Machine Learning - LC loans

Svetlana Mikheyenok

March 5, 2017

Contents

Introduction

The dataset I used for this project was obtained from LendingClub (LC), a peer to peer lender located in San Francisco, California. The data is representative of LC's initial loan issuances from 2007 through 2011. I thought it would be interesting to look into these specific years to see if there were any significant trends in predicting whether a borrower would pay off their loan or not (paid off or charged off respectively). As I will take you through the machine learning process we will be able to see the final results which will define whether or not this may have been useful for the company to utilize. Coming from loan originating, if the company was able to identify key characteristics that reflect a common trait of an individual who will most likely charge off their loan, then the underwriters and loan analysts could do an even better job in weeding these people out. This in turn would not only increase the revenue generated by the company but considering this is a peer to peer lender, this too will benefit investors who are looking to make a consistent return.

Exploratory Data Analysis

The raw data contained 42,542 observations wiht 111 features. I created a subset with data including features only relevant to my response outcome bringing the total number of features down to 26.

Data Cleaning

```
data<-subset(rawdata,select=c(loan_amnt,funded_amnt,funded_amnt_inv,term,int_rate,installment,grade,sub

data$had_delinq<-1
data[is.na(data$mths_since_last_delinq),]$had_delinq<-0
data$mths_since_last_delinq[is.na(data$mths_since_last_delinq)]<-0
```

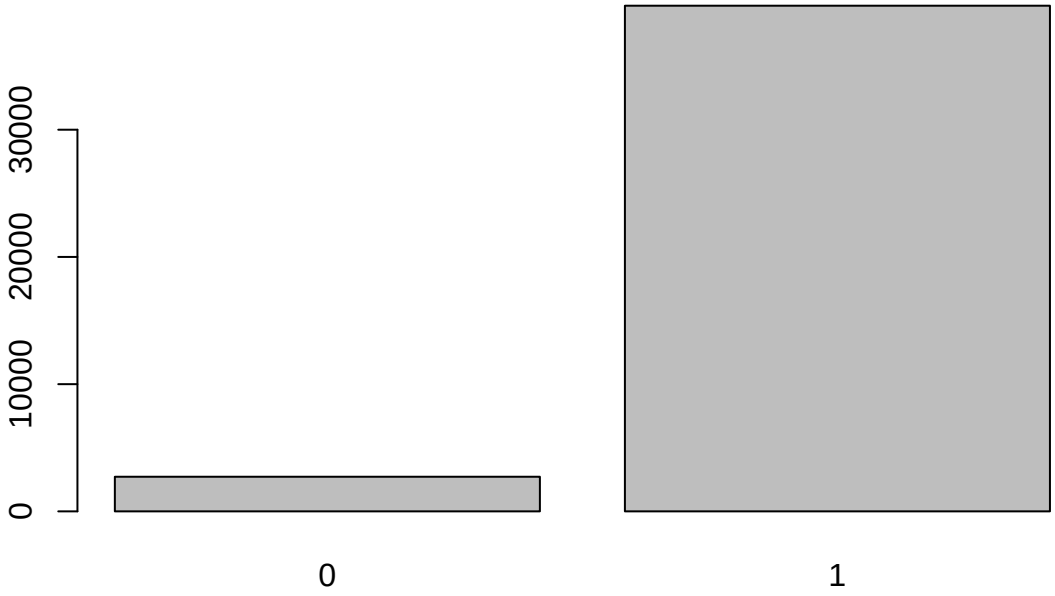
"Mthssincelastdelinq" had many N/A responses. Considering this variable is interpreted as the number of months since the borrowers last deliqnuency, these are N/A reponses which we do not want to remove as these are people with likely good credit. Instead, I created a new variabe "had_delinq" and set it to TRUE to represent the people who had ever gone delinquent, including those who were 'currently' delinquent. This way the 0 response between the N/A's and the currently delinquent borrowers was separated from one another. From doing this, I was able to see people who have a clean record without accidentally removing 26,933 entries. This produced 15,609 people who range from having their most recent delinquency almost half a year ago (120 months & up) till present day (date of collected data). With this set up, I removed actual NA observations, bringing down the observation count to 42,506. Then I went through each feature changing the amounts to purely be numeric such as removing the percentage signs as well as factors, in turn creating dummy variables.

```
data<-na.omit(data)
data$term<-as.numeric(gsub(" ","",gsub("months","",data$term)))
data$int_rate<-as.numeric(gsub("%","",data$int_rate))
data$grade<-as.factor(data$grade)
data$sub_grade<-as.factor(data$sub_grade)
data$emp_length<-as.factor(data$emp_length)
data$home_ownership<-as.factor(data$home_ownership)
data$verification_status<-as.factor(data$verification_status)
data$issue_d<-as.factor(substr(data$issue_d,5,6))

data$purpose<-as.factor(data$purpose)

data$addr_state<-as.factor(data$addr_state)
data$earliest_cr_line<-as.factor(substr(data$earliest_cr_line,5,5))

data$revol_util<-as.numeric(gsub("%","",data$revol_util))
```

Lastly, I removed all observations not inclusive of "Fully Paid" or "Charged Off", as these were the results I was in search of. The remainder of status's were removed considering on average each held only 1-5 observations. Additionally, I created a new feature, CP, indicative of whether the loan did or did not meet the credit policy requirements. As I would have expected majority of the loans: 39,697 fell into LendingClub's guidelines. The remaining 2,709 loans, totalling approximatley 6.4% were exceptions most likely made by a combination of management, capital markets and the underwriting teams.

```
data<-subset(data,!((data$loan_status=="Default")|(data$loan_status=="In Grace Period")|(data$loan_statu
data$CP<-1
data$CP[grep("Does not",data$loan_status)]<-0
data$CP<-as.factor(data$CP)
data$had_delinq<-as.factor(data$had_delinq)

data$funded_inv <- data$funded_amnt_inv
data$earliest_line <- data$earliest_cr_line
```

```
data$inq_l6months <- data$inq_last_6mths
data$mths_sincedelinq <- data$mths_since_last_delinq

data$loan_status[grep("Does not",data$loan_status)]<-gsub("Does not meet the credit policy. Status:","
data$loan_status<-as.factor(data$loan_status)
freq(data$CP)
```




```
## data$CP
##       Frequency Percent
## 0          2720   6.405
## 1         39747  93.595
## Total     42467 100.000
```

```
data<-na.exclude(data)
N<-nrow(data)
set.seed(1234)
```

The Data Explained

Variables utilized in the Supervised Learning dataset

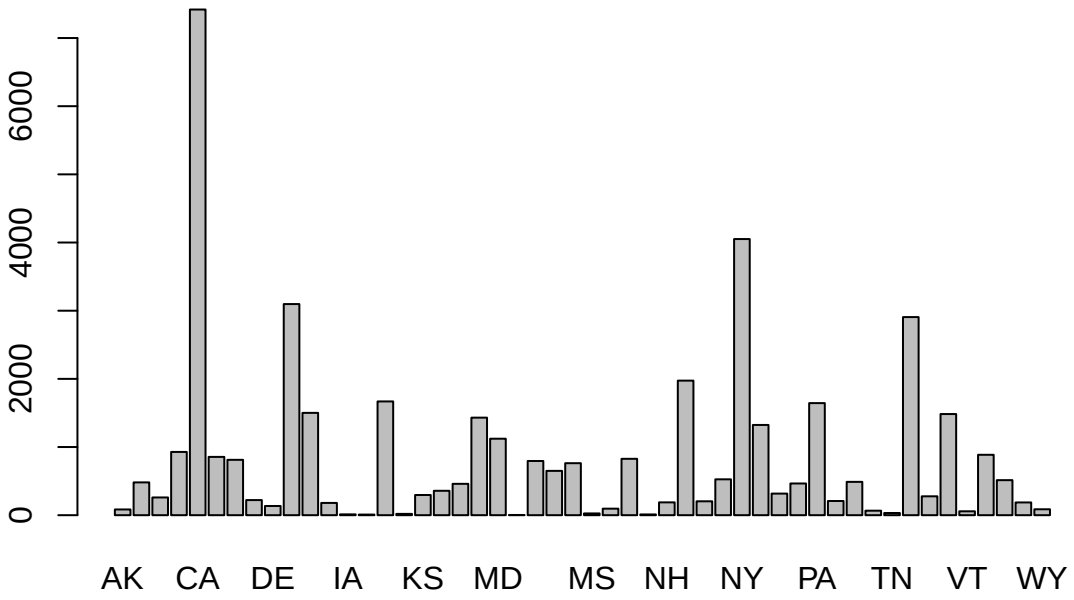
- Loan_Amnt::: The listed amount of the loan applied for by the borrower. If at some point in time, the credit # department reduces the loan amount, then it will be reflected in this value.
- funded_amnt:: The total amount committed to that loan at that point in time.
- funded_inv: The total amount committed by investors for that loan at that point in time.
- term: The number of payments on the loan. Values are in months and can be either 36 or 60.
- int_rate: Interest rate on the loan
- installment: The monhtly paymented owed by the borrower if the loan originates (when borrower is approved for the loan)

- grade: LendingClub assigned loan grade (associated with an interest rate, modeled return range, modeled default range, and origination fee. The loan grades range from "A", which is the lowest modeled risk/return, to "G", which is the highest modeled risk/return.)
- sub_grade: LendingClub assigned subgrade (incremental from A1 to G5)
- emp_length: Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
- home _ownership: The home ownership status provided by the borrower during registration. Value options: RENT, OWN, MORTGAGE, OTHER.
- annual_incom: The self-reported annual income provided by the borrower during registration.
- verification_status: Indicates if the co-borrowers' joint income was verified by LendingClub, not verified, or if the income source was verified
- issue_d: The month which the loan was funded
- loan_status: Current status of the loan
- purpose: A category provided by the borrower for the loan request.
- addr_state: The state provided by the borrower in the loan application
- dti: (debt-to-income) A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LendingClub loan, divided by the borrower's self-reported monthly income.
- earliest_line: The month the borrower's earliest reported credit line was opened
- inq_l6mths: The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
- mths_sincedelinq: The number of months since the borrower's last delinquency.
- open_acc: The number of open credit lines in the borrower's credit file.
- pub_rec: Number of derogatory public records
- revol_bal: Total credit revolving balance
- revol_util: Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
- total_acc: The total number of credit lines currently in the borrower's credit file
- CP: Whether the originated loan fell into the guidelines of LendingClub's credit policy or not. Return options are 0 or 1.

The graphs below are representative of all the features considered in the dataset used in the following predictive modeling. It gives a good overview of the frequency of attributes within the distribution of approved loans. Additionally, it allows us to get somewhat of a generalized picture as to the kind of borrowers LendingClub had from its starting years of 2007 - 2011.

Graphs for each explained variable

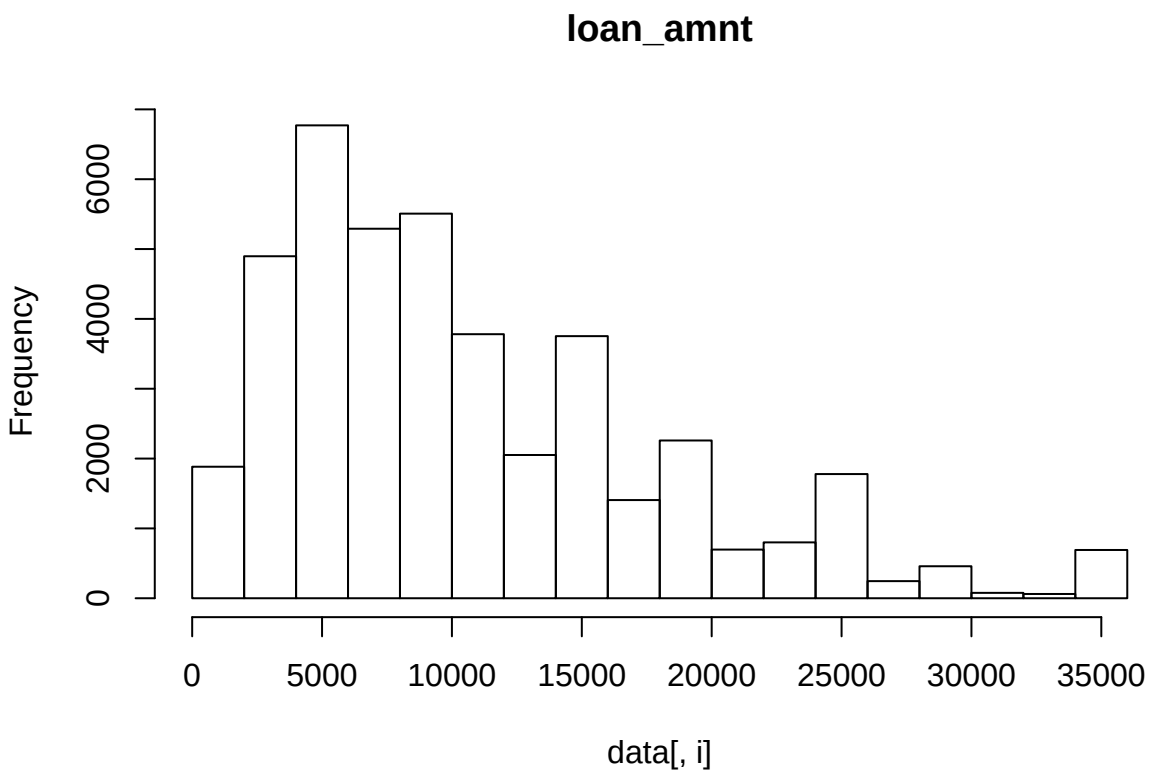
```
freq(data$addr_state)
```



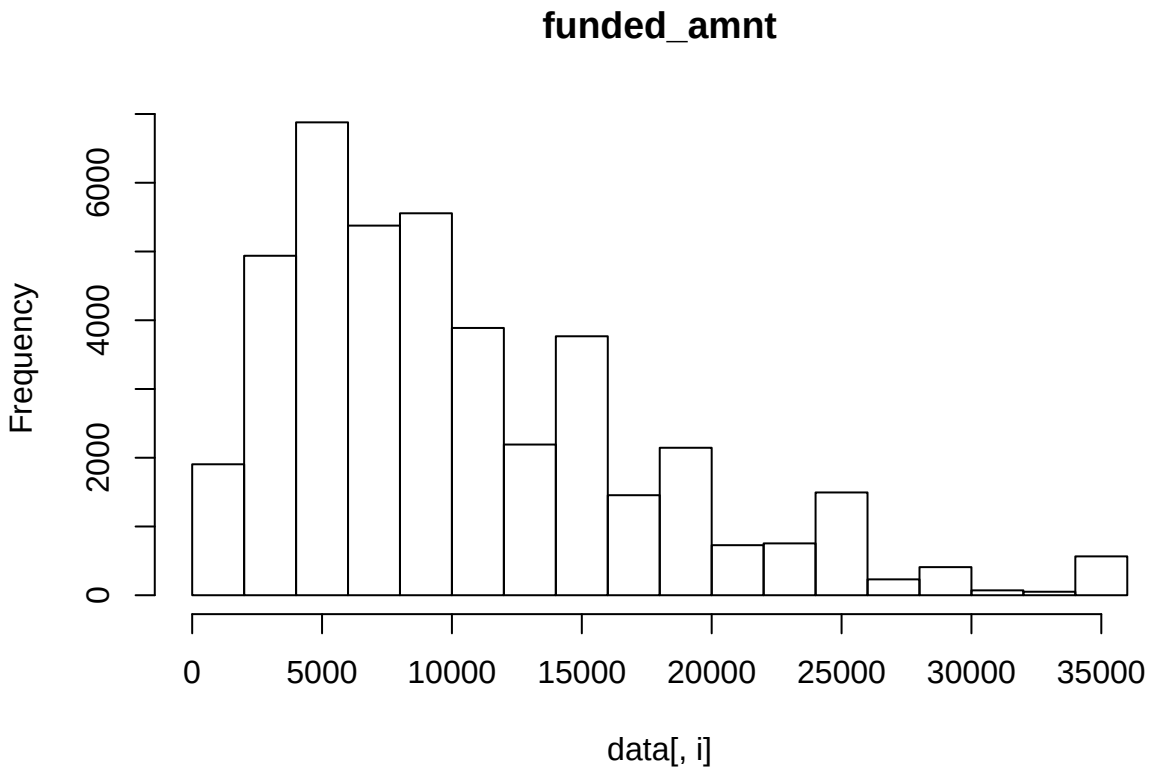

```
## data$addr_state
##       Frequency   Percent
## AK           84 1.981e-01
## AL          481 1.134e+00
## AR          260 6.131e-01
## AZ          928 2.188e+00
## CA         7418 1.749e+01
## CO          855 2.016e+00
## CT          812 1.915e+00
## DC          222 5.235e-01
## DE          135 3.184e-01
## FL         3097 7.303e+00
## GA         1501 3.540e+00
## HI          180 4.245e-01
## IA           12 2.830e-02
## ID            9 2.122e-02
## IL         1669 3.936e+00
## IN           19 4.480e-02
## KS          296 6.980e-01
## KY          358 8.442e-01
## LA          460 1.085e+00
## MA         1431 3.375e+00
## MD         1122 2.646e+00
## ME            3 7.074e-03
## MI          795 1.875e+00
## MN          650 1.533e+00
## MO          763 1.799e+00
## MS           26 6.131e-02
## MT           96 2.264e-01
## NC          827 1.950e+00
```

```
## NE           11 2.594e-02
## NH          188 4.433e-01
## NJ         1974 4.655e+00
## NM          204 4.811e-01
## NV          526 1.240e+00
## NY         4051 9.553e+00
## OH         1324 3.122e+00
## OK          317 7.475e-01
## OR          465 1.097e+00
## PA         1644 3.877e+00
## RI          208 4.905e-01
## SC          489 1.153e+00
## SD           66 1.556e-01
## TN           32 7.546e-02
## TX         2906 6.853e+00
## UT          277 6.532e-01
## VA         1484 3.500e+00
## VT           57 1.344e-01
## WA          886 2.089e+00
## WI          514 1.212e+00
## WV          187 4.410e-01
## WY           87 2.052e-01
## Total     42406 1.000e+02
```

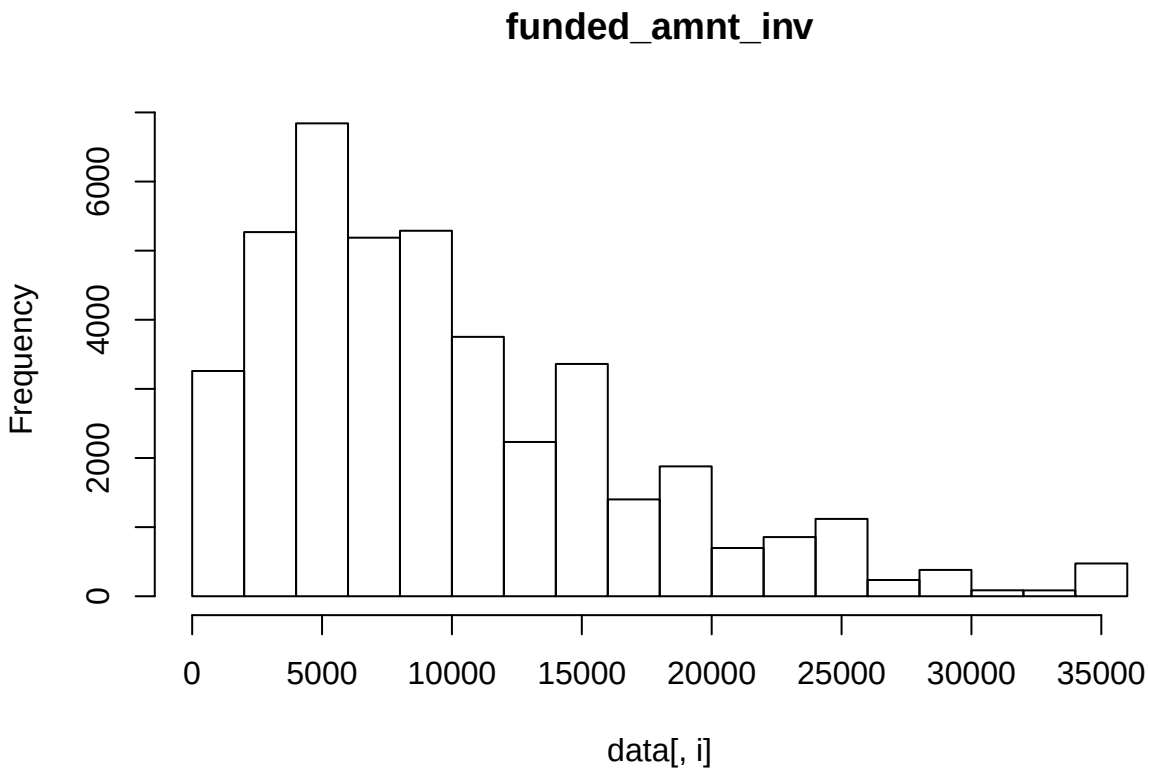
```
for(i in 1:ncol(data)){
  if(is.factor(data[,i])){
    freq(data[,i],main=colnames(data)[i])
    print(paste("freq(data[,",i,"],main=\"",colnames(data)[i],'")',sep=""))
  }else{

    hist(data[,i],main=colnames(data)[i])
    print(paste("hist(data[,",i,"],main=\"",colnames(data)[i],'")',sep=""))
  }
}
```



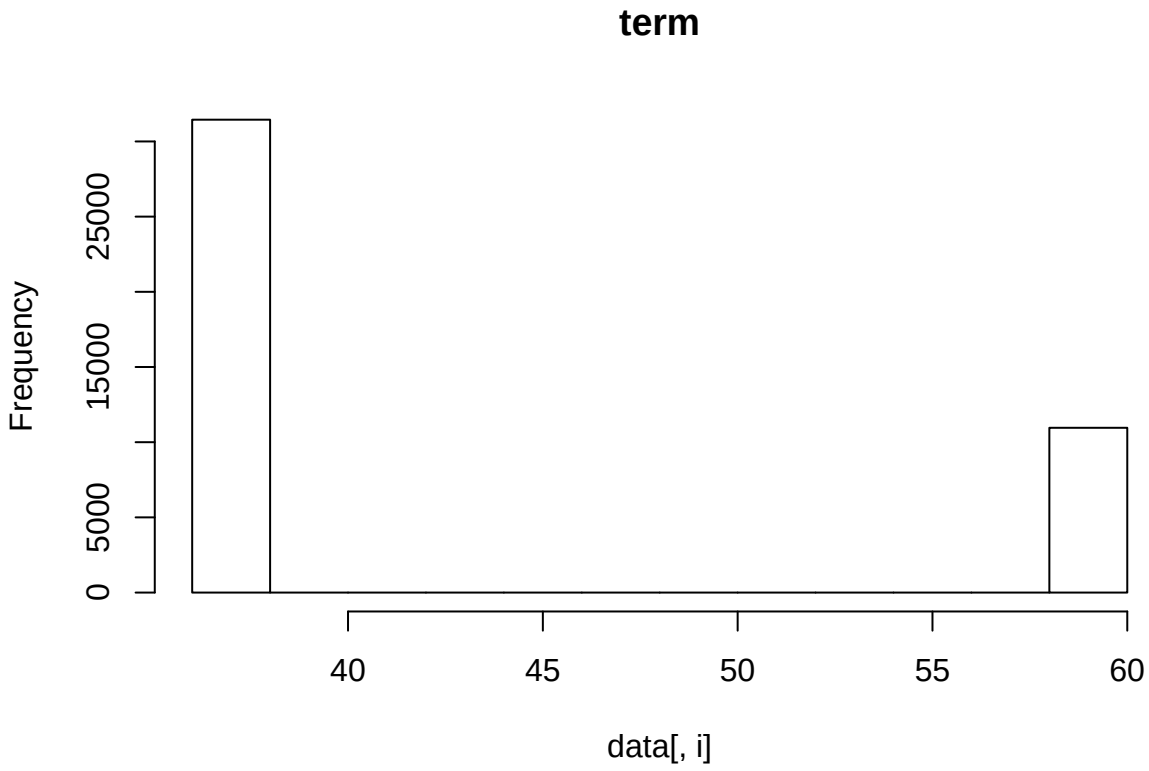

```
## [1] "hist(data[,1],main=\"loan_amnt\")"
```



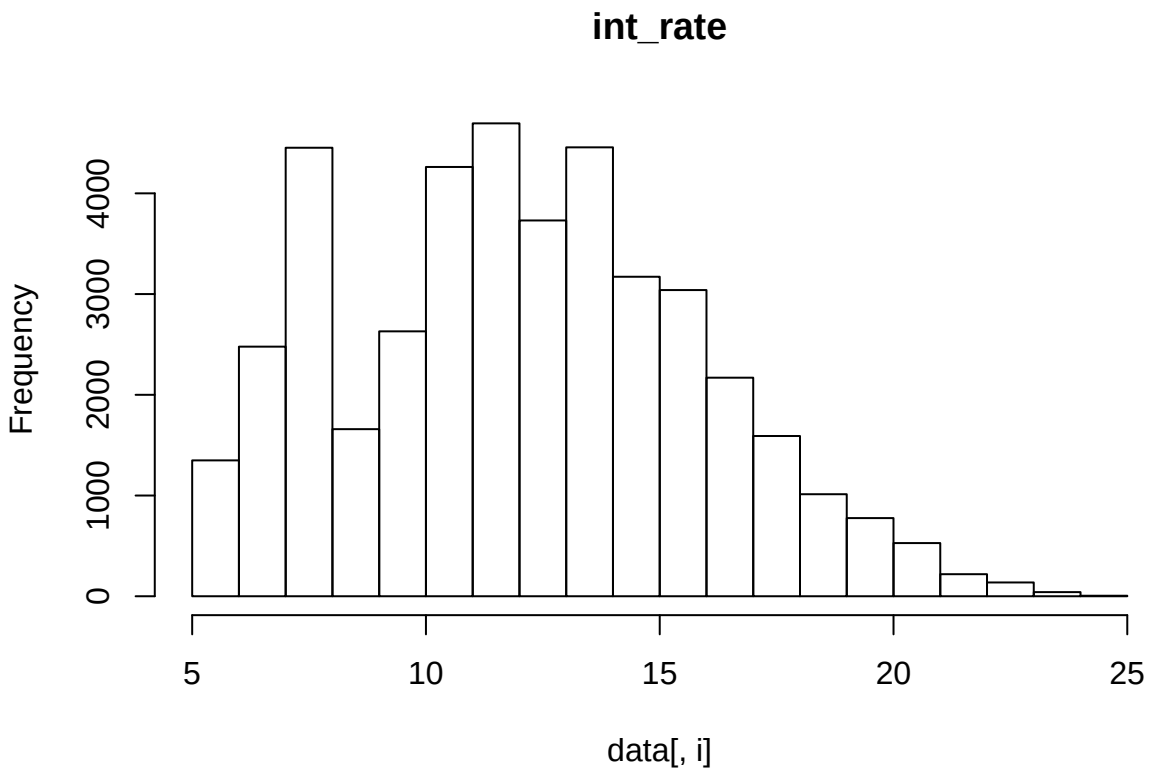

```
## [1] "hist(data[,2],main=\"funded_amnt\")"
```



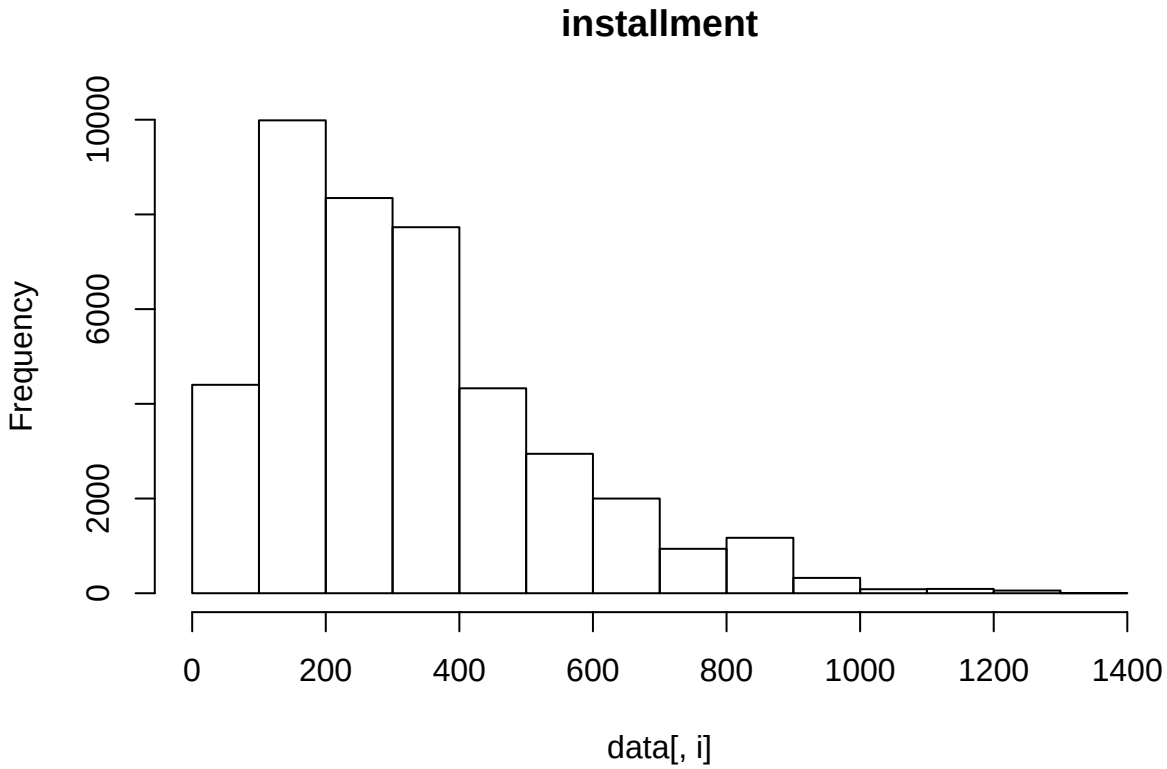

```
## [1] "hist(data[,3],main=\"funded_amnt_inv\")"
```



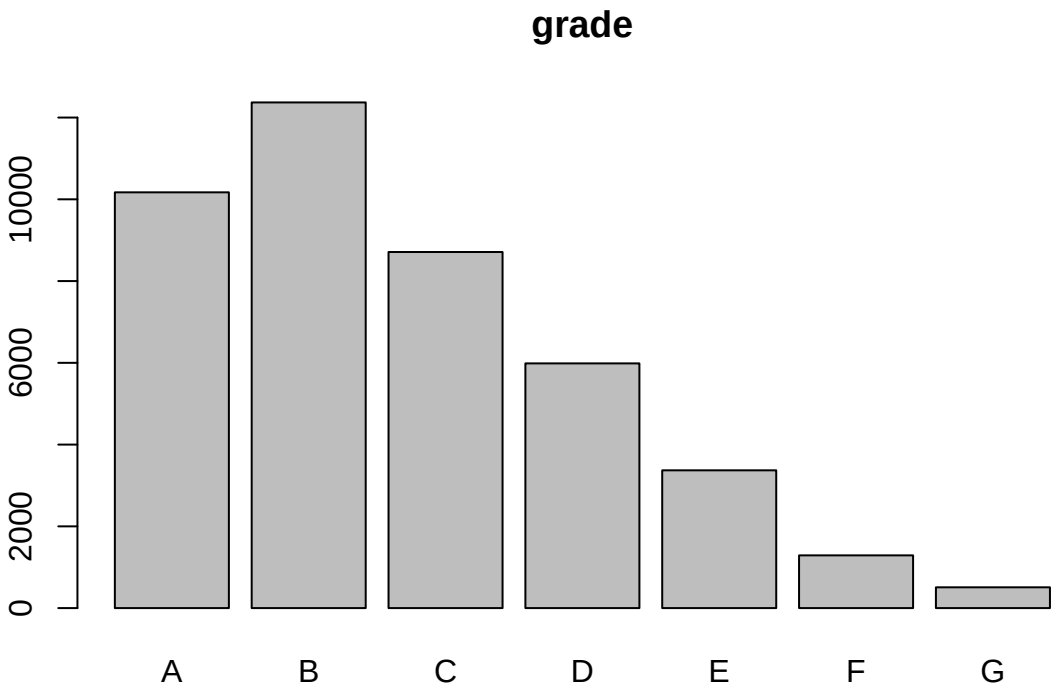

```
## [1] "hist(data[,4],main=\"term\")"
```



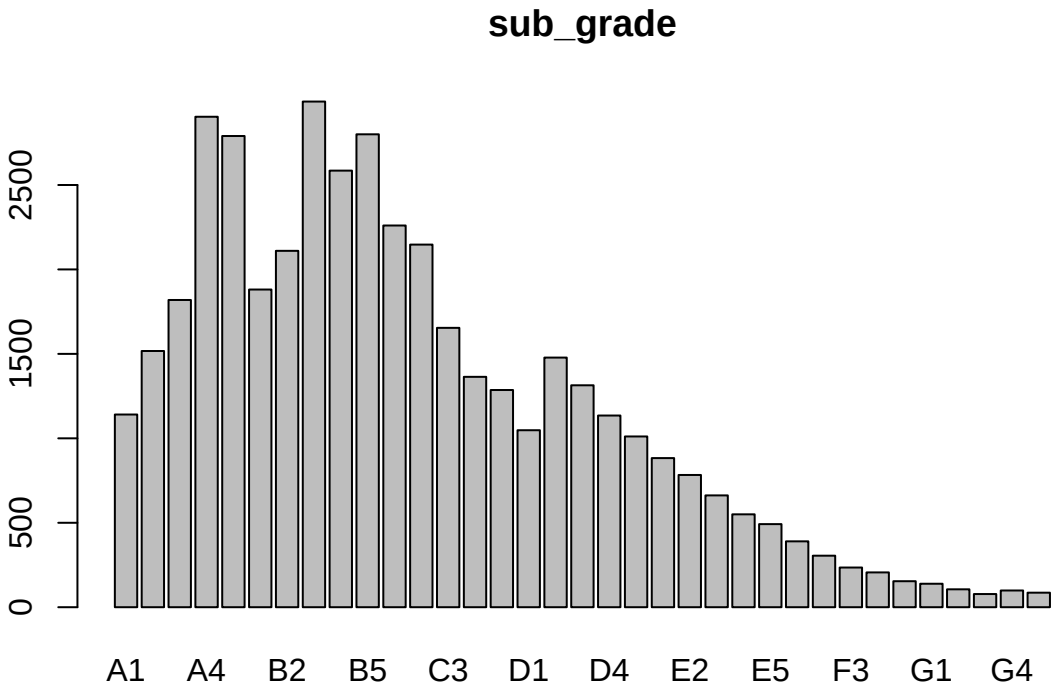

```
## [1] "hist(data[,5],main=\"int_rate\")"
```



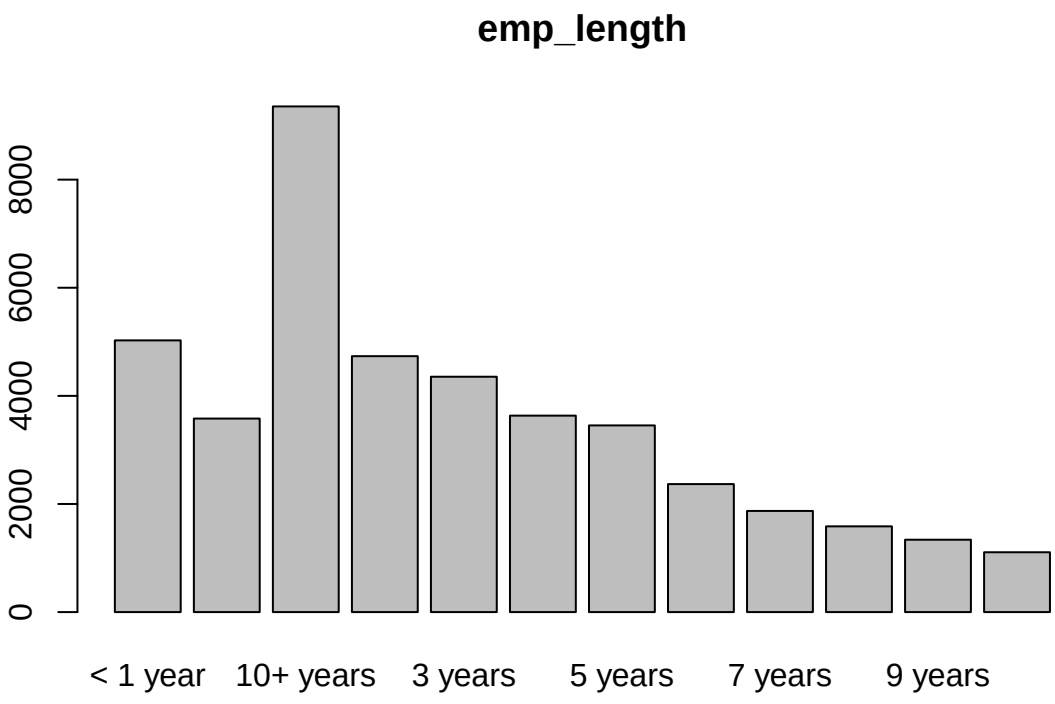

```
## [1] "hist(data[,6],main=\"installment\")"
```



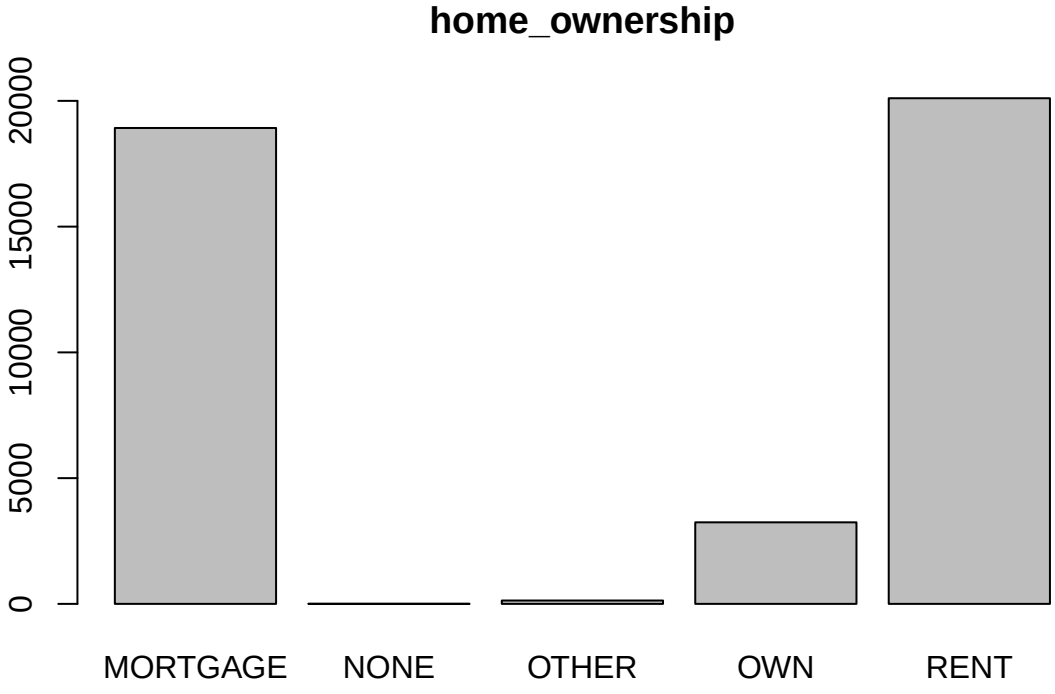

```
## [1] "freq(data[,7],main=\"grade\")"
```



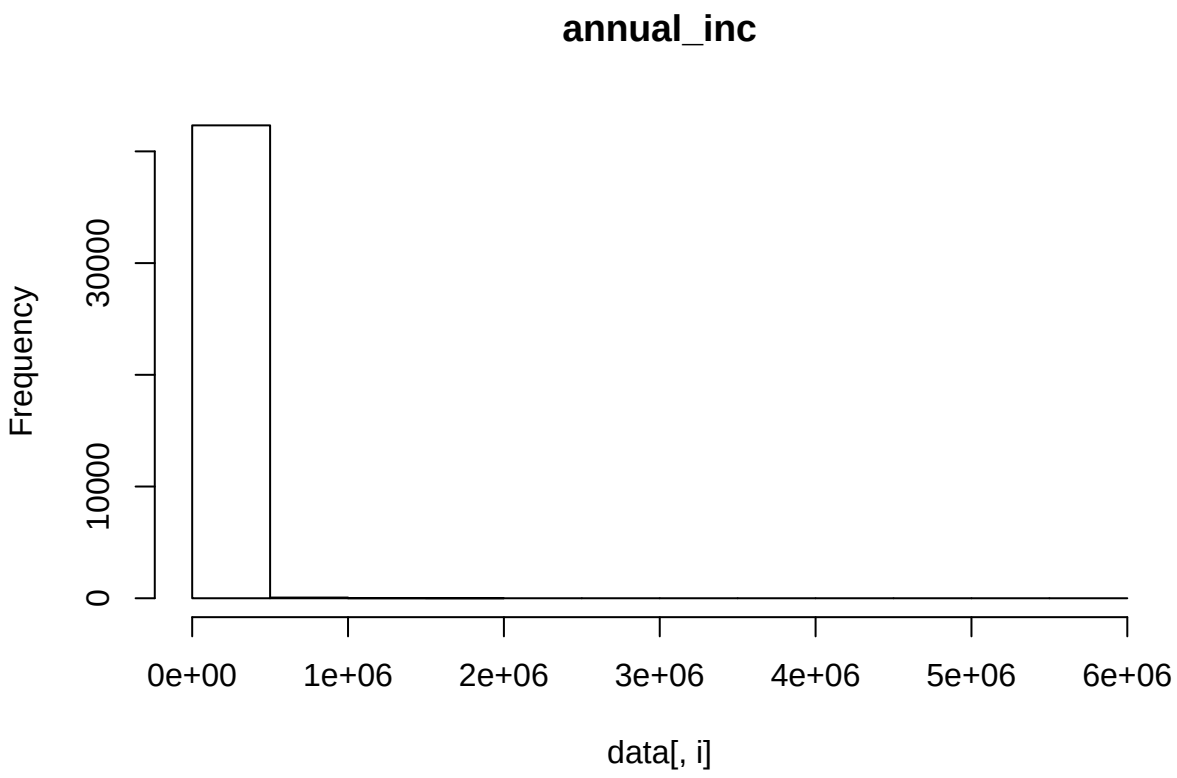

```
## [1] "freq(data[,8],main=\"sub_grade\")"
```



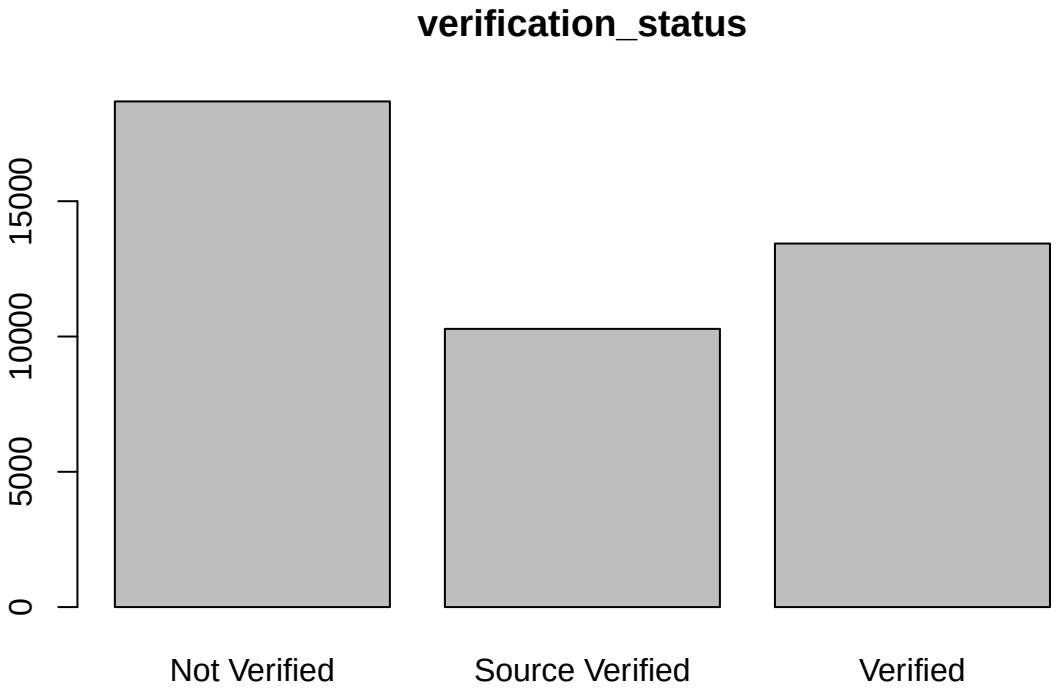

```
## [1] "freq(data[,9],main=\"emp_length\")"
```



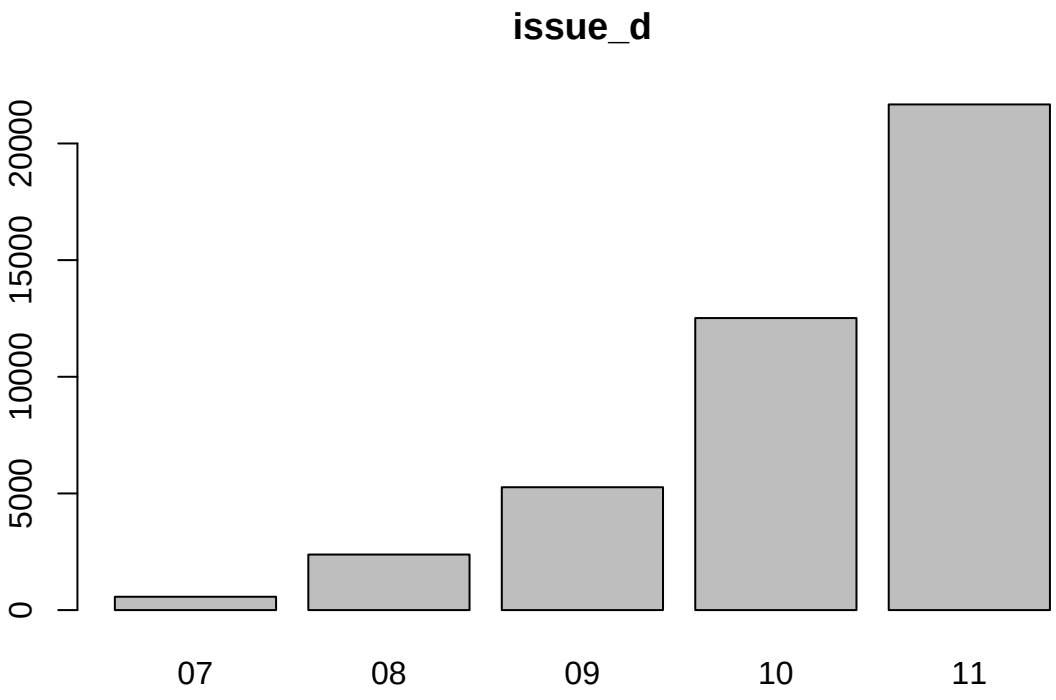

```
## [1] "freq(data[,10],main=\"home_ownership\")"
```



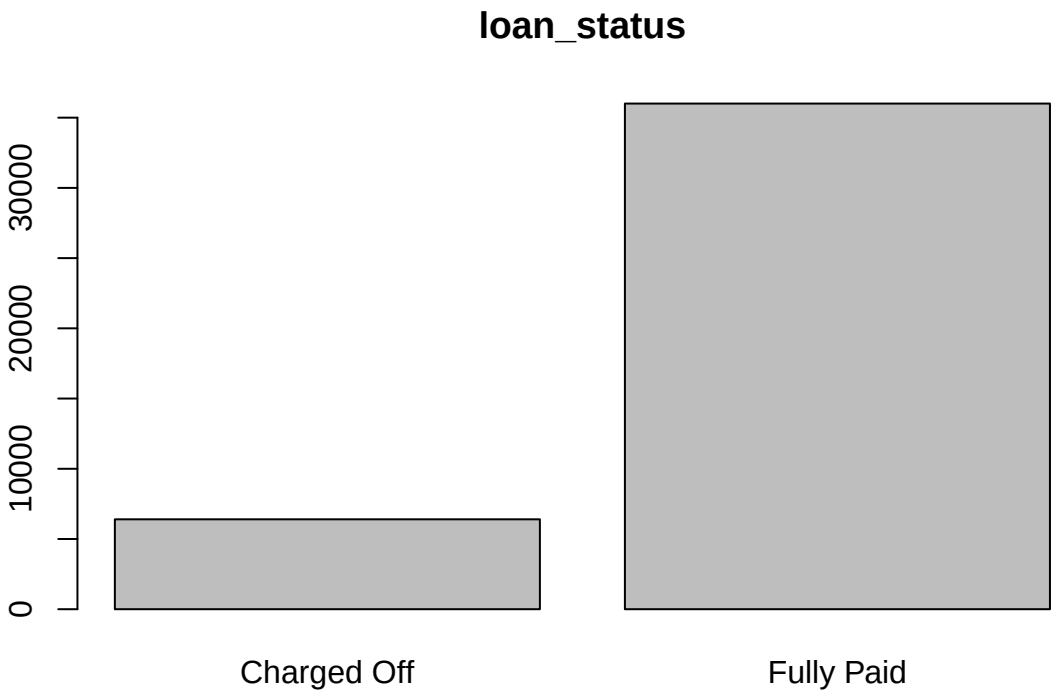

```
## [1] "hist(data[,11],main=\"annual_inc\")"
```



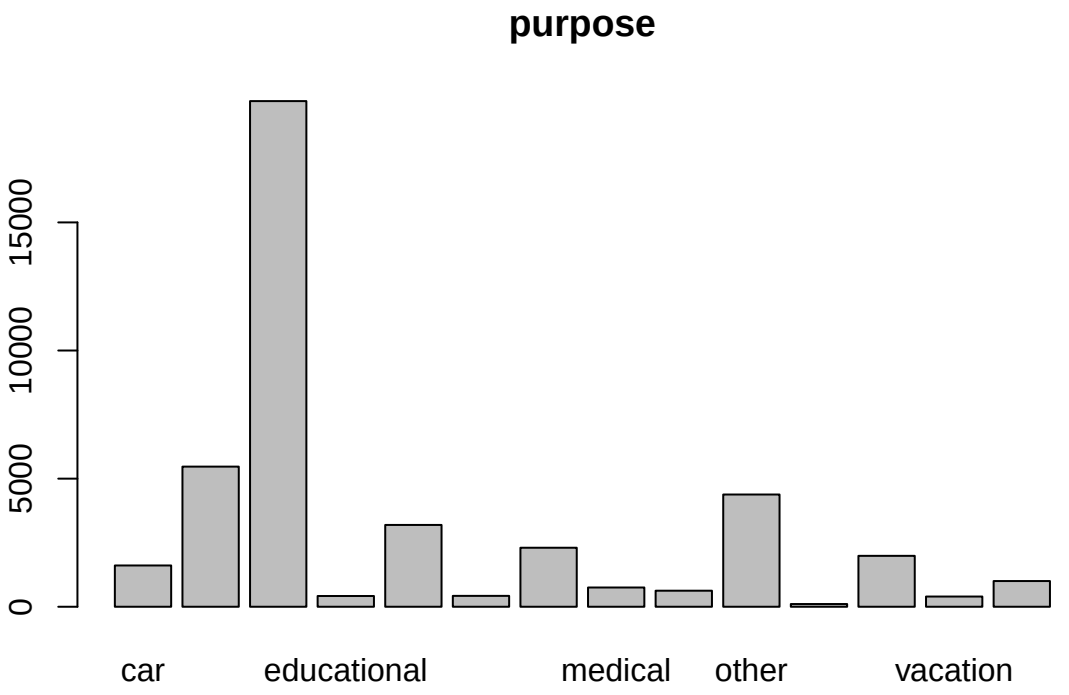

```
## [1] "freq(data[,12],main=\"verification_status\")"
```



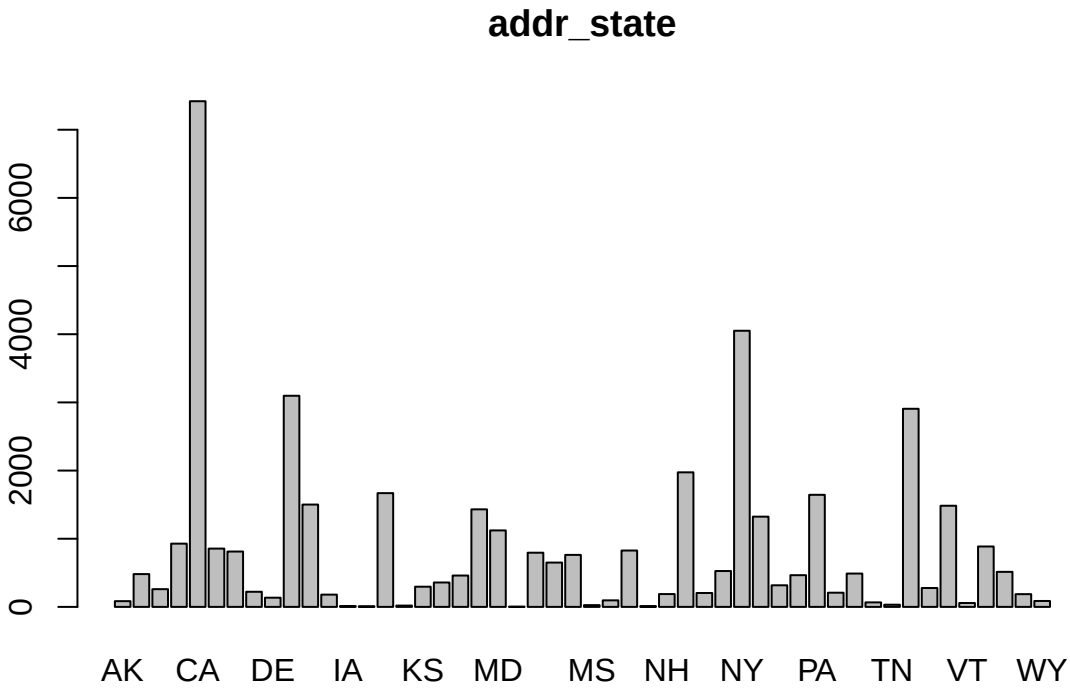

```
## [1] "freq(data[,13],main=\"issue_d\")"
```



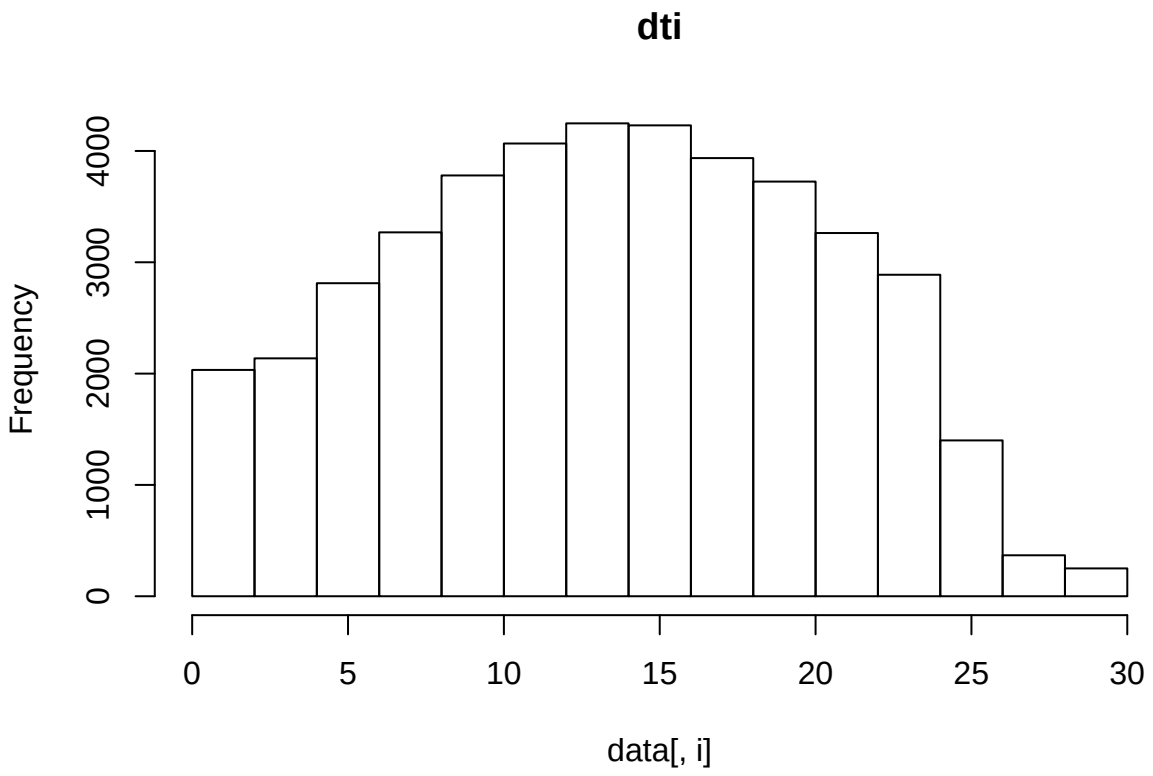

```
## [1] "freq(data[,14],main=\"loan_status\")"
```



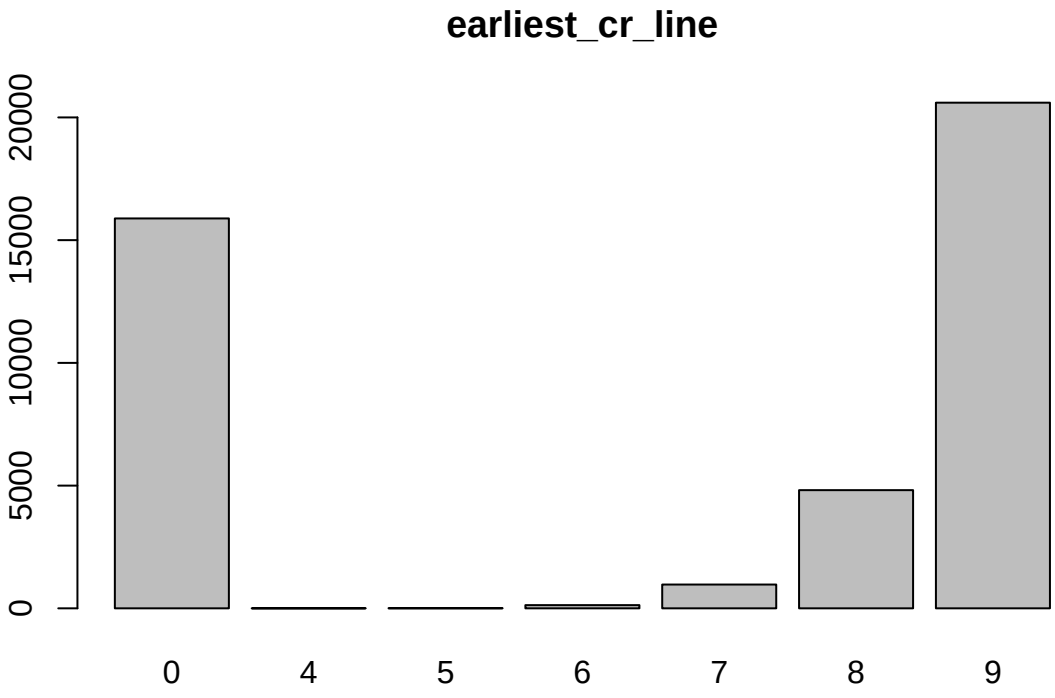

```
## [1] "freq(data[,15],main=\"purpose\")"
```



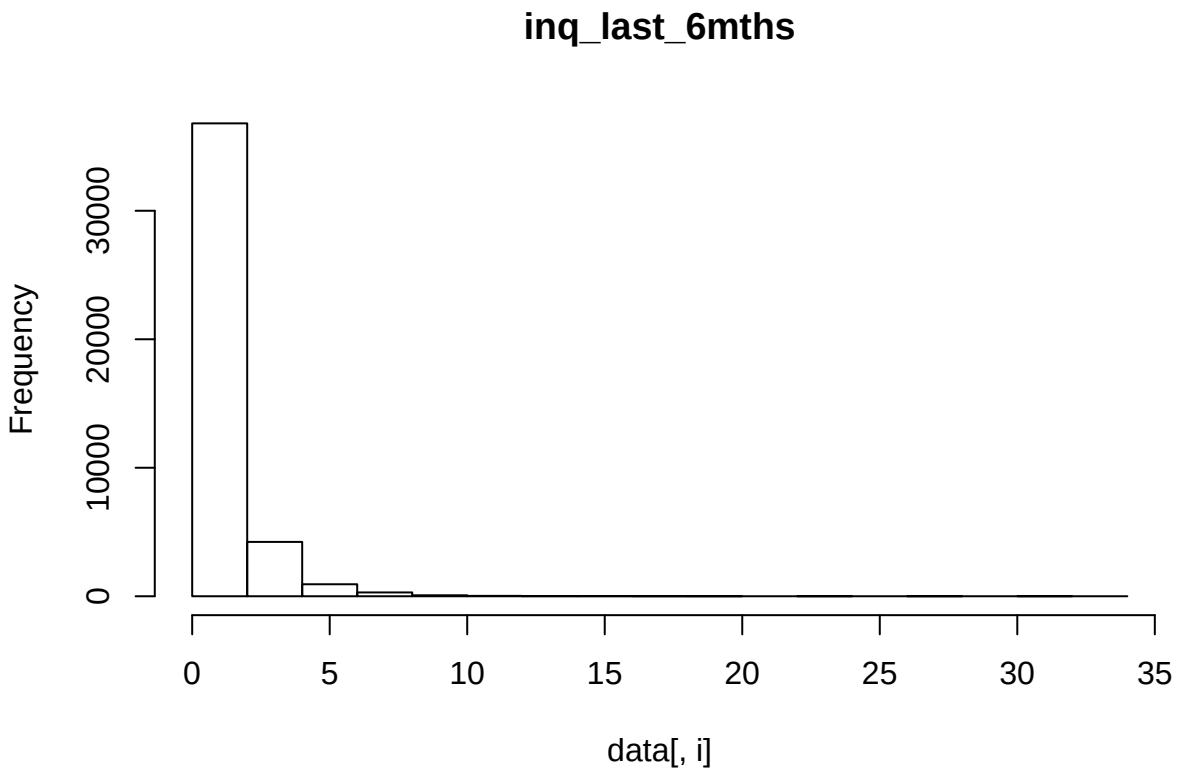

```
## [1] "freq(data[,16],main=\"addr_state\")"
```



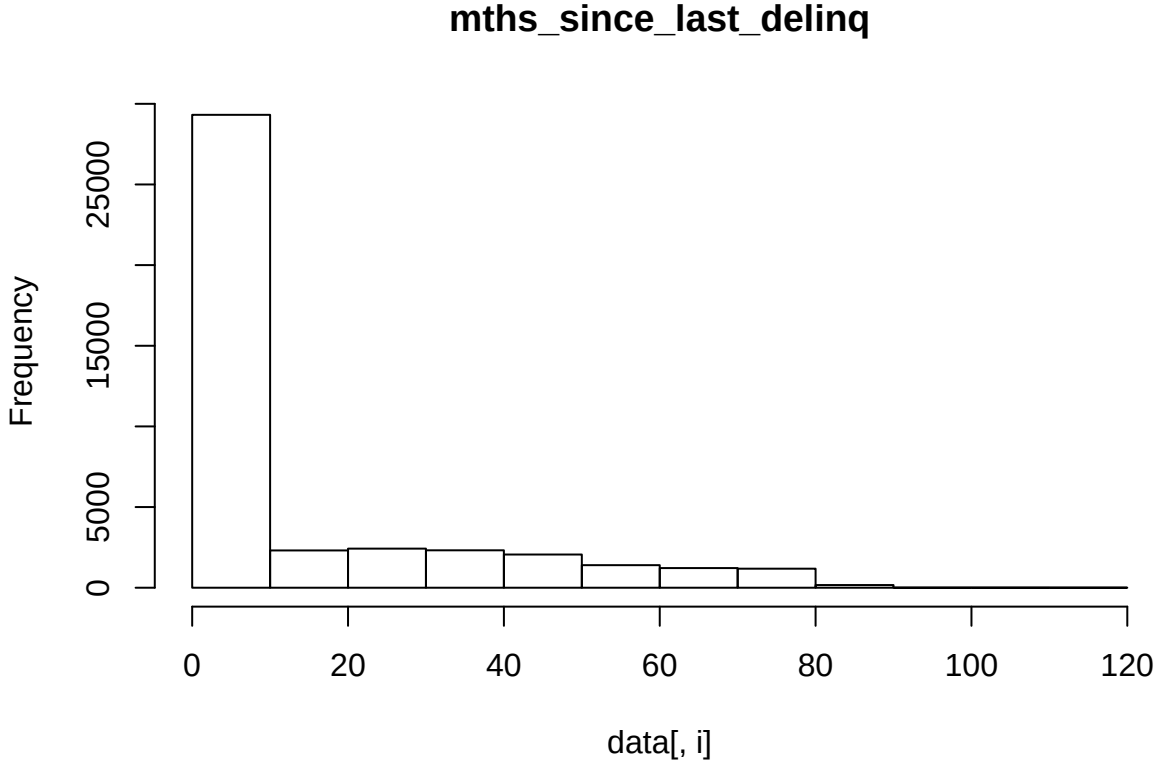

```
## [1] "hist(data[,17],main=\"dti\")"
```



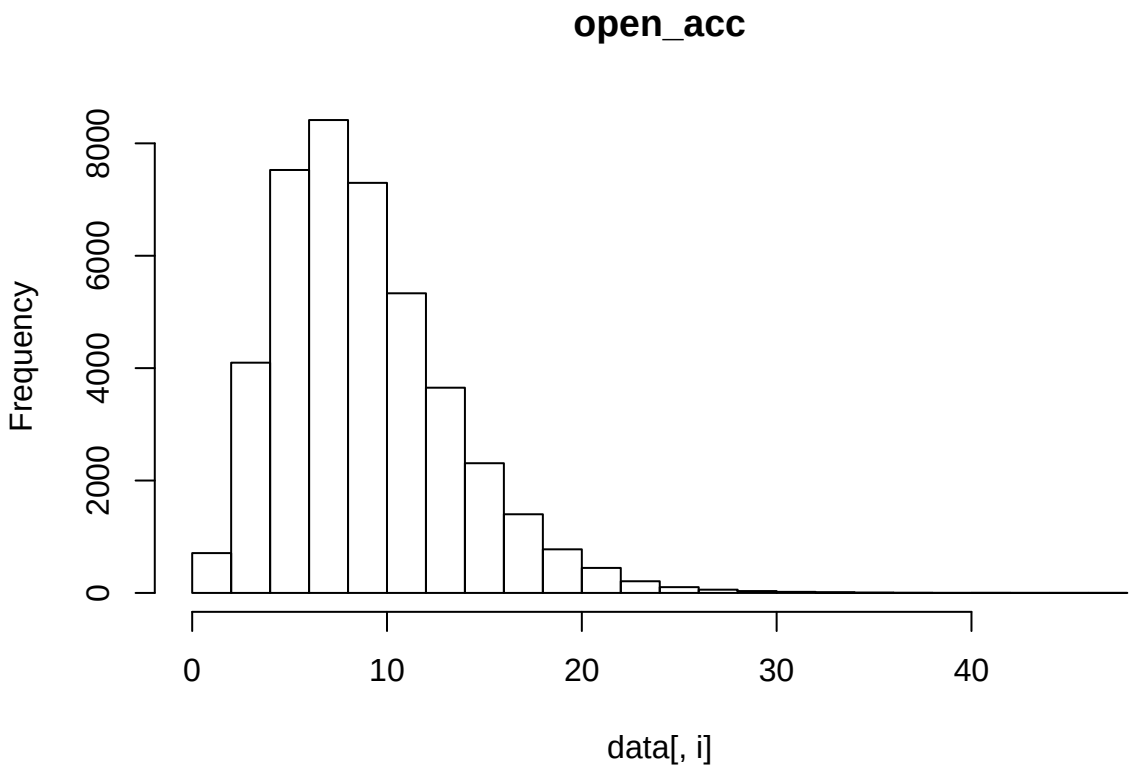

```
## [1] "freq(data[,18],main=\"earliest_cr_line\")"
```



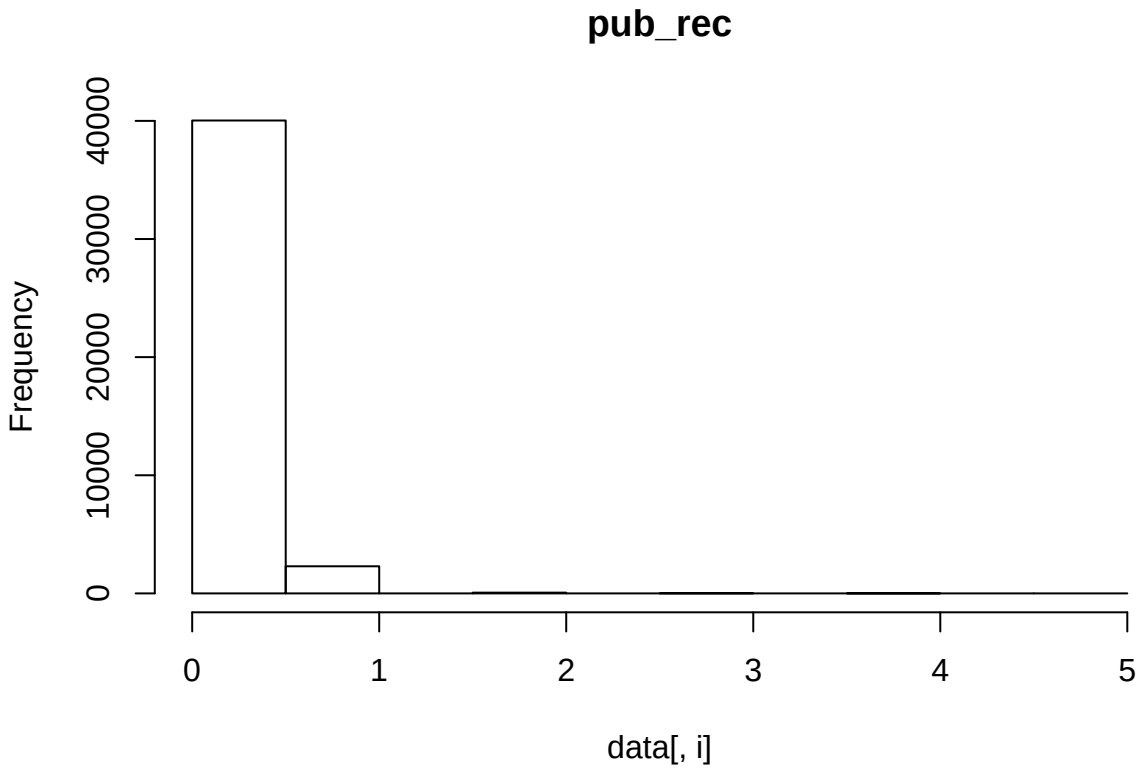

```
## [1] "hist(data[,19],main=\"inq_last_6mths\")"
```



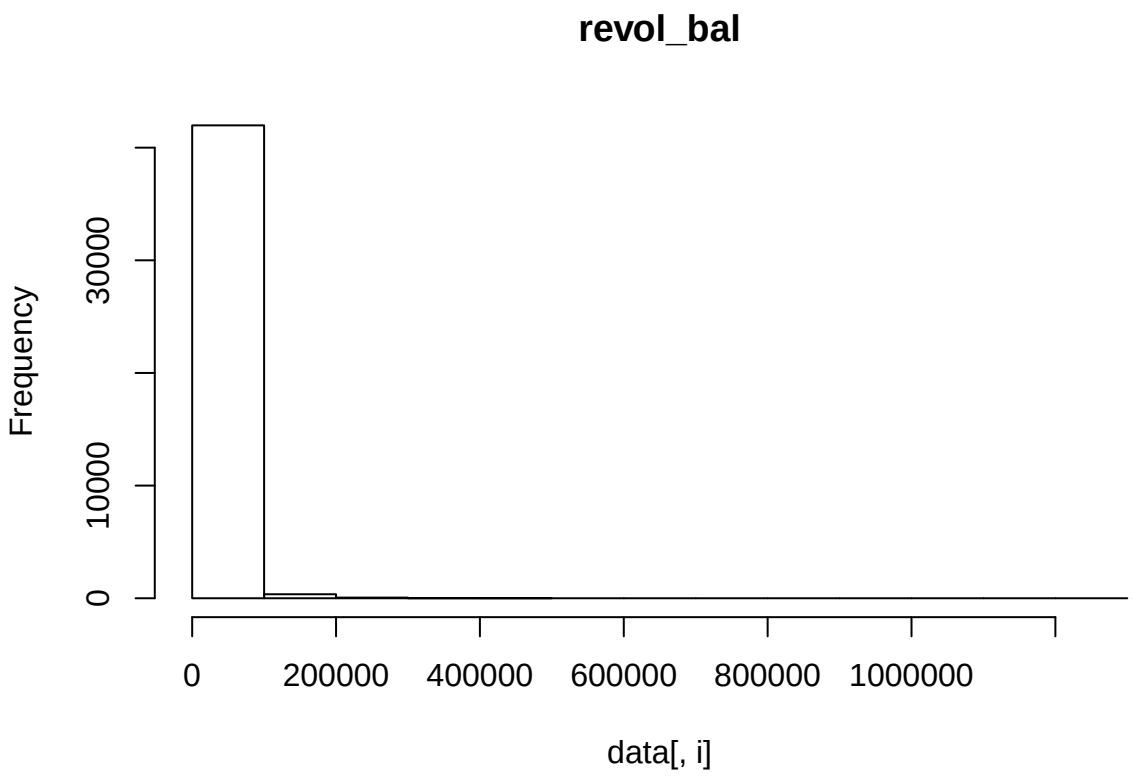

```
## [1] "hist(data[,20],main=\"mths_since_last_delinq\")"
```



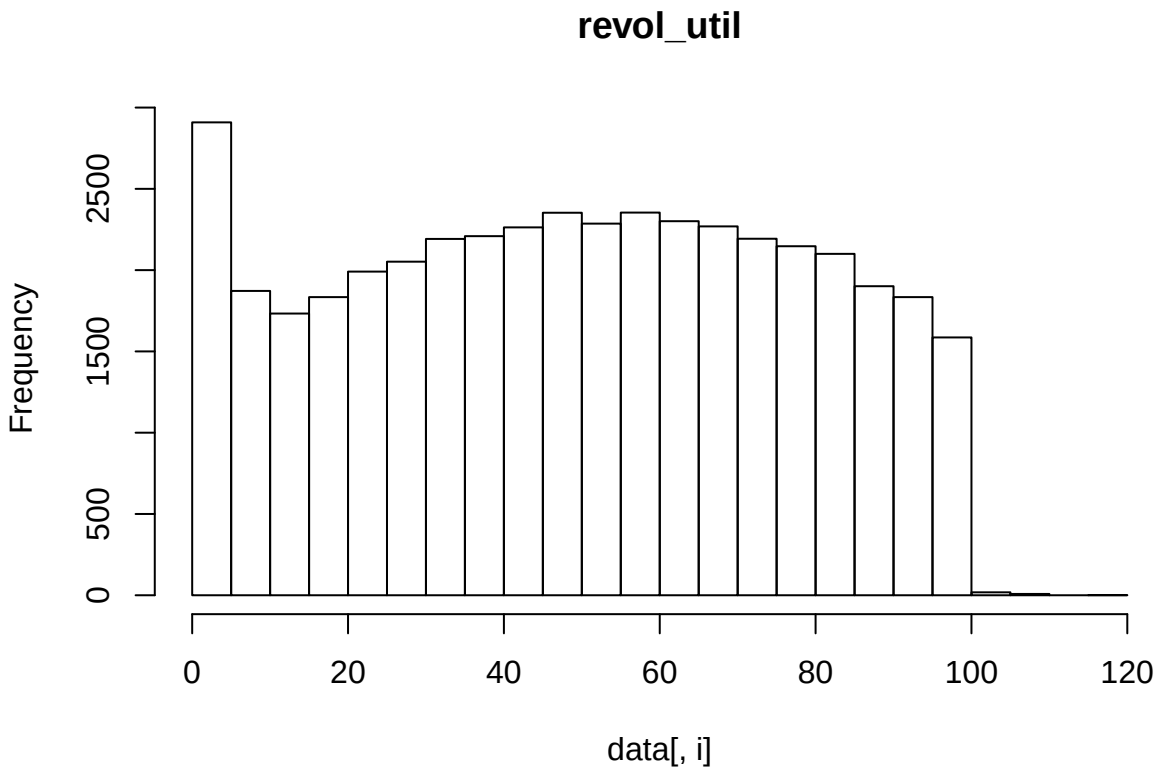

```
## [1] "hist(data[,21],main=\"open_acc\")"
```



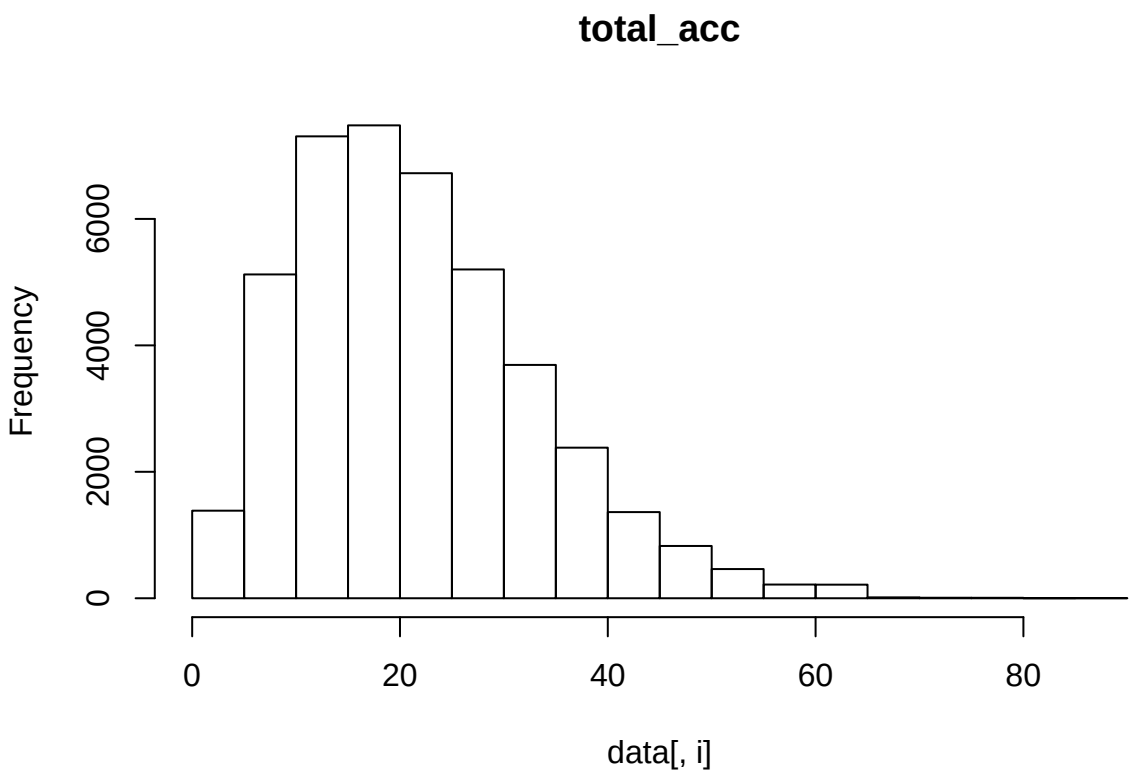

```
## [1] "hist(data[,22],main=\"pub_rec\")"
```



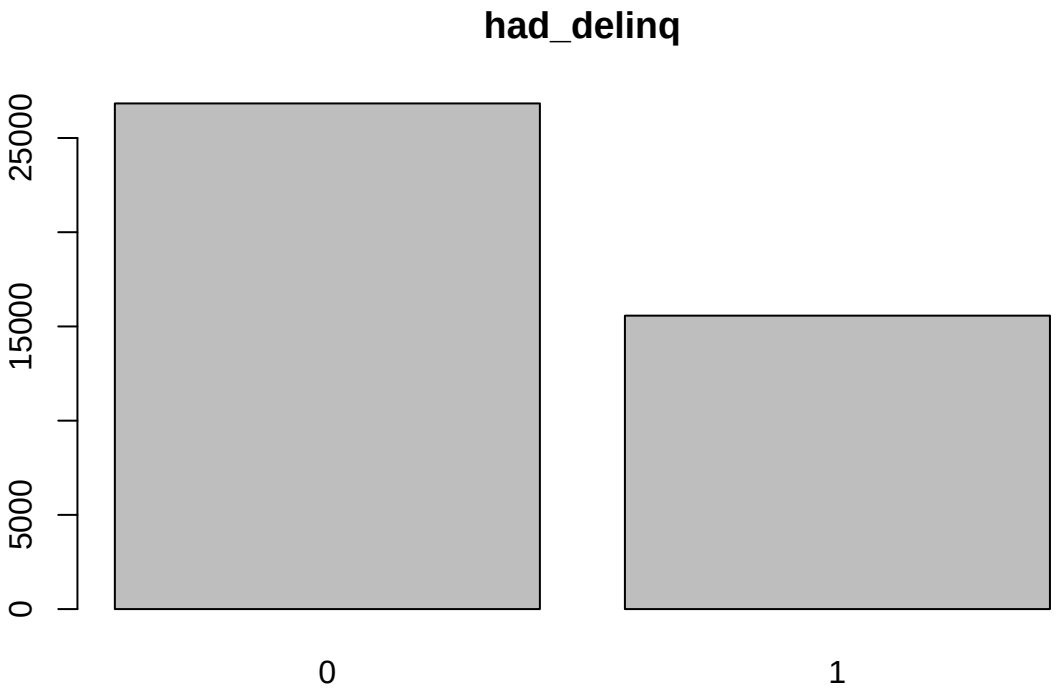

```
## [1] "hist(data[,23],main=\"revol_bal\")"
```



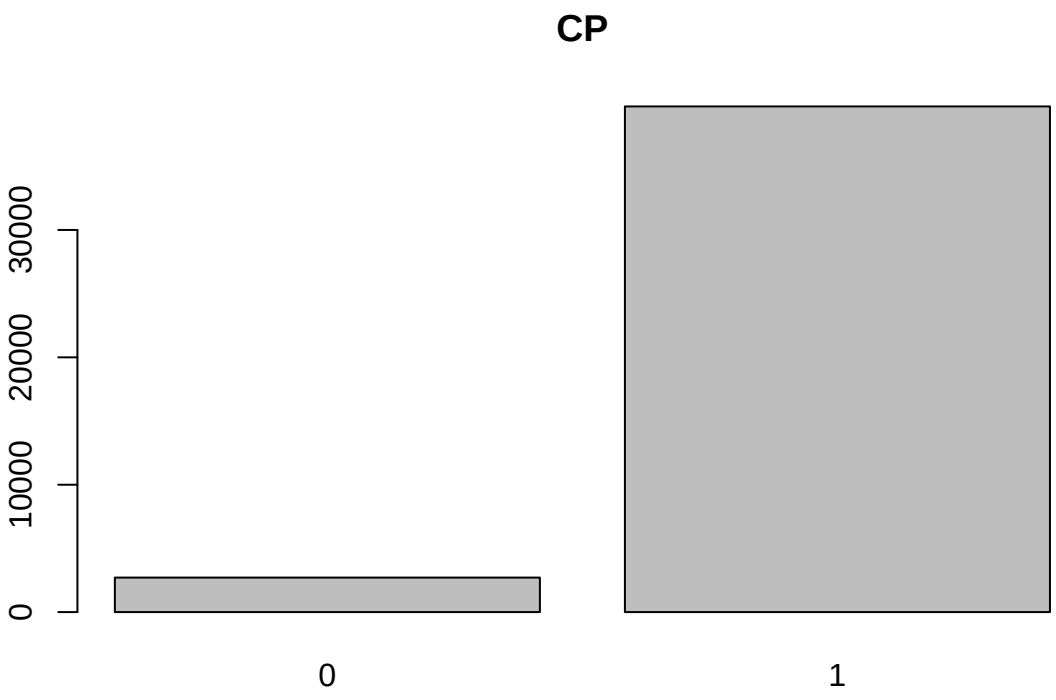

```
## [1] "hist(data[,24],main=\"revol_util\")"
```



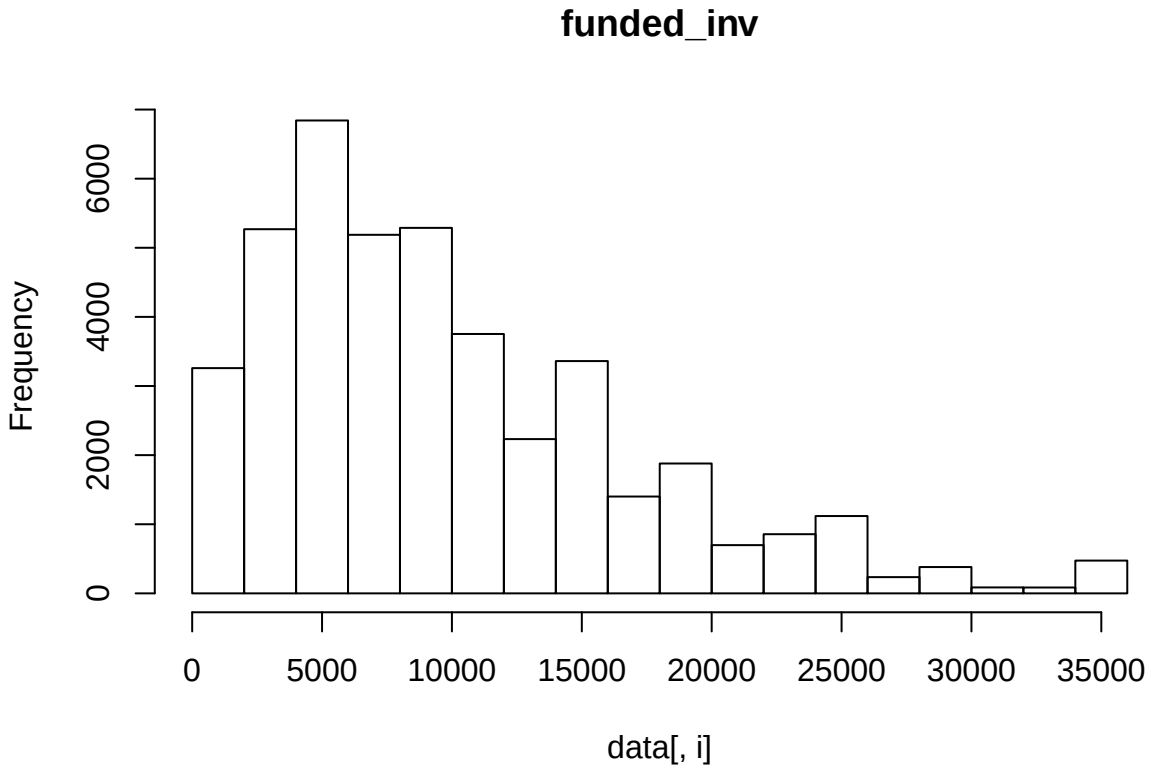

```
## [1] "hist(data[,25],main=\"total_acc\")"
```



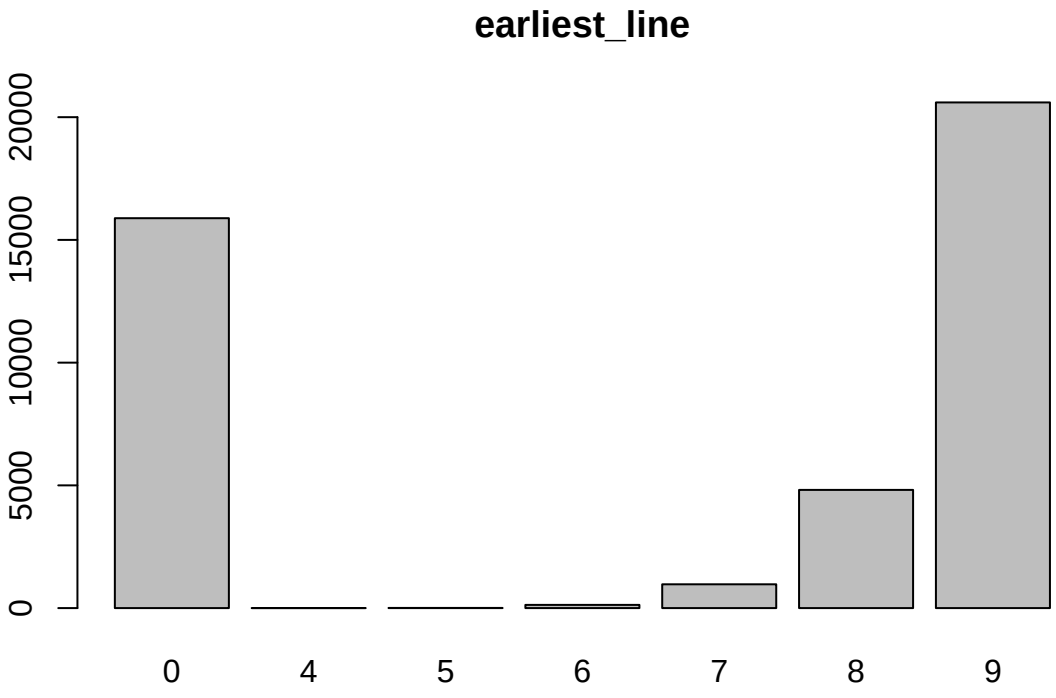

```
## [1] "freq(data[,26],main=\"had_delinq\")"
```



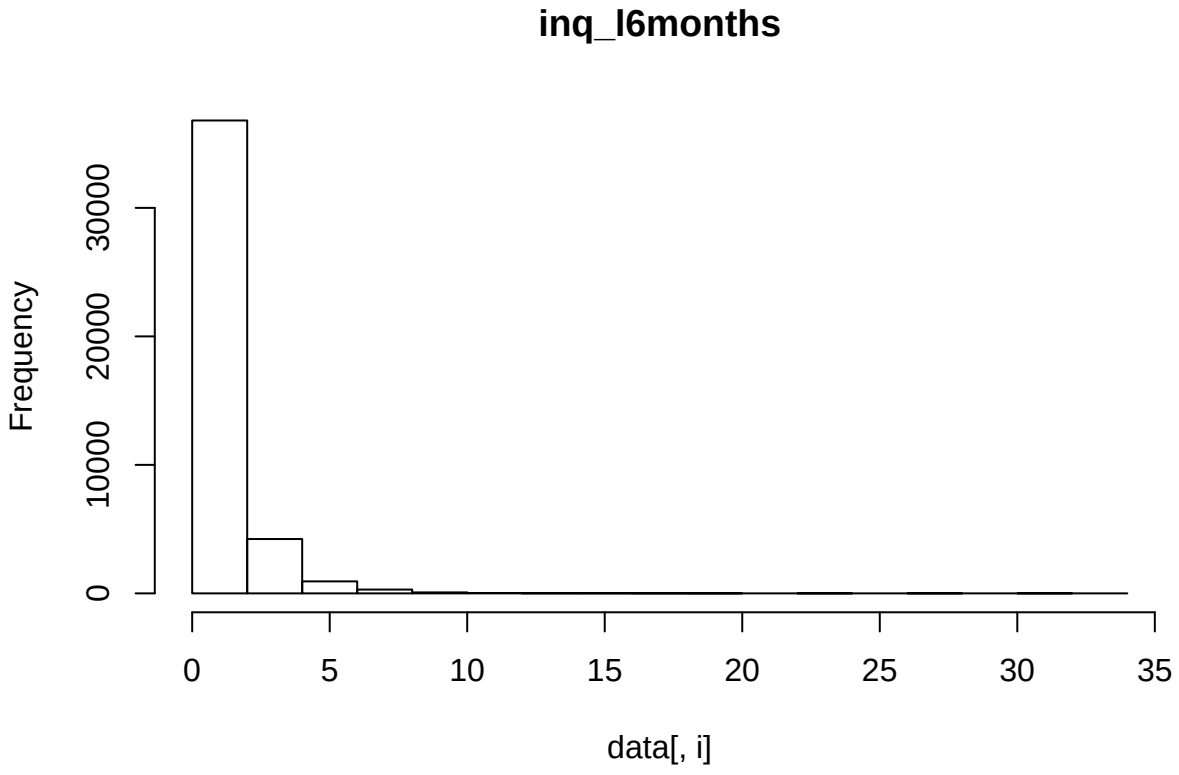

```
## [1] "freq(data[,27],main=\"CP\")"
```



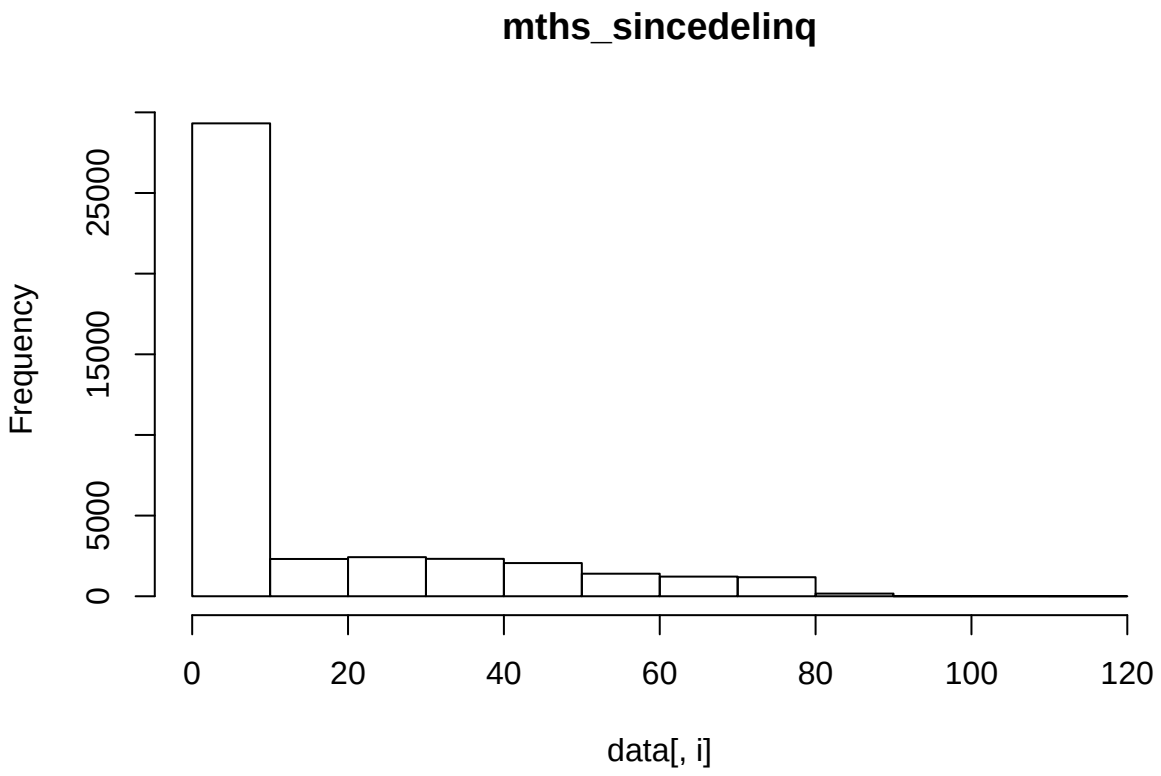

```
## [1] "hist(data[,28],main=\"funded_inv\")"
```




```
## [1] "freq(data[,29],main=\"earliest_line\")"
```




```
## [1] "hist(data[,30],main=\"inq_l6months\")"
```




```
## [1] "hist(data[,31],main=\"mths_sincedelinq\")"
```

Some facts which can be collected from the data are as follows:

*Majority of loan amounts funded fell under the $12K mark.

- There were approximately 2x more 36month term loans compared to 60 month term loans
- The interest rate of loans spanned, but the peak percentages were around: 8%, 11-12% and 14%.
- The distribution of loans indicates that per due dilligence, majority of the loans were classified to be Grade A and B loan. Indicating borrower stability.
- California proved to be the most prominant state the borrower's resided in per their application.

Machine Learning

I created an 'id' sample from the total 'N' observations with 70% of that data, totalling 29,684 loans. From here, I set the training data to consitute this newly created sample and utilized the remaining 30% as the test set.

Modeling

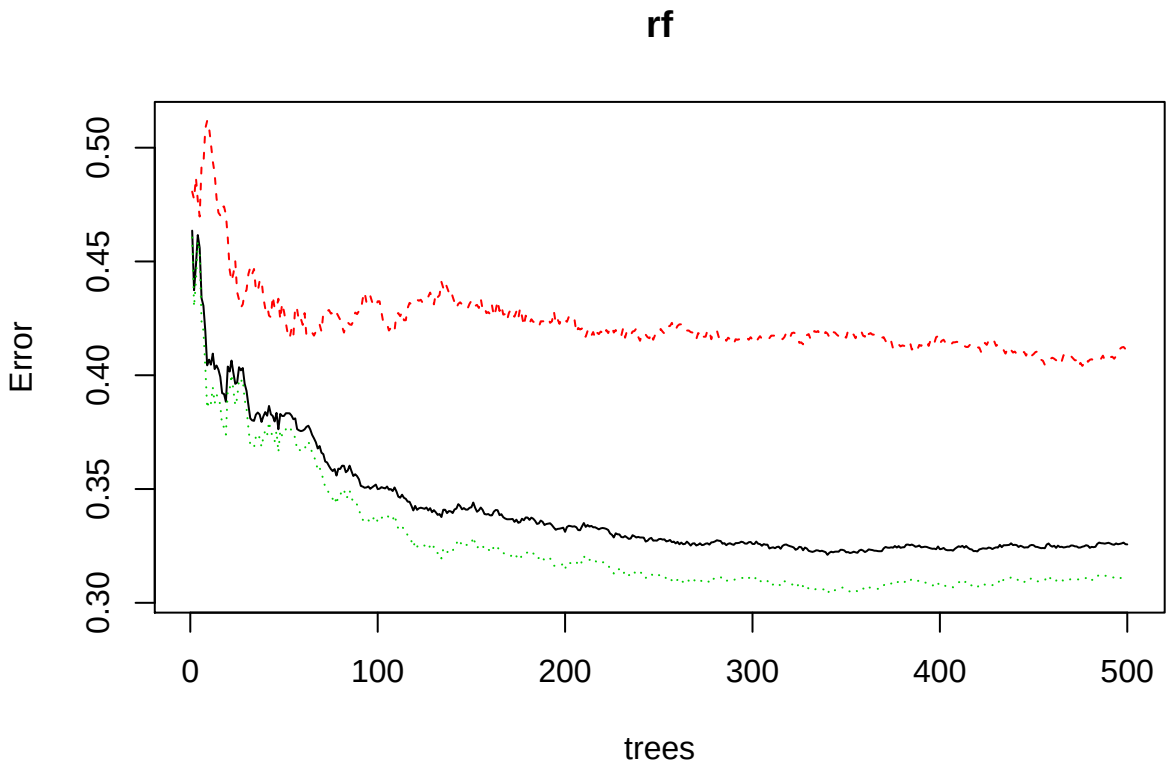
Random Forests

To start Random Forest, I regressed the Loan Status on all the previously cleaned features utilizing 500 trees. The number of variables tried at each split is 5. The graph represents a decreasing error rate with the increase of the number of trees for both Charged off and Fully paid loans. The confusion matrix produced shows that the Charged off classification error equals 0.3696 whereas the Fully paid classification error is only slightly less at 0.3531

```
rf <- randomForest(loan_status ~ ., data = d_train, ntree = 500)
rf
```

```
##
## Call:
##  randomForest(formula = loan_status ~ ., data = d_train, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 32.57%
## Confusion matrix:
##             Charged Off Fully Paid class.error
## Charged Off        2618       1819   0.4099617
## Fully Paid         7848      17399   0.3108488
```

```
plot(rf)
```

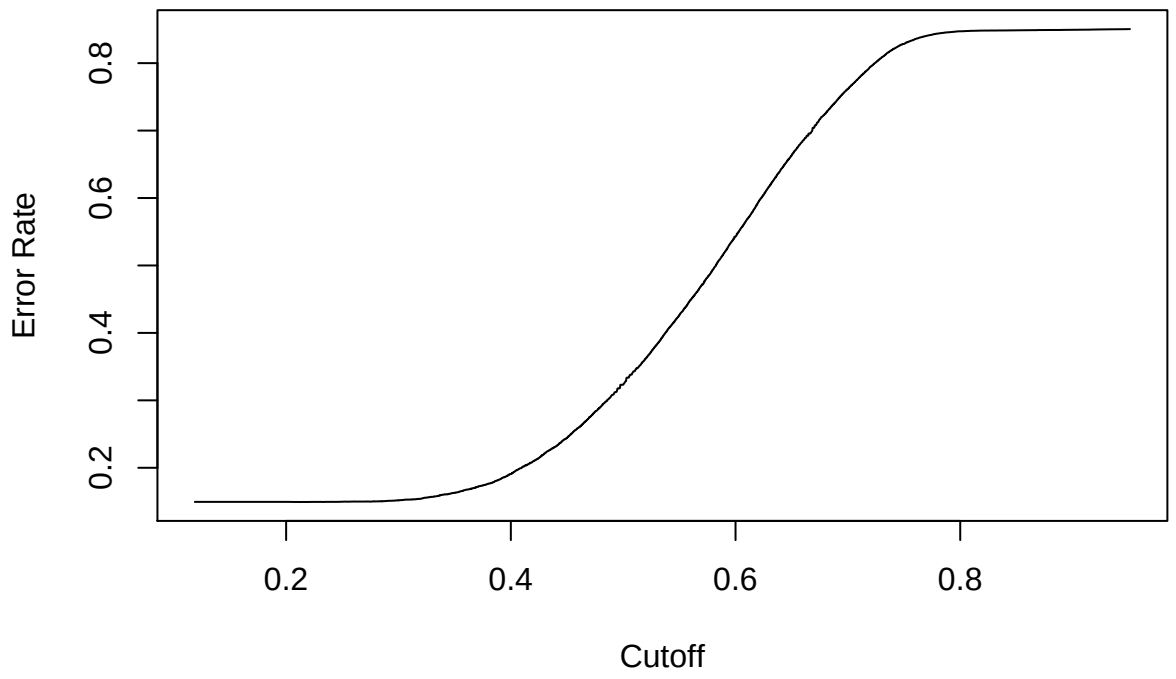


RF Model Evaluation

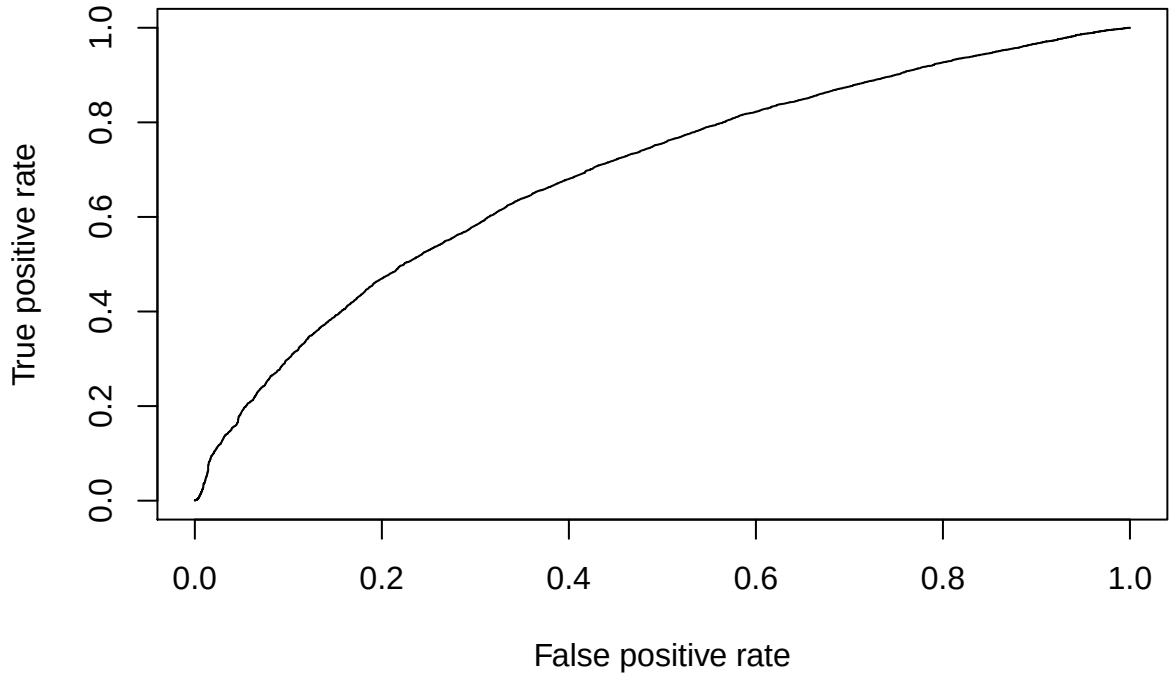
With the data trained, I go on to predict the probability of the test set observations and finalize the concluding outcome: Charged off or Fully Paid by indicating the class. I assigned 0.4 as the threshold of classification for the test set. The final performance result was very similar to that of the training set.

- Training Performance (AUC): 0.688
- Test Performance (AUC): 0.686

```r
phat_tr <- predict(rf, type = "prob")[,"Fully Paid"]
# phat_tr
rocr_obj_tr <- prediction(phat_tr, d_train$loan_status)
plot(performance(rocr_obj_tr, "err"))
```



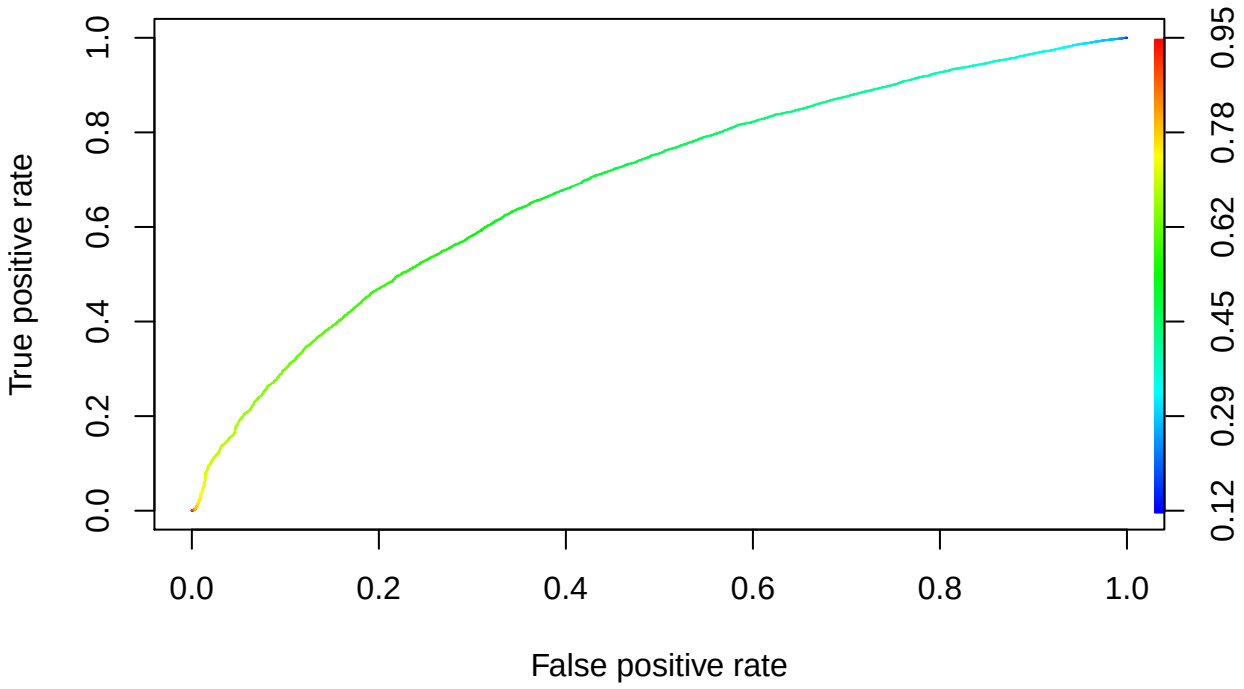

```r
plot(performance(rocr_obj_tr, "tpr", "fpr"))
```



```
#performance(rocr_obj_tr, "auc")
performance(rocr_obj_tr, "auc")@y.values[[1]]
```

```
## [1] 0.6920039
```

```
plot(performance(rocr_obj_tr, "tpr", "fpr"), colorize=TRUE)
```



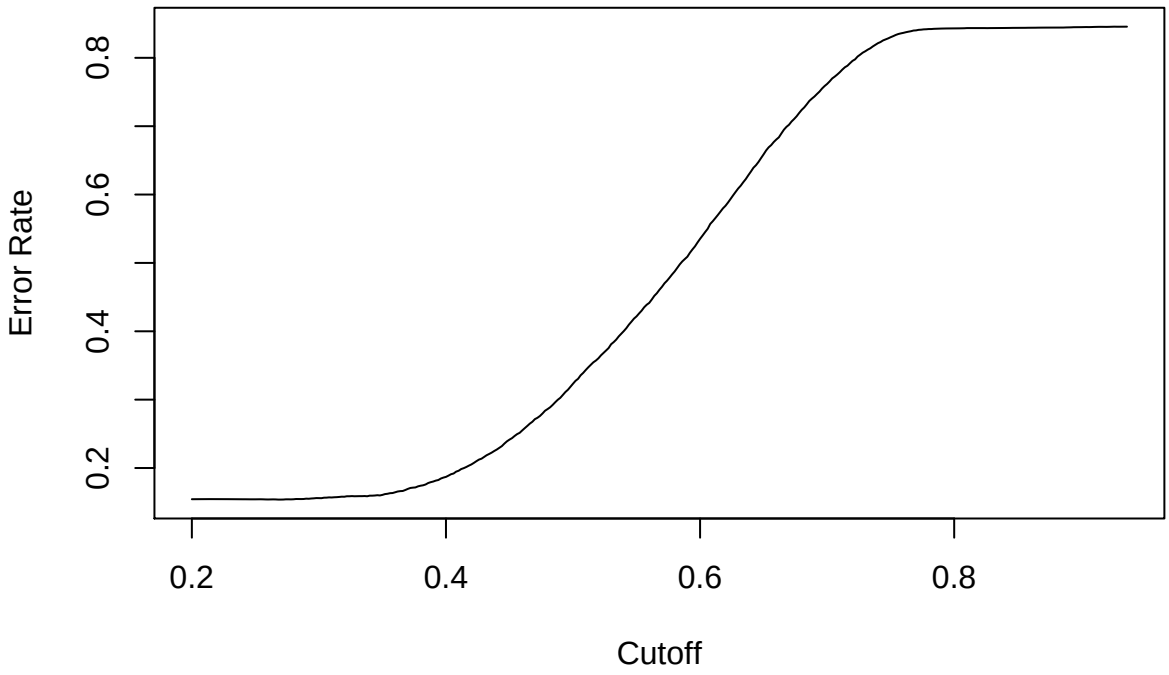

```
phat <- predict(rf, d_test, type = "prob")[,"Fully Paid"]
#phat
phatc <- predict(rf, d_test, type = "class")
#phatc

table(phatc,d_test$loan_status)
```

```
##
## phatc         Charged Off Fully Paid
##   Charged Off        1130       3302
##   Fully Paid          834       7456
```

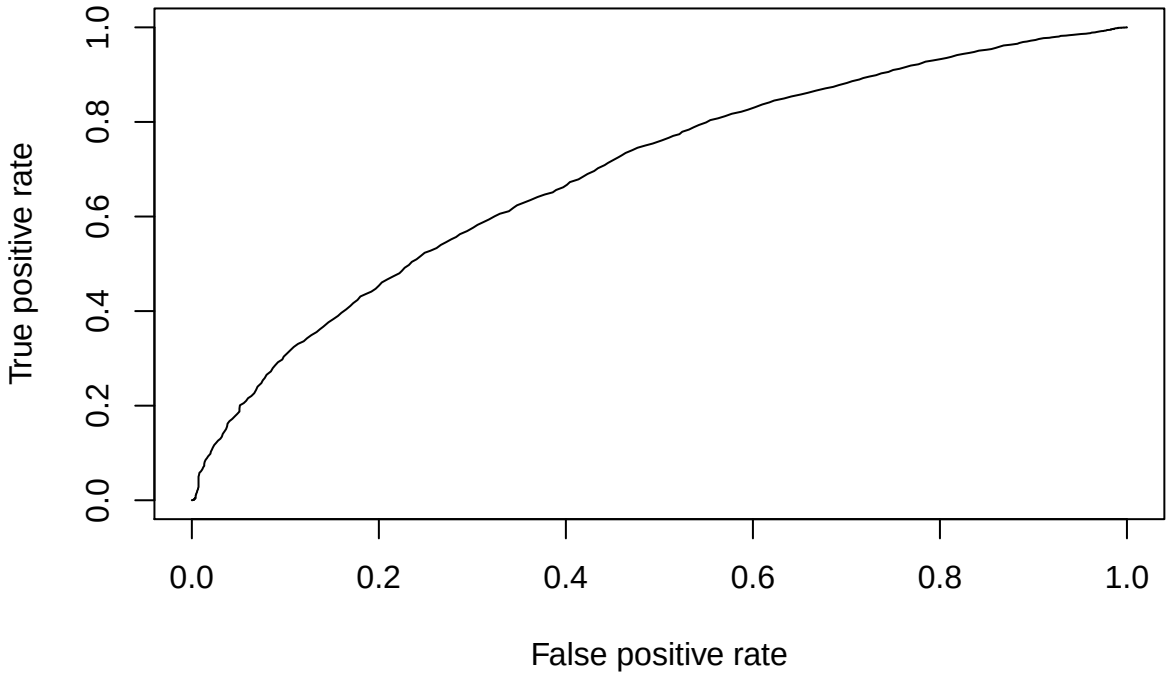
```
table(ifelse(phat>0.4,1,0), d_test$loan_status)
```

```
##
##     Charged Off Fully Paid
##   0         465        904
##   1        1499       9854
```

```
rocr_obj <- prediction(phat, d_test$loan_status)
plot(performance(rocr_obj, "err"))
```



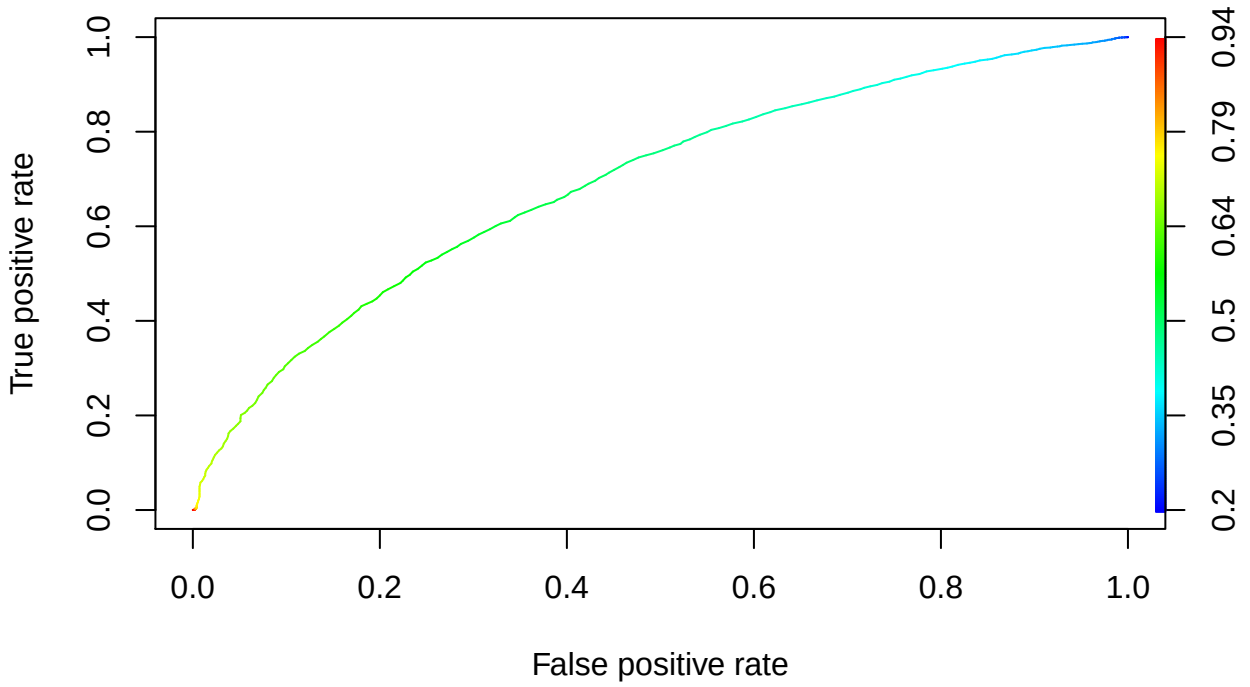

```
plot(performance(rocr_obj, "tpr", "fpr"))
```




```
#performance(rocr_obj, "auc")
performance(rocr_obj, "auc")@y.values[[1]]
```

```
## [1] 0.6922457
```

```
plot(performance(rocr_obj, "tpr", "fpr"), colorize=TRUE)
```



GBM Model Evaluation

From this model, only 9.7% of people Charged off their loan from those who were predicted to fully pay their loan and 73.4% paid off their loan from those who were predicted to Charge off.

```
d_train_ynum<-d_train
d_train_ynum$loan_status <- ifelse(d_train_ynum$loan_status=="Fully Paid",1,0)
d_test_ynum<-d_test
d_test_ynum$loan_status <- ifelse(d_test_ynum$loan_status=="Fully Paid",1,0)

mdgbm <- gbm(loan_status ~ ., data = d_train_ynum, distribution = "bernoulli",
             n.trees = 500, interaction.depth = 10, shrinkage = 0.03)
mdgbm
```

```
## gbm(formula = loan_status ~ ., distribution = "bernoulli", data = d_train_ynum,
##     n.trees = 500, interaction.depth = 10, shrinkage = 0.03)
## A gradient boosted model with bernoulli loss function.
## 500 iterations were performed.
## There were 30 predictors of which 26 had non-zero influence.
```

```
yhat <- predict(mdgbm, d_test, n.trees = 500)

CrossTable(ifelse(yhat>1.66,1,0), d_test$loan_status,prop.chisq=F)
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
```

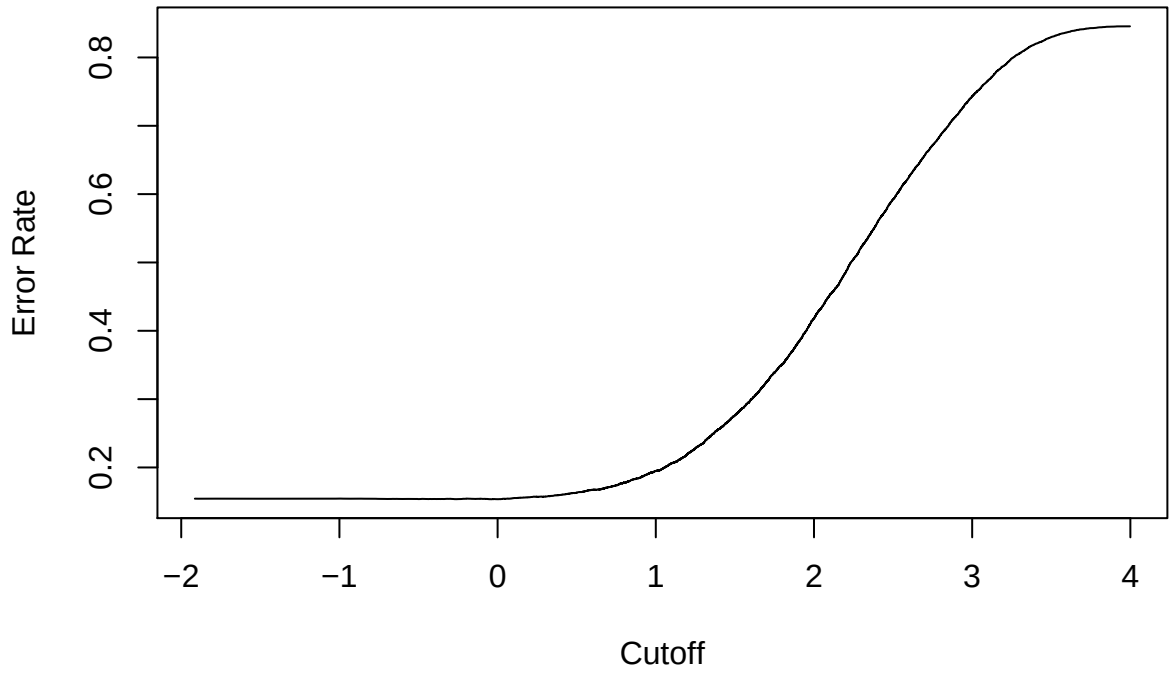
```
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
## ================================================================
##                               d_test$loan_status
## ifelse(yhat > 1.66, 1, 0)    Charged Off   Fully Paid   Total
## ----------------------------------------------------------------
## 0                                  1141         3166    4307
##                                   0.265        0.735   0.339
##                                   0.581        0.294
##                                   0.090        0.249
## ----------------------------------------------------------------
## 1                                   823         7592    8415
##                                   0.098        0.902   0.661
##                                   0.419        0.706
##                                   0.065        0.597
## ----------------------------------------------------------------
## Total                              1964        10758   12722
##                                   0.154        0.846
## ================================================================
```

GBM Test

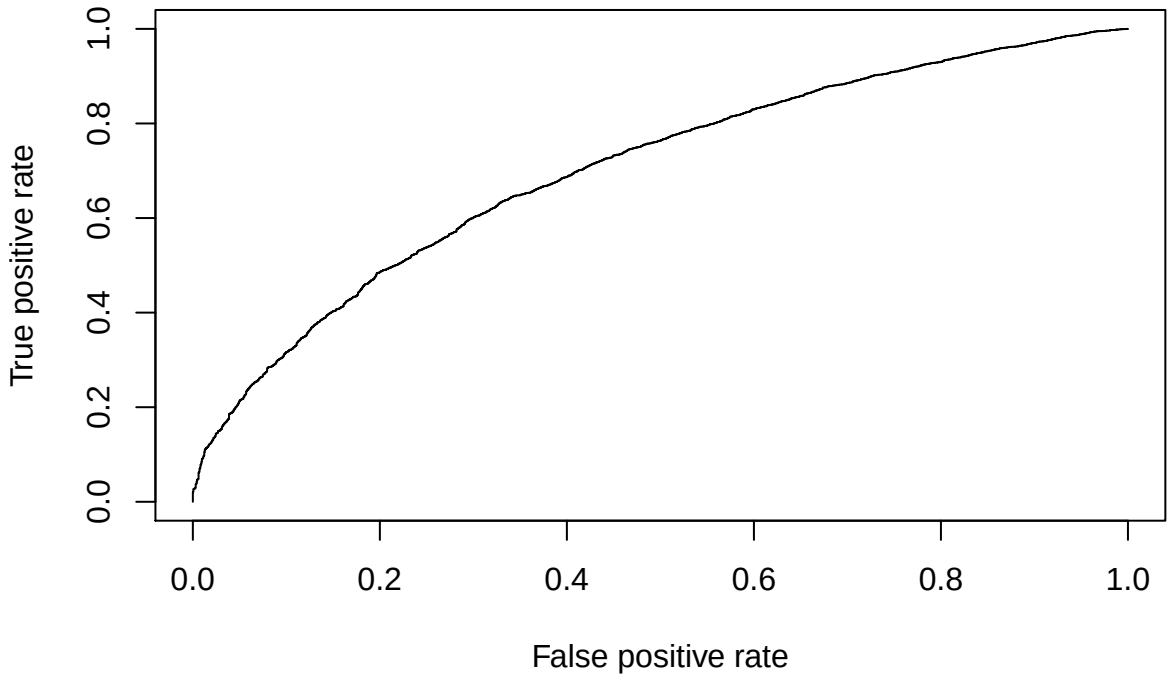
Here, I assigned 1.66 as the threshold of classification for the test set. The final performance result proved to be slightly better than that evaluated from the Random Forests.

- Test Performance (AUC): 0.701

```
rocr_obj <- prediction(yhat, d_test$loan_status)
plot(performance(rocr_obj, "err"))
```



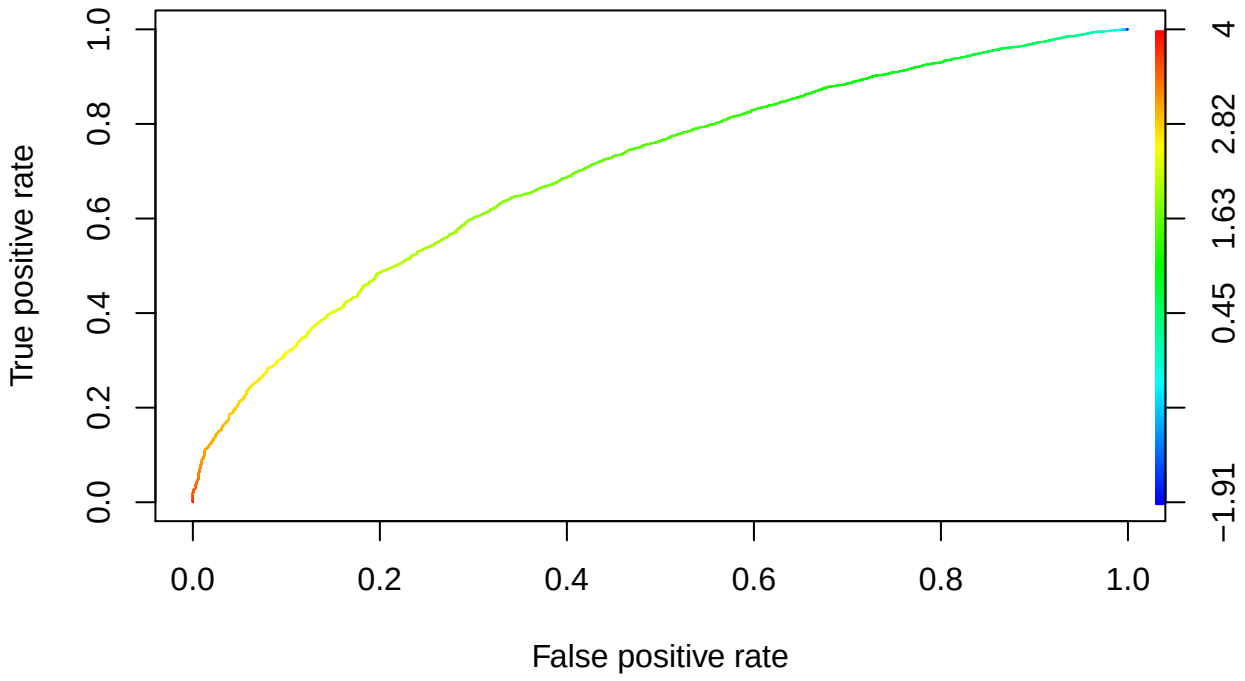
```
plot(performance(rocr_obj, "tpr", "fpr"))
```




```
#performance(rocr_obj, "auc")
performance(rocr_obj, "auc")@y.values[[1]]
```

```
## [1] 0.7020147
```

```
plot(performance(rocr_obj, "tpr", "fpr"), colorize=TRUE)
```



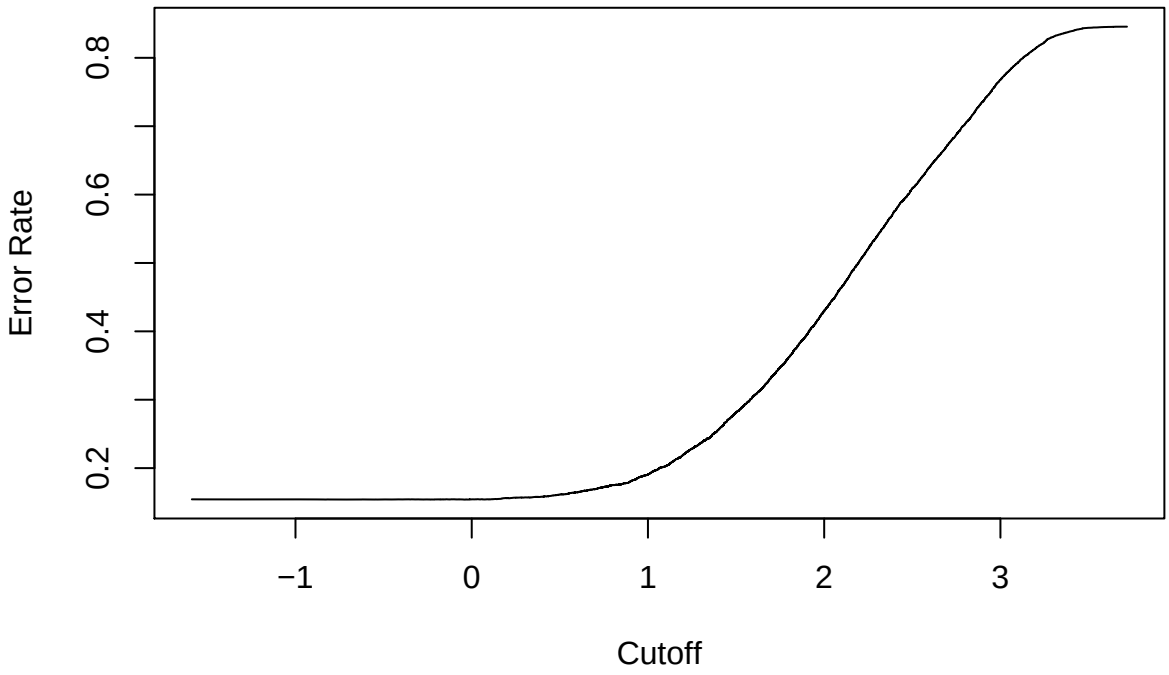
GBM with Cross Validation

Not too much of a difference could be seen from doing cross Validation. Yet again, the crosstable produced exact results reflecting only 9.7% of people Charged off their loan from those who were predicted to fully pay their loan. However, the prediction for those that paid off their loan from those who were predicted to Charge of their loan slightly increase by 0.4% (73.8%).

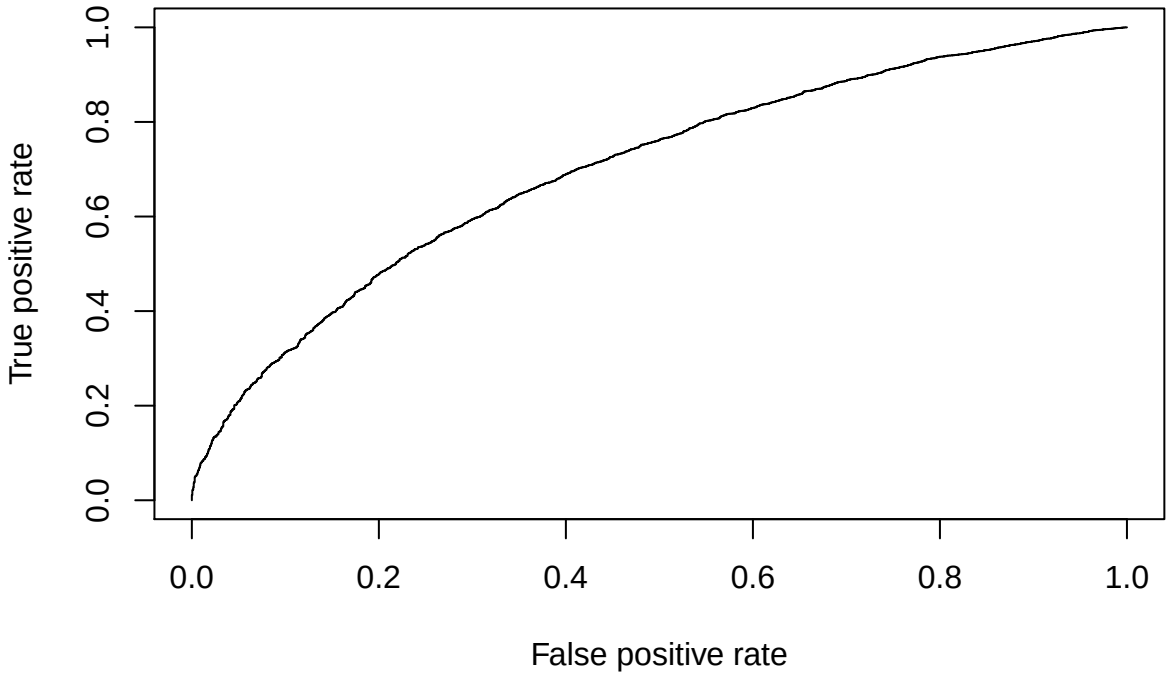
```
CrossTable(ifelse(yhatcv>1.66,1,0), d_test$loan_status,prop.chisq=F)
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
## ================================================================
##                                d_test$loan_status
## ifelse(yhatcv > 1.66, 1, 0)    Charged Off   Fully Paid   Total
## ----------------------------------------------------------------
## 0                                    1164         3275    4439
##                                     0.262        0.738   0.349
##                                     0.593        0.304
##                                     0.091        0.257
## ----------------------------------------------------------------
## 1                                     800         7483    8283
##                                     0.097        0.903   0.651
##                                     0.407        0.696
##                                     0.063        0.588
## ----------------------------------------------------------------
## Total                                1964        10758   12722
##                                     0.154        0.846
## ================================================================
```

```
rocr_obj <- prediction(yhatcv, d_test$loan_status)
plot(performance(rocr_obj, "err"))
```



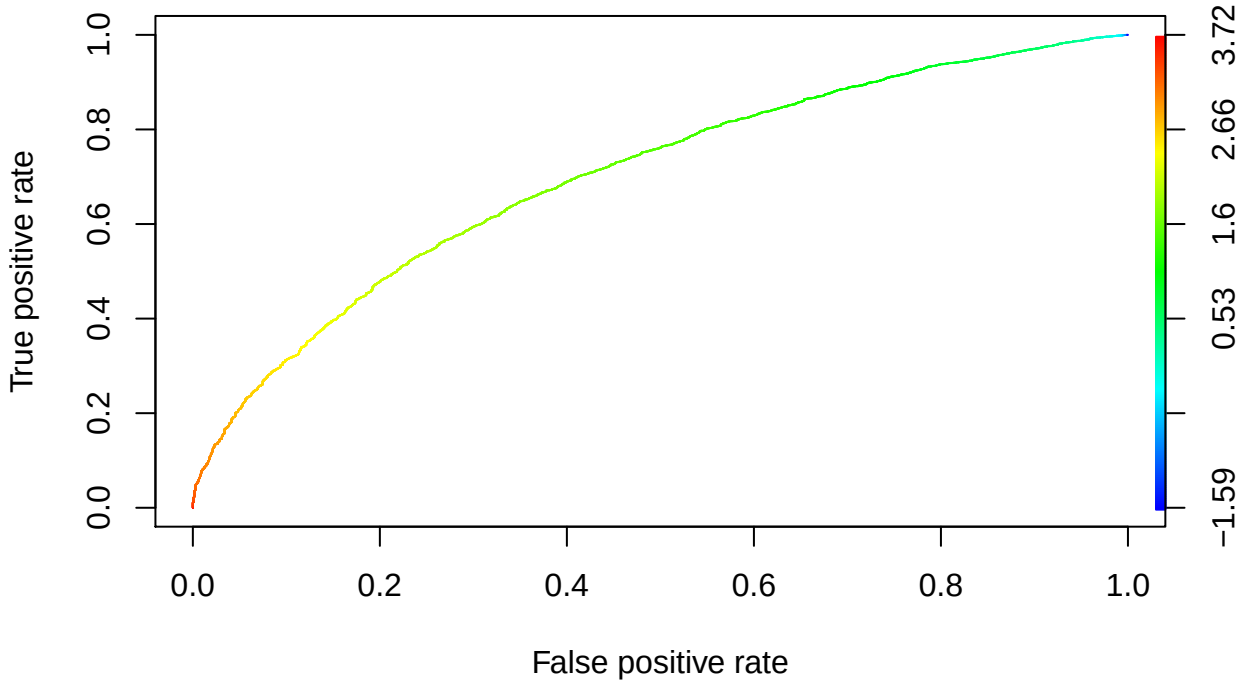

```
plot(performance(rocr_obj, "tpr", "fpr"))
```




```
#performance(rocr_obj, "auc")
performance(rocr_obj, "auc")@y.values[[1]]
```

```
## [1] 0.7008094
```

```
plot(performance(rocr_obj, "tpr", "fpr"), colorize=TRUE)
```



Once again, the model proved to increase ever so slighlty with the final performance result (AUC) equalling approximately 0.704.

Neural Network Model Evaluation

With neural networks I made sure to include a validation set in addition to the train and test set. After testing out a few different amounts, the concluding result I felt comfortable was with 2 hidden layers each containing 50 nodes with 50 iterations done. The model fell slightly short of the other models, reflecting:

- Test Performance (AUC): 0.677

```
N1<-nrow(data)
vt<-sample(1:N1,0.6*N)
d_train<-data[vt,]
d_vt<-data[-vt,]
N2<-nrow(d_vt)
t<-sample(1:N2,0.5*N2)
d_valid<-d_vt[t,]
d_test<-d_vt[-t,]
```

```
h2o.auc(h2odl)
```

```
## [1] 0.7022892
```

With a 0.9 threshold, the confusion matrix results indicated that out 807 borrowers which were predicted to charge off their loans indeed did, whereas 2572 who were predicted to charge off their loans, actually paid them off. On the other end, 4,613 borrowers who were predicted to pay off their loans did and only 490 ended up not paying off their loans from those predicted to be a good borrower.

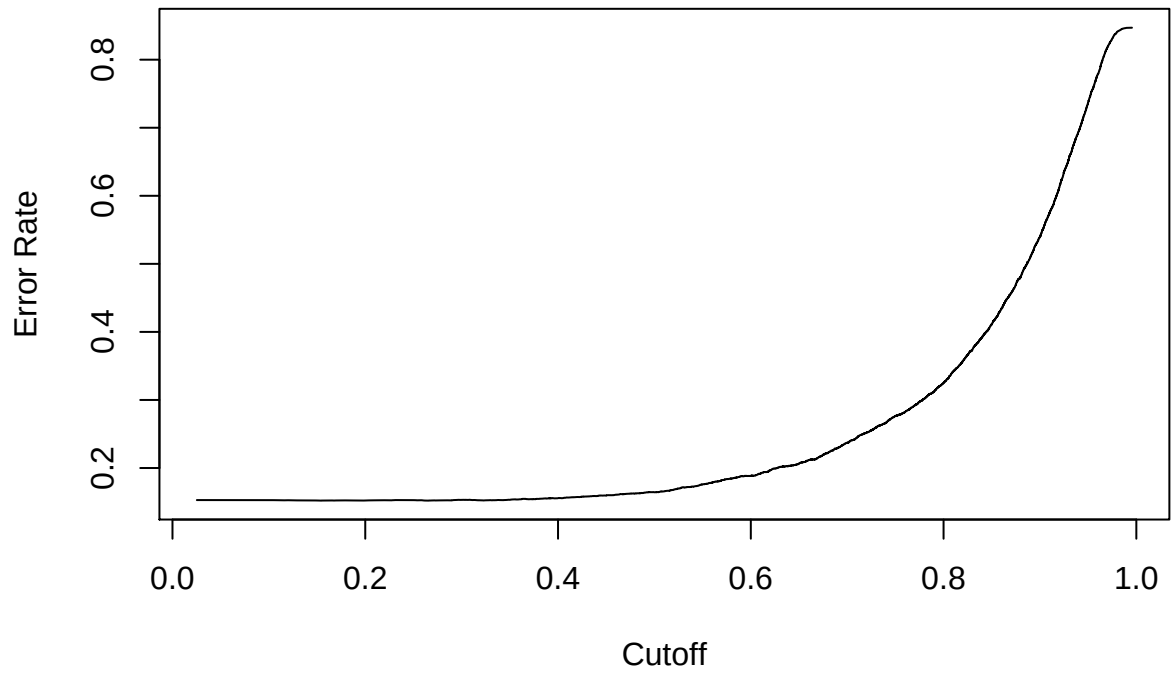
```
h2opred<-(h2o.predict(h2odl,newdata=dx_test))
```

```
##
  |
  |                                                                      |   0%
  |
  |======================================================================| 100%
```

```
h2opred<-as.data.frame(h2opred)
# table(h2opred$predict,d_test$loan_status)
nhat<-h2opred$Fully.Paid
length(d_test$loan_status)
```

```
## [1] 8482
```

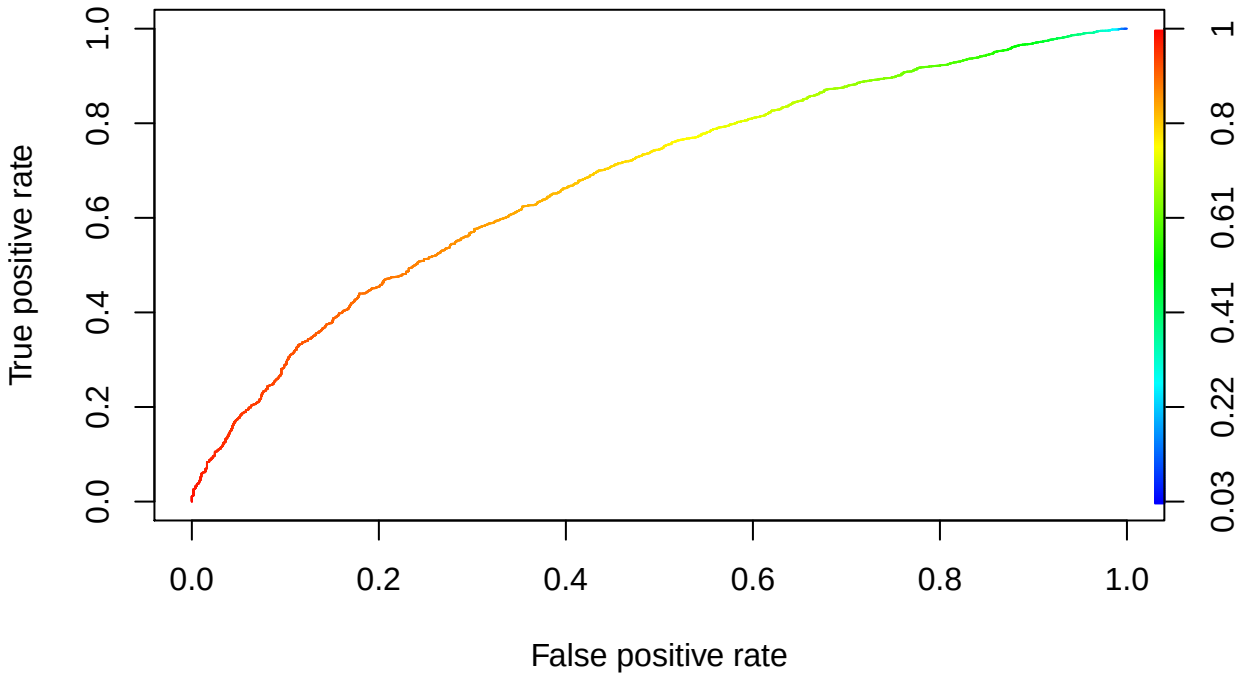
```
rocr_obj <- ROCR::prediction(nhat, d_test$loan_status)
plot(performance(rocr_obj, "err"))            # err vs cutoff
```




```
#performance(rocr_obj, "auc")
performance(rocr_obj, "auc")@y.values[[1]]
```

```
## [1] 0.6840629
```

```
plot(performance(rocr_obj, "tpr", "fpr"), colorize=TRUE)     # ROC curve
```




```
table(ifelse(nhat>0.9,1,0),d_test$loan_status)
```

```
##
##     Charged Off Fully Paid
##   0        1096       4377
##   1         201       2808
```

Discussion

The final chosen model from the ones tested above is GBM with Cross Validation leading to an AUROC of 0.704. Therefore, the charged off prediction per the given data and models has a precision of 70.4%. Although these results would be considered not bad, I would want to train and test more observations considering much of the data here was from when the company was starting and ramping up. Since this point LendingClub has grown exponentially with the number of annual borrowers to the iterative versions of their internal credit policy. When I started working on this project I believed that working with the first version of LC's figures would be insightful. However, with such manufactured results I would now be interested in testing the current algorithm on the newly approved borrowers along with their underlying attributes to see how it performs. If underwriters have vetted borrowers better we should see a less precise result.

It was interesting to play around with the Neural Networks model which proved to be rather volatile. As I altered the number of hidden layers, nodes and iterations the results returned ranged between 5 to 15 percentage points below my final presented precision rate of 67.7%. This proved to me that the probable cause was overfitting as had been thoroughly discussed in each class.