HW3

Eva

7/6/2018

##reference for glmnet  
##https://web.stanford.edu/~hastie/glmnet/glmnet\_alpha.html  
##https://www4.stat.ncsu.edu/~post/josh/LASSO\_Ridge\_Elastic\_Net\_-\_Examples.html  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(parsedate)  
library(lubridate)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(DescTools)  
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16

##   
## Attaching package: 'glmnet'

## The following object is masked from 'package:pROC':  
##   
## auc

set.seed(22)  
##question  
##Build the best logistic regression model to predict loan will be default (delay) or not. Add regularization to control for multicollinearity.  
rm(list=ls())  
setwd("~/evagit/BitTiger/Pro\_LendingClub")  
loan\_df= read.csv("loan.csv", stringsAsFactors = FALSE)  
#glimpse(loan\_df)  
#sum(duplicated(loan\_df))

##select features with less than 20% missing value  
num.NA = sort(sapply(loan\_df, function(x){sum(is.na(x))}), decreasing = TRUE)  
remain.col = names(num.NA[which(num.NA/dim(loan\_df)[1]<=0.2)])  
loan\_df = loan\_df[,remain.col]  
  
loan\_df <- within(loan\_df, rm("pymnt\_plan","policy\_code",'id', 'member\_id', 'url','desc', 'title','emp\_title','zip\_code'))  
  
loan\_df$issue\_year = sapply(loan\_df$issue\_d, function(x){substr(x, nchar(x)-3,nchar(x) )})  
#plot(table(loan\_df$issue\_year), main="Loan Issued Over the Year", ylab = "Amount", xlab = "Year")  
  
##create relative values  
loan\_df$last\_pay\_year = sapply(loan\_df$last\_pymnt\_d, function(x){substr(x, nchar(x)-3,nchar(x) )})  
table(loan\_df$last\_pay\_year)

##   
## 2007 2008 2009 2010 2011 2012 2013 2014 2015   
## 17659 2 335 838 2420 5748 11813 27194 66595 284625   
## 2016   
## 470150

loan\_df$payment\_length\_year=as.numeric(loan\_df$last\_pay\_year)-as.numeric(loan\_df$issue\_year)  
loan\_df$credit\_year = as.numeric(loan\_df$issue\_year) - sapply(loan\_df$earliest\_cr\_line, function(x){as.numeric(substr(x, nchar(x)-3,nchar(x)) )})  
table(loan\_df$credit\_year)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11   
## 11 67 100 4206 10611 13210 17531 24290 30409 35110 45736 55769   
## 12 13 14 15 16 17 18 19 20 21 22 23   
## 61028 63144 61670 56024 48987 42647 38431 36486 32354 28168 23015 18721   
## 24 25 26 27 28 29 30 31 32 33 34 35   
## 17912 17217 15301 13074 11713 9934 8865 7686 6241 4736 3948 3457   
## 36 37 38 39 40 41 42 43 44 45 46 47   
## 3224 3007 2502 1937 1538 1397 1160 969 690 658 558 461   
## 48 49 50 51 52 53 54 55 56 57 58 59   
## 368 270 242 141 113 76 58 43 42 18 17 14   
## 60 61 62 63 64 65 66 68 70 71   
## 11 3 3 6 5 6 1 1 1 1

##too many levels  
##we can group the states into region by economic level / geolocation / average interest rate  
##group the states by region  
#unique(loan\_df$addr\_state)  
west = c('CA', 'OR', 'UT','WA', 'CO', 'NV', 'AK', 'MT', 'HI', 'WY', 'ID')  
south\_west = c('AZ', 'TX', 'NM', 'OK')  
south\_east = c('GA', 'NC', 'VA', 'FL', 'KY', 'SC', 'LA', 'AL', 'WV', 'DC', 'AR', 'DE', 'MS', 'TN' )  
mid\_west = c('IL', 'MO', 'MN', 'OH', 'WI', 'KS', 'MI', 'SD', 'IA', 'NE', 'IN', 'ND')  
north\_east = c('CT', 'NY', 'PA', 'NJ', 'RI','MA', 'MD', 'VT', 'NH', 'ME')  
  
state\_to\_Reagion = function(s){  
 if (s %in% west){  
 return ("WEST")  
 }  
 if (s %in% south\_west){  
 return ("SOUTH\_WEST")  
 }  
 if (s %in% south\_east){  
 return ("SOUTH\_EAST")  
 }  
 if (s %in% mid\_west){  
 return ("MID\_WEST")  
 }  
 if (s %in% north\_east){  
 return ("NORTH\_EAST")  
 }  
 else{  
 return ("Missing\_Region")  
 }  
}  
  
loan\_df$Region = sapply(loan\_df$addr\_state, state\_to\_Reagion )  
##remove addr\_state  
loan\_df=within(loan\_df, rm('addr\_state'))  
  
## convert employment length into numeric value  
emp\_length\_c\_to\_n=function(x){  
 if(x=="< 1 year"){  
 return(0.5)  
 }  
 if(x=="1 year"){  
 return(1)  
 }  
 if(x=="2 years"){  
 return(2)  
 }  
 if(x=="3 years"){  
 return(3)  
 }  
 if(x=="4 years"){  
 return(4)  
 }  
 if(x=="5 years"){  
 return(5)  
 }  
 if(x=="6 years"){  
 return(6)  
 }  
 if(x=="7 years"){  
 return(7)  
 }  
 if(x=="8 years"){  
 return(8)  
 }  
 if(x=="9 years"){  
 return(9)  
 }  
 if(x=="10+ years"){  
 return(10)  
 }  
 else{  
 return(0)  
 }  
   
}  
  
loan\_df$Employment\_Length = sapply(loan\_df$emp\_length, emp\_length\_c\_to\_n)  
##delete some categorical variables  
loan\_df=loan\_df[,-which(colnames(loan\_df)%in%c("emp\_length", "last\_pymnt\_d", "last\_credit\_pull\_d", "issue\_d", "next\_pymnt\_d", "earliest\_cr\_line"))]  
glimpse(loan\_df)

## Observations: 887,379  
## Variables: 45  
## $ tot\_coll\_amt <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,...  
## $ tot\_cur\_bal <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,...  
## $ total\_rev\_hi\_lim <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,...  
## $ revol\_util <dbl> 83.70, 9.40, 98.50, 21.00, 53.90, 2...  
## $ collections\_12\_mths\_ex\_med <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## $ delinq\_2yrs <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## $ inq\_last\_6mths <dbl> 1, 5, 2, 1, 0, 3, 1, 2, 2, 0, 2, 0,...  
## $ open\_acc <dbl> 3, 3, 2, 10, 15, 9, 7, 4, 11, 2, 14...  
## $ pub\_rec <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## $ total\_acc <dbl> 9, 4, 10, 37, 38, 12, 11, 4, 13, 3,...  
## $ acc\_now\_delinq <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## $ annual\_inc <dbl> 24000.00, 30000.00, 12252.00, 49200...  
## $ loan\_amnt <dbl> 5000, 2500, 2400, 10000, 3000, 5000...  
## $ funded\_amnt <dbl> 5000, 2500, 2400, 10000, 3000, 5000...  
## $ funded\_amnt\_inv <dbl> 4975.00, 2500.00, 2400.00, 10000.00...  
## $ term <chr> " 36 months", " 60 months", " 36 mo...  
## $ int\_rate <dbl> 10.65, 15.27, 15.96, 13.49, 12.69, ...  
## $ installment <dbl> 162.87, 59.83, 84.33, 339.31, 67.79...  
## $ grade <chr> "B", "C", "C", "C", "B", "A", "C", ...  
## $ sub\_grade <chr> "B2", "C4", "C5", "C1", "B5", "A4",...  
## $ home\_ownership <chr> "RENT", "RENT", "RENT", "RENT", "RE...  
## $ verification\_status <chr> "Verified", "Source Verified", "Not...  
## $ loan\_status <chr> "Fully Paid", "Charged Off", "Fully...  
## $ purpose <chr> "credit\_card", "car", "small\_busine...  
## $ dti <dbl> 27.65, 1.00, 8.72, 20.00, 17.94, 11...  
## $ revol\_bal <dbl> 13648, 1687, 2956, 5598, 27783, 796...  
## $ initial\_list\_status <chr> "f", "f", "f", "f", "f", "f", "f", ...  
## $ out\_prncp <dbl> 0.00, 0.00, 0.00, 0.00, 766.90, 0.0...  
## $ out\_prncp\_inv <dbl> 0.00, 0.00, 0.00, 0.00, 766.90, 0.0...  
## $ total\_pymnt <dbl> 5861.071, 1008.710, 3003.654, 12226...  
## $ total\_pymnt\_inv <dbl> 5831.78, 1008.71, 3003.65, 12226.30...  
## $ total\_rec\_prncp <dbl> 5000.00, 456.46, 2400.00, 10000.00,...  
## $ total\_rec\_int <dbl> 861.07, 435.17, 603.65, 2209.33, 10...  
## $ total\_rec\_late\_fee <dbl> 0.00, 0.00, 0.00, 16.97, 0.00, 0.00...  
## $ recoveries <dbl> 0.00, 117.08, 0.00, 0.00, 0.00, 0.0...  
## $ collection\_recovery\_fee <dbl> 0.0000, 1.1100, 0.0000, 0.0000, 0.0...  
## $ last\_pymnt\_amnt <dbl> 171.62, 119.66, 649.91, 357.48, 67....  
## $ application\_type <chr> "INDIVIDUAL", "INDIVIDUAL", "INDIVI...  
## $ verification\_status\_joint <chr> "", "", "", "", "", "", "", "", "",...  
## $ issue\_year <chr> "2011", "2011", "2011", "2011", "20...  
## $ last\_pay\_year <chr> "2015", "2013", "2014", "2015", "20...  
## $ payment\_length\_year <dbl> 4, 2, 3, 4, 5, 4, 5, 4, 1, 1, 2, 2,...  
## $ credit\_year <dbl> 26, 12, 10, 15, 15, 7, 6, 4, 7, 7, ...  
## $ Region <chr> "SOUTH\_WEST", "SOUTH\_EAST", "MID\_WE...  
## $ Employment\_Length <dbl> 10.0, 0.5, 10.0, 10.0, 1.0, 3.0, 8....

sort(table(loan\_df$loan\_status))

##   
## Does not meet the credit policy. Status:Charged Off   
## 761   
## Default   
## 1219   
## Does not meet the credit policy. Status:Fully Paid   
## 1988   
## Late (16-30 days)   
## 2357   
## In Grace Period   
## 6253   
## Issued   
## 8460   
## Late (31-120 days)   
## 11591   
## Charged Off   
## 45248   
## Fully Paid   
## 207723   
## Current   
## 601779

loan\_df$loan\_status\_default\_binary = with(loan\_df, ifelse(loan\_status %in% c('Default','Late (16-30 days)',"Late (31-120 days)"), 1, 0))  
table(loan\_df$loan\_status\_default\_binary)

##   
## 0 1   
## 872212 15167

loan\_df$log\_annual\_inc = log(loan\_df$annual\_inc+1)  
loan\_df = within(loan\_df, rm("loan\_status", "annual\_inc" ))  
str(loan\_df)

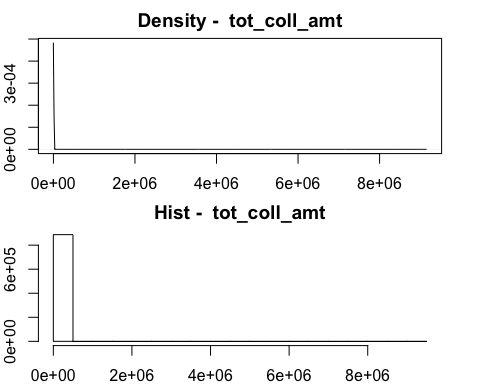
## 'data.frame': 887379 obs. of 45 variables:  
## $ tot\_coll\_amt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ tot\_cur\_bal : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_rev\_hi\_lim : num NA NA NA NA NA NA NA NA NA NA ...  
## $ revol\_util : num 83.7 9.4 98.5 21 53.9 28.3 85.6 87.5 32.6 36.5 ...  
## $ collections\_12\_mths\_ex\_med: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ delinq\_2yrs : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ inq\_last\_6mths : num 1 5 2 1 0 3 1 2 2 0 ...  
## $ open\_acc : num 3 3 2 10 15 9 7 4 11 2 ...  
## $ pub\_rec : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ total\_acc : num 9 4 10 37 38 12 11 4 13 3 ...  
## $ acc\_now\_delinq : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ loan\_amnt : num 5000 2500 2400 10000 3000 ...  
## $ funded\_amnt : num 5000 2500 2400 10000 3000 ...  
## $ funded\_amnt\_inv : num 4975 2500 2400 10000 3000 ...  
## $ term : chr " 36 months" " 60 months" " 36 months" " 36 months" ...  
## $ int\_rate : num 10.7 15.3 16 13.5 12.7 ...  
## $ installment : num 162.9 59.8 84.3 339.3 67.8 ...  
## $ grade : chr "B" "C" "C" "C" ...  
## $ sub\_grade : chr "B2" "C4" "C5" "C1" ...  
## $ home\_ownership : chr "RENT" "RENT" "RENT" "RENT" ...  
## $ verification\_status : chr "Verified" "Source Verified" "Not Verified" "Source Verified" ...  
## $ purpose : chr "credit\_card" "car" "small\_business" "other" ...  
## $ dti : num 27.65 1 8.72 20 17.94 ...  
## $ revol\_bal : num 13648 1687 2956 5598 27783 ...  
## $ initial\_list\_status : chr "f" "f" "f" "f" ...  
## $ out\_prncp : num 0 0 0 0 767 ...  
## $ out\_prncp\_inv : num 0 0 0 0 767 ...  
## $ total\_pymnt : num 5861 1009 3004 12226 3242 ...  
## $ total\_pymnt\_inv : num 5832 1009 3004 12226 3242 ...  
## $ total\_rec\_prncp : num 5000 456 2400 10000 2233 ...  
## $ total\_rec\_int : num 861 435 604 2209 1009 ...  
## $ total\_rec\_late\_fee : num 0 0 0 17 0 ...  
## $ recoveries : num 0 117 0 0 0 ...  
## $ collection\_recovery\_fee : num 0 1.11 0 0 0 0 0 0 2.09 2.52 ...  
## $ last\_pymnt\_amnt : num 171.6 119.7 649.9 357.5 67.8 ...  
## $ application\_type : chr "INDIVIDUAL" "INDIVIDUAL" "INDIVIDUAL" "INDIVIDUAL" ...  
## $ verification\_status\_joint : chr "" "" "" "" ...  
## $ issue\_year : chr "2011" "2011" "2011" "2011" ...  
## $ last\_pay\_year : chr "2015" "2013" "2014" "2015" ...  
## $ payment\_length\_year : num 4 2 3 4 5 4 5 4 1 1 ...  
## $ credit\_year : num 26 12 10 15 15 7 6 4 7 7 ...  
## $ Region : chr "SOUTH\_WEST" "SOUTH\_EAST" "MID\_WEST" "WEST" ...  
## $ Employment\_Length : num 10 0.5 10 10 1 3 8 9 4 0.5 ...  
## $ loan\_status\_default\_binary: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ log\_annual\_inc : num 10.09 10.31 9.41 10.8 11.29 ...

summary(loan\_df)

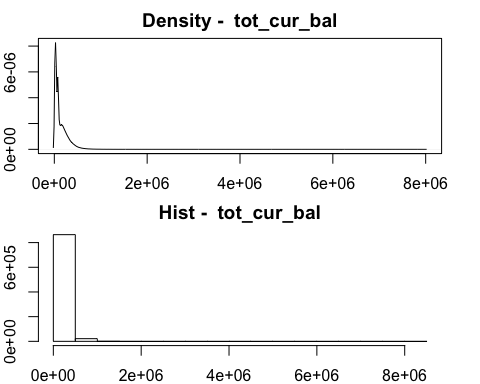
## tot\_coll\_amt tot\_cur\_bal total\_rev\_hi\_lim revol\_util   
## Min. : 0 Min. : 0 Min. : 0 Min. : 0.00   
## 1st Qu.: 0 1st Qu.: 29853 1st Qu.: 13900 1st Qu.: 37.70   
## Median : 0 Median : 80559 Median : 23700 Median : 56.00   
## Mean : 226 Mean : 139458 Mean : 32069 Mean : 55.07   
## 3rd Qu.: 0 3rd Qu.: 208205 3rd Qu.: 39800 3rd Qu.: 73.60   
## Max. :9152545 Max. :8000078 Max. :9999999 Max. :892.30   
## NA's :70276 NA's :70276 NA's :70276 NA's :502   
## collections\_12\_mths\_ex\_med delinq\_2yrs inq\_last\_6mths   
## Min. : 0.00000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 0.00000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median : 0.00000 Median : 0.0000 Median : 0.0000   
## Mean : 0.01438 Mean : 0.3144 Mean : 0.6946   
## 3rd Qu.: 0.00000 3rd Qu.: 0.0000 3rd Qu.: 1.0000   
## Max. :20.00000 Max. :39.0000 Max. :33.0000   
## NA's :145 NA's :29 NA's :29   
## open\_acc pub\_rec total\_acc acc\_now\_delinq   
## Min. : 0.00 Min. : 0.0000 Min. : 1.00 Min. : 0.000000   
## 1st Qu.: 8.00 1st Qu.: 0.0000 1st Qu.: 17.00 1st Qu.: 0.000000   
## Median :11.00 Median : 0.0000 Median : 24.00 Median : 0.000000   
## Mean :11.55 Mean : 0.1953 Mean : 25.27 Mean : 0.004991   
## 3rd Qu.:14.00 3rd Qu.: 0.0000 3rd Qu.: 32.00 3rd Qu.: 0.000000   
## Max. :90.00 Max. :86.0000 Max. :169.00 Max. :14.000000   
## NA's :29 NA's :29 NA's :29 NA's :29   
## loan\_amnt funded\_amnt funded\_amnt\_inv term   
## Min. : 500 Min. : 500 Min. : 0 Length:887379   
## 1st Qu.: 8000 1st Qu.: 8000 1st Qu.: 8000 Class :character   
## Median :13000 Median :13000 Median :13000 Mode :character   
## Mean :14755 Mean :14742 Mean :14702   
## 3rd Qu.:20000 3rd Qu.:20000 3rd Qu.:20000   
## Max. :35000 Max. :35000 Max. :35000   
##   
## int\_rate installment grade sub\_grade   
## Min. : 5.32 Min. : 15.67 Length:887379 Length:887379   
## 1st Qu.: 9.99 1st Qu.: 260.70 Class :character Class :character   
## Median :12.99 Median : 382.55 Mode :character Mode :character   
## Mean :13.25 Mean : 436.72   
## 3rd Qu.:16.20 3rd Qu.: 572.60   
## Max. :28.99 Max. :1445.46   
##   
## home\_ownership verification\_status purpose   
## Length:887379 Length:887379 Length:887379   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## dti revol\_bal initial\_list\_status out\_prncp   
## Min. : 0.00 Min. : 0 Length:887379 Min. : 0   
## 1st Qu.: 11.91 1st Qu.: 6443 Class :character 1st Qu.: 0   
## Median : 17.65 Median : 11875 Mode :character Median : 6458   
## Mean : 18.16 Mean : 16921 Mean : 8403   
## 3rd Qu.: 23.95 3rd Qu.: 20829 3rd Qu.:13659   
## Max. :9999.00 Max. :2904836 Max. :49373   
##   
## out\_prncp\_inv total\_pymnt total\_pymnt\_inv total\_rec\_prncp  
## Min. : 0 Min. : 0 Min. : 0 Min. : 0   
## 1st Qu.: 0 1st Qu.: 1915 1st Qu.: 1900 1st Qu.: 1201   
## Median : 6456 Median : 4895 Median : 4862 Median : 3215   
## Mean : 8400 Mean : 7559 Mean : 7521 Mean : 5758   
## 3rd Qu.:13654 3rd Qu.:10617 3rd Qu.:10566 3rd Qu.: 8000   
## Max. :49373 Max. :57778 Max. :57778 Max. :35000   
##   
## total\_rec\_int total\_rec\_late\_fee recoveries   
## Min. : 0.0 Min. : 0.0000 Min. : 0.00   
## 1st Qu.: 441.5 1st Qu.: 0.0000 1st Qu.: 0.00   
## Median : 1073.3 Median : 0.0000 Median : 0.00   
## Mean : 1754.8 Mean : 0.3967 Mean : 45.92   
## 3rd Qu.: 2238.3 3rd Qu.: 0.0000 3rd Qu.: 0.00   
## Max. :24205.6 Max. :358.6800 Max. :33520.27   
##   
## collection\_recovery\_fee last\_pymnt\_amnt application\_type   
## Min. : 0.000 Min. : 0.0 Length:887379   
## 1st Qu.: 0.000 1st Qu.: 280.2 Class :character   
## Median : 0.000 Median : 462.8 Mode :character   
## Mean : 4.881 Mean : 2164.2   
## 3rd Qu.: 0.000 3rd Qu.: 831.2   
## Max. :7002.190 Max. :36475.6   
##   
## verification\_status\_joint issue\_year last\_pay\_year   
## Length:887379 Length:887379 Length:887379   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## payment\_length\_year credit\_year Region Employment\_Length  
## Min. :0.000 Min. : 0.00 Length:887379 Min. : 0.000   
## 1st Qu.:1.000 1st Qu.:11.00 Class :character 1st Qu.: 2.000   
## Median :1.000 Median :15.00 Mode :character Median : 6.000   
## Mean :1.309 Mean :16.32 Mean : 5.745   
## 3rd Qu.:2.000 3rd Qu.:20.00 3rd Qu.:10.000   
## Max. :6.000 Max. :71.00 Max. :10.000   
## NA's :17659 NA's :29   
## loan\_status\_default\_binary log\_annual\_inc   
## Min. :0.00000 Min. : 0.00   
## 1st Qu.:0.00000 1st Qu.:10.71   
## Median :0.00000 Median :11.08   
## Mean :0.01709 Mean :11.08   
## 3rd Qu.:0.00000 3rd Qu.:11.41   
## Max. :1.00000 Max. :16.07   
## NA's :4

##deal with missing value  
loan\_df[which(is.na(loan\_df[,"credit\_year"])),]=0  
##replace all numeric missing value with medium  
data.type=sapply(loan\_df, class)  
plot.new()  
for ( i in colnames(loan\_df)){  
 if (class(loan\_df[,i])=='numeric'){  
 print (names(loan\_df[i]))  
 savename=names(loan\_df[i])  
 loan\_df[is.na(loan\_df[,i]), i]=median(loan\_df[,i], na.rm=TRUE)  
   
 d= density(loan\_df[,i])  
 par(mar=c(2,2,2,2), mfrow=c(2,1))  
 plot(d , main=paste("Density - ",savename))  
 hist(loan\_df[,i], main=paste("Hist - ",savename))  
  
 rm(d)  
 }  
}

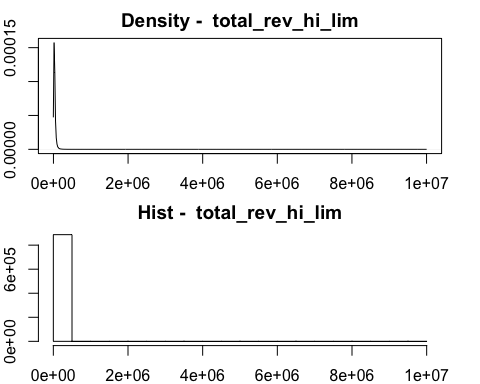
## [1] "tot\_coll\_amt"



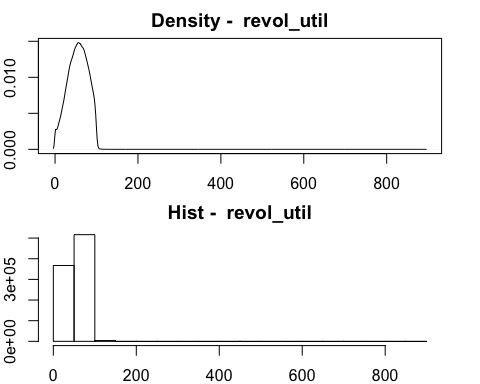
## [1] "tot\_cur\_bal"



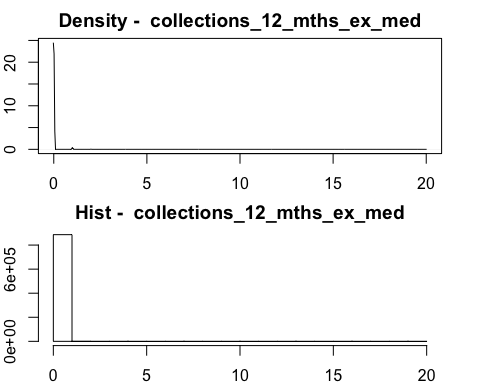
## [1] "total\_rev\_hi\_lim"



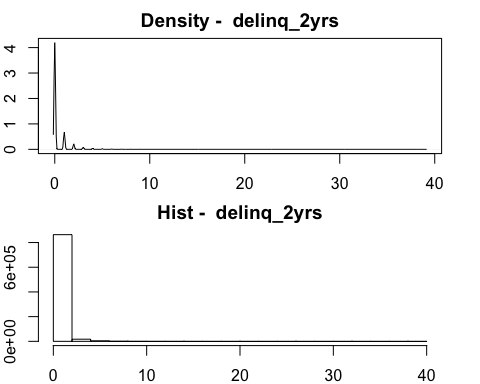
## [1] "revol\_util"



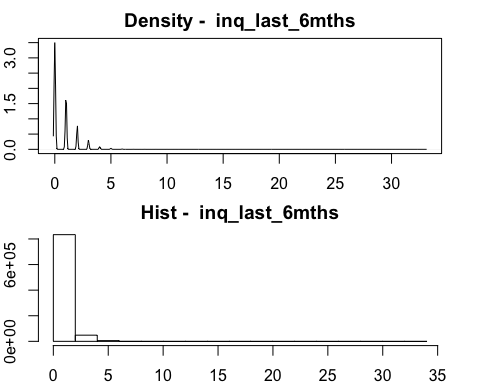
## [1] "collections\_12\_mths\_ex\_med"



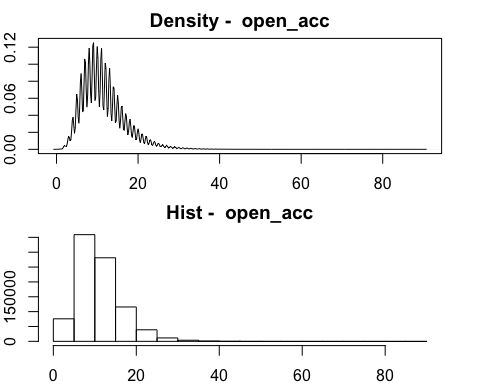
## [1] "delinq\_2yrs"



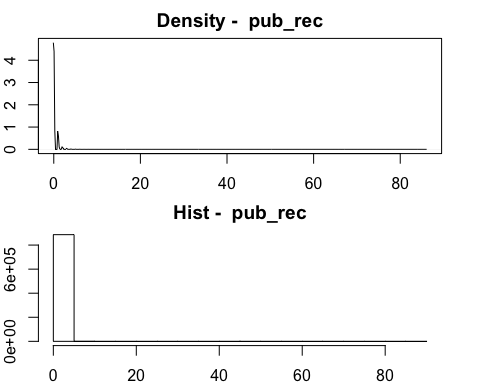
## [1] "inq\_last\_6mths"



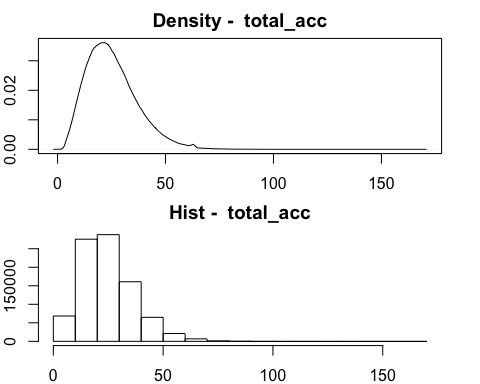
## [1] "open\_acc"



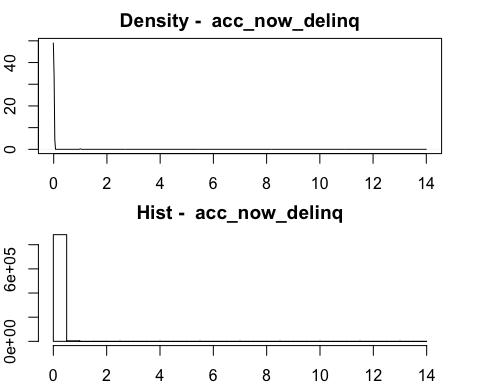
## [1] "pub\_rec"



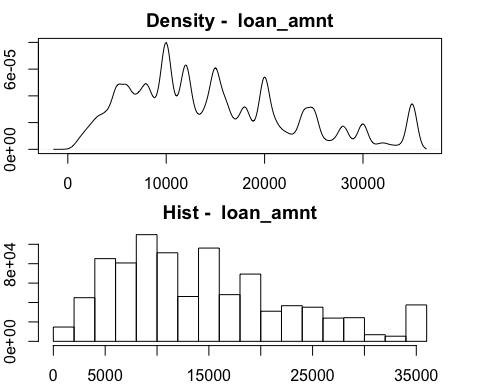
## [1] "total\_acc"



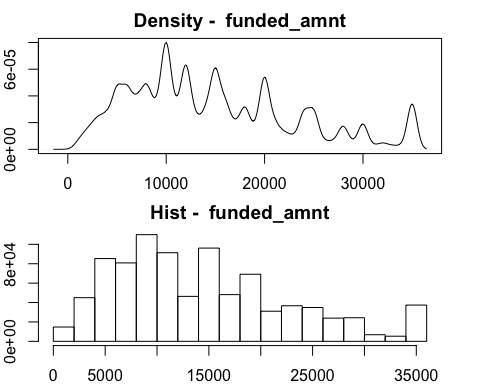
## [1] "acc\_now\_delinq"



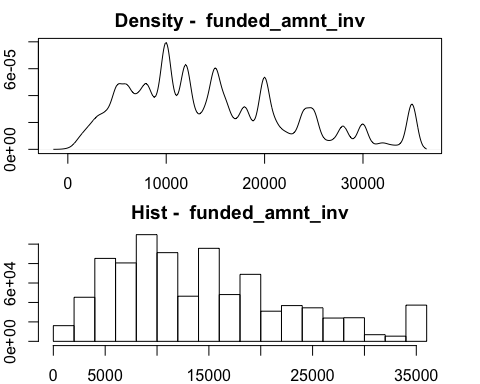
## [1] "loan\_amnt"



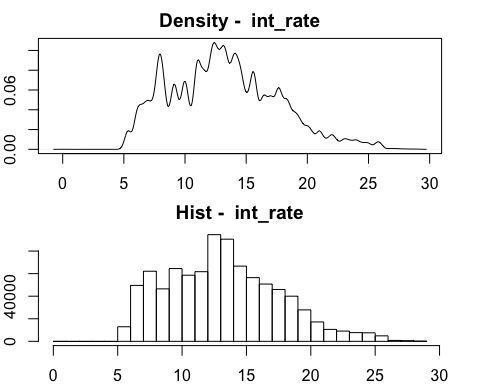
## [1] "funded\_amnt"



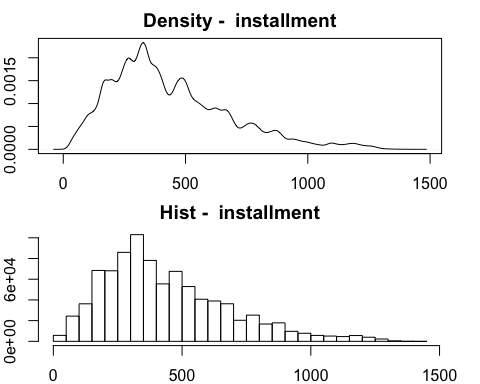
## [1] "funded\_amnt\_inv"



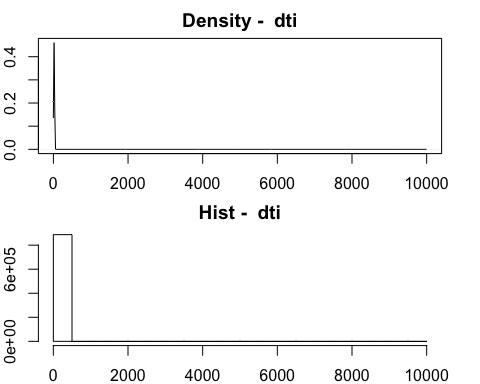
## [1] "int\_rate"



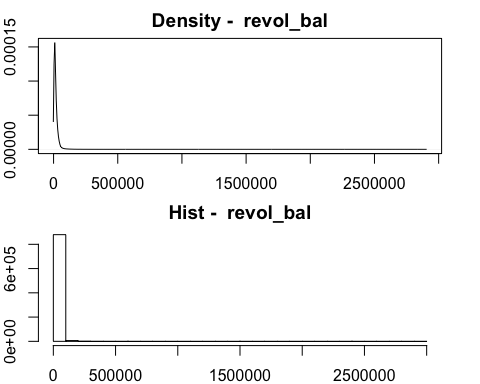
## [1] "installment"



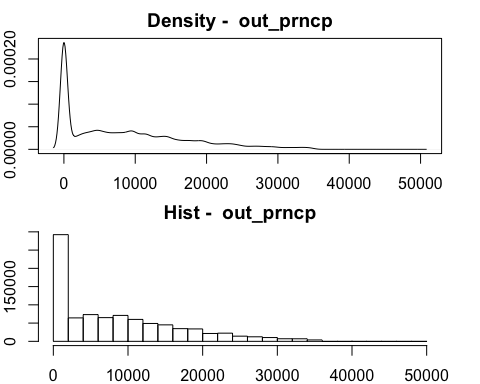
## [1] "dti"



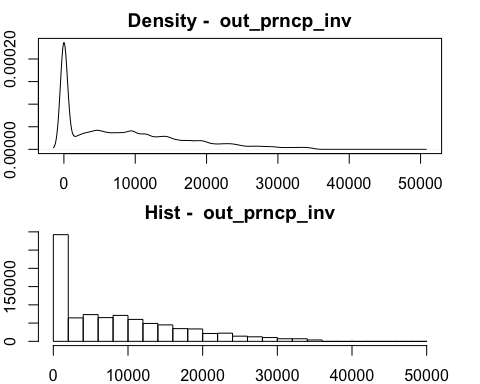
## [1] "revol\_bal"



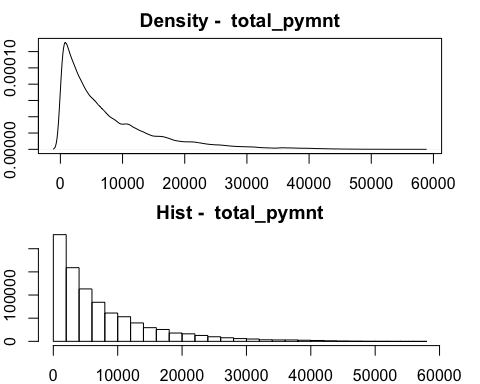
## [1] "out\_prncp"



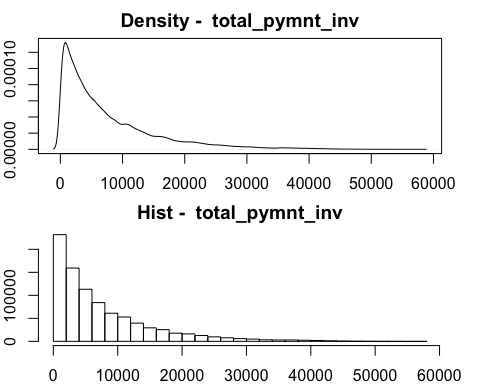
## [1] "out\_prncp\_inv"



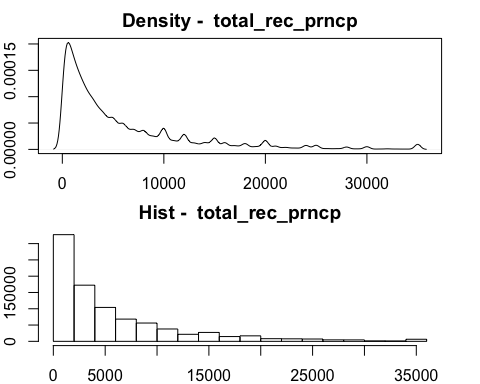
## [1] "total\_pymnt"



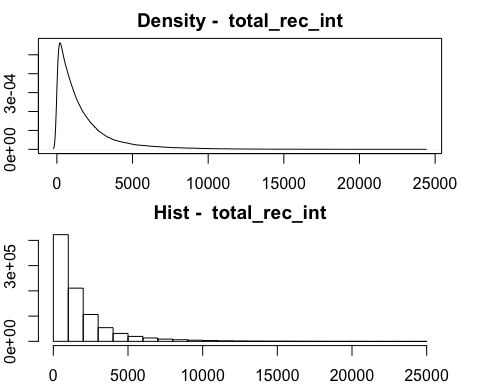
## [1] "total\_pymnt\_inv"



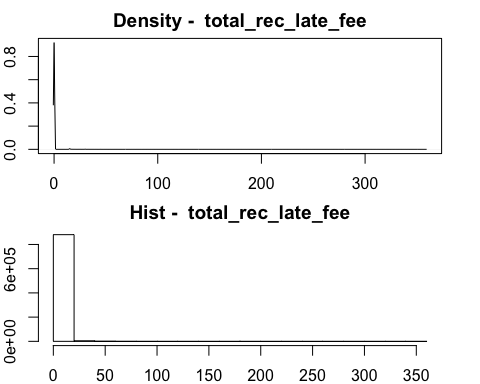
## [1] "total\_rec\_prncp"



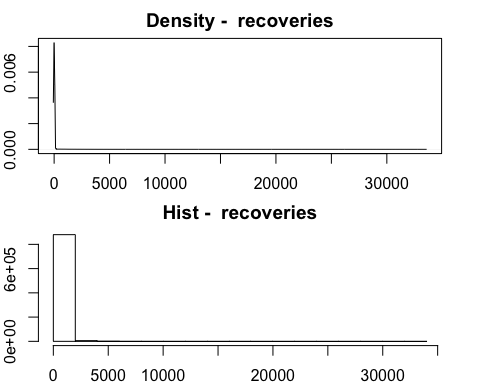
## [1] "total\_rec\_int"



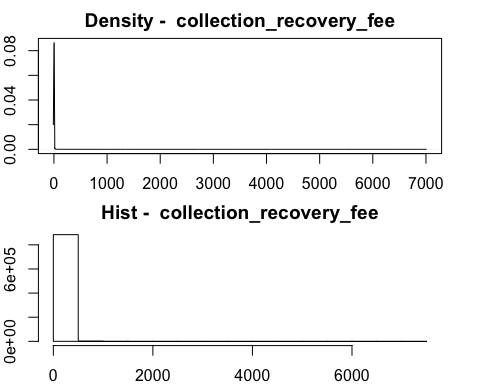
## [1] "total\_rec\_late\_fee"



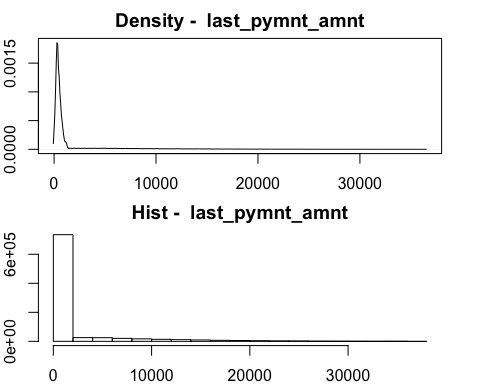
## [1] "recoveries"



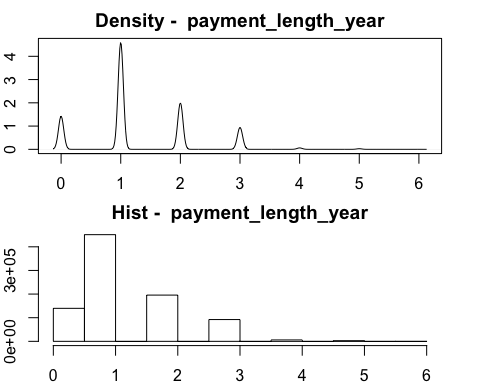
## [1] "collection\_recovery\_fee"



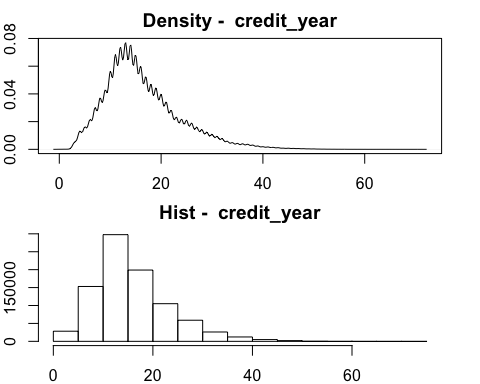
## [1] "last\_pymnt\_amnt"



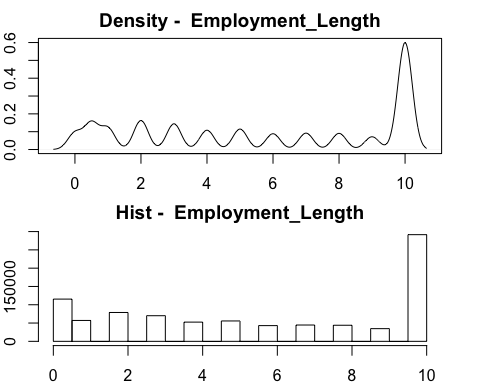
## [1] "payment\_length\_year"



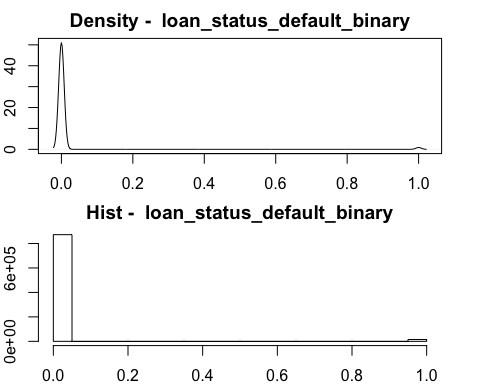
## [1] "credit\_year"



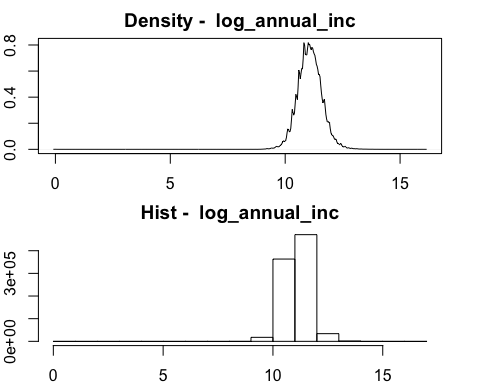
## [1] "Employment\_Length"



## [1] "loan\_status\_default\_binary"



## [1] "log\_annual\_inc"



##check feature "total\_rev\_hi\_lim" = 9999999  
  
total\_rev\_hi\_lim\_extreme= subset(loan\_df, total\_rev\_hi\_lim==9999999)  
total\_rev\_hi\_lim\_extreme

## tot\_coll\_amt tot\_cur\_bal total\_rev\_hi\_lim revol\_util  
## 129959 0 3881449 9999999 20.4  
## 301371 0 3840795 9999999 16.3  
## 755747 0 4447397 9999999 37.4  
## collections\_12\_mths\_ex\_med delinq\_2yrs inq\_last\_6mths open\_acc  
## 129959 0 0 2 13  
## 301371 0 0 0 16  
## 755747 0 0 0 9  
## pub\_rec total\_acc acc\_now\_delinq loan\_amnt funded\_amnt  
## 129959 0 28 0 28000 28000  
## 301371 0 31 0 26200 26200  
## 755747 1 17 0 35000 35000  
## funded\_amnt\_inv term int\_rate installment grade sub\_grade  
## 129959 28000 36 months 8.90 889.09 A A5  
## 301371 26200 36 months 7.69 817.28 A A4  
## 755747 35000 36 months 16.55 1240.03 D D2  
## home\_ownership verification\_status purpose dti  
## 129959 MORTGAGE Verified credit\_card 12.54  
## 301371 MORTGAGE Verified debt\_consolidation 13.30  
## 755747 OWN Verified moving 24.83  
## revol\_bal initial\_list\_status out\_prncp out\_prncp\_inv total\_pymnt  
## 129959 2568995 f 4351.81 4351.81 27557.23  
## 301371 2560703 f 16009.95 16009.95 12259.20  
## 755747 2904836 f 29474.30 29474.30 8715.95  
## total\_pymnt\_inv total\_rec\_prncp total\_rec\_int total\_rec\_late\_fee  
## 129959 27557.23 23648.19 3909.04 0  
## 301371 12259.20 10190.05 2069.15 0  
## 755747 8715.95 5525.70 3190.25 0  
## recoveries collection\_recovery\_fee last\_pymnt\_amnt application\_type  
## 129959 0 0 889.09 INDIVIDUAL  
## 301371 0 0 817.28 INDIVIDUAL  
## 755747 0 0 1240.03 INDIVIDUAL  
## verification\_status\_joint issue\_year last\_pay\_year  
## 129959 2013 2016  
## 301371 2014 2016  
## 755747 2015 2016  
## payment\_length\_year credit\_year Region Employment\_Length  
## 129959 3 27 NORTH\_EAST 1  
## 301371 2 28 NORTH\_EAST 3  
## 755747 1 19 SOUTH\_EAST 10  
## loan\_status\_default\_binary log\_annual\_inc  
## 129959 0 13.81551  
## 301371 0 13.81551  
## 755747 0 12.89922

## these three people have high annual income. the exreme values make snese to them  
##delete the rows with NA in categorical variables  
loan\_df=loan\_df[complete.cases(loan\_df),]  
  
  
  
##split the dataset into training and testing  
##take 70% for training  
train.ind = sample(1:dim(loan\_df)[1], 0.7\*dim(loan\_df)[1])  
train = loan\_df[train.ind,]  
test = loan\_df[-train.ind,]  
table(train$loan\_status\_default\_binary)

##   
## 0 1   
## 610559 10606

table(test$loan\_status\_default\_binary)

##   
## 0 1   
## 261653 4561

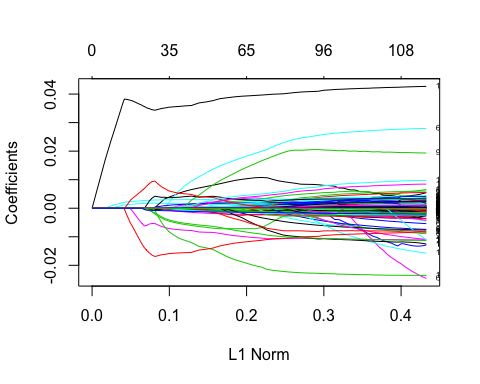
#regulization for feature selection  
ind\_v = within(train, rm("loan\_status\_default\_binary"))  
dim(train[!complete.cases(train),])

## [1] 0 45

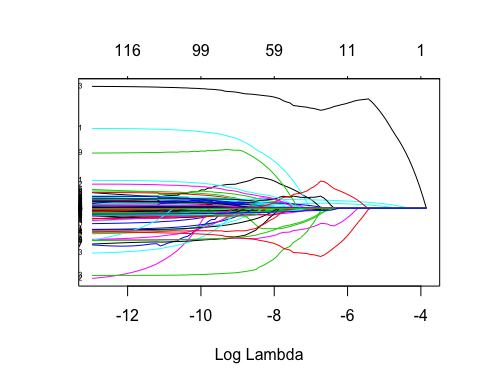
ind\_v\_matrix = model.matrix(~., ind\_v)  
dim(ind\_v\_matrix)

## [1] 621165 133

dep\_v = train$loan\_status\_default\_binary  
#elasticnet=> if we do not change the parameter, by default it's Lasso  
#glmnet function => standarize all features   
#if we don't have glmnet, we use scale to transform the featrues M=0, sd=1  
fit1 = glmnet(x = ind\_v\_matrix, y = dep\_v)  
plot(fit1, label = T)



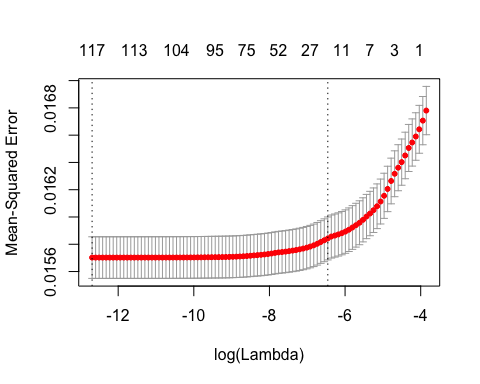
#get coefficient   
vnat =coef(fit1)  
##move the intercept  
vnat=vnat[-c(1,2), ncol(vnat)]  
plot.new()  
plot(fit1, label =T, xvar="lambda", yaxt='n', ylab="")



#deviance\_model = 2\*(loglikelihood\_saturated\_model - loglh\_current\_model)  
#deviance\_null = 2\*(loglikelihood\_saturated\_model - loglh\_intercept\_only\_model)  
print(fit1)

##   
## Call: glmnet(x = ind\_v\_matrix, y = dep\_v)   
##   
## Df %Dev Lambda  
## [1,] 0 0.000000 2.125e-02  
## [2,] 1 0.004566 1.936e-02  
## [3,] 1 0.008357 1.764e-02  
## [4,] 1 0.011500 1.607e-02  
## [5,] 1 0.014120 1.464e-02  
## [6,] 2 0.016310 1.334e-02  
## [7,] 2 0.019820 1.216e-02  
## [8,] 2 0.022730 1.108e-02  
## [9,] 2 0.025150 1.009e-02  
## [10,] 3 0.027790 9.197e-03  
## [11,] 5 0.030950 8.380e-03  
## [12,] 6 0.034450 7.635e-03  
## [13,] 6 0.037460 6.957e-03  
## [14,] 6 0.039950 6.339e-03  
## [15,] 6 0.042020 5.776e-03  
## [16,] 7 0.043740 5.263e-03  
## [17,] 7 0.045210 4.795e-03  
## [18,] 8 0.046510 4.369e-03  
## [19,] 9 0.047960 3.981e-03  
## [20,] 9 0.049240 3.627e-03  
## [21,] 9 0.050300 3.305e-03  
## [22,] 11 0.051310 3.011e-03  
## [23,] 11 0.052290 2.744e-03  
## [24,] 11 0.053100 2.500e-03  
## [25,] 11 0.053780 2.278e-03  
## [26,] 12 0.054350 2.076e-03  
## [27,] 13 0.054830 1.891e-03  
## [28,] 13 0.055260 1.723e-03  
## [29,] 16 0.056090 1.570e-03  
## [30,] 18 0.056900 1.431e-03  
## [31,] 24 0.057680 1.304e-03  
## [32,] 25 0.058450 1.188e-03  
## [33,] 26 0.059150 1.082e-03  
## [34,] 27 0.059700 9.861e-04  
## [35,] 27 0.060170 8.985e-04  
## [36,] 31 0.060580 8.187e-04  
## [37,] 35 0.060930 7.460e-04  
## [38,] 39 0.061240 6.797e-04  
## [39,] 41 0.061500 6.193e-04  
## [40,] 43 0.061730 5.643e-04  
## [41,] 47 0.061970 5.142e-04  
## [42,] 49 0.062140 4.685e-04  
## [43,] 52 0.062310 4.269e-04  
## [44,] 56 0.062530 3.889e-04  
## [45,] 58 0.062820 3.544e-04  
## [46,] 59 0.063050 3.229e-04  
## [47,] 64 0.063240 2.942e-04  
## [48,] 65 0.063410 2.681e-04  
## [49,] 69 0.063550 2.443e-04  
## [50,] 71 0.063660 2.226e-04  
## [51,] 72 0.063830 2.028e-04  
## [52,] 75 0.063970 1.848e-04  
## [53,] 76 0.064080 1.684e-04  
## [54,] 80 0.064180 1.534e-04  
## [55,] 82 0.064270 1.398e-04  
## [56,] 84 0.064340 1.274e-04  
## [57,] 87 0.064400 1.160e-04  
## [58,] 87 0.064450 1.057e-04  
## [59,] 88 0.064460 9.635e-05  
## [60,] 91 0.064490 8.779e-05  
## [61,] 95 0.064520 7.999e-05  
## [62,] 96 0.064530 7.288e-05  
## [63,] 96 0.064560 6.641e-05  
## [64,] 97 0.064610 6.051e-05  
## [65,] 101 0.064620 5.513e-05  
## [66,] 99 0.064660 5.023e-05  
## [67,] 102 0.064660 4.577e-05  
## [68,] 99 0.064690 4.171e-05  
## [69,] 103 0.064690 3.800e-05  
## [70,] 103 0.064690 3.462e-05  
## [71,] 102 0.064720 3.155e-05  
## [72,] 104 0.064720 2.875e-05  
## [73,] 104 0.064720 2.619e-05  
## [74,] 104 0.064740 2.387e-05  
## [75,] 104 0.064740 2.175e-05  
## [76,] 107 0.064740 1.981e-05  
## [77,] 108 0.064740 1.805e-05  
## [78,] 109 0.064740 1.645e-05  
## [79,] 110 0.064750 1.499e-05  
## [80,] 108 0.064760 1.366e-05  
## [81,] 110 0.064760 1.244e-05  
## [82,] 111 0.064760 1.134e-05  
## [83,] 111 0.064760 1.033e-05  
## [84,] 113 0.064760 9.413e-06  
## [85,] 114 0.064760 8.577e-06  
## [86,] 114 0.064760 7.815e-06  
## [87,] 114 0.064760 7.121e-06  
## [88,] 116 0.064760 6.488e-06  
## [89,] 116 0.064770 5.912e-06  
## [90,] 117 0.064770 5.386e-06  
## [91,] 118 0.064770 4.908e-06  
## [92,] 118 0.064770 4.472e-06  
## [93,] 118 0.064770 4.075e-06  
## [94,] 118 0.064770 3.713e-06  
## [95,] 118 0.064770 3.383e-06  
## [96,] 117 0.064770 3.082e-06  
## [97,] 120 0.064770 2.809e-06  
## [98,] 120 0.064770 2.559e-06  
## [99,] 120 0.064770 2.332e-06

##use cv.glmnet to make k-fold cross validation / default is 10 fold  
##select the best lambda   
cvfit = cv.glmnet(ind\_v\_matrix,dep\_v, nfolds = 5)  
plot(cvfit)



#pick log(lambda) = -8  
##check the coeficient when lambda is fixed   
coef(fit1, s = 1/exp(7))

## 134 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 5.340117e-03  
## (Intercept) .   
## tot\_coll\_amt .   
## tot\_cur\_bal -1.412780e-09  
## total\_rev\_hi\_lim .   
## revol\_util .   
## collections\_12\_mths\_ex\_med .   
## delinq\_2yrs .   
## inq\_last\_6mths 8.779489e-05  
## open\_acc .   
## pub\_rec .   
## total\_acc .   
## acc\_now\_delinq .   
## loan\_amnt -9.845680e-07  
## funded\_amnt -4.962101e-08  
## funded\_amnt\_inv .   
## term 60 months -3.896172e-03  
## term0 .   
## int\_rate 1.698447e-03  
## installment .   
## gradeA .   
## gradeB .   
## gradeC .   
## gradeD .   
## gradeE .   
## gradeF .   
## gradeG 3.103927e-03  
## sub\_gradeA1 .   
## sub\_gradeA2 .   
## sub\_gradeA3 .   
## sub\_gradeA4 .   
## sub\_gradeA5 .   
## sub\_gradeB1 .   
## sub\_gradeB2 .   
## sub\_gradeB3 .   
## sub\_gradeB4 .   
## sub\_gradeB5 .   
## sub\_gradeC1 .   
## sub\_gradeC2 .   
## sub\_gradeC3 .   
## sub\_gradeC4 .   
## sub\_gradeC5 .   
## sub\_gradeD1 .   
## sub\_gradeD2 .   
## sub\_gradeD3 .   
## sub\_gradeD4 .   
## sub\_gradeD5 .   
## sub\_gradeE1 .   
## sub\_gradeE2 .   
## sub\_gradeE3 .   
## sub\_gradeE4 .   
## sub\_gradeE5 .   
## sub\_gradeF1 .   
## sub\_gradeF2 .   
## sub\_gradeF3 .   
## sub\_gradeF4 .   
## sub\_gradeF5 .   
## sub\_gradeG1 .   
## sub\_gradeG2 .   
## sub\_gradeG3 .   
## sub\_gradeG4 .   
## sub\_gradeG5 .   
## home\_ownershipANY .   
## home\_ownershipMORTGAGE -5.167370e-04  
## home\_ownershipNONE .   
## home\_ownershipOTHER .   
## home\_ownershipOWN .   
## home\_ownershipRENT 3.179962e-04  
## verification\_statusNot Verified -4.073562e-04  
## verification\_statusSource Verified .   
## verification\_statusVerified .   
## purposecar .   
## purposecredit\_card .   
## purposedebt\_consolidation .   
## purposeeducational .   
## purposehome\_improvement .   
## purposehouse .   
## purposemajor\_purchase .   
## purposemedical .   
## purposemoving .   
## purposeother .   
## purposerenewable\_energy .   
## purposesmall\_business .   
## purposevacation .   
## purposewedding .   
## dti .   
## revol\_bal .   
## initial\_list\_statusf 8.840304e-04  
## initial\_list\_statusw .   
## out\_prncp 1.991816e-06  
## out\_prncp\_inv 6.001351e-08  
## total\_pymnt .   
## total\_pymnt\_inv .   
## total\_rec\_prncp .   
## total\_rec\_int 1.433286e-06  
## total\_rec\_late\_fee 2.933513e-03  
## recoveries -3.379565e-06  
## collection\_recovery\_fee -7.570525e-06  
## last\_pymnt\_amnt -1.197013e-06  
## application\_typeINDIVIDUAL .   
## application\_typeJOINT .   
## verification\_status\_joint0 .   
## verification\_status\_jointNot Verified .   
## verification\_status\_jointSource Verified .   
## verification\_status\_jointVerified .   
## issue\_year2007 .   
## issue\_year2008 .   
## issue\_year2009 .   
## issue\_year2010 .   
## issue\_year2011 .   
## issue\_year2012 -6.660459e-03  
## issue\_year2013 3.848418e-03  
## issue\_year2014 6.994339e-03  
## issue\_year2015 -6.262831e-03  
## last\_pay\_year0 .   
## last\_pay\_year2007 .   
## last\_pay\_year2008 .   
## last\_pay\_year2009 .   
## last\_pay\_year2010 .   
## last\_pay\_year2011 .   
## last\_pay\_year2012 .   
## last\_pay\_year2013 .   
## last\_pay\_year2014 .   
## last\_pay\_year2015 3.507998e-02  
## last\_pay\_year2016 -1.621408e-02  
## payment\_length\_year -3.451423e-03  
## credit\_year .   
## RegionMID\_WEST .   
## RegionNORTH\_EAST .   
## RegionSOUTH\_EAST .   
## RegionSOUTH\_WEST .   
## RegionWEST .   
## Employment\_Length -3.678688e-05  
## log\_annual\_inc -8.652786e-04

##only keep the selected features  
  
  
train\_selectedV=train[,c("tot\_cur\_bal","inq\_last\_6mths","funded\_amnt","term", "int\_rate", "home\_ownership", "verification\_status","initial\_list\_status","out\_prncp","out\_prncp\_inv", "total\_rec\_prncp","total\_rec\_int","total\_rec\_late\_fee","recoveries","collection\_recovery\_fee","last\_pymnt\_amnt", "issue\_year", "last\_pay\_year","Employment\_Length","log\_annual\_inc")]  
train\_selectedV$loan\_status\_default\_binary=train$loan\_status\_default\_binary  
  
logis.mod = glm(loan\_status\_default\_binary ~., train\_selectedV, family = 'binomial')

## Warning: glm.fit: algorithm did not converge

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

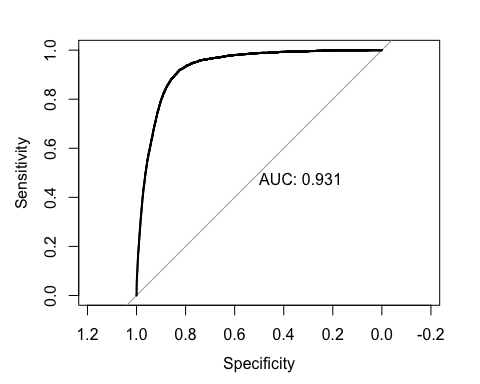
summary(logis.mod)

##   
## Call:  
## glm(formula = loan\_status\_default\_binary ~ ., family = "binomial",   
## data = train\_selectedV)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -7.4899 -0.1011 -0.0577 -0.0313 7.3744   
##   
## Coefficients: (5 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -6.079e+00 3.093e-01 -19.656 < 2e-16  
## tot\_cur\_bal -2.911e-07 1.015e-07 -2.869 0.00412  
## inq\_last\_6mths 6.842e-02 1.034e-02 6.619 3.62e-11  
## funded\_amnt -1.322e-03 1.089e-04 -12.146 < 2e-16  
## term 60 months -6.465e-01 3.225e-02 -20.048 < 2e-16  
## term0 -2.049e+01 7.771e+04 0.000 0.99979  
## int\_rate 1.253e-01 3.082e-03 40.659 < 2e-16  
## home\_ownershipANY -2.201e+01 1.821e+05 0.000 0.99990  
## home\_ownershipMORTGAGE -1.384e-01 2.646e-02 -5.229 1.70e-07  
## home\_ownershipNONE -2.026e+01 5.398e+04 0.000 0.99970  
## home\_ownershipOTHER -1.926e+01 2.146e+04 -0.001 0.99928  
## home\_ownershipOWN -8.294e-02 3.650e-02 -2.272 0.02307  
## home\_ownershipRENT NA NA NA NA  
## verification\_statusNot Verified -1.253e-01 3.005e-02 -4.169 3.05e-05  
## verification\_statusSource Verified 1.273e-02 2.466e-02 0.516 0.60576  
## verification\_statusVerified NA NA NA NA  
## initial\_list\_statusf 1.486e-01 2.279e-02 6.521 6.99e-11  
## initial\_list\_statusw NA NA NA NA  
## out\_prncp -1.345e-03 5.223e-04 -2.574 0.01004  
## out\_prncp\_inv 2.730e-03 5.112e-04 5.341 9.25e-08  
## total\_rec\_prncp 1.252e-03 1.088e-04 11.509 < 2e-16  
## total\_rec\_int 2.476e-05 8.210e-06 3.016 0.00256  
## total\_rec\_late\_fee 7.161e-02 1.414e-03 50.648 < 2e-16  
## recoveries -5.914e-01 4.212e+00 -0.140 0.88832  
## collection\_recovery\_fee -4.515e-01 4.446e+01 -0.010 0.99190  
## last\_pymnt\_amnt -8.348e-04 3.043e-05 -27.430 < 2e-16  
## issue\_year2007 -9.414e+00 1.632e+04 -0.001 0.99954  
## issue\_year2008 -1.072e+01 7.887e+03 -0.001 0.99892  
## issue\_year2009 -1.575e+01 3.847e+03 -0.004 0.99673  
## issue\_year2010 -1.305e+00 4.883e-01 -2.673 0.00752  
## issue\_year2011 8.355e-01 1.750e-01 4.774 1.81e-06  
## issue\_year2012 -7.277e-01 8.872e-02 -8.203 2.34e-16  
## issue\_year2013 1.025e+00 4.750e-02 21.577 < 2e-16  
## issue\_year2014 1.034e+00 3.035e-02 34.070 < 2e-16  
## issue\_year2015 NA NA NA NA  
## last\_pay\_year0 NA NA NA NA  
## last\_pay\_year2007 -8.524e+00 2.383e+05 0.000 0.99997  
## last\_pay\_year2008 -7.100e+00 1.979e+04 0.000 0.99971  
## last\_pay\_year2009 -7.388e+00 1.214e+04 -0.001 0.99951  
## last\_pay\_year2010 -1.552e+01 6.426e+03 -0.002 0.99807  
## last\_pay\_year2011 -1.672e+01 4.073e+03 -0.004 0.99673  
## last\_pay\_year2012 -1.779e+01 2.990e+03 -0.006 0.99525  
## last\_pay\_year2013 -1.838e+01 1.922e+03 -0.010 0.99237  
## last\_pay\_year2014 -1.883e+01 1.200e+03 -0.016 0.98748  
## last\_pay\_year2015 2.465e+00 9.709e-02 25.386 < 2e-16  
## last\_pay\_year2016 -1.090e+00 1.022e-01 -10.661 < 2e-16  
## Employment\_Length -1.194e-02 2.880e-03 -4.146 3.38e-05  
## log\_annual\_inc -8.643e-02 2.712e-02 -3.187 0.00144  
##   
## (Intercept) \*\*\*  
## tot\_cur\_bal \*\*   
## inq\_last\_6mths \*\*\*  
## funded\_amnt \*\*\*  
## term 60 months \*\*\*  
## term0   
## int\_rate \*\*\*  
## home\_ownershipANY   
## home\_ownershipMORTGAGE \*\*\*  
## home\_ownershipNONE   
## home\_ownershipOTHER   
## home\_ownershipOWN \*   
## home\_ownershipRENT   
## verification\_statusNot Verified \*\*\*  
## verification\_statusSource Verified   
## verification\_statusVerified   
## initial\_list\_statusf \*\*\*  
## initial\_list\_statusw   
## out\_prncp \*   
## out\_prncp\_inv \*\*\*  
## total\_rec\_prncp \*\*\*  
## total\_rec\_int \*\*   
## total\_rec\_late\_fee \*\*\*  
## recoveries   
## collection\_recovery\_fee   
## last\_pymnt\_amnt \*\*\*  
## issue\_year2007   
## issue\_year2008   
## issue\_year2009   
## issue\_year2010 \*\*   
## issue\_year2011 \*\*\*  
## issue\_year2012 \*\*\*  
## issue\_year2013 \*\*\*  
## issue\_year2014 \*\*\*  
## issue\_year2015   
## last\_pay\_year0   
## last\_pay\_year2007   
## last\_pay\_year2008   
## last\_pay\_year2009   
## last\_pay\_year2010   
## last\_pay\_year2011   
## last\_pay\_year2012   
## last\_pay\_year2013   
## last\_pay\_year2014   
## last\_pay\_year2015 \*\*\*  
## last\_pay\_year2016 \*\*\*  
## Employment\_Length \*\*\*  
## log\_annual\_inc \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 107366 on 621164 degrees of freedom  
## Residual deviance: 72493 on 621122 degrees of freedom  
## AIC: 72579  
##   
## Number of Fisher Scoring iterations: 25

test\_selectedV = test[,c("tot\_cur\_bal","inq\_last\_6mths","funded\_amnt","term", "int\_rate", "home\_ownership", "verification\_status","initial\_list\_status","out\_prncp","out\_prncp\_inv", "total\_rec\_prncp","total\_rec\_int","total\_rec\_late\_fee","recoveries","collection\_recovery\_fee","last\_pymnt\_amnt", "issue\_year", "last\_pay\_year","Employment\_Length","log\_annual\_inc")]  
pred = predict(logis.mod, test\_selectedV)

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

plot.roc(test$loan\_status\_default\_binary, pred, print.auc=T)



table(train\_selectedV$last\_pay\_year, train\_selectedV$issue\_year)

##   
## 0 2007 2008 2009 2010 2011 2012 2013 2014  
## 0 0 3 12 21 29 37 64 99  
## 0 21 0 0 0 0 0 0 0 0  
## 2007 0 2 0 0 0 0 0 0 0  
## 2008 0 89 144 0 0 0 0 0 0  
## 2009 0 73 296 211 0 0 0 0 0  
## 2010 0 200 339 676 427 0 0 0 0  
## 2011 0 17 848 847 1589 762 0 0 0  
## 2012 0 1 27 1854 1787 2842 1755 0 0  
## 2013 0 0 1 83 3552 3166 7327 4911 0  
## 2014 0 0 0 4 523 6011 9094 20664 10353  
## 2015 0 0 0 2 779 1427 17243 33888 66705  
## 2016 0 0 0 0 37 1029 2099 34794 87705  
##   
## 2015  
## 12160  
## 0 0  
## 2007 0  
## 2008 0  
## 2009 0  
## 2010 0  
## 2011 0  
## 2012 0  
## 2013 0  
## 2014 0  
## 2015 79127  
## 2016 203409