Can I Borrow Some Money?

Mike Rhinehart

June 5, 2018

Introduction

The is a dataset from a expired Kaggle competition where it requires participants to improve on the state of the art in credit scoring, by predicting the probability that somebody will experience financial distress in the next two years.

Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit. Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. The goal of this competition is to build a model that borrowers can use to help make the best financial decisions.

Data Preparation

First, we begin reading in the dataset. Then we need to look at the structure our data to view the data types before performing exploratory analysis. The five number summary gives us an idea of the value range of the predictor variables and identifying the target variable or dependent variable.

```r
credit = read.csv("cs-train.csv", header = TRUE) # reading in the dataset
str(credit) # View the structure of the data to see the data types
```

```
## 'data.frame':    150000 obs. of  12 variables:
##  $ X                                   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SeriousDlqin2yrs                    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ RevolvingUtilizationOfUnsecuredLines: num  0.766 0.957 0.658 0.234 0.907 ...
##  $ age                                 : int  45 40 38 30 49 74 57 39 27 57 ...
##  $ NumberOfTime30.59DaysPastDueNotWorse: int  2 0 1 0 1 0 0 0 0 0 ...
##  $ DebtRatio                           : num  0.803 0.1219 0.0851 0.036 0.0249 ...
##  $ MonthlyIncome                       : int  9120 2600 3042 3300 63588 3500 NA 3500 NA 23684 ...
##  $ NumberOfOpenCreditLinesAndLoans     : int  13 4 2 5 7 3 8 8 2 9 ...
##  $ NumberOfTimes90DaysLate             : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ NumberRealEstateLoansOrLines        : int  6 0 0 0 1 1 3 0 0 4 ...
##  $ NumberOfTime60.89DaysPastDueNotWorse: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NumberOfDependents                  : int  2 1 0 0 0 1 0 0 NA 2 ...
```

```r
summary(credit) # five number summary
```

```
##        X          SeriousDlqin2yrs  RevolvingUtilizationOfUnsecuredLines
##  Min.   :     1   Min.   :0.00000   Min.   :    0.00
##  1st Qu.: 37501   1st Qu.:0.00000   1st Qu.:    0.03
##  Median : 75001   Median :0.00000   Median :    0.15
##  Mean   : 75001   Mean   :0.06684   Mean   :    6.05
##  3rd Qu.:112500   3rd Qu.:0.00000   3rd Qu.:    0.56
##  Max.   :150000   Max.   :1.00000   Max.   :50708.00
##
##       age        NumberOfTime30.59DaysPastDueNotWorse   DebtRatio
##  Min.   :  0.0   Min.   : 0.000                       Min.   :     0.0
##  1st Qu.: 41.0   1st Qu.: 0.000                       1st Qu.:     0.2
```

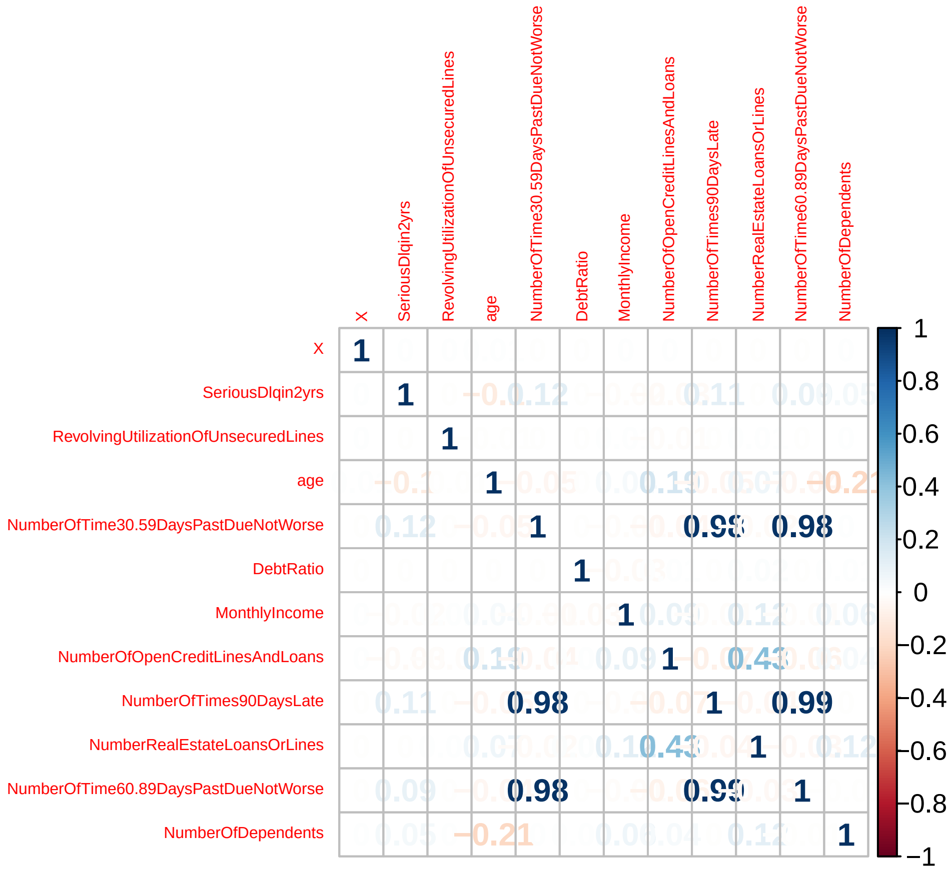
```
##  Median : 52.0   Median : 0.000                        Median :     0.4
##  Mean   : 52.3   Mean   : 0.421                        Mean   :   353.0
##  3rd Qu.: 63.0   3rd Qu.: 0.000                        3rd Qu.:     0.9
##  Max.   :109.0   Max.   :98.000                        Max.   :329664.0
##
##  MonthlyIncome     NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
##  Min.   :      0   Min.   : 0.000                  Min.   : 0.000
##  1st Qu.:   3400   1st Qu.: 5.000                  1st Qu.: 0.000
##  Median :   5400   Median : 8.000                  Median : 0.000
##  Mean   :   6670   Mean   : 8.453                  Mean   : 0.266
##  3rd Qu.:   8249   3rd Qu.:11.000                  3rd Qu.: 0.000
##  Max.   :3008750   Max.   :58.000                  Max.   :98.000
##  NA's   :29731
##  NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 1.000               Median : 0.0000
##  Mean   : 1.018               Mean   : 0.2404
##  3rd Qu.: 2.000               3rd Qu.: 0.0000
##  Max.   :54.000               Max.   :98.0000
##
##  NumberOfDependents
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 0.757
##  3rd Qu.: 1.000
##  Max.   :20.000
##  NA's   :3924
```

Here we look at the correlation matrix to measure the correlations between the predictors and the outcome variable. Noticed, there are cases of multi-collinearity between the NumberOfTime30.59DaysPastDueNotWorse, NumberOfTime60.89DaysPastDueNotWorse, and NumberOfTime60.89DaysPastDueNotWorse. This means we would have to drop two predictors and leave only one. Moreover, multicollinearity makes it tedious to assess the relative importance of the independent variables in explaining the variation caused by the dependent variable. Since they are closer to 1 on a scale to -1 to 1, we would keep the NumberOfTime30.59DaysPastDueNotWorse to avoid increases the standard errors of the coefficients.

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
credit_miss <- na.omit(credit) # temporarily remove NA's before correlation plot
corrplot(cor(credit_miss), method = "number", tl.cex = 0.5) # correlation matrix
```




```
credit_miss$SeriousDlqin2yrs[credit_miss$SeriousDlqin2yrs == 0] <- "No"
credit_miss$SeriousDlqin2yrs[credit_miss$SeriousDlqin2yrs == 1] <- "Yes"
credit_miss$SeriousDlqin2yrs <- as.factor(credit_miss$SeriousDlqin2yrs) #change the outcome variable to
```

We observed there are many missing values in the data and will deal those later. Next, we combined the three defaulted fields and removed two out of the three multi-collinearity predictor variables.

```
sum(is.na(credit_miss)) # count the number of NA's in the data
```

```
## [1] 0
```

```
credit_sub <- subset(credit_miss, select = -c(NumberOfTimes90DaysLate,NumberOfTime60.89DaysPastDueNotWo
str(credit_sub) # view to make sure both fields are removed
```

```
## 'data.frame':    120269 obs. of  10 variables:
##  $ X                                   : int  1 2 3 4 5 6 8 10 11 12 ...
##  $ SeriousDlqin2yrs                    : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ RevolvingUtilizationOfUnsecuredLines: num  0.766 0.957 0.658 0.234 0.907 ...
##  $ age                                 : int  45 40 38 30 49 74 39 57 30 51 ...
##  $ NumberOfTime30.59DaysPastDueNotWorse: int  2 0 1 0 1 0 0 0 0 0 ...
##  $ DebtRatio                           : num  0.803 0.1219 0.0851 0.036 0.0249 ...
##  $ MonthlyIncome                       : int  9120 2600 3042 3300 63588 3500 3500 23684 2500 6501 ..
##  $ NumberOfOpenCreditLinesAndLoans     : int  13 4 2 5 7 3 8 9 5 7 ...
##  $ NumberRealEstateLoansOrLines        : int  6 0 0 0 1 1 0 4 0 2 ...
##  $ NumberOfDependents                  : int  2 1 0 0 0 1 0 2 0 2 ...
```

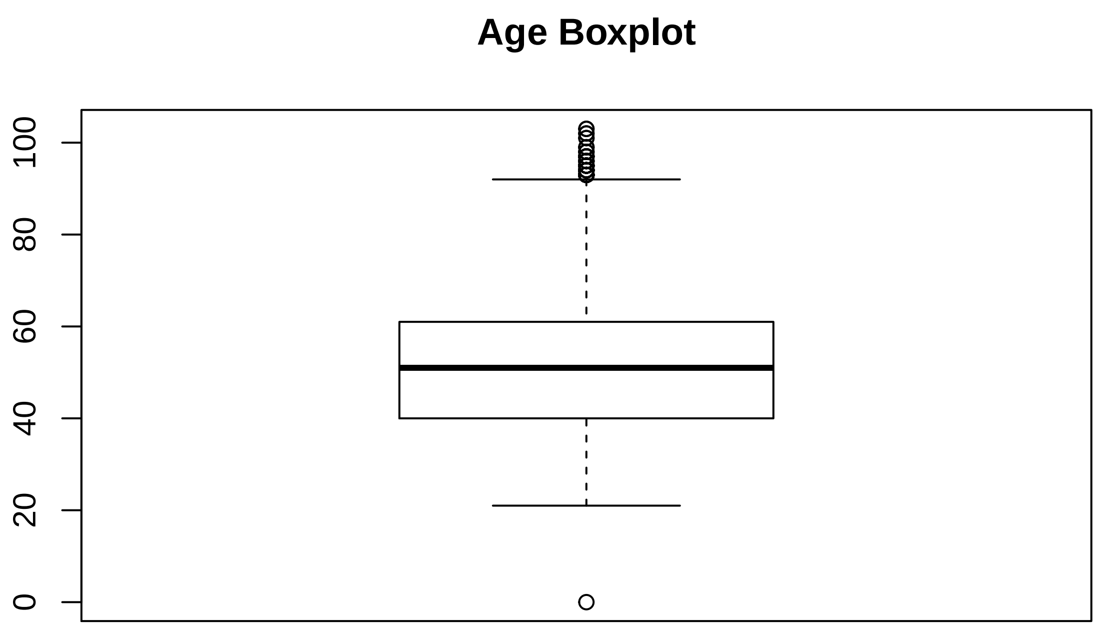
Data Cleaning/Exploratory Data Analysis

First, we noticed RevolvingUtilizationOfUnsecuredLines field had some individuals with higher credit utilization greater than 100 percent. These are cases we will need to remove because no one can go over their max usage of credit. So we will replace those values greater than 100 percent with NA's.

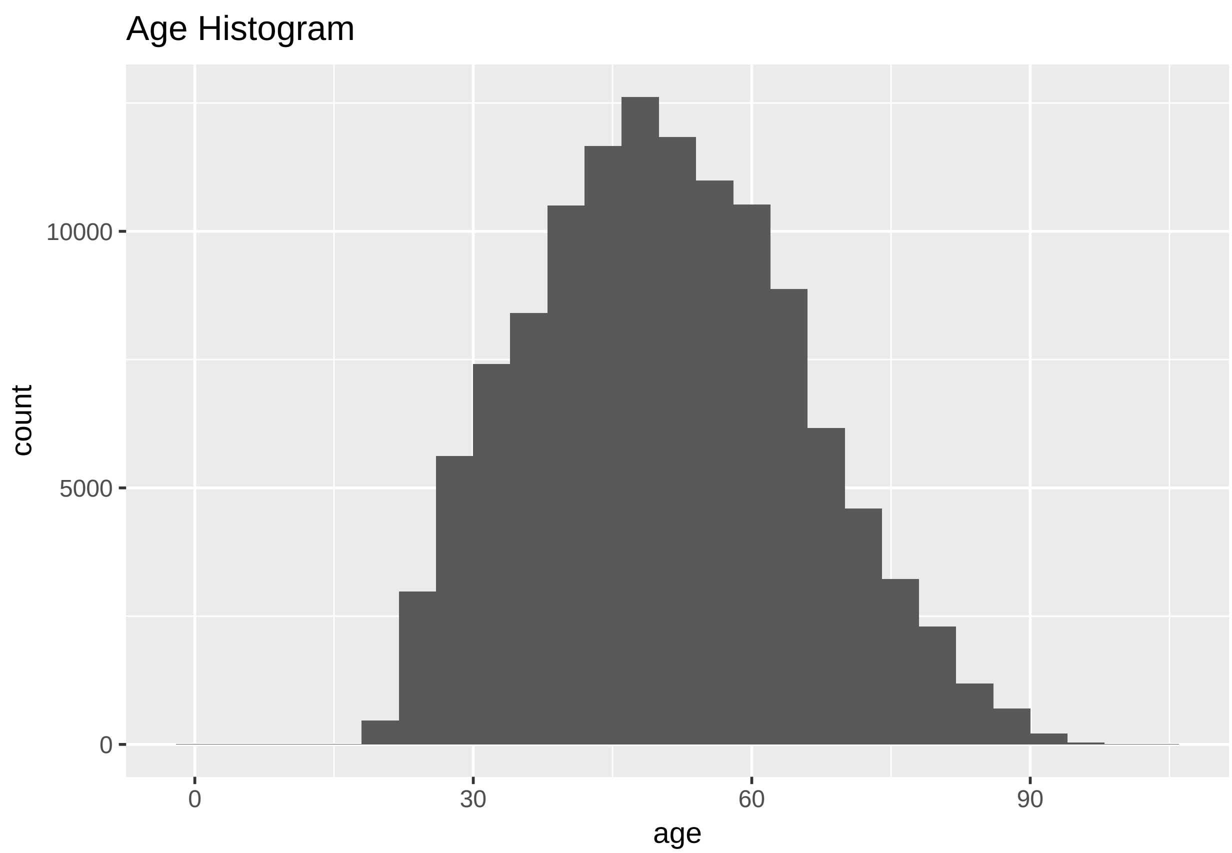
In the boxplot for Age, there are several outliers in the upper whisker and one in the lower whisker that needs to be removed. So we will replace those outliers with NA's as well.

In the histogram, the skewness of Age, looks normally distributed.

```
library(ggplot2)
credit_sub$RevolvingUtilizationOfUnsecuredLines[credit_sub$RevolvingUtilizationOfUnsecuredLines > 1] <-
boxplot(credit_sub$age, main = "Age Boxplot")
```




```
ggplot(credit_sub, aes(age)) + geom_histogram(binwidth = 4) + labs(title="Age Histogram")
```

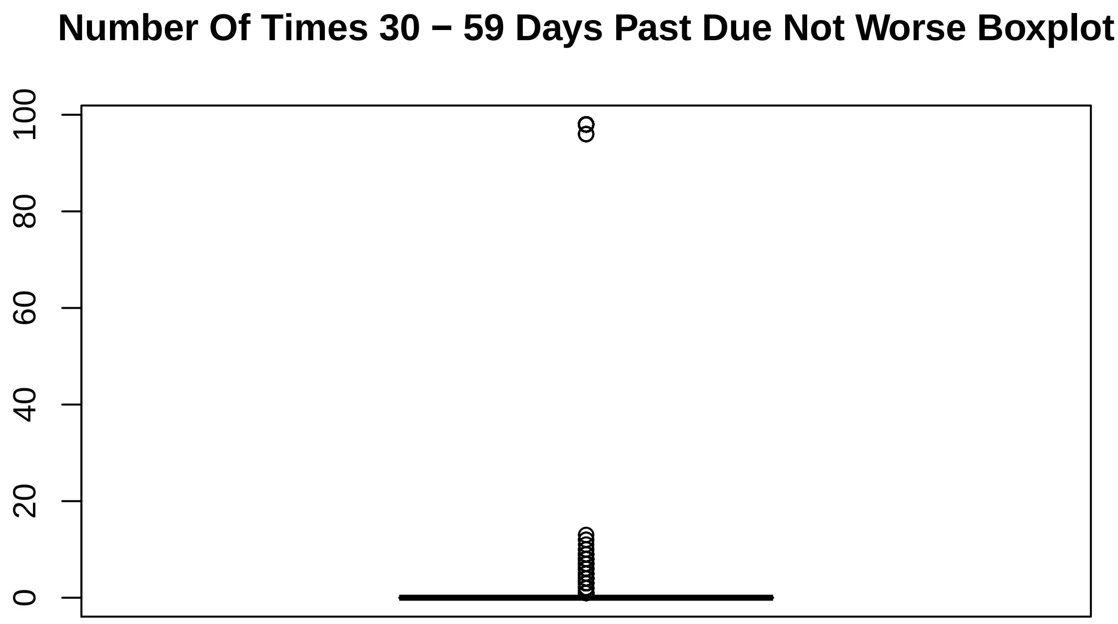



```
credit_sub$age[credit_sub$age < 21] <- NA
credit_sub$age[credit_sub$age > 90] <- NA
```

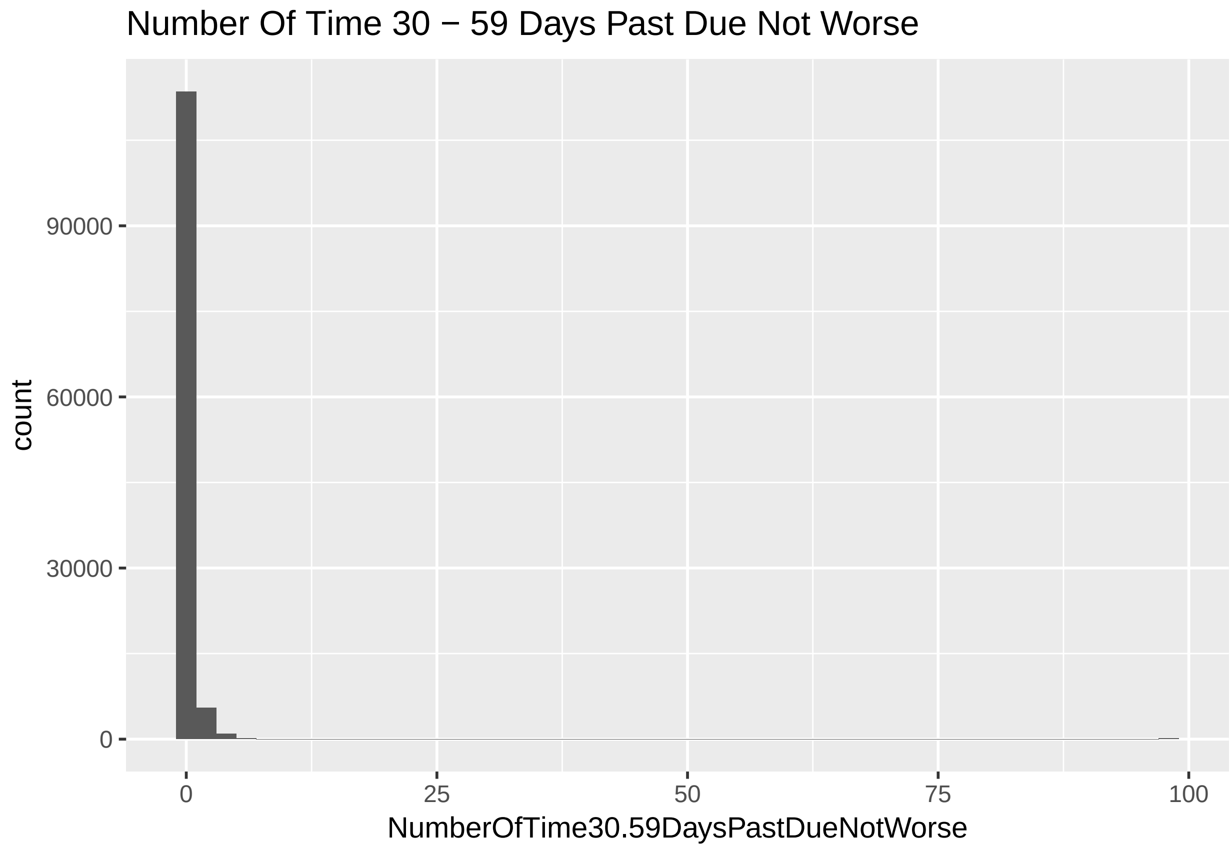
In the boxplot of NumberOfTime30.59DaysPastDueNotWorse, it's really difficult to find if there are cases of outliers, other than the two in the upper extreme, since there are no quartile boxes or whiskers to interpret. So, we will plot a histogram to see if we can get a better look at possible outliers. Noticed, the observations on the histogram displays values only less than approximately to 10.We can assume these values are outliers and replace them with NA.

The DebtRatio field had some individuals with credit usage greater than 100 percent. These are cases we will need to remove since no one can borrow money than their max credit given to them. So we will replace those values greater than 100 percent with NA's.

```
boxplot(credit_sub$NumberOfTime30.59DaysPastDueNotWorse, main = "Number Of Times 30 - 59 Days Past Due N
```




```
ggplot(credit_sub, aes(NumberOfTime30.59DaysPastDueNotWorse)) + geom_histogram(binwidth = 2) + labs(titl
```

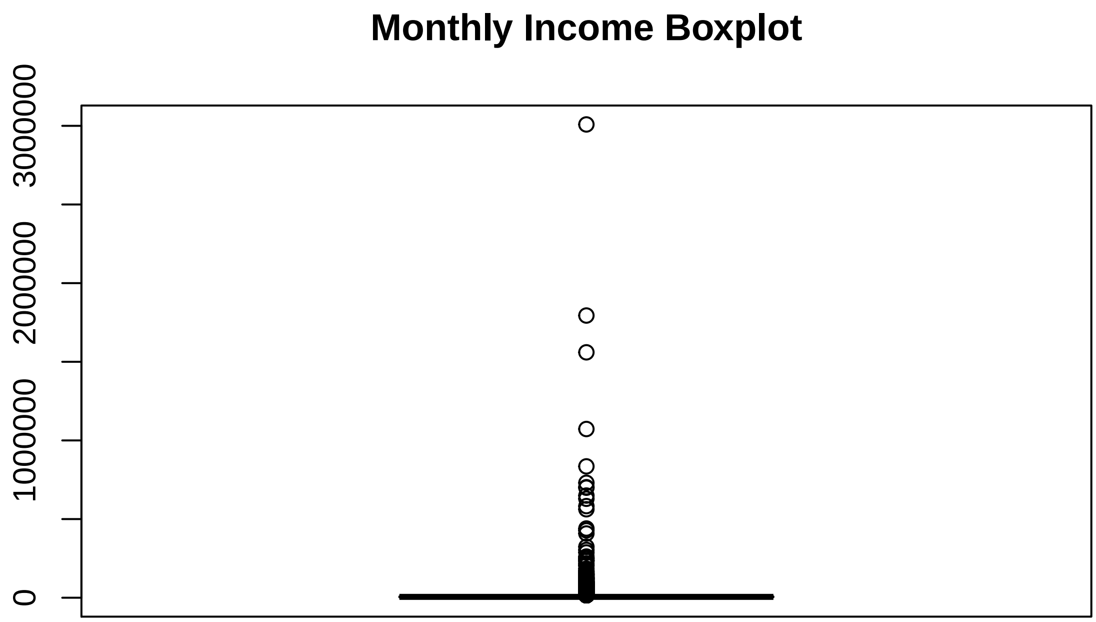



```
credit_sub$NumberOfTime30.59DaysPastDueNotWorse[credit_sub$NumberOfTime30.59DaysPastDueNotWorse > 10]
credit_sub$DebtRatio[credit_sub$DebtRatio > 1] <- NA
```

In the boxplot of MonthlyIncome, it's really difficult to find if there are cases of outliers, other than the one in the upper extreme, since there are no quartile boxes or whiskers to interpret. So, we will plot a histogram to see if we can get a better look of possible outliers. Noticed, the spread of the data is very skewed to the right and does not take shape of a normal distribution. Moroever, we are going to replace the Monthly Income greater 14000 to NA, so we can reduce inaccurate classifications errorsbefore using several machine learning techniques later.

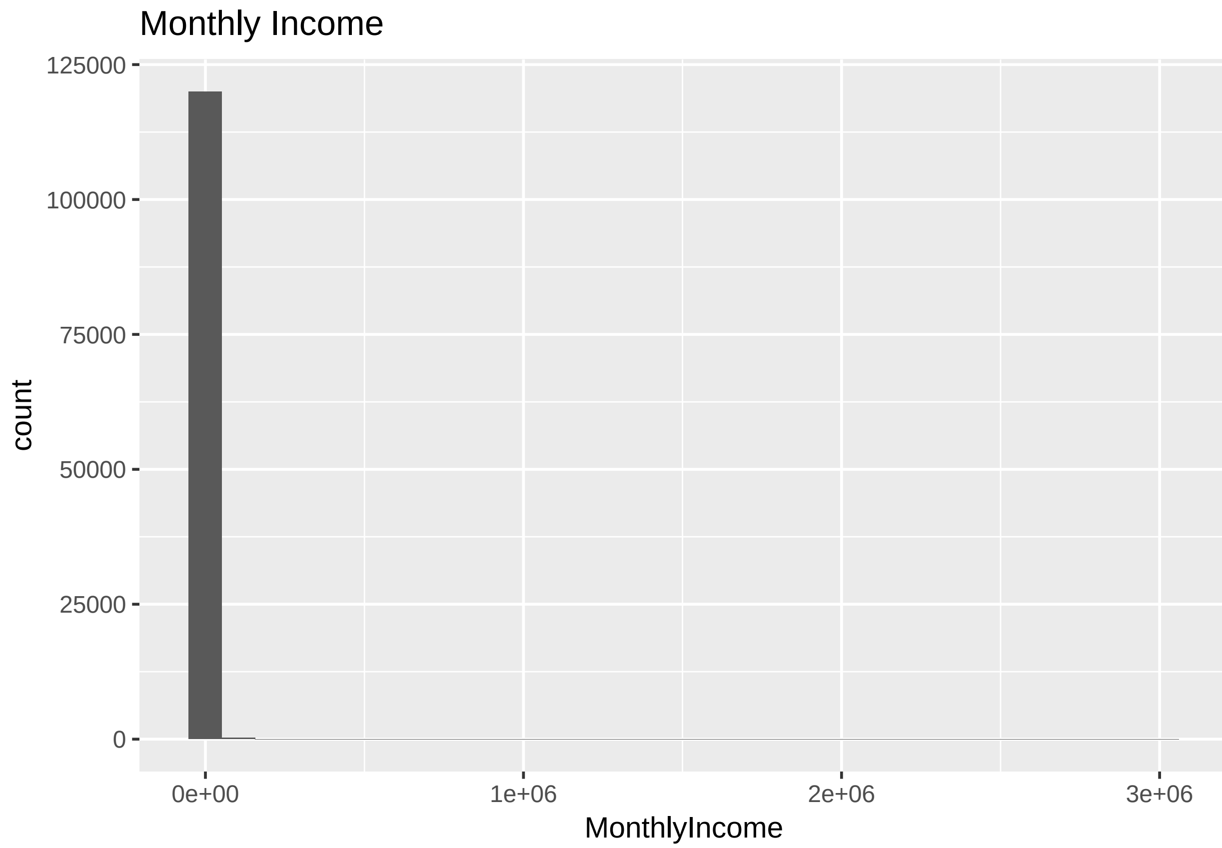
We plotted the Monthly Boxplot again to detect cases of outliers and they are no longer in our data.

```
boxplot(credit_sub$MonthlyIncome, main = "Monthly Income Boxplot")
```



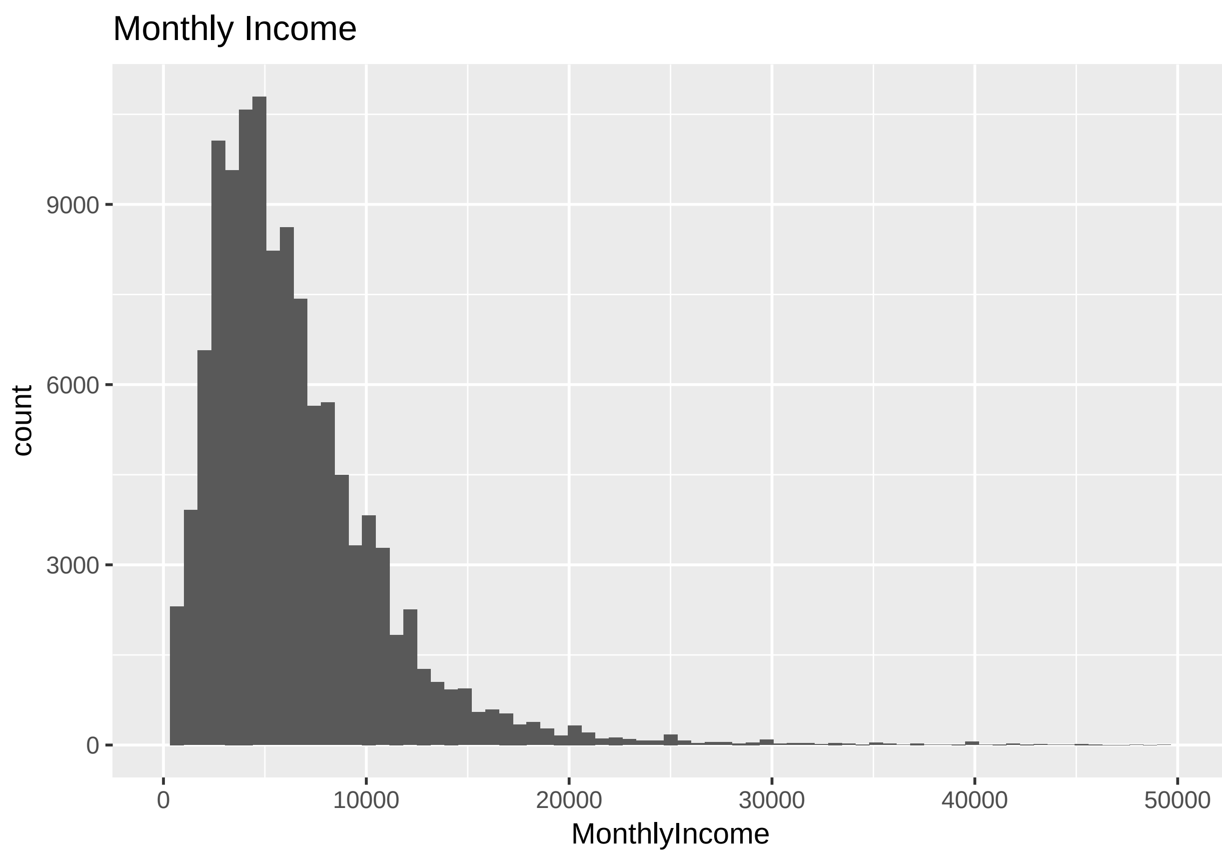

```r
ggplot(credit_sub, aes(MonthlyIncome)) + geom_histogram() + labs(title="Monthly Income")
```

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

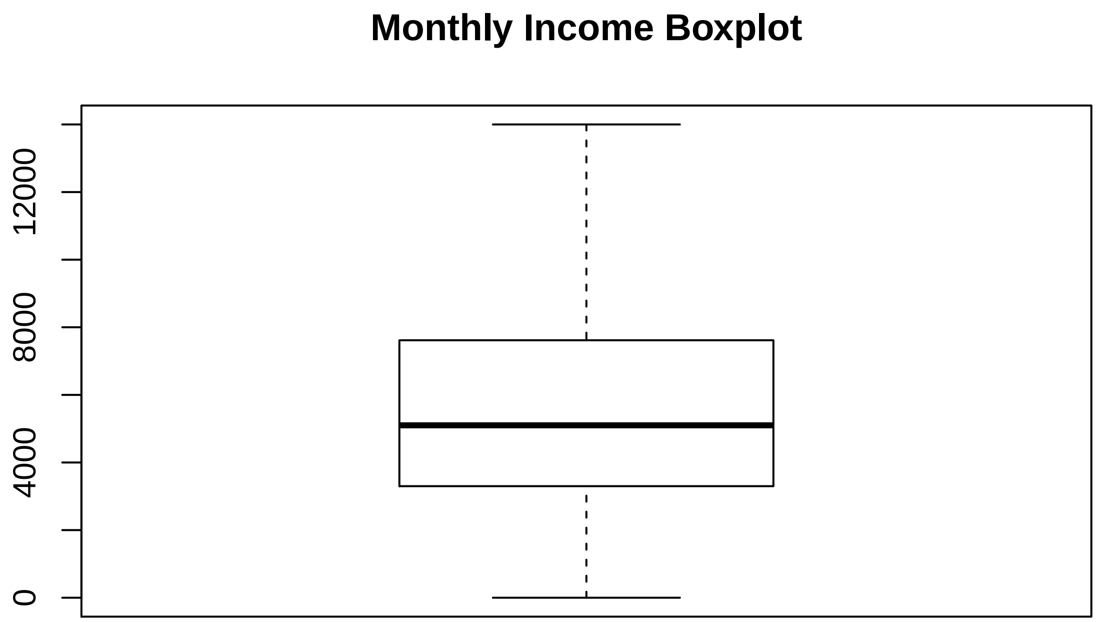



```
ggplot(credit_sub, aes(MonthlyIncome)) + geom_histogram(bins = 75) + labs(title="Monthly Income")+ xlim
```

```
## Warning: Removed 301 rows containing non-finite values (stat_bin).
```

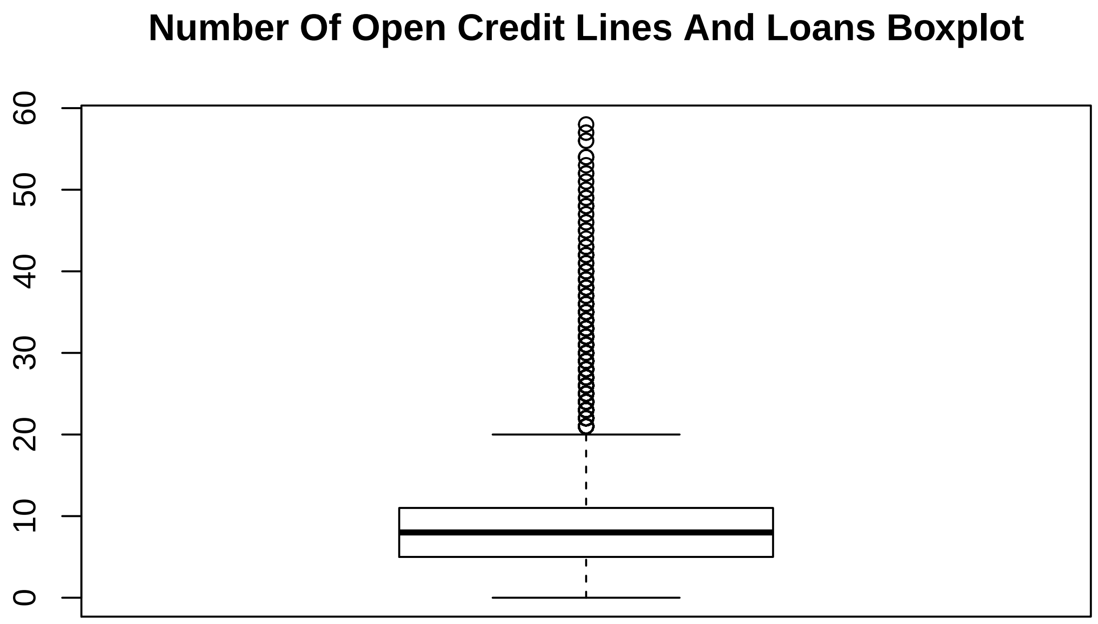



```r
credit_sub$MonthlyIncome[as.integer(credit_sub$MonthlyIncome) > 14000] <- NA
boxplot(credit_sub$MonthlyIncome, main = "Monthly Income Boxplot")
```

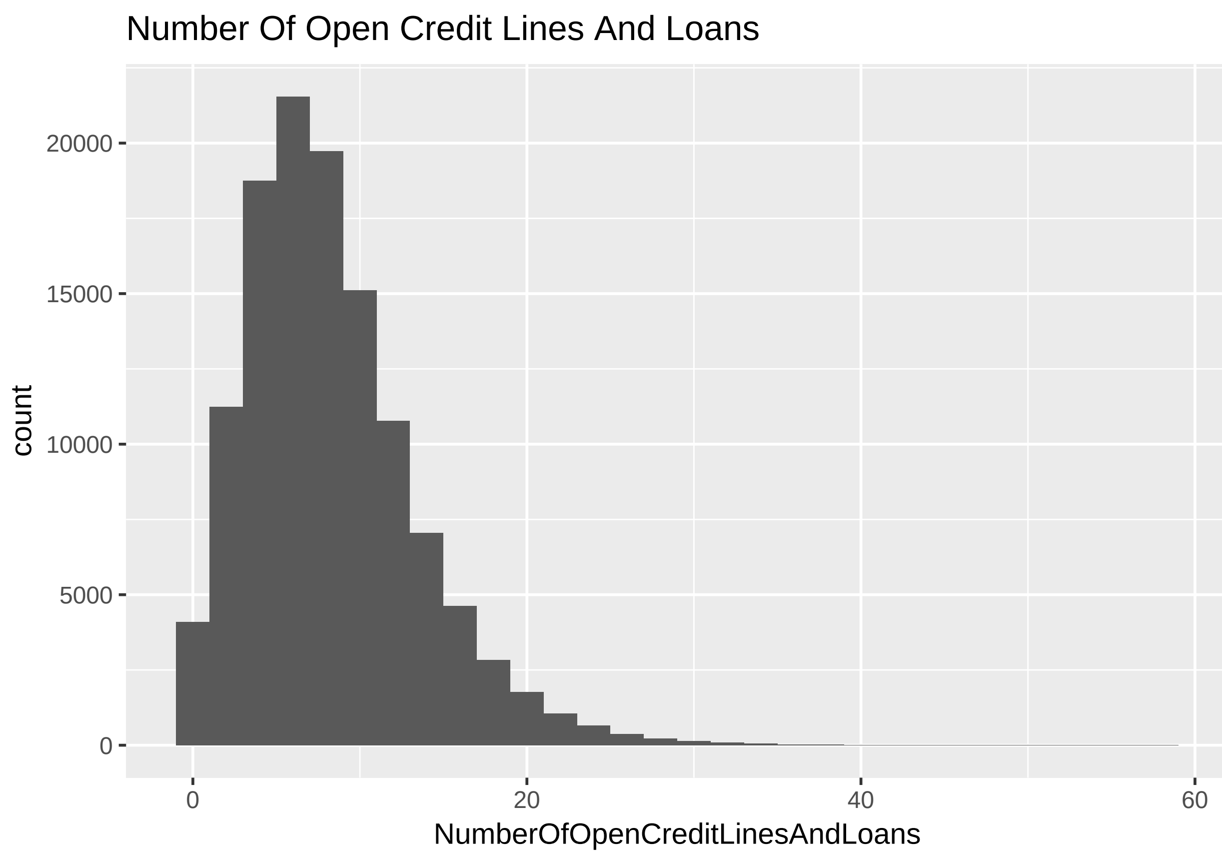


In the Number of Open Credit Lines And Loans boxplot, there are serveral outliers in the upper whisker and the data looks slightly skewed to the right. Let's take a look at the histogram to be sure they are outliers and data is slighltly skewed to the right. The outliers in the boxplot could have caused it to not take the shape of a normal distribution, so lets remove them.

```
boxplot(credit_sub$NumberOfOpenCreditLinesAndLoans, main = "Number Of Open Credit Lines And Loans Boxpl
```



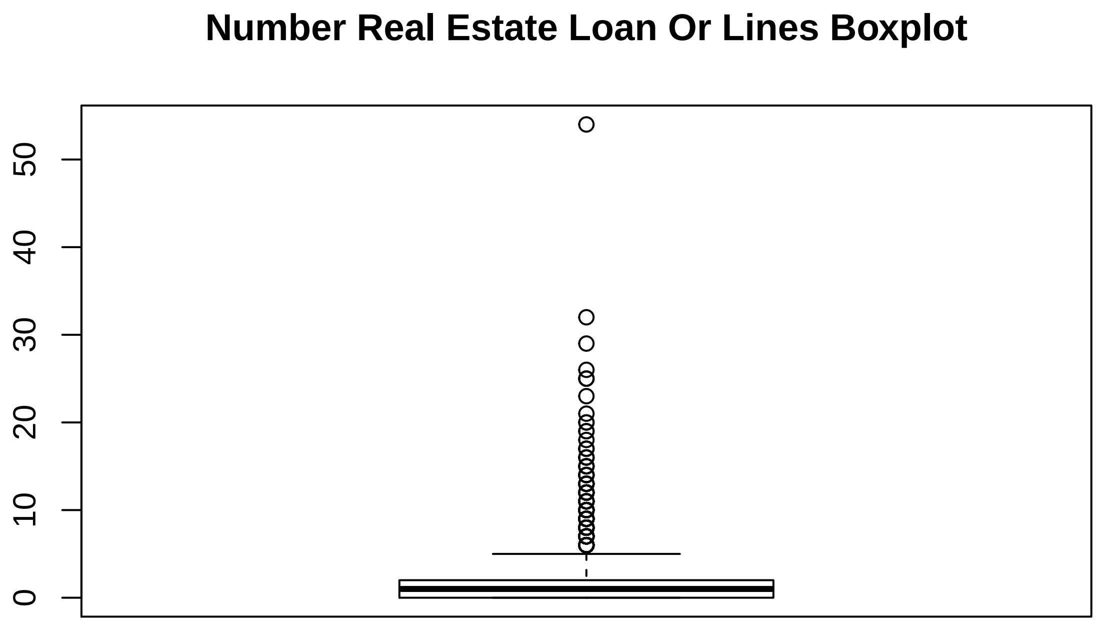

```
ggplot(credit_sub, aes(NumberOfOpenCreditLinesAndLoans)) + geom_histogram(bins = 30) + labs(title="Numbe
```



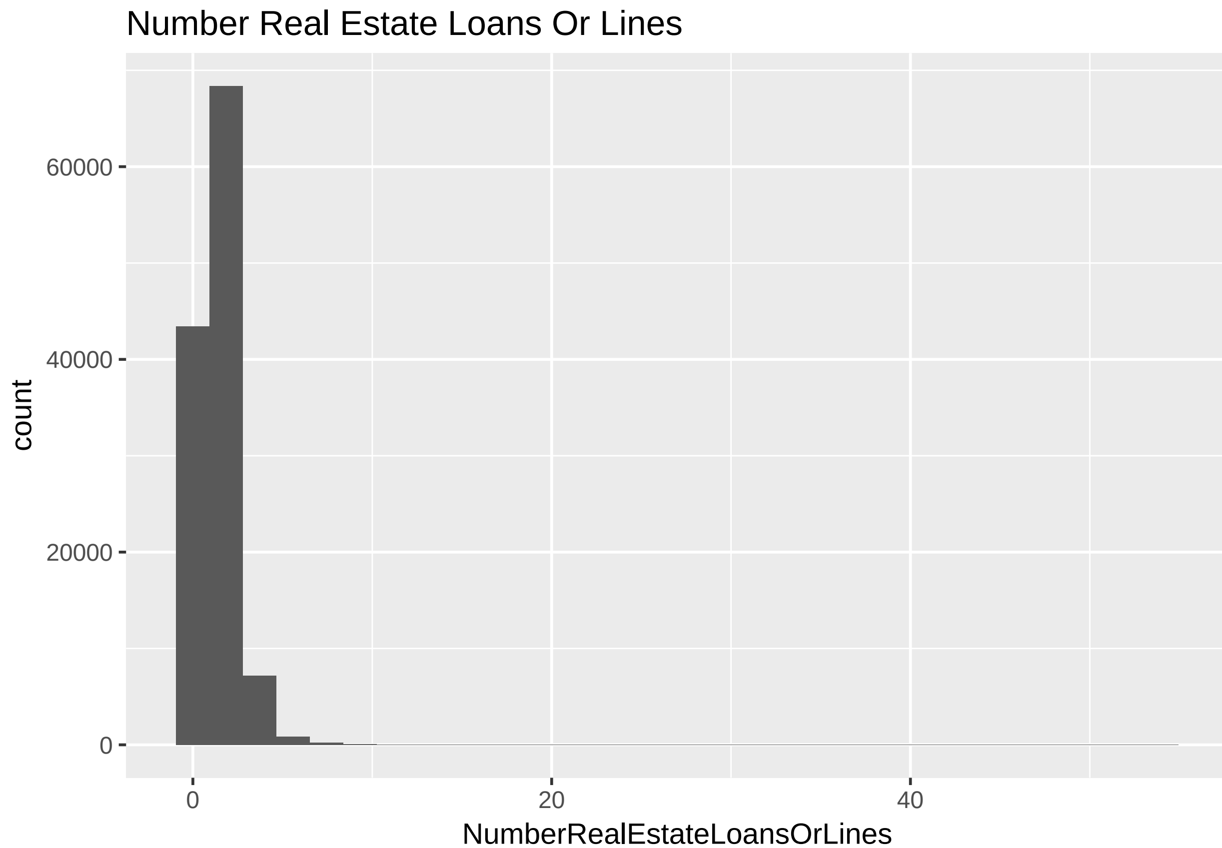

```
credit_sub$NumberOfOpenCreditLinesAndLoans[credit_sub$NumberOfOpenCreditLinesAndLoans > 20] <- NA
```

In the Number Real Estate Loans Or Lines boxplot, there are serveral outliers in the upper whisker and only one at the very top. Let's take a look at the histogram to be sure they are outliers. Noticed, the data is slighltly skewed to the right and does not have a bell-shaped curve. Moreover, the outliers in the boxplot could have caused it to not take the shape of a normal distribution, so lets replace those values greater than 7 them with NA.

```
boxplot(credit_sub$NumberRealEstateLoansOrLines, main = "Number Real Estate Loan Or Lines Boxplot")
```



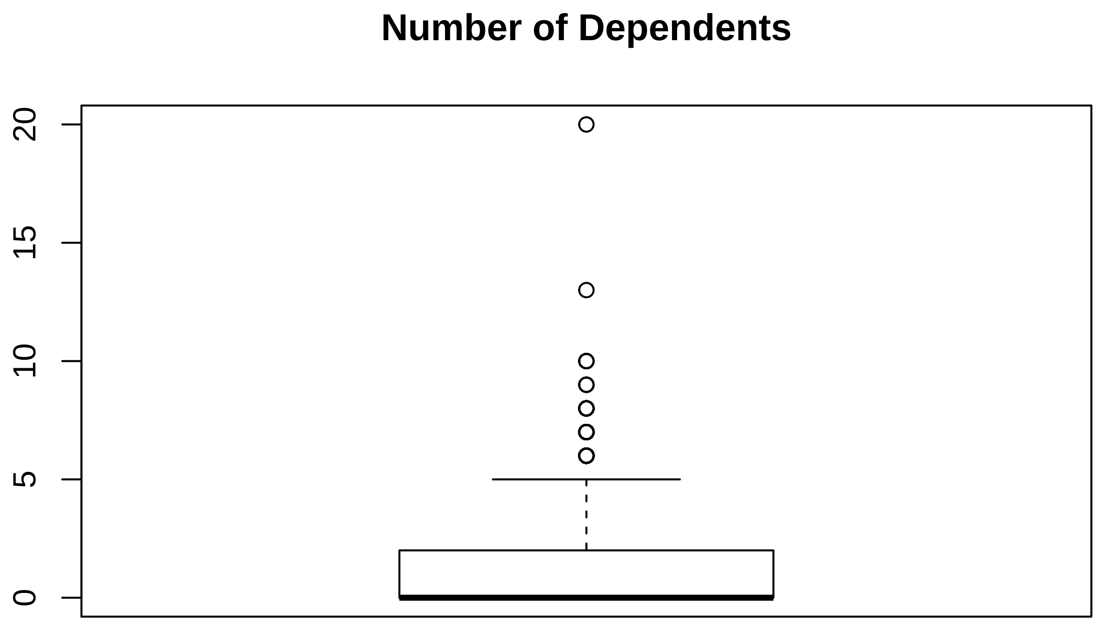

```
ggplot(credit_sub, aes(NumberRealEstateLoansOrLines)) + geom_histogram(bins = 30) + labs(title="Number R
```



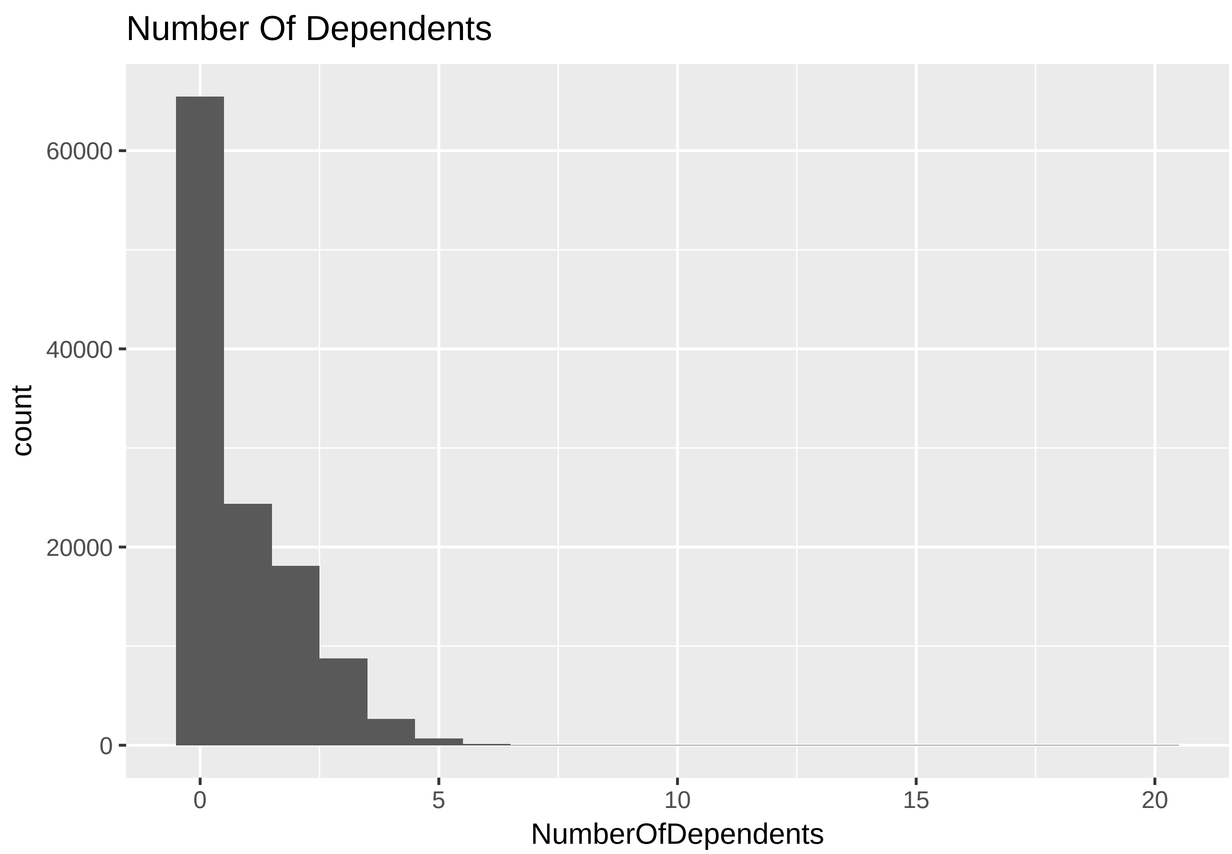

```
credit_sub$NumberRealEstateLoansOrLines[credit_sub$NumberRealEstateLoansOrLines > 7] <- NA
```

In the Number of Dependents boxplot, there are serveral outliers in the upper whisker. Let's take a look at the histogram to be sure they are outliers. Since it doesn't display the values greater than 5, lets assume these are outliers and replace them with NA.

```
boxplot(credit_sub$NumberOfDependents, main = "Number of Dependents")
```




```
ggplot(credit_sub, aes(NumberOfDependents)) + geom_histogram(binwidth = 1) + labs(title= "Number Of Depe
```




```
credit_sub$NumberOfDependents[credit_sub$NumberOfDependents > 5] <- NA
```

From the cleaning above, let's see how many NA's our data contain. There are quite a bit of NA's in our data, so lets remove them before performing EDA.

```
sum(is.na(credit_sub))
```

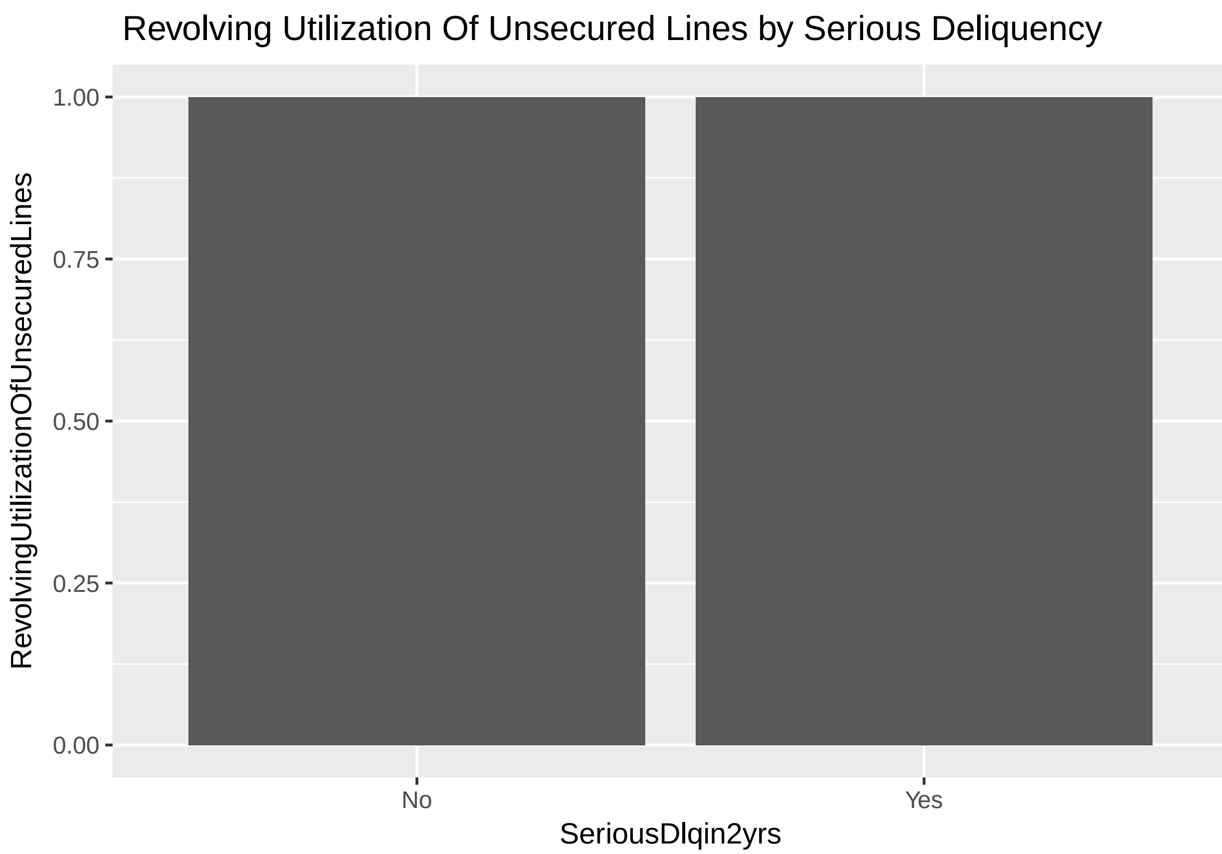
```
## [1] 21070
```

```
credit_clean <- na.omit(credit_sub)
attach(credit_clean)
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent according to Revolving Utilization Of Unsecured Lines.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(data=credit_clean, aes(SeriousDlqin2yrs,RevolvingUtilizationOfUnsecuredLines)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = " Revolving Utilization Of Unse
```



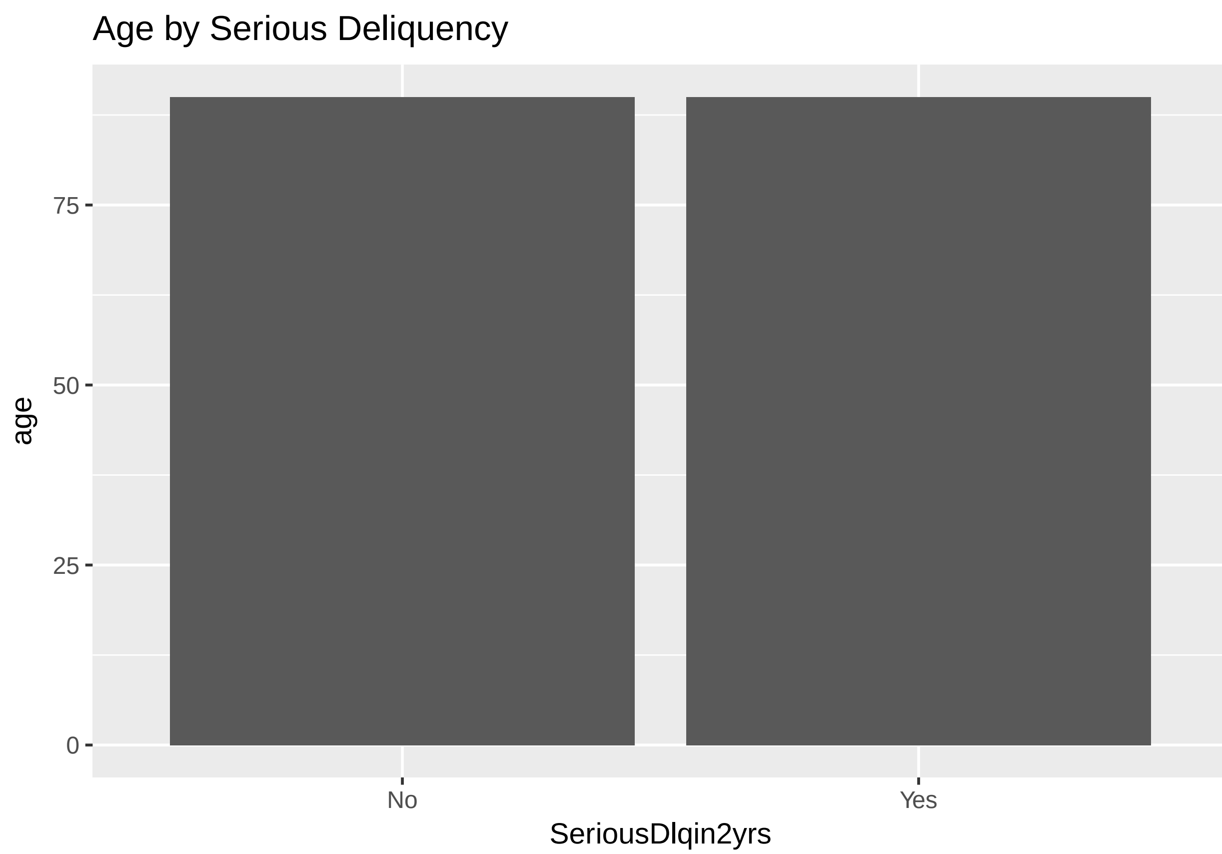

```
t.test(RevolvingUtilizationOfUnsecuredLines~SeriousDlqin2yrs, var.equal=FALSE) # testing the means of t
```

```
##
##  Welch Two Sample t-test
##
## data:  RevolvingUtilizationOfUnsecuredLines by SeriousDlqin2yrs
## t = -73.406, df = 6805.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3493288 -0.3311565
## sample estimates:
##  mean in group No mean in group Yes
##         0.2942551         0.6344977
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent according to Age.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(credit_clean, aes(SeriousDlqin2yrs,age)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "Age by Serious Deliquency")
```



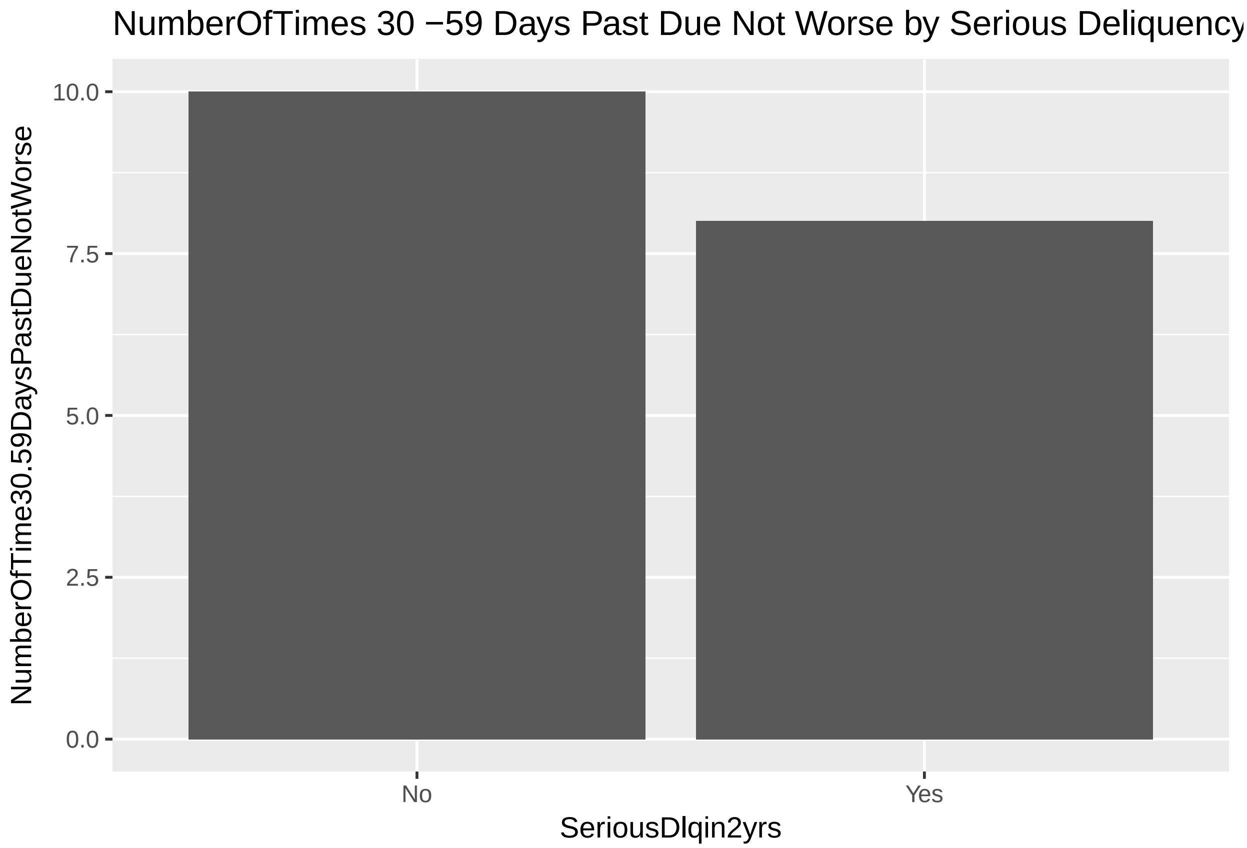

```r
t.test(age~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  age by SeriousDlqin2yrs
## t = 33.805, df = 7173.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  5.444296 6.114582
## sample estimates:
##  mean in group No mean in group Yes
##          51.48477          45.70533
```

From the barplot, there were more number of individuals were Seriously Deliquent than were not delinquent according to NumberOfTimes 30 -59 Days Past Due Not Worse.

According to the t-test, there is a significant difference within the mean of the groups.

```r
ggplot(credit_clean, aes(SeriousDlqin2yrs,NumberOfTime30.59DaysPastDueNotWorse)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "NumberOfTimes 30 -59 Days Past
```



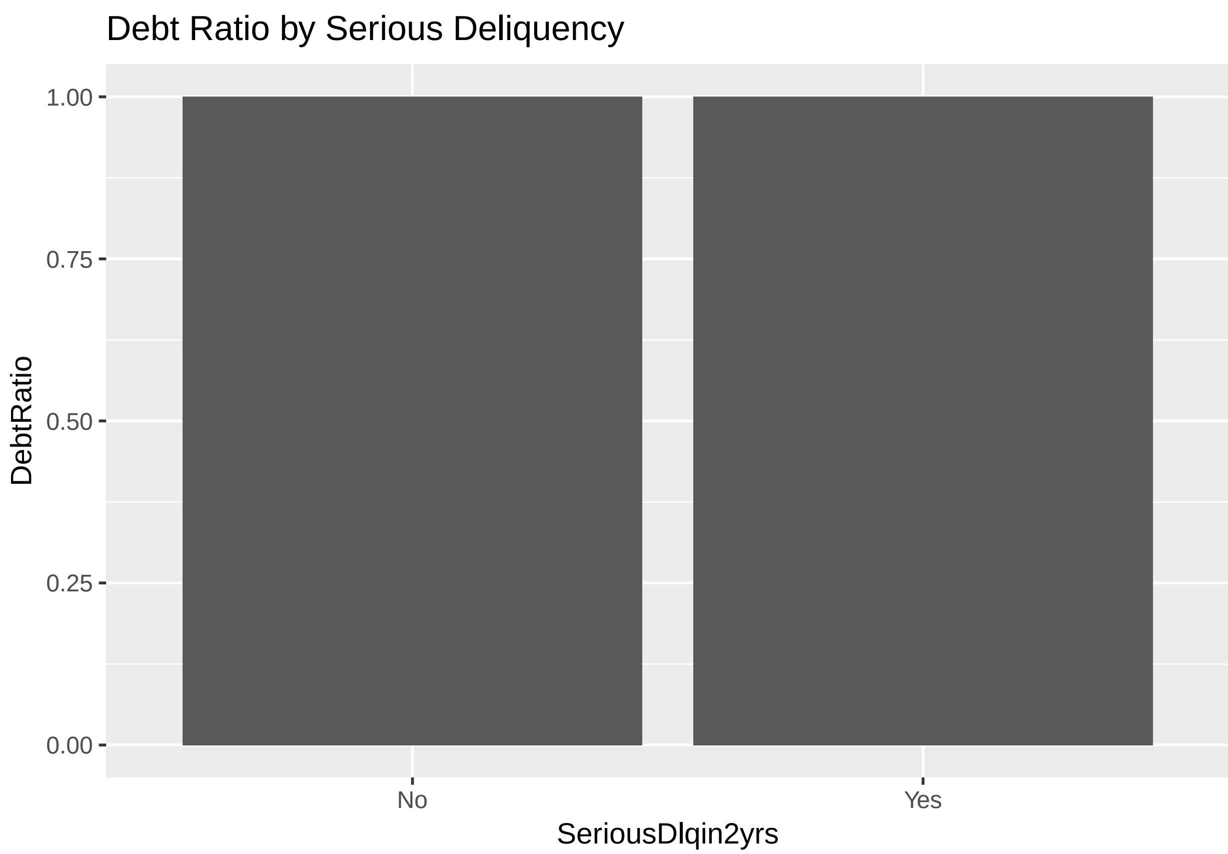

```
t.test(NumberOfTime30.59DaysPastDueNotWorse~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  NumberOfTime30.59DaysPastDueNotWorse by SeriousDlqin2yrs
## t = -42.352, df = 6279, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7218833 -0.6580127
## sample estimates:
##  mean in group No mean in group Yes
##         0.1940507         0.8839987
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent according to Debt Ratio.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(credit_clean, aes(SeriousDlqin2yrs,DebtRatio)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "Debt Ratio by Serious Deliquen
```



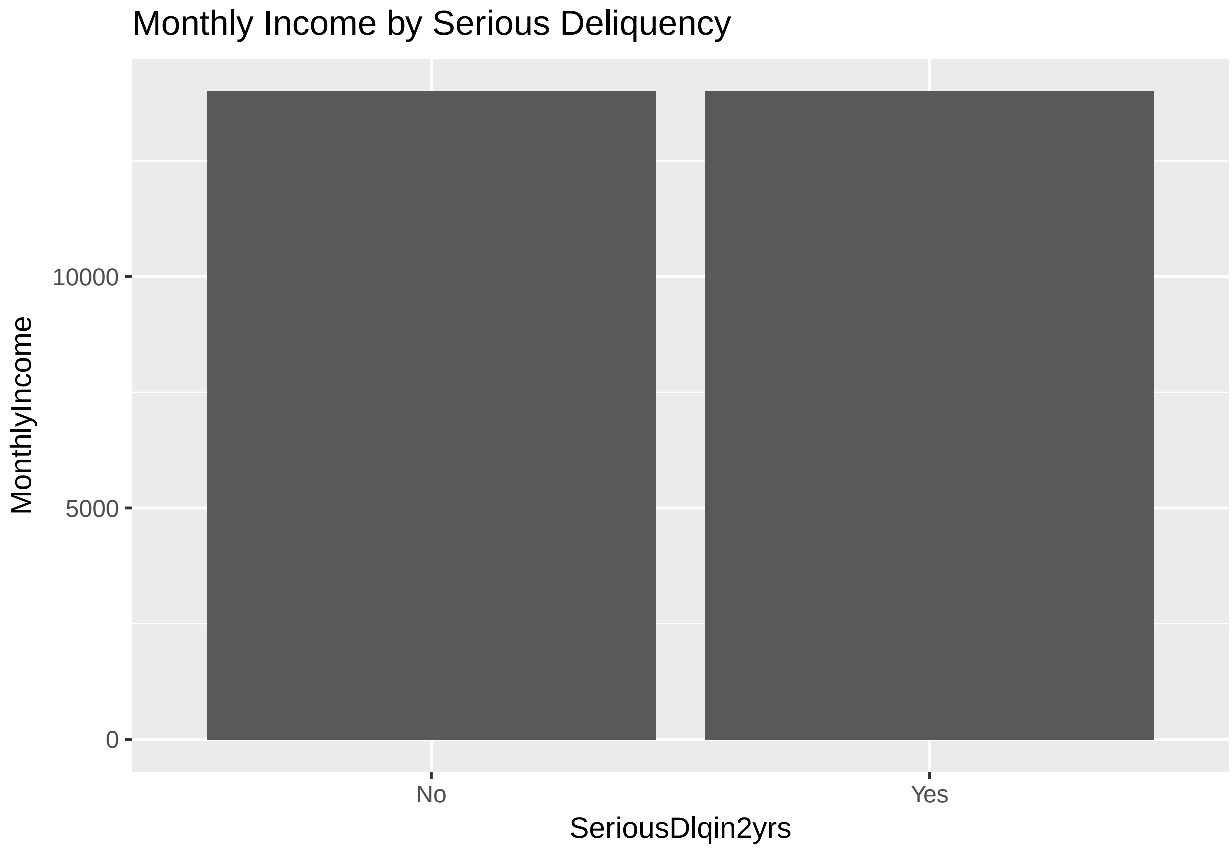

```
t.test(DebtRatio~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  DebtRatio by SeriousDlqin2yrs
## t = -16.357, df = 6766.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05934597 -0.04664392
## sample estimates:
##  mean in group No mean in group Yes
##         0.3037153         0.3567102
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent according to Monthly Income.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(credit_clean, aes(SeriousDlqin2yrs,MonthlyIncome)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "Monthly Income by Serious Delic
```



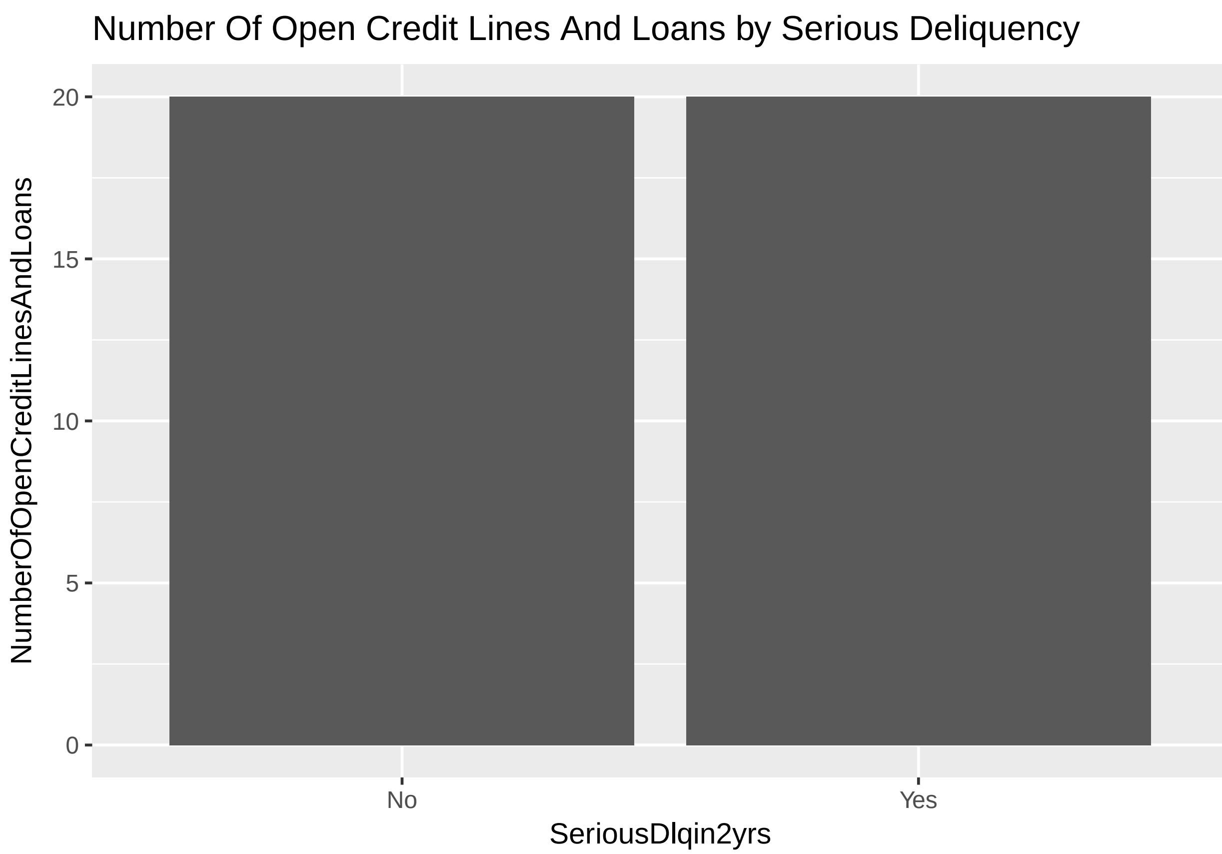

```
t.test(MonthlyIncome~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  MonthlyIncome by SeriousDlqin2yrs
## t = 20.66, df = 7083.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  684.1058 827.5337
## sample estimates:
##  mean in group No mean in group Yes
##          5886.675          5130.856
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent according to Number Of Open Credit Lines And Loans.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(credit_clean, aes(SeriousDlqin2yrs,NumberOfOpenCreditLinesAndLoans)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "Number Of Open Credit Lines And
```



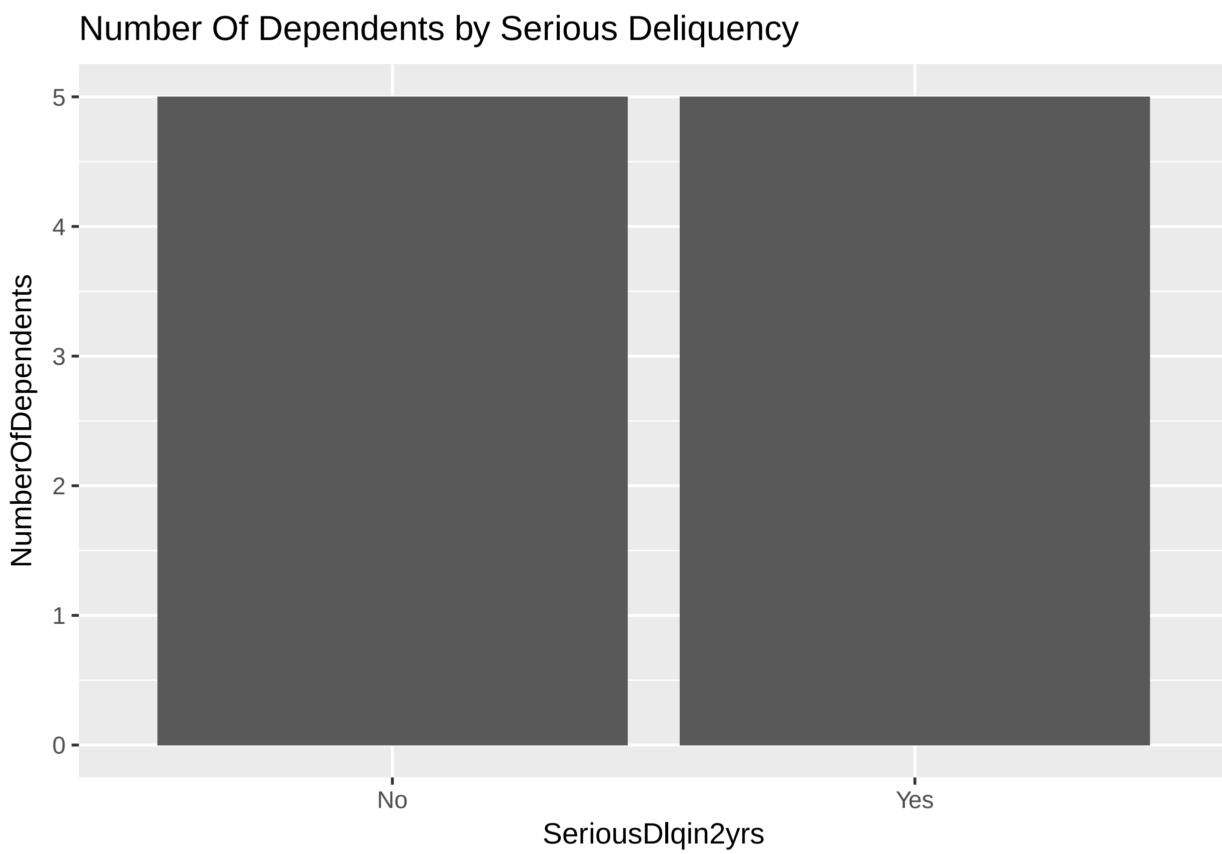

```
t.test(NumberOfOpenCreditLinesAndLoans~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  NumberOfOpenCreditLinesAndLoans by SeriousDlqin2yrs
## t = 10.261, df = 6772.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5156021 0.7591244
## sample estimates:
##  mean in group No mean in group Yes
##          8.187920          7.550556
```

From the barplot, there were an equivalent number of individuals who were seriously delinquent in according to Number of Dependents.

According to the t-test, there is a significant difference within the mean of the groups.

```
ggplot(credit_clean, aes(SeriousDlqin2yrs,NumberOfDependents)) +
    geom_bar(stat="identity", position=position_dodge()) + labs(title = "Number Of Dependents by Serious
```




```r
t.test(NumberOfDependents~SeriousDlqin2yrs, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  NumberOfDependents by SeriousDlqin2yrs
## t = -13.737, df = 6790.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2470072 -0.1853146
## sample estimates:
##  mean in group No mean in group Yes
##         0.8049451         1.0211060
```

Here, we checked the number of proportionate values for SeriousDlqin2yrs for unbalancing.

```r
table(credit_clean$SeriousDlqin2yrs) # view the balance of the out
```

```
##
##    No   Yes
## 94599  6112
```

Modeling

Now we are going to begin introducing our cleaned and prepared data to different classification techniques to predict if an individual will experience financial distress in the next two years.

First, we are going to split our data between 70/30 training and testing set.

```
library(caret)
attach(credit_clean)
sample_data <- createDataPartition(SeriousDlqin2yrs, p = 0.7, list = FALSE) # splitting 70/30 train and
train <- credit_clean[sample_data,] # training set
test <- credit_clean[-sample_data,] # testing set
```

```
table(train$SeriousDlqin2yrs) # viewing the train set proportionality of the outcome variable
```

```
##
##    No   Yes
## 66220  4279
```

```
table(test$SeriousDlqin2yrs) # viewing the test set proportionality of the outcome variable
```

```
##
##    No   Yes
## 28379  1833
```

After splitting the data, its good to make sure there are a equivalent amount of columns for the training and testing set.

```
dim(train)
```

```
## [1] 70499    10
```

```
dim(test)
```

```
## [1] 30212    10
```

Logistic Regression

First, we begin our modeling techniques with logistic regression. Since our outcome variable (SeriousDlqin2yrs) is dichotomous, we are going to use the non-linear approach because the its not a linear outcome. Also, note that we must specify family = "binomial" for a binary classification context.

```
library(ROCR)
library(ROSE)
logit_model <- glm(SeriousDlqin2yrs ~.-X, data = train, family = "binomial") # binomial for binary clas
summary(logit_model) # summary of the model
```

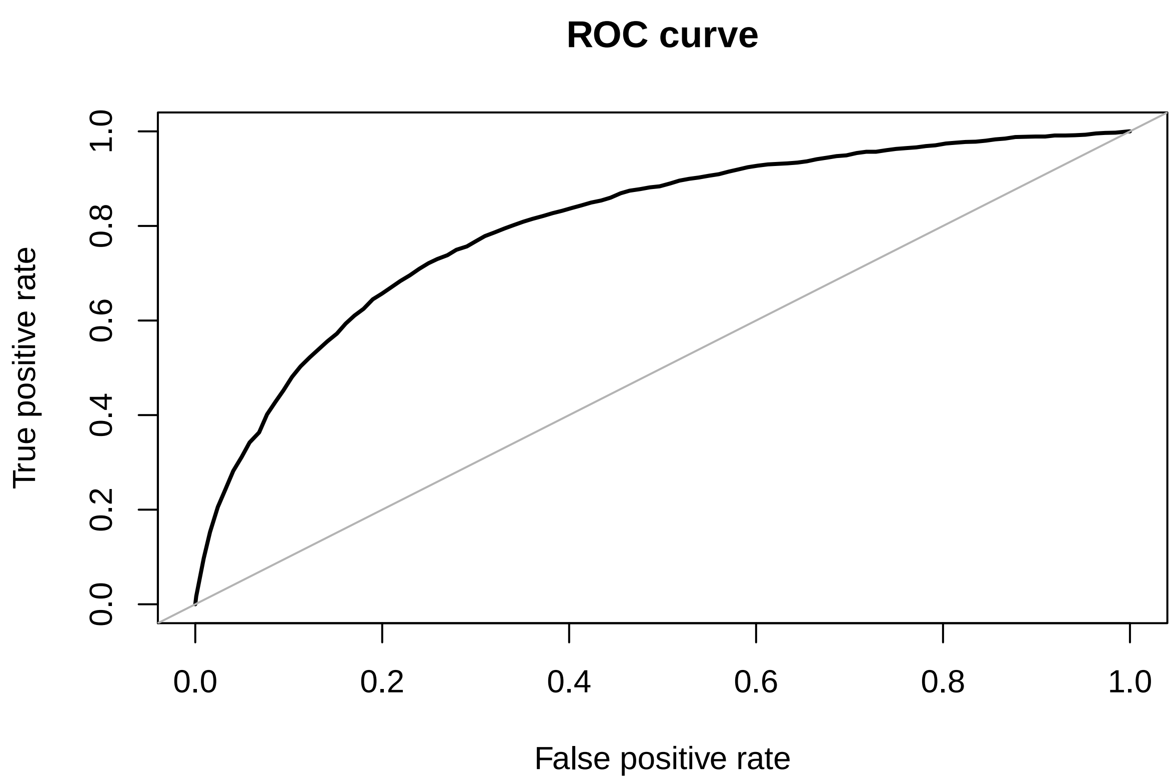
```
##
## Call:
## glm(formula = SeriousDlqin2yrs ~ . - X, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7812  -0.3400  -0.2278  -0.1792   3.1080
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                          -3.220e+00  8.293e-02 -38.825
## RevolvingUtilizationOfUnsecuredLines  2.110e+00  5.176e-02  40.765
## age                                  -1.289e-02  1.355e-03  -9.518
## NumberOfTime30.59DaysPastDueNotWorse  6.158e-01  1.607e-02  38.310
## DebtRatio                             1.648e-01  9.983e-02   1.650
```

```
## MonthlyIncome                          -7.263e-05  7.772e-06  -9.345
## NumberOfOpenCreditLinesAndLoans         8.950e-03  4.999e-03   1.790
## NumberRealEstateLoansOrLines            4.398e-02  2.562e-02   1.717
## NumberOfDependents                      1.047e-01  1.460e-02   7.173
##                                       Pr(>|z|)
## (Intercept)                            < 2e-16 ***
## RevolvingUtilizationOfUnsecuredLines   < 2e-16 ***
## age                                    < 2e-16 ***
## NumberOfTime30.59DaysPastDueNotWorse   < 2e-16 ***
## DebtRatio                               0.0989 .
## MonthlyIncome                          < 2e-16 ***
## NumberOfOpenCreditLinesAndLoans         0.0734 .
## NumberRealEstateLoansOrLines            0.0860 .
## NumberOfDependents                    7.34e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32271  on 70498  degrees of freedom
## Residual deviance: 26951  on 70490  degrees of freedom
## AIC: 26969
##
## Number of Fisher Scoring iterations: 6
```

```
logit_pred <- predict(logit_model,newdata = test, type = "response") # predicting the class on unseen d
logit_preds <- ifelse(logit_pred > 0.5, "Yes", "No") # threshold probabilities greater than 0.5
confusionMatrix(table(logit_preds,test$SeriousDlqin2yrs)) # confusion matrix and Kappa Statistic
```

```
## Confusion Matrix and Statistics
##
##
## logit_preds    No   Yes
##         No  28269  1751
##         Yes   110    82
##
##                Accuracy : 0.9384
##                  95% CI : (0.9356, 0.9411)
##     No Information Rate : 0.9393
##     P-Value [Acc > NIR] : 0.7545
##
##                   Kappa : 0.0703
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99612
##             Specificity : 0.04474
##          Pos Pred Value : 0.94167
##          Neg Pred Value : 0.42708
##              Prevalence : 0.93933
##          Detection Rate : 0.93569
##    Detection Prevalence : 0.99364
##       Balanced Accuracy : 0.52043
##
##        'Positive' Class : No
##
```

```r
roc.curve(test$SeriousDlqin2yrs, logit_pred) #
```




```
## Area under the curve (AUC): 0.803
```

Noticed, in the summary table, there are a couple of variables that were not significant to our model, DebitRatio, NumberOfOpenCreditLinesAndLoans and NumberRealEstateLoansOrLines. They had p-values greater than 0.05, so we are going to remove both them to see if they will make difference in improving the accuracy of our next model.

How do we know that 0.5 value is the "optimal" value for accuracy. In reality, other cutoff values may be better (although 0.5 will tend to be the best value if all model assumptions are true and the sample size is reasonably large since we are dealing with a binary outcome).

From our current model, it classification accuracy of 94% is very good but it seems like the learning algorithm has some issues with overfitting. If you look at the Kappa statistic, it has a value of 0.08 or 8% on a 100% scale. This means our model has a agreement equivalent to chance which means guessing in other words. Before, implementing our second model, we are going to balance the data, to improve the Kappa statistic and possibly the overall accuracy of our model.

For the ROC plot, we would like the curve to "hug" the right and upper borders of the plot (indicating high sensitivity and specificity). Although it's not as closer to the upper right boarders as I expected, we will evaluate it on our next model to see if has improved.

Oversampling unbalanced data

As stated above in our previous model, there were some problems with overfitting and unproportioned outcome variable imbalances. So we performed an oversampling method that works with minority class. It replicates the observations from minority class to balance the data. Since our training and testing data are severly unbalanced, we are going to perform this technique on both samples.

```
data_oversample_train <- ovun.sample(SeriousDlqin2yrs ~.,data = train, method = "over")$data
table(data_oversample_train$SeriousDlqin2yrs)
```

```
##
##    No   Yes
## 66220 66226
```

```
data_oversample_test <- ovun.sample(SeriousDlqin2yrs ~.,data = test, method = "over")$data
table(data_oversample_test$SeriousDlqin2yrs)
```

```
##
##    No   Yes
## 28379 28331
```

As you can see from the results above, they are now both balanced now.

Below, we have our second logistic regression model. We removed a few of the predictor variables that were not significant to our first model. Noticed, we also incorporated the oversampled samples for our training and testing data.

```
logit_model2 <- glm(SeriousDlqin2yrs ~ RevolvingUtilizationOfUnsecuredLines + age + NumberOfTime30.59Day
summary(logit_model2)
```

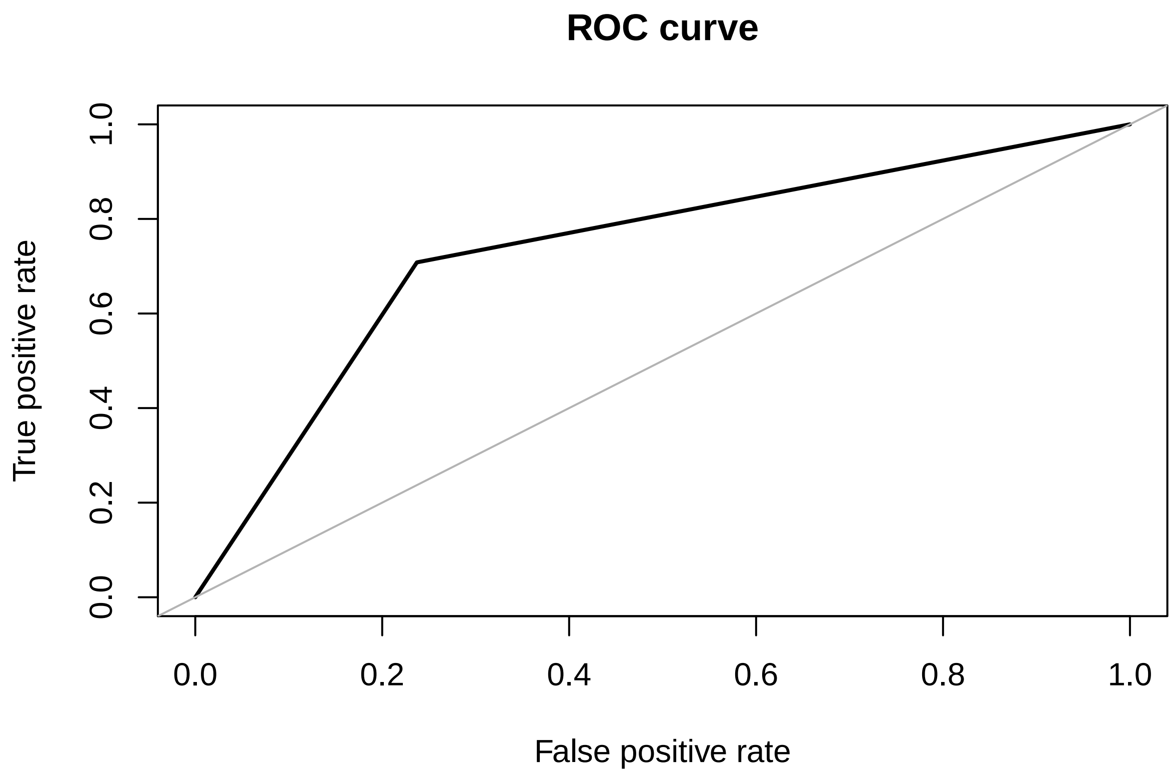
```
##
## Call:
## glm(formula = SeriousDlqin2yrs ~ RevolvingUtilizationOfUnsecuredLines +
##     age + NumberOfTime30.59DaysPastDueNotWorse + MonthlyIncome +
##     NumberOfDependents, family = "binomial", data = data_oversample_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0543  -0.8135   0.0369   0.8624   2.1223
##
## Coefficients:
##                                        Estimate Std. Error z value
## (Intercept)                          -4.416e-01  2.933e-02  -15.06
## RevolvingUtilizationOfUnsecuredLines  2.134e+00  1.829e-02  116.71
## age                                  -1.370e-02  4.922e-04  -27.84
## NumberOfTime30.59DaysPastDueNotWorse  8.262e-01  9.356e-03   88.31
## MonthlyIncome                        -6.160e-05  2.349e-06  -26.22
## NumberOfDependents                    1.022e-01  5.796e-03   17.63
##                                      Pr(>|z|)
## (Intercept)                            <2e-16 ***
## RevolvingUtilizationOfUnsecuredLines   <2e-16 ***
## age                                    <2e-16 ***
## NumberOfTime30.59DaysPastDueNotWorse   <2e-16 ***
## MonthlyIncome                          <2e-16 ***
## NumberOfDependents                     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 183609  on 132445  degrees of freedom
```

```
## Residual deviance: 142809  on 132440  degrees of freedom
## AIC: 142821
##
## Number of Fisher Scoring iterations: 5
```

```
os_pred <- predict(logit_model2, newdata = data_oversample_test, type = "response") # predicting the cl
os_preds <- ifelse(os_pred > 0.5,"Yes", "No") # threshold of probabilities greater than 0.5
confusionMatrix(table(data_oversample_test$SeriousDlqin2yrs,os_preds)) # confusion matrix
```

```
## Confusion Matrix and Statistics
##
##      os_preds
##          No   Yes
##   No  21650  6729
##   Yes  8269 20062
##
##                Accuracy : 0.7355
##                  95% CI : (0.7319, 0.7392)
##     No Information Rate : 0.5276
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.471
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7236
##             Specificity : 0.7488
##          Pos Pred Value : 0.7629
##          Neg Pred Value : 0.7081
##              Prevalence : 0.5276
##          Detection Rate : 0.3818
##    Detection Prevalence : 0.5004
##       Balanced Accuracy : 0.7362
##
##        'Positive' Class : No
##
```

```
roc.curve(data_oversample_test$SeriousDlqin2yrs, os_preds)
```




```
## Area under the curve (AUC): 0.736
```

From the results of our second model, it didn't performed as expected with a classification accuracy of 74%. Check out the Kappa statistic of a value of 49% has moderate agreement as oppose to no aggreement in our first model.

The ROC value was very similair to our overall classification our model, which means our model is no longer overfitting.

ROSE Sampling

The ROSE sampling method generates data synthetically and provides a better estimate of original data. We wanted to try another balancing technique for our outcome variable to measure if we can improve or receive a better accuracy than using the oversampling method above.

Noticed, the training and testing sample data is clearly proportionate to both levels.

```r
data_rose_train <- ROSE(SeriousDlqin2yrs ~., data = train)$data # synthetic training data generated enl
data_rose_test <- ROSE(SeriousDlqin2yrs ~., data = test)$data # synthetic testing data generated enlarg
table(data_rose_train$SeriousDlqin2yrs)
```

```
##
##    No   Yes
## 35381 35118
```

```r
table(data_rose_test$SeriousDlqin2yrs)
```

```
##
```

```
##    No   Yes
## 15064 15148
```

From incorporating ROSE training and testing sample data, the model performed less than the oversampling method above. It had an accuracy of 73%. The Kappa statistic of a value of 45% has moderate agreement as oppose to no aggreement.

```
logit_model3 <- glm(SeriousDlqin2yrs~.-X, data = data_rose_train, family = "binomial") # added the new
summary(logit_model3) # summary of the model
```

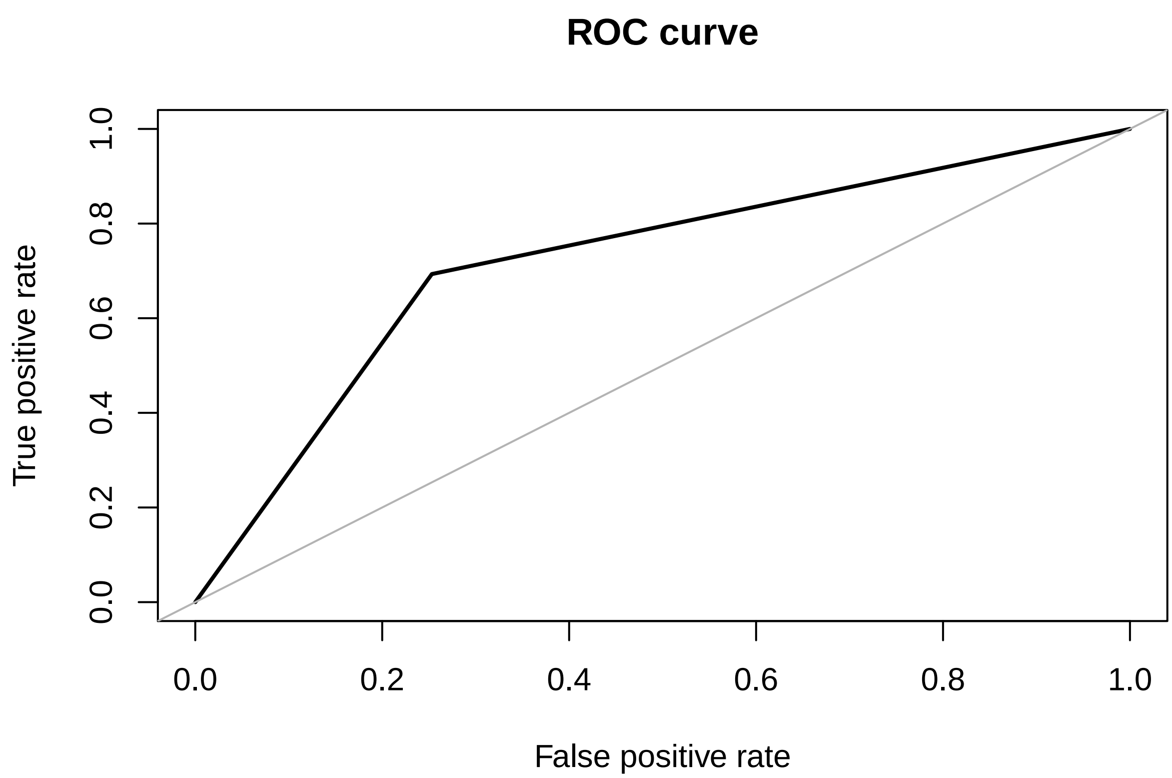
```
##
## Call:
## glm(formula = SeriousDlqin2yrs ~ . - X, family = "binomial",
##     data = data_rose_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6231  -0.8729  -0.4280   0.9058   2.5942
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                         -4.561e-01  3.785e-02 -12.051
## RevolvingUtilizationOfUnsecuredLines  1.956e+00  2.382e-02  82.107
## age                                 -1.363e-02  6.035e-04 -22.578
## NumberOfTime30.59DaysPastDueNotWorse  6.202e-01  1.011e-02  61.365
## DebtRatio                            3.028e-01  4.130e-02   7.333
## MonthlyIncome                       -5.778e-05  3.172e-06 -18.217
## NumberOfOpenCreditLinesAndLoans      5.324e-03  2.078e-03   2.562
## NumberRealEstateLoansOrLines         1.317e-02  1.017e-02   1.295
## NumberOfDependents                   9.647e-02  7.055e-03  13.674
##                                     Pr(>|z|)
## (Intercept)                          < 2e-16 ***
## RevolvingUtilizationOfUnsecuredLines  < 2e-16 ***
## age                                  < 2e-16 ***
## NumberOfTime30.59DaysPastDueNotWorse  < 2e-16 ***
## DebtRatio                           2.26e-13 ***
## MonthlyIncome                        < 2e-16 ***
## NumberOfOpenCreditLinesAndLoans       0.0104 *
## NumberRealEstateLoansOrLines          0.1952
## NumberOfDependents                   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 97731  on 70498  degrees of freedom
## Residual deviance: 78423  on 70490  degrees of freedom
## AIC: 78441
##
## Number of Fisher Scoring iterations: 4
```

```
os_pred3 <- predict(logit_model3, newdata = data_rose_test, type = "response") # predicting the class u
os_preds3 <- ifelse(os_pred3 > 0.5,"Yes", "No")# threshold of probabilities greater than 0.5
confusionMatrix(table(data_rose_test$SeriousDlqin2yrs,os_preds3)) # confusion matrix
```

```
## Confusion Matrix and Statistics
```

```
##
##      os_preds3
##          No   Yes
##   No  11251  3813
##   Yes  4646 10502
##
##                Accuracy : 0.72
##                  95% CI : (0.7149, 0.7251)
##     No Information Rate : 0.5262
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4401
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7077
##             Specificity : 0.7336
##          Pos Pred Value : 0.7469
##          Neg Pred Value : 0.6933
##              Prevalence : 0.5262
##          Detection Rate : 0.3724
##    Detection Prevalence : 0.4986
##       Balanced Accuracy : 0.7207
##
##        'Positive' Class : No
##
```

```r
roc.curve(data_rose_test$SeriousDlqin2yrs, os_preds3)
```




```
## Area under the curve (AUC): 0.720
```

As we can see there was little to no difference in the overall accuracy of model in comparision to oversampling technique above.

Decision Tree

Now we are going to fit a Tree to our data to predict if an individual is going to be seriously delinquent in two years. We are incorporating a cross-validation technique within the decision tree model to find the best optimal value for the complexity parameter for reducing the mean prediction error of our model.

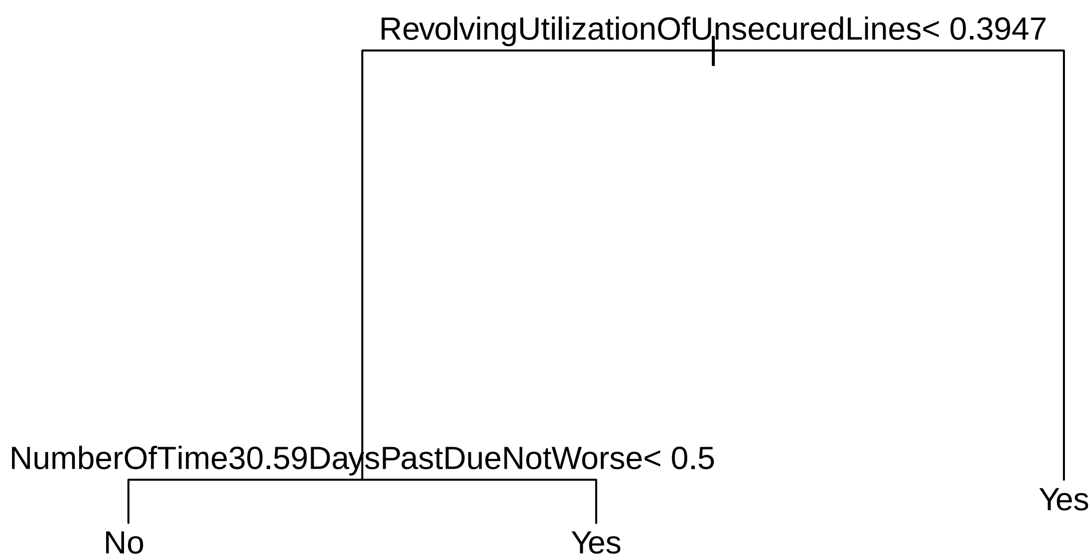
Our Decision Tree tells us Revolving Utilization of Unsecured Lines and Number of Times 30 - 59 Days Past Due were the two most important variables in the model.

```
library(rpart)
rf_model <- train(SeriousDlqin2yrs ~.-X, data = data_oversample_train, method = "rpart", trControl = tra
print(rf_model) # Plot the trees
```

```
## CART
##
## 132446 samples
##      9 predictor
##      2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 119202, 119201, 119201, 119201, 119201, 119202, ...
```

```
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00576865  0.7322757  0.4645510
##   0.04195107  0.7154006  0.4307979
##   0.41643008  0.6020220  0.2040062
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.00576865.
```

```
par(xpd = NA) # Avoid clipping the text in some device
plot(rf_model$finalModel)# Plot the final tree model
text(rf_model$finalModel,  digits = 3) # adding the names of the relevant variable names to the trees
```

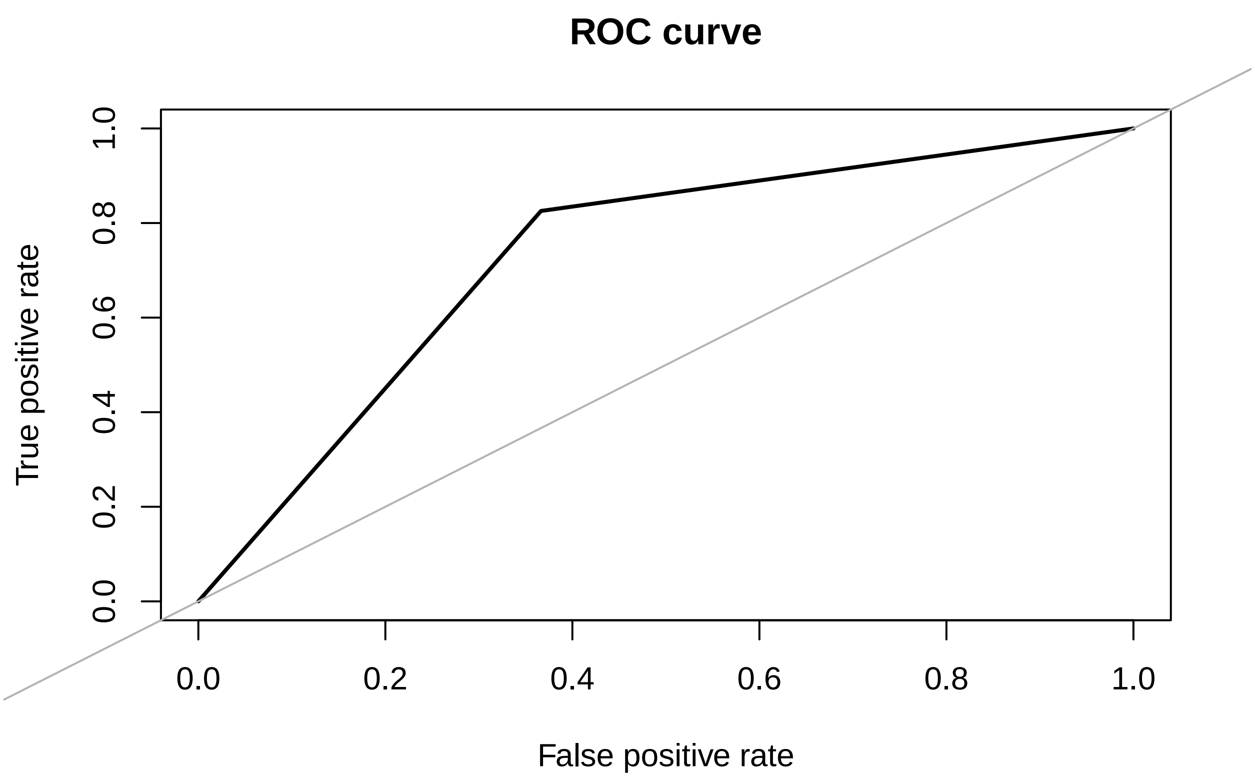



```
dt_y_hat <- predict(rf_model, data_oversample_test)
confusionMatrix(table(dt_y_hat,data_oversample_test$SeriousDlqin2yrs))
```

```
## Confusion Matrix and Statistics
##
##
## dt_y_hat    No   Yes
##      No  17979  4943
##      Yes 10400 23388
##
##                Accuracy : 0.7294
##                  95% CI : (0.7258, 0.7331)
##     No Information Rate : 0.5004
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.459
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6335
##             Specificity : 0.8255
##          Pos Pred Value : 0.7844
##          Neg Pred Value : 0.6922
##              Prevalence : 0.5004
##          Detection Rate : 0.3170
##    Detection Prevalence : 0.4042
##       Balanced Accuracy : 0.7295
##
##        'Positive' Class : No
##
```

```
roc.curve(data_oversample_test$SeriousDlqin2yrs, dt_y_hat)
```




```
## Area under the curve (AUC): 0.730
```

By adding using the optimal value for the complexity paramter from our previous model for another model,there will be no improvement or changes in the overall classification accuracy nor the confusion matrix itself. The reason its going to be no change is due to the cp value being very small.

K-Nearest Neighbor

Now we are going to implement a K-Nearest Neighboor technique identifying the k most similar training observations to our new observation. Also, we incorporae cross-validation along with the K-NN algorithm to find the optimal k value to reduce the mean prediction error.

```
knn_model <- train(SeriousDlqin2yrs ~.-X, data = data_oversample_train, method = "knn",trControl = trai
print(knn_model) # summary of our model
```

```
## k-Nearest Neighbors
##
## 132446 samples
##      9 predictor
##      2 classes: 'No', 'Yes'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 119201, 119202, 119201, 119201, 119201, 119202, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8944400  0.7888780
##    7  0.8678178  0.7356325
##    9  0.8459674  0.6919306
##   11  0.8281639  0.6563226
##   13  0.8153587  0.6307114
##   15  0.8055434  0.6110803
##   17  0.7990728  0.5981390
##   19  0.7942104  0.5884145
##   21  0.7908732  0.5817404
##   23  0.7877852  0.5755648
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
knn_model$bestTune # optimal value for k
```

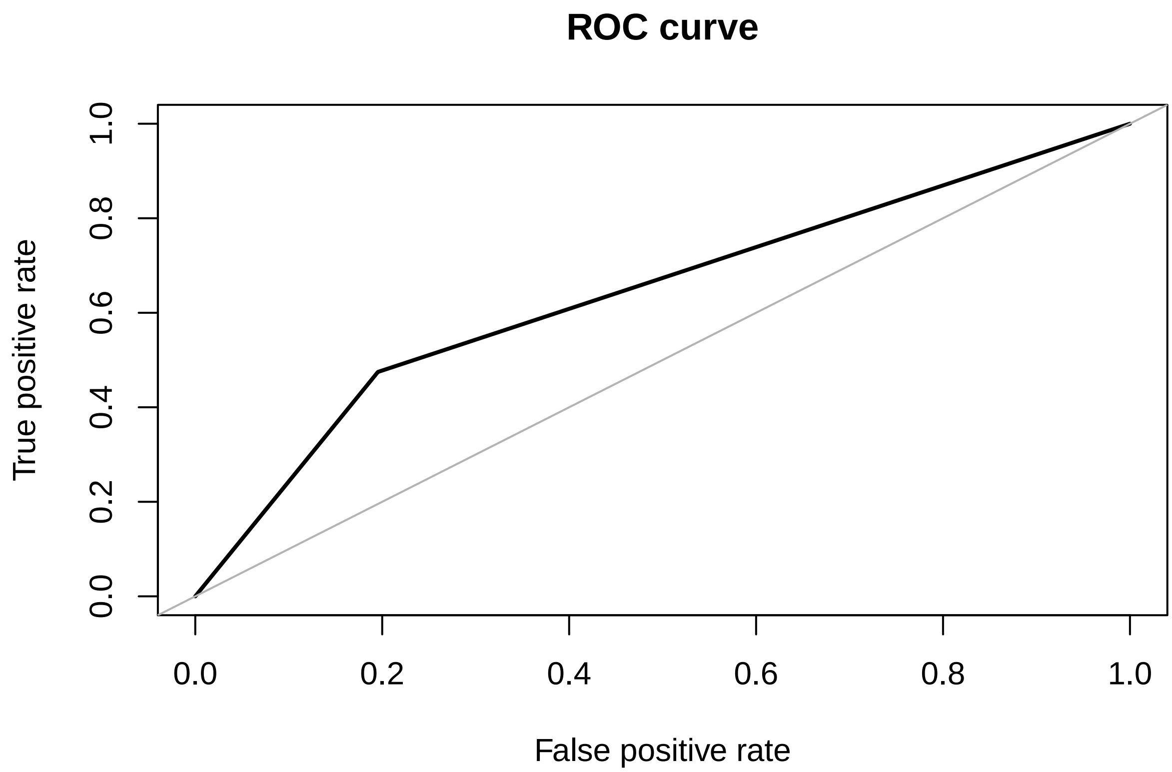
```
##   k
## 1 5
```

```
knn_y_hat <- predict(knn_model, data_oversample_test) # predicting the class on unseen data
confusionMatrix(table(knn_y_hat,data_oversample_test$SeriousDlqin2yrs)) # confusion matrix
```

```
## Confusion Matrix and Statistics
##
##
## knn_y_hat    No   Yes
##       No  22833 14879
##       Yes  5546 13452
##
##                Accuracy : 0.6398
##                  95% CI : (0.6359, 0.6438)
##     No Information Rate : 0.5004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2795
```

```
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8046
##             Specificity : 0.4748
##          Pos Pred Value : 0.6055
##          Neg Pred Value : 0.7081
##              Prevalence : 0.5004
##          Detection Rate : 0.4026
##    Detection Prevalence : 0.6650
##       Balanced Accuracy : 0.6397
##
##        'Positive' Class : No
##
```

```
roc.curve(data_oversample_test$SeriousDlqin2yrs, knn_y_hat)
```




```
## Area under the curve (AUC): 0.640
```

From the results from our K-NN model above, the optimal value for the best accuracy of our model is 5. We are going to use the optimal k value for our tuneLength to measure if our model accuracy will improve or not.

From the results of our model above, it had a classification accuracy of65-%. Check out the Kappa statistic of a value of 30% has slight agreement as oppose to no aggreement.

As you can see, we used 5 for our tuneLength for our second model.

```
ctrl <- trainControl(method="repeatedcv",repeats = 3,classProbs=TRUE,summaryFunction = twoClassSummary)
knn_model2 <- train(SeriousDlqin2yrs ~.-X, data = data_oversample_train, method = "knn",trControl = ctrl
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

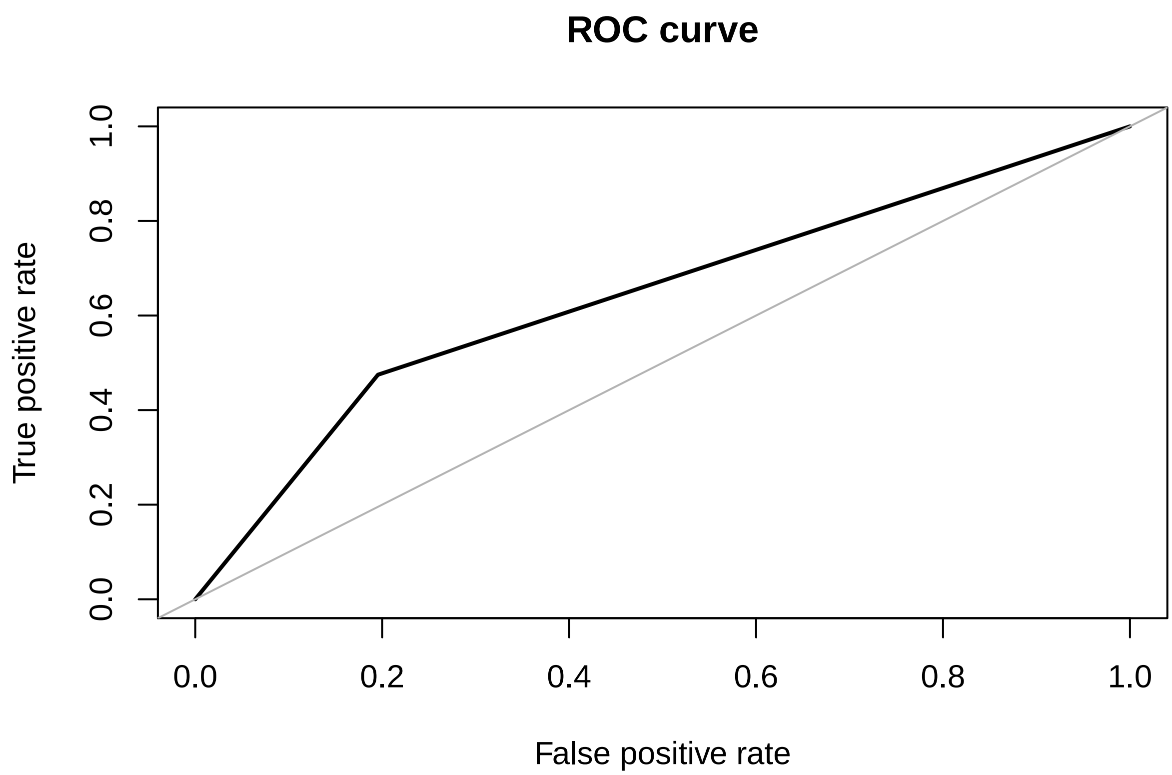
```r
knn_model2
```

```
## k-Nearest Neighbors
##
## 132446 samples
##      9 predictor
##      2 classes: 'No', 'Yes'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 119201, 119201, 119201, 119201, 119202, 119202, ...
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    5  0.9710719  0.7885030  0.9999547
##    7  0.9699722  0.7361019  0.9997131
##    9  0.9662008  0.6929276  0.9991594
##   11  0.9576316  0.6591966  0.9977400
##   13  0.9440488  0.6358502  0.9940557
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```r
knn_y_hat2 <- predict(knn_model2, data_oversample_test) # predicting the class on unseen data
confusionMatrix(knn_y_hat2,data_oversample_test$SeriousDlqin2yrs) # confusion matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  22832 14879
##        Yes  5547 13452
##
##                Accuracy : 0.6398
##                  95% CI : (0.6358, 0.6438)
##     No Information Rate : 0.5004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2794
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8045
##             Specificity : 0.4748
##          Pos Pred Value : 0.6054
##          Neg Pred Value : 0.7080
##              Prevalence : 0.5004
##          Detection Rate : 0.4026
##    Detection Prevalence : 0.6650
##       Balanced Accuracy : 0.6397
##
##        'Positive' Class : No
##
```

```
roc.curve(data_oversample_test$SeriousDlqin2yrs, knn_y_hat2) # ROC curve
```




```
## Area under the curve (AUC): 0.640
```

From the results from our K-NN model above, the optimal value for the best accuracy of our model is 65%. The model did not improve at all.

Ensemble Learning

Given a list of caret models, the caretStack() function can be used to specify a higher-order model to learn how to best combine the predictions of sub-models together.

Let's first look at creating 5 sub-models for to finish our analysis specifically:

```
Gradient Boosting (GBM)
Classification and Regression Trees (CART)
Logistic Regression (via Generalized Linear Model or GLM)
k-Nearest Neighbors (kNN)
Naive Bayes (NB)
```

Below is an example that creates these 5 sub-models. Note the new helpful caretList() function provided by the caretEnsemble package for creating a list of standard caret models.

```
library(caretEnsemble)
control <- trainControl(method="repeatedcv", number=10, repeats=3, savePredictions='final', classProbs=T
algorithmList <- c('gbm', 'rpart', 'glm', 'knn', 'nb') # stacking 5 modeling techniques
ensemble_learning <- caretList(SeriousDlqin2yrs~.-X, data=data_oversample_train, trControl=control, met
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0165
##      2        1.3263             nan     0.1000    0.0133
##      3        1.3038             nan     0.1000    0.0112
##      4        1.2824             nan     0.1000    0.0107
##      5        1.2649             nan     0.1000    0.0088
##      6        1.2484             nan     0.1000    0.0082
##      7        1.2344             nan     0.1000    0.0072
##      8        1.2214             nan     0.1000    0.0066
##      9        1.2099             nan     0.1000    0.0057
##     10        1.1996             nan     0.1000    0.0052
##     20        1.1356             nan     0.1000    0.0020
##     40        1.0935             nan     0.1000    0.0005
##     60        1.0780             nan     0.1000    0.0003
##     80        1.0703             nan     0.1000    0.0001
##    100        1.0649             nan     0.1000    0.0001
##    120        1.0606             nan     0.1000    0.0001
##    140        1.0575             nan     0.1000    0.0000
##    150        1.0562             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3446             nan     0.1000    0.0210
##      2        1.3083             nan     0.1000    0.0180
##      3        1.2803             nan     0.1000    0.0140
##      4        1.2558             nan     0.1000    0.0123
##      5        1.2342             nan     0.1000    0.0108
##      6        1.2146             nan     0.1000    0.0098
##      7        1.1989             nan     0.1000    0.0078
##      8        1.1846             nan     0.1000    0.0071
##      9        1.1718             nan     0.1000    0.0064
##     10        1.1613             nan     0.1000    0.0052
##     20        1.1041             nan     0.1000    0.0015
##     40        1.0700             nan     0.1000    0.0003
##     60        1.0576             nan     0.1000    0.0002
##     80        1.0497             nan     0.1000    0.0002
##    100        1.0445             nan     0.1000    0.0001
##    120        1.0405             nan     0.1000    0.0001
##    140        1.0371             nan     0.1000    0.0001
##    150        1.0356             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3391             nan     0.1000    0.0236
##      2        1.2998             nan     0.1000    0.0195
##      3        1.2681             nan     0.1000    0.0158
##      4        1.2413             nan     0.1000    0.0133
##      5        1.2196             nan     0.1000    0.0109
##      6        1.2011             nan     0.1000    0.0093
##      7        1.1854             nan     0.1000    0.0078
##      8        1.1715             nan     0.1000    0.0070
##      9        1.1598             nan     0.1000    0.0058
##     10        1.1494             nan     0.1000    0.0051
##     20        1.0919             nan     0.1000    0.0017
##     40        1.0592             nan     0.1000    0.0003
##     60        1.0461             nan     0.1000    0.0002
```

```
##     80        1.0379             nan     0.1000    0.0003
##    100        1.0318             nan     0.1000    0.0001
##    120        1.0274             nan     0.1000    0.0001
##    140        1.0235             nan     0.1000    0.0000
##    150        1.0217             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0165
##      2        1.3266             nan     0.1000    0.0135
##      3        1.3044             nan     0.1000    0.0111
##      4        1.2834             nan     0.1000    0.0106
##      5        1.2658             nan     0.1000    0.0088
##      6        1.2493             nan     0.1000    0.0083
##      7        1.2350             nan     0.1000    0.0071
##      8        1.2221             nan     0.1000    0.0065
##      9        1.2106             nan     0.1000    0.0057
##     10        1.2005             nan     0.1000    0.0050
##     20        1.1368             nan     0.1000    0.0020
##     40        1.0949             nan     0.1000    0.0005
##     60        1.0798             nan     0.1000    0.0002
##     80        1.0716             nan     0.1000    0.0002
##    100        1.0665             nan     0.1000    0.0001
##    120        1.0624             nan     0.1000    0.0001
##    140        1.0591             nan     0.1000    0.0001
##    150        1.0578             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3444             nan     0.1000    0.0210
##      2        1.3095             nan     0.1000    0.0174
##      3        1.2800             nan     0.1000    0.0148
##      4        1.2558             nan     0.1000    0.0119
##      5        1.2342             nan     0.1000    0.0108
##      6        1.2143             nan     0.1000    0.0100
##      7        1.1997             nan     0.1000    0.0073
##      8        1.1855             nan     0.1000    0.0070
##      9        1.1734             nan     0.1000    0.0060
##     10        1.1618             nan     0.1000    0.0057
##     20        1.1050             nan     0.1000    0.0017
##     40        1.0714             nan     0.1000    0.0004
##     60        1.0579             nan     0.1000    0.0002
##     80        1.0509             nan     0.1000    0.0001
##    100        1.0455             nan     0.1000    0.0001
##    120        1.0410             nan     0.1000    0.0001
##    140        1.0377             nan     0.1000    0.0000
##    150        1.0363             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3391             nan     0.1000    0.0235
##      2        1.3002             nan     0.1000    0.0194
##      3        1.2688             nan     0.1000    0.0158
##      4        1.2423             nan     0.1000    0.0132
##      5        1.2204             nan     0.1000    0.0109
##      6        1.2021             nan     0.1000    0.0092
##      7        1.1865             nan     0.1000    0.0078
```

```
##      8        1.1725             nan     0.1000    0.0069
##      9        1.1604             nan     0.1000    0.0061
##     10        1.1506             nan     0.1000    0.0049
##     20        1.0939             nan     0.1000    0.0013
##     40        1.0604             nan     0.1000    0.0004
##     60        1.0473             nan     0.1000    0.0002
##     80        1.0397             nan     0.1000    0.0001
##    100        1.0342             nan     0.1000    0.0001
##    120        1.0294             nan     0.1000    0.0001
##    140        1.0252             nan     0.1000    0.0001
##    150        1.0226             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0165
##      2        1.3265             nan     0.1000    0.0135
##      3        1.3042             nan     0.1000    0.0111
##      4        1.2824             nan     0.1000    0.0107
##      5        1.2648             nan     0.1000    0.0087
##      6        1.2480             nan     0.1000    0.0083
##      7        1.2338             nan     0.1000    0.0071
##      8        1.2206             nan     0.1000    0.0066
##      9        1.2090             nan     0.1000    0.0057
##     10        1.1987             nan     0.1000    0.0052
##     20        1.1353             nan     0.1000    0.0021
##     40        1.0938             nan     0.1000    0.0005
##     60        1.0780             nan     0.1000    0.0002
##     80        1.0699             nan     0.1000    0.0001
##    100        1.0647             nan     0.1000    0.0001
##    120        1.0606             nan     0.1000    0.0001
##    140        1.0575             nan     0.1000    0.0001
##    150        1.0562             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3437             nan     0.1000    0.0210
##      2        1.3077             nan     0.1000    0.0179
##      3        1.2793             nan     0.1000    0.0140
##      4        1.2550             nan     0.1000    0.0124
##      5        1.2343             nan     0.1000    0.0104
##      6        1.2151             nan     0.1000    0.0096
##      7        1.1979             nan     0.1000    0.0086
##      8        1.1848             nan     0.1000    0.0066
##      9        1.1730             nan     0.1000    0.0059
##     10        1.1618             nan     0.1000    0.0055
##     20        1.1042             nan     0.1000    0.0015
##     40        1.0705             nan     0.1000    0.0005
##     60        1.0577             nan     0.1000    0.0002
##     80        1.0500             nan     0.1000    0.0001
##    100        1.0449             nan     0.1000    0.0000
##    120        1.0406             nan     0.1000    0.0001
##    140        1.0369             nan     0.1000    0.0001
##    150        1.0353             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0237
```

```
##      2        1.3001             nan     0.1000    0.0194
##      3        1.2687             nan     0.1000    0.0159
##      4        1.2420             nan     0.1000    0.0133
##      5        1.2199             nan     0.1000    0.0110
##      6        1.2012             nan     0.1000    0.0093
##      7        1.1842             nan     0.1000    0.0083
##      8        1.1708             nan     0.1000    0.0066
##      9        1.1582             nan     0.1000    0.0062
##     10        1.1474             nan     0.1000    0.0053
##     20        1.0920             nan     0.1000    0.0017
##     40        1.0590             nan     0.1000    0.0004
##     60        1.0457             nan     0.1000    0.0003
##     80        1.0383             nan     0.1000    0.0001
##    100        1.0327             nan     0.1000    0.0001
##    120        1.0277             nan     0.1000    0.0002
##    140        1.0236             nan     0.1000    0.0001
##    150        1.0217             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3536             nan     0.1000    0.0165
##      2        1.3268             nan     0.1000    0.0135
##      3        1.3043             nan     0.1000    0.0112
##      4        1.2829             nan     0.1000    0.0108
##      5        1.2650             nan     0.1000    0.0089
##      6        1.2484             nan     0.1000    0.0084
##      7        1.2340             nan     0.1000    0.0070
##      8        1.2207             nan     0.1000    0.0065
##      9        1.2092             nan     0.1000    0.0057
##     10        1.1989             nan     0.1000    0.0052
##     20        1.1354             nan     0.1000    0.0020
##     40        1.0933             nan     0.1000    0.0005
##     60        1.0778             nan     0.1000    0.0003
##     80        1.0700             nan     0.1000    0.0001
##    100        1.0646             nan     0.1000    0.0001
##    120        1.0604             nan     0.1000    0.0001
##    140        1.0571             nan     0.1000    0.0001
##    150        1.0558             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3443             nan     0.1000    0.0210
##      2        1.3097             nan     0.1000    0.0171
##      3        1.2797             nan     0.1000    0.0150
##      4        1.2550             nan     0.1000    0.0123
##      5        1.2318             nan     0.1000    0.0116
##      6        1.2137             nan     0.1000    0.0091
##      7        1.1987             nan     0.1000    0.0076
##      8        1.1854             nan     0.1000    0.0066
##      9        1.1735             nan     0.1000    0.0058
##     10        1.1623             nan     0.1000    0.0056
##     20        1.1033             nan     0.1000    0.0018
##     40        1.0699             nan     0.1000    0.0004
##     60        1.0571             nan     0.1000    0.0002
##     80        1.0495             nan     0.1000    0.0001
##    100        1.0440             nan     0.1000    0.0001
```

```
##    120        1.0395             nan     0.1000    0.0000
##    140        1.0361             nan     0.1000    0.0000
##    150        1.0346             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0237
##      2        1.2996             nan     0.1000    0.0195
##      3        1.2680             nan     0.1000    0.0159
##      4        1.2417             nan     0.1000    0.0131
##      5        1.2197             nan     0.1000    0.0110
##      6        1.1998             nan     0.1000    0.0100
##      7        1.1828             nan     0.1000    0.0084
##      8        1.1693             nan     0.1000    0.0068
##      9        1.1578             nan     0.1000    0.0057
##     10        1.1469             nan     0.1000    0.0054
##     20        1.0914             nan     0.1000    0.0013
##     40        1.0584             nan     0.1000    0.0004
##     60        1.0456             nan     0.1000    0.0002
##     80        1.0373             nan     0.1000    0.0002
##    100        1.0321             nan     0.1000    0.0001
##    120        1.0268             nan     0.1000    0.0002
##    140        1.0228             nan     0.1000    0.0001
##    150        1.0210             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3535             nan     0.1000    0.0165
##      2        1.3264             nan     0.1000    0.0133
##      3        1.3042             nan     0.1000    0.0110
##      4        1.2826             nan     0.1000    0.0107
##      5        1.2650             nan     0.1000    0.0087
##      6        1.2481             nan     0.1000    0.0084
##      7        1.2339             nan     0.1000    0.0071
##      8        1.2208             nan     0.1000    0.0066
##      9        1.2093             nan     0.1000    0.0057
##     10        1.1992             nan     0.1000    0.0051
##     20        1.1354             nan     0.1000    0.0019
##     40        1.0937             nan     0.1000    0.0005
##     60        1.0782             nan     0.1000    0.0002
##     80        1.0700             nan     0.1000    0.0001
##    100        1.0646             nan     0.1000    0.0001
##    120        1.0607             nan     0.1000    0.0001
##    140        1.0574             nan     0.1000    0.0001
##    150        1.0560             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3441             nan     0.1000    0.0210
##      2        1.3081             nan     0.1000    0.0179
##      3        1.2798             nan     0.1000    0.0140
##      4        1.2547             nan     0.1000    0.0125
##      5        1.2339             nan     0.1000    0.0102
##      6        1.2135             nan     0.1000    0.0101
##      7        1.1986             nan     0.1000    0.0074
##      8        1.1845             nan     0.1000    0.0070
##      9        1.1726             nan     0.1000    0.0059
```

```
##     10        1.1624             nan     0.1000    0.0051
##     20        1.1032             nan     0.1000    0.0016
##     40        1.0695             nan     0.1000    0.0005
##     60        1.0570             nan     0.1000    0.0002
##     80        1.0491             nan     0.1000    0.0002
##    100        1.0433             nan     0.1000    0.0001
##    120        1.0390             nan     0.1000    0.0001
##    140        1.0359             nan     0.1000    0.0001
##    150        1.0344             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3391             nan     0.1000    0.0237
##      2        1.3009             nan     0.1000    0.0193
##      3        1.2684             nan     0.1000    0.0161
##      4        1.2424             nan     0.1000    0.0131
##      5        1.2197             nan     0.1000    0.0114
##      6        1.2011             nan     0.1000    0.0092
##      7        1.1844             nan     0.1000    0.0083
##      8        1.1701             nan     0.1000    0.0071
##      9        1.1585             nan     0.1000    0.0058
##     10        1.1476             nan     0.1000    0.0055
##     20        1.0916             nan     0.1000    0.0015
##     40        1.0586             nan     0.1000    0.0005
##     60        1.0460             nan     0.1000    0.0002
##     80        1.0384             nan     0.1000    0.0001
##    100        1.0326             nan     0.1000    0.0001
##    120        1.0275             nan     0.1000    0.0001
##    140        1.0228             nan     0.1000    0.0001
##    150        1.0210             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3528             nan     0.1000    0.0165
##      2        1.3259             nan     0.1000    0.0134
##      3        1.3034             nan     0.1000    0.0112
##      4        1.2824             nan     0.1000    0.0106
##      5        1.2650             nan     0.1000    0.0087
##      6        1.2487             nan     0.1000    0.0083
##      7        1.2347             nan     0.1000    0.0070
##      8        1.2217             nan     0.1000    0.0066
##      9        1.2105             nan     0.1000    0.0056
##     10        1.2003             nan     0.1000    0.0050
##     20        1.1359             nan     0.1000    0.0022
##     40        1.0943             nan     0.1000    0.0005
##     60        1.0792             nan     0.1000    0.0002
##     80        1.0715             nan     0.1000    0.0001
##    100        1.0661             nan     0.1000    0.0001
##    120        1.0622             nan     0.1000    0.0001
##    140        1.0591             nan     0.1000    0.0000
##    150        1.0578             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0210
##      2        1.3101             nan     0.1000    0.0171
##      3        1.2802             nan     0.1000    0.0149
```

```
##      4        1.2556             nan     0.1000    0.0122
##      5        1.2327             nan     0.1000    0.0115
##      6        1.2145             nan     0.1000    0.0091
##      7        1.1992             nan     0.1000    0.0076
##      8        1.1852             nan     0.1000    0.0070
##      9        1.1729             nan     0.1000    0.0062
##     10        1.1617             nan     0.1000    0.0057
##     20        1.1040             nan     0.1000    0.0017
##     40        1.0703             nan     0.1000    0.0004
##     60        1.0577             nan     0.1000    0.0001
##     80        1.0505             nan     0.1000    0.0002
##    100        1.0456             nan     0.1000    0.0001
##    120        1.0417             nan     0.1000    0.0001
##    140        1.0382             nan     0.1000    0.0001
##    150        1.0368             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3386             nan     0.1000    0.0236
##      2        1.3000             nan     0.1000    0.0193
##      3        1.2684             nan     0.1000    0.0158
##      4        1.2421             nan     0.1000    0.0132
##      5        1.2200             nan     0.1000    0.0109
##      6        1.2015             nan     0.1000    0.0093
##      7        1.1840             nan     0.1000    0.0087
##      8        1.1705             nan     0.1000    0.0067
##      9        1.1581             nan     0.1000    0.0062
##     10        1.1482             nan     0.1000    0.0048
##     20        1.0929             nan     0.1000    0.0014
##     40        1.0596             nan     0.1000    0.0004
##     60        1.0473             nan     0.1000    0.0003
##     80        1.0395             nan     0.1000    0.0001
##    100        1.0336             nan     0.1000    0.0001
##    120        1.0290             nan     0.1000    0.0001
##    140        1.0250             nan     0.1000    0.0001
##    150        1.0234             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3541             nan     0.1000    0.0164
##      2        1.3274             nan     0.1000    0.0134
##      3        1.3051             nan     0.1000    0.0112
##      4        1.2838             nan     0.1000    0.0106
##      5        1.2661             nan     0.1000    0.0087
##      6        1.2495             nan     0.1000    0.0083
##      7        1.2354             nan     0.1000    0.0070
##      8        1.2221             nan     0.1000    0.0066
##      9        1.2106             nan     0.1000    0.0057
##     10        1.2003             nan     0.1000    0.0052
##     20        1.1369             nan     0.1000    0.0022
##     40        1.0957             nan     0.1000    0.0005
##     60        1.0805             nan     0.1000    0.0002
##     80        1.0724             nan     0.1000    0.0001
##    100        1.0671             nan     0.1000    0.0001
##    120        1.0631             nan     0.1000    0.0001
##    140        1.0599             nan     0.1000    0.0000
```

```
##    150        1.0586             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3443             nan     0.1000    0.0209
##      2        1.3088             nan     0.1000    0.0178
##      3        1.2809             nan     0.1000    0.0139
##      4        1.2570             nan     0.1000    0.0119
##      5        1.2348             nan     0.1000    0.0111
##      6        1.2147             nan     0.1000    0.0100
##      7        1.2002             nan     0.1000    0.0073
##      8        1.1869             nan     0.1000    0.0066
##      9        1.1742             nan     0.1000    0.0063
##     10        1.1627             nan     0.1000    0.0058
##     20        1.1060             nan     0.1000    0.0018
##     40        1.0723             nan     0.1000    0.0004
##     60        1.0594             nan     0.1000    0.0002
##     80        1.0519             nan     0.1000    0.0001
##    100        1.0464             nan     0.1000    0.0001
##    120        1.0423             nan     0.1000    0.0001
##    140        1.0390             nan     0.1000    0.0001
##    150        1.0376             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0235
##      2        1.3004             nan     0.1000    0.0193
##      3        1.2689             nan     0.1000    0.0157
##      4        1.2427             nan     0.1000    0.0130
##      5        1.2209             nan     0.1000    0.0108
##      6        1.2022             nan     0.1000    0.0093
##      7        1.1860             nan     0.1000    0.0082
##      8        1.1726             nan     0.1000    0.0067
##      9        1.1602             nan     0.1000    0.0062
##     10        1.1497             nan     0.1000    0.0053
##     20        1.0943             nan     0.1000    0.0014
##     40        1.0619             nan     0.1000    0.0003
##     60        1.0491             nan     0.1000    0.0002
##     80        1.0413             nan     0.1000    0.0001
##    100        1.0353             nan     0.1000    0.0001
##    120        1.0306             nan     0.1000    0.0001
##    140        1.0264             nan     0.1000    0.0000
##    150        1.0243             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3531             nan     0.1000    0.0165
##      2        1.3260             nan     0.1000    0.0135
##      3        1.3034             nan     0.1000    0.0112
##      4        1.2822             nan     0.1000    0.0107
##      5        1.2645             nan     0.1000    0.0087
##      6        1.2480             nan     0.1000    0.0083
##      7        1.2337             nan     0.1000    0.0071
##      8        1.2207             nan     0.1000    0.0066
##      9        1.2093             nan     0.1000    0.0057
##     10        1.1992             nan     0.1000    0.0050
##     20        1.1356             nan     0.1000    0.0020
```

```
##     40        1.0942             nan     0.1000    0.0005
##     60        1.0791             nan     0.1000    0.0002
##     80        1.0708             nan     0.1000    0.0001
##    100        1.0654             nan     0.1000    0.0001
##    120        1.0614             nan     0.1000    0.0001
##    140        1.0583             nan     0.1000    0.0001
##    150        1.0570             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3446             nan     0.1000    0.0211
##      2        1.3104             nan     0.1000    0.0172
##      3        1.2805             nan     0.1000    0.0151
##      4        1.2567             nan     0.1000    0.0119
##      5        1.2352             nan     0.1000    0.0108
##      6        1.2158             nan     0.1000    0.0097
##      7        1.1984             nan     0.1000    0.0085
##      8        1.1843             nan     0.1000    0.0071
##      9        1.1714             nan     0.1000    0.0064
##     10        1.1608             nan     0.1000    0.0053
##     20        1.1042             nan     0.1000    0.0017
##     40        1.0708             nan     0.1000    0.0005
##     60        1.0583             nan     0.1000    0.0002
##     80        1.0508             nan     0.1000    0.0001
##    100        1.0452             nan     0.1000    0.0001
##    120        1.0410             nan     0.1000    0.0000
##    140        1.0375             nan     0.1000    0.0001
##    150        1.0362             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0237
##      2        1.3000             nan     0.1000    0.0195
##      3        1.2679             nan     0.1000    0.0159
##      4        1.2415             nan     0.1000    0.0133
##      5        1.2197             nan     0.1000    0.0109
##      6        1.2011             nan     0.1000    0.0093
##      7        1.1839             nan     0.1000    0.0084
##      8        1.1691             nan     0.1000    0.0074
##      9        1.1571             nan     0.1000    0.0060
##     10        1.1473             nan     0.1000    0.0049
##     20        1.0924             nan     0.1000    0.0018
##     40        1.0601             nan     0.1000    0.0004
##     60        1.0476             nan     0.1000    0.0002
##     80        1.0395             nan     0.1000    0.0001
##    100        1.0342             nan     0.1000    0.0001
##    120        1.0289             nan     0.1000    0.0001
##    140        1.0249             nan     0.1000    0.0001
##    150        1.0232             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3532             nan     0.1000    0.0165
##      2        1.3266             nan     0.1000    0.0134
##      3        1.3043             nan     0.1000    0.0112
##      4        1.2826             nan     0.1000    0.0108
##      5        1.2649             nan     0.1000    0.0087
```

```
##      6        1.2483             nan     0.1000    0.0084
##      7        1.2340             nan     0.1000    0.0071
##      8        1.2206             nan     0.1000    0.0067
##      9        1.2090             nan     0.1000    0.0057
##     10        1.1986             nan     0.1000    0.0052
##     20        1.1350             nan     0.1000    0.0020
##     40        1.0929             nan     0.1000    0.0005
##     60        1.0775             nan     0.1000    0.0002
##     80        1.0695             nan     0.1000    0.0002
##    100        1.0641             nan     0.1000    0.0001
##    120        1.0600             nan     0.1000    0.0001
##    140        1.0568             nan     0.1000    0.0000
##    150        1.0555             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3440             nan     0.1000    0.0211
##      2        1.3097             nan     0.1000    0.0171
##      3        1.2798             nan     0.1000    0.0150
##      4        1.2554             nan     0.1000    0.0123
##      5        1.2345             nan     0.1000    0.0105
##      6        1.2141             nan     0.1000    0.0103
##      7        1.1984             nan     0.1000    0.0079
##      8        1.1835             nan     0.1000    0.0073
##      9        1.1713             nan     0.1000    0.0060
##     10        1.1611             nan     0.1000    0.0051
##     20        1.1029             nan     0.1000    0.0019
##     40        1.0690             nan     0.1000    0.0004
##     60        1.0569             nan     0.1000    0.0002
##     80        1.0494             nan     0.1000    0.0001
##    100        1.0441             nan     0.1000    0.0001
##    120        1.0398             nan     0.1000    0.0001
##    140        1.0366             nan     0.1000    0.0001
##    150        1.0350             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3387             nan     0.1000    0.0237
##      2        1.2997             nan     0.1000    0.0194
##      3        1.2676             nan     0.1000    0.0160
##      4        1.2411             nan     0.1000    0.0131
##      5        1.2190             nan     0.1000    0.0110
##      6        1.2006             nan     0.1000    0.0093
##      7        1.1847             nan     0.1000    0.0080
##      8        1.1696             nan     0.1000    0.0075
##      9        1.1579             nan     0.1000    0.0058
##     10        1.1478             nan     0.1000    0.0050
##     20        1.0921             nan     0.1000    0.0013
##     40        1.0585             nan     0.1000    0.0004
##     60        1.0456             nan     0.1000    0.0002
##     80        1.0384             nan     0.1000    0.0002
##    100        1.0327             nan     0.1000    0.0001
##    120        1.0281             nan     0.1000    0.0001
##    140        1.0235             nan     0.1000    0.0001
##    150        1.0217             nan     0.1000    0.0001
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3538             nan     0.1000    0.0164
##      2        1.3272             nan     0.1000    0.0135
##      3        1.3048             nan     0.1000    0.0111
##      4        1.2832             nan     0.1000    0.0108
##      5        1.2657             nan     0.1000    0.0087
##      6        1.2489             nan     0.1000    0.0084
##      7        1.2346             nan     0.1000    0.0071
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2102             nan     0.1000    0.0057
##     10        1.1996             nan     0.1000    0.0053
##     20        1.1358             nan     0.1000    0.0020
##     40        1.0937             nan     0.1000    0.0005
##     60        1.0781             nan     0.1000    0.0003
##     80        1.0701             nan     0.1000    0.0001
##    100        1.0648             nan     0.1000    0.0001
##    120        1.0609             nan     0.1000    0.0001
##    140        1.0578             nan     0.1000    0.0000
##    150        1.0564             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3447             nan     0.1000    0.0211
##      2        1.3102             nan     0.1000    0.0171
##      3        1.2804             nan     0.1000    0.0150
##      4        1.2559             nan     0.1000    0.0123
##      5        1.2326             nan     0.1000    0.0116
##      6        1.2144             nan     0.1000    0.0091
##      7        1.1992             nan     0.1000    0.0074
##      8        1.1840             nan     0.1000    0.0075
##      9        1.1722             nan     0.1000    0.0058
##     10        1.1606             nan     0.1000    0.0057
##     20        1.1037             nan     0.1000    0.0016
##     40        1.0704             nan     0.1000    0.0004
##     60        1.0574             nan     0.1000    0.0002
##     80        1.0496             nan     0.1000    0.0002
##    100        1.0445             nan     0.1000    0.0001
##    120        1.0403             nan     0.1000    0.0001
##    140        1.0370             nan     0.1000    0.0001
##    150        1.0355             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3391             nan     0.1000    0.0236
##      2        1.2997             nan     0.1000    0.0195
##      3        1.2678             nan     0.1000    0.0159
##      4        1.2413             nan     0.1000    0.0132
##      5        1.2192             nan     0.1000    0.0109
##      6        1.2008             nan     0.1000    0.0093
##      7        1.1850             nan     0.1000    0.0079
##      8        1.1700             nan     0.1000    0.0075
##      9        1.1583             nan     0.1000    0.0058
##     10        1.1475             nan     0.1000    0.0054
##     20        1.0924             nan     0.1000    0.0014
##     40        1.0596             nan     0.1000    0.0004
##     60        1.0465             nan     0.1000    0.0002
```

```
##     80        1.0394             nan     0.1000    0.0001
##    100        1.0340             nan     0.1000    0.0001
##    120        1.0292             nan     0.1000    0.0001
##    140        1.0248             nan     0.1000    0.0001
##    150        1.0228             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3532             nan     0.1000    0.0166
##      2        1.3259             nan     0.1000    0.0135
##      3        1.3037             nan     0.1000    0.0112
##      4        1.2825             nan     0.1000    0.0106
##      5        1.2649             nan     0.1000    0.0087
##      6        1.2483             nan     0.1000    0.0083
##      7        1.2340             nan     0.1000    0.0071
##      8        1.2210             nan     0.1000    0.0065
##      9        1.2094             nan     0.1000    0.0058
##     10        1.1990             nan     0.1000    0.0052
##     20        1.1352             nan     0.1000    0.0022
##     40        1.0935             nan     0.1000    0.0006
##     60        1.0783             nan     0.1000    0.0002
##     80        1.0704             nan     0.1000    0.0001
##    100        1.0650             nan     0.1000    0.0001
##    120        1.0610             nan     0.1000    0.0001
##    140        1.0577             nan     0.1000    0.0001
##    150        1.0563             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3441             nan     0.1000    0.0211
##      2        1.3103             nan     0.1000    0.0171
##      3        1.2802             nan     0.1000    0.0150
##      4        1.2555             nan     0.1000    0.0123
##      5        1.2351             nan     0.1000    0.0101
##      6        1.2146             nan     0.1000    0.0103
##      7        1.1989             nan     0.1000    0.0079
##      8        1.1842             nan     0.1000    0.0073
##      9        1.1727             nan     0.1000    0.0058
##     10        1.1614             nan     0.1000    0.0055
##     20        1.1041             nan     0.1000    0.0018
##     40        1.0700             nan     0.1000    0.0005
##     60        1.0577             nan     0.1000    0.0002
##     80        1.0502             nan     0.1000    0.0001
##    100        1.0448             nan     0.1000    0.0001
##    120        1.0406             nan     0.1000    0.0001
##    140        1.0373             nan     0.1000    0.0001
##    150        1.0356             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0237
##      2        1.3002             nan     0.1000    0.0195
##      3        1.2683             nan     0.1000    0.0160
##      4        1.2416             nan     0.1000    0.0133
##      5        1.2200             nan     0.1000    0.0109
##      6        1.2014             nan     0.1000    0.0093
##      7        1.1842             nan     0.1000    0.0085
```

```
##      8        1.1702             nan     0.1000    0.0068
##      9        1.1582             nan     0.1000    0.0060
##     10        1.1474             nan     0.1000    0.0053
##     20        1.0918             nan     0.1000    0.0012
##     40        1.0592             nan     0.1000    0.0004
##     60        1.0462             nan     0.1000    0.0001
##     80        1.0390             nan     0.1000    0.0001
##    100        1.0338             nan     0.1000    0.0001
##    120        1.0291             nan     0.1000    0.0001
##    140        1.0246             nan     0.1000    0.0001
##    150        1.0227             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3531             nan     0.1000    0.0164
##      2        1.3262             nan     0.1000    0.0135
##      3        1.3039             nan     0.1000    0.0112
##      4        1.2827             nan     0.1000    0.0106
##      5        1.2651             nan     0.1000    0.0086
##      6        1.2483             nan     0.1000    0.0083
##      7        1.2338             nan     0.1000    0.0071
##      8        1.2208             nan     0.1000    0.0065
##      9        1.2096             nan     0.1000    0.0057
##     10        1.1994             nan     0.1000    0.0051
##     20        1.1367             nan     0.1000    0.0019
##     40        1.0943             nan     0.1000    0.0005
##     60        1.0789             nan     0.1000    0.0002
##     80        1.0709             nan     0.1000    0.0001
##    100        1.0656             nan     0.1000    0.0001
##    120        1.0616             nan     0.1000    0.0001
##    140        1.0584             nan     0.1000    0.0000
##    150        1.0571             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3443             nan     0.1000    0.0210
##      2        1.3083             nan     0.1000    0.0179
##      3        1.2791             nan     0.1000    0.0146
##      4        1.2556             nan     0.1000    0.0117
##      5        1.2357             nan     0.1000    0.0100
##      6        1.2148             nan     0.1000    0.0104
##      7        1.1990             nan     0.1000    0.0079
##      8        1.1854             nan     0.1000    0.0067
##      9        1.1724             nan     0.1000    0.0064
##     10        1.1625             nan     0.1000    0.0050
##     20        1.1050             nan     0.1000    0.0019
##     40        1.0713             nan     0.1000    0.0004
##     60        1.0587             nan     0.1000    0.0003
##     80        1.0511             nan     0.1000    0.0001
##    100        1.0459             nan     0.1000    0.0002
##    120        1.0421             nan     0.1000    0.0000
##    140        1.0382             nan     0.1000    0.0000
##    150        1.0367             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3388             nan     0.1000    0.0235
```

```
##      2        1.2999             nan     0.1000    0.0195
##      3        1.2687             nan     0.1000    0.0159
##      4        1.2423             nan     0.1000    0.0132
##      5        1.2205             nan     0.1000    0.0109
##      6        1.2006             nan     0.1000    0.0099
##      7        1.1849             nan     0.1000    0.0077
##      8        1.1701             nan     0.1000    0.0073
##      9        1.1571             nan     0.1000    0.0063
##     10        1.1468             nan     0.1000    0.0050
##     20        1.0923             nan     0.1000    0.0016
##     40        1.0601             nan     0.1000    0.0005
##     60        1.0470             nan     0.1000    0.0003
##     80        1.0396             nan     0.1000    0.0001
##    100        1.0336             nan     0.1000    0.0001
##    120        1.0289             nan     0.1000    0.0001
##    140        1.0247             nan     0.1000    0.0001
##    150        1.0229             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3535             nan     0.1000    0.0165
##      2        1.3268             nan     0.1000    0.0134
##      3        1.3042             nan     0.1000    0.0111
##      4        1.2828             nan     0.1000    0.0107
##      5        1.2654             nan     0.1000    0.0087
##      6        1.2484             nan     0.1000    0.0084
##      7        1.2343             nan     0.1000    0.0071
##      8        1.2214             nan     0.1000    0.0066
##      9        1.2100             nan     0.1000    0.0057
##     10        1.1997             nan     0.1000    0.0051
##     20        1.1357             nan     0.1000    0.0023
##     40        1.0939             nan     0.1000    0.0005
##     60        1.0783             nan     0.1000    0.0003
##     80        1.0706             nan     0.1000    0.0001
##    100        1.0652             nan     0.1000    0.0001
##    120        1.0612             nan     0.1000    0.0001
##    140        1.0578             nan     0.1000    0.0000
##    150        1.0564             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0210
##      2        1.3096             nan     0.1000    0.0174
##      3        1.2800             nan     0.1000    0.0149
##      4        1.2564             nan     0.1000    0.0117
##      5        1.2337             nan     0.1000    0.0113
##      6        1.2160             nan     0.1000    0.0087
##      7        1.1984             nan     0.1000    0.0088
##      8        1.1849             nan     0.1000    0.0067
##      9        1.1724             nan     0.1000    0.0062
##     10        1.1627             nan     0.1000    0.0048
##     20        1.1039             nan     0.1000    0.0016
##     40        1.0702             nan     0.1000    0.0004
##     60        1.0576             nan     0.1000    0.0002
##     80        1.0500             nan     0.1000    0.0001
##    100        1.0448             nan     0.1000    0.0001
```

```
##    120        1.0406             nan     0.1000    0.0001
##    140        1.0375             nan     0.1000    0.0001
##    150        1.0358             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0235
##      2        1.3001             nan     0.1000    0.0194
##      3        1.2682             nan     0.1000    0.0160
##      4        1.2413             nan     0.1000    0.0133
##      5        1.2195             nan     0.1000    0.0109
##      6        1.2008             nan     0.1000    0.0093
##      7        1.1838             nan     0.1000    0.0085
##      8        1.1702             nan     0.1000    0.0068
##      9        1.1579             nan     0.1000    0.0061
##     10        1.1470             nan     0.1000    0.0053
##     20        1.0924             nan     0.1000    0.0016
##     40        1.0595             nan     0.1000    0.0003
##     60        1.0458             nan     0.1000    0.0002
##     80        1.0381             nan     0.1000    0.0001
##    100        1.0326             nan     0.1000    0.0001
##    120        1.0276             nan     0.1000    0.0002
##    140        1.0241             nan     0.1000    0.0000
##    150        1.0219             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3535             nan     0.1000    0.0165
##      2        1.3267             nan     0.1000    0.0134
##      3        1.3043             nan     0.1000    0.0112
##      4        1.2834             nan     0.1000    0.0106
##      5        1.2656             nan     0.1000    0.0087
##      6        1.2488             nan     0.1000    0.0084
##      7        1.2346             nan     0.1000    0.0071
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2098             nan     0.1000    0.0057
##     10        1.1993             nan     0.1000    0.0052
##     20        1.1360             nan     0.1000    0.0020
##     40        1.0935             nan     0.1000    0.0005
##     60        1.0779             nan     0.1000    0.0003
##     80        1.0702             nan     0.1000    0.0001
##    100        1.0648             nan     0.1000    0.0001
##    120        1.0607             nan     0.1000    0.0000
##    140        1.0575             nan     0.1000    0.0000
##    150        1.0561             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3444             nan     0.1000    0.0210
##      2        1.3104             nan     0.1000    0.0169
##      3        1.2802             nan     0.1000    0.0150
##      4        1.2555             nan     0.1000    0.0124
##      5        1.2351             nan     0.1000    0.0100
##      6        1.2169             nan     0.1000    0.0091
##      7        1.1992             nan     0.1000    0.0088
##      8        1.1851             nan     0.1000    0.0069
##      9        1.1726             nan     0.1000    0.0062
```

```
##     10        1.1613             nan     0.1000    0.0056
##     20        1.1042             nan     0.1000    0.0019
##     40        1.0702             nan     0.1000    0.0005
##     60        1.0577             nan     0.1000    0.0002
##     80        1.0498             nan     0.1000    0.0001
##    100        1.0447             nan     0.1000    0.0001
##    120        1.0407             nan     0.1000    0.0001
##    140        1.0374             nan     0.1000    0.0000
##    150        1.0357             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3387             nan     0.1000    0.0237
##      2        1.2996             nan     0.1000    0.0195
##      3        1.2679             nan     0.1000    0.0158
##      4        1.2418             nan     0.1000    0.0132
##      5        1.2199             nan     0.1000    0.0109
##      6        1.2014             nan     0.1000    0.0092
##      7        1.1842             nan     0.1000    0.0085
##      8        1.1703             nan     0.1000    0.0070
##      9        1.1584             nan     0.1000    0.0058
##     10        1.1478             nan     0.1000    0.0052
##     20        1.0922             nan     0.1000    0.0015
##     40        1.0591             nan     0.1000    0.0004
##     60        1.0465             nan     0.1000    0.0003
##     80        1.0389             nan     0.1000    0.0001
##    100        1.0334             nan     0.1000    0.0001
##    120        1.0288             nan     0.1000    0.0001
##    140        1.0249             nan     0.1000    0.0000
##    150        1.0229             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3536             nan     0.1000    0.0165
##      2        1.3268             nan     0.1000    0.0135
##      3        1.3043             nan     0.1000    0.0113
##      4        1.2833             nan     0.1000    0.0106
##      5        1.2652             nan     0.1000    0.0088
##      6        1.2486             nan     0.1000    0.0084
##      7        1.2346             nan     0.1000    0.0070
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2100             nan     0.1000    0.0058
##     10        1.1995             nan     0.1000    0.0052
##     20        1.1357             nan     0.1000    0.0022
##     40        1.0937             nan     0.1000    0.0006
##     60        1.0781             nan     0.1000    0.0003
##     80        1.0703             nan     0.1000    0.0001
##    100        1.0648             nan     0.1000    0.0001
##    120        1.0609             nan     0.1000    0.0000
##    140        1.0577             nan     0.1000    0.0001
##    150        1.0564             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3441             nan     0.1000    0.0210
##      2        1.3086             nan     0.1000    0.0180
##      3        1.2803             nan     0.1000    0.0141
```

```
##      4        1.2550             nan     0.1000    0.0126
##      5        1.2347             nan     0.1000    0.0102
##      6        1.2171             nan     0.1000    0.0087
##      7        1.2000             nan     0.1000    0.0086
##      8        1.1846             nan     0.1000    0.0076
##      9        1.1727             nan     0.1000    0.0059
##     10        1.1611             nan     0.1000    0.0057
##     20        1.1029             nan     0.1000    0.0019
##     40        1.0699             nan     0.1000    0.0004
##     60        1.0572             nan     0.1000    0.0002
##     80        1.0496             nan     0.1000    0.0001
##    100        1.0444             nan     0.1000    0.0001
##    120        1.0404             nan     0.1000    0.0000
##    140        1.0368             nan     0.1000    0.0000
##    150        1.0353             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3388             nan     0.1000    0.0237
##      2        1.3001             nan     0.1000    0.0194
##      3        1.2680             nan     0.1000    0.0159
##      4        1.2420             nan     0.1000    0.0130
##      5        1.2204             nan     0.1000    0.0108
##      6        1.2018             nan     0.1000    0.0093
##      7        1.1861             nan     0.1000    0.0079
##      8        1.1711             nan     0.1000    0.0074
##      9        1.1584             nan     0.1000    0.0062
##     10        1.1478             nan     0.1000    0.0052
##     20        1.0926             nan     0.1000    0.0014
##     40        1.0596             nan     0.1000    0.0005
##     60        1.0466             nan     0.1000    0.0002
##     80        1.0394             nan     0.1000    0.0001
##    100        1.0336             nan     0.1000    0.0001
##    120        1.0282             nan     0.1000    0.0001
##    140        1.0244             nan     0.1000    0.0001
##    150        1.0225             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3531             nan     0.1000    0.0166
##      2        1.3259             nan     0.1000    0.0136
##      3        1.3034             nan     0.1000    0.0111
##      4        1.2818             nan     0.1000    0.0107
##      5        1.2639             nan     0.1000    0.0088
##      6        1.2471             nan     0.1000    0.0083
##      7        1.2329             nan     0.1000    0.0071
##      8        1.2198             nan     0.1000    0.0065
##      9        1.2082             nan     0.1000    0.0057
##     10        1.1980             nan     0.1000    0.0052
##     20        1.1345             nan     0.1000    0.0020
##     40        1.0926             nan     0.1000    0.0006
##     60        1.0773             nan     0.1000    0.0003
##     80        1.0694             nan     0.1000    0.0002
##    100        1.0640             nan     0.1000    0.0001
##    120        1.0599             nan     0.1000    0.0001
##    140        1.0566             nan     0.1000    0.0001
```

```
##    150        1.0553             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3441             nan     0.1000    0.0211
##      2        1.3098             nan     0.1000    0.0170
##      3        1.2793             nan     0.1000    0.0151
##      4        1.2545             nan     0.1000    0.0123
##      5        1.2312             nan     0.1000    0.0116
##      6        1.2132             nan     0.1000    0.0091
##      7        1.1971             nan     0.1000    0.0080
##      8        1.1833             nan     0.1000    0.0069
##      9        1.1718             nan     0.1000    0.0058
##     10        1.1606             nan     0.1000    0.0056
##     20        1.1021             nan     0.1000    0.0018
##     40        1.0692             nan     0.1000    0.0004
##     60        1.0563             nan     0.1000    0.0002
##     80        1.0487             nan     0.1000    0.0001
##    100        1.0433             nan     0.1000    0.0001
##    120        1.0392             nan     0.1000    0.0001
##    140        1.0356             nan     0.1000    0.0001
##    150        1.0339             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0238
##      2        1.3006             nan     0.1000    0.0194
##      3        1.2677             nan     0.1000    0.0162
##      4        1.2416             nan     0.1000    0.0132
##      5        1.2198             nan     0.1000    0.0110
##      6        1.2006             nan     0.1000    0.0095
##      7        1.1846             nan     0.1000    0.0080
##      8        1.1708             nan     0.1000    0.0068
##      9        1.1576             nan     0.1000    0.0066
##     10        1.1467             nan     0.1000    0.0054
##     20        1.0910             nan     0.1000    0.0016
##     40        1.0580             nan     0.1000    0.0004
##     60        1.0451             nan     0.1000    0.0002
##     80        1.0379             nan     0.1000    0.0001
##    100        1.0324             nan     0.1000    0.0001
##    120        1.0272             nan     0.1000    0.0000
##    140        1.0230             nan     0.1000    0.0001
##    150        1.0211             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0164
##      2        1.3266             nan     0.1000    0.0133
##      3        1.3045             nan     0.1000    0.0110
##      4        1.2832             nan     0.1000    0.0107
##      5        1.2657             nan     0.1000    0.0087
##      6        1.2490             nan     0.1000    0.0083
##      7        1.2346             nan     0.1000    0.0070
##      8        1.2212             nan     0.1000    0.0065
##      9        1.2099             nan     0.1000    0.0056
##     10        1.1995             nan     0.1000    0.0051
##     20        1.1370             nan     0.1000    0.0022
```

```
##     40        1.0952             nan     0.1000    0.0005
##     60        1.0798             nan     0.1000    0.0003
##     80        1.0722             nan     0.1000    0.0001
##    100        1.0669             nan     0.1000    0.0001
##    120        1.0629             nan     0.1000    0.0001
##    140        1.0598             nan     0.1000    0.0000
##    150        1.0584             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0209
##      2        1.3084             nan     0.1000    0.0179
##      3        1.2802             nan     0.1000    0.0142
##      4        1.2565             nan     0.1000    0.0118
##      5        1.2352             nan     0.1000    0.0106
##      6        1.2150             nan     0.1000    0.0101
##      7        1.2004             nan     0.1000    0.0072
##      8        1.1866             nan     0.1000    0.0068
##      9        1.1739             nan     0.1000    0.0064
##     10        1.1628             nan     0.1000    0.0054
##     20        1.1053             nan     0.1000    0.0016
##     40        1.0718             nan     0.1000    0.0004
##     60        1.0592             nan     0.1000    0.0004
##     80        1.0519             nan     0.1000    0.0001
##    100        1.0465             nan     0.1000    0.0001
##    120        1.0425             nan     0.1000    0.0001
##    140        1.0392             nan     0.1000    0.0000
##    150        1.0377             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3392             nan     0.1000    0.0235
##      2        1.3009             nan     0.1000    0.0191
##      3        1.2689             nan     0.1000    0.0161
##      4        1.2424             nan     0.1000    0.0132
##      5        1.2207             nan     0.1000    0.0109
##      6        1.2017             nan     0.1000    0.0095
##      7        1.1858             nan     0.1000    0.0079
##      8        1.1713             nan     0.1000    0.0073
##      9        1.1589             nan     0.1000    0.0061
##     10        1.1488             nan     0.1000    0.0050
##     20        1.0939             nan     0.1000    0.0016
##     40        1.0610             nan     0.1000    0.0004
##     60        1.0483             nan     0.1000    0.0003
##     80        1.0407             nan     0.1000    0.0002
##    100        1.0351             nan     0.1000    0.0001
##    120        1.0305             nan     0.1000    0.0001
##    140        1.0265             nan     0.1000    0.0001
##    150        1.0245             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0164
##      2        1.3267             nan     0.1000    0.0134
##      3        1.3044             nan     0.1000    0.0111
##      4        1.2830             nan     0.1000    0.0107
##      5        1.2653             nan     0.1000    0.0087
```

```
##      6        1.2489             nan     0.1000    0.0082
##      7        1.2345             nan     0.1000    0.0071
##      8        1.2215             nan     0.1000    0.0065
##      9        1.2100             nan     0.1000    0.0056
##     10        1.1997             nan     0.1000    0.0051
##     20        1.1365             nan     0.1000    0.0021
##     40        1.0950             nan     0.1000    0.0006
##     60        1.0798             nan     0.1000    0.0003
##     80        1.0719             nan     0.1000    0.0001
##    100        1.0665             nan     0.1000    0.0001
##    120        1.0624             nan     0.1000    0.0001
##    140        1.0591             nan     0.1000    0.0001
##    150        1.0577             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3445             nan     0.1000    0.0210
##      2        1.3086             nan     0.1000    0.0179
##      3        1.2808             nan     0.1000    0.0140
##      4        1.2575             nan     0.1000    0.0116
##      5        1.2354             nan     0.1000    0.0109
##      6        1.2149             nan     0.1000    0.0101
##      7        1.1998             nan     0.1000    0.0075
##      8        1.1866             nan     0.1000    0.0066
##      9        1.1740             nan     0.1000    0.0064
##     10        1.1632             nan     0.1000    0.0054
##     20        1.1047             nan     0.1000    0.0015
##     40        1.0716             nan     0.1000    0.0005
##     60        1.0594             nan     0.1000    0.0002
##     80        1.0519             nan     0.1000    0.0002
##    100        1.0463             nan     0.1000    0.0001
##    120        1.0419             nan     0.1000    0.0001
##    140        1.0386             nan     0.1000    0.0000
##    150        1.0371             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0235
##      2        1.3006             nan     0.1000    0.0192
##      3        1.2684             nan     0.1000    0.0161
##      4        1.2422             nan     0.1000    0.0130
##      5        1.2205             nan     0.1000    0.0108
##      6        1.2017             nan     0.1000    0.0094
##      7        1.1843             nan     0.1000    0.0086
##      8        1.1708             nan     0.1000    0.0066
##      9        1.1593             nan     0.1000    0.0058
##     10        1.1486             nan     0.1000    0.0054
##     20        1.0943             nan     0.1000    0.0013
##     40        1.0616             nan     0.1000    0.0003
##     60        1.0487             nan     0.1000    0.0002
##     80        1.0411             nan     0.1000    0.0001
##    100        1.0353             nan     0.1000    0.0001
##    120        1.0304             nan     0.1000    0.0001
##    140        1.0266             nan     0.1000    0.0001
##    150        1.0247             nan     0.1000    0.0001
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3533             nan     0.1000    0.0165
##      2        1.3264             nan     0.1000    0.0135
##      3        1.3039             nan     0.1000    0.0112
##      4        1.2824             nan     0.1000    0.0108
##      5        1.2650             nan     0.1000    0.0087
##      6        1.2482             nan     0.1000    0.0084
##      7        1.2338             nan     0.1000    0.0071
##      8        1.2205             nan     0.1000    0.0066
##      9        1.2093             nan     0.1000    0.0057
##     10        1.1989             nan     0.1000    0.0052
##     20        1.1353             nan     0.1000    0.0019
##     40        1.0933             nan     0.1000    0.0007
##     60        1.0781             nan     0.1000    0.0002
##     80        1.0700             nan     0.1000    0.0002
##    100        1.0646             nan     0.1000    0.0001
##    120        1.0607             nan     0.1000    0.0001
##    140        1.0575             nan     0.1000    0.0001
##    150        1.0561             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3438             nan     0.1000    0.0211
##      2        1.3077             nan     0.1000    0.0180
##      3        1.2794             nan     0.1000    0.0141
##      4        1.2550             nan     0.1000    0.0122
##      5        1.2348             nan     0.1000    0.0100
##      6        1.2150             nan     0.1000    0.0098
##      7        1.1997             nan     0.1000    0.0076
##      8        1.1844             nan     0.1000    0.0075
##      9        1.1725             nan     0.1000    0.0059
##     10        1.1618             nan     0.1000    0.0053
##     20        1.1036             nan     0.1000    0.0016
##     40        1.0697             nan     0.1000    0.0005
##     60        1.0568             nan     0.1000    0.0002
##     80        1.0495             nan     0.1000    0.0001
##    100        1.0440             nan     0.1000    0.0001
##    120        1.0404             nan     0.1000    0.0000
##    140        1.0369             nan     0.1000    0.0000
##    150        1.0351             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0237
##      2        1.3002             nan     0.1000    0.0193
##      3        1.2680             nan     0.1000    0.0162
##      4        1.2412             nan     0.1000    0.0133
##      5        1.2193             nan     0.1000    0.0109
##      6        1.2008             nan     0.1000    0.0093
##      7        1.1837             nan     0.1000    0.0085
##      8        1.1690             nan     0.1000    0.0072
##      9        1.1573             nan     0.1000    0.0057
##     10        1.1469             nan     0.1000    0.0052
##     20        1.0922             nan     0.1000    0.0013
##     40        1.0586             nan     0.1000    0.0004
##     60        1.0461             nan     0.1000    0.0002
```

```
##     80        1.0381             nan     0.1000    0.0002
##    100        1.0322             nan     0.1000    0.0001
##    120        1.0270             nan     0.1000    0.0001
##    140        1.0230             nan     0.1000    0.0002
##    150        1.0212             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3530             nan     0.1000    0.0164
##      2        1.3263             nan     0.1000    0.0134
##      3        1.3039             nan     0.1000    0.0112
##      4        1.2829             nan     0.1000    0.0105
##      5        1.2654             nan     0.1000    0.0088
##      6        1.2488             nan     0.1000    0.0083
##      7        1.2350             nan     0.1000    0.0071
##      8        1.2218             nan     0.1000    0.0066
##      9        1.2102             nan     0.1000    0.0058
##     10        1.1999             nan     0.1000    0.0050
##     20        1.1358             nan     0.1000    0.0022
##     40        1.0942             nan     0.1000    0.0005
##     60        1.0788             nan     0.1000    0.0002
##     80        1.0707             nan     0.1000    0.0001
##    100        1.0651             nan     0.1000    0.0001
##    120        1.0610             nan     0.1000    0.0001
##    140        1.0577             nan     0.1000    0.0001
##    150        1.0564             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0210
##      2        1.3084             nan     0.1000    0.0180
##      3        1.2790             nan     0.1000    0.0146
##      4        1.2556             nan     0.1000    0.0117
##      5        1.2351             nan     0.1000    0.0103
##      6        1.2155             nan     0.1000    0.0097
##      7        1.1985             nan     0.1000    0.0086
##      8        1.1849             nan     0.1000    0.0069
##      9        1.1726             nan     0.1000    0.0062
##     10        1.1612             nan     0.1000    0.0057
##     20        1.1047             nan     0.1000    0.0017
##     40        1.0705             nan     0.1000    0.0004
##     60        1.0574             nan     0.1000    0.0003
##     80        1.0499             nan     0.1000    0.0001
##    100        1.0448             nan     0.1000    0.0001
##    120        1.0410             nan     0.1000    0.0000
##    140        1.0374             nan     0.1000    0.0001
##    150        1.0358             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3397             nan     0.1000    0.0236
##      2        1.3006             nan     0.1000    0.0195
##      3        1.2686             nan     0.1000    0.0159
##      4        1.2426             nan     0.1000    0.0131
##      5        1.2200             nan     0.1000    0.0111
##      6        1.2001             nan     0.1000    0.0099
##      7        1.1847             nan     0.1000    0.0077
```

```
##      8        1.1699             nan     0.1000    0.0073
##      9        1.1575             nan     0.1000    0.0061
##     10        1.1471             nan     0.1000    0.0052
##     20        1.0919             nan     0.1000    0.0014
##     40        1.0588             nan     0.1000    0.0004
##     60        1.0457             nan     0.1000    0.0002
##     80        1.0383             nan     0.1000    0.0002
##    100        1.0325             nan     0.1000    0.0002
##    120        1.0275             nan     0.1000    0.0001
##    140        1.0237             nan     0.1000    0.0001
##    150        1.0219             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3536             nan     0.1000    0.0164
##      2        1.3267             nan     0.1000    0.0135
##      3        1.3044             nan     0.1000    0.0112
##      4        1.2832             nan     0.1000    0.0105
##      5        1.2657             nan     0.1000    0.0087
##      6        1.2488             nan     0.1000    0.0085
##      7        1.2346             nan     0.1000    0.0070
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2099             nan     0.1000    0.0058
##     10        1.1994             nan     0.1000    0.0052
##     20        1.1362             nan     0.1000    0.0022
##     40        1.0941             nan     0.1000    0.0005
##     60        1.0785             nan     0.1000    0.0003
##     80        1.0704             nan     0.1000    0.0002
##    100        1.0652             nan     0.1000    0.0001
##    120        1.0611             nan     0.1000    0.0001
##    140        1.0579             nan     0.1000    0.0001
##    150        1.0566             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0210
##      2        1.3083             nan     0.1000    0.0179
##      3        1.2802             nan     0.1000    0.0141
##      4        1.2558             nan     0.1000    0.0122
##      5        1.2357             nan     0.1000    0.0100
##      6        1.2165             nan     0.1000    0.0096
##      7        1.1992             nan     0.1000    0.0086
##      8        1.1852             nan     0.1000    0.0070
##      9        1.1727             nan     0.1000    0.0062
##     10        1.1621             nan     0.1000    0.0053
##     20        1.1043             nan     0.1000    0.0016
##     40        1.0705             nan     0.1000    0.0005
##     60        1.0576             nan     0.1000    0.0002
##     80        1.0500             nan     0.1000    0.0001
##    100        1.0443             nan     0.1000    0.0001
##    120        1.0402             nan     0.1000    0.0000
##    140        1.0368             nan     0.1000    0.0000
##    150        1.0353             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3388             nan     0.1000    0.0236
```

```
##      2        1.3007             nan     0.1000    0.0191
##      3        1.2683             nan     0.1000    0.0161
##      4        1.2418             nan     0.1000    0.0133
##      5        1.2197             nan     0.1000    0.0109
##      6        1.2010             nan     0.1000    0.0093
##      7        1.1856             nan     0.1000    0.0076
##      8        1.1707             nan     0.1000    0.0074
##      9        1.1580             nan     0.1000    0.0063
##     10        1.1472             nan     0.1000    0.0052
##     20        1.0920             nan     0.1000    0.0015
##     40        1.0599             nan     0.1000    0.0004
##     60        1.0465             nan     0.1000    0.0003
##     80        1.0392             nan     0.1000    0.0001
##    100        1.0332             nan     0.1000    0.0001
##    120        1.0284             nan     0.1000    0.0001
##    140        1.0242             nan     0.1000    0.0001
##    150        1.0222             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3533             nan     0.1000    0.0165
##      2        1.3264             nan     0.1000    0.0135
##      3        1.3040             nan     0.1000    0.0112
##      4        1.2829             nan     0.1000    0.0106
##      5        1.2652             nan     0.1000    0.0087
##      6        1.2482             nan     0.1000    0.0084
##      7        1.2340             nan     0.1000    0.0071
##      8        1.2208             nan     0.1000    0.0065
##      9        1.2094             nan     0.1000    0.0056
##     10        1.1989             nan     0.1000    0.0052
##     20        1.1351             nan     0.1000    0.0022
##     40        1.0932             nan     0.1000    0.0005
##     60        1.0777             nan     0.1000    0.0002
##     80        1.0697             nan     0.1000    0.0001
##    100        1.0642             nan     0.1000    0.0001
##    120        1.0601             nan     0.1000    0.0001
##    140        1.0568             nan     0.1000    0.0001
##    150        1.0555             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3439             nan     0.1000    0.0211
##      2        1.3080             nan     0.1000    0.0180
##      3        1.2802             nan     0.1000    0.0138
##      4        1.2555             nan     0.1000    0.0122
##      5        1.2329             nan     0.1000    0.0114
##      6        1.2145             nan     0.1000    0.0091
##      7        1.1971             nan     0.1000    0.0085
##      8        1.1836             nan     0.1000    0.0068
##      9        1.1709             nan     0.1000    0.0063
##     10        1.1603             nan     0.1000    0.0053
##     20        1.1040             nan     0.1000    0.0016
##     40        1.0701             nan     0.1000    0.0004
##     60        1.0573             nan     0.1000    0.0002
##     80        1.0497             nan     0.1000    0.0001
##    100        1.0445             nan     0.1000    0.0001
```

```
##    120        1.0403             nan     0.1000    0.0000
##    140        1.0367             nan     0.1000    0.0001
##    150        1.0353             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3392             nan     0.1000    0.0237
##      2        1.3003             nan     0.1000    0.0195
##      3        1.2685             nan     0.1000    0.0159
##      4        1.2423             nan     0.1000    0.0132
##      5        1.2204             nan     0.1000    0.0111
##      6        1.2006             nan     0.1000    0.0099
##      7        1.1837             nan     0.1000    0.0085
##      8        1.1701             nan     0.1000    0.0067
##      9        1.1575             nan     0.1000    0.0063
##     10        1.1473             nan     0.1000    0.0051
##     20        1.0919             nan     0.1000    0.0014
##     40        1.0587             nan     0.1000    0.0004
##     60        1.0457             nan     0.1000    0.0002
##     80        1.0384             nan     0.1000    0.0001
##    100        1.0329             nan     0.1000    0.0001
##    120        1.0282             nan     0.1000    0.0001
##    140        1.0242             nan     0.1000    0.0001
##    150        1.0224             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3533             nan     0.1000    0.0165
##      2        1.3265             nan     0.1000    0.0134
##      3        1.3040             nan     0.1000    0.0113
##      4        1.2827             nan     0.1000    0.0106
##      5        1.2653             nan     0.1000    0.0087
##      6        1.2487             nan     0.1000    0.0083
##      7        1.2345             nan     0.1000    0.0071
##      8        1.2212             nan     0.1000    0.0066
##      9        1.2099             nan     0.1000    0.0058
##     10        1.1995             nan     0.1000    0.0052
##     20        1.1358             nan     0.1000    0.0021
##     40        1.0937             nan     0.1000    0.0005
##     60        1.0783             nan     0.1000    0.0003
##     80        1.0705             nan     0.1000    0.0001
##    100        1.0652             nan     0.1000    0.0001
##    120        1.0613             nan     0.1000    0.0001
##    140        1.0580             nan     0.1000    0.0001
##    150        1.0567             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3438             nan     0.1000    0.0211
##      2        1.3081             nan     0.1000    0.0179
##      3        1.2798             nan     0.1000    0.0141
##      4        1.2552             nan     0.1000    0.0122
##      5        1.2345             nan     0.1000    0.0104
##      6        1.2148             nan     0.1000    0.0098
##      7        1.1977             nan     0.1000    0.0086
##      8        1.1840             nan     0.1000    0.0069
##      9        1.1723             nan     0.1000    0.0058
```

```
##     10        1.1608             nan     0.1000    0.0057
##     20        1.1041             nan     0.1000    0.0013
##     40        1.0702             nan     0.1000    0.0004
##     60        1.0578             nan     0.1000    0.0002
##     80        1.0503             nan     0.1000    0.0001
##    100        1.0449             nan     0.1000    0.0001
##    120        1.0409             nan     0.1000    0.0000
##    140        1.0371             nan     0.1000    0.0001
##    150        1.0355             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3392             nan     0.1000    0.0237
##      2        1.3007             nan     0.1000    0.0194
##      3        1.2681             nan     0.1000    0.0161
##      4        1.2417             nan     0.1000    0.0133
##      5        1.2199             nan     0.1000    0.0109
##      6        1.2012             nan     0.1000    0.0093
##      7        1.1855             nan     0.1000    0.0077
##      8        1.1706             nan     0.1000    0.0074
##      9        1.1579             nan     0.1000    0.0063
##     10        1.1473             nan     0.1000    0.0053
##     20        1.0916             nan     0.1000    0.0016
##     40        1.0600             nan     0.1000    0.0005
##     60        1.0474             nan     0.1000    0.0002
##     80        1.0397             nan     0.1000    0.0002
##    100        1.0339             nan     0.1000    0.0001
##    120        1.0291             nan     0.1000    0.0001
##    140        1.0249             nan     0.1000    0.0000
##    150        1.0231             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3530             nan     0.1000    0.0165
##      2        1.3262             nan     0.1000    0.0135
##      3        1.3040             nan     0.1000    0.0111
##      4        1.2828             nan     0.1000    0.0107
##      5        1.2651             nan     0.1000    0.0088
##      6        1.2481             nan     0.1000    0.0083
##      7        1.2339             nan     0.1000    0.0070
##      8        1.2208             nan     0.1000    0.0066
##      9        1.2095             nan     0.1000    0.0057
##     10        1.1990             nan     0.1000    0.0052
##     20        1.1355             nan     0.1000    0.0020
##     40        1.0938             nan     0.1000    0.0005
##     60        1.0782             nan     0.1000    0.0002
##     80        1.0704             nan     0.1000    0.0001
##    100        1.0649             nan     0.1000    0.0001
##    120        1.0608             nan     0.1000    0.0001
##    140        1.0576             nan     0.1000    0.0001
##    150        1.0561             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3439             nan     0.1000    0.0210
##      2        1.3080             nan     0.1000    0.0179
##      3        1.2798             nan     0.1000    0.0141
```

```
##      4        1.2551             nan     0.1000    0.0122
##      5        1.2343             nan     0.1000    0.0104
##      6        1.2142             nan     0.1000    0.0102
##      7        1.1994             nan     0.1000    0.0074
##      8        1.1847             nan     0.1000    0.0074
##      9        1.1728             nan     0.1000    0.0059
##     10        1.1617             nan     0.1000    0.0055
##     20        1.1037             nan     0.1000    0.0014
##     40        1.0696             nan     0.1000    0.0004
##     60        1.0570             nan     0.1000    0.0002
##     80        1.0494             nan     0.1000    0.0001
##    100        1.0440             nan     0.1000    0.0001
##    120        1.0397             nan     0.1000    0.0001
##    140        1.0360             nan     0.1000    0.0001
##    150        1.0343             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0237
##      2        1.3003             nan     0.1000    0.0194
##      3        1.2685             nan     0.1000    0.0159
##      4        1.2418             nan     0.1000    0.0133
##      5        1.2200             nan     0.1000    0.0109
##      6        1.2012             nan     0.1000    0.0093
##      7        1.1857             nan     0.1000    0.0077
##      8        1.1712             nan     0.1000    0.0072
##      9        1.1591             nan     0.1000    0.0060
##     10        1.1481             nan     0.1000    0.0055
##     20        1.0924             nan     0.1000    0.0014
##     40        1.0591             nan     0.1000    0.0006
##     60        1.0461             nan     0.1000    0.0003
##     80        1.0382             nan     0.1000    0.0001
##    100        1.0325             nan     0.1000    0.0001
##    120        1.0277             nan     0.1000    0.0001
##    140        1.0242             nan     0.1000    0.0001
##    150        1.0222             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3535             nan     0.1000    0.0165
##      2        1.3266             nan     0.1000    0.0135
##      3        1.3042             nan     0.1000    0.0111
##      4        1.2827             nan     0.1000    0.0108
##      5        1.2650             nan     0.1000    0.0087
##      6        1.2483             nan     0.1000    0.0083
##      7        1.2341             nan     0.1000    0.0072
##      8        1.2208             nan     0.1000    0.0066
##      9        1.2094             nan     0.1000    0.0056
##     10        1.1990             nan     0.1000    0.0052
##     20        1.1356             nan     0.1000    0.0022
##     40        1.0938             nan     0.1000    0.0005
##     60        1.0783             nan     0.1000    0.0002
##     80        1.0704             nan     0.1000    0.0001
##    100        1.0649             nan     0.1000    0.0001
##    120        1.0608             nan     0.1000    0.0001
##    140        1.0576             nan     0.1000    0.0000
```

```
##     150        1.0562             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3440             nan     0.1000    0.0210
##      2        1.3099             nan     0.1000    0.0170
##      3        1.2797             nan     0.1000    0.0150
##      4        1.2559             nan     0.1000    0.0120
##      5        1.2352             nan     0.1000    0.0103
##      6        1.2144             nan     0.1000    0.0104
##      7        1.1974             nan     0.1000    0.0085
##      8        1.1844             nan     0.1000    0.0066
##      9        1.1717             nan     0.1000    0.0063
##     10        1.1608             nan     0.1000    0.0054
##     20        1.1043             nan     0.1000    0.0015
##     40        1.0701             nan     0.1000    0.0005
##     60        1.0572             nan     0.1000    0.0002
##     80        1.0494             nan     0.1000    0.0002
##    100        1.0441             nan     0.1000    0.0001
##    120        1.0402             nan     0.1000    0.0000
##    140        1.0365             nan     0.1000    0.0001
##    150        1.0350             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0236
##      2        1.3000             nan     0.1000    0.0195
##      3        1.2683             nan     0.1000    0.0160
##      4        1.2415             nan     0.1000    0.0133
##      5        1.2199             nan     0.1000    0.0108
##      6        1.2011             nan     0.1000    0.0093
##      7        1.1852             nan     0.1000    0.0079
##      8        1.1705             nan     0.1000    0.0075
##      9        1.1582             nan     0.1000    0.0062
##     10        1.1480             nan     0.1000    0.0051
##     20        1.0923             nan     0.1000    0.0016
##     40        1.0593             nan     0.1000    0.0004
##     60        1.0460             nan     0.1000    0.0003
##     80        1.0380             nan     0.1000    0.0002
##    100        1.0325             nan     0.1000    0.0000
##    120        1.0280             nan     0.1000    0.0001
##    140        1.0240             nan     0.1000    0.0001
##    150        1.0223             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3537             nan     0.1000    0.0165
##      2        1.3264             nan     0.1000    0.0135
##      3        1.3039             nan     0.1000    0.0111
##      4        1.2827             nan     0.1000    0.0106
##      5        1.2647             nan     0.1000    0.0088
##      6        1.2481             nan     0.1000    0.0082
##      7        1.2336             nan     0.1000    0.0071
##      8        1.2207             nan     0.1000    0.0065
##      9        1.2092             nan     0.1000    0.0057
##     10        1.1991             nan     0.1000    0.0050
##     20        1.1359             nan     0.1000    0.0021
```

```
##     40        1.0938             nan     0.1000    0.0005
##     60        1.0784             nan     0.1000    0.0003
##     80        1.0703             nan     0.1000    0.0002
##    100        1.0648             nan     0.1000    0.0001
##    120        1.0610             nan     0.1000    0.0001
##    140        1.0578             nan     0.1000    0.0001
##    150        1.0565             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3447             nan     0.1000    0.0209
##      2        1.3105             nan     0.1000    0.0171
##      3        1.2807             nan     0.1000    0.0150
##      4        1.2558             nan     0.1000    0.0125
##      5        1.2350             nan     0.1000    0.0104
##      6        1.2173             nan     0.1000    0.0088
##      7        1.2000             nan     0.1000    0.0087
##      8        1.1854             nan     0.1000    0.0072
##      9        1.1730             nan     0.1000    0.0061
##     10        1.1627             nan     0.1000    0.0051
##     20        1.1041             nan     0.1000    0.0016
##     40        1.0707             nan     0.1000    0.0004
##     60        1.0579             nan     0.1000    0.0002
##     80        1.0505             nan     0.1000    0.0001
##    100        1.0449             nan     0.1000    0.0001
##    120        1.0410             nan     0.1000    0.0001
##    140        1.0376             nan     0.1000    0.0000
##    150        1.0359             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0235
##      2        1.3004             nan     0.1000    0.0195
##      3        1.2687             nan     0.1000    0.0159
##      4        1.2425             nan     0.1000    0.0131
##      5        1.2204             nan     0.1000    0.0111
##      6        1.2012             nan     0.1000    0.0097
##      7        1.1855             nan     0.1000    0.0077
##      8        1.1717             nan     0.1000    0.0069
##      9        1.1599             nan     0.1000    0.0059
##     10        1.1488             nan     0.1000    0.0055
##     20        1.0928             nan     0.1000    0.0016
##     40        1.0609             nan     0.1000    0.0004
##     60        1.0478             nan     0.1000    0.0002
##     80        1.0395             nan     0.1000    0.0002
##    100        1.0336             nan     0.1000    0.0002
##    120        1.0286             nan     0.1000    0.0001
##    140        1.0246             nan     0.1000    0.0001
##    150        1.0226             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3534             nan     0.1000    0.0165
##      2        1.3266             nan     0.1000    0.0133
##      3        1.3042             nan     0.1000    0.0112
##      4        1.2827             nan     0.1000    0.0107
##      5        1.2652             nan     0.1000    0.0086
```

```
##      6        1.2485             nan     0.1000    0.0084
##      7        1.2345             nan     0.1000    0.0071
##      8        1.2216             nan     0.1000    0.0065
##      9        1.2100             nan     0.1000    0.0057
##     10        1.2001             nan     0.1000    0.0050
##     20        1.1359             nan     0.1000    0.0021
##     40        1.0941             nan     0.1000    0.0006
##     60        1.0788             nan     0.1000    0.0003
##     80        1.0710             nan     0.1000    0.0001
##    100        1.0658             nan     0.1000    0.0001
##    120        1.0617             nan     0.1000    0.0001
##    140        1.0586             nan     0.1000    0.0001
##    150        1.0574             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3442             nan     0.1000    0.0210
##      2        1.3093             nan     0.1000    0.0172
##      3        1.2797             nan     0.1000    0.0148
##      4        1.2564             nan     0.1000    0.0116
##      5        1.2340             nan     0.1000    0.0112
##      6        1.2167             nan     0.1000    0.0086
##      7        1.1989             nan     0.1000    0.0089
##      8        1.1856             nan     0.1000    0.0067
##      9        1.1729             nan     0.1000    0.0063
##     10        1.1613             nan     0.1000    0.0058
##     20        1.1052             nan     0.1000    0.0014
##     40        1.0706             nan     0.1000    0.0004
##     60        1.0585             nan     0.1000    0.0002
##     80        1.0509             nan     0.1000    0.0002
##    100        1.0457             nan     0.1000    0.0001
##    120        1.0413             nan     0.1000    0.0001
##    140        1.0381             nan     0.1000    0.0001
##    150        1.0366             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0235
##      2        1.3004             nan     0.1000    0.0192
##      3        1.2683             nan     0.1000    0.0160
##      4        1.2424             nan     0.1000    0.0130
##      5        1.2201             nan     0.1000    0.0111
##      6        1.1995             nan     0.1000    0.0101
##      7        1.1828             nan     0.1000    0.0083
##      8        1.1695             nan     0.1000    0.0066
##      9        1.1575             nan     0.1000    0.0061
##     10        1.1477             nan     0.1000    0.0049
##     20        1.0930             nan     0.1000    0.0016
##     40        1.0606             nan     0.1000    0.0004
##     60        1.0475             nan     0.1000    0.0002
##     80        1.0397             nan     0.1000    0.0002
##    100        1.0342             nan     0.1000    0.0001
##    120        1.0294             nan     0.1000    0.0001
##    140        1.0252             nan     0.1000    0.0001
##    150        1.0233             nan     0.1000    0.0001
##
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3532             nan     0.1000    0.0165
##      2        1.3263             nan     0.1000    0.0134
##      3        1.3038             nan     0.1000    0.0112
##      4        1.2826             nan     0.1000    0.0106
##      5        1.2651             nan     0.1000    0.0088
##      6        1.2484             nan     0.1000    0.0083
##      7        1.2345             nan     0.1000    0.0070
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2096             nan     0.1000    0.0057
##     10        1.1993             nan     0.1000    0.0051
##     20        1.1361             nan     0.1000    0.0021
##     40        1.0942             nan     0.1000    0.0005
##     60        1.0788             nan     0.1000    0.0003
##     80        1.0709             nan     0.1000    0.0001
##    100        1.0656             nan     0.1000    0.0001
##    120        1.0615             nan     0.1000    0.0001
##    140        1.0583             nan     0.1000    0.0001
##    150        1.0569             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3445             nan     0.1000    0.0210
##      2        1.3088             nan     0.1000    0.0179
##      3        1.2807             nan     0.1000    0.0140
##      4        1.2558             nan     0.1000    0.0124
##      5        1.2348             nan     0.1000    0.0104
##      6        1.2151             nan     0.1000    0.0098
##      7        1.2000             nan     0.1000    0.0075
##      8        1.1849             nan     0.1000    0.0075
##      9        1.1733             nan     0.1000    0.0057
##     10        1.1620             nan     0.1000    0.0056
##     20        1.1053             nan     0.1000    0.0014
##     40        1.0712             nan     0.1000    0.0004
##     60        1.0582             nan     0.1000    0.0003
##     80        1.0502             nan     0.1000    0.0001
##    100        1.0453             nan     0.1000    0.0001
##    120        1.0415             nan     0.1000    0.0001
##    140        1.0383             nan     0.1000    0.0001
##    150        1.0365             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3390             nan     0.1000    0.0235
##      2        1.3002             nan     0.1000    0.0193
##      3        1.2687             nan     0.1000    0.0159
##      4        1.2426             nan     0.1000    0.0130
##      5        1.2203             nan     0.1000    0.0111
##      6        1.2017             nan     0.1000    0.0092
##      7        1.1843             nan     0.1000    0.0087
##      8        1.1690             nan     0.1000    0.0074
##      9        1.1569             nan     0.1000    0.0059
##     10        1.1466             nan     0.1000    0.0052
##     20        1.0924             nan     0.1000    0.0018
##     40        1.0605             nan     0.1000    0.0005
##     60        1.0473             nan     0.1000    0.0002
```

```
##     80        1.0396             nan     0.1000    0.0001
##    100        1.0340             nan     0.1000    0.0001
##    120        1.0288             nan     0.1000    0.0001
##    140        1.0248             nan     0.1000    0.0000
##    150        1.0230             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3535             nan     0.1000    0.0165
##      2        1.3267             nan     0.1000    0.0135
##      3        1.3044             nan     0.1000    0.0112
##      4        1.2830             nan     0.1000    0.0106
##      5        1.2655             nan     0.1000    0.0087
##      6        1.2490             nan     0.1000    0.0083
##      7        1.2346             nan     0.1000    0.0071
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2101             nan     0.1000    0.0056
##     10        1.1997             nan     0.1000    0.0052
##     20        1.1363             nan     0.1000    0.0021
##     40        1.0949             nan     0.1000    0.0005
##     60        1.0797             nan     0.1000    0.0002
##     80        1.0719             nan     0.1000    0.0001
##    100        1.0666             nan     0.1000    0.0001
##    120        1.0626             nan     0.1000    0.0001
##    140        1.0593             nan     0.1000    0.0000
##    150        1.0580             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3439             nan     0.1000    0.0210
##      2        1.3078             nan     0.1000    0.0179
##      3        1.2794             nan     0.1000    0.0141
##      4        1.2543             nan     0.1000    0.0124
##      5        1.2338             nan     0.1000    0.0103
##      6        1.2170             nan     0.1000    0.0084
##      7        1.1992             nan     0.1000    0.0090
##      8        1.1856             nan     0.1000    0.0066
##      9        1.1733             nan     0.1000    0.0064
##     10        1.1626             nan     0.1000    0.0053
##     20        1.1055             nan     0.1000    0.0014
##     40        1.0708             nan     0.1000    0.0004
##     60        1.0584             nan     0.1000    0.0002
##     80        1.0507             nan     0.1000    0.0001
##    100        1.0456             nan     0.1000    0.0001
##    120        1.0419             nan     0.1000    0.0001
##    140        1.0385             nan     0.1000    0.0000
##    150        1.0366             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3392             nan     0.1000    0.0237
##      2        1.3003             nan     0.1000    0.0195
##      3        1.2684             nan     0.1000    0.0159
##      4        1.2422             nan     0.1000    0.0131
##      5        1.2202             nan     0.1000    0.0110
##      6        1.2000             nan     0.1000    0.0101
##      7        1.1834             nan     0.1000    0.0081
```

```
##      8        1.1699             nan     0.1000    0.0067
##      9        1.1586             nan     0.1000    0.0057
##     10        1.1488             nan     0.1000    0.0048
##     20        1.0945             nan     0.1000    0.0014
##     40        1.0599             nan     0.1000    0.0004
##     60        1.0474             nan     0.1000    0.0002
##     80        1.0395             nan     0.1000    0.0002
##    100        1.0340             nan     0.1000    0.0002
##    120        1.0291             nan     0.1000    0.0001
##    140        1.0251             nan     0.1000    0.0000
##    150        1.0234             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3532             nan     0.1000    0.0164
##      2        1.3263             nan     0.1000    0.0134
##      3        1.3040             nan     0.1000    0.0112
##      4        1.2828             nan     0.1000    0.0105
##      5        1.2653             nan     0.1000    0.0087
##      6        1.2487             nan     0.1000    0.0083
##      7        1.2345             nan     0.1000    0.0071
##      8        1.2215             nan     0.1000    0.0066
##      9        1.2100             nan     0.1000    0.0057
##     10        1.1996             nan     0.1000    0.0052
##     20        1.1362             nan     0.1000    0.0020
##     40        1.0944             nan     0.1000    0.0005
##     60        1.0791             nan     0.1000    0.0003
##     80        1.0711             nan     0.1000    0.0001
##    100        1.0658             nan     0.1000    0.0001
##    120        1.0617             nan     0.1000    0.0001
##    140        1.0586             nan     0.1000    0.0000
##    150        1.0571             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3446             nan     0.1000    0.0210
##      2        1.3083             nan     0.1000    0.0179
##      3        1.2797             nan     0.1000    0.0142
##      4        1.2558             nan     0.1000    0.0119
##      5        1.2362             nan     0.1000    0.0098
##      6        1.2162             nan     0.1000    0.0100
##      7        1.1984             nan     0.1000    0.0089
##      8        1.1843             nan     0.1000    0.0070
##      9        1.1728             nan     0.1000    0.0058
##     10        1.1621             nan     0.1000    0.0052
##     20        1.1047             nan     0.1000    0.0014
##     40        1.0705             nan     0.1000    0.0004
##     60        1.0584             nan     0.1000    0.0002
##     80        1.0507             nan     0.1000    0.0001
##    100        1.0455             nan     0.1000    0.0001
##    120        1.0415             nan     0.1000    0.0001
##    140        1.0378             nan     0.1000    0.0000
##    150        1.0364             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3394             nan     0.1000    0.0236
```

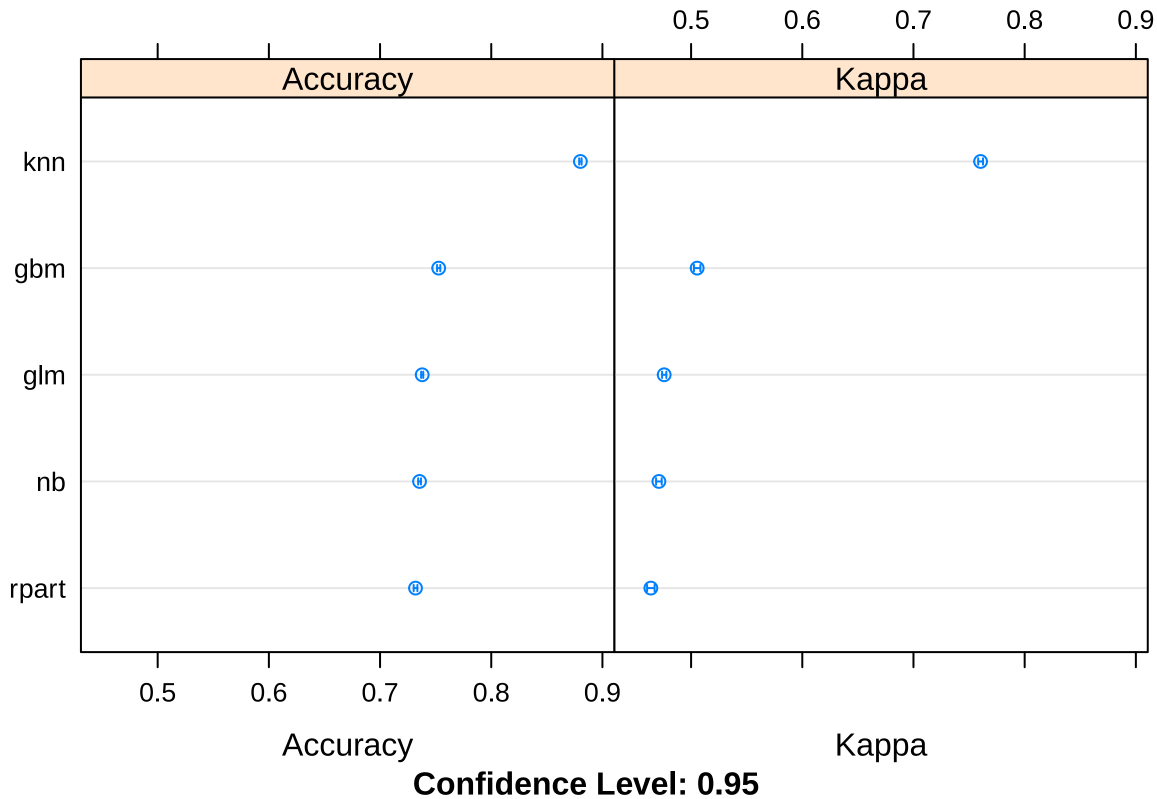
```
##      2        1.3008             nan     0.1000    0.0195
##      3        1.2687             nan     0.1000    0.0160
##      4        1.2425             nan     0.1000    0.0131
##      5        1.2205             nan     0.1000    0.0111
##      6        1.2004             nan     0.1000    0.0100
##      7        1.1849             nan     0.1000    0.0079
##      8        1.1704             nan     0.1000    0.0073
##      9        1.1580             nan     0.1000    0.0061
##     10        1.1474             nan     0.1000    0.0054
##     20        1.0931             nan     0.1000    0.0014
##     40        1.0597             nan     0.1000    0.0004
##     60        1.0469             nan     0.1000    0.0001
##     80        1.0391             nan     0.1000    0.0001
##    100        1.0333             nan     0.1000    0.0001
##    120        1.0285             nan     0.1000    0.0001
##    140        1.0249             nan     0.1000    0.0000
##    150        1.0229             nan     0.1000    0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.3389             nan     0.1000    0.0237
##      2        1.3007             nan     0.1000    0.0192
##      3        1.2686             nan     0.1000    0.0161
##      4        1.2419             nan     0.1000    0.0133
##      5        1.2201             nan     0.1000    0.0109
##      6        1.2014             nan     0.1000    0.0093
##      7        1.1855             nan     0.1000    0.0079
##      8        1.1715             nan     0.1000    0.0069
##      9        1.1588             nan     0.1000    0.0064
##     10        1.1489             nan     0.1000    0.0049
##     20        1.0932             nan     0.1000    0.0014
##     40        1.0605             nan     0.1000    0.0005
##     60        1.0474             nan     0.1000    0.0002
##     80        1.0396             nan     0.1000    0.0001
##    100        1.0341             nan     0.1000    0.0001
##    120        1.0298             nan     0.1000    0.0000
##    140        1.0257             nan     0.1000    0.0001
##    150        1.0235             nan     0.1000    0.0001
```

```r
results <- resamples(ensemble_learning)
summary(results) # summary of all the combined models
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: gbm, rpart, glm, knn, nb
## Number of resamples: 30
##
## Accuracy
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## gbm   0.7468479 0.7497310 0.7522461 0.7527319 0.7556248 0.7629870    0
## rpart 0.7217818 0.7287038 0.7320498 0.7319008 0.7342004 0.7444881    0
## glm   0.7333333 0.7359947 0.7374283 0.7379033 0.7389343 0.7452431    0
## knn   0.8752737 0.8782371 0.8803700 0.8801826 0.8820098 0.8857682    0
## nb    0.7310683 0.7328942 0.7342948 0.7355224 0.7379200 0.7426004    0
```

```
##
## Kappa
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## gbm   0.4936953 0.4994613 0.5044913 0.5054633 0.5112486 0.5259740    0
## rpart 0.4435547 0.4574015 0.4640913 0.4637990 0.4684008 0.4889761    0
## glm   0.4666684 0.4719907 0.4748565 0.4758074 0.4778697 0.4904863    0
## knn   0.7505427 0.7564697 0.7607356 0.7603627 0.7640166 0.7715325    0
## nb    0.4621422 0.4657925 0.4685895 0.4710480 0.4758451 0.4852008    0
```

```r
dotplot(results) # plot to check Kappa and Accuracy differences
```



We can see that the KNN creates the most accurate model with an accuracy of 89%.

```r
stackControl <- trainControl(method="repeatedcv", number=10, repeats=3, savePredictions='final', classP
stack.glm <- caretStack(ensemble_learning, method="glm", metric="Accuracy", trControl=stackControl)
print(stack.glm)
```

```
## A glm ensemble of 2 base models: gbm, rpart, glm, knn, nb
##
## Ensemble results:
## Generalized Linear Model
##
## 397338 samples
##      5 predictor
##      2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
```

```
## Summary of sample sizes: 357604, 357605, 357604, 357604, 357604, 357605, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9706371  0.941274
```

From the model above, we combine the predictions of different models using stacking, it is desirable that the predictions made by the sub-models have low correlation. This would suggest that the models are skillful but in different ways, allowing a new classifier to figure out how to get the best from each model for an improved score.

Let's check these predictions from our training model on our testing set.

```
stacked_pred <- predict(stack.glm, data_oversample_test)
confusionMatrix(stacked_pred,data_oversample_test$SeriousDlqin2yrs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  26907 25249
##        Yes  1472  3082
##
##                Accuracy : 0.5288
##                  95% CI : (0.5247, 0.5329)
##     No Information Rate : 0.5004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.057
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9481
##             Specificity : 0.1088
##          Pos Pred Value : 0.5159
##          Neg Pred Value : 0.6768
##              Prevalence : 0.5004
##          Detection Rate : 0.4745
##    Detection Prevalence : 0.9197
##       Balanced Accuracy : 0.5285
##
##        'Positive' Class : No
##
```

The model didn't perform like I expected. From the combine predictions from the 5 classification models, the accuracyof 52% was lower than all the previous individuals models shown above. We are going to use second model of logistic regression as our final model to submit to the kaggle competition.

Lastly, we write our results to a csv file for kaggle submission.

```
submission <- data.frame(ID = data_oversample_test$X, Serious_Deliquency = os_pred)
head(submission)
```

```
##   ID Serious_Deliquency
## 1  3          0.7467114
## 2  4          0.3642371
## 3 21          0.2042830
## 4 23          0.7318677
```

```
## 5 30          0.7142131
## 6 34          0.2218167
```

```r
write.csv(submission, file = "MySubmission.csv", row.names = F)
```