Final Project

Amol Gote

08/09/2020

lendingClubLoanData <- fread("data/Lending\_club\_dataset.csv", header = TRUE)

nrow(lendingClubLoanData)

## [1] 2260701

lendingClubLoanData$orig\_year<-substr(lendingClubLoanData$issue\_d,5,8)

lendingClubLoanData\_2018 <- lendingClubLoanData %>%  
 filter(orig\_year == 2018)

nrow(lendingClubLoanData\_2018)

## [1] 495242

nrow(lendingClubLoanData\_2018)

## [1] 495242

write.csv(lendingClubLoanData\_2018, file = "data/lending\_club\_loan\_data\_2018\_final.csv", row.names=FALSE)

final\_dataset <- fread("data/lending\_club\_loan\_data\_2018\_final.csv", header = TRUE)  
drops <- c("id","member\_id", "url", "desc")  
final\_dataset[ , !(names(final\_dataset) %in% drops)]

## [1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [13] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE  
## [25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [49] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [85] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [97] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [109] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [133] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [145] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

names(final\_dataset)

## [1] "id"   
## [2] "member\_id"   
## [3] "loan\_amnt"   
## [4] "funded\_amnt"   
## [5] "funded\_amnt\_inv"   
## [6] "term"   
## [7] "int\_rate"   
## [8] "installment"   
## [9] "grade"   
## [10] "sub\_grade"   
## [11] "emp\_title"   
## [12] "emp\_length"   
## [13] "home\_ownership"   
## [14] "annual\_inc"   
## [15] "verification\_status"   
## [16] "issue\_d"   
## [17] "loan\_status"   
## [18] "pymnt\_plan"   
## [19] "url"   
## [20] "desc"   
## [21] "purpose"   
## [22] "title"   
## [23] "zip\_code"   
## [24] "addr\_state"   
## [25] "dti"   
## [26] "delinq\_2yrs"   
## [27] "earliest\_cr\_line"   
## [28] "fico\_range\_low"   
## [29] "fico\_range\_high"   
## [30] "inq\_last\_6mths"   
## [31] "mths\_since\_last\_delinq"   
## [32] "mths\_since\_last\_record"   
## [33] "open\_acc"   
## [34] "pub\_rec"   
## [35] "revol\_bal"   
## [36] "revol\_util"   
## [37] "total\_acc"   
## [38] "initial\_list\_status"   
## [39] "out\_prncp"   
## [40] "out\_prncp\_inv"   
## [41] "total\_pymnt"   
## [42] "total\_pymnt\_inv"   
## [43] "total\_rec\_prncp"   
## [44] "total\_rec\_int"   
## [45] "total\_rec\_late\_fee"   
## [46] "recoveries"   
## [47] "collection\_recovery\_fee"   
## [48] "last\_pymnt\_d"   
## [49] "last\_pymnt\_amnt"   
## [50] "next\_pymnt\_d"   
## [51] "last\_credit\_pull\_d"   
## [52] "last\_fico\_range\_high"   
## [53] "last\_fico\_range\_low"   
## [54] "collections\_12\_mths\_ex\_med"   
## [55] "mths\_since\_last\_major\_derog"   
## [56] "policy\_code"   
## [57] "application\_type"   
## [58] "annual\_inc\_joint"   
## [59] "dti\_joint"   
## [60] "verification\_status\_joint"   
## [61] "acc\_now\_delinq"   
## [62] "tot\_coll\_amt"   
## [63] "tot\_cur\_bal"   
## [64] "open\_acc\_6m"   
## [65] "open\_act\_il"   
## [66] "open\_il\_12m"   
## [67] "open\_il\_24m"   
## [68] "mths\_since\_rcnt\_il"   
## [69] "total\_bal\_il"   
## [70] "il\_util"   
## [71] "open\_rv\_12m"   
## [72] "open\_rv\_24m"   
## [73] "max\_bal\_bc"   
## [74] "all\_util"   
## [75] "total\_rev\_hi\_lim"   
## [76] "inq\_fi"   
## [77] "total\_cu\_tl"   
## [78] "inq\_last\_12m"   
## [79] "acc\_open\_past\_24mths"   
## [80] "avg\_cur\_bal"   
## [81] "bc\_open\_to\_buy"   
## [82] "bc\_util"   
## [83] "chargeoff\_within\_12\_mths"   
## [84] "delinq\_amnt"   
## [85] "mo\_sin\_old\_il\_acct"   
## [86] "mo\_sin\_old\_rev\_tl\_op"   
## [87] "mo\_sin\_rcnt\_rev\_tl\_op"   
## [88] "mo\_sin\_rcnt\_tl"   
## [89] "mort\_acc"   
## [90] "mths\_since\_recent\_bc"   
## [91] "mths\_since\_recent\_bc\_dlq"   
## [92] "mths\_since\_recent\_inq"   
## [93] "mths\_since\_recent\_revol\_delinq"   
## [94] "num\_accts\_ever\_120\_pd"   
## [95] "num\_actv\_bc\_tl"   
## [96] "num\_actv\_rev\_tl"   
## [97] "num\_bc\_sats"   
## [98] "num\_bc\_tl"   
## [99] "num\_il\_tl"   
## [100] "num\_op\_rev\_tl"   
## [101] "num\_rev\_accts"   
## [102] "num\_rev\_tl\_bal\_gt\_0"   
## [103] "num\_sats"   
## [104] "num\_tl\_120dpd\_2m"   
## [105] "num\_tl\_30dpd"   
## [106] "num\_tl\_90g\_dpd\_24m"   
## [107] "num\_tl\_op\_past\_12m"   
## [108] "pct\_tl\_nvr\_dlq"   
## [109] "percent\_bc\_gt\_75"   
## [110] "pub\_rec\_bankruptcies"   
## [111] "tax\_liens"   
## [112] "tot\_hi\_cred\_lim"   
## [113] "total\_bal\_ex\_mort"   
## [114] "total\_bc\_limit"   
## [115] "total\_il\_high\_credit\_limit"   
## [116] "revol\_bal\_joint"   
## [117] "sec\_app\_fico\_range\_low"   
## [118] "sec\_app\_fico\_range\_high"   
## [119] "sec\_app\_earliest\_cr\_line"   
## [120] "sec\_app\_inq\_last\_6mths"   
## [121] "sec\_app\_mort\_acc"   
## [122] "sec\_app\_open\_acc"   
## [123] "sec\_app\_revol\_util"   
## [124] "sec\_app\_open\_act\_il"   
## [125] "sec\_app\_num\_rev\_accts"   
## [126] "sec\_app\_chargeoff\_within\_12\_mths"   
## [127] "sec\_app\_collections\_12\_mths\_ex\_med"   
## [128] "sec\_app\_mths\_since\_last\_major\_derog"   
## [129] "hardship\_flag"   
## [130] "hardship\_type"   
## [131] "hardship\_reason"   
## [132] "hardship\_status"   
## [133] "deferral\_term"   
## [134] "hardship\_amount"   
## [135] "hardship\_start\_date"   
## [136] "hardship\_end\_date"   
## [137] "payment\_plan\_start\_date"   
## [138] "hardship\_length"   
## [139] "hardship\_dpd"   
## [140] "hardship\_loan\_status"   
## [141] "orig\_projected\_additional\_accrued\_interest"  
## [142] "hardship\_payoff\_balance\_amount"   
## [143] "hardship\_last\_payment\_amount"   
## [144] "disbursement\_method"   
## [145] "debt\_settlement\_flag"   
## [146] "debt\_settlement\_flag\_date"   
## [147] "settlement\_status"   
## [148] "settlement\_date"   
## [149] "settlement\_amount"   
## [150] "settlement\_percentage"   
## [151] "settlement\_term"   
## [152] "orig\_year"

ds\_lc <- final\_dataset[,c("loan\_amnt",  
 "funded\_amnt",  
 "funded\_amnt\_inv",  
 "term",  
 "int\_rate",  
 "installment",  
 "grade",  
 "sub\_grade",  
 "home\_ownership",  
 "annual\_inc",  
 "loan\_status",  
 "dti",  
 "delinq\_2yrs",  
 "fico\_range\_low",  
 "fico\_range\_high",  
 "inq\_last\_6mths",  
 "mths\_since\_last\_delinq",  
 "mths\_since\_last\_record",  
 "open\_acc",  
 "pub\_rec",  
 "revol\_bal",  
 "revol\_util",  
 "total\_acc",  
 "out\_prncp",  
 "out\_prncp\_inv",  
 "total\_pymnt",  
 "total\_pymnt\_inv",  
 "total\_rec\_prncp",  
 "total\_rec\_int",  
 "total\_rec\_late\_fee",  
 "recoveries",  
 "collection\_recovery\_fee",  
 "last\_fico\_range\_high",  
 "last\_fico\_range\_low",  
 "mths\_since\_last\_major\_derog",  
 "acc\_now\_delinq",  
 "avg\_cur\_bal"  
 )]  
  
  
  
ds\_lc <- final\_dataset[,c(  
 "loan\_amnt",  
 "term",  
 "int\_rate",  
 "installment",  
 "grade",  
 "sub\_grade",  
 "home\_ownership",  
 "annual\_inc",  
 "loan\_status",  
 "dti",  
 "delinq\_2yrs",  
 "fico\_range\_low",  
 "fico\_range\_high",  
 "inq\_last\_6mths",  
 "open\_acc",  
 "pub\_rec"  
 )]

nrow(ds\_lc)

## [1] 495242

ds\_lc <- na.omit(ds\_lc, cols = c("dti", "int\_rate"))  
nrow(ds\_lc)

## [1] 494110

levels(factor(ds\_lc$loan\_status))

## [1] "Charged Off" "Current" "Default"   
## [4] "Fully Paid" "In Grace Period" "Late (16-30 days)"   
## [7] "Late (31-120 days)"

levels(factor(ds\_lc$grade))

## [1] "A" "B" "C" "D" "E" "F" "G"

levels(factor(ds\_lc$sub\_grade))

## [1] "A1" "A2" "A3" "A4" "A5" "B1" "B2" "B3" "B4" "B5" "C1" "C2" "C3" "C4" "C5"  
## [16] "D1" "D2" "D3" "D4" "D5" "E1" "E2" "E3" "E4" "E5" "F1" "F2" "F3" "F4" "F5"  
## [31] "G1" "G2" "G3" "G4" "G5"

levels(factor(ds\_lc$home\_ownership))

## [1] "ANY" "MORTGAGE" "OWN" "RENT"

ds\_lc$term <- as.integer(as.factor(ds\_lc$term))   
ds\_lc$grade <- as.integer(as.factor(ds\_lc$grade))  
ds\_lc$sub\_grade <- as.integer(as.factor(ds\_lc$sub\_grade))  
ds\_lc$home\_ownership <- as.integer(as.factor(ds\_lc$home\_ownership))  
ds\_lc$int\_rate <- as.numeric(ds\_lc$int\_rate)  
#ds\_lc$mths\_since\_last\_record[is.na(ds\_lc$mths\_since\_last\_record)] <- 0  
#ds\_lc$mths\_since\_last\_major\_derog[is.na(ds\_lc$mths\_since\_last\_major\_derog)] <- 0  
#ds\_lc$revol\_util[is.na(ds\_lc$revol\_util)] <- mean(ds\_lc$revol\_util, na.rm = T)  
ds\_lc$loan\_status <- as.integer(as.factor(ds\_lc$loan\_status))  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 1] <- 1  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 2] <- 0  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 3] <- 1  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 4] <- 0  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 5] <- 1  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 6] <- 0  
ds\_lc$is\_acct\_delinquent[ds\_lc$loan\_status == 7] <- 1  
ds\_lc$is\_acct\_delinquent <- factor(ds\_lc$is\_acct\_delinquent)  
levels(factor(ds\_lc$is\_acct\_delinquent))

## [1] "0" "1"

ds\_lc = subset(ds\_lc, select = -c(loan\_status))  
#ds\_lc = subset(ds\_lc, select = -c(loan\_status,  
 #mths\_since\_last\_delinq,  
 #mths\_since\_last\_record,  
 #open\_acc,  
 #pub\_rec,  
 #revol\_bal,  
 #revol\_util,  
 #total\_acc,  
 #out\_prncp,  
 #out\_prncp\_inv,  
 #total\_pymnt,  
 #total\_pymnt\_inv,  
 #total\_rec\_prncp,  
 #total\_rec\_int,  
 #total\_rec\_late\_fee,  
 #recoveries  
 #collection\_recovery\_fee,  
 #last\_fico\_range\_high,  
 #last\_fico\_range\_low,  
 #mths\_since\_last\_major\_derog,  
 #acc\_now\_delinq  
 #avg\_cur\_bal  
 #))  
#ds\_lc$is\_acct\_delinquent  
  
  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 1] <- 1  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 2] <- 0  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 3] <- 0  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 4] <- 0  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 5] <- 0  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 6] <- 0  
#ds\_lc$is\_acct\_chargedoff[ds\_lc$loan\_status == 7] <- 0  
#ds\_lc$is\_acct\_chargedoff <- factor(ds\_lc$is\_acct\_chargedoff)  
#levels(factor(ds\_lc$is\_acct\_chargedoff))  
#ds\_lc$is\_acct\_chargedoff  
#ds\_lc$int\_rate

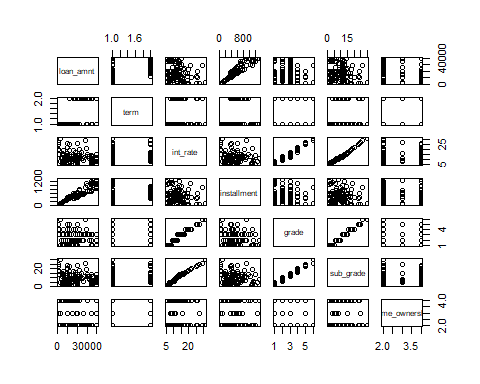
str(ds\_lc)

## Classes 'data.table' and 'data.frame': 494110 obs. of 16 variables:  
## $ loan\_amnt : int 5000 15000 11200 25000 3000 17000 20000 19200 6500 10000 ...  
## $ term : int 1 1 2 2 1 2 1 2 1 2 ...  
## $ int\_rate : num 20.39 9.92 30.79 21.85 7.34 ...  
## $ installment : num 186.8 483.4 367.8 688.4 93.1 ...  
## $ grade : int 4 2 7 4 1 4 2 4 1 3 ...  
## $ sub\_grade : int 19 7 31 20 4 19 8 16 2 11 ...  
## $ home\_ownership : int 4 3 4 2 4 4 2 4 2 2 ...  
## $ annual\_inc : num 50000 196000 44000 65000 52000 52000 19000 36500 50000 80000 ...  
## $ dti : num 21.8 18.29 43.97 12.89 0.58 ...  
## $ delinq\_2yrs : int 1 0 1 1 0 0 0 3 0 0 ...  
## $ fico\_range\_low : int 665 700 665 665 760 670 795 675 705 660 ...  
## $ fico\_range\_high : int 669 704 669 669 764 674 799 679 709 664 ...  
## $ inq\_last\_6mths : int 0 0 2 1 0 0 1 0 0 1 ...  
## $ open\_acc : int 5 19 8 7 7 9 6 14 7 17 ...  
## $ pub\_rec : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ is\_acct\_delinquent: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

summary(ds\_lc)

## loan\_amnt term int\_rate installment   
## Min. : 1000 Min. :1.000 Min. : 5.31 Min. : 29.76   
## 1st Qu.: 8000 1st Qu.:1.000 1st Qu.: 8.46 1st Qu.: 254.50   
## Median :14000 Median :1.000 Median :11.80 Median : 386.76   
## Mean :16017 Mean :1.304 Mean :12.73 Mean : 466.38   
## 3rd Qu.:22000 3rd Qu.:2.000 3rd Qu.:16.01 3rd Qu.: 628.67   
## Max. :40000 Max. :2.000 Max. :30.99 Max. :1670.15   
## grade sub\_grade home\_ownership annual\_inc dti   
## Min. :1.000 Min. : 1 Min. :1.00 Min. : 0 Min. : 0.00   
## 1st Qu.:1.000 1st Qu.: 5 1st Qu.:2.00 1st Qu.: 46000 1st Qu.: 11.43   
## Median :2.000 Median : 9 Median :3.00 Median : 66000 Median : 17.71   
## Mean :2.409 Mean :10 Mean :2.91 Mean : 80277 Mean : 19.67   
## 3rd Qu.:3.000 3rd Qu.:14 3rd Qu.:4.00 3rd Qu.: 96000 3rd Qu.: 25.03   
## Max. :7.000 Max. :35 Max. :4.00 Max. :9930475 Max. :999.00   
## delinq\_2yrs fico\_range\_low fico\_range\_high inq\_last\_6mths   
## Min. : 0.0000 Min. :660.0 Min. :664.0 Min. :0.0000   
## 1st Qu.: 0.0000 1st Qu.:680.0 1st Qu.:684.0 1st Qu.:0.0000   
## Median : 0.0000 Median :700.0 Median :704.0 Median :0.0000   
## Mean : 0.2295 Mean :706.4 Mean :710.4 Mean :0.4426   
## 3rd Qu.: 0.0000 3rd Qu.:725.0 3rd Qu.:729.0 3rd Qu.:1.0000   
## Max. :58.0000 Max. :845.0 Max. :850.0 Max. :5.0000   
## open\_acc pub\_rec is\_acct\_delinquent  
## Min. : 0.0 Min. : 0.0000 0:475138   
## 1st Qu.: 7.0 1st Qu.: 0.0000 1: 18972   
## Median : 10.0 Median : 0.0000   
## Mean : 11.5 Mean : 0.1346   
## 3rd Qu.: 14.0 3rd Qu.: 0.0000   
## Max. :101.0 Max. :52.0000

corr\_ds\_subset <- sample(nrow(ds\_lc), 100)  
corr\_ds = ds\_lc[corr\_ds\_subset, ]  
pairs(~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership  
 , corr\_ds)



head(ds\_lc)

## loan\_amnt term int\_rate installment grade sub\_grade home\_ownership  
## 1: 5000 1 20.39 186.82 4 19 4  
## 2: 15000 1 9.92 483.45 2 7 3  
## 3: 11200 2 30.79 367.82 7 31 4  
## 4: 25000 2 21.85 688.35 4 20 2  
## 5: 3000 1 7.34 93.10 1 4 4  
## 6: 17000 2 20.39 454.10 4 19 4  
## annual\_inc dti delinq\_2yrs fico\_range\_low fico\_range\_high inq\_last\_6mths  
## 1: 50000 21.80 1 665 669 0  
## 2: 196000 18.29 0 700 704 0  
## 3: 44000 43.97 1 665 669 2  
## 4: 65000 12.89 1 665 669 1  
## 5: 52000 0.58 0 760 764 0  
## 6: 52000 15.65 0 670 674 0  
## open\_acc pub\_rec is\_acct\_delinquent  
## 1: 5 0 0  
## 2: 19 0 0  
## 3: 8 0 0  
## 4: 7 0 0  
## 5: 7 0 0  
## 6: 9 0 0

sample\_set<-sample(nrow(ds\_lc),nrow(ds\_lc)\*0.6)  
training\_dataset <- ds\_lc[sample\_set,]  
test\_dataset <- ds\_lc[-sample\_set,]  
regression\_model <- glm(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec,  
 data = training\_dataset, family = "binomial")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5, 1, 0)  
validation\_error\_rate1 <- mean(prediction != test\_dataset$is\_acct\_delinquent)  
validation\_error\_rate1

## [1] 0.03885268

sample\_set<-sample(nrow(ds\_lc),nrow(ds\_lc)\*0.7)  
training\_dataset <- ds\_lc[sample\_set,]  
test\_dataset <- ds\_lc[-sample\_set,]  
regression\_model <- glm(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec,  
 data = training\_dataset, family = "binomial")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5, 1, 0)  
validation\_error\_rate2 <- mean(prediction != test\_dataset$is\_acct\_delinquent)  
validation\_error\_rate2

## [1] 0.0395391

sample\_set<-sample(nrow(ds\_lc),nrow(ds\_lc)\*0.75)  
training\_dataset <- ds\_lc[sample\_set,]  
test\_dataset <- ds\_lc[-sample\_set,]  
regression\_model <- glm(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec,  
 data = training\_dataset, family = "binomial")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5, 1, 0)  
validation\_error\_rate3 <- mean(prediction != test\_dataset$is\_acct\_delinquent)  
validation\_error\_rate3

## [1] 0.03870378

sample\_set<-sample(nrow(ds\_lc),nrow(ds\_lc)\*0.8)  
training\_dataset <- ds\_lc[sample\_set,]  
test\_dataset <- ds\_lc[-sample\_set,]  
regression\_model <- glm(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec,  
 data = training\_dataset, family = "binomial")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5, 1, 0)  
validation\_error\_rate4 <- mean(prediction != test\_dataset$is\_acct\_delinquent)  
validation\_error\_rate4

## [1] 0.03870596

Following is the data split in training and test data ration along with validation error rate/misclassification

1. Tranining : Test -> 60 : 40, validation error rate 0.0388527. or 3.8852685%   
2. Tranining : Test -> 70 : 30, validation error rate 0.0395391. or 3.9539104%   
3. Tranining : Test -> 75 : 25, validation error rate 0.0387038. or 3.8703776%   
3. Tranining : Test -> 80 : 20, validation error rate 0.0387038. or 3.8705956%

Validation error rate is different for different sample splits of taining and test dataset. This indicates that validation error rate varies by which observations are in the training/validation sets.

data\_partition <- createDataPartition(y = ds\_lc$is\_acct\_delinquent, p = 0.75, list = FALSE)  
train\_dataset <- ds\_lc[data\_partition,]  
test\_dataset <- ds\_lc[-data\_partition,]

nrow(train\_dataset)

## [1] 370583

nrow(test\_dataset)

## [1] 123527

logistic\_regression <- glm(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec  
 ,data = train\_dataset, family = binomial)  
  
predictions <- predict(logistic\_regression, test\_dataset, type="response")  
predicted\_direction <- ifelse(predictions > 0.5, 1, 0)  
error\_rate\_lr <- mean(predicted\_direction != test\_dataset$is\_acct\_delinquent)  
error\_rate\_lr

## [1] 0.03863123

lda\_model <- lda(is\_acct\_delinquent ~   
 loan\_amnt +   
 term +   
 int\_rate +   
 installment +   
 grade +   
 sub\_grade +   
 home\_ownership +   
 annual\_inc +   
 dti +   
 delinq\_2yrs +   
 fico\_range\_low +   
 fico\_range\_high +   
 inq\_last\_6mths +   
 open\_acc +   
 pub\_rec  
 ,data = train\_dataset)  
predictions <- predict(lda\_model, test\_dataset, type="response")  
confusion\_matrix <- table(predictions$class,   
 test\_dataset$is\_acct\_delinquent,   
 dnn = c("Predicted Status", "Observed Status"))  
confusion\_matrix

## Observed Status  
## Predicted Status 0 1  
## 0 118755 4738  
## 1 29 5

error\_rate\_lda <- mean(predictions$class != test\_dataset$is\_acct\_delinquent)  
error\_rate\_lda

## [1] 0.03859075

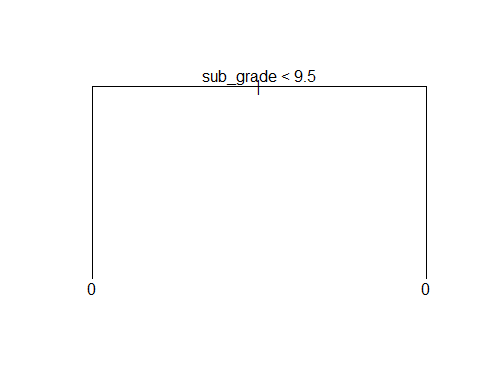
tree\_lc <- tree(is\_acct\_delinquent ~ .,train\_dataset)  
summary(tree\_lc)

##   
## Classification tree:  
## tree(formula = is\_acct\_delinquent ~ ., data = train\_dataset)  
## Variables actually used in tree construction:  
## [1] "sub\_grade"  
## Number of terminal nodes: 2   
## Residual mean deviance: 0.3135 = 116200 / 370600   
## Misclassification error rate: 0.0384 = 14229 / 370583

tree\_lc

## node), split, n, deviance, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 370583 120700 0 ( 0.96160 0.03840 )   
## 2) sub\_grade < 9.5 185587 33140 0 ( 0.98221 0.01779 ) \*  
## 3) sub\_grade > 9.5 184996 83030 0 ( 0.94093 0.05907 ) \*

plot(tree\_lc)  
text(tree\_lc, pretty=0)



prediction <- predict(tree\_lc, test\_dataset, type="class")  
error\_rate\_tree <- mean(prediction != test\_dataset$is\_acct\_delinquent)  
error\_rate\_tree

## [1] 0.03839646

optimal\_tree <- cv.tree(tree\_lc, FUN = prune.tree)  
optimal\_tree

## $size  
## [1] 2 1  
##   
## $dev  
## [1] 116257.6 120674.5  
##   
## $k  
## [1] -Inf 4499.304  
##   
## $method  
## [1] "deviance"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

tree\_plot <- data.frame(x=optimal\_tree$size, y=optimal\_tree$dev)  
ggplot(tree\_plot, aes(x=x,y=y)) +   
 geom\_point() +   
 geom\_line() +   
 xlab("Tree Size") +   
 ylab("Deviance")



pruned\_tree <- prune.tree(tree\_lc, best = 2)

summary(pruned\_tree)

##   
## Classification tree:  
## tree(formula = is\_acct\_delinquent ~ ., data = train\_dataset)  
## Variables actually used in tree construction:  
## [1] "sub\_grade"  
## Number of terminal nodes: 2   
## Residual mean deviance: 0.3135 = 116200 / 370600   
## Misclassification error rate: 0.0384 = 14229 / 370583

summary(tree\_lc)

##   
## Classification tree:  
## tree(formula = is\_acct\_delinquent ~ ., data = train\_dataset)  
## Variables actually used in tree construction:  
## [1] "sub\_grade"  
## Number of terminal nodes: 2   
## Residual mean deviance: 0.3135 = 116200 / 370600   
## Misclassification error rate: 0.0384 = 14229 / 370583

test\_error\_rate\_pruned <- mean(predict(pruned\_tree, test\_dataset, type = "class") != test\_dataset$is\_acct\_delinquent)  
test\_error\_rate\_pruned

## [1] 0.03839646

# Predicting the Interest Rate

test\_interest\_mean <- mean(test\_dataset$int\_rate)  
test\_interest\_mse <- mean((test\_dataset$int\_rate - test\_interest\_mean)^2)  
test\_interest\_mean

## [1] 12.73598

test\_interest\_mse

## [1] 26.48413

# 9b Fit a linear model using least squares on the training set, and report the test error obtained.

lm\_fit <- lm(int\_rate ~ . , data = train\_dataset)  
lm\_predictions <- predict(lm\_fit, test\_dataset)  
lm\_mse <- mean((lm\_predictions - test\_dataset$int\_rate)^2)  
lm\_mse

## [1] 0.375435

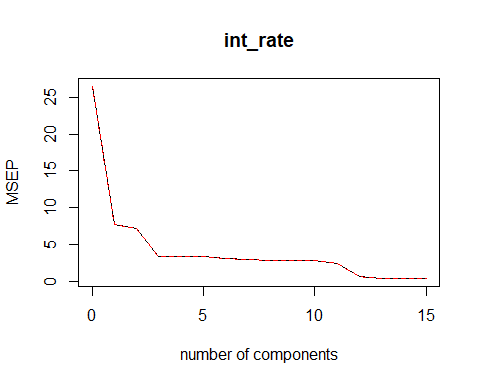
train\_ds\_matrix <- model.matrix(int\_rate ~ ., data = train\_dataset)  
test\_ds\_matrix <- model.matrix(int\_rate ~ ., data = test\_dataset)  
  
grid <- 10 ^ seq(4, -2, length = 100)  
  
ridge\_reg\_model <- cv.glmnet(train\_ds\_matrix, train\_dataset$int\_rate, alpha = 0, lambda = grid, thresh = 1e-12)  
ridge\_reg\_predictions <- predict(ridge\_reg\_model, test\_ds\_matrix, s = ridge\_reg\_model$lambda.min)  
ridge\_mse <- mean((test\_dataset$int\_rate - ridge\_reg\_predictions)^2)  
ridge\_mse

## [1] 0.3765767

lasso\_model <- cv.glmnet(train\_ds\_matrix, train\_dataset$int\_rate, alpha = 1, lambda = grid, thresh = 1e-12)  
lasso\_predictions <- predict(lasso\_model, test\_ds\_matrix, s = lasso\_model$lambda.min)  
lasso\_mse <- mean((test\_dataset$int\_rate - lasso\_predictions)^2)  
lasso\_rmse <- sqrt(mean((test\_dataset$int\_rate - lasso\_predictions)^2))  
lasso\_mse

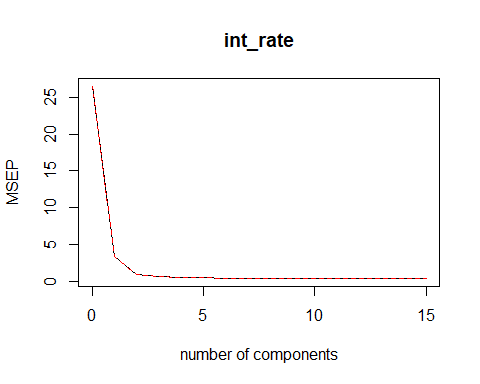
## [1] 0.3873661

pcr\_model <- pcr(int\_rate ~ . , data = train\_dataset, scale=T, validation="CV")  
validationplot(pcr\_model, val.type = "MSEP")



pcr\_predictions <- predict(pcr\_model, test\_dataset, ncomp = 10)  
pcr\_mse <- mean((test\_dataset$int\_rate - pcr\_predictions)^2)  
pcr\_rmse <- sqrt(mean((test\_dataset$int\_rate - pcr\_predictions)^2))

pls\_model <- plsr(int\_rate ~ . , data = train\_dataset, scale=T, validation="CV")  
validationplot(pls\_model, val.type = "MSEP")



pls\_predictions <- predict(pls\_model, test\_dataset, ncomp = 10)  
pls\_mse <- mean((test\_dataset$int\_rate - pls\_predictions)^2)  
pls\_rmse <- sqrt(mean((test\_dataset$int\_rate - pls\_predictions)^2))  
pls\_mse

## [1] 0.3759499

lm\_test\_r2 <- (1 - (lm\_mse/test\_interest\_mse))  
ridge\_test\_r2 <- (1 - (ridge\_mse/test\_interest\_mse))  
lasso\_test\_r2 <- (1 - (lasso\_mse/test\_interest\_mse))  
pcr\_test\_r2 <- (1 - (pcr\_mse/test\_interest\_mse))  
pls\_test\_r2 <- (1 - (pls\_mse/test\_interest\_mse))  
  
cat("R square with linear model : ", lm\_test\_r2, "\n")

## R square with linear model : 0.9858242

cat("R square with ridge model : ", ridge\_test\_r2, "\n")

## R square with ridge model : 0.985781

cat("R square with lasso model : ", lasso\_test\_r2, "\n")

## R square with lasso model : 0.9853737

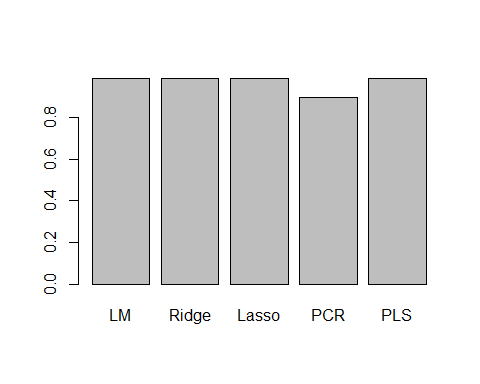
cat("R square with pcr : ", pcr\_test\_r2, "\n")

## R square with pcr : 0.8955828

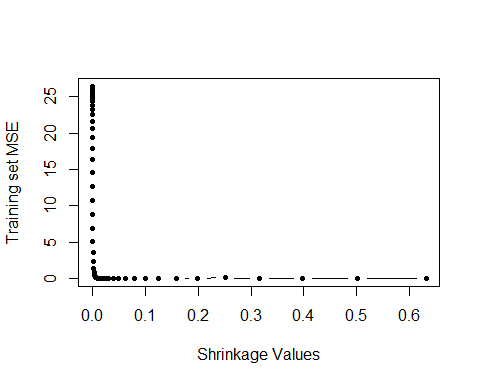
cat("R square with pls : ", pls\_test\_r2, "\n")

## R square with pls : 0.9858047

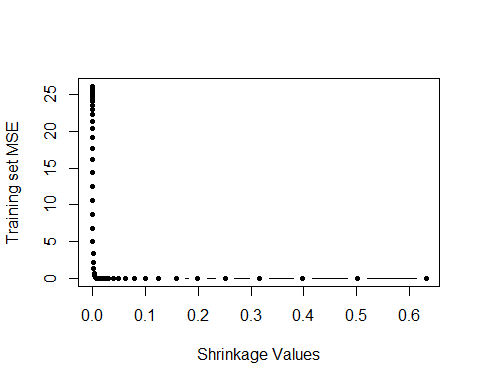
barplot(c(lm\_test\_r2, ridge\_test\_r2, lasso\_test\_r2, pcr\_test\_r2, pls\_test\_r2),  
 names.arg = c("LM", "Ridge", "Lasso", "PCR", "PLS"))



train\_dataset\_mini <- train\_dataset[1:5000,]  
test\_dataset\_mini <- test\_dataset[(1:5000),]  
  
pows <- seq(-10, -0.2, by = 0.1)  
lambdas <- 10^pows  
training\_errors <- rep(NA, length(lambdas))  
  
  
for (i in 1:length(lambdas)) {  
 boosting\_model <- gbm(int\_rate ~ . , data = train\_dataset\_mini, distribution = "gaussian",   
 n.trees = 1000, shrinkage = lambdas[i])  
   
 training\_predictions <- predict(boosting\_model, train\_dataset\_mini, n.trees = 1000)  
 training\_errors[i] <- mean((training\_predictions - train\_dataset\_mini$int\_rate)^2)  
}  
  
plot(lambdas, training\_errors, xlab = "Shrinkage Values", ylab = "Training set MSE", type = "b", pch = 20)



test\_errors <- rep(NA, length(lambdas))  
  
  
for (i in 1:length(lambdas)) {  
 boosting\_model <- gbm(int\_rate ~ . , data = train\_dataset\_mini, distribution = "gaussian",   
 n.trees = 1000, shrinkage = lambdas[i])  
   
 test\_predictions <- predict(boosting\_model, test\_dataset\_mini, n.trees = 1000)  
 test\_errors[i] <- mean((test\_predictions - test\_dataset\_mini$int\_rate)^2)  
}  
  
plot(lambdas, test\_errors, xlab = "Shrinkage Values", ylab = "Training set MSE", type = "b", pch = 20)



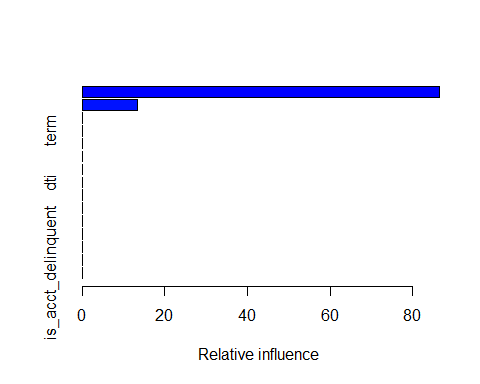
min(test\_errors)

## [1] 0.001951323

min\_test\_err <- min(test\_errors)  
min\_test\_err\_at <- lambdas[which.min(test\_errors)]

Both regression approaches lm and lasso have higher MSE compared to that of boosting.

boosting\_model <- gbm(int\_rate ~ . , data = train\_dataset\_mini, distribution = "gaussian",   
 n.trees = 1000, shrinkage = min\_test\_err)  
summary(boosting\_model)



## var rel.inf  
## sub\_grade sub\_grade 86.56652  
## grade grade 13.43348  
## loan\_amnt loan\_amnt 0.00000  
## term term 0.00000  
## installment installment 0.00000  
## home\_ownership home\_ownership 0.00000  
## annual\_inc annual\_inc 0.00000  
## dti dti 0.00000  
## delinq\_2yrs delinq\_2yrs 0.00000  
## fico\_range\_low fico\_range\_low 0.00000  
## fico\_range\_high fico\_range\_high 0.00000  
## inq\_last\_6mths inq\_last\_6mths 0.00000  
## open\_acc open\_acc 0.00000  
## pub\_rec pub\_rec 0.00000  
## is\_acct\_delinquent is\_acct\_delinquent 0.00000

random\_forest\_model <- randomForest(int\_rate ~ . , data = train\_dataset\_mini, ntree = 500, mtry = ncol(train\_dataset\_mini)-1)  
random\_forest\_predictions <- predict(random\_forest\_model, test\_dataset\_mini)  
  
random\_forest\_test\_mse <- mean((random\_forest\_predictions - test\_dataset\_mini$int\_rate)^2)  
random\_forest\_test\_mse

## [1] 0.03471787

Test MSE for bagging is 0.0347179 which is better than 0.0019513 which is best MSE from boosting

# Unspervised Learning - PCA

ds\_lc\_pca <- ds\_lc[,c(  
 "loan\_amnt",  
 "term",  
 "int\_rate",  
 "installment",  
 "grade",  
 "sub\_grade",  
 "home\_ownership",  
 "annual\_inc",  
 "dti",  
 "delinq\_2yrs",  
 "fico\_range\_low",  
 "fico\_range\_high",  
 "inq\_last\_6mths",  
 "open\_acc",  
 "pub\_rec"  
 )]  
ds\_lc\_pca\_subset <- sample(nrow(ds\_lc), 1000)  
ds\_lc\_pca = ds\_lc\_pca[ds\_lc\_pca\_subset, ]  
  
str(ds\_lc\_pca)

## Classes 'data.table' and 'data.frame': 1000 obs. of 15 variables:  
## $ loan\_amnt : int 15000 6000 1000 25000 25000 14000 9000 20000 36000 18000 ...  
## $ term : int 1 1 1 2 1 2 1 2 1 2 ...  
## $ int\_rate : num 10.9 16.14 18.94 7.21 14.47 ...  
## $ installment : num 490.4 211.4 36.6 497.5 860.2 ...  
## $ grade : int 2 3 4 1 3 2 1 4 1 3 ...  
## $ sub\_grade : int 9 14 17 3 12 6 4 17 1 14 ...  
## $ home\_ownership : int 2 4 2 2 2 3 2 3 2 4 ...  
## $ annual\_inc : num 44232 25920 90000 85000 120000 ...  
## $ dti : num 3.18 25.69 17.67 30.37 29.85 ...  
## $ delinq\_2yrs : int 1 0 0 0 0 2 0 0 0 0 ...  
## $ fico\_range\_low : int 710 670 680 730 685 700 775 685 730 710 ...  
## $ fico\_range\_high: int 714 674 684 734 689 704 779 689 734 714 ...  
## $ inq\_last\_6mths : int 3 0 0 0 1 0 0 0 0 0 ...  
## $ open\_acc : int 6 7 7 15 18 9 9 10 16 9 ...  
## $ pub\_rec : int 0 0 0 0 0 0 0 0 0 0 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

pr.out <- prcomp (ds\_lc\_pca, scale =TRUE)  
names(pr.out)

## [1] "sdev" "rotation" "center" "scale" "x"

pr.out$center

## loan\_amnt term int\_rate installment grade   
## 15317.27500 1.28000 12.81135 448.54785 2.42500   
## sub\_grade home\_ownership annual\_inc dti delinq\_2yrs   
## 10.11300 2.93000 75935.35083 18.90746 0.22900   
## fico\_range\_low fico\_range\_high inq\_last\_6mths open\_acc pub\_rec   
## 706.13000 710.13000 0.42200 11.46400 0.12600

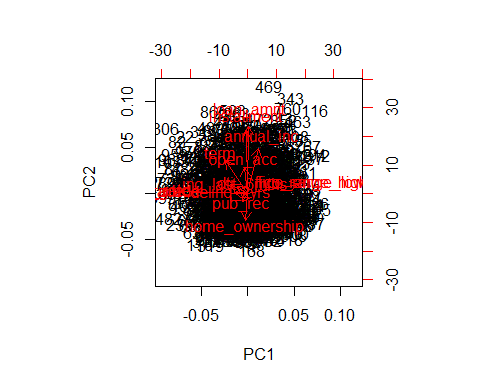
pr.out$scale

## loan\_amnt term int\_rate installment grade   
## 1.028473e+04 4.492236e-01 5.299601e+00 2.912727e+02 1.222239e+00   
## sub\_grade home\_ownership annual\_inc dti delinq\_2yrs   
## 6.148661e+00 9.380725e-01 4.491856e+04 1.113957e+01 6.595407e-01   
## fico\_range\_low fico\_range\_high inq\_last\_6mths open\_acc pub\_rec   
## 3.572253e+01 3.572253e+01 6.843861e-01 6.006395e+00 3.553171e-01

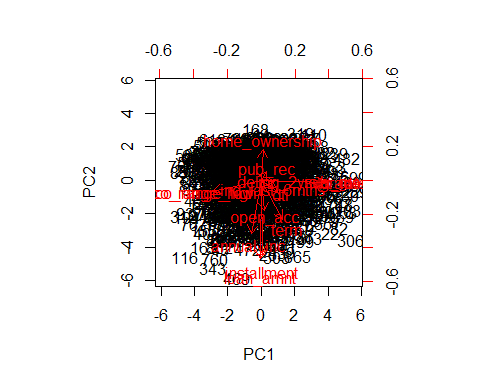
pr.out$rotation

## PC1 PC2 PC3 PC4 PC5  
## loan\_amnt 0.006605629 0.58003110 -0.0545297304 0.19285218 -0.17585157  
## term -0.155732755 0.28974736 -0.2799346661 -0.09668434 0.08062965  
## int\_rate -0.475999649 0.01935206 -0.2332918130 -0.05129638 0.08532478  
## installment -0.007636083 0.54179639 -0.0009614438 0.23145002 -0.21746121  
## grade -0.473040209 0.01853215 -0.2293123788 -0.04570269 0.10073042  
## sub\_grade -0.480611622 0.01490580 -0.2202432839 -0.04987475 0.07827835  
## home\_ownership -0.013758165 -0.22918545 -0.2056042435 0.29086456 -0.27620951  
## annual\_inc 0.074692455 0.39291474 0.1863401313 0.18437788 0.31447475  
## dti -0.121580547 0.08220175 0.0711923377 -0.54214809 -0.56065336  
## delinq\_2yrs -0.096167167 0.01056164 0.2171211547 0.05118269 0.34488031  
## fico\_range\_low 0.358588475 0.07902587 -0.4480652494 -0.26701115 0.14604703  
## fico\_range\_high 0.358588475 0.07902587 -0.4480652494 -0.26701115 0.14604703  
## inq\_last\_6mths -0.072326118 0.06089252 0.2110780589 -0.30660043 0.47624057  
## open\_acc -0.025148507 0.22578832 0.3247671860 -0.48889421 -0.05120646  
## pub\_rec -0.038497021 -0.06458991 0.2870322696 -0.04393885 -0.10575414  
## PC6 PC7 PC8 PC9 PC10  
## loan\_amnt -0.02681484 0.04910305 -0.13782456 -0.16600454 -0.054920334  
## term -0.12512666 -0.28299552 0.36043680 -0.23129187 -0.679649223  
## int\_rate -0.02335245 0.02877098 -0.07533751 0.15515159 0.121206140  
## installment 0.01047814 0.14762192 -0.28210635 -0.09640162 0.161646206  
## grade -0.03177112 0.01757674 -0.07646625 0.14673702 0.120994084  
## sub\_grade -0.02729377 0.03061077 -0.07711553 0.13243359 0.116339940  
## home\_ownership 0.09887902 0.66261640 -0.09859539 0.19149113 -0.466589440  
## annual\_inc -0.04221689 0.07621762 0.21908281 0.46378985 0.128893259  
## dti 0.17503734 -0.01961296 -0.11787268 -0.15206793 0.100116146  
## delinq\_2yrs 0.61988540 -0.21970311 -0.53213982 -0.01891549 -0.313764734  
## fico\_range\_low -0.01769485 0.01502545 -0.22120987 0.14720230 0.022387567  
## fico\_range\_high -0.01769485 0.01502545 -0.22120987 0.14720230 0.022387567  
## inq\_last\_6mths -0.24724173 0.55611776 -0.10469986 -0.48847374 -0.006536942  
## open\_acc 0.17649080 0.21304469 0.21441021 0.46838104 -0.215062739  
## pub\_rec -0.67959974 -0.21214663 -0.49441843 0.26445059 -0.275640831  
## PC11 PC12 PC13 PC14  
## loan\_amnt 0.123752471 0.0917245308 0.654332720 -0.2963986742  
## term -0.114604821 -0.0285033906 -0.188783628 0.0831779240  
## int\_rate 0.021643040 -0.4221955610 -0.208776069 -0.6637743399  
## installment 0.197742079 -0.0784016937 -0.594647488 0.2641510613  
## grade 0.026584534 0.8131419932 -0.075505731 0.0112023465  
## sub\_grade 0.029656772 -0.3804724298 0.364857832 0.6278580626  
## home\_ownership -0.176255642 0.0108109927 0.001209976 0.0004513594  
## annual\_inc -0.620190555 -0.0079710725 -0.009481488 0.0182728933  
## dti -0.535634769 0.0123576590 -0.003686119 -0.0006688116  
## delinq\_2yrs -0.086122246 -0.0027414897 0.002166760 -0.0010575670  
## fico\_range\_low -0.006813201 -0.0039521047 0.003261541 0.0081845348  
## fico\_range\_high -0.006813201 -0.0039521047 0.003261541 0.0081845348  
## inq\_last\_6mths -0.074969889 -0.0004538554 -0.004631504 -0.0070156304  
## open\_acc 0.461577477 0.0061156644 0.006402930 -0.0002815421  
## pub\_rec -0.040041654 -0.0030967974 0.000792840 -0.0026310452  
## PC15  
## loan\_amnt 8.513487e-16  
## term 1.468208e-16  
## int\_rate 2.744999e-15  
## installment -6.922222e-16  
## grade 3.038121e-16  
## sub\_grade -4.094823e-15  
## home\_ownership -7.462468e-17  
## annual\_inc -2.392466e-16  
## dti -9.443205e-17  
## delinq\_2yrs -3.759133e-16  
## fico\_range\_low 7.071068e-01  
## fico\_range\_high -7.071068e-01  
## inq\_last\_6mths -5.728263e-19  
## open\_acc -1.061237e-16  
## pub\_rec -1.493534e-16

biplot(pr.out)



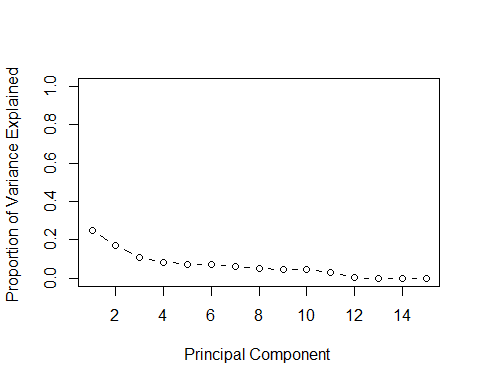
pr.out$rotation=-pr.out$rotation  
pr.out$x=-pr.out$x  
biplot (pr.out , scale =0)



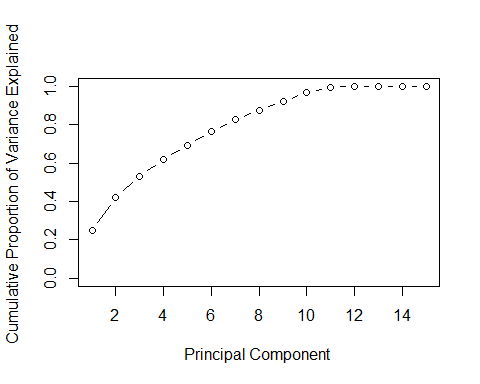
#pr.out$sdev  
pr.var =pr.out$sdev ^2  
pve=pr.var/sum(pr.var )  
pve

## [1] 2.518557e-01 1.708062e-01 1.096674e-01 8.472282e-02 7.482386e-02  
## [6] 7.283161e-02 6.035401e-02 5.103440e-02 4.632298e-02 4.514161e-02  
## [11] 2.901450e-02 2.229948e-03 7.383321e-04 4.564958e-04 3.026850e-31

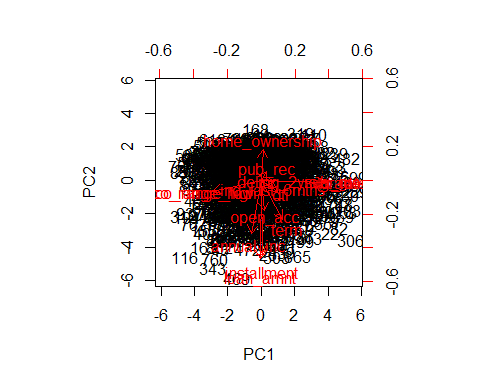
plot(pve , xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1) ,type="b")



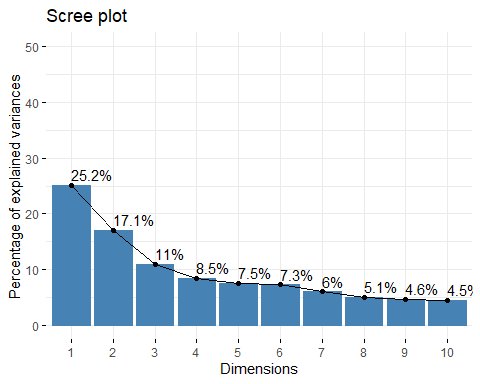
plot(cumsum (pve ), xlab="Principal Component ", ylab ="Cumulative Proportion of Variance Explained ", ylim=c(0,1) , type="b")



#autoplot(pr.out)  
biplot (pr.out , scale =0)



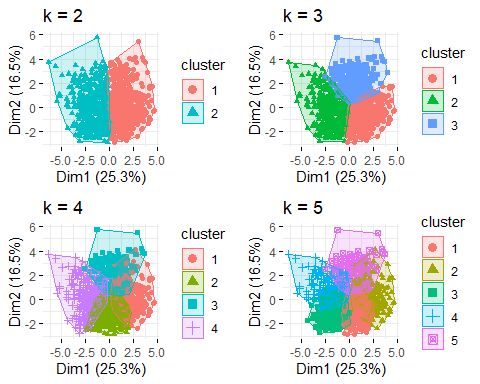
fviz\_eig(pr.out, addlabels = TRUE, ylim = c(0, 50))



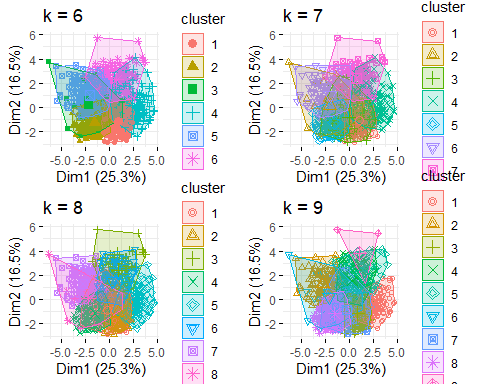
PCA does not bring in much value. Very dimension were identified with PCA.

ds\_lc\_kmean <- ds\_lc[,c(  
 "loan\_amnt",  
 "term",  
 "int\_rate",  
 "installment",  
 "grade",  
 "sub\_grade",  
 "home\_ownership",  
 "annual\_inc",  
 "dti",  
 "delinq\_2yrs",  
 "fico\_range\_low",  
 "fico\_range\_high",  
 "inq\_last\_6mths",  
 "open\_acc",  
 "pub\_rec"  
 )]  
ds\_lc\_kmean\_subset <- sample(nrow(ds\_lc\_kmean), 1000)  
ds\_lc\_kmean = ds\_lc\_kmean[ds\_lc\_kmean\_subset, ]  
ds\_lc\_kmean\_scaled <- as.data.frame(scale(ds\_lc\_kmean))

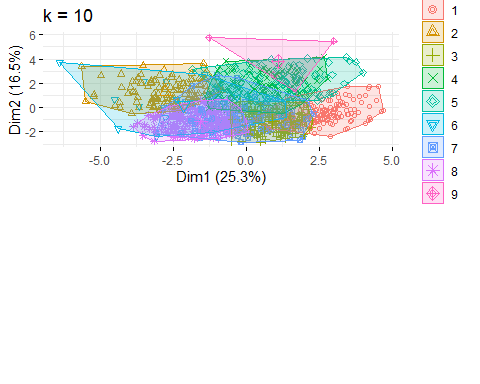
k2 <- kmeans(ds\_lc\_kmean\_scaled, centers = 2, nstart = 25)  
k3 <- kmeans(ds\_lc\_kmean\_scaled, centers = 3, nstart = 25)  
k4 <- kmeans(ds\_lc\_kmean\_scaled, centers = 4, nstart = 25)  
k5 <- kmeans(ds\_lc\_kmean\_scaled, centers = 5, nstart = 25)  
k6 <- kmeans(ds\_lc\_kmean\_scaled, centers = 6, nstart = 25)  
k7 <- kmeans(ds\_lc\_kmean\_scaled, centers = 7, nstart = 25)  
k8 <- kmeans(ds\_lc\_kmean\_scaled, centers = 8, nstart = 25)  
k9 <- kmeans(ds\_lc\_kmean\_scaled, centers = 9, nstart = 25)  
  
  
# plots to compare  
p2 <- fviz\_cluster(k2, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 2")  
p3 <- fviz\_cluster(k3, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 3")  
p4 <- fviz\_cluster(k4, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 4")  
p5 <- fviz\_cluster(k5, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 5")  
p6 <- fviz\_cluster(k6, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 6")  
p7 <- fviz\_cluster(k7, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 7")  
p8 <- fviz\_cluster(k8, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 8")  
p9 <- fviz\_cluster(k9, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 9")  
p10 <- fviz\_cluster(k9, geom = "point", data = ds\_lc\_kmean\_scaled, ggtheme = theme\_minimal()) + ggtitle("k = 10")  
  
  
grid.arrange(p2, p3, p4, p5, nrow = 2)



grid.arrange(p6, p7, p8, p9, nrow = 2)



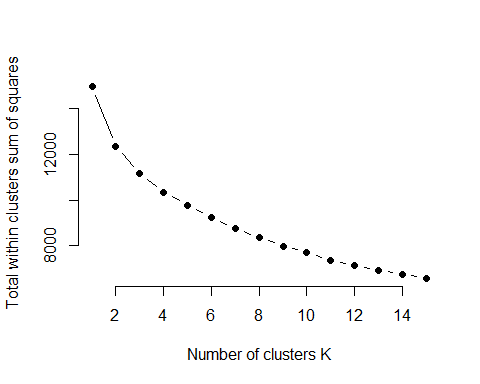
grid.arrange(p10, nrow = 2)



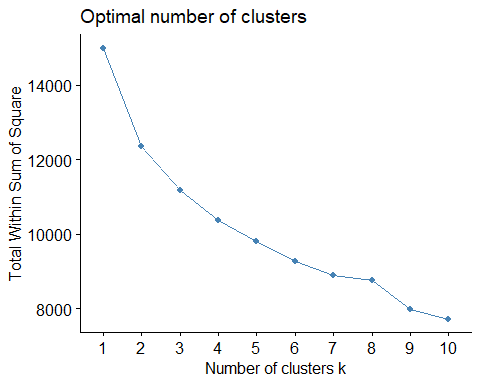
set.seed(1234)  
#Function to compute total within cluster sum of squares   
wss <- function(k) {  
 kmeans(ds\_lc\_kmean\_scaled, k, nstart = 25, iter.max = 10)$tot.withinss  
}  
  
#Compute and plot the within sum of squares (wss) for k = 1 to k = 10  
k.values <- 1:15  
  
#Extract wss for 2 - 10 clusters  
wss\_values <- map\_dbl(k.values, wss)

## Warning: did not converge in 10 iterations  
  
## Warning: did not converge in 10 iterations  
  
## Warning: did not converge in 10 iterations

plot(k.values, wss\_values,  
 type = "b", pch = 19, frame = FALSE,  
 xlab = "Number of clusters K",  
 ylab = "Total within clusters sum of squares")



set.seed(1234)  
fviz\_nbclust(ds\_lc\_kmean\_scaled, kmeans, method = "wss")



Based on the visualization 3 or 4 clusters seems to be better classification. 3 clusters have very less overlap compared to that of the 4 clusters.

ds\_lc\_kmean$cluster <- k3$cluster

ds\_lc\_kmean$category <- ""  
for(i in 1:nrow(ds\_lc\_kmean)){  
   
 if (ds\_lc\_kmean$cluster[i] == 1){  
 ds\_lc\_kmean$category[i] <- "Low"  
 }  
 else if (ds\_lc\_kmean$cluster[i] == 2){  
 ds\_lc\_kmean$category[i] <- "Medium"  
 }  
 else if (ds\_lc\_kmean$cluster[i] == 3){  
 ds\_lc\_kmean$category[i] <- "High"  
 }  
}  
  
low\_risk\_loans <- ds\_lc\_kmean %>%  
 filter(category == "Low")  
  
medium\_risk\_loans <- ds\_lc\_kmean %>%  
 filter(category == "Medium")  
  
high\_risk\_loans <- ds\_lc\_kmean %>%  
 filter(category == "High")

mean(low\_risk\_loans$int\_rate)

## [1] 8.76289

mean(medium\_risk\_loans$int\_rate)

## [1] 17.97925

mean(high\_risk\_loans$int\_rate)

## [1] 11.51484

mean(low\_risk\_loans$annual\_inc)

## [1] 67194.69

mean(medium\_risk\_loans$annual\_inc)

## [1] 66523.53

mean(high\_risk\_loans$annual\_inc)

## [1] 128719.9

mean(low\_risk\_loans$dti)

## [1] 16.79985

mean(medium\_risk\_loans$dti)

## [1] 20.90247

mean(high\_risk\_loans$dti)

## [1] 19.32208

mean(low\_risk\_loans$fico\_range\_low)

## [1] 719.5908

mean(medium\_risk\_loans$fico\_range\_low)

## [1] 683.2732

mean(high\_risk\_loans$fico\_range\_low)

## [1] 716.6063

mean(low\_risk\_loans$fico\_range\_high)

## [1] 723.5934

mean(medium\_risk\_loans$fico\_range\_high)

## [1] 687.2732

mean(high\_risk\_loans$fico\_range\_high)

## [1] 720.6063

mean(low\_risk\_loans$inq\_last\_6mths)

## [1] 0.3554987

mean(medium\_risk\_loans$inq\_last\_6mths)

## [1] 0.5386598

mean(high\_risk\_loans$inq\_last\_6mths)

## [1] 0.4660633

mean(low\_risk\_loans$delinq\_2yrs)

## [1] 0.1227621

mean(medium\_risk\_loans$delinq\_2yrs)

## [1] 0.435567

mean(high\_risk\_loans$delinq\_2yrs)

## [1] 0.1809955

mean(low\_risk\_loans$grade)

## [1] 1.478261

mean(medium\_risk\_loans$grade)

## [1] 3.618557

mean(high\_risk\_loans$grade)

## [1] 2.167421

mean(low\_risk\_loans$sub\_grade)

## [1] 5.199488

mean(medium\_risk\_loans$sub\_grade)

## [1] 16.15206

mean(high\_risk\_loans$sub\_grade)

## [1] 8.769231