Stat 412

Yolanda Jin

24/11/2021

NA data (Income NA), Minority, Majority Groups

- Added column for NA_indicator
- split groups to minority vs majority group
- Q1: do we want NA to be a separate group? based on EDA, might or might not do so

```
# Read Credit Scoring Data Training Set
#cs_train <- cs_training
cs_train = read.csv("cs-training.csv")
train <- cs_train
raw_data <- cs_train
cs_train_omit <- na.omit(raw_data)
Predictor_Variables <- subset.data.frame(cs_train_omit, select = c(RevolvingUtilizationOfUnsecuredLines
))
# If we want to split NA to another group
#cs_train_NA <- cs_train[cs_train$MonthlyIncome==NA,]
#head(cs_train_NA)


## 0. NA Monthly Income

# Add col to indicate whether monthly Income is NA or not
cs_train$NA_Indicator <- 0  # Set all NA_Indicator to zero
cs_train$NA_Indicator[is.na(cs_train$MonthlyIncome)] <- 1  # Change NA income indexes to 1
head(cs_train[cs_train$NA_Indicator==1,])  # Check the ones indicated as NA
```

```
##     X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 7   7                0                            0.30568247  57
## 9   9                0                            0.11695064  27
## 17 17                0                            0.06108612  78
## 33 33                0                            0.08341801  62
## 42 42                0                            0.07289757  81
## 53 53                0                            0.99999990  62
##    NumberOfTime30.59DaysPastDueNotWorse DebtRatio MonthlyIncome
## 7                                     0      5710            NA
## 9                                     0        46            NA
## 17                                    0      2058            NA
## 33                                    0       977            NA
## 42                                    0        75            NA
## 53                                    0         0            NA
##    NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 7                                8                       0
```

```
## 9                                2                       0
## 17                              10                       0
## 33                               6                       0
## 42                               7                       0
## 53                               1                       0
##    NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 7                             3                                     0
## 9                             0                                     0
## 17                            2                                     0
## 33                            1                                     0
## 42                            0                                     0
## 53                            0                                     0
##    NumberOfDependents NA_Indicator
## 7                   0            1
## 9                  NA            1
## 17                  0            1
## 33                  0            1
## 42                  0            1
## 53                  0            1
```

```
## 1. Separate minority data vs majority data vs NA data total 150k
cs_train_min <- cs_train[cs_train$SeriousDlqin2yrs==1,]  # 10,026 obs
cs_train_maj <- cs_train[cs_train$SeriousDlqin2yrs==0,]  # 139,974 obs
```

```
summary(cs_train)
```

```
##        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
```

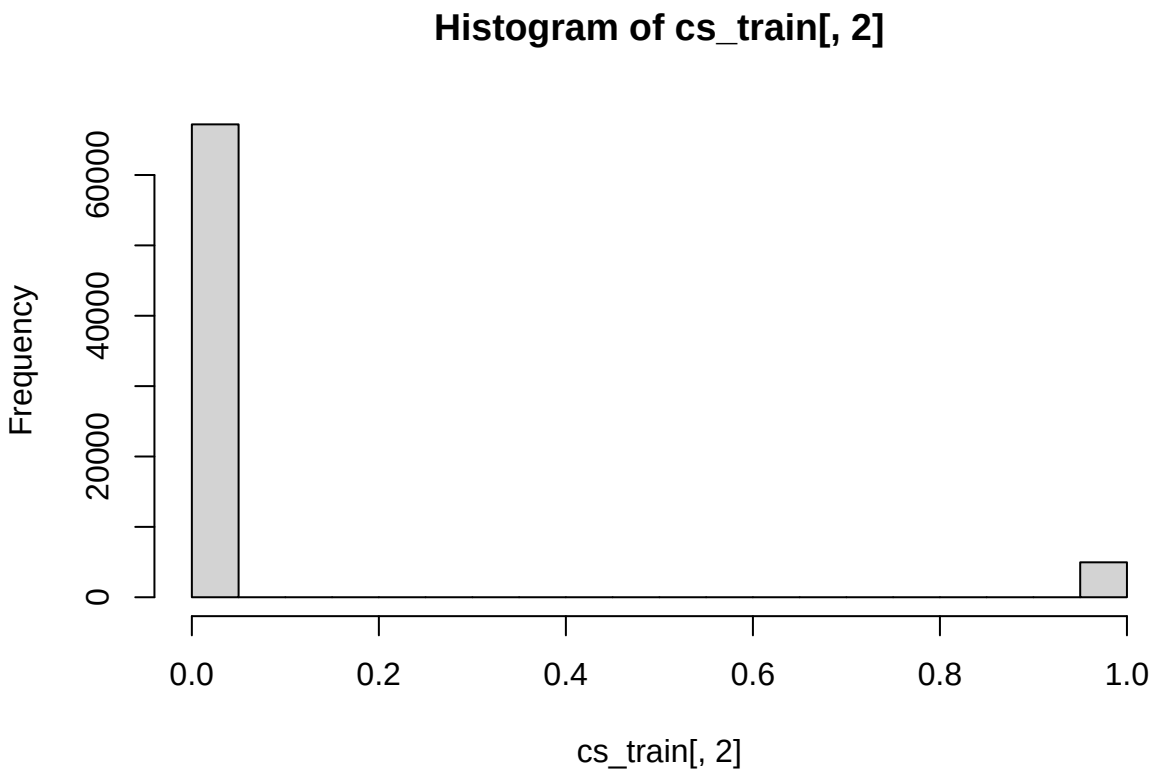
```
##  3rd Qu.: 2.000                3rd Qu.: 0.0000
##  Max.   :54.000                Max.   :98.0000
##
##  NumberOfDependents  NA_Indicator
##  Min.   : 0.000     Min.   :0.0000
##  1st Qu.: 0.000     1st Qu.:0.0000
##  Median : 0.000     Median :0.0000
##  Mean   : 0.757     Mean   :0.1982
##  3rd Qu.: 1.000     3rd Qu.:0.0000
##  Max.   :20.000     Max.   :1.0000
##  NA's   :3924
```

Split the data to Testing/Training

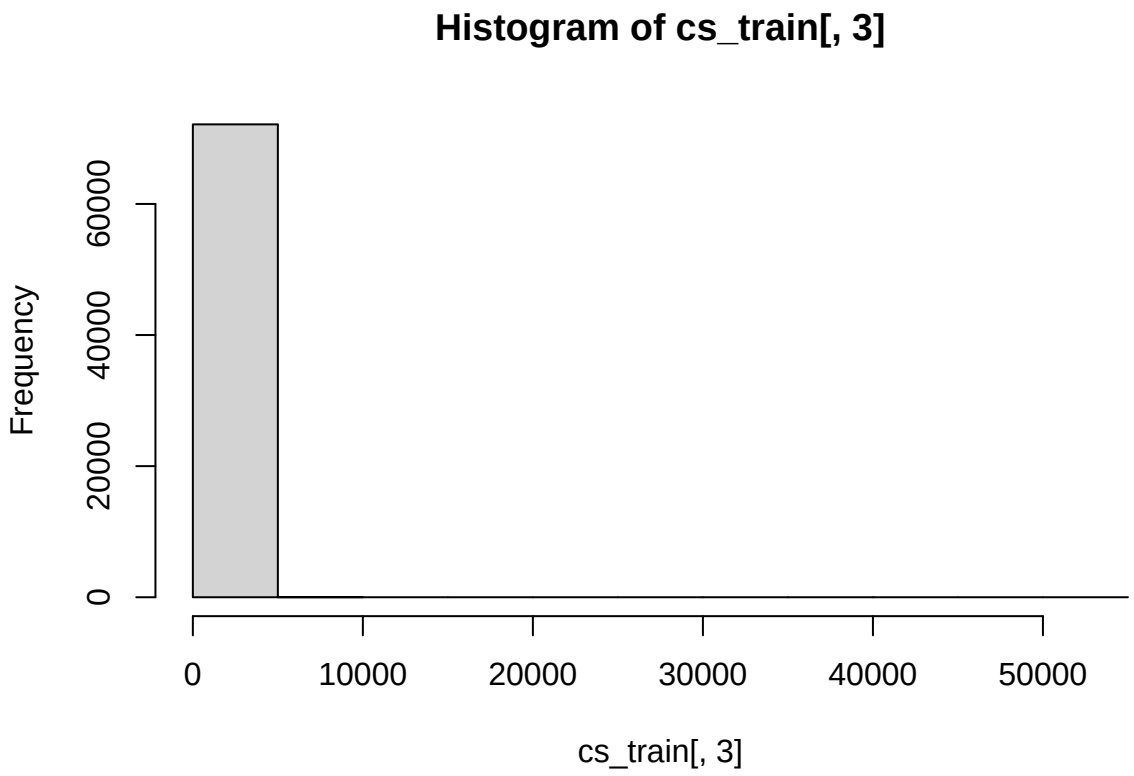
```
# Remove NA cases otherwise cannot predict
#raw_data <- cs_train
cs_train_omit <- na.omit(raw_data)   # 150k -> 120,269 obs

# Sample 60% of data for training Purpose
n = nrow(cs_train_omit)
na.idx = raw_data$X[-cs_train_omit$X]  # indexes of data with NA values that we removed
n.idx = sample(n, n*0.6) # Indexes for test train split

cs_train = cs_train_omit[n.idx,]  # Training data 72,161 obs
cs_test = cs_train_omit[-n.idx,]  # Testing data  48,108 obs.
cs_NA = raw_data[na.idx,]  # data with NA value  29,731 obs
```

```
# Split x and y variables
train.x = cs_train[,-which(names(cs_train) == "SeriousDlqin2yrs")]
train.x = cs_train[,-c(which(names(cs_train) == "SeriousDlqin2yrs"), which(names(cs_train) == "X"), whi
train.y = cs_train$SeriousDlqin2yrs
test.x = cs_test[,-which(names(cs_test) == "SeriousDlqin2yrs")]
test.x = cs_test[,-c(which(names(cs_test) == "SeriousDlqin2yrs"), which(names(cs_test) == "X"))]
test.y = cs_test$SeriousDlqin2yrs
```

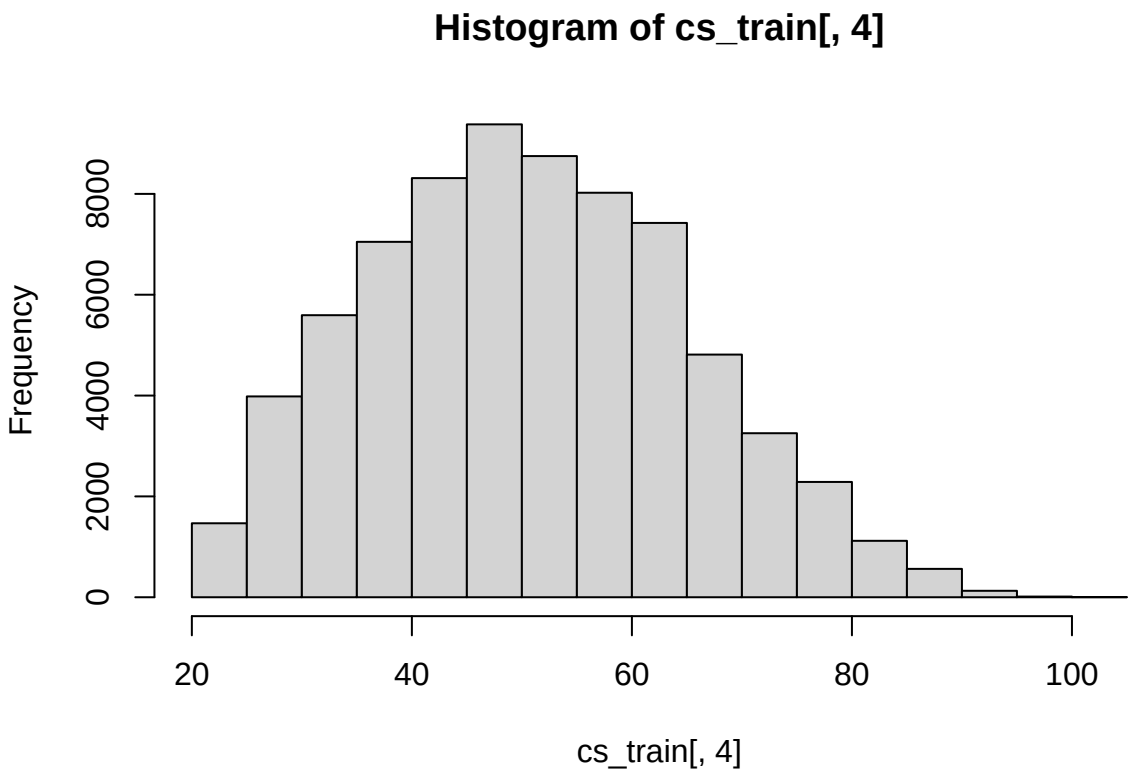
```
hist(cs_train[,2]) #Response Variable
```



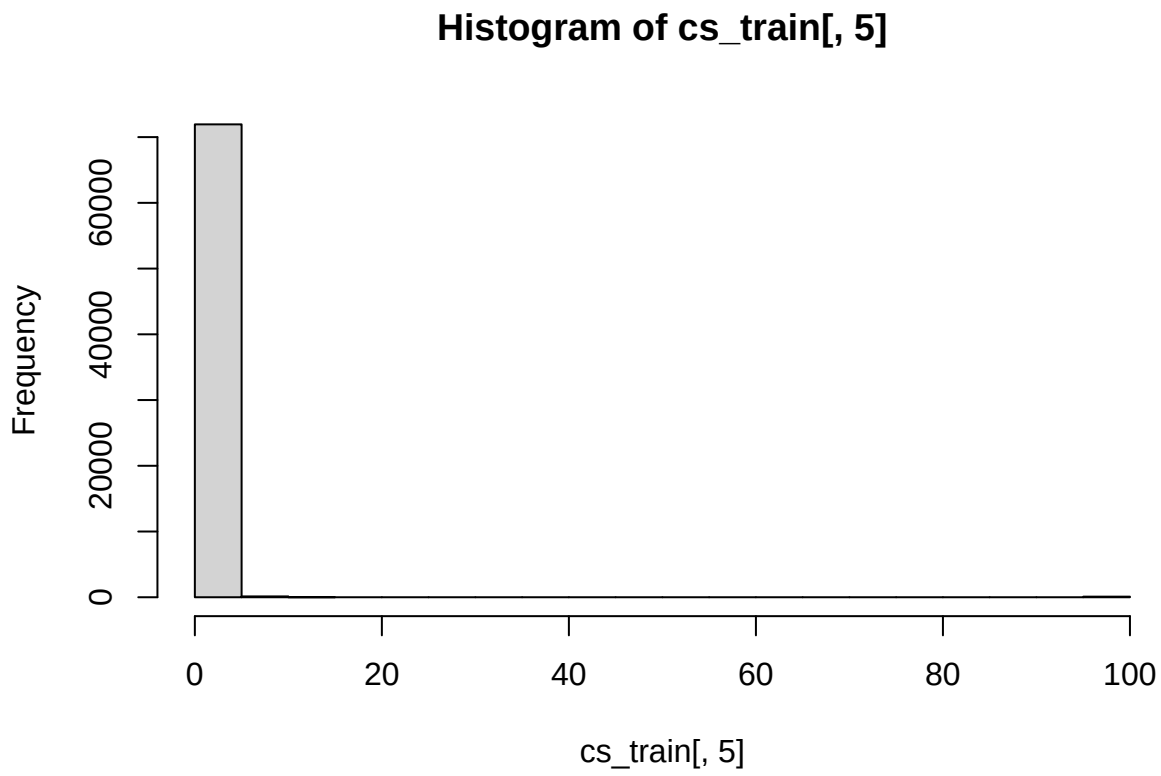

```
hist(cs_train[,3]) #RevolvingUtilizationOfUnsecuredLines
```



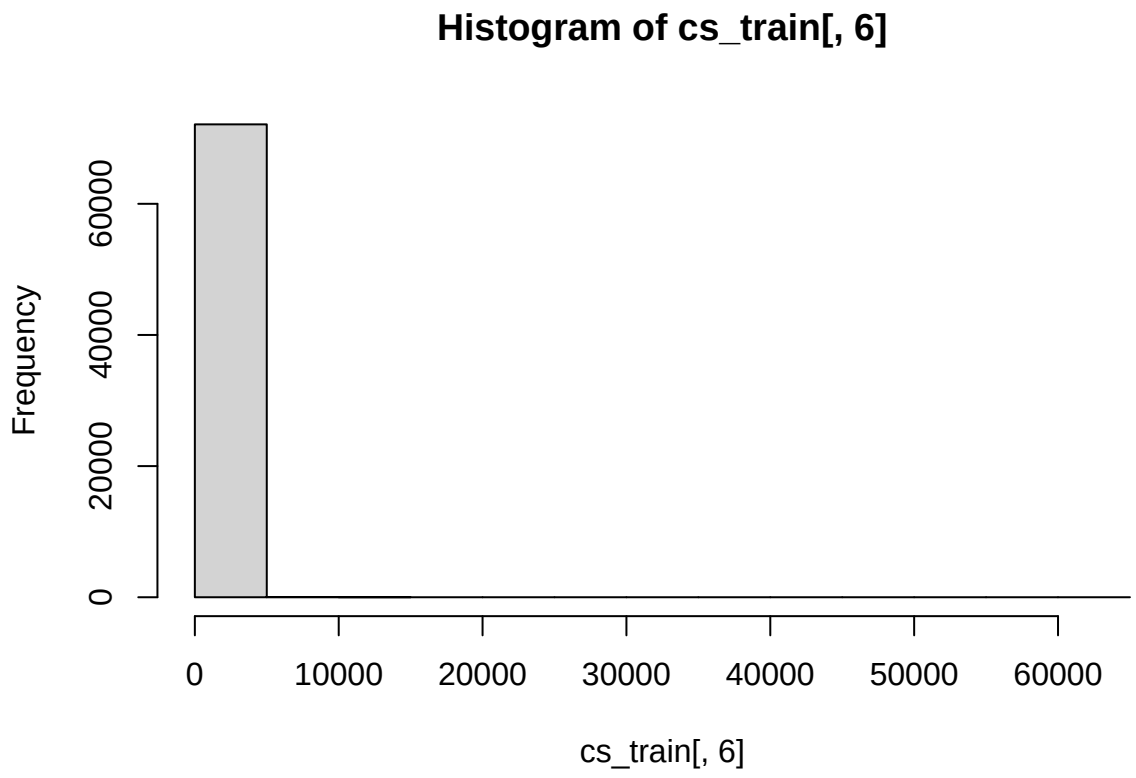

```
hist(cs_train[,4]) #age
```



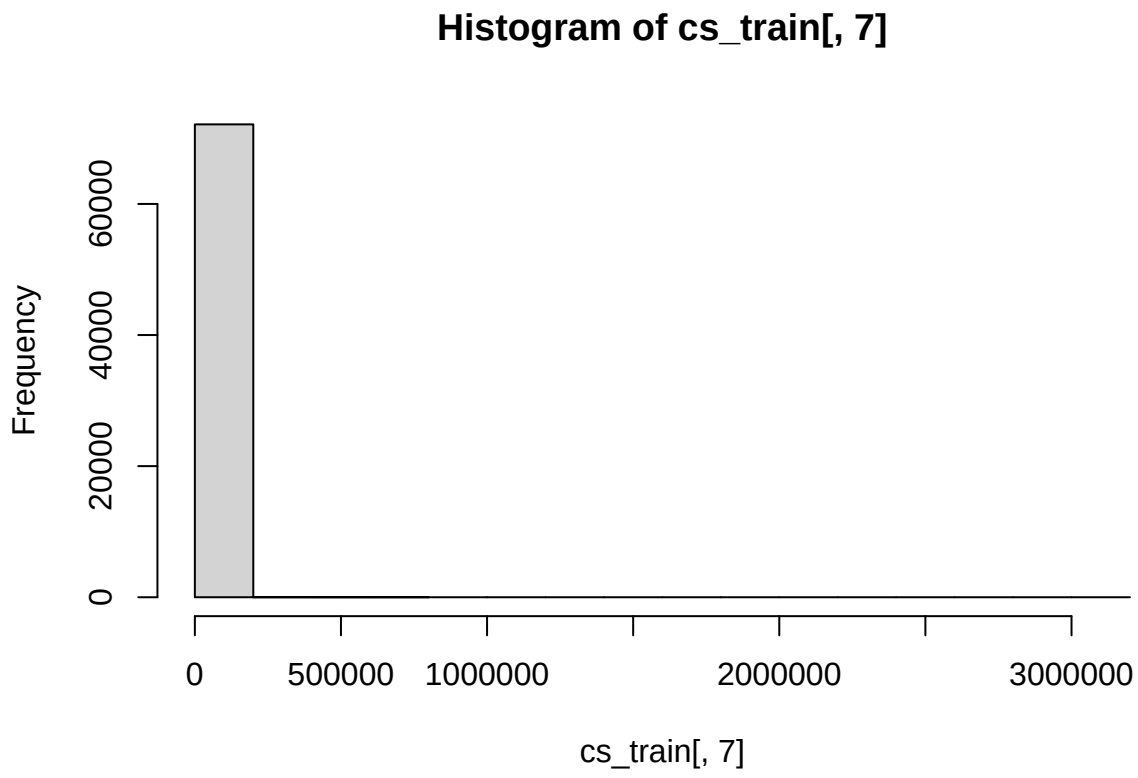

```
hist(cs_train[,5]) #NumberOfTime30.59DaysPastDueNotWorse
```



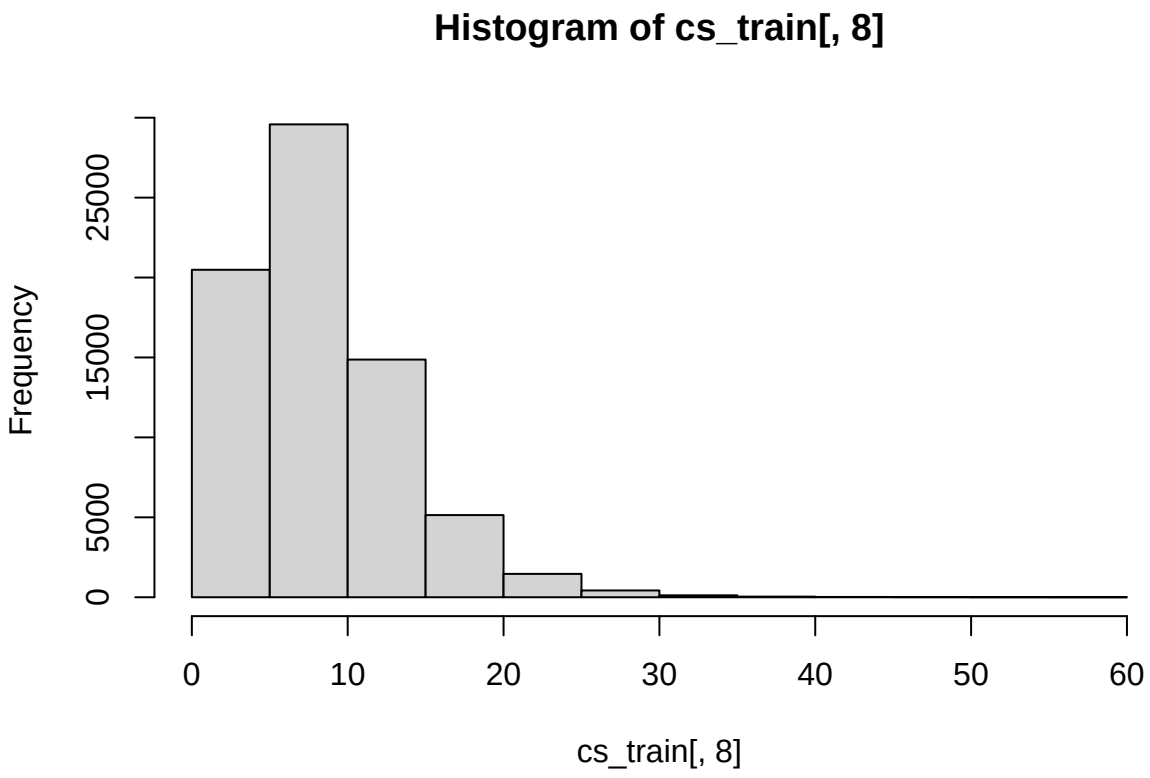

```
hist(cs_train[,6]) #DebtRatio
```



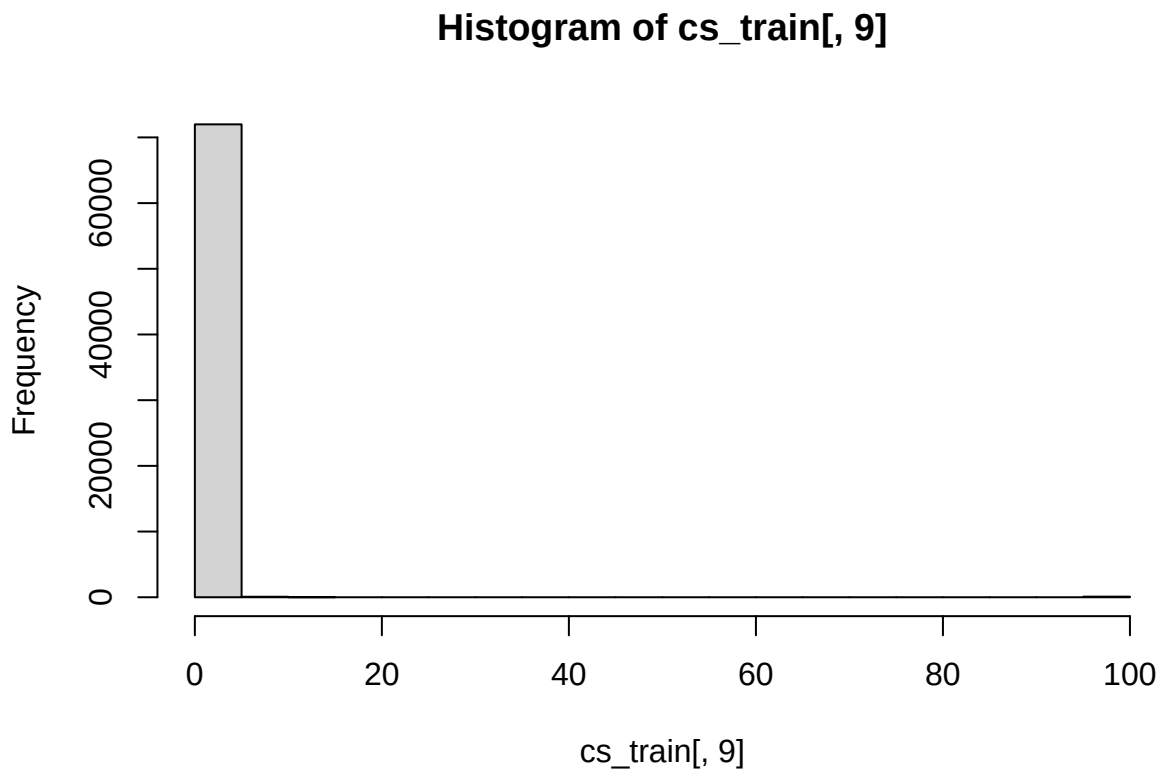

```
hist(cs_train[,7]) #MonthlyIncome
```



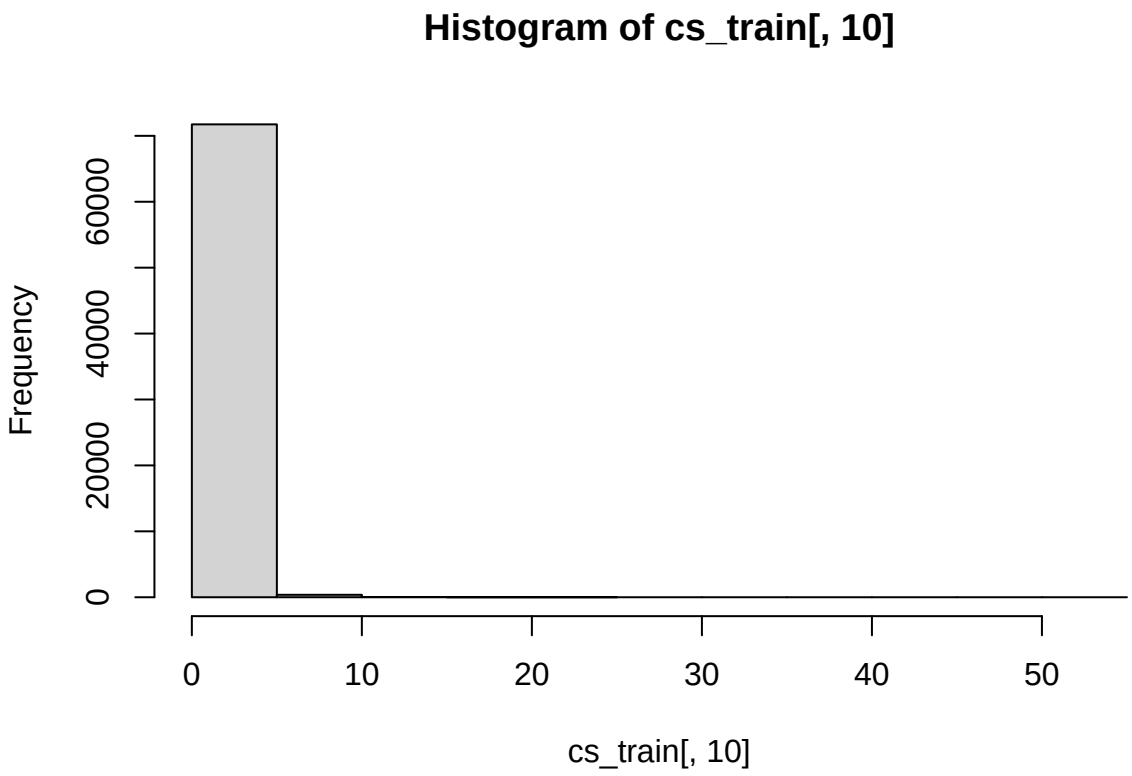

```
hist(cs_train[,8]) #NumberOfOpenCreditLinesAndLoans
```



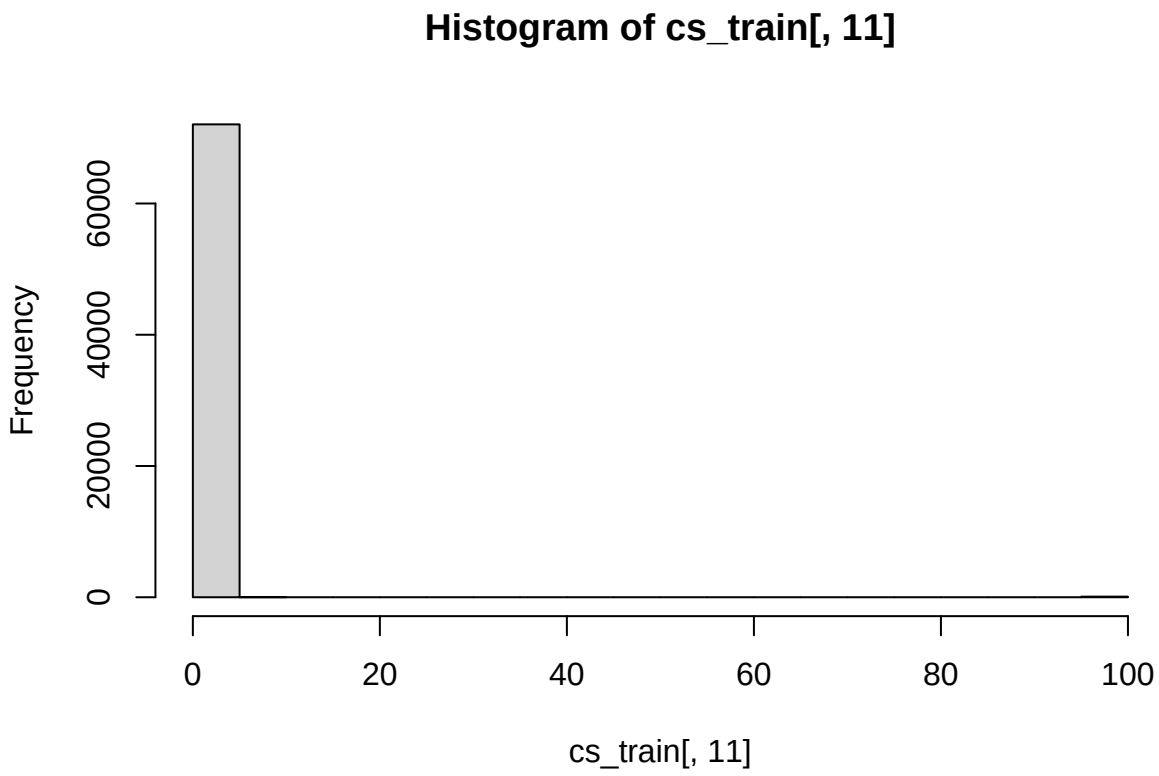

```
hist(cs_train[,9]) #NumberOfTimes90DaysLate
```



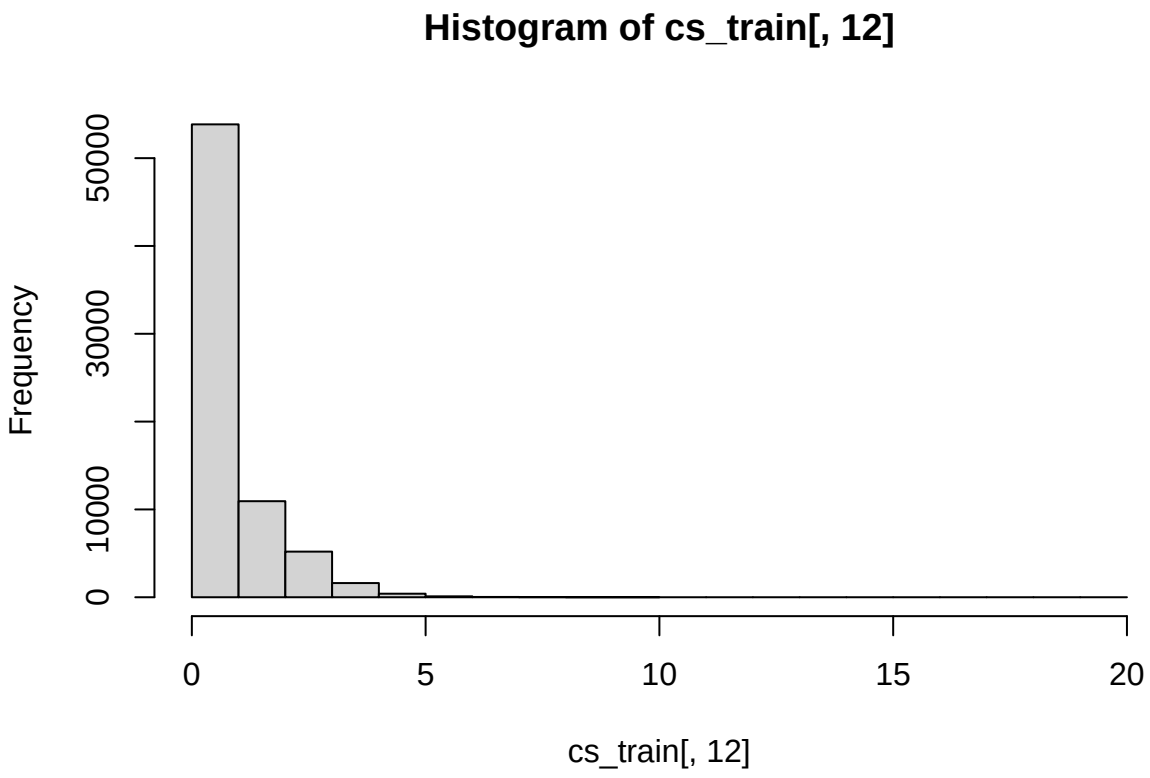

```
hist(cs_train[,10])#NumberRealEstateLoansOrLines
```




```
hist(cs_train[,11])#NumberOfTime60.89DaysPastDueNotWorse
```




```
hist(cs_train[,12])#NumberOfDependents
```




```
Jimmys_Bad_Variables <- cs_train
Jimmys_Bad_Variables$HighRevolving <- 0
Jimmys_Bad_Variables$HighRevolving[quantile(Jimmys_Bad_Variables[,3],0.95)] <- 1

Jimmys_Bad_Variables$Many_LessThan2MonthsLate <- 0
Jimmys_Bad_Variables$Many_LessThan2MonthsLate[quantile(Jimmys_Bad_Variables[,5],0.95)] <- 1

Jimmys_Bad_Variables$HighDebtRatio <- 0
Jimmys_Bad_Variables$HighDebtRatio[quantile(Jimmys_Bad_Variables[,6],0.95)] <- 1

Jimmys_Bad_Variables$Rich <- 0
Jimmys_Bad_Variables$Rich[quantile(Jimmys_Bad_Variables[,6],0.95)] <- 1

Jimmys_Bad_Variables$Many_CurrentLoans <- 0
Jimmys_Bad_Variables$Many_CurrentLoans[quantile(Jimmys_Bad_Variables[,8],0.95)] <- 1

Jimmys_Bad_Variables$Many_AtLeastThreeMonthsLate <- 0
Jimmys_Bad_Variables$Many_AtLeastThreeMonthsLate[quantile(Jimmys_Bad_Variables[,9],0.95)] <- 1

Jimmys_Bad_Variables$Many_HouseLoans <- 0
Jimmys_Bad_Variables$Many_HouseLoans[quantile(Jimmys_Bad_Variables[,10],0.95)] <- 1

Jimmys_Bad_Variables$Many_TwoToThreeMonthsLate <- 0
Jimmys_Bad_Variables$Many_TwoToThreeMonthsLate[quantile(Jimmys_Bad_Variables[,11],0.95)] <- 1

Jimmys_Bad_Variables$NA_Dependents_are_Mean <- Jimmys_Bad_Variables[,12] #Need to Change NA values to M
```

```
Jimmys_Bad_Variables$NA_Dependents_are_Mean[is.na(Jimmys_Bad_Variables[,12])] <- mean(Jimmys_Bad_Variabl
Jimmys_Bad_Variables$Many_Dependents <- 0
Jimmys_Bad_Variables$Many_Dependents[quantile(Jimmys_Bad_Variables$NA_Dependents_are_Mean,0.95)] <- 1

train1 <-Jimmys_Bad_Variables[Jimmys_Bad_Variables[,3]< quantile(Jimmys_Bad_Variables[,3],0.95),]
```

```
head(Jimmys_Bad_Variables)
```

```
##            X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 38516   38516                0                            0.20669915  55
## 78641   78641                0                            0.33463754  62
## 30110   30110                0                            0.90113992  51
## 37357   37357                0                            0.85201558  22
## 54665   54665                1                            0.99999990  40
## 106597 106597                0                            0.07304362  63
##        NumberOfTime30.59DaysPastDueNotWorse   DebtRatio MonthlyIncome
## 38516                                     0 0.776049767          4500
## 78641                                     2 0.377449020          2500
## 30110                                     0 0.210798492         11408
## 37357                                     0 0.644572526          1040
## 54665                                     0 0.009996002          2500
## 106597                                    0 0.101024412          9175
##        NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 38516                                8                       0
## 78641                                5                       0
## 30110                                7                       0
## 37357                                5                       0
## 54665                                1                       0
## 106597                              11                       0
##        NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 38516                             2                                    0
## 78641                             1                                    0
## 30110                             2                                    0
## 37357                             0                                    0
## 54665                             0                                    0
## 106597                            0                                    0
##        NumberOfDependents HighRevolving Many_LessThan2MonthsLate HighDebtRatio
## 38516                   0             0                        0             1
## 78641                   0             0                        1             0
## 30110                   0             0                        0             0
## 37357                   0             0                        0             0
## 54665                   0             0                        0             0
## 106597                  0             0                        0             0
##        Rich Many_CurrentLoans Many_AtLeastThreeMonthsLate Many_HouseLoans
## 38516     1                 0                           1               0
## 78641     0                 0                           0               0
## 30110     0                 0                           0               1
## 37357     0                 0                           0               0
## 54665     0                 0                           0               0
## 106597    0                 0                           0               0
##        Many_TwoToThreeMonthsLate NA_Dependents_are_Mean Many_Dependents
## 38516                          1                     0               0
## 78641                          0                     0               0
## 30110                          0                     0               1
```

```
## 37357                          0                       0              0
## 54665                          0                       0              0
## 106597                         0                       0              0
```

```
Just_Bad_Variables <- subset.data.frame(Jimmys_Bad_Variables, select = c(X,SeriousDlqin2yrs, age, HighR
```

##PCA using all 22 variables

```
pca_including_dummy_variables <- prcomp(na.omit(Jimmys_Bad_Variables[,-c(which(names(Jimmys_Bad_Variabl
```

```
summary(pca_including_dummy_variables) #First PC has like 99.66% of variance
```

```
## Importance of components:
##                             PC1       PC2       PC3   PC4   PC5   PC6  PC7
## Standard deviation     1.624e+04 466.60543 290.07477 14.46 5.962 5.073 1.59
## Proportion of Variance 9.989e-01   0.00082   0.00032  0.00 0.000 0.000 0.00
## Cumulative Proportion  9.989e-01   0.99968   1.00000  1.00 1.000 1.000 1.00
##                            PC8    PC9   PC10     PC11     PC12     PC13     PC14
## Standard deviation     1.028 0.5673 0.3456 0.007445 0.005265 0.003723 0.003722
## Proportion of Variance 0.000 0.0000 0.0000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.000 1.0000 1.0000 1.000000 1.000000 1.000000 1.000000
##                             PC15      PC16      PC17      PC18      PC19
## Standard deviation     6.588e-15 5.872e-17 5.262e-17 3.716e-19 4.715e-23
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC20
## Standard deviation     7.139e-36
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
pca_including_dummy_variables$rotation[,1] #I think it really likes Monthly Income as it's close to 1.
```

```
## RevolvingUtilizationOfUnsecuredLines                            age
##                         1.354991e-04                   3.150021e-05
## NumberOfTime30.59DaysPastDueNotWorse                      DebtRatio
##                         -2.057925e-06                  -6.581250e-04
##                         MonthlyIncome NumberOfOpenCreditLinesAndLoans
##                          9.999998e-01                   2.653969e-05
##              NumberOfTimes90DaysLate    NumberRealEstateLoansOrLines
##                         -2.516389e-06                   7.943599e-06
## NumberOfTime60.89DaysPastDueNotWorse              NumberOfDependents
##                         -2.190207e-06                   3.938171e-06
##                         HighRevolving       Many_LessThan2MonthsLate
##                          0.000000e+00                  -2.221966e-10
##                         HighDebtRatio                            Rich
##                         -1.171538e-10                  -1.171538e-10
##                    Many_CurrentLoans    Many_AtLeastThreeMonthsLate
##                         -2.878485e-10                  -1.171538e-10
##                      Many_HouseLoans      Many_TwoToThreeMonthsLate
##                          2.456645e-10                  -1.171538e-10
##               NA_Dependents_are_Mean                 Many_Dependents
##                         3.938171e-06                   2.456645e-10
```

```
glm_pca_including_dummy_variables <- glm(train.y ~ pca_including_dummy_variables$x[,1], family = "binom
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm_pca_including_dummy_variables)
```

```
##
## Call:
## glm(formula = train.y ~ pca_including_dummy_variables$x[, 1],
##     family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4215  -0.3951  -0.3796  -0.3564   4.8268
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -2.628e+00  1.510e-02 -174.02   <2e-16 ***
## pca_including_dummy_variables$x[, 1] -3.708e-05  3.609e-06  -10.27   <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: 36147  on 72160  degrees of freedom
## Residual deviance: 36017  on 72159  degrees of freedom
## AIC: 36021
##
## Number of Fisher Scoring iterations: 6
```

##PCA using original 10 variables

```
pca_original_data <- prcomp(train.x)
```

```
summary(pca_original_data) #PC1 has 99.66% variance explained
```

```
## Importance of components:
##                             PC1       PC2       PC3   PC4   PC5   PC6   PC7
## Standard deviation     1.624e+04 466.60543 290.07477 14.46 5.962 5.073 1.147
## Proportion of Variance 9.989e-01   0.00082   0.00032  0.00 0.000 0.000 0.000
## Cumulative Proportion  9.989e-01   0.99968   1.00000  1.00 1.000 1.000 1.000
##                            PC8    PC9   PC10
## Standard deviation     1.008 0.5672 0.3456
## Proportion of Variance 0.000 0.0000 0.0000
## Cumulative Proportion  1.000 1.0000 1.0000
```

```
pca_original_data$rotation[,1] #It also really likes MonthlyIncome and not anything else
```

```
## RevolvingUtilizationOfUnsecuredLines                             age
##                         1.354991e-04                    3.150021e-05
## NumberOfTime30.59DaysPastDueNotWorse                       DebtRatio
##                        -2.057925e-06                   -6.581250e-04
##                        MonthlyIncome NumberOfOpenCreditLinesAndLoans
##                         9.999998e-01                    2.653969e-05
##             NumberOfTimes90DaysLate    NumberRealEstateLoansOrLines
##                        -2.516389e-06                    7.943599e-06
## NumberOfTime60.89DaysPastDueNotWorse                 NumberOfDependents
##                        -2.190207e-06                    3.938171e-06
```

```
glm_pca_original_data <- glm(train.y ~ pca_original_data$x[,1], family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm_pca_original_data)
```

```
##
## Call:
## glm(formula = train.y ~ pca_original_data$x[, 1], family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4215  -0.3951  -0.3796  -0.3564   4.8268
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.628e+00  1.510e-02 -174.02   <2e-16 ***
## pca_original_data$x[, 1] -3.708e-05  3.609e-06  -10.27   <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: 36147  on 72160  degrees of freedom
## Residual deviance: 36017  on 72159  degrees of freedom
## AIC: 36021
##
## Number of Fisher Scoring iterations: 6
```

##PCA Using Dummy variables

```
pca_dummy <- prcomp(Just_Bad_Variables[,-c(which(names(Just_Bad_Variables) == "SeriousDlqin2yrs"), which
```

```
summary(pca_dummy) #PC 1 is 100%?
```

```
## Importance of components:
##                           PC1      PC2      PC3      PC4      PC5       PC6
## Standard deviation     14.43 0.007445 0.005265 0.003723 0.003723 3.048e-16
## Proportion of Variance  1.00 0.000000 0.000000 0.000000 0.000000 0.000e+00
## Cumulative Proportion   1.00 1.000000 1.000000 1.000000 1.000000 1.000e+00
##                              PC7       PC8       PC9      PC10
## Standard deviation     2.921e-16 8.856e-20 9.313e-23 1.711e-35
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00
```

```
pca_dummy$rotation[,1] #Age is significant
```

```
##                           age                HighRevolving
##                -1.000000e+00                -2.117582e-22
##     Many_LessThan2MonthsLate                HighDebtRatio
##                -7.134093e-07                -2.473981e-07
##                          Rich            Many_CurrentLoans
##                -2.473981e-07                -1.808250e-07
## Many_AtLeastThreeMonthsLate              Many_HouseLoans
##                -2.473981e-07                 1.889408e-08
##    Many_TwoToThreeMonthsLate              Many_Dependents
##                -2.473981e-07                 1.889408e-08
```

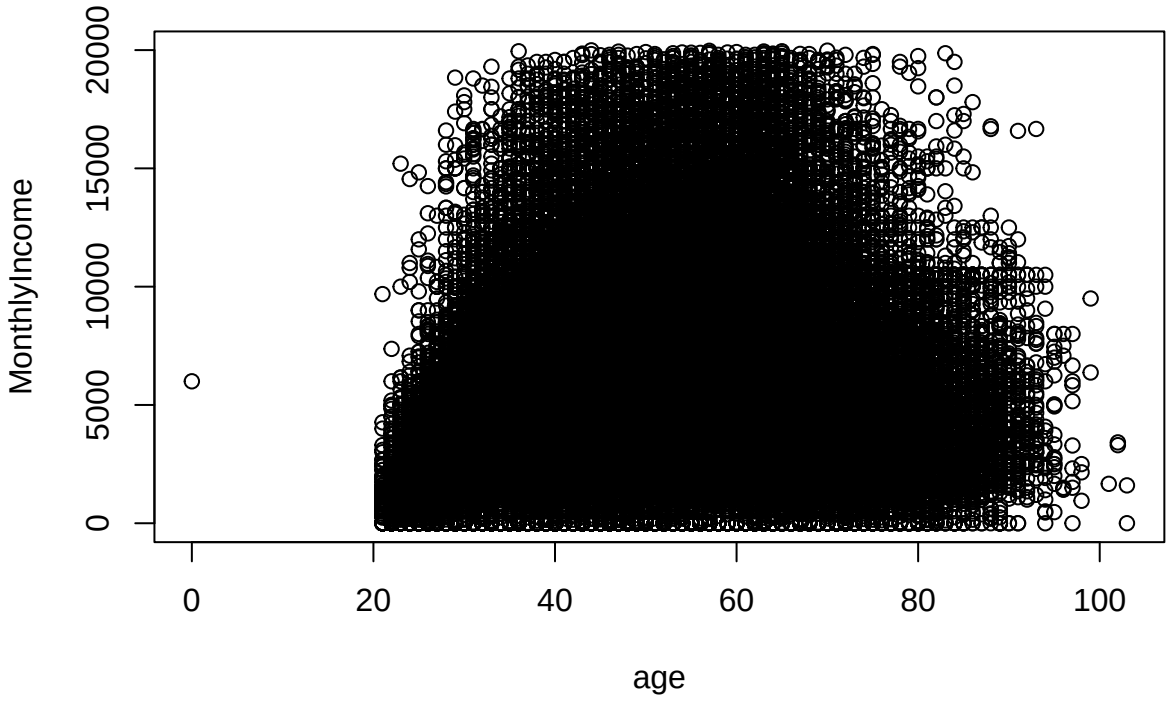
```r
glm_pca_dummy <- glm(train.y ~ pca_dummy$x[,1], family = "binomial")
summary(glm_pca_dummy)
```

```
##
## Call:
## glm(formula = train.y ~ pca_dummy$x[, 1], family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5526  -0.4148  -0.3558  -0.3048   2.8751
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.678271   0.015727 -170.30   <2e-16 ***
## pca_dummy$x[, 1]  0.028938   0.001092   26.49   <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: 36147  on 72160  degrees of freedom
## Residual deviance: 35406  on 72159  degrees of freedom
## AIC: 35410
##
## Number of Fisher Scoring iterations: 5
```

EDA

- monthly income = 0
- 30k monthly income = NA
- Dependent = NA 4k
- age 0 remove - only 1 point
- age very old
- group by age group etc
- 13 - 101 or older (maybe cut at 100)
- 80 or more

```r
#plot(SeriousDlqin2yrs ~ age, data = cs_train_maj)
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$MonthlyIncome<500000,] # less than 500k monthly income
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$MonthlyIncome<20000,] # less than 20k monthly income
plot(MonthlyIncome ~ age, data = cs_train_maj_g1)
```




```
train <- cs_train
##Shelly EDA Code
tapply(cs_train$RevolvingUtilizationOfUnsecuredLines, train$SeriousDlqin2yrs, median) #use this; data v
```

```
##         0         1
## 0.1556587 0.8169803
```

```
##Next Steps
# ***need final data set which excludes outliers for which we will create final model from
# do box plots for age group and income; fix axes so that box is readable -Shelly
# histogram of groupings like income by age groups, different colors -Yolanda
# sampling of data --> randomly select instead of adding additional data for response variable -Shelly
# ClassDiscovery package and DataExplorer --> provide initial graphs and analyses of data; also initial



 #Remove outliers; maybe only keep anything greater than 10
 tapply(train$RevolvingUtilizationOfUnsecuredLines, train$SeriousDlqin2yrs, mean) #d
```

```
##        0        1
## 6.136323 3.996190
```

```
 tapply(train$age, train$SeriousDlqin2yrs, median) #median age of 45 for people defaulting
```

```
##  0  1
## 51 45
```

```
#tapply(train$`NumberOfTime30-59DaysPastDueNotWorse`, train$SeriousDlqin2yrs, mean) #2.4 times for tho
tapply(train$NumberOfTimes90DaysLate, train$SeriousDlqin2yrs, mean) #2.1 times for those with delinque
```

```
##         0         1
## 0.1026816 1.6705622
```

```
tapply(train$DebtRatio, train$SeriousDlqin2yrs, median) #0.43 for delinquency incidence vs 0.36 for no
```

```
##         0         1
## 0.2920266 0.3600771
```

```
#tapply(train$NumIncome, train$SeriousDlqin2yrs, median) #delinquency monthly income of $3.8K vs $4.4K
tapply(train$NumberOfTimes90DaysLate, train$SeriousDlqin2yrs, mean)
```

```
##         0         1
## 0.1026816 1.6705622
```

```
#summary(train$NumIncome)
```

Question/Goal

- Comparing our model performance against paper's model
- Most important factors

Ensemble Learning

- Lasso Ensemble Algorithm
- Aggregating base learner: Weighted base=learner

Balancing Data

- Use clustering and pick one sub-group from majority
  - can try Age/Income
  - try Income/debt ratio
- Use bagging algorithm to create more minority data
- Use NA indicator 0 or 1 (maybe use mean/median)

```
# Cluster Grouping Majority Data (Try to cluster data by age/monthly income)
set.seed(1) # for reproducibility

head(cs_train_maj)
```

```
##   X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 2 2                0                             0.9571510  40
## 3 3                0                             0.6581801  38
## 4 4                0                             0.2338098  30
## 5 5                0                             0.9072394  49
## 6 6                0                             0.2131787  74
## 7 7                0                             0.3056825  57
##   NumberOfTime30.59DaysPastDueNotWorse    DebtRatio MonthlyIncome
## 2                                    0 1.218762e-01          2600
## 3                                    1 8.511338e-02          3042
## 4                                    0 3.604968e-02          3300
## 5                                    1 2.492570e-02         63588
```

```
## 6                                    0 3.756070e-01          3500
## 7                                    0 5.710000e+03            NA
##   NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 2                               4                       0
## 3                               2                       1
## 4                               5                       0
## 5                               7                       0
## 6                               3                       0
## 7                               8                       0
##   NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 2                            0                                     0
## 3                            0                                     0
## 4                            0                                     0
## 5                            1                                     0
## 6                            1                                     0
## 7                            3                                     0
##   NumberOfDependents NA_Indicator
## 2                  1            0
## 3                  0            0
## 4                  0            0
## 5                  0            0
## 6                  1            0
## 7                  0            1
```

```
# Test with smaller group based on monthly income range
cs_train_maj_g1 <- cs_train_maj[cs_train_maj$NA_Indicator==0,] # Filter out NA monthly income first
cs_train_maj_g1 <- cs_train_maj_g1[cs_train_maj_g1$MonthlyIncome<20000,]  #38035 obs
head(cs_train_maj_g1)
```

```
##     X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 2   2                0                             0.9571510  40
## 3   3                0                             0.6581801  38
## 4   4                0                             0.2338098  30
## 6   6                0                             0.2131787  74
## 8   8                0                             0.7544636  39
## 11 11                0                             0.6442260  30
##    NumberOfTime30.59DaysPastDueNotWorse  DebtRatio MonthlyIncome
## 2                                     0 0.12187620          2600
## 3                                     1 0.08511338          3042
## 4                                     0 0.03604968          3300
## 6                                     0 0.37560697          3500
## 8                                     0 0.20994002          3500
## 11                                    0 0.30947621          2500
##    NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 2                                4                       0
## 3                                2                       1
## 4                                5                       0
## 6                                3                       0
## 8                                8                       0
## 11                               5                       0
##    NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 2                             0                                     0
## 3                             0                                     0
## 4                             0                                     0
## 6                             1                                     0
```

```
## 8                            0                                     0
## 11                           0                                     0
##    NumberOfDependents NA_Indicator
## 2                   1            0
## 3                   0            0
## 4                   0            0
## 6                   1            0
## 8                   0            0
## 11                  0            0
```

```
# Create a dat with the two predictors of interest
dat <- cs_train_maj_g1[,c(4,7)]  # Age and MonthlyIncome
head(dat)
```

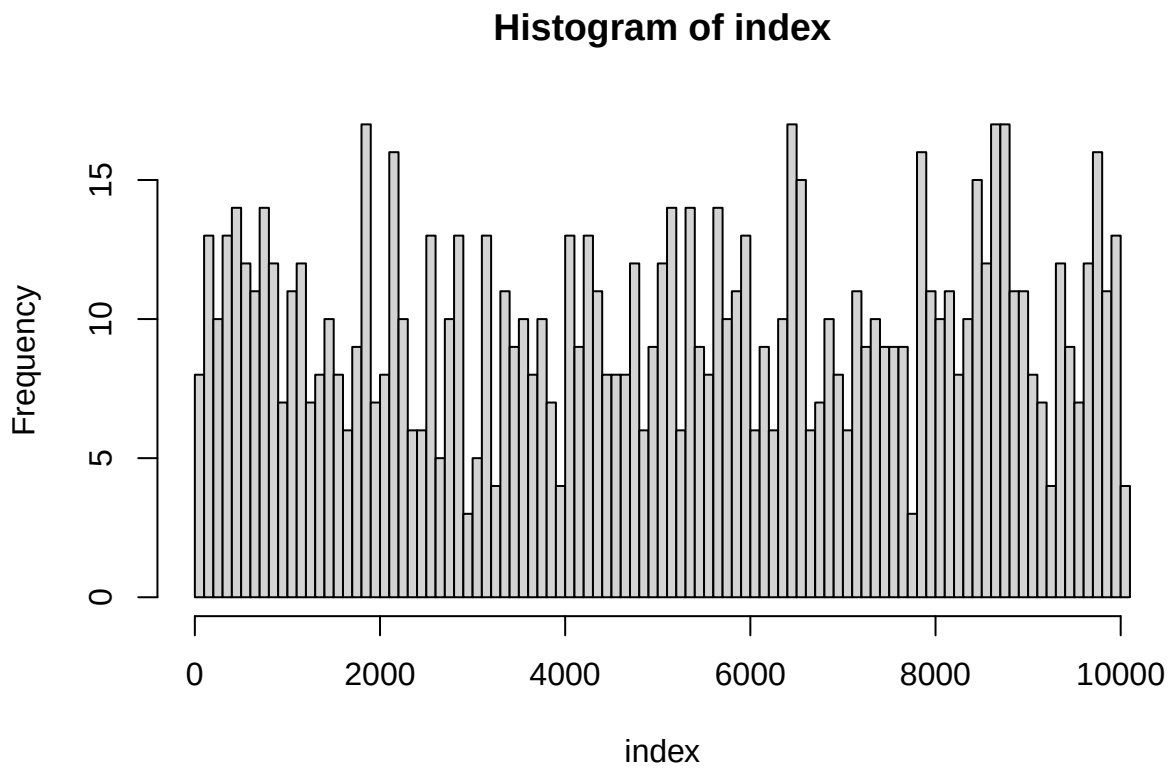
```
##    age MonthlyIncome
## 2   40          2600
## 3   38          3042
## 4   30          3300
## 6   74          3500
## 8   39          3500
## 11  30          2500
```

```
n_maj <- nrow(dat) # get number of rows

# Initial assignments to three groups that will need to update
assignments <- factor(sample(c(1,2,3), n_maj, replace = TRUE))
#plot(dat, col=assignments, xlim = c(0,110), asp=1)
#plot(dat, col=assignments)
```

```
# Boostrapping Minority Data
set.seed(1) # for reproducibility
# set number of minority data to reproduce
n_add <- 1000

n_min <- nrow(cs_train_min)
n_min
```

```
## [1] 10026
```

```
index <- sample(n_min, n_add, replace = TRUE)
#plot(density(index),  main="")  # show density curve of the index we randomized
hist(index, breaks = 100)
```




```
min(index)
```

```
## [1] 27
```

```
max(index)
```

```
## [1] 10014
```

```
length(index)
```

```
## [1] 1000
```

```
# We add the additional data for future analysis
cs_train_min_add <- cs_train_min[index,]
head(cs_train_min_add)
```

```
##             X SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age
## 15807   15807                1                            0.99999990  46
## 120332 120332                1                            0.04713019  45
## 71375   71375                1                            0.71069704  52
## 145782 145782                1                            0.99999990  52
## 127189 127189                1                            1.25614754  56
## 60636   60636                1                            0.41318696  35
##        NumberOfTime30.59DaysPastDueNotWorse    DebtRatio MonthlyIncome
## 15807                                     0    0.6327286          2700
## 120332                                    0    0.5847736          7000
## 71375                                     0    1.1807549          3761
## 145782                                    0    0.0000000          3200
## 127189                                    3    0.1938939          5600
```

```
## 60636                                      1 1203.0000000            NA
##        NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 15807                                3                       1
## 120332                               8                       0
## 71375                               11                       0
## 145782                               0                       0
## 127189                               7                       2
## 60636                               10                       0
##        NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 15807                             1                                    1
## 120332                            2                                    0
## 71375                             2                                    0
## 145782                            0                                    0
## 127189                            0                                    1
## 60636                             0                                    0
##        NumberOfDependents NA_Indicator
## 15807                   0            0
## 120332                  0            0
## 71375                   2            0
## 145782                  2            0
## 127189                  0            0
## 60636                   0            1
```

```
###SHELLY -  Balance Response Variable
 set.seed(2) # for reproducibility
 raw_data <- cs_train #150k rows
 cs_train_omit <- na.omit(raw_data)   # 150k -> 120,269 obs
 new_train <- filter(cs_train_omit, cs_train_omit$SeriousDlqin2yrs == 0) #112k rows

 #Check Response Variable Balance in cs_train_omit data
 table(cs_train_omit$SeriousDlqin2yrs) #8,357 1's
```

```
##
##     0     1
## 67198  4963
```

```
 new_train2 <- new_train[sample(1:nrow(new_train)),] #112k rows
 new_train3 <- new_train2[1:8357,]

 ########**Merge Data Together (use new_train5 variable)
 new_train4 <- filter(cs_train_omit, cs_train_omit$SeriousDlqin2yrs == 1)
 #nrow(new_train4)
 new_train5 <- rbind(new_train3, new_train4)
 nrow(new_train5) #16,714 rows as there should be (8357*2)
```

```
## [1] 13320
```

```
 table(new_train5$SeriousDlqin2yrs) #correct; equal # of 0's and 1's
```

```
##
##    0    1
## 8357 4963
```

##Split into Training and Testing

Which models to try

- Logistic regression (compare the different link functions)
- look maybe merge Lasso with logistic regression
- RF

Evaluating Model/Comparing results

- AUC

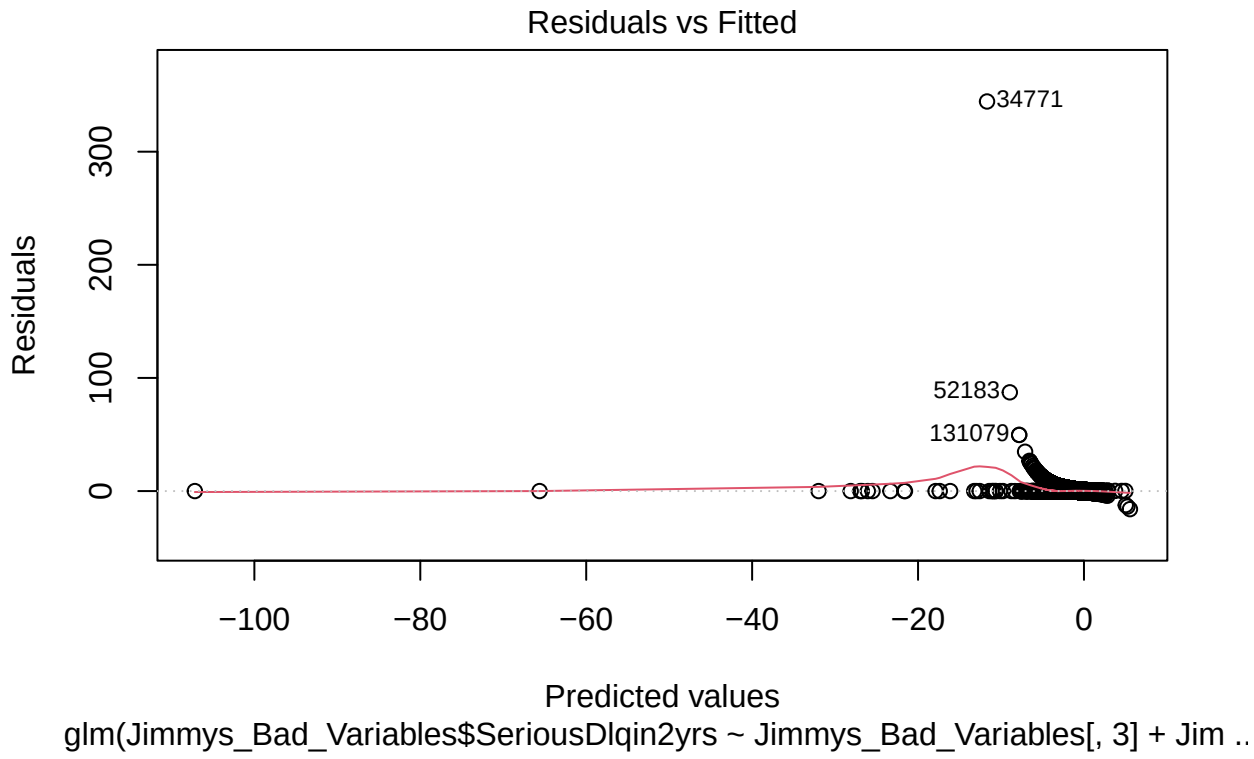
```
model <- glm(Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,3] +Jimmys_Bad_Variables[,4]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#To Change family link use family = quasi(variance = "mu^3", link = "log") change quasi
summary(model)
```

```
##
## Call:
## glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     3] + Jimmys_Bad_Variables[, 4] + Jimmys_Bad_Variables[, 5] +
##     Jimmys_Bad_Variables[, 6] + Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[,
##     8] + Jimmys_Bad_Variables[, 9] + Jimmys_Bad_Variables[, 10] +
##     Jimmys_Bad_Variables[, 11] + Jimmys_Bad_Variables[, 12],
##     family = "binomial", data = Jimmys_Bad_Variables)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3305  -0.3938  -0.3254  -0.2649   4.8339
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.550e+00  6.030e-02 -25.700  < 2e-16 ***
## Jimmys_Bad_Variables[, 3]   -3.818e-05  8.753e-05  -0.436 0.662690
## Jimmys_Bad_Variables[, 4]   -2.470e-02  1.205e-03 -20.490  < 2e-16 ***
## Jimmys_Bad_Variables[, 5]    5.086e-01  1.530e-02  33.254  < 2e-16 ***
## Jimmys_Bad_Variables[, 6]   -2.916e-04  8.438e-05  -3.456 0.000548 ***
## Jimmys_Bad_Variables[, 7]   -3.479e-05  4.047e-06  -8.595  < 2e-16 ***
## Jimmys_Bad_Variables[, 8]   -3.986e-03  3.506e-03  -1.137 0.255529
## Jimmys_Bad_Variables[, 9]    4.821e-01  2.207e-02  21.842  < 2e-16 ***
## Jimmys_Bad_Variables[, 10]   7.692e-02  1.407e-02   5.466 4.60e-08 ***
## Jimmys_Bad_Variables[, 11] -9.523e-01  2.541e-02 -37.475  < 2e-16 ***
## Jimmys_Bad_Variables[, 12]  9.509e-02  1.251e-02   7.600 2.97e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36147  on 72160  degrees of freedom
## Residual deviance: 33281  on 72150  degrees of freedom
## AIC: 33303
##
## Number of Fisher Scoring iterations: 6
```

```
plot(model, which = 1) #Outliers skew residuals plot
```




```
stepAIC(model)
```

```
## Start:  AIC=33303.09
## Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     3] + Jimmys_Bad_Variables[, 4] + Jimmys_Bad_Variables[, 5] +
##     Jimmys_Bad_Variables[, 6] + Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[,
##     8] + Jimmys_Bad_Variables[, 9] + Jimmys_Bad_Variables[, 10] +
##     Jimmys_Bad_Variables[, 11] + Jimmys_Bad_Variables[, 12]
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##                              Df Deviance   AIC
## - Jimmys_Bad_Variables[, 3]   1    33281 33301
## - Jimmys_Bad_Variables[, 8]   1    33282 33302
## <none>                             33281 33303
## - Jimmys_Bad_Variables[, 6]   1    33298 33318
## - Jimmys_Bad_Variables[, 10]  1    33310 33330
## - Jimmys_Bad_Variables[, 12]  1    33337 33357
## - Jimmys_Bad_Variables[, 7]   1    33372 33392
## - Jimmys_Bad_Variables[, 4]   1    33723 33743
## - Jimmys_Bad_Variables[, 9]   1    33757 33777
## - Jimmys_Bad_Variables[, 5]   1    34274 34294
## - Jimmys_Bad_Variables[, 11]  1    34651 34671
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=33301.32
## Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     4] + Jimmys_Bad_Variables[, 5] + Jimmys_Bad_Variables[, 6] +
##     Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[, 8] + Jimmys_Bad_Variables[,
##     9] + Jimmys_Bad_Variables[, 10] + Jimmys_Bad_Variables[,
##     11] + Jimmys_Bad_Variables[, 12]
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##                              Df Deviance   AIC
## - Jimmys_Bad_Variables[, 8]   1    33283 33301
## <none>                             33281 33301
## - Jimmys_Bad_Variables[, 6]   1    33299 33317
## - Jimmys_Bad_Variables[, 10]  1    33310 33328
## - Jimmys_Bad_Variables[, 12]  1    33337 33355
## - Jimmys_Bad_Variables[, 7]   1    33373 33391
## - Jimmys_Bad_Variables[, 4]   1    33723 33741
## - Jimmys_Bad_Variables[, 9]   1    33757 33775
## - Jimmys_Bad_Variables[, 5]   1    34275 34293
## - Jimmys_Bad_Variables[, 11]  1    34651 34669
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##
## Step:  AIC=33300.61
```

```
## Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     4] + Jimmys_Bad_Variables[, 5] + Jimmys_Bad_Variables[, 6] +
##     Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[, 9] + Jimmys_Bad_Variables[,
##     10] + Jimmys_Bad_Variables[, 11] + Jimmys_Bad_Variables[,
##     12]
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##                              Df Deviance   AIC
## <none>                             33283 33301
## - Jimmys_Bad_Variables[, 6]   1    33300 33316
## - Jimmys_Bad_Variables[, 10]  1    33311 33327
## - Jimmys_Bad_Variables[, 12]  1    33338 33354
## - Jimmys_Bad_Variables[, 7]   1    33378 33394
## - Jimmys_Bad_Variables[, 4]   1    33749 33765
## - Jimmys_Bad_Variables[, 9]   1    33768 33784
## - Jimmys_Bad_Variables[, 5]   1    34280 34296
## - Jimmys_Bad_Variables[, 11]  1    34653 34669
```

```
##
## Call:  glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     4] + Jimmys_Bad_Variables[, 5] + Jimmys_Bad_Variables[, 6] +
##     Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[, 9] + Jimmys_Bad_Variables[,
##     10] + Jimmys_Bad_Variables[, 11] + Jimmys_Bad_Variables[,
##     12], family = "binomial", data = Jimmys_Bad_Variables)
##
## Coefficients:
##                (Intercept)   Jimmys_Bad_Variables[, 4]
##                 -1.561e+00                  -2.496e-02
##  Jimmys_Bad_Variables[, 5]   Jimmys_Bad_Variables[, 6]
##                  5.063e-01                  -2.936e-04
##  Jimmys_Bad_Variables[, 7]   Jimmys_Bad_Variables[, 9]
##                 -3.536e-05                   4.850e-01
## Jimmys_Bad_Variables[, 10]  Jimmys_Bad_Variables[, 11]
##                  7.098e-02                  -9.528e-01
## Jimmys_Bad_Variables[, 12]
##                  9.497e-02
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72152 Residual
## Null Deviance:       36150
## Residual Deviance: 33280     AIC: 33300
```

```
model2 <- glm(Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,13] + Jimmys
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model2
```

```
##
## Call:  glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome +
##     Jimmys_Bad_Variables[, 13] + Jimmys_Bad_Variables[, 14] +
##     Jimmys_Bad_Variables[, 15] + Jimmys_Bad_Variables[, 16] +
##     Jimmys_Bad_Variables[, 17] + Jimmys_Bad_Variables[, 18] +
##     Jimmys_Bad_Variables[, 19] + Jimmys_Bad_Variables[, 20] +
##     Jimmys_Bad_Variables[, 22], family = "binomial", data = Jimmys_Bad_Variables)
##
## Coefficients:
##                 (Intercept)                MonthlyIncome
##                  -2.378e+00                   -3.709e-05
## Jimmys_Bad_Variables[, 13]  Jimmys_Bad_Variables[, 14]
##                          NA                   -8.095e+00
## Jimmys_Bad_Variables[, 15]  Jimmys_Bad_Variables[, 16]
##                  -8.021e+00                           NA
## Jimmys_Bad_Variables[, 17]  Jimmys_Bad_Variables[, 18]
##                  -8.142e+00                           NA
## Jimmys_Bad_Variables[, 19]  Jimmys_Bad_Variables[, 20]
##                  -7.765e+00                           NA
## Jimmys_Bad_Variables[, 22]
##                          NA
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72155 Residual
## Null Deviance:       36150
## Residual Deviance: 36020     AIC: 36030
```

```
model2$rank # equals 7 which is how many variables are not NA
```

```
## [1] 6
```

```
#Certain Dummy Variables are a  Singular Matrix Meaning that some of our variables can be constructed u
#Not Sure what to do, but I'll remove these variables in the meantime
```

```
model3 <- glm(Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,15] + Jimmys
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model3
```

```
##
## Call:  glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome +
##     Jimmys_Bad_Variables[, 15] + Jimmys_Bad_Variables[, 16] +
##     Jimmys_Bad_Variables[, 17] + Jimmys_Bad_Variables[, 18] +
##     Jimmys_Bad_Variables[, 19], family = "binomial", data = Jimmys_Bad_Variables)
##
## Coefficients:
##                 (Intercept)                MonthlyIncome
##                  -2.378e+00                   -3.709e-05
## Jimmys_Bad_Variables[, 15]  Jimmys_Bad_Variables[, 16]
##                  -8.021e+00                           NA
## Jimmys_Bad_Variables[, 17]  Jimmys_Bad_Variables[, 18]
##                  -8.142e+00                           NA
## Jimmys_Bad_Variables[, 19]
##                  -7.765e+00
```

```
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36150
## Residual Deviance: 36020     AIC: 36030
```

```
model4 <- glm(Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,15] + Jimmys
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model4
```

```
##
## Call:  glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome +
##     Jimmys_Bad_Variables[, 15] + Jimmys_Bad_Variables[, 16] +
##     Jimmys_Bad_Variables[, 17] + Jimmys_Bad_Variables[, 18] +
##     Jimmys_Bad_Variables[, 19], family = binomial(link = "probit"),
##     data = Jimmys_Bad_Variables)
##
## Coefficients:
##                (Intercept)               MonthlyIncome
##                  -1.399e+00                  -1.375e-05
## Jimmys_Bad_Variables[, 15]  Jimmys_Bad_Variables[, 16]
##                  -2.705e+00                          NA
## Jimmys_Bad_Variables[, 17]  Jimmys_Bad_Variables[, 18]
##                  -2.750e+00                          NA
## Jimmys_Bad_Variables[, 19]
##                  -2.610e+00
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36150
## Residual Deviance: 36040     AIC: 36050
```

```
model5 <- glm(Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,15] + Jimmys
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model5
```

```
##
## Call:  glm(formula = Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome +
##     Jimmys_Bad_Variables[, 15] + Jimmys_Bad_Variables[, 16] +
##     Jimmys_Bad_Variables[, 17] + Jimmys_Bad_Variables[, 18] +
##     Jimmys_Bad_Variables[, 19], family = binomial(link = "cloglog"),
##     data = Jimmys_Bad_Variables)
##
## Coefficients:
##                (Intercept)               MonthlyIncome
##                  -2.418e+00                  -3.647e-05
## Jimmys_Bad_Variables[, 15]  Jimmys_Bad_Variables[, 16]
##                  -7.894e+00                          NA
## Jimmys_Bad_Variables[, 17]  Jimmys_Bad_Variables[, 18]
##                  -8.013e+00                          NA
## Jimmys_Bad_Variables[, 19]
##                  -7.642e+00
##
## Degrees of Freedom: 72160 Total (i.e. Null);  72156 Residual
## Null Deviance:       36150
```

```
## Residual Deviance: 36010     AIC: 36020
```

```
anova(model,model2,model3,model4,model5) #model 2 is the one with NA values
```

```
## Analysis of Deviance Table
##
## Model 1: Jimmys_Bad_Variables$SeriousDlqin2yrs ~ Jimmys_Bad_Variables[,
##     3] + Jimmys_Bad_Variables[, 4] + Jimmys_Bad_Variables[, 5] +
##     Jimmys_Bad_Variables[, 6] + Jimmys_Bad_Variables[, 7] + Jimmys_Bad_Variables[,
##     8] + Jimmys_Bad_Variables[, 9] + Jimmys_Bad_Variables[, 10] +
##     Jimmys_Bad_Variables[, 11] + Jimmys_Bad_Variables[, 12]
## Model 2: Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,
##     13] + Jimmys_Bad_Variables[, 14] + Jimmys_Bad_Variables[,
##     15] + Jimmys_Bad_Variables[, 16] + Jimmys_Bad_Variables[,
##     17] + Jimmys_Bad_Variables[, 18] + Jimmys_Bad_Variables[,
##     19] + Jimmys_Bad_Variables[, 20] + Jimmys_Bad_Variables[,
##     22]
## Model 3: Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,
##     15] + Jimmys_Bad_Variables[, 16] + Jimmys_Bad_Variables[,
##     17] + Jimmys_Bad_Variables[, 18] + Jimmys_Bad_Variables[,
##     19]
## Model 4: Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,
##     15] + Jimmys_Bad_Variables[, 16] + Jimmys_Bad_Variables[,
##     17] + Jimmys_Bad_Variables[, 18] + Jimmys_Bad_Variables[,
##     19]
## Model 5: Jimmys_Bad_Variables$SeriousDlqin2yrs ~ MonthlyIncome + Jimmys_Bad_Variables[,
##     15] + Jimmys_Bad_Variables[, 16] + Jimmys_Bad_Variables[,
##     17] + Jimmys_Bad_Variables[, 18] + Jimmys_Bad_Variables[,
##     19]
##   Resid. Df Resid. Dev Df Deviance
## 1     72150      33281
## 2     72155      36017 -5 -2735.47
## 3     72156      36017 -1    -0.16
## 4     72156      36039  0   -22.75
## 5     72156      36015  0    24.93
```

```
#model 1 is the best but we can check model 5 using our dummy variables
```

```
x <- data.frame(Jimmys_Bad_Variables$SeriousDlqin2yrs, Jimmys_Bad_Variables$MonthlyIncome, Jimmys_Bad_Va
x <- na.omit(x) #Remember Monthly Income contains NA values
x_for_ridge <- data.matrix(x[,-1]) #Everything but Response Variable
ridge_model5 <- cv.glmnet(x_for_ridge,x[,1], alpha = 0, standardize = TRUE, nfolds = length(x))
ridge_model5
```

```
##
## Call:  cv.glmnet(x = x_for_ridge, y = x[, 1], nfolds = length(x), alpha = 0,      standardize = TRUE
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure        SE Nonzero
## min  0.842    18 0.06404 0.0007541       6
## 1se  4.093     1 0.06405 0.0007531       6
```

```
lasso_model5 <- cv.glmnet(x_for_ridge,x[,1], alpha = 1, standardize = TRUE, nfolds = length(x))
lasso_model5
```

```
##
```

```
## Call:  cv.glmnet(x = x_for_ridge, y = x[, 1], nfolds = length(x), alpha = 1,      standardize = TRUE
##
## Measure: Mean-Squared Error
##
##       Lambda Index Measure        SE Nonzero
## min 0.001614    11 0.06404 0.0004029       1
## 1se 0.004093     1 0.06405 0.0004053       0
```

Evaluate Final Model

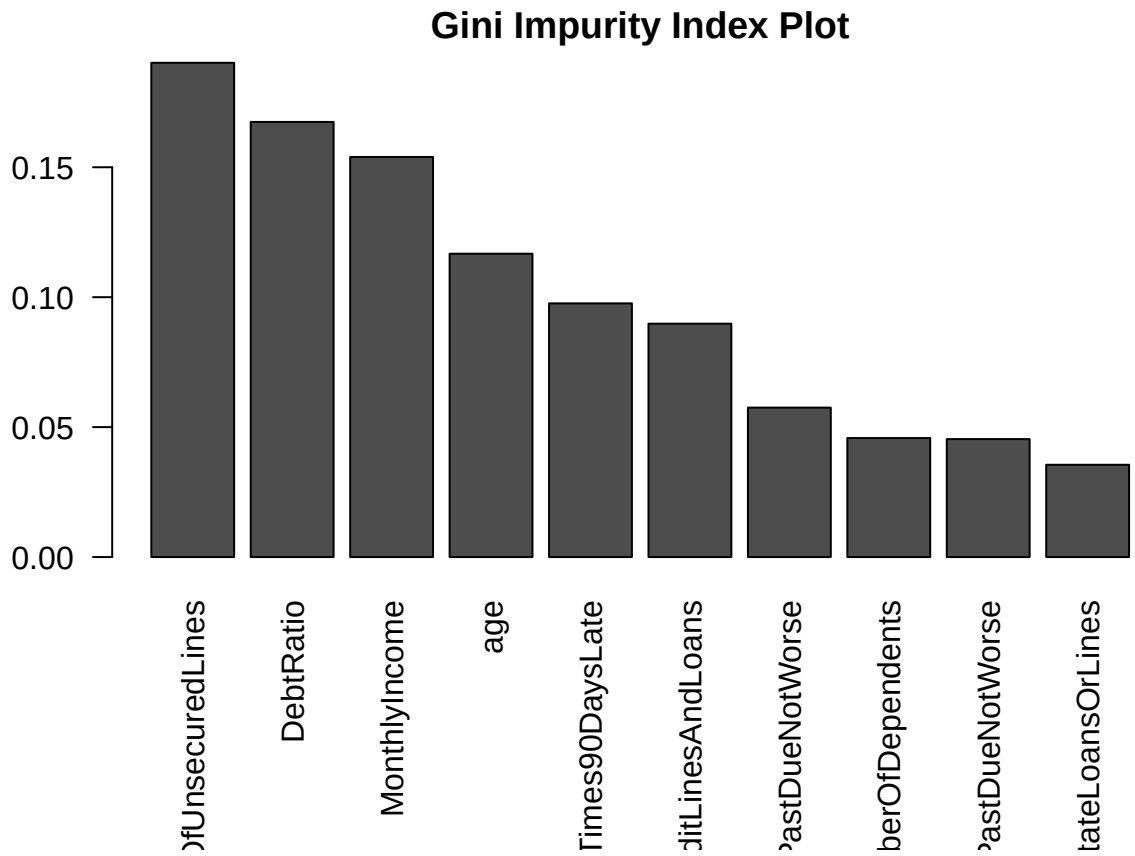
- mmp plots to see if model fits the data, as well as the pearson Chi-square test
- checking for outliers and influential points
- Some measure of pesudo R-square and accuracy of the model
- Use confusion matrix, ROC/AUC curve, AIC to evaluate the different models

Model after recoding Random Forest

```
# Fits Random Forest
model.rf = randomForest(y = as.factor(train.y), x=train.x, xtest=test.x, ytest=as.factor(test.y), mtry =

model.rf # Output shows confusion matrix for both train and test
```

```
##
## Call:
##  randomForest(x = train.x, y = as.factor(train.y), xtest = test.x,      ytest = as.factor(test.y), m
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 6.58%
## Confusion matrix:
##       0   1 class.error
## 0 66499 699   0.0104021
## 1  4052 911   0.8164417
##                 Test set error rate: 6.74%
## Confusion matrix:
##       0   1 class.error
## 0 44297 417 0.009325938
## 1  2826 568 0.832645846
```

```
# Random Forest Output
var.imp = data.frame(importance(model.rf, type=2))
var.imp$Variables = row.names(var.imp)
varimp = var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),]
par(mar = c(7.5,3,2,2))
giniplot = barplot(t(varimp[-2]/sum(varimp[-2])),las=2,cex.names=1, main="Gini Impurity Index Plot")
```

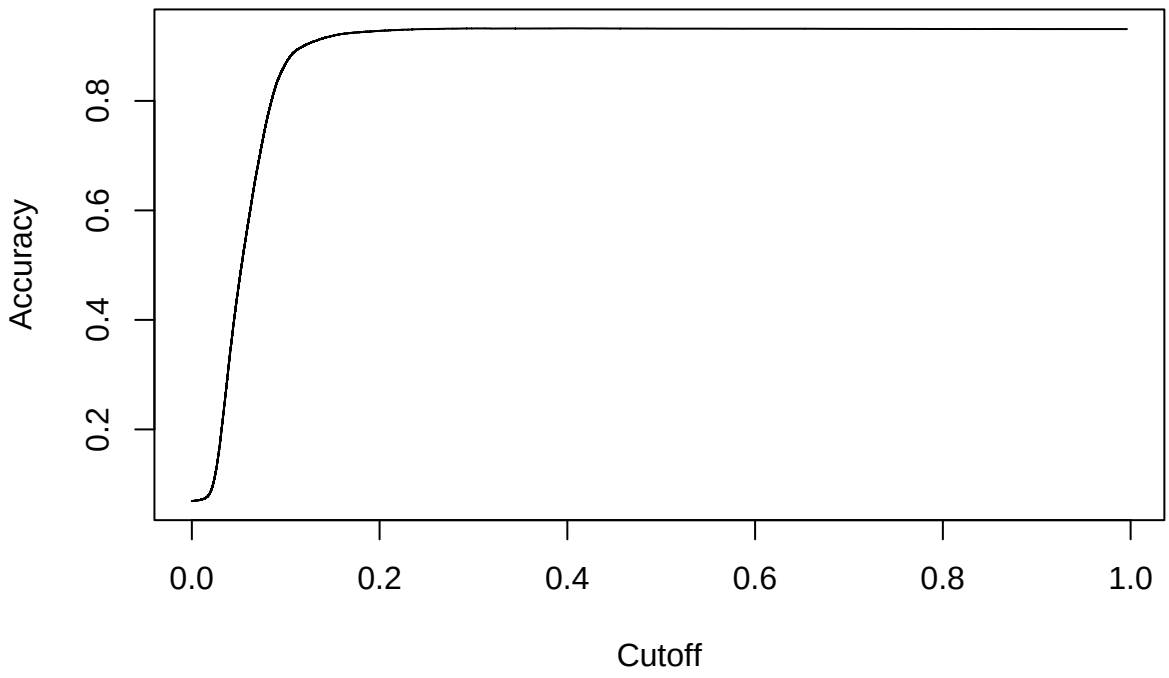


Evaluation on Final Model using Training Data

```
## Set model as final model
model.final <- model

## Evaluation on Final model using Training Data
train_preds = predict(model.final, newdata=train.x, type="response")
head(train_preds[is.na(train_preds)])
```

```
## named numeric(0)
```

```
#train.x[c(7,9),]
#train_preds[is.na(train_preds)]

pred_compare = prediction(train_preds, train.y)
plot(performance(pred_compare, "acc"))
```




```
table(train.y, train_preds>0.2) # accurary on train
```

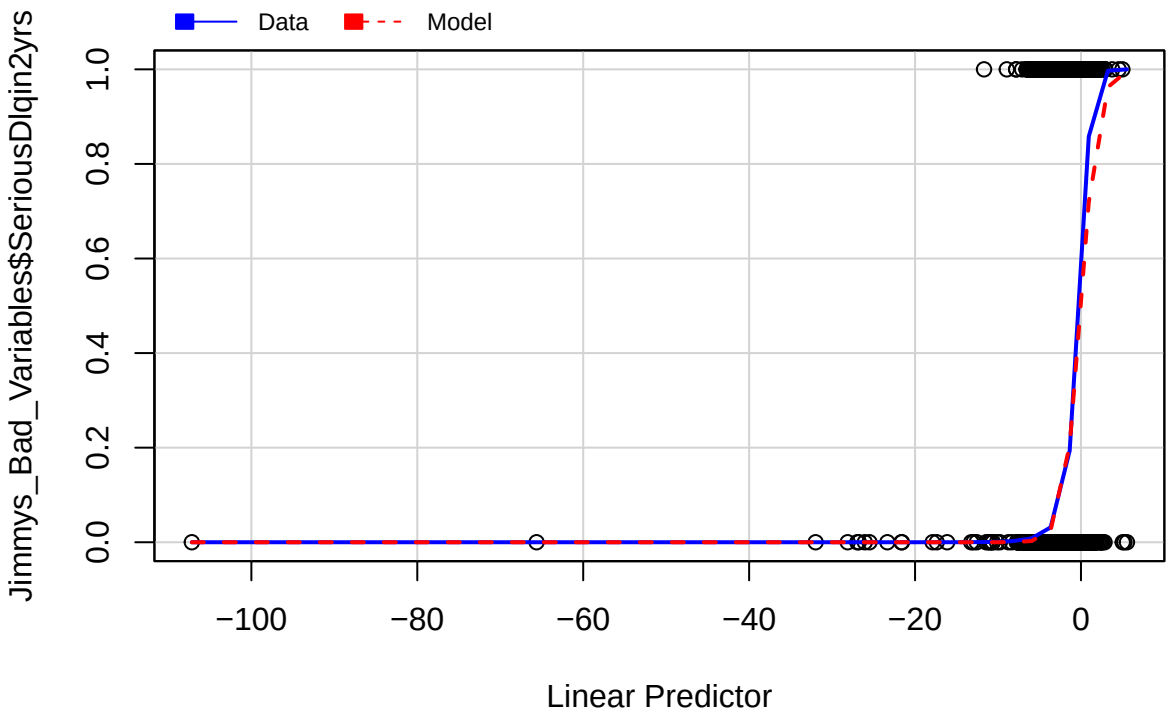
```
##
## train.y FALSE  TRUE
##       0 66094  1104
##       1  4096   867
```

Evaluating Model/Comparing results, Plots, Tests, chi-square test, influential points

Need to Update!!

```
# residual plots
#residualPlots(model.final)

# Marginial Model Plots
library(car)
mmp(model.final)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



Goodness of Fit using Hosmer-Lemeshow Test

The p-value is 0, meaning that we want to reject the null hypothesis that the model is adequate.

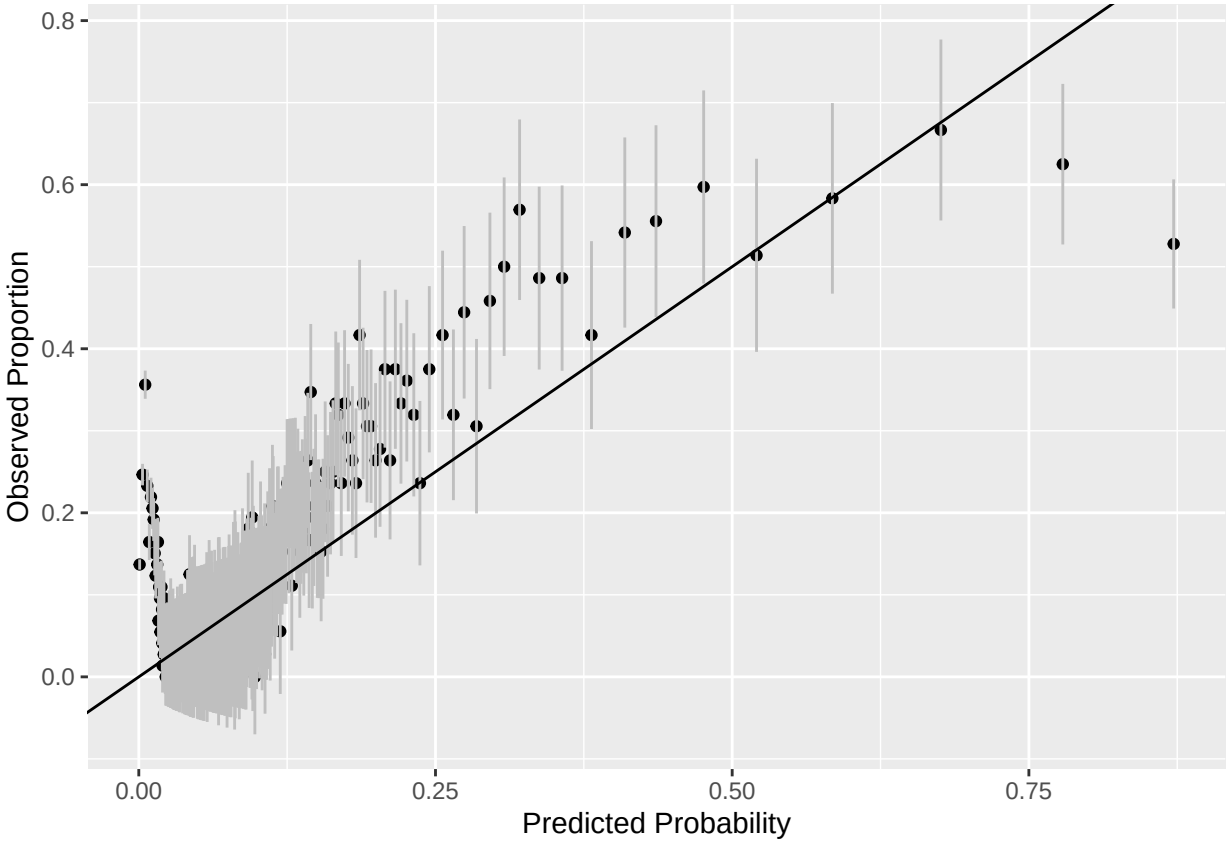
```
# Goodness of Fit using Hosmer-Lemeshow Test
linpred=predict(model.final)

cs_train_m <- mutate(cs_train, predprob=predict(model.final, type="response"))  # cal p^_i
gdf <- group_by(cs_train_m, ntile(linpred, 1000)) # group up the data by eta_x into 100 groups
hldf <- summarise(gdf, y=sum(SeriousDlqin2yrs==1), ppred=mean(predprob), count=n())
head(hldf)
```

```
## # A tibble: 6 x 4
##   `ntile(linpred, 1000)`     y    ppred count
##                    <int> <int>    <dbl> <int>
## 1                      1    10 0.000716    73
## 2                      2    18 0.00315     73
## 3                      3    26 0.00542     73
## 4                      4    17 0.00719     73
## 5                      5    12 0.00885     73
## 6                      6    16 0.0103      73
```

```
# We adjust the size of the bins until there's only one group with less than 5
hldf[hldf$count<5,]
```

```
## # A tibble: 0 x 4
## # ... with 4 variables: ntile(linpred, 1000) <int>, y <int>, ppred <dbl>,
## #   count <int>
```

```
# Observed Proportion Confidence Interval vs Predicted Probability
hldf <- mutate(hldf, se.fit=sqrt(ppred*(1-ppred)/count))

ggplot(hldf,aes(x=ppred,y=y/count,ymin=y/count-2*se.fit, ymax=y/count+2*se.fit))+
  geom_point()+geom_linerange(color=grey(0.75))+
  geom_abline(intercept = 0,slope = 1)+
  xlab("Predicted Probability")+
  ylab("Observed Proportion")
```




```
# Hosmer-Lemeshow statistics
hlstat <- with(hldf, sum((y-count*ppred) ^2/(count * ppred * (1-ppred))))
c(hlstat, nrow(hldf))
```

```
## [1] 8495.698 1000.000
```

```
# The p-value is given by:
1-pchisq(hlstat, nrow(hldf)-2)
```

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

AUC

Need to fix the Prediction function!!!

Predict -> Currently gives 120k, expecting 80k

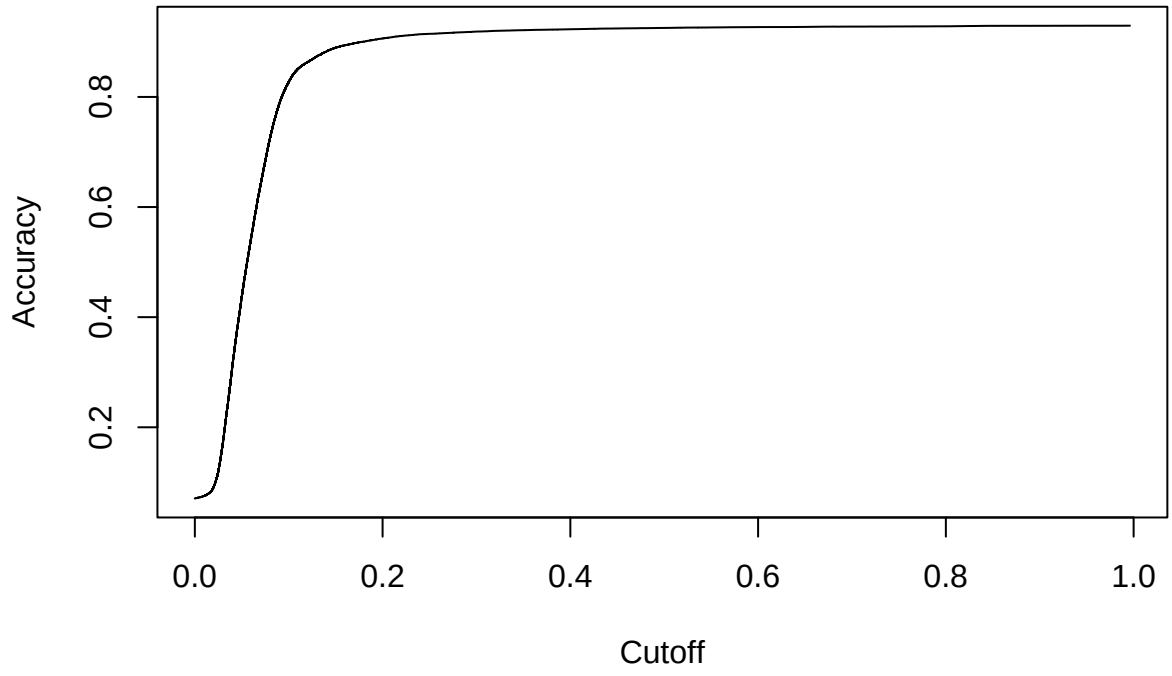
```
#Final Model(w/ interaction term ReugularMedicine * PhysicallyActive)
result_m2 = predict(model.final, newdata=test.x, type="response")
```

```
## Warning: 'newdata' had 48108 rows but variables found have 72161 rows
```

```
# Fix Later!!! Forced the length to be the same
result_m2 <- result_m2[1:length(test.y)]
head(test.y)
```

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

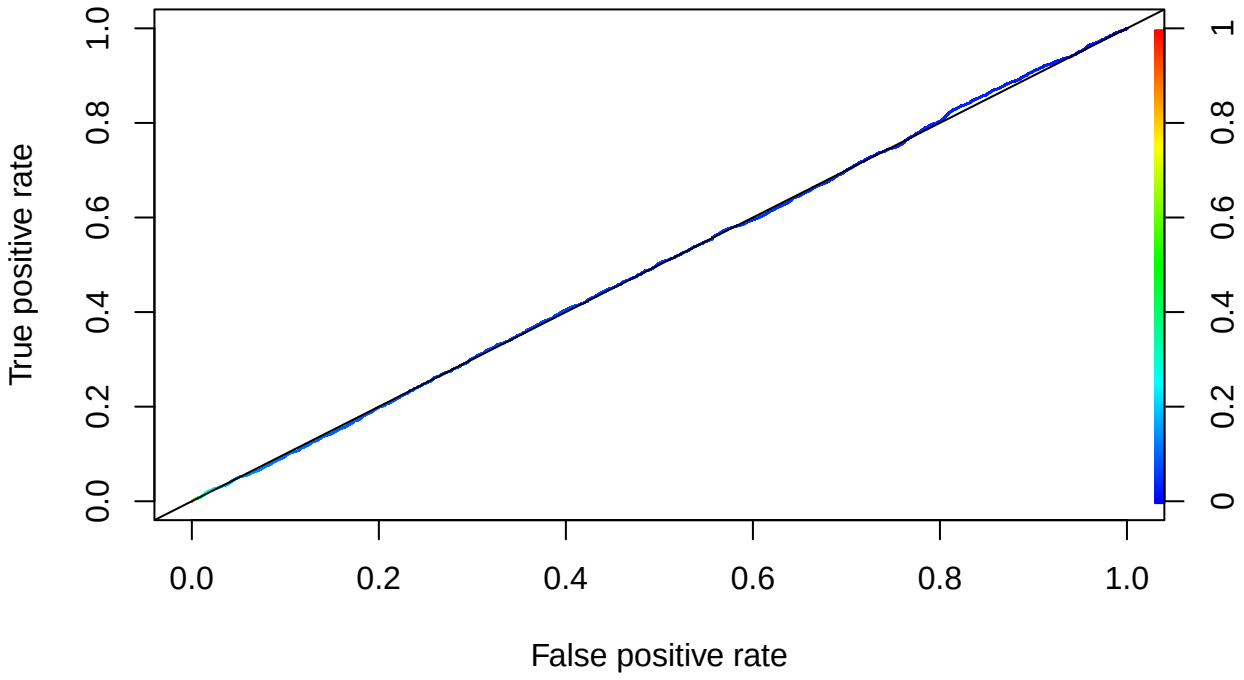
```
pred_m2 = prediction(result_m2, test.y)


plot(performance(pred_m2, "acc")) #It seems like 0.52 cutoff has the highest accuracy
```




```
table(test.y, result_m2>0.2)
```

```
##
## test.y FALSE  TRUE
##      0 43496  1218
##      1  3299    95
```

```
#Accuracy :
#Sensitivity :
#Specificity :
#The Specificity and accuracy improved a bit compared to the previous model without interaction term, s
plot(performance(pred_m2,"tpr","fpr"), colorize=T)
abline(0,1)
```




```
#Now we calculate the area under the curve (AUC) and accuracy of the model given above (glmModel2)
auc_ROCR2 <- performance(pred_m2, measure = "auc")
auc_ROCR2@y.values[[1]]
```

```
## [1] 0.5011228
```

Not Sure if we use this!!

```
# Check how the model fits the data
# From model.final, we can calculate the difference in the two deviances from the summary(model): 987.2
#Number of regressors in the model: 813-802=11
pchisq(473.75,12)
```

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

```
#The area below 473.75 is one which means the area above it is almost zero. This means that our model h
print(paste("Pearson's X^2 =",round(sum(residuals(model.final,type="pearson")^2),3)))
```

```
## [1] "Pearson's X^2 = 224463.067"
```

```
qchisq(0.95,802)
```

```
## [1] 868.9936
```

```
#781.61<868.99, so we fail to reject the null hypothesis and conclude that the logistic model fits the
```

```
sum(cs_train[,2])
```

```
## [1] 4963
```

```r
sum(cs_test[,2])
```

```
## [1] 3394
```

```r
training_dataset <- data.frame(train.x, train.y)
kmeans_model_train  <- kmeans(training_dataset, centers = 5) # centers because paper had 5 clusters
kmeans_model_train$centers #what each cluster's average values are for each variable. The most obvious
```

```
##   RevolvingUtilizationOfUnsecuredLines      age
## 1                        14.265756681 53.57759
## 2                         3.646322114 50.61780
## 3                         0.269884828 53.97500
## 4                         0.003664065 60.00000
## 5                         0.219734040 56.33333
##   NumberOfTime30.59DaysPastDueNotWorse    DebtRatio MonthlyIncome
## 1                           0.2564584  0.291091129      12853.40
## 2                           0.4181155 33.562334538       4521.34
## 3                           0.2000000  0.097257466      83362.68
## 4                           0.0000000  0.002122935    2401405.00
## 5                           0.3333333  0.003112791     632973.22
##   NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 1                       10.689998              0.05135423
## 2                        8.199739              0.25641942
## 3                       11.825000              0.10000000
## 4                       12.500000              0.00000000
## 5                       10.111111              0.00000000
##   NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 1                    1.6594733                            0.05235504
## 2                    0.8781695                            0.22568483
## 3                    2.3650000                            0.04000000
## 4                    1.0000000                            0.00000000
## 5                    0.8888889                            0.22222222
##   NumberOfDependents    train.y
## 1          1.1534997 0.04753863
## 2          0.7684006 0.07483516
## 3          1.2100000 0.07500000
## 4          1.5000000 0.00000000
## 5          0.8888889 0.00000000
```

```r
#Cluster 3 and 4 are different than Cluster 1 and 2 in terms of the Response Variable.
```

```r
testing_data <- data.frame(test.x,test.y)
kmeans_model_test  <- kmeans(testing_data, centers = 5) #testing to see if test data has similar cluste
kmeans_model_test$centers #Clusters are grouped as (1,2), (3,5) and (4) for Response Variables.
```

```
##   RevolvingUtilizationOfUnsecuredLines      age
## 1                           0.3164407 44.00000
## 2                           9.8305759 53.44267
## 3                          29.5137331 52.74916
## 4                           4.0291137 50.47701
## 5                           0.1574056 51.64286
##   NumberOfTime30.59DaysPastDueNotWorse   DebtRatio MonthlyIncome
## 1                           0.5000000  0.00427481   1316300.000
## 2                           0.2478626  0.31004064     11403.951
## 3                           0.2608696  0.12796824     48432.478
```

```
## 4                           0.4335168 37.81703238      4234.529
## 5                           0.5000000  0.01222468    222821.143
##   NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
## 1                       10.500000              0.00000000
## 2                       10.592443              0.05931298
## 3                       11.120401              0.04347826
## 4                        8.038423              0.27400340
## 5                        8.928571              0.07142857
##   NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
## 1                     3.000000                           0.00000000
## 2                     1.595191                           0.05885496
## 3                     1.903010                           0.05685619
## 4                     0.841063                           0.23993313
## 5                     1.214286                           0.07142857
##   NumberOfDependents     test.y
## 1          1.5000000 0.00000000
## 2          1.1265649 0.04458015
## 3          1.3311037 0.03678930
## 4          0.7371804 0.08065028
## 5          1.3571429 0.07142857
```

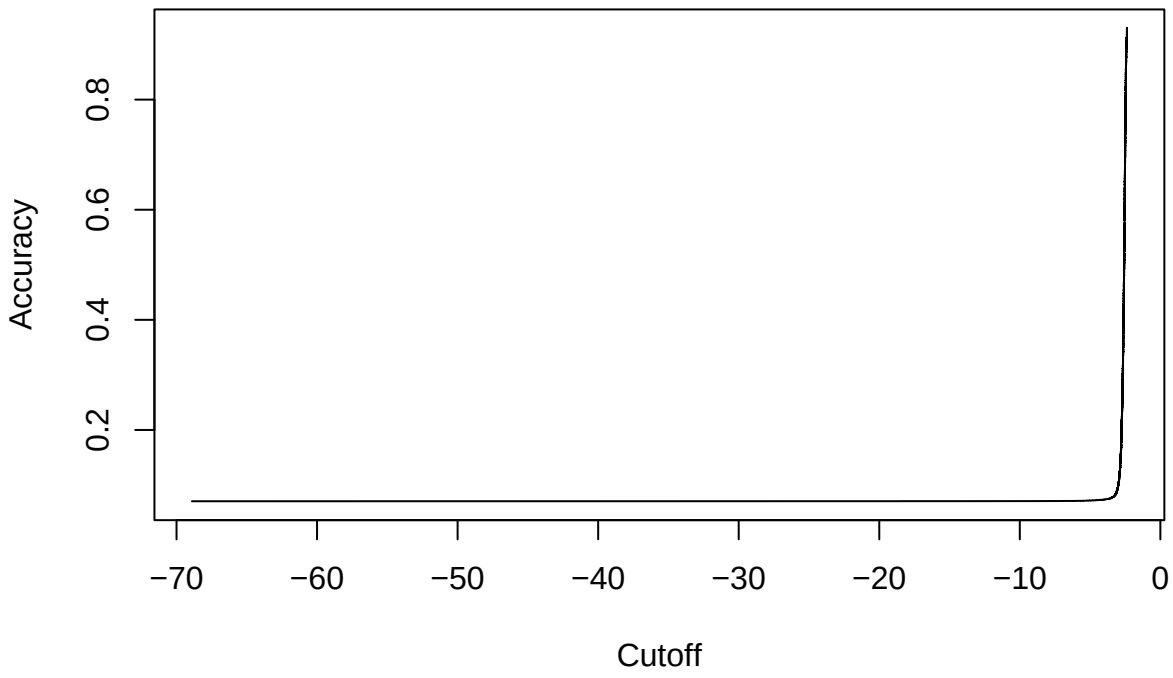
```
#predict(kmeans_model_train, newdata=test.x, type="response")
prediction_result <- predict.glm(glm_pca_original_data, newdata = testing_data)
```

```
## Warning: 'newdata' had 48108 rows but variables found have 72161 rows
```

```
prediction_result <- prediction_result[1:length(test.y)]
head(test.y)
```

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

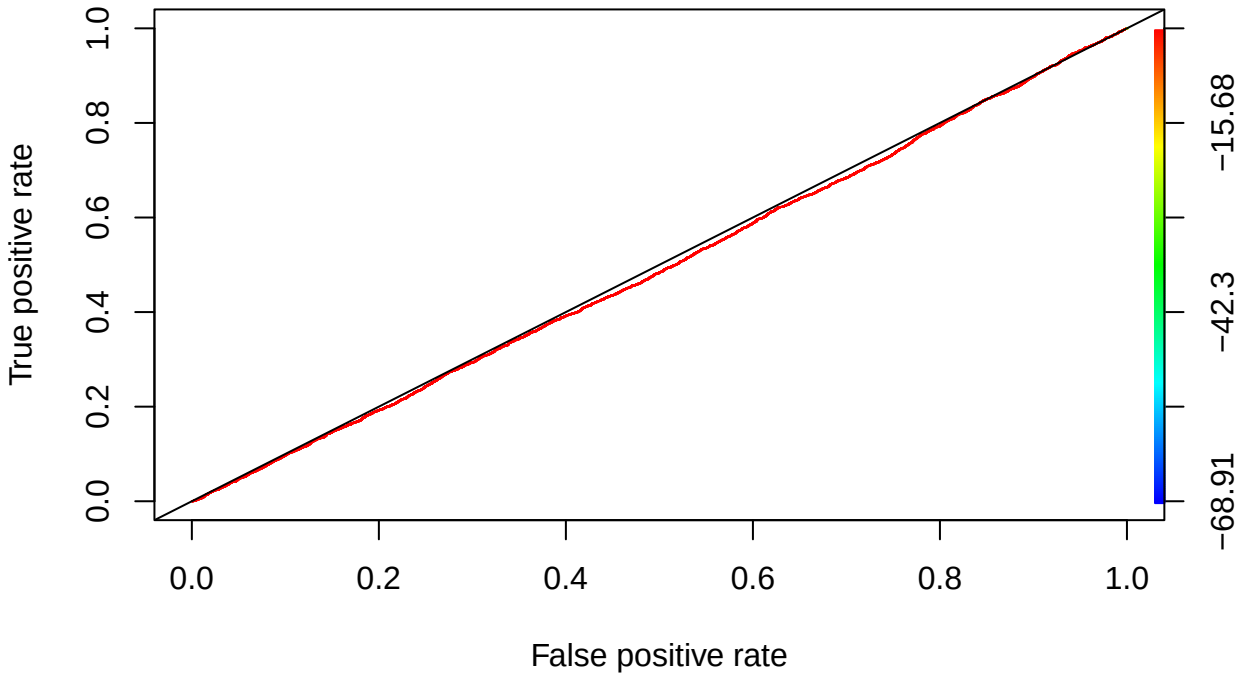
```
prediction_pca = prediction(prediction_result, test.y)
plot(performance(prediction_pca, "acc"))
```




```
table(test.y, result_m2>0.2)
```

```
##
## test.y FALSE  TRUE
##      0 43496  1218
##      1  3299    95
```

```
plot(performance(prediction_pca,"tpr","fpr"), colorize=T)
abline(0,1)
```




```
pca_auc_ROCR2 <- performance(prediction_pca, measure = "auc")
pca_auc_ROCR2@y.values[[1]]
```

```
## [1] 0.492578
```