651  -0.0624   3.8560
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -3.56906    0.22760 -15.681  < 2e-16 ***
## acct_length                            0.18630    0.06654   2.800  0.00512 **
## num_vm_mess                            0.38983    0.28765   1.355  0.17534
## tot_day_min                          518.56670  219.04406   2.367  0.01791 *
## tot_day_calls                         -0.06970    0.06926  -1.006  0.31423
## tot_day_chg                         -517.28606  219.04097  -2.362  0.01820 *
## tot_eve_min                           81.98465  103.59169   0.791  0.42870
## total_eve_charge                     -81.35228  103.59327  -0.785  0.43227
## tot_night_min                        -50.39245   58.25188  -0.865  0.38700
## tot_night_chg                         50.69713   58.25040   0.870  0.38412
## tot_intl_min                           7.24003   17.05365   0.425  0.67117
## tot_intl_calls                        -0.07154    0.08303  -0.862  0.38887
## tot_intl_chg                          -7.05646   17.05470  -0.414  0.67905
## num_cust_serv_calls                    1.18702    0.08052  14.742  < 2e-16 ***
## area_code_4151                        -0.10988    0.13296  -0.826  0.40857
## intl_plan_yes1                         2.66168    0.20114  13.233  < 2e-16 ***
## vm_plan_yes1                          -2.12068    0.69434  -3.054  0.00226 **
## tot_day_min:num_cust_serv_calls     -299.31623  175.78324  -1.703  0.08861 .
## tot_day_min:tot_day_chg                0.54287    0.04662  11.644  < 2e-16 ***
## tot_intl_min:intl_plan_yes1            1.35657    0.19794   6.854 7.20e-12 ***
## tot_eve_min:num_cust_serv_calls     -159.65789   85.29265  -1.872  0.06122 .
```

```
## tot_day_min:vm_plan_yes1                 -1.07864    0.16497  -6.539 6.21e-11 ***
## tot_day_min:tot_eve_min                   0.47061    0.06323   7.443 9.85e-14 ***
## tot_day_chg:tot_night_min                69.77118   46.95393   1.486  0.13729
## tot_intl_calls:intl_plan_yes1            -0.96848    0.17788  -5.444 5.20e-08 ***
## tot_day_chg:intl_plan_yes1               -0.73092    0.16220  -4.506 6.60e-06 ***
## tot_eve_min:vm_plan_yes1                 -0.57258    0.17741  -3.227  0.00125 **
## num_cust_serv_calls:intl_plan_yes1       -0.76519    0.15509  -4.934 8.06e-07 ***
## tot_night_chg:vm_plan_yes1             -318.60461  142.02538  -2.243  0.02488 *
## tot_night_min:num_cust_serv_calls        -0.18921    0.05879  -3.218  0.00129 **
## acct_length:num_vm_mess                   0.20644    0.07825   2.638  0.00834 **
## total_eve_charge:tot_night_min           71.19028   44.60521   1.596  0.11049
## tot_intl_min:tot_intl_calls               0.14273    0.06496   2.197  0.02802 *
## num_cust_serv_calls:vm_plan_yes1          0.43939    0.15672   2.804  0.00505 **
## tot_day_calls:total_eve_charge            0.16134    0.06516   2.476  0.01329 *
## tot_night_min:vm_plan_yes1              318.06833  142.01489   2.240  0.02511 *
## total_eve_charge:num_cust_serv_calls    159.28413   85.29617   1.867  0.06184 .
## intl_plan_yes1:vm_plan_yes1               0.92269    0.40597   2.273  0.02304 *
## num_vm_mess:area_code_4151               -0.29807    0.15230  -1.957  0.05034 .
## tot_eve_min:total_eve_charge              0.10734    0.04756   2.257  0.02401 *
## tot_intl_chg:num_cust_serv_calls         -0.11584    0.05983  -1.936  0.05285 .
## tot_day_calls:num_cust_serv_calls         0.08847    0.05668   1.561  0.11854
## tot_day_chg:num_cust_serv_calls         298.51828  175.78054   1.698  0.08946 .
## tot_intl_calls:vm_plan_yes1               0.30881    0.17724   1.742  0.08145 .
## tot_eve_min:tot_night_chg               -71.01069   44.60234  -1.592  0.11137
## tot_day_min:tot_night_chg               -69.43228   46.95412  -1.479  0.13921
## acct_length:tot_night_chg               -88.48917   53.17043  -1.664  0.09606 .
## acct_length:tot_night_min                88.39733   53.17298   1.662  0.09642 .
## tot_intl_calls:num_cust_serv_calls       -0.08955    0.06284  -1.425  0.15414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3261.7  on 4000  degrees of freedom
## Residual deviance: 1711.3  on 3952  degrees of freedom
## AIC: 1809.3
##
## Number of Fisher Scoring iterations: 7
```

```r
anova(model1, model3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: churn_yes ~ state + acct_length + num_vm_mess + tot_day_min +
##     tot_day_calls + tot_day_chg + tot_eve_min + tot_eve_calls +
##     total_eve_charge + tot_night_min + tot_night_calls + tot_night_chg +
##     tot_intl_min + tot_intl_calls + tot_intl_chg + num_cust_serv_calls +
##     area_code_408 + area_code_415 + area_code_510 + intl_plan_yes +
##     vm_plan_yes
## Model 2: churn_yes ~ acct_length + num_vm_mess + tot_day_min + tot_day_calls +
##     tot_day_chg + tot_eve_min + total_eve_charge + tot_night_min +
##     tot_night_chg + tot_intl_min + tot_intl_calls + tot_intl_chg +
##     num_cust_serv_calls + area_code_415 + intl_plan_yes + vm_plan_yes +
##     tot_day_min:num_cust_serv_calls + tot_day_min:tot_day_chg +
```

```
##     tot_intl_min:intl_plan_yes + tot_eve_min:num_cust_serv_calls +
##     tot_day_min:vm_plan_yes + tot_day_min:tot_eve_min + tot_day_chg:tot_night_min +
##     tot_intl_calls:intl_plan_yes + tot_day_chg:intl_plan_yes +
##     tot_eve_min:vm_plan_yes + num_cust_serv_calls:intl_plan_yes +
##     tot_night_chg:vm_plan_yes + tot_night_min:num_cust_serv_calls +
##     acct_length:num_vm_mess + total_eve_charge:tot_night_min +
##     tot_intl_min:tot_intl_calls + num_cust_serv_calls:vm_plan_yes +
##     tot_day_calls:total_eve_charge + tot_night_min:vm_plan_yes +
##     total_eve_charge:num_cust_serv_calls + intl_plan_yes:vm_plan_yes +
##     num_vm_mess:area_code_415 + tot_eve_min:total_eve_charge +
##     tot_intl_chg:num_cust_serv_calls + tot_day_calls:num_cust_serv_calls +
##     tot_day_chg:num_cust_serv_calls + tot_intl_calls:vm_plan_yes +
##     tot_eve_min:tot_night_chg + tot_day_min:tot_night_chg + acct_length:tot_night_chg +
##     acct_length:tot_night_min + tot_intl_calls:num_cust_serv_calls
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      3931     2409.9
## 2      3952     1711.3 -21   698.62
```

```r
# A test IIA hypothesis (independence of irrelevant alternatives) for a multinomial logit model. Basica
list(model1 = pR2(model1)["McFadden"],
     model3 = pR2(model3)["McFadden"])
```

```
## fitting null model for pseudo-r2
## fitting null model for pseudo-r2
```

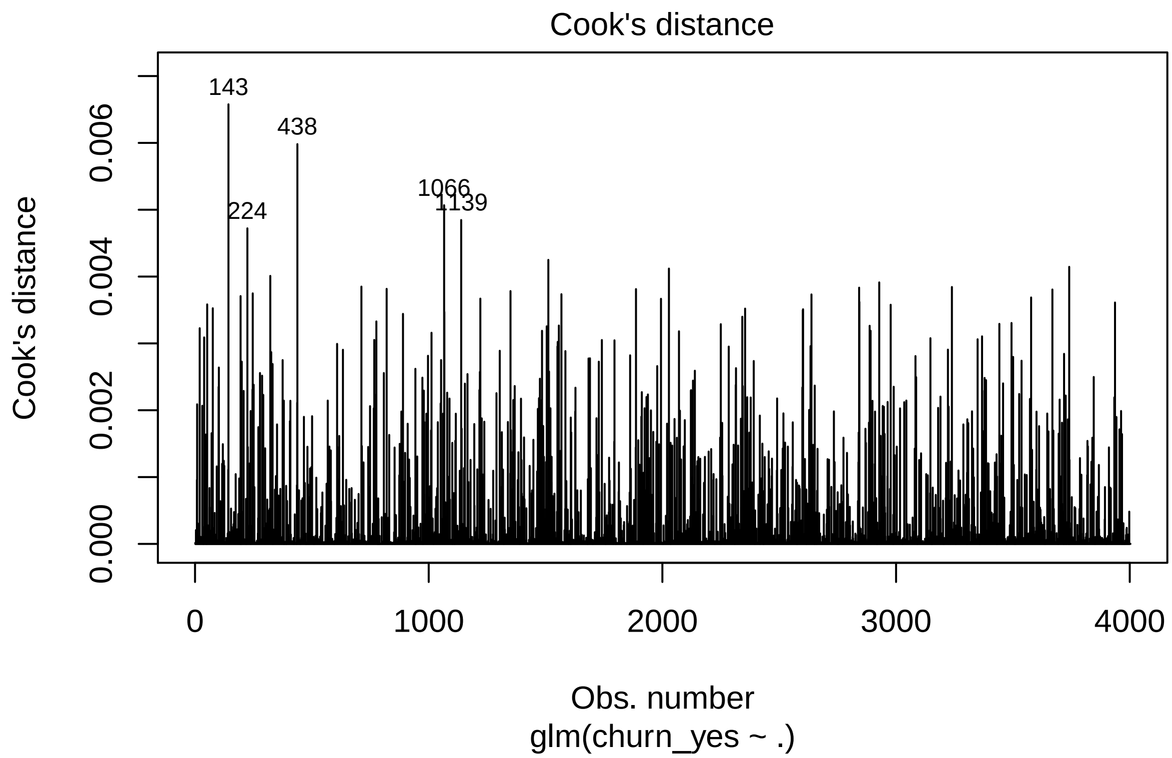
```
## $model1
##  McFadden
## 0.2611485
##
## $model3
##  McFadden
## 0.4753362
```

```r
model1_data <- augment(model1) %>%
  mutate(index = 1:n())

model3_data <- augment(model1) %>%
  mutate(index = 1:n())

model1_data %>%  # Used to estimate the influence of a data point when performing a least-squares regre
  filter(abs(.std.resid) > 3)
```

```
## # A tibble: 4 x 29
##   churn_yes state acct_length num_vm_mess tot_day_min tot_day_calls tot_day_chg
##   <fct>     <fct>       <dbl>       <dbl>       <dbl>         <dbl>       <dbl>
## 1 1         SD        -0.0651      -0.576      -3.34         -5.08       -3.34
## 2 1         SD        -0.241       -0.576      -0.506         1.17       -0.506
## 3 1         WY         1.47        -0.576      -0.239        -1.63       -0.239
## 4 1         IN        -0.316       -0.576      -1.81         -0.306      -1.81
## # ... with 22 more variables: tot_eve_min <dbl>, tot_eve_calls <dbl>,
## #   total_eve_charge <dbl>, tot_night_min <dbl>, tot_night_calls <dbl>,
## #   tot_night_chg <dbl>, tot_intl_min <dbl>, tot_intl_calls <dbl>,
```

```
## #   tot_intl_chg <dbl>, num_cust_serv_calls <dbl>, area_code_408 <fct>,
## #   area_code_415 <fct>, area_code_510 <fct>, intl_plan_yes <fct>,
## #   vm_plan_yes <fct>, .fitted <dbl>, .resid <dbl>, .std.resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, index <int>
```

```r
plot(model1, which = 4, id.n = 5)
```




```r
model1_data %>%
  top_n(5, .cooksd)
```

```
## # A tibble: 5 x 29
##   churn_yes state acct_length num_vm_mess tot_day_min tot_day_calls tot_day_chg
##   <fct>     <fct>       <dbl>       <dbl>       <dbl>         <dbl>       <dbl>
## 1 1         AK         0.637       -0.576      -2.26         -0.306      -2.26
## 2 1         SD         1.09         2.95        0.176        -0.205       0.176
## 3 1         AK         0.261       -0.576      -0.139        -2.13       -0.139
## 4 1         SD        -0.0651      -0.576      -3.34         -5.08       -3.34
## 5 1         IL        -0.291        0.820      -1.25          0.709      -1.25
## # ... with 22 more variables: tot_eve_min <dbl>, tot_eve_calls <dbl>,
## #   total_eve_charge <dbl>, tot_night_min <dbl>, tot_night_calls <dbl>,
## #   tot_night_chg <dbl>, tot_intl_min <dbl>, tot_intl_calls <dbl>,
## #   tot_intl_chg <dbl>, num_cust_serv_calls <dbl>, area_code_408 <fct>,
## #   area_code_415 <fct>, area_code_510 <fct>, intl_plan_yes <fct>,
## #   vm_plan_yes <fct>, .fitted <dbl>, .resid <dbl>, .std.resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, index <int>
```

```r
# Now checking how well the models perform on the validation set.
valid <- predict(norm, valid)

test_m1 <- predict(model1, newdata = valid, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```r
test_m3 <- predict(model3, newdata = valid, type = "response")



list(
  model1 = table(valid$churn_yes, test_m1 > 0.5) %>%
    prop.table() %>%
    round(3),
  model3 = table(valid$churn_yes, test_m3 > 0.5) %>%
    prop.table() %>%
    round(3)
)
```

```
## $model1
##
##     FALSE  TRUE
##   0 0.837 0.022
##   1 0.112 0.029
##
## $model3
##
##     FALSE  TRUE
##   0 0.843 0.016
##   1 0.065 0.076
```

```r
table(valid$churn_yes, test_m1 > .5)
```

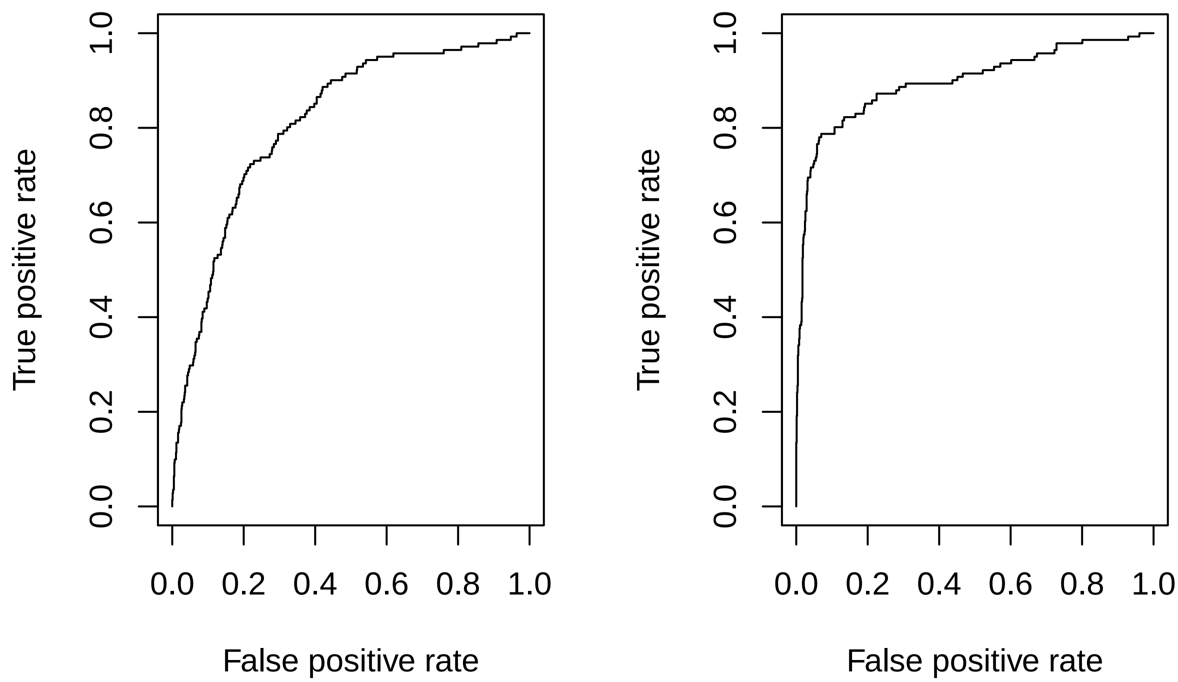
```
##
##     FALSE TRUE
##   0   836   22
##   1   112   29
```

```r
table(valid$churn_yes, test_m3 > .5)
```

```
##
##     FALSE TRUE
##   0   842   16
##   1    65   76
```

ROC AND AUC

```r
library(ROCR)

par(mfrow=c(1, 2))

prediction(test_m1, valid$churn_yes) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot()

prediction(test_m3, valid$churn_yes) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot()
```




```r
# model 2 AUC
prediction(test_m1, valid$churn_yes) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.8117509
```

```r
# model 2 AUC
prediction(test_m3, valid$churn_yes) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.8972788
```

‘