Capstone Project - Week1-8

Amol Gote

12/18/2020

set.seed(123)  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 1.0.0  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(corrplot)

## corrplot 0.84 loaded

library(tidyverse)  
library(ROCR)  
library(gbm)

## Loaded gbm 2.1.8

library(ROSE)

## Loaded ROSE 0.0-3

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(parallel)   
library(tree)

## Registered S3 method overwritten by 'tree':  
## method from  
## print.tree cli

library(rpart)  
library(rpart.plot)  
library(ResourceSelection)

## ResourceSelection 0.3-5 2019-07-22

library(generalhoslem)

## Loading required package: reshape

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

## The following objects are masked from 'package:tidyr':  
##   
## expand, smiths

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

fundamentals\_ds <- read.csv("./data/Fundamentals\_DS.csv", na.strings=c(""," "))  
nrow(fundamentals\_ds)

## [1] 17416

### Filter the dataset with the sector assigned.

fundamentals\_ds\_sjm <- fundamentals\_ds %>%  
 filter(tic == 'SJM')  
gsector\_sjm <- head(fundamentals\_ds\_sjm$gsector, 1)  
fundamentals\_ds <- fundamentals\_ds %>%  
 filter(gsector == gsector\_sjm)  
nrow(fundamentals\_ds)

## [1] 2323

gsector\_sjm

## [1] 30

### Filter the data to restrict the dataset to contain all annually reported statements, and exclude restatements.

names(fundamentals\_ds)[names(fundamentals\_ds) == "ï..gvkey"] <- "gvkey"  
fundamentals\_ds\_filter <- fundamentals\_ds %>%  
 filter(datafmt == 'STD')   
nrow(fundamentals\_ds\_filter)

## [1] 1243

target\_company <- fundamentals\_ds\_filter %>%  
 filter(tic == 'SJM')  
target\_company\_gv\_key <- head(target\_company$gvkey, 1)  
target\_company\_gv\_key

## [1] 9777

ncol(fundamentals\_ds\_filter)

## [1] 1768

#summary(fundamentals\_ds\_filter)

summary(fundamentals\_ds\_filter$mkvalt)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.03 29.65 261.19 7270.94 2406.22 241440.44 284

### Remove all the columns which contain 25% or greater NA or null values

fundamentals\_ds\_filter\_1 <- fundamentals\_ds\_filter[ lapply( fundamentals\_ds\_filter, function(x) sum(is.na(x)) / length(x) ) < 0.25 ]  
ncol(fundamentals\_ds\_filter\_1)

## [1] 318

nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

colnames(fundamentals\_ds\_filter\_1)

## [1] "gvkey" "datadate" "fyear" "indfmt" "consol"   
## [6] "popsrc" "datafmt" "tic" "conm" "acctstd"   
## [11] "ajex" "ajp" "curcd" "curncd" "currtr"   
## [16] "final" "fyr" "ismod" "pddur" "scf"   
## [21] "src" "upd" "apdedate" "fdate" "acchg"   
## [26] "acdo" "aco" "acodo" "acominc" "acox"   
## [31] "act" "aldo" "am" "ao" "aocidergl"   
## [36] "aociother" "aocipen" "aocisecgl" "aodo" "aoloch"   
## [41] "aox" "ap" "aqc" "at" "bkvlps"   
## [46] "caps" "capx" "capxv" "ceq" "ceql"   
## [51] "ceqt" "ch" "che" "chech" "ci"   
## [56] "cibegni" "cicurr" "cidergl" "cimii" "ciother"   
## [61] "cipen" "cisecgl" "citotal" "cogs" "cshfd"   
## [66] "cshi" "csho" "cshpri" "cshr" "cstk"   
## [71] "cstkcv" "cstke" "dc" "dclo" "dcom"   
## [76] "dcpstk" "dcvsr" "dcvsub" "dcvt" "dd"   
## [81] "dd1" "dd2" "dd3" "dd4" "dd5"   
## [86] "diladj" "dilavx" "dlc" "dltis" "dlto"   
## [91] "dltp" "dltr" "dltt" "dm" "dn"   
## [96] "do" "donr" "dp" "dpact" "dpc"   
## [101] "dpvieb" "drc" "drlt" "ds" "dudd"   
## [106] "dv" "dvc" "dvp" "dvpa" "dvt"   
## [111] "ebit" "ebitda" "emp" "epsfi" "epsfx"   
## [116] "epspi" "epspx" "esopct" "esopnr" "esopr"   
## [121] "esopt" "esub" "esubc" "exre" "fatb"   
## [126] "fatc" "fatn" "fatp" "fiao" "fincf"   
## [131] "fopo" "fopox" "gdwl" "gp" "ib"   
## [136] "ibadj" "ibc" "ibcom" "ibmii" "icapt"   
## [141] "intan" "intano" "intc" "intpn" "invch"   
## [146] "invt" "itcb" "ivaco" "ivaeq" "ivao"   
## [151] "ivch" "ivncf" "ivst" "ivstch" "lco"   
## [156] "lcox" "lcoxdr" "lct" "lifr" "lo"   
## [161] "loxdr" "lse" "lt" "mib" "mibn"   
## [166] "mibt" "mii" "mrc1" "mrcta" "msa"   
## [171] "ni" "niadj" "nopi" "nopio" "np"   
## [176] "oancf" "oiadp" "oibdp" "opeps" "oprepsx"   
## [181] "pi" "pnca" "pncad" "pncaeps" "pnrsho"   
## [186] "ppegt" "ppent" "ppeveb" "prca" "prcad"   
## [191] "prcaeps" "prsho" "prstkc" "pstk" "pstkc"   
## [196] "pstkl" "pstkn" "pstkr" "pstkrv" "rdip"   
## [201] "rdipa" "rdipd" "rdipeps" "re" "rea"   
## [206] "reajo" "recch" "recco" "recd" "rect"   
## [211] "recta" "rectr" "reuna" "revt" "sale"   
## [216] "seq" "seqo" "siv" "spce" "spced"   
## [221] "spceeps" "spi" "sppe" "sppiv" "sstk"   
## [226] "stkco" "teq" "tstk" "tstkc" "tstkn"   
## [231] "tstkp" "txbco" "txbcof" "txc" "txdb"   
## [236] "txdba" "txdbca" "txdbcl" "txdc" "txdi"   
## [241] "txditc" "txfed" "txfo" "txndb" "txndba"   
## [246] "txndbl" "txndbr" "txo" "txp" "txpd"   
## [251] "txr" "txs" "txt" "txw" "wcap"   
## [256] "xacc" "xi" "xido" "xidoc" "xint"   
## [261] "xintopt" "xopr" "xoptd" "xopteps" "xrent"   
## [266] "xsga" "exchg" "costat" "fic" "naicsh"   
## [271] "sich" "cshtr\_c" "dvpsp\_c" "dvpsx\_c" "prcc\_c"   
## [276] "prch\_c" "prcl\_c" "adjex\_c" "cshtr\_f" "dvpsp\_f"   
## [281] "dvpsx\_f" "mkvalt" "prcc\_f" "prch\_f" "prcl\_f"   
## [286] "adjex\_f" "rank" "au" "auop" "auopic"   
## [291] "ceoso" "cfoso" "dpact\_fn" "rdipa\_fn" "rdipd\_fn"   
## [296] "rdipeps\_fn" "stkco\_fn" "add1" "addzip" "busdesc"   
## [301] "city" "conml" "ein" "fyrc" "ggroup"   
## [306] "gind" "gsector" "gsubind" "idbflag" "incorp"   
## [311] "loc" "naics" "phone" "priusa" "sic"   
## [316] "state" "stko" "weburl"

write.csv(fundamentals\_ds\_filter\_1, file = "data/fundamentals\_ds\_filter\_1.csv", row.names=FALSE)

### Remove all the columns which are related to general information of the company like address, phone, url etc…

#fundamentals\_ds\_filter\_1 <- fundamentals\_ds\_filter\_1[ lapply( fundamentals\_ds\_filter\_1, function(x) sum(is.na(x)) / length(x) ) < 0.25 ]  
ncol(fundamentals\_ds\_filter\_1)

## [1] 318

fundamentals\_ds\_filter\_1 <- subset(fundamentals\_ds\_filter\_1, select = -c(datadate,indfmt,curncd,consol,popsrc,conm,curcd,apdedate,fdate,add1,addzip,busdesc,  
 city,conml,weburl,phone,loc,final,fyr,acchg,fic,  
 xido,xidoc,naicsh,sich,au,auop,auopic,fyrc,ggroup,gind,gsector,gsubind,priusa))  
write.csv(fundamentals\_ds\_filter\_1, file = "data/fundamentals\_ds\_filter\_2.csv", row.names=FALSE, na="")  
ncol(fundamentals\_ds\_filter\_1)

## [1] 284

nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

### Identify columns which have values which are new to zero variance

nzv\_ds <- nearZeroVar(fundamentals\_ds\_filter\_1, saveMetrics = TRUE)  
nzv\_ds <- nzv\_ds[nzv\_ds[,"nzv"] > 0, ]  
nzv\_ds

## freqRatio percentUnique zeroVar nzv  
## datafmt 0.00000 0.08045052 TRUE TRUE  
## ajex 50.59091 3.05711987 FALSE TRUE  
## ajp 50.59091 3.05711987 FALSE TRUE  
## currtr 121.77778 5.79243765 FALSE TRUE  
## ismod 45.48000 0.16090105 FALSE TRUE  
## pddur 1239.00000 0.40225261 FALSE TRUE  
## scf 0.00000 0.08045052 TRUE TRUE  
## upd 94.61538 0.16090105 FALSE TRUE  
## acdo 1161.00000 3.13757039 FALSE TRUE  
## aldo 1187.00000 1.04585680 FALSE TRUE  
## aocisecgl 0.00000 0.08045052 TRUE TRUE  
## ciother 217.20000 7.88415125 FALSE TRUE  
## cstke 272.75000 7.88415125 FALSE TRUE  
## dcom 1177.00000 1.76991150 FALSE TRUE  
## dcvsr 276.75000 5.87288817 FALSE TRUE  
## dcvsub 291.75000 1.44810941 FALSE TRUE  
## dcvt 217.60000 6.99919549 FALSE TRUE  
## diladj 220.60000 4.82703138 FALSE TRUE  
## donr 538.50000 5.47063556 FALSE TRUE  
## drlt 561.00000 5.06838294 FALSE TRUE  
## ds 284.00000 2.73531778 FALSE TRUE  
## dudd 374.33333 5.14883347 FALSE TRUE  
## dvp 223.60000 5.06838294 FALSE TRUE  
## dvpa 1176.00000 1.20675784 FALSE TRUE  
## esopct 1168.00000 2.33306516 FALSE TRUE  
## esopnr 598.50000 0.64360418 FALSE TRUE  
## esopr 0.00000 0.08045052 TRUE TRUE  
## esopt 598.50000 0.64360418 FALSE TRUE  
## fatn 254.00000 1.20675784 FALSE TRUE  
## itcb 0.00000 0.08045052 TRUE TRUE  
## mib 383.66667 4.02252615 FALSE TRUE  
## pnrsho 208.60000 7.16009654 FALSE TRUE  
## prcad 21.23404 2.01126307 FALSE TRUE  
## prcaeps 20.72917 2.01126307 FALSE TRUE  
## prsho 393.33333 1.28720837 FALSE TRUE  
## pstk 93.90909 7.96460177 FALSE TRUE  
## pstkc 135.87500 5.55108608 FALSE TRUE  
## pstkl 92.63636 8.68865648 FALSE TRUE  
## pstkn 96.09091 6.19469027 FALSE TRUE  
## pstkr 393.33333 1.93081255 FALSE TRUE  
## pstkrv 92.63636 8.68865648 FALSE TRUE  
## rdip 1202.00000 0.16090105 FALSE TRUE  
## rdipa 1194.00000 0.16090105 FALSE TRUE  
## rdipd 1157.00000 0.16090105 FALSE TRUE  
## rdipeps 1157.00000 0.16090105 FALSE TRUE  
## rea 957.00000 5.06838294 FALSE TRUE  
## tstkp 297.00000 0.16090105 FALSE TRUE  
## txndbr 1120.00000 0.16090105 FALSE TRUE  
## txo 1106.00000 3.29847144 FALSE TRUE  
## txw 1026.00000 8.20595334 FALSE TRUE  
## xi 1202.00000 0.16090105 FALSE TRUE  
## xintopt 1025.00000 0.16090105 FALSE TRUE  
## xoptd 0.00000 0.08045052 TRUE TRUE  
## xopteps 0.00000 0.08045052 TRUE TRUE  
## adjex\_c 50.28571 2.97666935 FALSE TRUE  
## adjex\_f 50.38095 2.97666935 FALSE TRUE  
## rank 0.00000 0.08045052 TRUE TRUE  
## dpact\_fn 23.02174 0.32180209 FALSE TRUE  
## rdipa\_fn 0.00000 0.08045052 TRUE TRUE  
## rdipd\_fn 285.75000 0.16090105 FALSE TRUE  
## rdipeps\_fn 228.40000 0.16090105 FALSE TRUE  
## stkco\_fn 0.00000 0.08045052 TRUE TRUE

### Remove columns which have values which are of zero variance

nzv\_ds\_cols <- nearZeroVar(fundamentals\_ds\_filter\_1)  
fundamentals\_ds\_filter\_1 <- fundamentals\_ds\_filter\_1[, -nzv\_ds\_cols]  
write.csv(fundamentals\_ds\_filter\_1, file = "data/fundamentals\_ds\_filter\_3.csv", row.names=FALSE, na="")

ncol(fundamentals\_ds\_filter\_1)

## [1] 222

nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

### Based on manual analysis of the columns and considering the end goal, remove the variables which logically do not make sense or might have correlattion with existing set of variables.

fundamentals\_ds\_filter\_1 <- subset(fundamentals\_ds\_filter\_1, select = -c(acctstd,src,acodo,acox,  
 aox,capxv,ceql,cibegni,cicurr,cidergl,  
 cimii,cipen,cisecgl,citotal,cshfd,cshpri,dclo,dcpstk,  
 dltis,dlto,dltr,do,dp,dpvieb,drc,  
 dv,dvc,epsfx,epspx,exre,  
 fopox,ibadj,ibc,ibcom,ibmii,  
 invch,ivaco,ivao,lco,  
 lcox,lcoxdr,lct,loxdr,mibn,  
 mibt,mii,msa,niadj,np,oprepsx,pnca,ppent,ppeveb,  
 recco,rectr,sale,spced,spceeps,  
 tstkc,txbco,txbcof,txdb,  
 txdbca,txdi,  
 txditc,txndb,xopr,exchg,costat,  
 ceoso,cfoso,idbflag,naics,sic,stko))  
write.csv(fundamentals\_ds\_filter\_1, file = "data/fundamentals\_ds\_filter\_4.csv", row.names=FALSE, na="")

colnames(fundamentals\_ds\_filter\_1)

## [1] "gvkey" "fyear" "tic" "aco" "acominc" "act"   
## [7] "am" "ao" "aocidergl" "aociother" "aocipen" "aodo"   
## [13] "aoloch" "ap" "aqc" "at" "bkvlps" "caps"   
## [19] "capx" "ceq" "ceqt" "ch" "che" "chech"   
## [25] "ci" "cogs" "cshi" "csho" "cshr" "cstk"   
## [31] "cstkcv" "dc" "dd" "dd1" "dd2" "dd3"   
## [37] "dd4" "dd5" "dilavx" "dlc" "dltp" "dltt"   
## [43] "dm" "dn" "dpact" "dpc" "dvt" "ebit"   
## [49] "ebitda" "emp" "epsfi" "epspi" "esub" "esubc"   
## [55] "fatb" "fatc" "fatp" "fiao" "fincf" "fopo"   
## [61] "gdwl" "gp" "ib" "icapt" "intan" "intano"   
## [67] "intc" "intpn" "invt" "ivaeq" "ivch" "ivncf"   
## [73] "ivst" "ivstch" "lifr" "lo" "lse" "lt"   
## [79] "mrc1" "mrcta" "ni" "nopi" "nopio" "oancf"   
## [85] "oiadp" "oibdp" "opeps" "pi" "pncad" "pncaeps"   
## [91] "ppegt" "prca" "prstkc" "re" "reajo" "recch"   
## [97] "recd" "rect" "recta" "reuna" "revt" "seq"   
## [103] "seqo" "siv" "spce" "spi" "sppe" "sppiv"   
## [109] "sstk" "stkco" "teq" "tstk" "tstkn" "txc"   
## [115] "txdba" "txdbcl" "txdc" "txfed" "txfo" "txndba"   
## [121] "txndbl" "txp" "txpd" "txr" "txs" "txt"   
## [127] "wcap" "xacc" "xint" "xrent" "xsga" "cshtr\_c"   
## [133] "dvpsp\_c" "dvpsx\_c" "prcc\_c" "prch\_c" "prcl\_c" "cshtr\_f"   
## [139] "dvpsp\_f" "dvpsx\_f" "mkvalt" "prcc\_f" "prch\_f" "prcl\_f"   
## [145] "ein" "incorp" "state"

nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

#fundamentals\_ds\_filter\_1

### Filter and create a seperate dataset for restatement

#### Remove all the columns which contain 10% or greater NA or null values

fundamentals\_restmt\_ds\_filter <- fundamentals\_ds %>%  
 filter(datafmt == 'SUMM\_STD' & gsector == gsector\_sjm)  
 #filter(datafmt == 'SUMM\_STD')  
   
std\_cols <- colnames(fundamentals\_ds\_filter\_1)  
fundamentals\_restmt\_ds\_filter <- subset(fundamentals\_restmt\_ds\_filter, select = c(std\_cols))  
fundamentals\_restmt\_ds\_filter <- fundamentals\_restmt\_ds\_filter[ lapply( fundamentals\_restmt\_ds\_filter, function(x) sum(is.na(x)) / length(x) ) < 0.1 ]  
summary(fundamentals\_restmt\_ds\_filter)

## gvkey fyear tic at   
## Min. : 1239 Min. :2009 0161A : 5 Min. : 0.00   
## 1st Qu.: 10852 1st Qu.:2010 0173A : 5 1st Qu.: 19.66   
## Median : 29517 Median :2011 AOI : 5 Median : 317.63   
## Mean : 75238 Mean :2011 BF.B : 5 Mean : 5919.97   
## 3rd Qu.:162517 3rd Qu.:2012 BNNY : 5 3rd Qu.: 2767.12   
## Max. :264393 Max. :2013 CASY : 5 Max. :204751.00   
## (Other):1050 NA's :20   
## capx cogs dltt epsfi   
## Min. : 0.000 Min. : 0 Min. : 0.00 Min. : -70.300   
## 1st Qu.: 0.366 1st Qu.: 13 1st Qu.: 0.00 1st Qu.: -0.035   
## Median : 12.813 Median : 245 Median : 16.23 Median : 0.430   
## Mean : 229.376 Mean : 5040 Mean : 1469.99 Mean : 4.928   
## 3rd Qu.: 101.290 3rd Qu.: 2172 3rd Qu.: 789.47 3rd Qu.: 1.985   
## Max. :13510.000 Max. :349199 Max. :47079.00 Max. :1126.180   
## NA's :41 NA's :40 NA's :33 NA's :65   
## epspi ib ni nopi   
## Min. : -70.300 Min. :-1728.282 Min. :-1575.621 Min. :-2366.000   
## 1st Qu.: -0.040 1st Qu.: -0.682 1st Qu.: -0.693 1st Qu.: -5.813   
## Median : 0.395 Median : 11.068 Median : 10.489 Median : 0.000   
## Mean : 5.007 Mean : 424.227 Mean : 413.637 Mean : 10.569   
## 3rd Qu.: 1.960 3rd Qu.: 108.980 3rd Qu.: 110.281 3rd Qu.: 0.738   
## Max. :1126.180 Max. :16963.000 Max. :16999.000 Max. : 8234.000   
## NA's :86 NA's :27 NA's :30 NA's :44   
## pi reuna seq teq   
## Min. :-2251.837 Min. :-7883.37 Min. :-7766.00 Min. :-6274.00   
## 1st Qu.: -0.712 1st Qu.: -9.31 1st Qu.: 4.14 1st Qu.: 4.13   
## Median : 16.577 Median : 27.36 Median : 110.65 Median : 110.66   
## Mean : 577.995 Mean : 2151.53 Mean : 2123.17 Mean : 2184.02   
## 3rd Qu.: 158.523 3rd Qu.: 443.86 3rd Qu.: 924.87 3rd Qu.: 976.52   
## Max. :25662.000 Max. :80197.00 Max. :76343.00 Max. :81738.00   
## NA's :33 NA's :35 NA's :17 NA's :21   
## txt wcap xint xsga   
## Min. :-523.555 Min. :-11878.000 Min. : 0.000 Min. : 0.00   
## 1st Qu.: 0.000 1st Qu.: 0.337 1st Qu.: 0.150 1st Qu.: 7.22   
## Median : 3.554 Median : 27.381 Median : 3.711 Median : 87.06   
## Mean : 166.817 Mean : 213.857 Mean : 89.108 Mean : 1392.23   
## 3rd Qu.: 48.900 3rd Qu.: 257.084 3rd Qu.: 61.965 3rd Qu.: 662.60   
## Max. :8105.000 Max. : 9900.000 Max. :3341.000 Max. :90920.00   
## NA's :30 NA's :64 NA's :107 NA's :73   
## dvpsp\_c dvpsx\_c dvpsp\_f dvpsx\_f   
## Min. : 0.0000 Min. : 0.0000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median : 0.0000 Median : 0.0000 Median : 0.0000 Median : 0.0000   
## Mean : 0.4543 Mean : 0.4558 Mean : 0.4490 Mean : 0.4505   
## 3rd Qu.: 0.5867 3rd Qu.: 0.5950 3rd Qu.: 0.5769 3rd Qu.: 0.5825   
## Max. :21.0000 Max. :21.0000 Max. :21.0000 Max. :21.0000   
## NA's :65 NA's :65 NA's :62 NA's :62   
## ein incorp   
## 13-4306188: 5 DE :495   
## 16-0716709: 5 NV :179   
## 16-0733425: 5 FL : 43   
## 20-1266625: 5 VA : 35   
## 23-1614034: 5 CO : 32   
## (Other) :1014 (Other):241   
## NA's : 41 NA's : 55

nrow(fundamentals\_restmt\_ds\_filter)

## [1] 1080

### Add restement variables and the magnitude as pecentages

sample\_restmt\_ds\_filter <- fundamentals\_restmt\_ds\_filter #%>%  
 #filter(gvkey == 1076)  
sample\_ds\_filter <- fundamentals\_ds\_filter\_1 #%>%  
 #filter(gvkey == 1076)  
  
fundamentals\_ds\_filter\_1$restmt\_at <- 0  
fundamentals\_ds\_filter\_1$restmt\_at\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_capx <- 0  
fundamentals\_ds\_filter\_1$restmt\_capx\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_cogs <- 0  
fundamentals\_ds\_filter\_1$restmt\_cogs\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_dltt <- 0  
fundamentals\_ds\_filter\_1$restmt\_dltt\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_epsfi <- 0  
fundamentals\_ds\_filter\_1$restmt\_epsfi\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_epspi <- 0  
fundamentals\_ds\_filter\_1$restmt\_epspi\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_ib <- 0  
fundamentals\_ds\_filter\_1$restmt\_ib\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_ni <- 0  
fundamentals\_ds\_filter\_1$restmt\_ni\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_nopi <- 0  
fundamentals\_ds\_filter\_1$restmt\_nopi\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_pi <- 0  
fundamentals\_ds\_filter\_1$restmt\_pi\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_reuna <- 0  
fundamentals\_ds\_filter\_1$restmt\_reuna\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_seq <- 0  
fundamentals\_ds\_filter\_1$restmt\_seq\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_teq <- 0  
fundamentals\_ds\_filter\_1$restmt\_teq\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_txt <- 0  
fundamentals\_ds\_filter\_1$restmt\_txt\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_wcap <- 0  
fundamentals\_ds\_filter\_1$restmt\_wcap\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_xint <- 0  
fundamentals\_ds\_filter\_1$restmt\_xint\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_xsga <- 0  
fundamentals\_ds\_filter\_1$restmt\_xsga\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_dvpsp\_f <- 0  
fundamentals\_ds\_filter\_1$restmt\_dvpsp\_f\_mag <- 0.0  
  
fundamentals\_ds\_filter\_1$restmt\_dvpsx\_f <- 0  
fundamentals\_ds\_filter\_1$restmt\_dvpsx\_f\_mag <- 0.0  
  
for (row in 1:nrow(sample\_restmt\_ds\_filter)){  
 restmt\_item\_gvkey <- as.integer(sample\_restmt\_ds\_filter[row, "gvkey"])  
 restmt\_item\_fyear <- sample\_restmt\_ds\_filter[row, "fyear"]  
 restmt\_item\_at <- sample\_restmt\_ds\_filter[row, "at"]  
 restmt\_item\_capx <- sample\_restmt\_ds\_filter[row, "capx"]  
 restmt\_item\_cogs <- sample\_restmt\_ds\_filter[row, "cogs"]  
 restmt\_item\_dltt <- sample\_restmt\_ds\_filter[row, "dltt"]  
 restmt\_item\_epsfi <- sample\_restmt\_ds\_filter[row, "epsfi"]  
 restmt\_item\_epspi <- sample\_restmt\_ds\_filter[row, "epspi"]  
   
 restmt\_item\_ib <- sample\_restmt\_ds\_filter[row, "ib"]  
 restmt\_item\_ni <- sample\_restmt\_ds\_filter[row, "ni"]  
 restmt\_item\_nopi <- sample\_restmt\_ds\_filter[row, "nopi"]  
 restmt\_item\_pi <- sample\_restmt\_ds\_filter[row, "pi"]  
 restmt\_item\_reuna <- sample\_restmt\_ds\_filter[row, "reuna"]  
 restmt\_item\_seq <- sample\_restmt\_ds\_filter[row, "seq"]  
 restmt\_item\_teq <- sample\_restmt\_ds\_filter[row, "teq"]  
 restmt\_item\_txt <- sample\_restmt\_ds\_filter[row, "txt"]  
 restmt\_item\_wcap <- sample\_restmt\_ds\_filter[row, "wcap"]  
   
 restmt\_item\_xint <- sample\_restmt\_ds\_filter[row, "xint"]  
 restmt\_item\_xsga <- sample\_restmt\_ds\_filter[row, "xsga"]  
 restmt\_item\_dvpsp\_f <- sample\_restmt\_ds\_filter[row, "dvpsp\_f"]  
 restmt\_item\_dvpsx\_f <- sample\_restmt\_ds\_filter[row, "dvpsx\_f"]  
  
 row\_count <- as.integer(nrow(subset(fundamentals\_ds\_filter\_1, gvkey == restmt\_item\_gvkey & fyear == restmt\_item\_fyear)))  
   
 if (row\_count > 0){  
 fundamental\_stmt\_row <- fundamentals\_ds\_filter\_1 %>%  
 filter(gvkey == restmt\_item\_gvkey & fyear == restmt\_item\_fyear)  
  
 stmt\_item\_gvkey <- fundamental\_stmt\_row["gvkey"]  
 stmt\_item\_fyear <- fundamental\_stmt\_row["fyear"]  
 stmt\_item\_at <- fundamental\_stmt\_row["at"]  
 stmt\_item\_capx <- fundamental\_stmt\_row["capx"]  
 stmt\_item\_cogs <- fundamental\_stmt\_row["cogs"]  
 stmt\_item\_dltt <- fundamental\_stmt\_row["dltt"]  
 stmt\_item\_epsfi <- fundamental\_stmt\_row["epsfi"]  
 stmt\_item\_epspi <- fundamental\_stmt\_row["epspi"]  
 stmt\_item\_ib <- fundamental\_stmt\_row["ib"]  
 stmt\_item\_ni <- fundamental\_stmt\_row["ni"]  
 stmt\_item\_nopi <- fundamental\_stmt\_row["nopi"]  
 stmt\_item\_pi <- fundamental\_stmt\_row["pi"]  
 stmt\_item\_reuna <- fundamental\_stmt\_row["reuna"]  
 stmt\_item\_seq <- fundamental\_stmt\_row["seq"]  
 stmt\_item\_teq <- fundamental\_stmt\_row["teq"]  
 stmt\_item\_txt <- fundamental\_stmt\_row["txt"]  
 stmt\_item\_wcap <- fundamental\_stmt\_row["wcap"]  
 stmt\_item\_xint <- fundamental\_stmt\_row["xint"]  
 stmt\_item\_xsga <- fundamental\_stmt\_row["xsga"]  
 stmt\_item\_dvpsp\_f <- fundamental\_stmt\_row["dvpsp\_f"]  
 stmt\_item\_dvpsx\_f <- fundamental\_stmt\_row["dvpsx\_f"]  
  
   
  
 if (!is.na(restmt\_item\_at) & !is.na(stmt\_item\_at) & stmt\_item\_at != 0 & restmt\_item\_at != stmt\_item\_at){  
 fundamentals\_ds\_filter\_1$restmt\_at[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_at - stmt\_item\_at)/stmt\_item\_at) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_at\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_capx) & !is.na(stmt\_item\_capx) & restmt\_item\_capx != stmt\_item\_capx){  
 fundamentals\_ds\_filter\_1$restmt\_capx[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_capx == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_capx - stmt\_item\_capx)/stmt\_item\_capx) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_capx\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_cogs) & !is.na(stmt\_item\_cogs) & restmt\_item\_cogs != stmt\_item\_cogs){  
 fundamentals\_ds\_filter\_1$restmt\_cogs[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_cogs == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_cogs - stmt\_item\_cogs)/stmt\_item\_cogs) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_cogs\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_dltt) & !is.na(stmt\_item\_dltt) & restmt\_item\_dltt != stmt\_item\_dltt){  
 fundamentals\_ds\_filter\_1$restmt\_dltt[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_dltt == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_dltt - stmt\_item\_dltt)/stmt\_item\_dltt) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_dltt\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_epsfi) & !is.na(stmt\_item\_epsfi) & restmt\_item\_epsfi != stmt\_item\_epsfi){  
 fundamentals\_ds\_filter\_1$restmt\_epsfi[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_epsfi == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_epsfi - stmt\_item\_epsfi)/stmt\_item\_epsfi) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_epsfi\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_epspi) & !is.na(stmt\_item\_epspi) & restmt\_item\_epspi != stmt\_item\_epspi){  
 fundamentals\_ds\_filter\_1$restmt\_epspi[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_epspi == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_epspi - stmt\_item\_epspi)/stmt\_item\_epspi) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_epspi\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_ib) & !is.na(stmt\_item\_ib) & restmt\_item\_ib != stmt\_item\_ib){  
 fundamentals\_ds\_filter\_1$restmt\_ib[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_ib - stmt\_item\_ib)/stmt\_item\_ib) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_ib\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_ni) & !is.na(stmt\_item\_ni) & restmt\_item\_ni != stmt\_item\_ni){  
 fundamentals\_ds\_filter\_1$restmt\_ni[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_ni - stmt\_item\_ni)/stmt\_item\_ni) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_ni\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_nopi) & !is.na(stmt\_item\_nopi) & restmt\_item\_nopi != stmt\_item\_nopi){  
 fundamentals\_ds\_filter\_1$restmt\_nopi[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_nopi == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_nopi - stmt\_item\_nopi)/stmt\_item\_nopi) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_nopi\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_pi) & !is.na(stmt\_item\_pi) & restmt\_item\_pi != stmt\_item\_pi){  
 fundamentals\_ds\_filter\_1$restmt\_pi[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_pi - stmt\_item\_pi)/stmt\_item\_pi) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_pi\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_reuna) & !is.na(stmt\_item\_reuna) & restmt\_item\_reuna != stmt\_item\_reuna){  
 fundamentals\_ds\_filter\_1$restmt\_reuna[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_reuna - stmt\_item\_reuna)/stmt\_item\_reuna) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_reuna\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_seq) & !is.na(stmt\_item\_seq) & restmt\_item\_seq != stmt\_item\_seq){  
 fundamentals\_ds\_filter\_1$restmt\_seq[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_seq - stmt\_item\_seq)/stmt\_item\_seq) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_seq\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_teq) & !is.na(stmt\_item\_teq) & restmt\_item\_teq != stmt\_item\_teq){  
 fundamentals\_ds\_filter\_1$restmt\_teq[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_teq - stmt\_item\_teq)/stmt\_item\_teq) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_teq\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_txt) & !is.na(stmt\_item\_txt) & restmt\_item\_txt != stmt\_item\_txt){  
 fundamentals\_ds\_filter\_1$restmt\_txt[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 if (stmt\_item\_txt == 0.0){  
 magnitude <- 100.00  
 }  
 else{  
 magnitude <- ((restmt\_item\_txt - stmt\_item\_txt)/stmt\_item\_txt) \* 100.0  
 }  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_txt\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
  
 if (!is.na(restmt\_item\_wcap) & !is.na(stmt\_item\_wcap) & restmt\_item\_wcap != stmt\_item\_wcap){  
 fundamentals\_ds\_filter\_1$restmt\_wcap[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_wcap - stmt\_item\_wcap)/stmt\_item\_wcap) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_wcap\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
   
 if (!is.na(restmt\_item\_xint) & !is.na(stmt\_item\_xint) & stmt\_item\_xint != 0 & restmt\_item\_xint != stmt\_item\_xint){  
 fundamentals\_ds\_filter\_1$restmt\_xint[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_xint - stmt\_item\_xint)/stmt\_item\_xint) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_xint\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
   
 if (!is.na(restmt\_item\_xsga) & !is.na(stmt\_item\_xsga) & restmt\_item\_xsga != stmt\_item\_xsga){  
 fundamentals\_ds\_filter\_1$restmt\_xsga[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_xsga - stmt\_item\_xsga)/stmt\_item\_xsga) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_xsga\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
   
 if (!is.na(restmt\_item\_dvpsp\_f) & !is.na(stmt\_item\_dvpsp\_f) & restmt\_item\_dvpsp\_f != stmt\_item\_dvpsp\_f){  
 fundamentals\_ds\_filter\_1$restmt\_dvpsp\_f[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_dvpsp\_f - stmt\_item\_dvpsp\_f)/stmt\_item\_dvpsp\_f) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_dvpsp\_f\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
   
 if (!is.na(restmt\_item\_dvpsx\_f) & !is.na(stmt\_item\_dvpsx\_f) & restmt\_item\_dvpsx\_f != stmt\_item\_dvpsx\_f){  
 fundamentals\_ds\_filter\_1$restmt\_dvpsx\_f[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- 1  
 magnitude <- ((restmt\_item\_dvpsx\_f - stmt\_item\_dvpsx\_f)/stmt\_item\_dvpsx\_f) \* 100.0  
 magnitude <- as.double(round(magnitude, digits = 3))  
 fundamentals\_ds\_filter\_1$restmt\_dvpsx\_f\_mag[fundamentals\_ds\_filter\_1$gvkey == restmt\_item\_gvkey & fundamentals\_ds\_filter\_1$fyear == restmt\_item\_fyear] <- magnitude  
 }  
   
 }  
}  
nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

### Removing aco == NA as all those rows do not have any data

nrow(fundamentals\_ds\_filter\_1)

## [1] 1243

fundamentals\_ds\_filter\_2 <- subset(fundamentals\_ds\_filter\_1, !is.na(aco))  
nrow(fundamentals\_ds\_filter\_2)

## [1] 1205

fundamentals\_ds\_filter\_2[is.na(fundamentals\_ds\_filter\_2)] <- 0  
#summary(fundamentals\_ds\_filter\_2)

colnames(fundamentals\_ds\_filter\_2)

## [1] "gvkey" "fyear" "tic"   
## [4] "aco" "acominc" "act"   
## [7] "am" "ao" "aocidergl"   
## [10] "aociother" "aocipen" "aodo"   
## [13] "aoloch" "ap" "aqc"   
## [16] "at" "bkvlps" "caps"   
## [19] "capx" "ceq" "ceqt"   
## [22] "ch" "che" "chech"   
## [25] "ci" "cogs" "cshi"   
## [28] "csho" "cshr" "cstk"   
## [31] "cstkcv" "dc" "dd"   
## [34] "dd1" "dd2" "dd3"   
## [37] "dd4" "dd5" "dilavx"   
## [40] "dlc" "dltp" "dltt"   
## [43] "dm" "dn" "dpact"   
## [46] "dpc" "dvt" "ebit"   
## [49] "ebitda" "emp" "epsfi"   
## [52] "epspi" "esub" "esubc"   
## [55] "fatb" "fatc" "fatp"   
## [58] "fiao" "fincf" "fopo"   
## [61] "gdwl" "gp" "ib"   
## [64] "icapt" "intan" "intano"   
## [67] "intc" "intpn" "invt"   
## [70] "ivaeq" "ivch" "ivncf"   
## [73] "ivst" "ivstch" "lifr"   
## [76] "lo" "lse" "lt"   
## [79] "mrc1" "mrcta" "ni"   
## [82] "nopi" "nopio" "oancf"   
## [85] "oiadp" "oibdp" "opeps"   
## [88] "pi" "pncad" "pncaeps"   
## [91] "ppegt" "prca" "prstkc"   
## [94] "re" "reajo" "recch"   
## [97] "recd" "rect" "recta"   
## [100] "reuna" "revt" "seq"   
## [103] "seqo" "siv" "spce"   
## [106] "spi" "sppe" "sppiv"   
## [109] "sstk" "stkco" "teq"   
## [112] "tstk" "tstkn" "txc"   
## [115] "txdba" "txdbcl" "txdc"   
## [118] "txfed" "txfo" "txndba"   
## [121] "txndbl" "txp" "txpd"   
## [124] "txr" "txs" "txt"   
## [127] "wcap" "xacc" "xint"   
## [130] "xrent" "xsga" "cshtr\_c"   
## [133] "dvpsp\_c" "dvpsx\_c" "prcc\_c"   
## [136] "prch\_c" "prcl\_c" "cshtr\_f"   
## [139] "dvpsp\_f" "dvpsx\_f" "mkvalt"   
## [142] "prcc\_f" "prch\_f" "prcl\_f"   
## [145] "ein" "incorp" "state"   
## [148] "restmt\_at" "restmt\_at\_mag" "restmt\_capx"   
## [151] "restmt\_capx\_mag" "restmt\_cogs" "restmt\_cogs\_mag"   
## [154] "restmt\_dltt" "restmt\_dltt\_mag" "restmt\_epsfi"   
## [157] "restmt\_epsfi\_mag" "restmt\_epspi" "restmt\_epspi\_mag"   
## [160] "restmt\_ib" "restmt\_ib\_mag" "restmt\_ni"   
## [163] "restmt\_ni\_mag" "restmt\_nopi" "restmt\_nopi\_mag"   
## [166] "restmt\_pi" "restmt\_pi\_mag" "restmt\_reuna"   
## [169] "restmt\_reuna\_mag" "restmt\_seq" "restmt\_seq\_mag"   
## [172] "restmt\_teq" "restmt\_teq\_mag" "restmt\_txt"   
## [175] "restmt\_txt\_mag" "restmt\_wcap" "restmt\_wcap\_mag"   
## [178] "restmt\_xint" "restmt\_xint\_mag" "restmt\_xsga"   
## [181] "restmt\_xsga\_mag" "restmt\_dvpsp\_f" "restmt\_dvpsp\_f\_mag"  
## [184] "restmt\_dvpsx\_f" "restmt\_dvpsx\_f\_mag"

nrow(fundamentals\_ds\_filter\_2)

## [1] 1205

### The dataset contains company wise data across financial year, this dataset needs to be consolidate to single ror for each of the company

### So group the dataset by gvkey and summarize all variables

final\_ds\_initial <- fundamentals\_ds\_filter\_2 %>%  
 group\_by(gvkey,tic) %>%  
 dplyr::summarize(  
 aco = mean(aco),  
 acominc = mean(acominc),  
 act = mean(act),  
 ao = mean(ao),  
 aocidergl = mean(aocidergl),  
 aocipen = mean(aocipen),  
 aodo = mean(aodo),  
 aoloch = mean(aoloch),  
 ap = mean(ap),  
 aqc = mean(aqc),  
 at = mean(at),  
 bkvlps = mean(bkvlps),  
 caps = mean(caps),  
 capx = mean(capx),  
 ceq = mean(ceq),  
 ceqt = mean(ceqt),  
 ch = mean(ch),  
 che = mean(che),  
 chech = mean(chech),  
 ci = mean(ci),  
 cogs = mean(cogs),  
 cshi = mean(cshi),  
 csho = mean(csho),  
 cstk = mean(cstk),  
 cstkcv = mean(cstkcv),  
 dd1 = mean(dd1),  
 dilavx = mean(dilavx),  
 dlc = mean(dlc),  
 dltt = mean(dltt),  
 dm = mean(dm),  
 dn = mean(dn),  
 dpact = mean(dpact),  
 dpc = mean(dpc),  
 dvt = mean(dvt),  
 ebit = mean(ebit),  
 ebitda = mean(ebitda),  
 epsfi = mean(epsfi),  
 epspi = mean(epspi),  
 fiao = mean(fiao),  
 fincf = mean(fincf),  
 fopo = mean(fopo),  
 gdwl = mean(gdwl),  
 gp = mean(gp),  
 ib = mean(ib),  
 icapt = mean(icapt),  
 intan = mean(intan),  
 intano = mean(intano),  
 invt = mean(invt),  
 ivch = mean(ivch),  
 ivncf = mean(ivncf),  
 ivst = mean(ivst),  
 lo = mean(lo),  
 lse = mean(lse),  
 lt = mean(lt),  
 ni = mean(ni),  
 nopi = mean(nopi),  
 nopio = mean(nopio),  
 oancf = mean(oancf),  
 oiadp = mean(oiadp),  
 oibdp = mean(oibdp),  
 opeps = mean(opeps),  
 pi = mean(pi),  
 ppegt = mean(ppegt),  
 re = mean(re),  
 reajo = mean(reajo),  
 rect = mean(rect),  
 recta = mean(recta),  
 reuna = mean(reuna),  
 revt = mean(revt),  
 seq = mean( seq ),  
 siv = mean( siv ),  
 spce = mean(spce),  
 spi = mean(spi),  
 sppiv = mean(sppiv),  
 sstk = mean(sstk),  
 teq = mean(teq),  
 tstk = mean(tstk),  
 tstkn = mean(tstkn),  
 txp = mean(txp),  
 txr = mean(txr),  
 txt = mean(txt),  
 wcap = mean(wcap),  
 xint = mean(xint),  
 restmt\_at = mean(restmt\_at),  
 restmt\_at\_mag = mean(restmt\_at\_mag),  
 restmt\_capx = mean(restmt\_capx),  
 restmt\_capx\_mag = mean(restmt\_capx\_mag),  
 restmt\_cogs = mean(restmt\_cogs),  
 restmt\_cogs\_mag = mean(restmt\_cogs\_mag),  
 restmt\_dltt = mean(restmt\_dltt),  
 restmt\_dltt\_mag = mean(restmt\_dltt\_mag),  
 restmt\_epsfi = mean(restmt\_epsfi),  
 restmt\_epsfi\_mag = mean(restmt\_epsfi\_mag),  
 restmt\_epspi = mean(restmt\_epspi),  
 restmt\_epspi\_mag = mean(restmt\_epspi\_mag),  
 restmt\_ib = mean(restmt\_ib),  
 restmt\_ib\_mag = mean(restmt\_ib\_mag),  
 restmt\_ni = mean(restmt\_ni),  
 restmt\_ni\_mag = mean(restmt\_ni\_mag),  
 restmt\_nopi = mean(restmt\_nopi),  
 restmt\_nopi\_mag = mean(restmt\_nopi\_mag),  
 restmt\_pi = mean(restmt\_pi),  
 restmt\_pi\_mag = mean(restmt\_pi\_mag),  
 restmt\_reuna = mean(restmt\_reuna),  
 restmt\_reuna\_mag = mean(restmt\_reuna\_mag),  
 restmt\_seq = mean(restmt\_seq),  
 restmt\_seq\_mag = mean(restmt\_seq\_mag),  
 restmt\_teq = mean(restmt\_teq),  
 restmt\_teq\_mag = mean(restmt\_teq\_mag),  
 restmt\_txt = mean(restmt\_txt),  
 restmt\_txt\_mag = mean(restmt\_txt\_mag),  
 restmt\_wcap = mean(restmt\_wcap),  
 restmt\_wcap\_mag = mean(restmt\_wcap\_mag),  
   
 restmt\_xint = mean(restmt\_xint),  
 restmt\_xint\_mag = mean(restmt\_xint\_mag),  
   
 restmt\_xsga = mean(restmt\_xsga),  
 restmt\_xsga\_mag = mean(restmt\_xsga\_mag),  
   
 restmt\_dvpsp\_f = mean(restmt\_dvpsp\_f),  
 restmt\_dvpsp\_f\_mag = mean(restmt\_dvpsp\_f\_mag),  
   
 restmt\_dvpsx\_f = mean(restmt\_dvpsx\_f),  
 restmt\_dvpsx\_f\_mag = mean(restmt\_dvpsx\_f\_mag)  
   
 )

## `summarise()` regrouping output by 'gvkey' (override with `.groups` argument)

#summary(final\_ds\_initial)  
nrow(final\_ds\_initial)

## [1] 348

### Adjust the restement values, in case if the restate percentage is greater than 50% then mark restatement as 1 or else mark as 0 For 0/false restatement magnitude is marked as 0.

final\_ds\_initial\_1 <- final\_ds\_initial   
  
for (row in 1:nrow(final\_ds\_initial\_1)){  
 row\_item\_gvkey <- as.integer(final\_ds\_initial\_1[row, "gvkey"])  
   
 restmt\_at <- final\_ds\_initial\_1[row, "restmt\_at"]  
 restmt\_at\_mag <- final\_ds\_initial\_1[row, "restmt\_at\_mag"]  
 if (restmt\_at >= 0.5){  
 restmt\_at <- 1  
 restmt\_at\_mag <- as.double(restmt\_at\_mag)  
 }  
 else{  
 restmt\_at <- 0  
 restmt\_at\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_at[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_at  
 final\_ds\_initial\_1$restmt\_at\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_at\_mag  
   
 restmt\_capx <- final\_ds\_initial\_1[row, "restmt\_capx"]  
 restmt\_capx\_mag <- final\_ds\_initial\_1[row, "restmt\_capx\_mag"]  
 if (restmt\_capx >= 0.5){  
 restmt\_capx <- 1  
 restmt\_capx\_mag <- as.double(restmt\_capx\_mag)  
 }  
 else{  
 restmt\_capx <- 0  
 restmt\_capx\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_capx[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_capx  
 final\_ds\_initial\_1$restmt\_capx\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_capx\_mag  
   
 restmt\_cogs <- final\_ds\_initial\_1[row, "restmt\_cogs"]  
 restmt\_cogs\_mag <- final\_ds\_initial\_1[row, "restmt\_cogs\_mag"]  
 if (restmt\_cogs >= 0.5){  
 restmt\_cogs <- 1  
 restmt\_cogs\_mag <- as.double(restmt\_cogs\_mag)  
 }  
 else{  
 restmt\_cogs <- 0  
 restmt\_cogs\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_cogs[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_cogs  
 final\_ds\_initial\_1$restmt\_cogs\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_cogs\_mag  
   
 restmt\_dltt <- final\_ds\_initial\_1[row, "restmt\_dltt"]  
 restmt\_dltt\_mag <- final\_ds\_initial\_1[row, "restmt\_dltt\_mag"]  
 if (restmt\_dltt >= 0.5){  
 restmt\_dltt <- 1  
 restmt\_dltt\_mag <- as.double(restmt\_dltt\_mag)  
 }  
 else{  
 restmt\_dltt <- 0  
 restmt\_dltt\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_dltt[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dltt  
 final\_ds\_initial\_1$restmt\_dltt\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dltt\_mag  
   
   
 restmt\_epsfi <- final\_ds\_initial\_1[row, "restmt\_epsfi"]  
 restmt\_epsfi\_mag <- final\_ds\_initial\_1[row, "restmt\_epsfi\_mag"]  
 if (restmt\_epsfi >= 0.5){  
 restmt\_epsfi <- 1  
 restmt\_epsfi\_mag <- as.double(restmt\_epsfi\_mag)  
 }  
 else{  
 restmt\_epsfi <- 0  
 restmt\_epsfi\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_epsfi[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_epsfi  
 final\_ds\_initial\_1$restmt\_epsfi\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_epsfi\_mag  
   
   
 restmt\_epspi <- final\_ds\_initial\_1[row, "restmt\_epspi"]  
 restmt\_epspi\_mag <- final\_ds\_initial\_1[row, "restmt\_epspi\_mag"]  
 if (restmt\_epspi >= 0.5){  
 restmt\_epspi <- 1  
 restmt\_epspi\_mag <- as.double(restmt\_epspi\_mag)  
 }  
 else{  
 restmt\_epspi <- 0  
 restmt\_epspi\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_epspi[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_epspi  
 final\_ds\_initial\_1$restmt\_epspi\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_epspi\_mag  
   
 restmt\_ib <- final\_ds\_initial\_1[row, "restmt\_ib"]  
 restmt\_ib\_mag <- final\_ds\_initial\_1[row, "restmt\_ib\_mag"]  
 if (restmt\_ib >= 0.5){  
 restmt\_ib <- 1  
 restmt\_ib\_mag <- as.double(restmt\_ib\_mag)  
 }  
 else{  
 restmt\_ib <- 0  
 restmt\_ib\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_ib[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_ib  
 final\_ds\_initial\_1$restmt\_ib\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_ib\_mag  
   
 restmt\_ni <- final\_ds\_initial\_1[row, "restmt\_ni"]  
 restmt\_ni\_mag <- final\_ds\_initial\_1[row, "restmt\_ni\_mag"]  
 if (restmt\_ni >= 0.5){  
 restmt\_ni <- 1  
 restmt\_ni\_mag <- as.double(restmt\_ni\_mag)  
 }  
 else{  
 restmt\_ni <- 0  
 restmt\_ni\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_ni[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_ni  
 final\_ds\_initial\_1$restmt\_ni\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_ni\_mag  
   
 restmt\_nopi <- final\_ds\_initial\_1[row, "restmt\_nopi"]  
 restmt\_nopi\_mag <- final\_ds\_initial\_1[row, "restmt\_nopi\_mag"]  
 if (restmt\_nopi >= 0.5){  
 restmt\_nopi <- 1  
 restmt\_nopi\_mag <- as.double(restmt\_nopi\_mag)  
 }  
 else{  
 restmt\_nopi <- 0  
 restmt\_nopi\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_nopi[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_nopi  
 final\_ds\_initial\_1$restmt\_nopi\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_nopi\_mag  
   
 restmt\_pi <- final\_ds\_initial\_1[row, "restmt\_pi"]  
 restmt\_pi\_mag <- final\_ds\_initial\_1[row, "restmt\_pi\_mag"]  
 if (restmt\_pi >= 0.5){  
 restmt\_pi <- 1  
 restmt\_pi\_mag <- as.double(restmt\_pi\_mag)  
 }  
 else{  
 restmt\_pi <- 0  
 restmt\_pi\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_pi[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_pi  
 final\_ds\_initial\_1$restmt\_pi\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_pi\_mag  
   
 restmt\_reuna <- final\_ds\_initial\_1[row, "restmt\_reuna"]  
 restmt\_reuna\_mag <- final\_ds\_initial\_1[row, "restmt\_reuna\_mag"]  
 if (restmt\_reuna >= 0.5){  
 restmt\_reuna <- 1  
 restmt\_reuna\_mag <- as.double(restmt\_reuna\_mag)  
 }  
 else{  
 restmt\_reuna <- 0  
 restmt\_reuna\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_reuna[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_reuna  
 final\_ds\_initial\_1$restmt\_reuna\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_reuna\_mag  
   
 restmt\_seq <- final\_ds\_initial\_1[row, "restmt\_seq"]  
 restmt\_seq\_mag <- final\_ds\_initial\_1[row, "restmt\_seq\_mag"]  
 if (restmt\_seq >= 0.5){  
 restmt\_seq <- 1  
 restmt\_seq\_mag <- as.double(restmt\_seq\_mag)  
 }  
 else{  
 restmt\_seq <- 0  
 restmt\_seq\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_seq[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_seq  
 final\_ds\_initial\_1$restmt\_seq\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_seq\_mag  
   
 restmt\_teq <- final\_ds\_initial\_1[row, "restmt\_teq"]  
 restmt\_teq\_mag <- final\_ds\_initial\_1[row, "restmt\_teq\_mag"]  
 if (restmt\_teq >= 0.5){  
 restmt\_teq <- 1  
 restmt\_teq\_mag <- as.double(restmt\_teq\_mag)  
 }  
 else{  
 restmt\_teq <- 0  
 restmt\_teq\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_teq[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_teq  
 final\_ds\_initial\_1$restmt\_teq\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_teq\_mag  
   
 restmt\_txt <- final\_ds\_initial\_1[row, "restmt\_txt"]  
 restmt\_txt\_mag <- final\_ds\_initial\_1[row, "restmt\_txt\_mag"]  
 if (restmt\_txt >= 0.5){  
 restmt\_txt <- 1  
 restmt\_txt\_mag <- as.double(restmt\_txt\_mag)  
 }  
 else{  
 restmt\_txt <- 0  
 restmt\_txt\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_txt[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_txt  
 final\_ds\_initial\_1$restmt\_txt\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_txt\_mag  
   
 restmt\_wcap <- final\_ds\_initial\_1[row, "restmt\_wcap"]  
 restmt\_wcap\_mag <- final\_ds\_initial\_1[row, "restmt\_wcap\_mag"]  
 if (restmt\_wcap >= 0.5){  
 restmt\_wcap <- 1  
 restmt\_wcap\_mag <- as.double(restmt\_wcap\_mag)  
 }  
 else{  
 restmt\_wcap <- 0  
 restmt\_wcap\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_wcap[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_wcap  
 final\_ds\_initial\_1$restmt\_wcap\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_wcap\_mag  
   
 restmt\_xint <- final\_ds\_initial\_1[row, "restmt\_xint"]  
 restmt\_xint\_mag <- final\_ds\_initial\_1[row, "restmt\_xint\_mag"]  
 if (restmt\_xint >= 0.5){  
 restmt\_xint <- 1  
 restmt\_xint\_mag <- as.double(restmt\_xint\_mag)  
 }  
 else{  
 restmt\_xint <- 0  
 restmt\_xint\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_xint[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_xint  
 final\_ds\_initial\_1$restmt\_xint\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_xint\_mag  
   
 restmt\_xsga <- final\_ds\_initial\_1[row, "restmt\_xsga"]  
 restmt\_xsga\_mag <- final\_ds\_initial\_1[row, "restmt\_xsga\_mag"]  
 if (restmt\_xsga >= 0.5){  
 restmt\_xsga <- 1  
 restmt\_xsga\_mag <- as.double(restmt\_xsga\_mag)  
 }  
 else{  
 restmt\_xsga <- 0  
 restmt\_xsga\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_xsga[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_xsga  
 final\_ds\_initial\_1$restmt\_xsga\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_xsga\_mag  
   
 restmt\_dvpsp\_f <- final\_ds\_initial\_1[row, "restmt\_dvpsp\_f"]  
 restmt\_dvpsp\_f\_mag <- final\_ds\_initial\_1[row, "restmt\_dvpsp\_f\_mag"]  
 if (restmt\_dvpsp\_f >= 0.5){  
 restmt\_dvpsp\_f <- 1  
 restmt\_dvpsp\_f\_mag <- as.double(restmt\_dvpsp\_f\_mag)  
 }  
 else{  
 restmt\_dvpsp\_f <- 0  
 restmt\_dvpsp\_f\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_dvpsp\_f[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dvpsp\_f  
 final\_ds\_initial\_1$restmt\_dvpsp\_f\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dvpsp\_f\_mag  
   
 restmt\_dvpsx\_f <- final\_ds\_initial\_1[row, "restmt\_dvpsx\_f"]  
 restmt\_dvpsx\_f\_mag <- final\_ds\_initial\_1[row, "restmt\_dvpsx\_f\_mag"]  
 if (restmt\_dvpsx\_f >= 0.5){  
 restmt\_dvpsx\_f <- 1  
 restmt\_dvpsx\_f\_mag <- as.double(restmt\_dvpsx\_f\_mag)  
 }  
 else{  
 restmt\_dvpsx\_f <- 0  
 restmt\_dvpsx\_f\_mag <- 0.0  
 }  
 final\_ds\_initial\_1$restmt\_dvpsx\_f[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dvpsx\_f  
 final\_ds\_initial\_1$restmt\_dvpsx\_f\_mag[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- restmt\_dvpsx\_f\_mag  
}  
  
restmt\_var\_ds <- subset(final\_ds\_initial\_1, select = c(gvkey,  
 restmt\_at,restmt\_at\_mag,  
 restmt\_capx,restmt\_capx\_mag,  
 restmt\_cogs, restmt\_cogs\_mag,  
 restmt\_dltt, restmt\_dltt\_mag,  
 restmt\_epsfi, restmt\_epsfi\_mag,  
 restmt\_epspi, restmt\_epspi\_mag,  
 restmt\_ib, restmt\_ib\_mag,  
 restmt\_ni, restmt\_ni\_mag,  
 restmt\_nopi, restmt\_nopi\_mag,  
 restmt\_pi, restmt\_pi\_mag,  
 restmt\_reuna, restmt\_reuna\_mag,  
 restmt\_seq, restmt\_seq\_mag,  
 restmt\_teq, restmt\_teq\_mag,  
 restmt\_txt, restmt\_txt\_mag,  
 restmt\_wcap, restmt\_wcap\_mag,  
   
 restmt\_xint, restmt\_xint\_mag,  
 restmt\_xsga, restmt\_xsga\_mag,  
 restmt\_dvpsp\_f, restmt\_dvpsp\_f\_mag,  
 restmt\_dvpsx\_f, restmt\_dvpsx\_f\_mag  
 ))  
  
#summary(restmt\_var\_ds)

final\_ds\_initial\_2 <- final\_ds\_initial\_1  
#summary(final\_ds\_initial\_2)  
nrow(final\_ds\_initial\_2)

## [1] 348

write.csv(final\_ds\_initial\_2, file = "data/final\_ds\_initial\_2.csv", row.names=FALSE, na="")

### This first step where all the correlated variables are idenfied and then removed. This will reduce the Collinearity.

cor\_matrix\_ds <- subset(final\_ds\_initial\_2, select = -c(gvkey,tic, aodo,seq,ivch,nopio,spce,reuna,dilavx,ebitda,cshi,epsfi,   
 ib,pi,  
 oiadp,oibdp,gdwl))  
cor\_matrix <- cor(cor\_matrix\_ds)  
cor\_matrix %>%  
 as.data.frame() %>%  
 mutate(var1 = rownames(.)) %>%  
 gather(var2, value, -var1) %>%  
 arrange(desc(value)) %>%  
 group\_by(value) %>%  
 filter(row\_number()==1)

## # A tibble: 5,052 x 3  
## # Groups: value [5,052]  
## var1 var2 value  
## <chr> <chr> <dbl>  
## 1 aco aco 1   
## 2 restmt\_epspi\_mag restmt\_epsfi\_mag 1.00   
## 3 restmt\_teq\_mag restmt\_seq\_mag 1.00   
## 4 opeps epspi 1.00   
## 5 restmt\_pi\_mag restmt\_ib\_mag 0.998  
## 6 restmt\_ni\_mag restmt\_ib\_mag 0.998  
## 7 restmt\_xsga\_mag restmt\_ni\_mag 0.997  
## 8 restmt\_xsga\_mag restmt\_pi\_mag 0.997  
## 9 restmt\_pi\_mag restmt\_ni\_mag 0.997  
## 10 restmt\_xsga\_mag restmt\_ib\_mag 0.997  
## # ... with 5,042 more rows

fundamentals\_final\_ds <- subset(final\_ds\_initial\_2, select = -c(aodo,seq,ivch,nopio,spce,reuna,dilavx,ebitda,cshi,epsfi,   
 ib,pi,  
 oiadp,oibdp,gdwl))  
#summary(fundamentals\_final\_ds)  
nrow(fundamentals\_final\_ds)

## [1] 348

### Loan stocks data file

stocks\_init\_ds <- read.csv("./data/Stocks\_DS.csv", na.strings=c(""," "))  
nrow(stocks\_init\_ds)

## [1] 4187047

### From Stocks daat file choose only certain variables

### prccd ==> Price - Close - Daily

### prchd ==> Price - High - Daily

### prcld ==> Price - Low - Daily

### prcod ==> Price - Open - Daily

### trfd ==> Daily Total Return Factor

### cshtrd ==> Trading Volume - Daily

names(stocks\_init\_ds)[names(stocks\_init\_ds) == "ï..gvkey"] <- "gvkey"  
stocks\_init\_limited\_cols <- subset(stocks\_init\_ds, select = c(gvkey,cshtrd,prccd,prchd,prcld,prcod,trfd))  
stocks\_init\_limited\_cols <- stocks\_init\_limited\_cols[!is.na(stocks\_init\_limited\_cols$cshtrd)&!is.na(stocks\_init\_limited\_cols$prccd)  
 &!is.na(stocks\_init\_limited\_cols$prchd)&!is.na(stocks\_init\_limited\_cols$prcld)  
 &!is.na(stocks\_init\_limited\_cols$trfd),]  
stocks\_init\_limited\_cols$prcod[is.na(stocks\_init\_limited\_cols$prcod)] <- (stocks\_init\_limited\_cols$prchd + stocks\_init\_limited\_cols$prcld)/2  
  
stocks\_grouped\_data <- stocks\_init\_limited\_cols %>%  
 group\_by(gvkey) %>%  
 dplyr::summarize(  
 cshtrd\_m = mean(cshtrd),  
 prccd\_m = mean(prccd),  
 prchd\_m = mean(prchd),  
 prcld\_m = mean(prcld),  
 prcod\_m = mean(prcod),  
 trfd\_m = mean(trfd)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

ncol(stocks\_init\_ds)

## [1] 76

### Merge the stocks dataset with final dataset.

for (row in 1:nrow(fundamentals\_final\_ds)){  
 row\_item\_gvkey <- as.integer(fundamentals\_final\_ds[row, "gvkey"])  
   
 specific\_stock <- stocks\_grouped\_data %>%  
 filter(gvkey == row\_item\_gvkey)   
   
 if (nrow(specific\_stock) > 0){  
 specific\_stock <- head(specific\_stock, 1)  
 fundamentals\_final\_ds$cshtrd\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$cshtrd\_m  
 fundamentals\_final\_ds$prccd\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$prccd\_m  
 fundamentals\_final\_ds$prchd\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$prchd\_m  
 fundamentals\_final\_ds$prcld\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$prcld\_m  
 fundamentals\_final\_ds$prcod\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$prcod\_m  
 fundamentals\_final\_ds$trfd\_m[final\_ds\_initial\_1$gvkey == row\_item\_gvkey] <- specific\_stock$trfd\_m  
 }  
}  
#summary(fundamentals\_final\_ds)  
nrow(fundamentals\_final\_ds)

## [1] 348

fundamental\_stocks\_data <- fundamentals\_final\_ds[!is.na(fundamentals\_final\_ds$cshtrd\_m) &!is.na(fundamentals\_final\_ds$prccd\_m)  
 &!is.na(fundamentals\_final\_ds$prchd\_m) &!is.na(fundamentals\_final\_ds$prcld\_m)  
 &!is.na(fundamentals\_final\_ds$prcod\_m) &!is.na(fundamentals\_final\_ds$trfd\_m),]

fundamental\_stocks\_data\_final <- fundamental\_stocks\_data  
#summary(fundamental\_stocks\_data\_final)  
nrow(fundamental\_stocks\_data\_final)

## [1] 333

### Load securities dataset

securities\_init\_ds <- read.csv("./data/Securities\_DS.csv", na.strings=c(""," "))  
names(securities\_init\_ds)[names(securities\_init\_ds) == "ï..gvkey"] <- "gvkey"  
ncol(securities\_init\_ds)

## [1] 55

securities\_init\_ds\_1 <- subset(securities\_init\_ds, select = -c(iid,isalrt,primiss,ajexm,  
 spgim,spiim,spmim,cheqvm,curcddvm,dvpsxm,  
 sphcusip,sphiid, sphmid,sphname,sphsec,sphtic,sphvg,sph100,  
 cyear,mkvalincl,exchg,tpci,city,  
 conml,costat,ggroup,gind, gsubind,loc,naics,sic,state, curcdm,   
 navm,adrrm,rawpm,rawxm,cshoq,csfsm,  
 datadate,tic,conm,cmth  
 ))  
  
  
#summary(securities\_init\_ds\_1)

### Choose following variables from the securities dataset

### trfm ==> Monthly Total Return Factor

### dvrate ==> Dividend Rate - Monthly

fund\_stock\_securities\_ds <- fundamental\_stocks\_data\_final   
securities\_init\_ds\_2 <- securities\_init\_ds\_1 %>%  
 filter(!is.na(trfm) & !is.na(trt1m)) %>%  
 group\_by(gvkey) %>%  
 dplyr::summarize(  
 trfm\_m = mean(trfm)  
 )  
  
fund\_stock\_securities\_ds$trfm\_m <- NA  
for (row in 1:nrow(fund\_stock\_securities\_ds)){  
 row\_item\_gvkey <- as.integer(fund\_stock\_securities\_ds[row, "gvkey"])  
 specific\_security <- securities\_init\_ds\_2 %>%  
 filter(gvkey == row\_item\_gvkey)   
 if (nrow(specific\_security) > 0){  
 security\_row <- head(specific\_security, 1)  
 trfm\_m <- as.numeric(security\_row$trfm\_m)   
 fund\_stock\_securities\_ds$trfm\_m[fund\_stock\_securities\_ds$gvkey == row\_item\_gvkey] <- trfm\_m  
 }  
}  
  
  
  
securities\_init\_ds\_3 <- securities\_init\_ds\_1 %>%  
 filter(!is.na(dvrate)) %>%  
 group\_by(gvkey) %>%  
 dplyr::summarize(  
 dvrate\_m = mean(dvrate)  
 )  
  
fund\_stock\_securities\_ds$dvrate\_m <- NA  
for (row in 1:nrow(fund\_stock\_securities\_ds)){  
 row\_item\_gvkey <- as.integer(fund\_stock\_securities\_ds[row, "gvkey"])  
 specific\_security <- securities\_init\_ds\_3 %>%  
 filter(gvkey == row\_item\_gvkey)   
 if (nrow(specific\_security) > 0){  
 security\_row <- head(specific\_security, 1)  
 dvrate\_m <- as.numeric(security\_row$dvrate\_m)   
 fund\_stock\_securities\_ds$dvrate\_m[fund\_stock\_securities\_ds$gvkey == row\_item\_gvkey] <- dvrate\_m  
 }  
}  
  
  
#summary(fund\_stock\_securities\_ds)

### Load ratings dataset.

### Load this variable splticrm ==> S&P Domestic Long Term Issuer Credit Rating

### splticrm - This is categorical variable with unique ratings, each of the rating are given numeric values.

### Highest rating gets high numeric values and as rating decreases the numerica value assigned decreases.

ratings\_init\_ds <- read.csv("./data/Ratings\_DS.csv", na.strings=c("", " "))  
names(ratings\_init\_ds)[names(ratings\_init\_ds) == "ï..gvkey"] <- "gvkey"  
ratings\_init\_ds$datadate <- as.Date(ratings\_init\_ds$datadate, "%m/%d/%Y")  
ratings\_init\_ds$splticrm = factor(ratings\_init\_ds$splticrm, levels=c(levels(ratings\_init\_ds$splticrm), "NR"))  
ratings\_init\_ds$splticrm[is.na(ratings\_init\_ds$splticrm)] = "NR"  
  
ratings\_init\_ds$splticrm\_num\_value <- 0  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "AAA"] <- 100  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "AA"] <- 90  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "AA-"] <- 85  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "A+"] <- 80  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "A"] <- 75  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "A-"] <- 70  
  
  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BBB+"] <- 65  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BBB"] <- 60  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BBB-"] <- 55  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BB+"] <- 50  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BB"] <- 45  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "BB-"] <- 40  
  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "B+"] <- 35  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "B"] <- 30  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "B-"] <- 25  
  
  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "CCC+"] <- 20  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "CCC"] <- 19  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "CCC-"] <- 18  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "CC"] <- 17  
  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "D"] <- 10  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "SD"] <- 10  
ratings\_init\_ds$splticrm\_num\_value[ratings\_init\_ds$splticrm == "NR"] <- 0  
ratings\_init\_ds$splticrm\_num\_value <- factor(ratings\_init\_ds$splticrm\_num\_value)

### Merge the ratings dataset and fundamentals dataset, by assigning the rating from ratings dataset to each company using gvkey

### Along with rating, additional variable is added which indicates whether the rating has

### 1. Increased

### 2. Decreased

### 3. NOCHANGE

fund\_stock\_securities\_rating\_ds <- fund\_stock\_securities\_ds #%>%  
 #filter(gvkey == 1078)  
fund\_stock\_securities\_rating\_ds$sp\_rating <- "NOTRATED"   
rated\_companies <- ratings\_init\_ds %>%  
 filter(splticrm != "NR")  
  
for (row in 1:nrow(fund\_stock\_securities\_rating\_ds)){  
 row\_item\_gvkey <- as.integer(fund\_stock\_securities\_rating\_ds[row, "gvkey"])  
  
 specific\_rating <- rated\_companies %>%  
 filter(gvkey == row\_item\_gvkey) %>%  
 arrange(datadate)  
 if (nrow(specific\_rating) > 0){  
 first\_row <- head(specific\_rating, 1)  
 last\_row <- tail(specific\_rating, 1)  
 start\_value <- as.integer(first\_row$splticrm\_num\_value)  
 end\_value <- as.integer(last\_row$splticrm\_num\_value)  
 if (start\_value == end\_value){  
 fund\_stock\_securities\_rating\_ds$sp\_rating[fund\_stock\_securities\_rating\_ds$gvkey == row\_item\_gvkey] <- "NoCHANGE"  
 }else if (start\_value < end\_value){  
 fund\_stock\_securities\_rating\_ds$sp\_rating[fund\_stock\_securities\_rating\_ds$gvkey == row\_item\_gvkey] <- "INCREASED"  
 }else if (start\_value > end\_value){  
 fund\_stock\_securities\_rating\_ds$sp\_rating[fund\_stock\_securities\_rating\_ds$gvkey == row\_item\_gvkey] <- "DECREASED"  
 }  
 }  
}  
fund\_stock\_securities\_rating\_ds$sp\_rating <- factor(fund\_stock\_securities\_rating\_ds$sp\_rating)  
#summary(fund\_stock\_securities\_rating\_ds)

### Load Sca filing dataset for the target variable

sca\_fillings\_ds <- read.csv("./data/SCA\_Filings\_and\_Settlements.csv", na.strings=c(""," "))  
sca\_fillings\_ds$SettlementAmount = gsub("\\$", "", sca\_fillings\_ds$SettlementAmount)  
sca\_fillings\_ds$SettlementAmount = as.numeric(gsub("\\,", "", sca\_fillings\_ds$SettlementAmount))  
#summary(sca\_fillings\_ds)  
ncol(sca\_fillings\_ds)

## [1] 6

### Add the target variable litigated to the dataset under analysis.

### From each of the conany record identify if has any sca filing from the sca fileing dataset,if entry exists then mark for that company lititgated = true or else if the record ### does not exist in sca filing for that particular comapny then mark litigate attribute as false.

### Also in case of litigation indentify if there is any settlement amount and add same to main dataset. In case if multiple settlement amount exists for particular company then ### take the maximum settlement amount.

fund\_stock\_securities\_rating\_ds$litigated <- 0  
fund\_stock\_securities\_rating\_ds$litigation\_settlement <- NA  
fund\_stock\_securities\_rating\_ds\_1 <- fund\_stock\_securities\_rating\_ds   
  
for (row in 1:nrow(fund\_stock\_securities\_rating\_ds\_1)){  
 row\_item\_tic <- lapply(fund\_stock\_securities\_rating\_ds\_1[row, "tic"], as.character)  
 row\_item\_gvkey <- as.integer(fund\_stock\_securities\_rating\_ds[row, "gvkey"])  
 specific\_sca\_filings <- sca\_fillings\_ds %>%  
 filter(Ticker == row\_item\_tic)   
 if (nrow(specific\_sca\_filings) > 0){  
 fund\_stock\_securities\_rating\_ds$litigated[fund\_stock\_securities\_rating\_ds$gvkey == row\_item\_gvkey] <- 1  
 specific\_sca\_filings\_max <- specific\_sca\_filings %>%  
 filter(!is.na(SettlementAmount)) %>%  
 arrange(SettlementAmount)  
   
 if (nrow(specific\_sca\_filings\_max) > 0){  
 last\_row <- tail(specific\_sca\_filings\_max, 1)  
 settlement\_amount <- as.numeric(last\_row$SettlementAmount)  
 fund\_stock\_securities\_rating\_ds$litigation\_settlement[fund\_stock\_securities\_rating\_ds$gvkey == row\_item\_gvkey] <- settlement\_amount  
 }  
 }  
}  
fund\_stock\_securities\_rating\_ds$litigated <- as.factor(fund\_stock\_securities\_rating\_ds$litigated)  
#summary(fund\_stock\_securities\_rating\_ds)

### Based on manual review. idetify the columns that need to be removed from the final dataset.

fund\_stock\_securities\_rating\_ds\_final <- subset(fund\_stock\_securities\_rating\_ds, select = -c(aocidergl, aocipen, ceqt, cstkcv, dd1,  
 dpc,icapt, intan, intano, ivncf, ivst,  
 lo, lse, opeps, reajo, recta,  
 spi, tstkn, txp, txr,  
 restmt\_epsfi\_mag, restmt\_epsfi,  
 restmt\_pi, restmt\_pi\_mag,  
 restmt\_seq, restmt\_seq\_mag,  
 restmt\_xsga, restmt\_xsga\_mag,  
 restmt\_dvpsp\_f, restmt\_dvpsp\_f\_mag,   
 restmt\_dvpsx\_f, restmt\_dvpsx\_f\_mag  
 ))  
#summary(fund\_stock\_securities\_rating\_ds\_final)  
ncol(fund\_stock\_securities\_rating\_ds\_final)

## [1] 87

# DE ==> Debt to equity ratio

# wc ==> Working capital ratio

# pe ==> Pricing to earning ratio

# ROE ==> Return on Equity

fund\_stock\_securities\_rating\_ds\_final$epspi[fund\_stock\_securities\_rating\_ds\_final$epspi == 0] <- 0.000001  
fund\_stock\_securities\_rating\_ds\_final$lt[fund\_stock\_securities\_rating\_ds\_final$lt == 0] <- 0.000001  
fund\_stock\_securities\_rating\_ds\_final$teq[fund\_stock\_securities\_rating\_ds\_final$teq == 0] <- 0.000001  
fund\_stock\_securities\_rating\_ds\_final$pe\_ratio <- fund\_stock\_securities\_rating\_ds\_final$prccd\_m/fund\_stock\_securities\_rating\_ds\_final$epspi  
fund\_stock\_securities\_rating\_ds\_final$wc\_ratio <- fund\_stock\_securities\_rating\_ds\_final$act/fund\_stock\_securities\_rating\_ds\_final$lt  
fund\_stock\_securities\_rating\_ds\_final$de\_ratio <- fund\_stock\_securities\_rating\_ds\_final$lt/fund\_stock\_securities\_rating\_ds\_final$teq  
fund\_stock\_securities\_rating\_ds\_final$roe\_ratio <- fund\_stock\_securities\_rating\_ds\_final$ni/fund\_stock\_securities\_rating\_ds\_final$teq  
  
trfm\_median <- median(as.numeric(fund\_stock\_securities\_rating\_ds\_final$trfm\_m),na.rm=TRUE)  
fund\_stock\_securities\_rating\_ds\_final$trfm\_m[is.na(fund\_stock\_securities\_rating\_ds\_final$trfm\_m)] <- trfm\_median

### From the final dataset identify correlated variables

cor\_matrix\_ds <- subset(fund\_stock\_securities\_rating\_ds\_final, select = -c(gvkey,tic, sp\_rating, litigated))  
cor\_matrix <- cor(cor\_matrix\_ds)  
cor\_matrix %>%  
 as.data.frame() %>%  
 mutate(var1 = rownames(.)) %>%  
 gather(var2, value, -var1) %>%  
 arrange(desc(value)) %>%  
 group\_by(value) %>%  
 filter(row\_number() == 1)

## # A tibble: 3,572 x 3  
## # Groups: value [3,572]  
## var1 var2 value  
## <chr> <chr> <dbl>  
## 1 aco aco 1   
## 2 prchd\_m prccd\_m 1.00   
## 3 prcld\_m prccd\_m 1.00   
## 4 prcld\_m prchd\_m 0.998  
## 5 restmt\_ni\_mag restmt\_ib\_mag 0.998  
## 6 teq ceq 0.996  
## 7 ni ci 0.994  
## 8 oancf ebit 0.988  
## 9 revt cogs 0.988  
## 10 ppegt capx 0.988  
## # ... with 3,562 more rows

#### Remove Highly correlated variables, in general if correlation value is > 0.9 and check the correlation matrix values

cor\_matrix\_ds <- subset(fund\_stock\_securities\_rating\_ds\_final, select = -c(gvkey,tic, sp\_rating, litigated, prchd\_m, prcld\_m, ni, restmt\_ib\_mag, ceq, oancf, cogs, ppegt,  
 lt, ci, restmt\_dltt\_mag, invt, che, ap, at, xint, gp, act, txt, capx,  
 dm, dn, dpact,fiao,fincf,sppiv,fopo)) #latest removed variables  
cor\_matrix <- cor(cor\_matrix\_ds)  
cor\_matrix %>%  
 as.data.frame() %>%  
 mutate(var1 = rownames(.)) %>%  
 gather(var2, value, -var1) %>%  
 arrange(desc(value)) %>%  
 group\_by(value) %>%  
 filter(row\_number() == 1)

## # A tibble: 1,655 x 3  
## # Groups: value [1,655]  
## var1 var2 value  
## <chr> <chr> <dbl>  
## 1 aco aco 1   
## 2 prccd\_m epspi 0.945  
## 3 ebit dvt 0.945  
## 4 restmt\_teq\_mag restmt\_ni\_mag 0.934  
## 5 teq ebit 0.892  
## 6 ebit dltt 0.892  
## 7 teq ao 0.879  
## 8 prcod\_m prccd\_m 0.879  
## 9 ebit ao 0.874  
## 10 re ebit 0.859  
## # ... with 1,645 more rows

### Remove Highly correlated variables from above analysis.

fund\_stock\_securities\_rating\_ds\_final\_10 <- subset(fund\_stock\_securities\_rating\_ds\_final, select = -c(prchd\_m, prcld\_m, ni, restmt\_ib\_mag,   
 ceq, oancf, cogs, ppegt,  
 lt, ci, restmt\_dltt\_mag, invt, che,   
 ap, at, xint, gp, act, txt, capx,  
 dm, dn, dpact,fiao,fincf,sppiv,fopo)) #latest removed variables

### Convert all categorical variables to factor

ds\_final <- fund\_stock\_securities\_rating\_ds\_final\_10  
ds\_final <- subset(ds\_final, select = -c(dvrate\_m)) #, litigation\_settlement))  
ds\_final$restmt\_at <- as.factor(ds\_final$restmt\_at)  
ds\_final$restmt\_capx <- as.factor(ds\_final$restmt\_capx)  
ds\_final$restmt\_cogs <- as.factor(ds\_final$restmt\_cogs)  
ds\_final$restmt\_dltt <- as.factor(ds\_final$restmt\_dltt)  
ds\_final$restmt\_epspi <- as.factor(ds\_final$restmt\_epspi)  
ds\_final$restmt\_ib <- as.factor(ds\_final$restmt\_ib)  
ds\_final$restmt\_ni <- as.factor(ds\_final$restmt\_ni)  
ds\_final$restmt\_nopi <- as.factor(ds\_final$restmt\_nopi)  
ds\_final$restmt\_reuna <- as.factor(ds\_final$restmt\_reuna)  
ds\_final$restmt\_teq <- as.factor(ds\_final$restmt\_teq)  
ds\_final$restmt\_txt <- as.factor(ds\_final$restmt\_txt)  
ds\_final$restmt\_wcap <- as.factor(ds\_final$restmt\_wcap)  
ds\_final$restmt\_xint <- as.factor(ds\_final$restmt\_xint)  
ds\_final$sp\_rating <- as.factor(ds\_final$sp\_rating)  
ds\_final$litigated <- as.factor(ds\_final$litigated)

### Perform target ecoding to all the categorical variables

# split the data into training and (held-out) test sets  
training\_ind <- createDataPartition(ds\_final$litigated,  
p = 0.75,  
list = FALSE,  
times = 1)  
training\_set <- ds\_final[training\_ind, ]  
test\_set <- ds\_final[-training\_ind, ]  
  
nrow(training\_set)

## [1] 250

nrow(test\_set)

## [1] 83

threshold <- 250  
  
  
threshold <- 250  
target\_enc\_train <- function(variable, level) {  
 training\_set$litigated <- as.numeric(as.vector(training\_set$litigated))  
 train\_avg\_target <- colMeans(training\_set[, "litigated"])  
 if (nrow(training\_set[training\_set[, variable]==level, ])==0) {  
 return(train\_avg\_target)  
 } else {  
 level\_num\_obs <- nrow(training\_set[training\_set[, variable]==level,])  
 level\_avg\_target <- colMeans(training\_set[training\_set[, variable]==level, "litigated"])  
 return((level\_num\_obs\*level\_avg\_target+threshold\*train\_avg\_target)/(level\_num\_obs+threshold))  
 }  
}  
  
sp\_rating\_target <- mapply(target\_enc\_train, variable = "sp\_rating", level = levels(training\_set$sp\_rating), USE.NAMES = FALSE)  
names(sp\_rating\_target) <- levels(training\_set$sp\_rating)  
training\_set$sp\_rating\_target <- 0  
for (level in levels(training\_set$sp\_rating)) {  
 training\_set[training\_set[, "sp\_rating"]==level, "sp\_rating\_target"] <- sp\_rating\_target[level]  
}  
  
test\_set$sp\_rating\_target <- 0  
for (level in levels(training\_set$sp\_rating)) {  
 test\_set[test\_set[, "sp\_rating"]==level, "sp\_rating\_target"] <- sp\_rating\_target[level]  
}  
  
  
restmt\_at\_target <- mapply(target\_enc\_train, variable = "restmt\_at", level = levels(training\_set$restmt\_at), USE.NAMES = FALSE)  
names(restmt\_at\_target) <- levels(training\_set$restmt\_at)  
training\_set$restmt\_at\_target <- 0  
for (level in levels(training\_set$restmt\_at)) {  
 training\_set[training\_set[, "restmt\_at"]==level, "restmt\_at\_target"] <- restmt\_at\_target[level]  
}  
  
test\_set$restmt\_at\_target <- 0  
for (level in levels(training\_set$restmt\_at)) {  
 test\_set[test\_set[, "restmt\_at"]==level, "restmt\_at\_target"] <- restmt\_at\_target[level]  
}  
  
restmt\_capx\_target <- mapply(target\_enc\_train, variable = "restmt\_capx", level = levels(training\_set$restmt\_capx), USE.NAMES = FALSE)  
names(restmt\_capx\_target) <- levels(training\_set$restmt\_capx)  
training\_set$restmt\_capx\_target <- 0  
for (level in levels(training\_set$restmt\_capx)) {  
 training\_set[training\_set[, "restmt\_capx"]==level, "restmt\_capx\_target"] <- restmt\_capx\_target[level]  
}  
  
test\_set$restmt\_capx\_target <- 0  
for (level in levels(training\_set$restmt\_capx)) {  
 test\_set[test\_set[, "restmt\_capx"]==level, "restmt\_capx\_target"] <- restmt\_capx\_target[level]  
}  
  
restmt\_cogs\_target <- mapply(target\_enc\_train, variable = "restmt\_cogs", level = levels(training\_set$restmt\_cogs), USE.NAMES = FALSE)  
names(restmt\_cogs\_target) <- levels(training\_set$restmt\_cogs)  
training\_set$restmt\_cogs\_target <- 0  
for (level in levels(training\_set$restmt\_cogs)) {  
 training\_set[training\_set[, "restmt\_cogs"]==level, "restmt\_cogs\_target"] <- restmt\_cogs\_target[level]  
}  
  
test\_set$restmt\_cogs\_target <- 0  
for (level in levels(training\_set$restmt\_cogs)) {  
 test\_set[test\_set[, "restmt\_cogs"]==level, "restmt\_cogs\_target"] <- restmt\_cogs\_target[level]  
}  
  
restmt\_dltt\_target <- mapply(target\_enc\_train, variable = "restmt\_dltt", level = levels(training\_set$restmt\_dltt), USE.NAMES = FALSE)  
names(restmt\_dltt\_target) <- levels(training\_set$restmt\_dltt)  
training\_set$restmt\_dltt\_target <- 0  
for (level in levels(training\_set$restmt\_dltt)) {  
 training\_set[training\_set[, "restmt\_dltt"]==level, "restmt\_dltt\_target"] <- restmt\_dltt\_target[level]  
}  
  
test\_set$restmt\_dltt\_target <- 0  
for (level in levels(training\_set$restmt\_dltt)) {  
 test\_set[test\_set[, "restmt\_dltt"]==level, "restmt\_dltt\_target"] <- restmt\_dltt\_target[level]  
}  
  
restmt\_epspi\_target <- mapply(target\_enc\_train, variable = "restmt\_epspi", level = levels(training\_set$restmt\_epspi), USE.NAMES = FALSE)  
names(restmt\_epspi\_target) <- levels(training\_set$restmt\_epspi)  
training\_set$restmt\_epspi\_target <- 0  
for (level in levels(training\_set$restmt\_epspi)) {  
 training\_set[training\_set[, "restmt\_epspi"]==level, "restmt\_epspi\_target"] <- restmt\_epspi\_target[level]  
}  
  
test\_set$restmt\_epspi\_target <- 0  
for (level in levels(training\_set$restmt\_epspi)) {  
 test\_set[test\_set[, "restmt\_epspi"]==level, "restmt\_epspi\_target"] <- restmt\_epspi\_target[level]  
}  
  
restmt\_ib\_target <- mapply(target\_enc\_train, variable = "restmt\_ib", level = levels(training\_set$restmt\_ib), USE.NAMES = FALSE)  
names(restmt\_ib\_target) <- levels(training\_set$restmt\_ib)  
training\_set$restmt\_ib\_target <- 0  
for (level in levels(training\_set$restmt\_ib)) {  
 training\_set[training\_set[, "restmt\_ib"]==level, "restmt\_ib\_target"] <- restmt\_ib\_target[level]  
}  
  
test\_set$restmt\_ib\_target <- 0  
for (level in levels(training\_set$restmt\_ib)) {  
 test\_set[test\_set[, "restmt\_ib"]==level, "restmt\_ib\_target"] <- restmt\_ib\_target[level]  
}  
  
  
restmt\_ni\_target <- mapply(target\_enc\_train, variable = "restmt\_ni", level = levels(training\_set$restmt\_ni), USE.NAMES = FALSE)  
names(restmt\_ni\_target) <- levels(training\_set$restmt\_ni)  
training\_set$restmt\_ni\_target <- 0  
for (level in levels(training\_set$restmt\_ni)) {  
 training\_set[training\_set[, "restmt\_ni"]==level, "restmt\_ni\_target"] <- restmt\_ni\_target[level]  
}  
  
test\_set$restmt\_ni\_target <- 0  
for (level in levels(training\_set$restmt\_ni)) {  
 test\_set[test\_set[, "restmt\_ni"]==level, "restmt\_ni\_target"] <- restmt\_ni\_target[level]  
}  
  
restmt\_nopi\_target <- mapply(target\_enc\_train, variable = "restmt\_nopi", level = levels(training\_set$restmt\_nopi), USE.NAMES = FALSE)  
names(restmt\_nopi\_target) <- levels(training\_set$restmt\_nopi)  
training\_set$restmt\_nopi\_target <- 0  
for (level in levels(training\_set$restmt\_nopi)) {  
 training\_set[training\_set[, "restmt\_nopi"]==level, "restmt\_nopi\_target"] <- restmt\_nopi\_target[level]  
}  
  
test\_set$restmt\_nopi\_target <- 0  
for (level in levels(training\_set$restmt\_nopi)) {  
 test\_set[test\_set[, "restmt\_nopi"]==level, "restmt\_nopi\_target"] <- restmt\_nopi\_target[level]  
}  
  
  
restmt\_reuna\_target <- mapply(target\_enc\_train, variable = "restmt\_reuna", level = levels(training\_set$restmt\_reuna), USE.NAMES = FALSE)  
names(restmt\_reuna\_target) <- levels(training\_set$restmt\_reuna)  
training\_set$restmt\_reuna\_target <- 0  
for (level in levels(training\_set$restmt\_reuna)) {  
 training\_set[training\_set[, "restmt\_reuna"]==level, "restmt\_reuna\_target"] <- restmt\_reuna\_target[level]  
}  
  
test\_set$restmt\_reuna\_target <- 0  
for (level in levels(training\_set$restmt\_reuna)) {  
 test\_set[test\_set[, "restmt\_reuna"]==level, "restmt\_reuna\_target"] <- restmt\_reuna\_target[level]  
}  
  
restmt\_teq\_target <- mapply(target\_enc\_train, variable = "restmt\_teq", level = levels(training\_set$restmt\_teq), USE.NAMES = FALSE)  
names(restmt\_teq\_target) <- levels(training\_set$restmt\_teq)  
training\_set$restmt\_teq\_target <- 0  
for (level in levels(training\_set$restmt\_teq)) {  
 training\_set[training\_set[, "restmt\_teq"]==level, "restmt\_teq\_target"] <- restmt\_teq\_target[level]  
}  
  
test\_set$restmt\_teq\_target <- 0  
for (level in levels(training\_set$restmt\_teq)) {  
 test\_set[test\_set[, "restmt\_teq"]==level, "restmt\_teq\_target"] <- restmt\_teq\_target[level]  
}  
  
restmt\_txt\_target <- mapply(target\_enc\_train, variable = "restmt\_txt", level = levels(training\_set$restmt\_txt), USE.NAMES = FALSE)  
names(restmt\_txt\_target) <- levels(training\_set$restmt\_txt)  
training\_set$restmt\_txt\_target <- 0  
for (level in levels(training\_set$restmt\_txt)) {  
 training\_set[training\_set[, "restmt\_txt"]==level, "restmt\_txt\_target"] <- restmt\_txt\_target[level]  
}  
  
test\_set$restmt\_txt\_target <- 0  
for (level in levels(training\_set$restmt\_txt)) {  
 test\_set[test\_set[, "restmt\_txt"]==level, "restmt\_txt\_target"] <- restmt\_txt\_target[level]  
}  
  
restmt\_wcap\_target <- mapply(target\_enc\_train, variable = "restmt\_wcap", level = levels(training\_set$restmt\_wcap), USE.NAMES = FALSE)  
names(restmt\_wcap\_target) <- levels(training\_set$restmt\_wcap)  
training\_set$restmt\_wcap\_target <- 0  
for (level in levels(training\_set$restmt\_wcap)) {  
 training\_set[training\_set[, "restmt\_wcap"]==level, "restmt\_wcap\_target"] <- restmt\_wcap\_target[level]  
}  
  
test\_set$restmt\_wcap\_target <- 0  
for (level in levels(training\_set$restmt\_wcap)) {  
 test\_set[test\_set[, "restmt\_wcap"]==level, "restmt\_wcap\_target"] <- restmt\_wcap\_target[level]  
}  
  
restmt\_xint\_target <- mapply(target\_enc\_train, variable = "restmt\_xint", level = levels(training\_set$restmt\_xint), USE.NAMES = FALSE)  
names(restmt\_xint\_target) <- levels(training\_set$restmt\_xint)  
training\_set$restmt\_xint\_target <- 0  
for (level in levels(training\_set$restmt\_xint)) {  
 training\_set[training\_set[, "restmt\_xint"]==level, "restmt\_xint\_target"] <- restmt\_xint\_target[level]  
}  
  
test\_set$restmt\_xint\_target <- 0  
for (level in levels(training\_set$restmt\_xint)) {  
 test\_set[test\_set[, "restmt\_xint"]==level, "restmt\_xint\_target"] <- restmt\_xint\_target[level]  
}

nrow(training\_set)

## [1] 250

nrow(test\_set)

## [1] 83

target\_company\_row <- training\_set %>%   
 filter(gvkey == target\_company\_gv\_key)  
  
if (nrow(target\_company\_row) > 0) {  
 training\_set <- subset(training\_set, gvkey != target\_company\_gv\_key)  
} else {  
   
 target\_company\_row <- test\_set %>%   
 filter(gvkey == target\_company\_gv\_key)  
 test\_set <- subset(test\_set, gvkey != target\_company\_gv\_key)  
}  
  
nrow(training\_set)

## [1] 249

nrow(test\_set)

## [1] 83

target\_company\_row

## Warning: `...` is not empty.  
##   
## We detected these problematic arguments:  
## \* `needs\_dots`  
##   
## These dots only exist to allow future extensions and should be empty.  
## Did you misspecify an argument?

## # A tibble: 1 x 77  
## # Groups: gvkey [1]  
## gvkey tic aco acominc ao aoloch aqc bkvlps caps ch chech csho  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9777 SJM 93.3 -45.8 111. -7.08 168. 47.1 4262. 249. -60.6 110.  
## # ... with 65 more variables: cstk <dbl>, dlc <dbl>, dltt <dbl>, dvt <dbl>,  
## # ebit <dbl>, epspi <dbl>, nopi <dbl>, re <dbl>, rect <dbl>, revt <dbl>,  
## # siv <dbl>, sstk <dbl>, teq <dbl>, tstk <dbl>, wcap <dbl>, restmt\_at <fct>,  
## # restmt\_at\_mag <dbl>, restmt\_capx <fct>, restmt\_capx\_mag <dbl>,  
## # restmt\_cogs <fct>, restmt\_cogs\_mag <dbl>, restmt\_dltt <fct>,  
## # restmt\_epspi <fct>, restmt\_epspi\_mag <dbl>, restmt\_ib <fct>,  
## # restmt\_ni <fct>, restmt\_ni\_mag <dbl>, restmt\_nopi <fct>,  
## # restmt\_nopi\_mag <dbl>, restmt\_reuna <fct>, restmt\_reuna\_mag <dbl>,  
## # restmt\_teq <fct>, restmt\_teq\_mag <dbl>, restmt\_txt <fct>,  
## # restmt\_txt\_mag <dbl>, restmt\_wcap <fct>, restmt\_wcap\_mag <dbl>,  
## # restmt\_xint <fct>, restmt\_xint\_mag <dbl>, cshtrd\_m <dbl>, prccd\_m <dbl>,  
## # prcod\_m <dbl>, trfd\_m <dbl>, trfm\_m <dbl>, sp\_rating <fct>,  
## # litigated <fct>, litigation\_settlement <dbl>, pe\_ratio <dbl>,  
## # wc\_ratio <dbl>, de\_ratio <dbl>, roe\_ratio <dbl>, sp\_rating\_target <dbl>,  
## # restmt\_at\_target <dbl>, restmt\_capx\_target <dbl>, restmt\_cogs\_target <dbl>,  
## # restmt\_dltt\_target <dbl>, restmt\_epspi\_target <dbl>,  
## # restmt\_ib\_target <dbl>, restmt\_ni\_target <dbl>, restmt\_nopi\_target <dbl>,  
## # restmt\_reuna\_target <dbl>, restmt\_teq\_target <dbl>,  
## # restmt\_txt\_target <dbl>, restmt\_wcap\_target <dbl>, restmt\_xint\_target <dbl>

### Remove the restatement variables which have near zero variance

training\_subset\_ds\_final\_1 <- subset(training\_set, select = -c(sp\_rating,  
 restmt\_at,  
 restmt\_capx,  
 restmt\_cogs,  
 restmt\_dltt,  
 restmt\_epspi,  
 restmt\_ib,  
 restmt\_ni,  
 restmt\_nopi,  
 restmt\_reuna,  
 restmt\_teq,  
 restmt\_txt,  
 restmt\_wcap,  
 restmt\_xint))  
  
  
test\_subset\_ds\_final\_1 <- subset(test\_set, select = -c(sp\_rating,  
 restmt\_at,  
 restmt\_capx,  
 restmt\_cogs,  
 restmt\_dltt,  
 restmt\_epspi,  
 restmt\_ib,  
 restmt\_ni,  
 restmt\_nopi,  
 restmt\_reuna,  
 restmt\_teq,  
 restmt\_txt,  
 restmt\_wcap,  
 restmt\_xint))  
  
target\_company\_row\_1 <- subset(target\_company\_row, select = -c(sp\_rating,  
 restmt\_at,  
 restmt\_capx,  
 restmt\_cogs,  
 restmt\_dltt,  
 restmt\_epspi,  
 restmt\_ib,  
 restmt\_ni,  
 restmt\_nopi,  
 restmt\_reuna,  
 restmt\_teq,  
 restmt\_txt,  
 restmt\_wcap,  
 restmt\_xint))  
  
  
training\_subset\_ds\_final\_1 <- training\_subset\_ds\_final\_1 %>%  
 relocate(litigated, .after = last\_col())  
  
test\_subset\_ds\_final\_1 <- test\_subset\_ds\_final\_1 %>%  
 relocate(litigated, .after = last\_col())  
  
target\_company\_row\_final <- target\_company\_row\_1 %>%  
 relocate(litigated, .after = last\_col())  
  
  
training\_subset\_ds\_final\_1 <- training\_subset\_ds\_final\_1 %>%  
 relocate(litigation\_settlement , .after = last\_col())  
  
test\_subset\_ds\_final\_1 <- test\_subset\_ds\_final\_1 %>%  
 relocate(litigation\_settlement, .after = last\_col())  
  
target\_company\_row\_final <- target\_company\_row\_final %>%  
 relocate(litigation\_settlement, .after = last\_col())  
  
training\_subset\_ds\_final <- training\_subset\_ds\_final\_1  
test\_subset\_ds\_final <- test\_subset\_ds\_final\_1  
  
ncol(training\_subset\_ds\_final)

## [1] 63

summary(training\_subset\_ds\_final)

## gvkey tic aco acominc   
## Min. : 1266 0161A : 1 Min. : 0.000 Min. :-19306.57   
## 1st Qu.: 12575 0170A : 1 1st Qu.: 0.351 1st Qu.: -24.15   
## Median : 31392 0173A : 1 Median : 8.138 Median : 0.00   
## Mean : 80970 3AHII : 1 Mean : 175.864 Mean : -187.26   
## 3rd Qu.:164700 3AIRW : 1 3rd Qu.: 93.742 3rd Qu.: 0.00   
## Max. :264393 3CAGZ : 1 Max. :4706.135 Max. : 3495.34   
## (Other):243   
## ao aoloch aqc bkvlps   
## Min. : 0.000 Min. :-465.250 Min. : -12.45 Min. : -82.3   
## 1st Qu.: 0.075 1st Qu.: -1.647 1st Qu.: 0.00 1st Qu.: 0.1   
## Median : 7.711 Median : 0.000 Median : 0.00 Median : 3.6   
## Mean : 180.350 Mean : 4.895 Mean : 97.10 Mean : 8339.9   
## 3rd Qu.: 91.925 3rd Qu.: 1.500 3rd Qu.: 12.32 3rd Qu.: 12.6   
## Max. :5132.600 Max. : 655.000 Max. :5559.02 Max. :1216746.9   
##   
## caps ch chech csho   
## Min. : -701.475 Min. : 0.000 Min. :-305.7500 Min. : 0.52   
## 1st Qu.: 6.498 1st Qu.: 1.202 1st Qu.: -0.1535 1st Qu.: 18.88   
## Median : 42.415 Median : 18.018 Median : 0.4822 Median : 52.49   
## Mean : 706.132 Mean : 363.447 Mean : 39.8367 Mean : 249.12   
## 3rd Qu.: 412.843 3rd Qu.: 181.018 3rd Qu.: 10.6358 3rd Qu.: 144.58   
## Max. :28658.250 Max. :7382.800 Max. :1358.0000 Max. :6252.56   
##   
## cstk dlc dltt dvt   
## Min. : 0.000 Min. : 0.000 Min. : 0.00 Min. : -0.006   
## 1st Qu.: 0.035 1st Qu.: 0.269 1st Qu.: 0.03 1st Qu.: 0.000   
## Median : 0.264 Median : 4.004 Median : 14.73 Median : 0.000   
## Mean : 196.959 Mean : 311.379 Mean : 1309.06 Mean : 203.163   
## 3rd Qu.: 12.908 3rd Qu.: 101.250 3rd Qu.: 929.11 3rd Qu.: 46.679   
## Max. :7290.750 Max. :15926.126 Max. :42659.60 Max. :6572.535   
##   
## ebit epspi nopi re   
## Min. : -208.760 Min. :-14.0200 Min. : -38.5478 Min. :-7570.29   
## 1st Qu.: -0.366 1st Qu.: -0.0550 1st Qu.: -0.0013 1st Qu.: -11.17   
## Median : 17.853 Median : 0.2425 Median : 0.1790 Median : 18.87   
## Mean : 700.042 Mean : 1.7230 Mean : 43.4463 Mean : 1704.11   
## 3rd Qu.: 351.466 3rd Qu.: 1.8200 3rd Qu.: 4.8362 3rd Qu.: 411.00   
## Max. :24345.400 Max. :230.7025 Max. :2224.4000 Max. :68884.60   
##   
## rect revt siv sstk   
## Min. : 0.000 Min. : 0.0 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 1.479 1st Qu.: 16.1 1st Qu.: 0.0000 1st Qu.: 0.0012   
## Median : 23.143 Median : 308.6 Median : 0.0000 Median : 0.9955   
## Mean : 499.237 Mean : 7936.7 Mean : 34.1581 Mean : 37.3398   
## 3rd Qu.: 343.470 3rd Qu.: 3793.0 3rd Qu.: 0.3503 3rd Qu.: 13.3750   
## Max. :15020.067 Max. :442511.4 Max. :1622.0000 Max. :1057.0000   
##   
## teq tstk wcap restmt\_at\_mag   
## Min. :-2208.96 Min. : 0.000 Min. :-8236.800 Min. :-1.011750   
## 1st Qu.: 3.38 1st Qu.: 0.000 1st Qu.: -0.014 1st Qu.: 0.000000   
## Median : 103.36 Median : 0.000 Median : 22.957 Median : 0.000000   
## Mean : 2434.16 Mean : 591.651 Mean : 263.686 Mean : 0.001972   
## 3rd Qu.: 1118.67 3rd Qu.: 8.386 3rd Qu.: 288.837 3rd Qu.: 0.000000   
## Max. :76602.80 Max. :25036.250 Max. :12261.750 Max. : 1.275750   
##   
## restmt\_capx\_mag restmt\_cogs\_mag restmt\_epspi\_mag restmt\_ni\_mag   
## Min. :-22.7162 Min. :-50.0000 Min. : -50.0 Min. : -6.611   
## 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 0.0 1st Qu.: 0.000   
## Median : 0.0000 Median : 0.0000 Median : 0.0 Median : 0.000   
## Mean : -0.1057 Mean : 0.4436 Mean : 424.5 Mean : 10.776   
## 3rd Qu.: 0.0000 3rd Qu.: 0.0000 3rd Qu.: 0.0 3rd Qu.: 0.000   
## Max. : 8.3335 Max. :100.0000 Max. :77081.7 Max. :2683.890   
##   
## restmt\_nopi\_mag restmt\_reuna\_mag restmt\_teq\_mag restmt\_txt\_mag   
## Min. :-1868600.0 Min. : -85.20 Min. : -105.39 Min. :-88.7704   
## 1st Qu.: -77.3 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.0000   
## Median : 0.0 Median : 0.00 Median : 0.00 Median : 0.0000   
## Mean : -7977.9 Mean : 16.96 Mean : 49.75 Mean : -0.9797   
## 3rd Qu.: 15.1 3rd Qu.: 0.00 3rd Qu.: 0.00 3rd Qu.: 0.0000   
## Max. : 68865.1 Max. :4181.70 Max. :12541.75 Max. : 17.8622   
##   
## restmt\_wcap\_mag restmt\_xint\_mag cshtrd\_m prccd\_m   
## Min. : -3.149 Min. :-62.735 Min. : 0 Min. : 0.0018   
## 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 18025 1st Qu.: 1.1177   
## Median : 0.000 Median : 0.000 Median : 111926 Median : 8.6464   
## Mean : 1.642 Mean : -1.035 Mean : 872188 Mean : 32.3903   
## 3rd Qu.: 0.000 3rd Qu.: 0.000 3rd Qu.: 614194 3rd Qu.: 30.4042   
## Max. :412.500 Max. : 0.562 Max. :13129451 Max. :2217.8253   
##   
## prcod\_m trfd\_m trfm\_m pe\_ratio   
## Min. : 0.0362 Min. : 1.000 Min. : 1.000 Min. : -1220   
## 1st Qu.: 3.1852 1st Qu.: 1.063 1st Qu.: 1.000 1st Qu.: -5   
## Median : 11.3376 Median : 1.227 Median : 1.000 Median : 12   
## Mean : 40.1783 Mean : 3.017 Mean : 2.007 Mean : 504488   
## 3rd Qu.: 34.8010 3rd Qu.: 1.827 3rd Qu.: 1.475 3rd Qu.: 22   
## Max. :2217.8208 Max. :218.416 Max. :34.527 Max. :76903023   
##   
## wc\_ratio de\_ratio roe\_ratio sp\_rating\_target  
## Min. : 0.0000 Min. :-60.4827 Min. :-192.20000 Min. :0.1226   
## 1st Qu.: 0.3719 1st Qu.: 0.2035 1st Qu.: -0.00117 1st Qu.:0.1226   
## Median : 0.7205 Median : 0.7431 Median : 0.11160 Median :0.1226   
## Mean : 1.2785 Mean : 0.8113 Mean : -0.68051 Mean :0.1292   
## 3rd Qu.: 1.3596 3rd Qu.: 1.4790 3rd Qu.: 0.25771 3rd Qu.:0.1343   
## Max. :12.4471 Max. : 80.2000 Max. : 10.15006 Max. :0.1483   
##   
## restmt\_at\_target restmt\_capx\_target restmt\_cogs\_target restmt\_dltt\_target  
## Min. :0.1278 Min. :0.1341 Min. :0.1311 Min. :0.1354   
## 1st Qu.:0.1278 1st Qu.:0.1341 1st Qu.:0.1311 1st Qu.:0.1354   
## Median :0.1278 Median :0.1341 Median :0.1311 Median :0.1354   
## Mean :0.1292 Mean :0.1343 Mean :0.1345 Mean :0.1354   
## 3rd Qu.:0.1278 3rd Qu.:0.1341 3rd Qu.:0.1424 3rd Qu.:0.1354   
## Max. :0.1509 Max. :0.1395 Max. :0.1424 Max. :0.1373   
##   
## restmt\_epspi\_target restmt\_ib\_target restmt\_ni\_target restmt\_nopi\_target  
## Min. :0.1258 Min. :0.1250 Min. :0.1263 Min. :0.1279   
## 1st Qu.:0.1258 1st Qu.:0.1250 1st Qu.:0.1263 1st Qu.:0.1279   
## Median :0.1258 Median :0.1250 Median :0.1263 Median :0.1429   
## Mean :0.1292 Mean :0.1283 Mean :0.1273 Mean :0.1372   
## 3rd Qu.:0.1258 3rd Qu.:0.1250 3rd Qu.:0.1263 3rd Qu.:0.1429   
## Max. :0.1530 Max. :0.1547 Max. :0.1544 Max. :0.1429   
##   
## restmt\_reuna\_target restmt\_teq\_target restmt\_txt\_target restmt\_wcap\_target  
## Min. :0.1286 Min. :0.1273 Min. :0.1266 Min. :0.1324   
## 1st Qu.:0.1286 1st Qu.:0.1273 1st Qu.:0.1266 1st Qu.:0.1324   
## Median :0.1286 Median :0.1273 Median :0.1266 Median :0.1324   
## Mean :0.1301 Mean :0.1294 Mean :0.1285 Mean :0.1328   
## 3rd Qu.:0.1286 3rd Qu.:0.1273 3rd Qu.:0.1266 3rd Qu.:0.1324   
## Max. :0.1493 Max. :0.1513 Max. :0.1530 Max. :0.1429   
##   
## restmt\_xint\_target litigated litigation\_settlement  
## Min. :0.1303 0:215 Min. : 1500000   
## 1st Qu.:0.1303 1: 34 1st Qu.: 3362500   
## Median :0.1303 Median : 6375000   
## Mean :0.1322 Mean :10993750   
## 3rd Qu.:0.1303 3rd Qu.: 9562500   
## Max. :0.1454 Max. :47500000   
## NA's :241

target\_company\_row\_final

## # A tibble: 1 x 63  
## # Groups: gvkey [1]  
## gvkey tic aco acominc ao aoloch aqc bkvlps caps ch chech csho  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9777 SJM 93.3 -45.8 111. -7.08 168. 47.1 4262. 249. -60.6 110.  
## # ... with 51 more variables: cstk <dbl>, dlc <dbl>, dltt <dbl>, dvt <dbl>,  
## # ebit <dbl>, epspi <dbl>, nopi <dbl>, re <dbl>, rect <dbl>, revt <dbl>,  
## # siv <dbl>, sstk <dbl>, teq <dbl>, tstk <dbl>, wcap <dbl>,  
## # restmt\_at\_mag <dbl>, restmt\_capx\_mag <dbl>, restmt\_cogs\_mag <dbl>,  
## # restmt\_epspi\_mag <dbl>, restmt\_ni\_mag <dbl>, restmt\_nopi\_mag <dbl>,  
## # restmt\_reuna\_mag <dbl>, restmt\_teq\_mag <dbl>, restmt\_txt\_mag <dbl>,  
## # restmt\_wcap\_mag <dbl>, restmt\_xint\_mag <dbl>, cshtrd\_m <dbl>,  
## # prccd\_m <dbl>, prcod\_m <dbl>, trfd\_m <dbl>, trfm\_m <dbl>, pe\_ratio <dbl>,  
## # wc\_ratio <dbl>, de\_ratio <dbl>, roe\_ratio <dbl>, sp\_rating\_target <dbl>,  
## # restmt\_at\_target <dbl>, restmt\_capx\_target <dbl>, restmt\_cogs\_target <dbl>,  
## # restmt\_dltt\_target <dbl>, restmt\_epspi\_target <dbl>,  
## # restmt\_ib\_target <dbl>, restmt\_ni\_target <dbl>, restmt\_nopi\_target <dbl>,  
## # restmt\_reuna\_target <dbl>, restmt\_teq\_target <dbl>,  
## # restmt\_txt\_target <dbl>, restmt\_wcap\_target <dbl>,  
## # restmt\_xint\_target <dbl>, litigated <fct>, litigation\_settlement <dbl>

training\_subset\_ds\_final\_settlement <- training\_subset\_ds\_final  
test\_subset\_ds\_final\_settlement <- test\_subset\_ds\_final

output\_ds <- rbind(training\_subset\_ds\_final, test\_subset\_ds\_final)  
write.csv(output\_ds, file = "data/litigation\_classification\_ds.csv", row.names=FALSE)

### Scale the variables.

num\_var\_start\_index <- 3  
num\_var\_end\_index <- ncol(training\_subset\_ds\_final) - 2  
target\_var\_index <- ncol(training\_subset\_ds\_final) - 1  
  
num\_var\_start\_index

## [1] 3

num\_var\_end\_index

## [1] 61

target\_var\_index

## [1] 62

test\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index] <- scale(test\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index],   
 center = apply(training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index], 2, mean),   
 scale = apply(training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index], 2, sd))  
  
#target\_company\_row\_final  
target\_company\_row\_final[, num\_var\_start\_index:num\_var\_end\_index] = scale(target\_company\_row\_final[, num\_var\_start\_index:num\_var\_end\_index],   
 center = apply(training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index], 2, mean),   
 scale = apply(training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index], 2, sd))  
  
training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index] <- scale(training\_subset\_ds\_final[, num\_var\_start\_index:num\_var\_end\_index])  
  
levels(training\_subset\_ds\_final$litigated)[levels(training\_subset\_ds\_final$litigated) == 1] <- "Yes"  
levels(training\_subset\_ds\_final$litigated)[levels(training\_subset\_ds\_final$litigated) == 0] <- "No"  
  
target\_company\_row\_final

## # A tibble: 1 x 63  
## # Groups: gvkey [1]  
## gvkey tic aco acominc ao aoloch aqc bkvlps caps ch chech  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9777 SJM -0.155 0.107 -0.142 -0.121 0.174 -0.0885 1.54 -0.118 -0.649  
## # ... with 52 more variables: csho <dbl>, cstk <dbl>, dlc <dbl>, dltt <dbl>,  
## # dvt <dbl>, ebit <dbl>, epspi <dbl>, nopi <dbl>, re <dbl>, rect <dbl>,  
## # revt <dbl>, siv <dbl>, sstk <dbl>, teq <dbl>, tstk <dbl>, wcap <dbl>,  
## # restmt\_at\_mag <dbl>, restmt\_capx\_mag <dbl>, restmt\_cogs\_mag <dbl>,  
## # restmt\_epspi\_mag <dbl>, restmt\_ni\_mag <dbl>, restmt\_nopi\_mag <dbl>,  
## # restmt\_reuna\_mag <dbl>, restmt\_teq\_mag <dbl>, restmt\_txt\_mag <dbl>,  
## # restmt\_wcap\_mag <dbl>, restmt\_xint\_mag <dbl>, cshtrd\_m <dbl>,  
## # prccd\_m <dbl>, prcod\_m <dbl>, trfd\_m <dbl>, trfm\_m <dbl>, pe\_ratio <dbl>,  
## # wc\_ratio <dbl>, de\_ratio <dbl>, roe\_ratio <dbl>, sp\_rating\_target <dbl>,  
## # restmt\_at\_target <dbl>, restmt\_capx\_target <dbl>, restmt\_cogs\_target <dbl>,  
## # restmt\_dltt\_target <dbl>, restmt\_epspi\_target <dbl>,  
## # restmt\_ib\_target <dbl>, restmt\_ni\_target <dbl>, restmt\_nopi\_target <dbl>,  
## # restmt\_reuna\_target <dbl>, restmt\_teq\_target <dbl>,  
## # restmt\_txt\_target <dbl>, restmt\_wcap\_target <dbl>,  
## # restmt\_xint\_target <dbl>, litigated <fct>, litigation\_settlement <dbl>

summary(training\_subset\_ds\_final)

## gvkey tic aco acominc   
## Min. : 1266 0161A : 1 Min. :-0.3300 Min. :-14.4470   
## 1st Qu.: 12575 0170A : 1 1st Qu.:-0.3293 1st Qu.: 0.1232   
## Median : 31392 0173A : 1 Median :-0.3147 Median : 0.1415   
## Mean : 80970 3AHII : 1 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:164700 3AIRW : 1 3rd Qu.:-0.1541 3rd Qu.: 0.1415   
## Max. :264393 3CAGZ : 1 Max. : 8.5009 Max. : 2.7827   
## (Other):243   
## ao aoloch aqc bkvlps   
## Min. :-0.3698 Min. :-4.74499 Min. :-0.2692 Min. :-0.08984   
## 1st Qu.:-0.3696 1st Qu.:-0.06602 1st Qu.:-0.2386 1st Qu.:-0.08897   
## Median :-0.3540 Median :-0.04940 Median :-0.2386 Median :-0.08893   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.:-0.1813 3rd Qu.:-0.03426 3rd Qu.:-0.2083 3rd Qu.:-0.08883   
## Max. :10.1539 Max. : 6.56127 Max. :13.4226 Max. :12.89060   
##   
## caps ch chech csho   
## Min. :-0.6113 Min. :-0.3742 Min. :-2.2327 Min. :-0.3866   
## 1st Qu.:-0.3038 1st Qu.:-0.3729 1st Qu.:-0.2584 1st Qu.:-0.3580   
## Median :-0.2882 Median :-0.3556 Median :-0.2543 Median :-0.3058   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1274 3rd Qu.:-0.1878 3rd Qu.:-0.1887 3rd Qu.:-0.1626   
## Max. :12.1385 Max. : 7.2263 Max. : 8.5161 Max. : 9.3353   
##   
## cstk dlc dltt dvt   
## Min. :-0.2397 Min. :-0.2440 Min. :-0.3488 Min. :-0.2885   
## 1st Qu.:-0.2397 1st Qu.:-0.2438 1st Qu.:-0.3488 1st Qu.:-0.2885   
## Median :-0.2394 Median :-0.2408 Median :-0.3448 Median :-0.2885   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.2240 3rd Qu.:-0.1646 3rd Qu.:-0.1012 3rd Qu.:-0.2222   
## Max. : 8.6348 Max. :12.2342 Max. :11.0168 Max. : 9.0453   
##   
## ebit epspi nopi re   
## Min. :-0.4071 Min. :-1.070685 Min. :-0.4232 Min. :-1.4979   
## 1st Qu.:-0.3137 1st Qu.:-0.120920 1st Qu.:-0.2243 1st Qu.:-0.2770   
## Median :-0.3056 Median :-0.100687 Median :-0.2233 Median :-0.2722   
## Mean : 0.0000 Mean : 0.000000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1561 3rd Qu.: 0.006599 3rd Qu.:-0.1993 3rd Qu.:-0.2089   
## Max. :10.5920 Max. :15.572979 Max. :11.2569 Max. :10.8504   
##   
## rect revt siv sstk   
## Min. :-0.3542 Min. :-0.2501 Min. :-0.2110 Min. :-0.3282   
## 1st Qu.:-0.3531 1st Qu.:-0.2496 1st Qu.:-0.2110 1st Qu.:-0.3282   
## Median :-0.3377 Median :-0.2403 Median :-0.2110 Median :-0.3195   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1105 3rd Qu.:-0.1306 3rd Qu.:-0.2088 3rd Qu.:-0.2106   
## Max. :10.3009 Max. :13.6928 Max. : 9.8069 Max. : 8.9623   
##   
## teq tstk wcap restmt\_at\_mag   
## Min. :-0.5964 Min. :-0.2350 Min. :-6.64063 Min. :-7.79754   
## 1st Qu.:-0.3122 1st Qu.:-0.2350 1st Qu.:-0.20600 1st Qu.:-0.01517   
## Median :-0.2994 Median :-0.2350 Median :-0.18806 Median :-0.01517   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.:-0.1690 3rd Qu.:-0.2317 3rd Qu.: 0.01965 3rd Qu.:-0.01517   
## Max. : 9.5272 Max. : 9.7101 Max. : 9.37296 Max. : 9.79789   
##   
## restmt\_capx\_mag restmt\_cogs\_mag restmt\_epspi\_mag restmt\_ni\_mag   
## Min. :-13.68831 Min. :-4.25205 Min. :-0.09336 Min. :-0.10223   
## 1st Qu.: 0.06399 1st Qu.:-0.03739 1st Qu.:-0.08352 1st Qu.:-0.06336   
## Median : 0.06399 Median :-0.03739 Median :-0.08352 Median :-0.06336   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.06399 3rd Qu.:-0.03739 3rd Qu.:-0.08352 3rd Qu.:-0.06336   
## Max. : 5.10905 Max. : 8.39193 Max. :15.08304 Max. :15.71626   
##   
## restmt\_nopi\_mag restmt\_reuna\_mag restmt\_teq\_mag restmt\_txt\_mag   
## Min. :-15.67603 Min. :-0.38523 Min. :-0.19517 Min. :-10.4469   
## 1st Qu.: 0.06656 1st Qu.:-0.06395 1st Qu.:-0.06259 1st Qu.: 0.1166   
## Median : 0.06721 Median :-0.06395 Median :-0.06259 Median : 0.1166   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.06734 3rd Qu.:-0.06395 3rd Qu.:-0.06259 3rd Qu.: 0.1166   
## Max. : 0.64741 Max. :15.70489 Max. :15.71558 Max. : 2.2421   
##   
## restmt\_wcap\_mag restmt\_xint\_mag cshtrd\_m prccd\_m   
## Min. :-0.18326 Min. :-11.0587 Min. :-0.4389 Min. :-0.21813   
## 1st Qu.:-0.06281 1st Qu.: 0.1856 1st Qu.:-0.4299 1st Qu.:-0.21061   
## Median :-0.06281 Median : 0.1856 Median :-0.3826 Median :-0.15991   
## Mean : 0.00000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.:-0.06281 3rd Qu.: 0.1856 3rd Qu.:-0.1298 3rd Qu.:-0.01338   
## Max. :15.71528 Max. : 0.2863 Max. : 6.1685 Max. :14.71819   
##   
## prcod\_m trfd\_m trfm\_m pe\_ratio   
## Min. :-0.22997 Min. :-0.1347 Min. :-0.2986 Min. :-0.08941   
## 1st Qu.:-0.21193 1st Qu.:-0.1305 1st Qu.:-0.2986 1st Qu.:-0.08920   
## Median :-0.16523 Median :-0.1195 Median :-0.2986 Median :-0.08919   
## Mean : 0.00000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.:-0.03081 3rd Qu.:-0.0795 3rd Qu.:-0.1578 3rd Qu.:-0.08919   
## Max. :12.47556 Max. :14.3859 Max. : 9.6410 Max. :13.50746   
##   
## wc\_ratio de\_ratio roe\_ratio sp\_rating\_target   
## Min. :-0.75253 Min. :-8.870033 Min. :-15.43447 Min. :-0.6249   
## 1st Qu.:-0.53360 1st Qu.:-0.087951 1st Qu.: 0.05475 1st Qu.:-0.6249   
## Median :-0.32842 Median :-0.009862 Median : 0.06384 Median :-0.6249   
## Mean : 0.00000 Mean : 0.000000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.04772 3rd Qu.: 0.096630 3rd Qu.: 0.07561 3rd Qu.: 0.4963   
## Max. : 6.57371 Max. :11.488569 Max. : 0.87283 Max. : 1.8344   
##   
## restmt\_at\_target restmt\_capx\_target restmt\_cogs\_target restmt\_dltt\_target  
## Min. :-0.2527 Min. :-0.1818 Min. :-0.6427 Min. :-0.1429   
## 1st Qu.:-0.2527 1st Qu.:-0.1818 1st Qu.:-0.6427 1st Qu.:-0.1429   
## Median :-0.2527 Median :-0.1818 Median :-0.6427 Median :-0.1429   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.2527 3rd Qu.:-0.1818 3rd Qu.: 1.5496 3rd Qu.:-0.1429   
## Max. : 3.9417 Max. : 5.4776 Max. : 1.5496 Max. : 6.9717   
##   
## restmt\_epspi\_target restmt\_ib\_target restmt\_ni\_target restmt\_nopi\_target  
## Min. :-0.3763 Min. :-0.3552 Min. :-0.1933 Min. :-1.2815   
## 1st Qu.:-0.3763 1st Qu.:-0.3552 1st Qu.:-0.1933 1st Qu.:-1.2815   
## Median :-0.3763 Median :-0.3552 Median :-0.1933 Median : 0.7772   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.3763 3rd Qu.:-0.3552 3rd Qu.:-0.1933 3rd Qu.: 0.7772   
## Max. : 2.6465 Max. : 2.8038 Max. : 5.1536 Max. : 0.7772   
##   
## restmt\_reuna\_target restmt\_teq\_target restmt\_txt\_target restmt\_wcap\_target  
## Min. :-0.2786 Min. :-0.3029 Min. :-0.2786 Min. :-0.1933   
## 1st Qu.:-0.2786 1st Qu.:-0.3029 1st Qu.:-0.2786 1st Qu.:-0.1933   
## Median :-0.2786 Median :-0.3029 Median :-0.2786 Median :-0.1933   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.2786 3rd Qu.:-0.3029 3rd Qu.:-0.2786 3rd Qu.:-0.1933   
## Max. : 3.5752 Max. : 3.2884 Max. : 3.5752 Max. : 5.1536   
##   
## restmt\_xint\_target litigated litigation\_settlement  
## Min. :-0.3763 No :215 Min. : 1500000   
## 1st Qu.:-0.3763 Yes: 34 1st Qu.: 3362500   
## Median :-0.3763 Median : 6375000   
## Mean : 0.0000 Mean :10993750   
## 3rd Qu.:-0.3763 3rd Qu.: 9562500   
## Max. : 2.6465 Max. :47500000   
## NA's :241

training\_subset\_ds\_final\_orig <- training\_subset\_ds\_final

colnames(training\_subset\_ds\_final)

## [1] "gvkey" "tic" "aco"   
## [4] "acominc" "ao" "aoloch"   
## [7] "aqc" "bkvlps" "caps"   
## [10] "ch" "chech" "csho"   
## [13] "cstk" "dlc" "dltt"   
## [16] "dvt" "ebit" "epspi"   
## [19] "nopi" "re" "rect"   
## [22] "revt" "siv" "sstk"   
## [25] "teq" "tstk" "wcap"   
## [28] "restmt\_at\_mag" "restmt\_capx\_mag" "restmt\_cogs\_mag"   
## [31] "restmt\_epspi\_mag" "restmt\_ni\_mag" "restmt\_nopi\_mag"   
## [34] "restmt\_reuna\_mag" "restmt\_teq\_mag" "restmt\_txt\_mag"   
## [37] "restmt\_wcap\_mag" "restmt\_xint\_mag" "cshtrd\_m"   
## [40] "prccd\_m" "prcod\_m" "trfd\_m"   
## [43] "trfm\_m" "pe\_ratio" "wc\_ratio"   
## [46] "de\_ratio" "roe\_ratio" "sp\_rating\_target"   
## [49] "restmt\_at\_target" "restmt\_capx\_target" "restmt\_cogs\_target"   
## [52] "restmt\_dltt\_target" "restmt\_epspi\_target" "restmt\_ib\_target"   
## [55] "restmt\_ni\_target" "restmt\_nopi\_target" "restmt\_reuna\_target"   
## [58] "restmt\_teq\_target" "restmt\_txt\_target" "restmt\_wcap\_target"   
## [61] "restmt\_xint\_target" "litigated" "litigation\_settlement"

training\_subset\_ds\_final <- training\_subset\_ds\_final\_orig  
table(training\_subset\_ds\_final$litigated)

##   
## No Yes   
## 215 34

training\_subset\_ds\_final\_no <- training\_subset\_ds\_final %>%  
 filter(litigated == 'No')  
no\_count <- nrow(training\_subset\_ds\_final\_no)  
training\_subset\_ds\_final\_yes <- training\_subset\_ds\_final %>%  
 filter(litigated == 'Yes')  
yes\_count <- nrow(training\_subset\_ds\_final\_yes)  
#training\_subset\_ds\_final %>%  
# arrange(gvkey)  
  
table(training\_subset\_ds\_final$litigated)

##   
## No Yes   
## 215 34

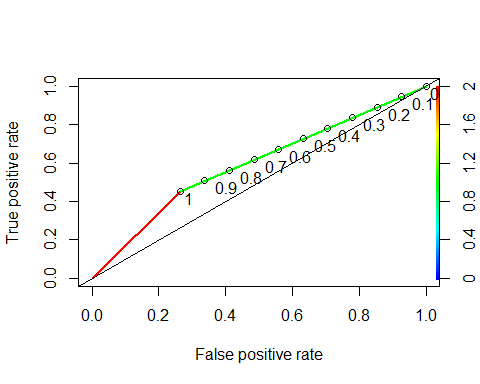
# Sampling both (Over and under combination)  
training\_subset\_ds\_final <- ovun.sample(litigated ~   
 .,   
 data = training\_subset\_ds\_final, method = "both", N = no\_count + yes\_count, p = 0.35, seed = 222, na.action = na.pass)$data  
  
##### Sampling both (Over)  
#training\_subset\_ds\_final <- ovun.sample(litigated ~., data = training\_subset\_ds\_final, method = "over", N = no\_count\*2)$data  
  
##### Sampling both (Under)  
#training\_subset\_ds\_final <- ovun.sample(litigated ~., data = training\_subset\_ds\_final, method = "under", N = yes\_count\*2)$data  
  
##### Manual sample adjustment  
# litigated\_no <- training\_subset\_ds\_final %>%   
# filter(litigated == 'No')  
#   
# litigated\_yes <- training\_subset\_ds\_final %>%   
# filter(litigated == 'Yes')  
#   
# litigated\_no\_sample\_index <- sample(nrow(litigated\_no), 97)  
# litigated\_no\_sample <- litigated\_no[litigated\_no\_sample\_index, ]  
# #litigated\_no\_sample  
#   
# training\_subset\_ds\_final <- bind\_rows(litigated\_yes, litigated\_no\_sample)   
#   
# rows <- sample(nrow(training\_subset\_ds\_final))  
# training\_subset\_ds\_final <- training\_subset\_ds\_final[rows, ]  
  
table(training\_subset\_ds\_final$litigated)

##   
## No Yes   
## 158 91

glm\_control <- trainControl(  
 method = "cv",  
 number = 10,  
 summaryFunction = twoClassSummary,  
 classProbs = TRUE   
)  
set.seed(123)  
glm\_model\_1 <- train(litigated ~  
 aco + acominc + ao + aoloch + aqc + bkvlps + caps +   
 ch + chech + csho + cstk + dlc + dltt + dvt +  
 ebit + epspi + nopi + re + rect + revt + siv +  
 sstk + teq + tstk + wcap +  
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_target +   
 restmt\_capx\_target +   
 restmt\_dltt\_target +   
 restmt\_wcap\_target +   
 restmt\_teq\_target + restmt\_teq\_mag +  
 restmt\_cogs\_target + restmt\_cogs\_mag +  
 restmt\_nopi\_target + restmt\_nopi\_mag  
 ,   
 data = training\_subset\_ds\_final, method = "glm", family = "binomial", trControl = glm\_control)  
class\_probabilities <- predict(glm\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glm\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glm\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glm\_confusion\_matrix <- confusionMatrix(factor(glm\_class\_probabilities\_litigated), factor(glm\_litigated), positive = "Yes")  
glm\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 53 6  
## Yes 19 5  
##   
## Accuracy : 0.6988   
## 95% CI : (0.5882, 0.7947)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : 0.1271   
##   
## Mcnemar's Test P-Value : 0.0164   
##   
## Sensitivity : 0.45455   
## Specificity : 0.73611   
## Pos Pred Value : 0.20833   
## Neg Pred Value : 0.89831   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.28916   
## Balanced Accuracy : 0.59533   
##   
## 'Positive' Class : Yes   
##

glm\_accuracy\_1 <- glm\_confusion\_matrix$overall["Accuracy"]  
glm\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
glm\_rocr\_roc <- performance(glm\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(glm\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



glm\_rocr\_auc <- performance(glm\_rocr\_pred, measure = "auc")  
glm\_auc\_1 <- glm\_rocr\_auc@y.values[[1]]  
glm\_auc\_1

## [1] 0.5953283

summary(glm\_model\_1)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -8.49 0.00 0.00 0.00 8.49   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.131e+15 4.605e+07 -111431553 <2e-16 \*\*\*  
## aco 4.043e+15 4.460e+07 90648900 <2e-16 \*\*\*  
## acominc 3.220e+15 2.348e+07 137178692 <2e-16 \*\*\*  
## ao -5.428e+14 3.610e+07 -15035192 <2e-16 \*\*\*  
## aoloch 7.484e+14 1.134e+07 65985858 <2e-16 \*\*\*  
## aqc -1.799e+15 1.886e+07 -95415693 <2e-16 \*\*\*  
## bkvlps -4.083e+16 4.532e+08 -90093108 <2e-16 \*\*\*  
## caps -5.965e+14 2.821e+07 -21147286 <2e-16 \*\*\*  
## ch 2.603e+15 2.856e+07 91136017 <2e-16 \*\*\*  
## chech -4.722e+14 1.055e+07 -44767007 <2e-16 \*\*\*  
## csho -8.314e+14 1.753e+07 -47427961 <2e-16 \*\*\*  
## cstk 5.620e+14 1.347e+07 41728260 <2e-16 \*\*\*  
## dlc -1.026e+16 6.144e+07 -166928123 <2e-16 \*\*\*  
## dltt 5.309e+15 4.711e+07 112700161 <2e-16 \*\*\*  
## dvt -7.486e+15 7.284e+07 -102767882 <2e-16 \*\*\*  
## ebit 3.307e+15 7.892e+07 41904311 <2e-16 \*\*\*  
## epspi -3.465e+14 2.924e+07 -11848284 <2e-16 \*\*\*  
## nopi -1.091e+15 3.578e+07 -30508757 <2e-16 \*\*\*  
## re 1.722e+15 9.209e+07 18697194 <2e-16 \*\*\*  
## rect -4.393e+14 3.454e+07 -12718259 <2e-16 \*\*\*  
## revt -3.625e+14 3.580e+07 -10125245 <2e-16 \*\*\*  
## siv 1.275e+15 1.655e+07 77072451 <2e-16 \*\*\*  
## sstk 8.848e+14 9.141e+06 96797074 <2e-16 \*\*\*  
## teq -7.777e+14 1.127e+08 -6900627 <2e-16 \*\*\*  
## tstk -1.571e+12 4.632e+07 -33925 <2e-16 \*\*\*  
## wcap -1.513e+15 2.238e+07 -67587267 <2e-16 \*\*\*  
## cshtrd\_m -2.089e+14 7.604e+06 -27467564 <2e-16 \*\*\*  
## prccd\_m 2.320e+14 3.865e+07 6003860 <2e-16 \*\*\*  
## prcod\_m 5.635e+13 1.431e+07 3936974 <2e-16 \*\*\*  
## trfd\_m 8.241e+13 3.554e+06 23187236 <2e-16 \*\*\*  
## trfm\_m -3.220e+14 1.715e+07 -18778428 <2e-16 \*\*\*  
## pe\_ratio 1.959e+13 8.171e+06 2397887 <2e-16 \*\*\*  
## wc\_ratio 2.331e+14 5.077e+06 45920525 <2e-16 \*\*\*  
## de\_ratio 4.476e+14 6.794e+06 65883443 <2e-16 \*\*\*  
## roe\_ratio 5.034e+14 6.450e+06 78034543 <2e-16 \*\*\*  
## sp\_rating\_target -9.879e+13 8.245e+06 -11982371 <2e-16 \*\*\*  
## restmt\_at\_target 4.130e+14 8.428e+06 49001365 <2e-16 \*\*\*  
## restmt\_capx\_target 2.637e+14 6.838e+06 38570407 <2e-16 \*\*\*  
## restmt\_dltt\_target 6.535e+13 6.252e+06 10452290 <2e-16 \*\*\*  
## restmt\_wcap\_target -8.275e+13 4.911e+06 -16848970 <2e-16 \*\*\*  
## restmt\_teq\_target -3.090e+14 8.789e+06 -35155034 <2e-16 \*\*\*  
## restmt\_teq\_mag 8.047e+14 3.556e+08 2262754 <2e-16 \*\*\*  
## restmt\_cogs\_target -1.795e+14 5.858e+06 -30637310 <2e-16 \*\*\*  
## restmt\_cogs\_mag -6.047e+13 5.223e+06 -11577591 <2e-16 \*\*\*  
## restmt\_nopi\_target -7.571e+13 5.641e+06 -13421319 <2e-16 \*\*\*  
## restmt\_nopi\_mag 2.500e+14 4.314e+06 57938573 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 326.94 on 248 degrees of freedom  
## Residual deviance: 3532.28 on 203 degrees of freedom  
## AIC: 3624.3  
##   
## Number of Fisher Scoring iterations: 19

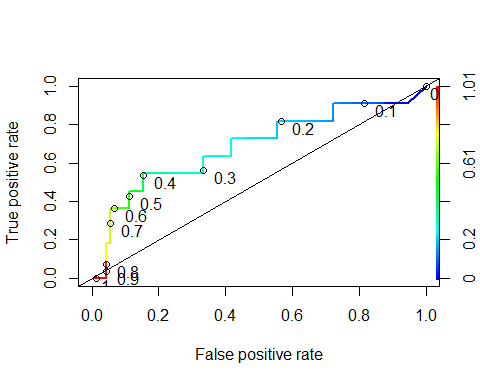
logitgof(training\_subset\_ds\_final$litigated, fitted(glm\_model\_1), g = 10)

##   
## Hosmer and Lemeshow test (binary model)  
##   
## data: training\_subset\_ds\_final$litigated, fitted(glm\_model\_1)  
## X-squared = 0.8803, df = -1, p-value = NA

set.seed(123)  
glm\_model\_2 <- train(litigated ~  
 acominc +   
 bkvlps +   
 caps +   
 dlc +   
 dvt +   
 nopi +   
 re +   
 rect +   
 revt +   
 tstk +   
 wcap +   
 prccd\_m +   
 prcod\_m +   
 trfm\_m +   
 pe\_ratio +   
 wc\_ratio +   
 sp\_rating\_target +   
 restmt\_at\_target +   
 restmt\_capx\_target +   
 restmt\_dltt\_target +   
 restmt\_cogs\_target +   
 restmt\_nopi\_target +   
 restmt\_nopi\_mag  
 ,   
 data = training\_subset\_ds\_final, method = "glm", family = "binomial", trControl = glm\_control)  
class\_probabilities <- predict(glm\_model\_2, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glm\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glm\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glm\_confusion\_matrix <- confusionMatrix(factor(glm\_class\_probabilities\_litigated), factor(glm\_litigated), positive = "Yes")  
glm\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 64 7  
## Yes 8 4  
##   
## Accuracy : 0.8193   
## 95% CI : (0.7195, 0.8952)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.9225   
##   
## Kappa : 0.2432   
##   
## Mcnemar's Test P-Value : 1.0000   
##   
## Sensitivity : 0.36364   
## Specificity : 0.88889   
## Pos Pred Value : 0.33333   
## Neg Pred Value : 0.90141   
## Prevalence : 0.13253   
## Detection Rate : 0.04819   
## Detection Prevalence : 0.14458   
## Balanced Accuracy : 0.62626   
##   
## 'Positive' Class : Yes   
##

glm\_accuracy\_2 <- glm\_confusion\_matrix$overall["Accuracy"]  
glm\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
glm\_rocr\_roc <- performance(glm\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(glm\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



glm\_rocr\_auc <- performance(glm\_rocr\_pred, measure = "auc")  
glm\_auc\_2 <- glm\_rocr\_auc@y.values[[1]]  
glm\_auc\_2

## [1] 0.6856061

summary(glm\_model\_2)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5378 -0.6855 -0.3338 0.7187 2.3352   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.251e+03 7.944e+02 -1.574 0.11541   
## acominc 1.485e+00 1.565e+00 0.949 0.34279   
## bkvlps -3.087e+03 1.396e+03 -2.212 0.02699 \*   
## caps 1.166e-01 7.493e-01 0.156 0.87634   
## dlc -5.101e+00 2.907e+00 -1.755 0.07930 .   
## dvt -5.324e+00 2.054e+00 -2.591 0.00956 \*\*  
## nopi 2.268e+00 9.762e-01 2.324 0.02014 \*   
## re -1.076e-01 1.728e+00 -0.062 0.95037   
## rect -4.942e-02 1.618e+00 -0.031 0.97563   
## revt 3.644e+00 1.592e+00 2.289 0.02210 \*   
## tstk 1.540e+00 9.650e-01 1.596 0.11046   
## wcap -5.575e-02 7.998e-01 -0.070 0.94443   
## prccd\_m 7.919e+00 5.656e+00 1.400 0.16148   
## prcod\_m -5.030e+00 6.412e+00 -0.785 0.43272   
## trfm\_m -5.185e-02 4.802e-01 -0.108 0.91400   
## pe\_ratio -1.092e+04 8.644e+03 -1.264 0.20632   
## wc\_ratio 1.574e-01 1.789e-01 0.880 0.37906   
## sp\_rating\_target 7.720e-01 2.800e-01 2.757 0.00583 \*\*  
## restmt\_at\_target 4.130e-01 2.188e-01 1.888 0.05904 .   
## restmt\_capx\_target 5.081e-01 2.200e-01 2.309 0.02094 \*   
## restmt\_dltt\_target -1.908e-01 2.505e-01 -0.762 0.44614   
## restmt\_cogs\_target -4.161e-01 2.336e-01 -1.782 0.07482 .   
## restmt\_nopi\_target -2.037e-01 2.187e-01 -0.931 0.35171   
## restmt\_nopi\_mag 4.413e+00 7.017e+00 0.629 0.52941   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 326.94 on 248 degrees of freedom  
## Residual deviance: 206.35 on 225 degrees of freedom  
## AIC: 254.35  
##   
## Number of Fisher Scoring iterations: 18

logitgof(training\_subset\_ds\_final$litigated, fitted(glm\_model\_2), g = 10)

##   
## Hosmer and Lemeshow test (binary model)  
##   
## data: training\_subset\_ds\_final$litigated, fitted(glm\_model\_2)  
## X-squared = 14.107, df = 8, p-value = 0.07902

set.seed(123)  
glm\_model\_3 <- train(litigated ~  
 dvt +   
 nopi +   
 revt +   
 tstk +   
 prccd\_m +   
 trfm\_m +   
 pe\_ratio +   
 wc\_ratio +   
 sp\_rating\_target  
 ,   
 data = training\_subset\_ds\_final, method = "glm", family = "binomial", trControl = glm\_control)  
class\_probabilities <- predict(glm\_model\_3, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glm\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glm\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glm\_confusion\_matrix <- confusionMatrix(factor(glm\_class\_probabilities\_litigated), factor(glm\_litigated), positive = "Yes")  
glm\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 64 6  
## Yes 8 5  
##   
## Accuracy : 0.8313   
## 95% CI : (0.7332, 0.9046)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.8698   
##   
## Kappa : 0.3189   
##   
## Mcnemar's Test P-Value : 0.7893   
##   
## Sensitivity : 0.45455   
## Specificity : 0.88889   
## Pos Pred Value : 0.38462   
## Neg Pred Value : 0.91429   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.15663   
## Balanced Accuracy : 0.67172   
##   
## 'Positive' Class : Yes   
##

glm\_accuracy\_3 <- glm\_confusion\_matrix$overall["Accuracy"]  
glm\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
glm\_rocr\_roc <- performance(glm\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(glm\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



glm\_rocr\_auc <- performance(glm\_rocr\_pred, measure = "auc")  
glm\_auc\_3 <- glm\_rocr\_auc@y.values[[1]]  
glm\_auc\_3

## [1] 0.7537879

summary(glm\_model\_3)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8622 -0.7650 -0.6942 0.9230 2.5164   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -264.6458 491.3706 -0.539 0.59017   
## dvt -3.7066 1.3471 -2.752 0.00593 \*\*   
## nopi 0.6730 0.4909 1.371 0.17044   
## revt 1.9624 0.7308 2.685 0.00725 \*\*   
## tstk 0.7940 0.4234 1.875 0.06074 .   
## prccd\_m -0.1059 0.2135 -0.496 0.61971   
## trfm\_m -0.3501 0.4549 -0.770 0.44158   
## pe\_ratio -2956.1840 5508.9445 -0.537 0.59153   
## wc\_ratio 0.2332 0.1596 1.462 0.14385   
## sp\_rating\_target 0.8367 0.2019 4.145 3.4e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 326.94 on 248 degrees of freedom  
## Residual deviance: 251.56 on 239 degrees of freedom  
## AIC: 271.56  
##   
## Number of Fisher Scoring iterations: 18

logitgof(training\_subset\_ds\_final$litigated, fitted(glm\_model\_3), g = 10)

##   
## Hosmer and Lemeshow test (binary model)  
##   
## data: training\_subset\_ds\_final$litigated, fitted(glm\_model\_3)  
## X-squared = 20.63, df = 8, p-value = 0.008197

glmnet\_control <- trainControl(  
 method = "cv",  
 number = 10,  
 summaryFunction = twoClassSummary,  
 classProbs = TRUE   
)  
set.seed(123)  
glmnet\_model\_1 <- train(litigated ~   
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data = training\_subset\_ds\_final, method = "glmnet", family = "binomial", trControl = glmnet\_control)  
class\_probabilities <- predict(glmnet\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glmnet\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glmnet\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glmnet\_confusion\_matrix <- confusionMatrix(factor(glmnet\_class\_probabilities\_litigated), factor(glmnet\_litigated), positive = "Yes")  
glmnet\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 59 6  
## Yes 13 5  
##   
## Accuracy : 0.7711   
## 95% CI : (0.6658, 0.8562)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.9948   
##   
## Kappa : 0.2158   
##   
## Mcnemar's Test P-Value : 0.1687   
##   
## Sensitivity : 0.45455   
## Specificity : 0.81944   
## Pos Pred Value : 0.27778   
## Neg Pred Value : 0.90769   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.21687   
## Balanced Accuracy : 0.63699   
##   
## 'Positive' Class : Yes   
##

glmnet\_accuracy\_1 <- glmnet\_confusion\_matrix$overall["Accuracy"]

set.seed(123)  
glmnet\_model\_2 <- train(litigated ~   
 aco + acominc + ao + aoloch + aqc + bkvlps + caps +   
 ch + chech + csho + cstk + dlc + dltt + dvt +  
 ebit + epspi + nopi + re + rect + revt + siv +  
 sstk + teq + tstk + wcap +  
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_target +   
 restmt\_capx\_target +   
 restmt\_dltt\_target +   
 restmt\_wcap\_target +   
 restmt\_teq\_target + restmt\_teq\_mag +  
 restmt\_cogs\_target + restmt\_cogs\_mag +  
 restmt\_nopi\_target + restmt\_nopi\_mag,   
 data = training\_subset\_ds\_final, method = "glmnet", family = "binomial", trControl = glmnet\_control)  
class\_probabilities <- predict(glmnet\_model\_2, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glmnet\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glmnet\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glmnet\_confusion\_matrix <- confusionMatrix(factor(glmnet\_class\_probabilities\_litigated), factor(glmnet\_litigated), positive = "Yes")  
glmnet\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 60 8  
## Yes 12 3  
##   
## Accuracy : 0.759   
## 95% CI : (0.6527, 0.8462)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.9977   
##   
## Kappa : 0.0919   
##   
## Mcnemar's Test P-Value : 0.5023   
##   
## Sensitivity : 0.27273   
## Specificity : 0.83333   
## Pos Pred Value : 0.20000   
## Neg Pred Value : 0.88235   
## Prevalence : 0.13253   
## Detection Rate : 0.03614   
## Detection Prevalence : 0.18072   
## Balanced Accuracy : 0.55303   
##   
## 'Positive' Class : Yes   
##

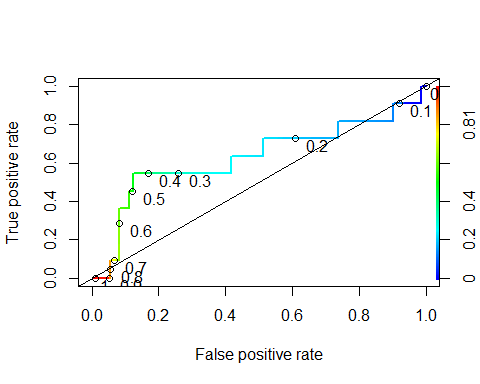
glmnet\_accuracy\_2 <- glmnet\_confusion\_matrix$overall["Accuracy"]

set.seed(123)  
glmnet\_model\_3 <- train(litigated ~   
 bkvlps +   
 dlc +   
 dvt +   
 nopi +   
 rect +   
 revt +   
 tstk +   
 wcap +   
 prccd\_m +   
 prcod\_m +   
 pe\_ratio +   
 wc\_ratio +   
 sp\_rating\_target +   
 restmt\_at\_target +   
 restmt\_capx\_target +   
 restmt\_nopi\_target,   
 data = training\_subset\_ds\_final, method = "glmnet", family = "binomial", trControl = glmnet\_control)  
class\_probabilities <- predict(glmnet\_model\_3, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
glmnet\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
glmnet\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
glmnet\_confusion\_matrix <- confusionMatrix(factor(glmnet\_class\_probabilities\_litigated), factor(glmnet\_litigated), positive = "Yes")  
glmnet\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 64 6  
## Yes 8 5  
##   
## Accuracy : 0.8313   
## 95% CI : (0.7332, 0.9046)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.8698   
##   
## Kappa : 0.3189   
##   
## Mcnemar's Test P-Value : 0.7893   
##   
## Sensitivity : 0.45455   
## Specificity : 0.88889   
## Pos Pred Value : 0.38462   
## Neg Pred Value : 0.91429   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.15663   
## Balanced Accuracy : 0.67172   
##   
## 'Positive' Class : Yes   
##

glmnet\_accuracy\_3 <- glmnet\_confusion\_matrix$overall["Accuracy"]

glmnet\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
glmnet\_rocr\_roc <- performance(glmnet\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(glmnet\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



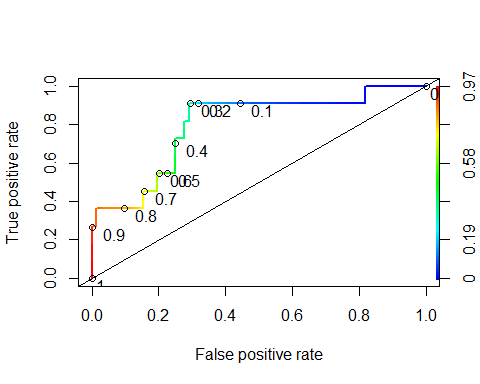
glmnet\_rocr\_auc <- performance(glmnet\_rocr\_pred, measure = "auc")  
glmnet\_auc <- glmnet\_rocr\_auc@y.values[[1]]  
glmnet\_auc

## [1] 0.6275253

gbm\_control <- trainControl(method = "repeatedcv", number = 10, repeats = 10)  
set.seed(123)  
gbm\_model\_1 <- train(litigated~  
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data=training\_subset\_ds\_final,   
 method = "gbm",  
 trControl = gbm\_control,  
 verbose = FALSE)  
  
class\_probabilities <- predict(gbm\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
gbm\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
gbm\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
gbm\_confusion\_matrix <- confusionMatrix(factor(gbm\_class\_probabilities\_litigated), factor(gbm\_litigated), positive = "Yes")  
gbm\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 56 5  
## Yes 16 6  
##   
## Accuracy : 0.747   
## 95% CI : (0.6396, 0.8361)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.9990   
##   
## Kappa : 0.2271   
##   
## Mcnemar's Test P-Value : 0.0291   
##   
## Sensitivity : 0.54545   
## Specificity : 0.77778   
## Pos Pred Value : 0.27273   
## Neg Pred Value : 0.91803   
## Prevalence : 0.13253   
## Detection Rate : 0.07229   
## Detection Prevalence : 0.26506   
## Balanced Accuracy : 0.66162   
##   
## 'Positive' Class : Yes   
##

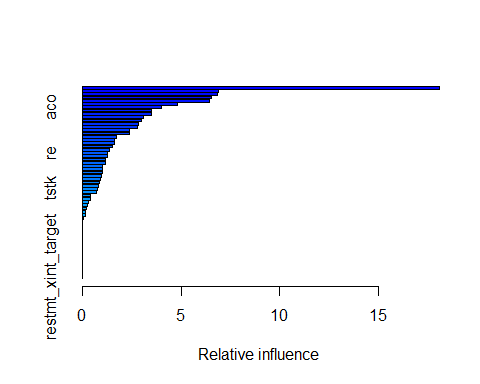
gbm\_accuracy\_1 <- gbm\_confusion\_matrix$overall["Accuracy"]  
gbm\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
gbm\_rocr\_roc <- performance(gbm\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(gbm\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



gbm\_rocr\_auc <- performance(gbm\_rocr\_pred, measure = "auc")  
gbm\_auc\_1 <- gbm\_rocr\_auc@y.values[[1]]  
gbm\_auc\_1

## [1] 0.7954545

summary(gbm\_model\_1)

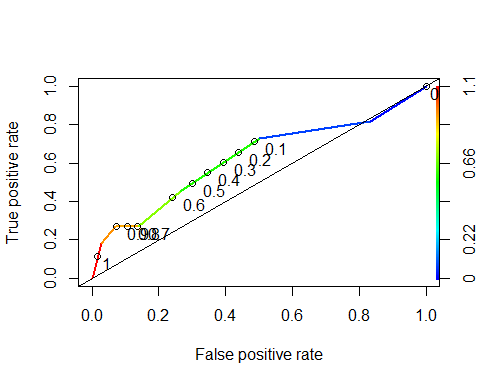


## var rel.inf  
## caps caps 18.07831944  
## dlc dlc 6.88627029  
## cshtrd\_m cshtrd\_m 6.84867012  
## prccd\_m prccd\_m 6.53401449  
## ebit ebit 6.41413120  
## cstk cstk 4.80156804  
## aco aco 4.00091708  
## prcod\_m prcod\_m 3.52031428  
## acominc acominc 3.48222060  
## wcap wcap 3.07182895  
## bkvlps bkvlps 2.99672348  
## pe\_ratio pe\_ratio 2.83105359  
## restmt\_nopi\_mag restmt\_nopi\_mag 2.78159259  
## wc\_ratio wc\_ratio 2.41058341  
## chech chech 2.37753171  
## trfd\_m trfd\_m 1.72339810  
## restmt\_ni\_target restmt\_ni\_target 1.63082527  
## ao ao 1.61775626  
## teq teq 1.54141432  
## aqc aqc 1.38748239  
## re re 1.28733137  
## restmt\_epspi\_target restmt\_epspi\_target 1.26962616  
## nopi nopi 1.19159279  
## roe\_ratio roe\_ratio 1.18979540  
## ch ch 1.02505827  
## siv siv 1.02127206  
## de\_ratio de\_ratio 0.99702012  
## sstk sstk 0.97114918  
## dltt dltt 0.93250197  
## trfm\_m trfm\_m 0.88367793  
## revt revt 0.80302419  
## tstk tstk 0.79369172  
## aoloch aoloch 0.72953898  
## epspi epspi 0.42120511  
## sp\_rating\_target sp\_rating\_target 0.39767194  
## dvt dvt 0.31475530  
## csho csho 0.25636091  
## restmt\_txt\_target restmt\_txt\_target 0.22033979  
## restmt\_epspi\_mag restmt\_epspi\_mag 0.16102464  
## restmt\_teq\_mag restmt\_teq\_mag 0.14313209  
## restmt\_cogs\_target restmt\_cogs\_target 0.04490493  
## restmt\_ib\_target restmt\_ib\_target 0.00870954  
## rect rect 0.00000000  
## restmt\_at\_mag restmt\_at\_mag 0.00000000  
## restmt\_capx\_mag restmt\_capx\_mag 0.00000000  
## restmt\_cogs\_mag restmt\_cogs\_mag 0.00000000  
## restmt\_ni\_mag restmt\_ni\_mag 0.00000000  
## restmt\_reuna\_mag restmt\_reuna\_mag 0.00000000  
## restmt\_txt\_mag restmt\_txt\_mag 0.00000000  
## restmt\_wcap\_mag restmt\_wcap\_mag 0.00000000  
## restmt\_xint\_mag restmt\_xint\_mag 0.00000000  
## restmt\_at\_target restmt\_at\_target 0.00000000  
## restmt\_capx\_target restmt\_capx\_target 0.00000000  
## restmt\_dltt\_target restmt\_dltt\_target 0.00000000  
## restmt\_nopi\_target restmt\_nopi\_target 0.00000000  
## restmt\_reuna\_target restmt\_reuna\_target 0.00000000  
## restmt\_teq\_target restmt\_teq\_target 0.00000000  
## restmt\_wcap\_target restmt\_wcap\_target 0.00000000  
## restmt\_xint\_target restmt\_xint\_target 0.00000000

set.seed(123)  
tree\_model\_1 <- rpart(litigated ~   
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data = training\_subset\_ds\_final, method = "class")  
class\_probabilities <- predict(tree\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- 1 - class\_probabilities[1: nrow(test\_subset\_ds\_final)]  
tree\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
  
tree\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
tree\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
tree\_confusion\_matrix <- confusionMatrix(factor(tree\_class\_probabilities\_litigated), factor(tree\_litigated), positive = "Yes")  
tree\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 53 6  
## Yes 19 5  
##   
## Accuracy : 0.6988   
## 95% CI : (0.5882, 0.7947)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : 0.1271   
##   
## Mcnemar's Test P-Value : 0.0164   
##   
## Sensitivity : 0.45455   
## Specificity : 0.73611   
## Pos Pred Value : 0.20833   
## Neg Pred Value : 0.89831   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.28916   
## Balanced Accuracy : 0.59533   
##   
## 'Positive' Class : Yes   
##

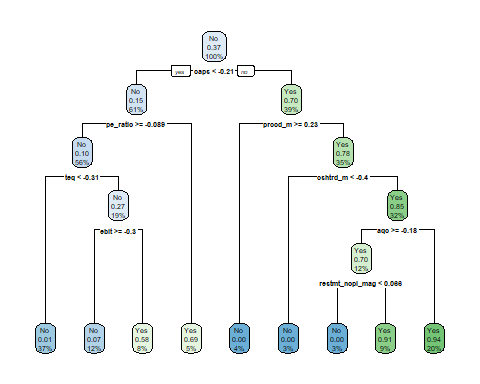
tree\_accuracy\_1 <- tree\_confusion\_matrix$overall["Accuracy"]  
  
tree\_rocr\_roc <- performance(tree\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(tree\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



tree\_rocr\_auc <- performance(tree\_rocr\_pred, measure = "auc")  
tree\_auc <- tree\_rocr\_auc@y.values[[1]]  
tree\_auc

## [1] 0.625

rpart.plot(tree\_model\_1)



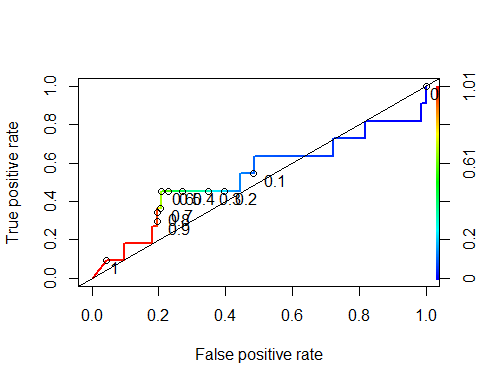
lda\_control <- trainControl(  
 method = "cv",  
 number = 10  
)  
set.seed(123)  
lda\_model\_1 <- train(litigated ~  
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data = training\_subset\_ds\_final, method = "lda", family = "binomial", trControl = lda\_control)  
class\_probabilities <- predict(lda\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_onehot <- class\_probabilities$Yes  
lda\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
lda\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
lda\_confusion\_matrix <- confusionMatrix(factor(lda\_class\_probabilities\_litigated), factor(lda\_litigated), positive = "Yes")  
lda\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 53 6  
## Yes 19 5  
##   
## Accuracy : 0.6988   
## 95% CI : (0.5882, 0.7947)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : 0.1271   
##   
## Mcnemar's Test P-Value : 0.0164   
##   
## Sensitivity : 0.45455   
## Specificity : 0.73611   
## Pos Pred Value : 0.20833   
## Neg Pred Value : 0.89831   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.28916   
## Balanced Accuracy : 0.59533   
##   
## 'Positive' Class : Yes   
##

lda\_accuracy\_1 <- lda\_confusion\_matrix$overall["Accuracy"]  
lda\_accuracy\_1

## Accuracy   
## 0.6987952

lda\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_onehot, test\_subset\_ds\_final$litigated)  
lda\_rocr\_roc <- performance(lda\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(lda\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



lda\_rocr\_auc <- performance(lda\_rocr\_pred, measure = "auc")  
lda\_auc\_1 <- lda\_rocr\_auc@y.values[[1]]  
lda\_auc\_1

## [1] 0.5309343

rf\_x <- training\_subset\_ds\_final[,num\_var\_start\_index:num\_var\_end\_index]  
rf\_y <- training\_subset\_ds\_final[,target\_var\_index]  
recommended\_mtry <- sqrt(ncol(rf\_x))  
rfGrid <- expand.grid(mtry=recommended\_mtry)

rfControl <- trainControl(method='repeatedcv', number=10, repeats=3)  
set.seed(123)  
rf\_model\_1 <- train(litigated~  
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data=training\_subset\_ds\_final,   
 method='rf',   
 metric='Accuracy',   
 tuneGrid=rfGrid,   
 trControl=rfControl,   
 ntree = 100  
 )  
  
class\_probabilities <- predict(rf\_model\_1, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
  
rf\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
rf\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
rf\_confusion\_matrix <- confusionMatrix(factor(rf\_class\_probabilities\_litigated), factor(rf\_litigated), positive = "Yes")  
rf\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 67 7  
## Yes 5 4  
##   
## Accuracy : 0.8554   
## 95% CI : (0.7611, 0.923)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.6969   
##   
## Kappa : 0.3187   
##   
## Mcnemar's Test P-Value : 0.7728   
##   
## Sensitivity : 0.36364   
## Specificity : 0.93056   
## Pos Pred Value : 0.44444   
## Neg Pred Value : 0.90541   
## Prevalence : 0.13253   
## Detection Rate : 0.04819   
## Detection Prevalence : 0.10843   
## Balanced Accuracy : 0.64710   
##   
## 'Positive' Class : Yes   
##

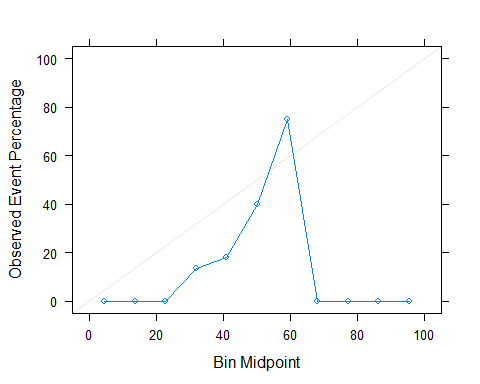
rf\_accuracy\_1 <- rf\_confusion\_matrix$overall["Accuracy"]  
rf\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
rf\_rocr\_roc <- performance(rf\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(rf\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



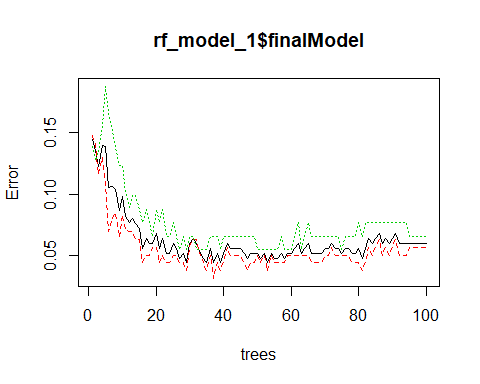
rf\_rocr\_auc <- performance(rf\_rocr\_pred, measure = "auc")  
rf\_auc\_1 <- rf\_rocr\_auc@y.values[[1]]  
rf\_auc\_1

## [1] 0.8863636

calibration\_curve <- calibration(litigated ~ class\_probabilities\_litigated,  
 data = test\_subset\_ds\_final,  
 class = 1)  
plot(calibration\_curve)



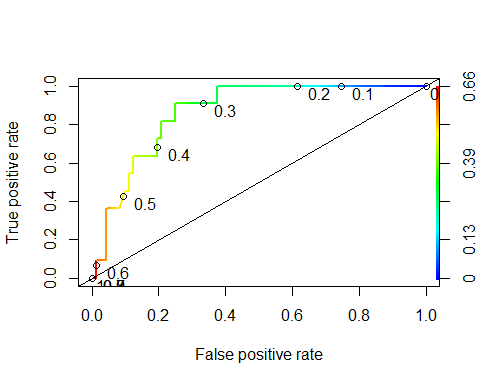
plot(rf\_model\_1$finalModel)



set.seed(123)  
rf\_model\_2 <- train(litigated~  
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data=training\_subset\_ds\_final,   
 method='rf',   
 metric='Accuracy',   
 tuneGrid=rfGrid,   
 trControl=rfControl,   
 ntree = 300  
 )  
  
class\_probabilities <- predict(rf\_model\_2, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
  
rf\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
rf\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
rf\_confusion\_matrix <- confusionMatrix(factor(rf\_class\_probabilities\_litigated), factor(rf\_litigated), positive = "Yes")  
rf\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 66 7  
## Yes 6 4  
##   
## Accuracy : 0.8434   
## 95% CI : (0.7471, 0.9139)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.795   
##   
## Kappa : 0.2915   
##   
## Mcnemar's Test P-Value : 1.000   
##   
## Sensitivity : 0.36364   
## Specificity : 0.91667   
## Pos Pred Value : 0.40000   
## Neg Pred Value : 0.90411   
## Prevalence : 0.13253   
## Detection Rate : 0.04819   
## Detection Prevalence : 0.12048   
## Balanced Accuracy : 0.64015   
##   
## 'Positive' Class : Yes   
##

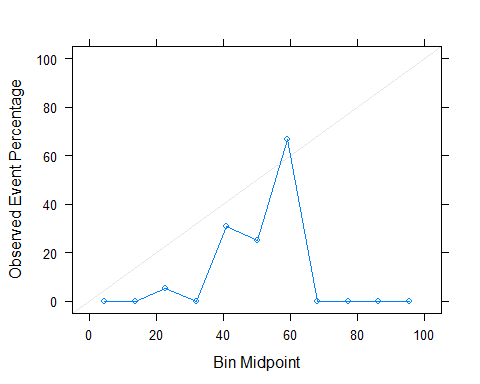
rf\_accuracy\_2 <- rf\_confusion\_matrix$overall["Accuracy"]  
rf\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
rf\_rocr\_roc <- performance(rf\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(rf\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



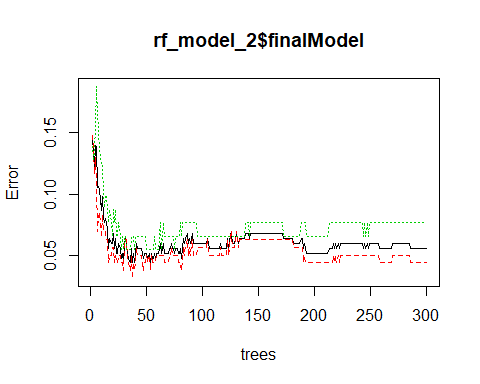
rf\_rocr\_auc <- performance(rf\_rocr\_pred, measure = "auc")  
rf\_auc\_2 <- rf\_rocr\_auc@y.values[[1]]  
rf\_auc\_2

## [1] 0.8642677

calibration\_curve <- calibration(litigated ~ class\_probabilities\_litigated,  
 data = test\_subset\_ds\_final,  
 class = 1)  
plot(calibration\_curve)



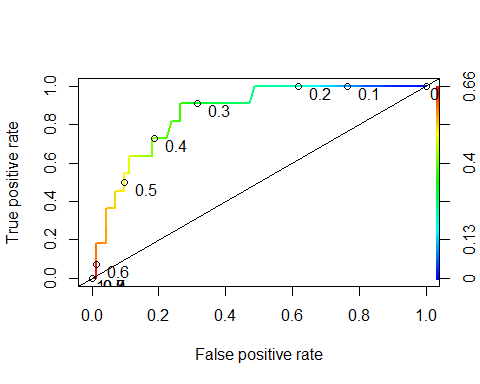
plot(rf\_model\_2$finalModel)



set.seed(123)  
rf\_model\_3 <- train(litigated~  
 aco + acominc + ao + aoloch + aqc +  
 bkvlps + caps + ch + chech + csho + cstk + dlc +  
 dltt + dvt + ebit + epspi + nopi + re + rect +  
 revt + siv + sstk + teq + tstk + wcap +   
 cshtrd\_m + prccd\_m + prcod\_m + trfd\_m + trfm\_m + pe\_ratio +  
 wc\_ratio + de\_ratio + roe\_ratio + sp\_rating\_target +  
 restmt\_at\_mag + restmt\_capx\_mag + restmt\_cogs\_mag +   
 restmt\_epspi\_mag + restmt\_ni\_mag + restmt\_nopi\_mag +   
 restmt\_reuna\_mag + restmt\_teq\_mag + restmt\_txt\_mag +   
 restmt\_wcap\_mag + restmt\_xint\_mag +   
 restmt\_at\_target + restmt\_capx\_target + restmt\_cogs\_target +   
 restmt\_dltt\_target + restmt\_epspi\_target + restmt\_ib\_target +   
 restmt\_ni\_target + restmt\_nopi\_target + restmt\_reuna\_target +   
 restmt\_teq\_target + restmt\_txt\_target + restmt\_wcap\_target +   
 restmt\_xint\_target,   
 data=training\_subset\_ds\_final,   
 method='rf',   
 metric='Accuracy',   
 tuneGrid=rfGrid,   
 trControl=rfControl,   
 ntree = 500  
 )  
  
class\_probabilities <- predict(rf\_model\_3, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
  
rf\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
rf\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
rf\_confusion\_matrix <- confusionMatrix(factor(rf\_class\_probabilities\_litigated), factor(rf\_litigated), positive = "Yes")  
rf\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 65 6  
## Yes 7 5  
##   
## Accuracy : 0.8434   
## 95% CI : (0.7471, 0.9139)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.795   
##   
## Kappa : 0.3441   
##   
## Mcnemar's Test P-Value : 1.000   
##   
## Sensitivity : 0.45455   
## Specificity : 0.90278   
## Pos Pred Value : 0.41667   
## Neg Pred Value : 0.91549   
## Prevalence : 0.13253   
## Detection Rate : 0.06024   
## Detection Prevalence : 0.14458   
## Balanced Accuracy : 0.67866   
##   
## 'Positive' Class : Yes   
##

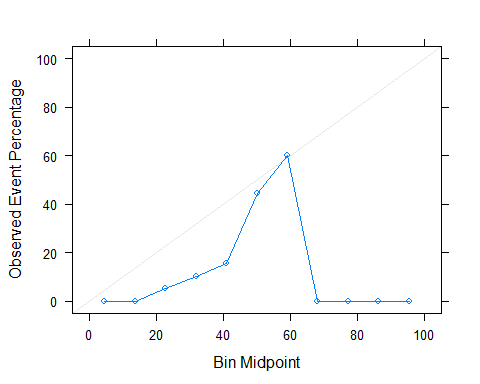
rf\_accuracy\_3 <- rf\_confusion\_matrix$overall["Accuracy"]  
rf\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
rf\_rocr\_roc <- performance(rf\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(rf\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



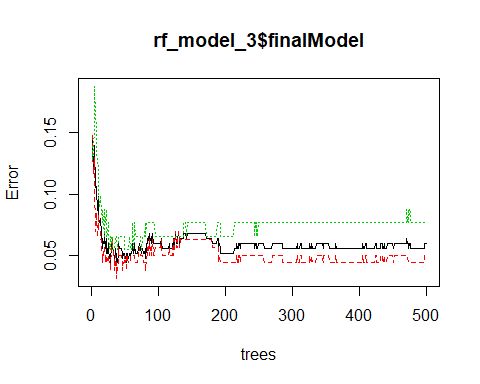
rf\_rocr\_auc <- performance(rf\_rocr\_pred, measure = "auc")  
rf\_auc\_3 <- rf\_rocr\_auc@y.values[[1]]  
rf\_auc\_3

## [1] 0.8598485

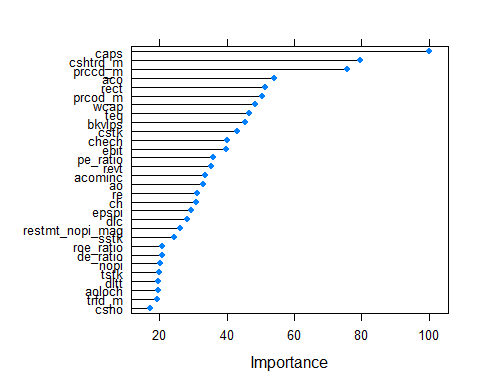
calibration\_curve <- calibration(litigated ~ class\_probabilities\_litigated,  
 data = test\_subset\_ds\_final,  
 class = 1)  
plot(calibration\_curve)



plot(rf\_model\_3$finalModel)



rf\_varImp <- varImp(rf\_model\_3, type = 2)  
plot(rf\_varImp, top = 30)



var\_imp\_ds <- rf\_varImp$importance  
var\_imp\_ds <- var\_imp\_ds %>%   
 as.data.frame() %>%  
 rownames\_to\_column() %>%  
 arrange(desc(Overall))  
var\_imp\_ds

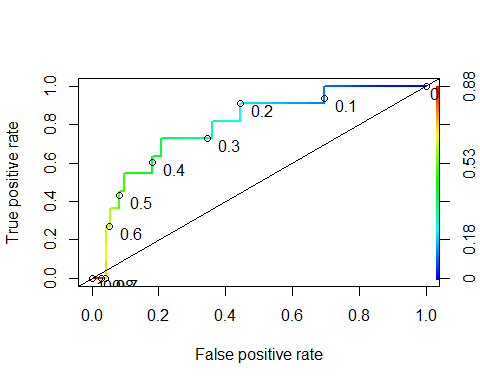
## rowname Overall  
## 1 caps 100.0000000  
## 2 cshtrd\_m 79.4194040  
## 3 prccd\_m 75.6368393  
## 4 aco 54.0231931  
## 5 rect 51.3476987  
## 6 prcod\_m 50.5521603  
## 7 wcap 48.2702510  
## 8 teq 46.5031948  
## 9 bkvlps 45.4259259  
## 10 cstk 43.1477802  
## 11 chech 39.9661312  
## 12 ebit 39.9171856  
## 13 pe\_ratio 35.9751181  
## 14 revt 35.2311891  
## 15 acominc 33.5397407  
## 16 ao 33.0993100  
## 17 re 31.1518851  
## 18 ch 30.9108924  
## 19 epspi 29.3656036  
## 20 dlc 28.3540475  
## 21 restmt\_nopi\_mag 26.2759188  
## 22 sstk 24.4141120  
## 23 roe\_ratio 20.9414244  
## 24 de\_ratio 20.8365221  
## 25 nopi 20.2377197  
## 26 tstk 20.0359429  
## 27 dltt 19.6315370  
## 28 aoloch 19.6117649  
## 29 trfd\_m 19.4111230  
## 30 csho 17.3033236  
## 31 aqc 17.0805341  
## 32 wc\_ratio 17.0044080  
## 33 restmt\_ni\_target 14.5559931  
## 34 restmt\_txt\_target 13.4750466  
## 35 trfm\_m 11.7773451  
## 36 dvt 10.7226866  
## 37 restmt\_epspi\_mag 8.0844478  
## 38 siv 7.7162546  
## 39 restmt\_cogs\_mag 7.4181810  
## 40 sp\_rating\_target 7.3408214  
## 41 restmt\_teq\_mag 5.8596958  
## 42 restmt\_ib\_target 5.8312712  
## 43 restmt\_nopi\_target 4.8757243  
## 44 restmt\_epspi\_target 4.7699079  
## 45 restmt\_teq\_target 4.5993146  
## 46 restmt\_at\_mag 3.8817843  
## 47 restmt\_xint\_mag 3.5342492  
## 48 restmt\_txt\_mag 3.4468243  
## 49 restmt\_ni\_mag 3.1899483  
## 50 restmt\_at\_target 2.5844875  
## 51 restmt\_reuna\_mag 2.3211074  
## 52 restmt\_cogs\_target 1.5612726  
## 53 restmt\_capx\_mag 1.3239253  
## 54 restmt\_capx\_target 0.8517064  
## 55 restmt\_wcap\_mag 0.7616944  
## 56 restmt\_reuna\_target 0.7064857  
## 57 restmt\_xint\_target 0.2319740  
## 58 restmt\_dltt\_target 0.1785672  
## 59 restmt\_wcap\_target 0.0000000

write.csv(var\_imp\_ds, file = "data/var\_imp\_ds.csv", row.names=FALSE)

set.seed(123)  
rf\_model\_4 <- train(litigated ~  
 caps +  
 cshtrd\_m +  
 prccd\_m +   
 aco +  
 rect +  
 prcod\_m +  
 wcap +  
 teq +  
 bkvlps +  
 cstk +  
 chech +  
 ebit +  
 pe\_ratio +  
 revt +  
 acominc +  
 ao +  
 re +  
 ch +  
 epspi +  
 dlc  
 ,   
 data=training\_subset\_ds\_final,   
 method='rf',   
 metric='Accuracy',   
 tuneGrid=rfGrid,   
 trControl=rfControl,   
 ntree = 300)  
  
class\_probabilities <- predict(rf\_model\_4, newdata = test\_subset\_ds\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
test\_subset\_ds\_final$class\_probabilities\_litigated <- class\_probabilities$Yes  
  
rf\_litigated <- ifelse(test\_subset\_ds\_final$litigated == 1, "Yes", "No")  
rf\_class\_probabilities\_litigated <- ifelse(test\_subset\_ds\_final$class\_probabilities\_litigated > 0.50, "Yes", "No")  
rf\_confusion\_matrix <- confusionMatrix(factor(rf\_class\_probabilities\_litigated), factor(rf\_litigated), positive = "Yes")  
rf\_confusion\_matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 66 7  
## Yes 6 4  
##   
## Accuracy : 0.8434   
## 95% CI : (0.7471, 0.9139)  
## No Information Rate : 0.8675   
## P-Value [Acc > NIR] : 0.795   
##   
## Kappa : 0.2915   
##   
## Mcnemar's Test P-Value : 1.000   
##   
## Sensitivity : 0.36364   
## Specificity : 0.91667   
## Pos Pred Value : 0.40000   
## Neg Pred Value : 0.90411   
## Prevalence : 0.13253   
## Detection Rate : 0.04819   
## Detection Prevalence : 0.12048   
## Balanced Accuracy : 0.64015   
##   
## 'Positive' Class : Yes   
##

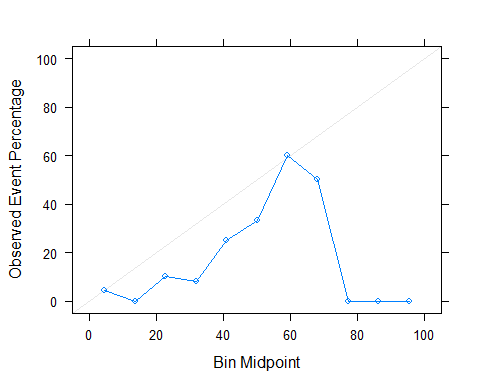
rf\_accuracy\_4 <- rf\_confusion\_matrix$overall["Accuracy"]  
  
  
rf\_rocr\_pred <- prediction(test\_subset\_ds\_final$class\_probabilities\_litigated, test\_subset\_ds\_final$litigated)  
rf\_rocr\_roc <- performance(rf\_rocr\_pred, measure = "tpr", x.measure = "fpr")  
plot(rf\_rocr\_roc,  
colorize = TRUE,  
print.cutoffs.at = seq(0, 1, by = 0.1),  
text.adj = c(-0.5, 1),  
lwd = 2)  
abline(a = 0, b = 1)



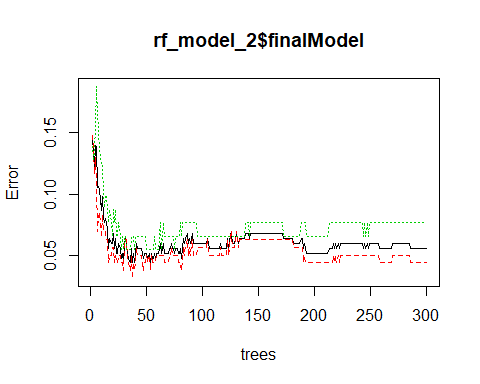
rf\_rocr\_auc <- performance(rf\_rocr\_pred, measure = "auc")  
rf\_auc\_4 <- rf\_rocr\_auc@y.values[[1]]  
rf\_auc\_4

## [1] 0.7954545

calibration\_curve <- calibration(litigated ~ class\_probabilities\_litigated,  
 data = test\_subset\_ds\_final,  
 class = 1)  
plot(calibration\_curve)



plot(rf\_model\_2$finalModel)



auc\_df <- data.frame("algorithm\_method" = c("glm\_1"), "auc" = c(glm\_auc\_1) \* 100, "accuracy" = c(glm\_accuracy\_1) \* 100, "notes" = c("All variables"))  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("glm\_2"), "auc" = c(glm\_auc\_2) \* 100, "accuracy" = c(glm\_accuracy\_2) \* 100,   
 "notes" = c("First Set of Signigficant variables variables"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("glm\_3"), "auc" = c(glm\_auc\_3) \* 100, "accuracy" = c(glm\_accuracy\_3) \* 100,  
 "notes" = c("Second set of signigficant variables variables"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("glmnet\_1"), "auc" = c(glmnet\_auc) \* 100, "accuracy" = c(glmnet\_accuracy\_1) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("glmnet\_1"), "auc" = c(glmnet\_auc) \* 100, "accuracy" = c(glmnet\_accuracy\_2) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("glmnet\_1"), "auc" = c(glmnet\_auc) \* 100, "accuracy" = c(glmnet\_accuracy\_3) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
   
auc\_df\_temp <- data.frame("algorithm\_method" = c("lda\_1"), "auc" = c(lda\_auc\_1) \* 100, "accuracy" = c(lda\_accuracy\_1) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("gbm\_1"), "auc" = c(gbm\_auc\_1) \* 100, "accuracy" = c(gbm\_accuracy\_1) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("tree\_1"), "auc" = c(tree\_auc) \* 100, "accuracy" = c(tree\_accuracy\_1) \* 100,"notes" = c("Default"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
   
auc\_df\_temp <- data.frame("algorithm\_method" = c("rf\_1"), "auc" = c(rf\_auc\_1) \* 100, "accuracy" = c(rf\_accuracy\_1) \* 100,"notes" = c("Trees = 100"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("rf\_2"), "auc" = c(rf\_auc\_2) \* 100, "accuracy" = c(rf\_accuracy\_2) \* 100,"notes" = c("Trees = 300"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("rf\_3"), "auc" = c(rf\_auc\_3) \* 100, "accuracy" = c(rf\_accuracy\_3) \* 100,"notes" = c("Trees = 500"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df\_temp <- data.frame("algorithm\_method" = c("rf\_4"), "auc" = c(rf\_auc\_4) \* 100, "accuracy" = c(rf\_accuracy\_4) \* 100,"notes" = c("Trees = 500 and Only important variables"))  
auc\_df <- rbind(auc\_df,auc\_df\_temp)  
  
auc\_df <- auc\_df %>%   
 arrange(desc(auc))  
  
auc\_df

## algorithm\_method auc accuracy  
## Accuracy9 rf\_1 88.63636 85.54217  
## Accuracy10 rf\_2 86.42677 84.33735  
## Accuracy11 rf\_3 85.98485 84.33735  
## Accuracy12 rf\_4 79.54545 84.33735  
## Accuracy7 gbm\_1 79.54545 74.69880  
## Accuracy2 glm\_3 75.37879 83.13253  
## Accuracy1 glm\_2 68.56061 81.92771  
## Accuracy3 glmnet\_1 62.75253 77.10843  
## Accuracy4 glmnet\_1 62.75253 75.90361  
## Accuracy5 glmnet\_1 62.75253 83.13253  
## Accuracy8 tree\_1 62.50000 69.87952  
## Accuracy glm\_1 59.53283 69.87952  
## Accuracy6 lda\_1 53.09343 69.87952  
## notes  
## Accuracy9 Trees = 100  
## Accuracy10 Trees = 300  
## Accuracy11 Trees = 500  
## Accuracy12 Trees = 500 and Only important variables  
## Accuracy7 Default  
## Accuracy2 Second set of signigficant variables variables  
## Accuracy1 First Set of Signigficant variables variables  
## Accuracy3 Default  
## Accuracy4 Default  
## Accuracy5 Default  
## Accuracy8 Default  
## Accuracy All variables  
## Accuracy6 Default

target\_company\_row\_final

## # A tibble: 1 x 63  
## # Groups: gvkey [1]  
## gvkey tic aco acominc ao aoloch aqc bkvlps caps ch chech  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9777 SJM -0.155 0.107 -0.142 -0.121 0.174 -0.0885 1.54 -0.118 -0.649  
## # ... with 52 more variables: csho <dbl>, cstk <dbl>, dlc <dbl>, dltt <dbl>,  
## # dvt <dbl>, ebit <dbl>, epspi <dbl>, nopi <dbl>, re <dbl>, rect <dbl>,  
## # revt <dbl>, siv <dbl>, sstk <dbl>, teq <dbl>, tstk <dbl>, wcap <dbl>,  
## # restmt\_at\_mag <dbl>, restmt\_capx\_mag <dbl>, restmt\_cogs\_mag <dbl>,  
## # restmt\_epspi\_mag <dbl>, restmt\_ni\_mag <dbl>, restmt\_nopi\_mag <dbl>,  
## # restmt\_reuna\_mag <dbl>, restmt\_teq\_mag <dbl>, restmt\_txt\_mag <dbl>,  
## # restmt\_wcap\_mag <dbl>, restmt\_xint\_mag <dbl>, cshtrd\_m <dbl>,  
## # prccd\_m <dbl>, prcod\_m <dbl>, trfd\_m <dbl>, trfm\_m <dbl>, pe\_ratio <dbl>,  
## # wc\_ratio <dbl>, de\_ratio <dbl>, roe\_ratio <dbl>, sp\_rating\_target <dbl>,  
## # restmt\_at\_target <dbl>, restmt\_capx\_target <dbl>, restmt\_cogs\_target <dbl>,  
## # restmt\_dltt\_target <dbl>, restmt\_epspi\_target <dbl>,  
## # restmt\_ib\_target <dbl>, restmt\_ni\_target <dbl>, restmt\_nopi\_target <dbl>,  
## # restmt\_reuna\_target <dbl>, restmt\_teq\_target <dbl>,  
## # restmt\_txt\_target <dbl>, restmt\_wcap\_target <dbl>,  
## # restmt\_xint\_target <dbl>, litigated <fct>, litigation\_settlement <dbl>

class\_probabilities\_prediction <- predict(rf\_model\_1, newdata = target\_company\_row\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
class\_probabilities\_prediction

## No Yes  
## 1 0.61 0.39

litigated\_probablity <- class\_probabilities\_prediction$Yes  
rf\_litigated <- ifelse(litigated\_probablity > 0.50, "Yes", "No")  
rf\_actual\_litigated\_num <- as.numeric(as.character(target\_company\_row\_final$litigated))  
rf\_actual\_litigated <- ifelse(rf\_actual\_litigated\_num == 1, "Yes", "No")  
rf\_litigated

## [1] "No"

rf\_actual\_litigated

## [1] "No"

litigated\_probablity

## [1] 0.39

class\_probabilities\_prediction <- predict(gbm\_model\_1, newdata = target\_company\_row\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
litigated\_probablity <- class\_probabilities\_prediction$Yes  
gbm\_litigated <- ifelse(litigated\_probablity > 0.50, "Yes", "No")  
gbm\_actual\_litigated\_num <- as.numeric(as.character(target\_company\_row\_final$litigated))  
gbm\_actual\_litigated <- ifelse(rf\_actual\_litigated\_num == 1, "Yes", "No")  
gbm\_litigated

## [1] "No"

gbm\_actual\_litigated

## [1] "No"

class\_probabilities\_prediction <- predict(glm\_model\_3, newdata = target\_company\_row\_final[, -1\*c(target\_var\_index:target\_var\_index)], type = "prob")  
litigated\_probablity <- class\_probabilities\_prediction$Yes  
glm\_litigated <- ifelse(litigated\_probablity > 0.50, "Yes", "No")  
glm\_actual\_litigated\_num <- as.numeric(as.character(target\_company\_row\_final$litigated))  
glm\_actual\_litigated <- ifelse(rf\_actual\_litigated\_num == 1, "Yes", "No")  
glm\_litigated

## [1] "No"

glm\_actual\_litigated

## [1] "No"

library(Rmisc)

## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':  
##   
## rename, round\_any

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

settlement\_training\_ds <- training\_subset\_ds\_final\_settlement %>%  
 filter(!is.na(litigation\_settlement))  
  
settlement\_test\_ds <- test\_subset\_ds\_final\_settlement %>%  
 filter(!is.na(litigation\_settlement))  
  
settlement\_ds <- rbind(settlement\_training\_ds,settlement\_test\_ds)  
settlement\_ds

## # A tibble: 10 x 63  
## # Groups: gvkey [10]  
## gvkey tic aco acominc ao aoloch aqc bkvlps caps ch  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 3505 TAP 126. 31. 301. -84.0 580. 43.8 3623. 841.   
## 2 5301 GAPTQ 39.9 -77.4 143. -42.2 0 -16.0 505. 303.   
## 3 7691 NAFC 23.5 -13.8 10.2 -1.25 29.1 29.5 116. 0.965  
## 4 12785 PPC 98.2 -46.0 105. 31.4 0 4.21 1545. 181.   
## 5 24316 MNST 52.3 -0.106 65.4 -10.5 0 7.60 268. 287.   
## 6 112968 MTEX 5.17 -0.789 7.55 0.881 0 4.71 42.4 18.6   
## 7 149379 SPU 0.568 13.9 14.3 -2.43 0 5.59 59.2 63.7   
## 8 178795 GRO 116. -18.7 11.9 1.82 0.693 3.44 356. 34.6   
## 9 114959 YUII 0.338 3.75 43.3 1.18 0 5.32 115. 34.5   
## 10 163627 DMND 26.8 6.25 25.7 -1.20 154. 14.6 322. 4.48   
## # ... with 53 more variables: chech <dbl>, csho <dbl>, cstk <dbl>, dlc <dbl>,  
## # dltt <dbl>, dvt <dbl>, ebit <dbl>, epspi <dbl>, nopi <dbl>, re <dbl>,  
## # rect <dbl>, revt <dbl>, siv <dbl>, sstk <dbl>, teq <dbl>, tstk <dbl>,  
## # wcap <dbl>, restmt\_at\_mag <dbl>, restmt\_capx\_mag <dbl>,  
## # restmt\_cogs\_mag <dbl>, restmt\_epspi\_mag <dbl>, restmt\_ni\_mag <dbl>,  
## # restmt\_nopi\_mag <dbl>, restmt\_reuna\_mag <dbl>, restmt\_teq\_mag <dbl>,  
## # restmt\_txt\_mag <dbl>, restmt\_wcap\_mag <dbl>, restmt\_xint\_mag <dbl>,  
## # cshtrd\_m <dbl>, prccd\_m <dbl>, prcod\_m <dbl>, trfd\_m <dbl>, trfm\_m <dbl>,  
## # pe\_ratio <dbl>, wc\_ratio <dbl>, de\_ratio <dbl>, roe\_ratio <dbl>,  
## # sp\_rating\_target <dbl>, restmt\_at\_target <dbl>, restmt\_capx\_target <dbl>,  
## # restmt\_cogs\_target <dbl>, restmt\_dltt\_target <dbl>,  
## # restmt\_epspi\_target <dbl>, restmt\_ib\_target <dbl>, restmt\_ni\_target <dbl>,  
## # restmt\_nopi\_target <dbl>, restmt\_reuna\_target <dbl>,  
## # restmt\_teq\_target <dbl>, restmt\_txt\_target <dbl>, restmt\_wcap\_target <dbl>,  
## # restmt\_xint\_target <dbl>, litigated <fct>, litigation\_settlement <dbl>

settlement\_ds\_subset <- subset(settlement\_ds, select = c(gvkey,tic, csho, prccd\_m, prcod\_m,litigated,litigation\_settlement))  
settlement\_ds\_subset$stk\_price\_m <- (settlement\_ds\_subset$prccd\_m + settlement\_ds\_subset$prcod\_m)/2  
settlement\_ds\_subset$market\_cap <- settlement\_ds\_subset$stk\_price\_m \* settlement\_ds\_subset$csho \* 1000000  
settlement\_ds\_subset$settlement\_ratio <- as.numeric(settlement\_ds\_subset$litigation\_settlement/settlement\_ds\_subset$market\_cap)  
settlement\_ds\_subset

## # A tibble: 10 x 10  
## # Groups: gvkey [10]  
## gvkey tic csho prccd\_m prcod\_m litigated litigation\_sett~ stk\_price\_m  
## <int> <fct> <dbl> <dbl> <dbl> <fct> <dbl> <dbl>  
## 1 3505 TAP 183. 47.2 40.0 1 6000000 43.6   
## 2 5301 GAPTQ 54.9 6.99 6.99 1 9000000 6.99  
## 3 7691 NAFC 12.2 29.5 29.5 1 6750000 29.5   
## 4 12785 PPC 237. 9.68 9.67 1 1500000 9.68  
## 5 24316 MNST 127. 61.2 61.1 1 47500000 61.1   
## 6 112968 MTEX 14.6 6.96 6.85 1 11250000 6.91  
## 7 149379 SPU 26.2 2.93 2.94 1 2200000 2.93  
## 8 178795 GRO 55.4 1.31 1.32 1 3750000 1.31  
## 9 114959 YUII 20.2 2.97 4.42 1 2700000 3.70  
## 10 163627 DMND 22.1 35.1 35.1 1 11000000 35.1   
## # ... with 2 more variables: market\_cap <dbl>, settlement\_ratio <dbl>

settlement\_gvkey <- settlement\_ds\_subset$gvkey  
settlement\_gvkey

## [1] 3505 5301 7691 12785 24316 112968 149379 178795 114959 163627

fundamentals\_ds\_settlement <- fundamentals\_ds\_filter %>%  
 filter(gvkey %in% settlement\_gvkey)  
  
fundamentals\_ds\_settlement\_1 <- subset(fundamentals\_ds\_settlement, select = c(gvkey,tic, mkvalt))  
  
fundamentals\_ds\_settlement\_2 <- fundamentals\_ds\_settlement\_1 %>%  
 filter(!is.na(mkvalt))   
  
fundamentals\_ds\_settlement\_3 <- fundamentals\_ds\_settlement\_2 %>%  
 group\_by(gvkey) %>%  
 dplyr::summarize(  
 market\_value = mean(mkvalt)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

settlement\_ds\_subset$market\_value <- NA  
for (row in 1:nrow(settlement\_ds\_subset)){  
 settlement\_item\_gvkey <- as.integer(settlement\_ds\_subset[row, "gvkey"])  
 fundamental\_settlement\_row <- fundamentals\_ds\_settlement\_3 %>%  
 filter(gvkey == settlement\_item\_gvkey)  
 market\_value <- as.numeric(fundamental\_settlement\_row["market\_value"])  
 settlement\_ds\_subset$market\_value[settlement\_ds\_subset$gvkey == settlement\_item\_gvkey] <- market\_value \* 1000000  
}  
settlement\_ds\_subset\_display <- subset(settlement\_ds\_subset, select = c(gvkey,tic, market\_value))  
settlement\_ds\_subset\_display

## # A tibble: 10 x 3  
## # Groups: gvkey [10]  
## gvkey tic market\_value  
## <int> <fct> <dbl>  
## 1 3505 TAP 8830106250  
## 2 5301 GAPTQ 209809000  
## 3 7691 NAFC 377569300  
## 4 12785 PPC 2209474450  
## 5 24316 MNST 8186498150  
## 6 112968 MTEX 29807250  
## 7 149379 SPU 66938975  
## 8 178795 GRO NA  
## 9 114959 YUII 181237500  
## 10 163627 DMND 842417650

settlement\_ds\_subset$market\_value[settlement\_ds\_subset$tic == 'GRO'] <- as.numeric(72720449)  
  
settlement\_ds\_subset$settlement\_ratio\_2 <- as.numeric(settlement\_ds\_subset$litigation\_settlement/settlement\_ds\_subset$market\_value)  
  
settlement\_ds\_subset

## # A tibble: 10 x 12  
## # Groups: gvkey [10]  
## gvkey tic csho prccd\_m prcod\_m litigated litigation\_sett~ stk\_price\_m  
## <int> <fct> <dbl> <dbl> <dbl> <fct> <dbl> <dbl>  
## 1 3505 TAP 183. 47.2 40.0 1 6000000 43.6   
## 2 5301 GAPTQ 54.9 6.99 6.99 1 9000000 6.99  
## 3 7691 NAFC 12.2 29.5 29.5 1 6750000 29.5   
## 4 12785 PPC 237. 9.68 9.67 1 1500000 9.68  
## 5 24316 MNST 127. 61.2 61.1 1 47500000 61.1   
## 6 112968 MTEX 14.6 6.96 6.85 1 11250000 6.91  
## 7 149379 SPU 26.2 2.93 2.94 1 2200000 2.93  
## 8 178795 GRO 55.4 1.31 1.32 1 3750000 1.31  
## 9 114959 YUII 20.2 2.97 4.42 1 2700000 3.70  
## 10 163627 DMND 22.1 35.1 35.1 1 11000000 35.1   
## # ... with 4 more variables: market\_cap <dbl>, settlement\_ratio <dbl>,  
## # market\_value <dbl>, settlement\_ratio\_2 <dbl>

settlement\_ratio\_mean <- mean(settlement\_ds\_subset$settlement\_ratio\_2)  
settlement\_ratio\_medain <- median(settlement\_ds\_subset$settlement\_ratio\_2)  
settlement\_ratio\_mean

## [1] 0.05577476

settlement\_ratio\_medain

## [1] 0.01638755

settlement\_ds\_subset\_display <- subset(settlement\_ds\_subset, select = c(gvkey,tic, csho, prccd\_m, market\_value, litigation\_settlement, settlement\_ratio\_2))  
settlement\_ds\_subset\_display

## # A tibble: 10 x 7  
## # Groups: gvkey [10]  
## gvkey tic csho prccd\_m market\_value litigation\_settlem~ settlement\_ratio~  
## <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 3505 TAP 183. 47.2 8830106250 6000000 0.000679  
## 2 5301 GAPTQ 54.9 6.99 209809000 9000000 0.0429   
## 3 7691 NAFC 12.2 29.5 377569300 6750000 0.0179   
## 4 12785 PPC 237. 9.68 2209474450 1500000 0.000679  
## 5 24316 MNST 127. 61.2 8186498150 47500000 0.00580   
## 6 112968 MTEX 14.6 6.96 29807250 11250000 0.377   
## 7 149379 SPU 26.2 2.93 66938975 2200000 0.0329   
## 8 178795 GRO 55.4 1.31 72720449 3750000 0.0516   
## 9 114959 YUII 20.2 2.97 181237500 2700000 0.0149   
## 10 163627 DMND 22.1 35.1 842417650 11000000 0.0131

write.csv(settlement\_ds\_subset\_display, file = "data/settlement\_ds.csv", row.names=FALSE)

target\_company\_ds\_settlement <- fundamentals\_ds\_filter %>%  
 filter(tic == 'SJM')  
target\_company\_market\_value <- mean(target\_company\_ds\_settlement$mkvalt) \* 1000000  
target\_company\_settlement\_amt <- target\_company\_market\_value \* settlement\_ratio\_mean  
target\_company\_market\_value

## [1] 9090440580

settlement\_ratio\_mean

## [1] 0.05577476

target\_company\_settlement\_amt

## [1] 507017124

settlement\_1 <- settlement\_ds\_subset$litigation\_settlement  
#inference(settlement\_1, est = "mean", type = "ci", method = "theoretical")  
std\_error <- function(x) sd(x)/sqrt(length(x))  
std\_error\_settlement\_amout <- std\_error(settlement\_1)  
std\_error\_settlement\_amout

## [1] 4296698

settlement\_amt\_high <- target\_company\_settlement\_amt + (1.96 \* std\_error\_settlement\_amout)  
settlement\_amt\_low <- target\_company\_settlement\_amt - (1.96 \* std\_error\_settlement\_amout)   
settlement\_amt\_high

## [1] 515438652

settlement\_amt\_low

## [1] 498595595

grouped\_data <- ds\_final %>%  
 group\_by(litigated) %>%  
 dplyr::summarise(  
 mean\_caps = mean(caps),  
 median\_caps = median(caps),  
 mean\_cshtrd\_m = mean(cshtrd\_m),  
 median\_cshtrd\_m = median(cshtrd\_m),  
 mean\_prccd\_m = mean(prccd\_m),  
 median\_prccd\_m = median(prccd\_m),  
 mean\_aco = mean(aco),  
 median\_aco = median(aco),  
 mean\_rect = mean(rect),  
 median\_rect = median(rect),  
 mean\_prcod\_m = mean(prcod\_m),  
 median\_prcod\_m = median(prcod\_m),  
 mean\_wcap = mean(wcap),  
 median\_wcap = median(wcap),  
 mean\_teq = mean(teq),  
 median\_teq = median(teq),  
 mean\_bkvlps = mean(bkvlps),  
 median\_bkvlps = median(bkvlps),  
 mean\_cstk = mean(cstk),  
 median\_cstk = median(cstk),  
 mean\_chech = mean(chech),  
 median\_chech = median(chech),  
 mean\_ebit = mean(ebit),  
 median\_ebit = median(ebit),  
 mean\_pe\_ratio = mean(pe\_ratio),  
 median\_pe\_ratio = median(pe\_ratio),  
 mean\_revt = mean(revt),  
 median\_revt = median(revt),  
 mean\_acominc = mean(acominc),  
 median\_acominc = median(acominc),  
 mean\_ao = mean(ao),  
 median\_ao = median(ao),  
 mean\_re = mean(re),  
 median\_re = median(re),  
 mean\_ch = mean(ch),  
 median\_ch = median(ch),  
 mean\_epspi = mean(epspi),  
 median\_epspi = median(epspi)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

grouped\_data

## # A tibble: 2 x 39  
## litigated mean\_caps median\_caps mean\_cshtrd\_m median\_cshtrd\_m mean\_prccd\_m  
## <fct> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 874. 28.6 746804. 91751. 31.0  
## 2 1 1681. 316. 1935593. 494572. 28.1  
## # ... with 33 more variables: median\_prccd\_m <dbl>, mean\_aco <dbl>,  
## # median\_aco <dbl>, mean\_rect <dbl>, median\_rect <dbl>, mean\_prcod\_m <dbl>,  
## # median\_prcod\_m <dbl>, mean\_wcap <dbl>, median\_wcap <dbl>, mean\_teq <dbl>,  
## # median\_teq <dbl>, mean\_bkvlps <dbl>, median\_bkvlps <dbl>, mean\_cstk <dbl>,  
## # median\_cstk <dbl>, mean\_chech <dbl>, median\_chech <dbl>, mean\_ebit <dbl>,  
## # median\_ebit <dbl>, mean\_pe\_ratio <dbl>, median\_pe\_ratio <dbl>,  
## # mean\_revt <dbl>, median\_revt <dbl>, mean\_acominc <dbl>,  
## # median\_acominc <dbl>, mean\_ao <dbl>, median\_ao <dbl>, mean\_re <dbl>,  
## # median\_re <dbl>, mean\_ch <dbl>, median\_ch <dbl>, mean\_epspi <dbl>,  
## # median\_epspi <dbl>

ggplot(data = ds\_final)+  
 geom\_bar(aes(x = sp\_rating), fill = "#0073c2ff") +  
 xlab("age segments") +  
 ylab("number of patients") +  
 facet\_grid(litigated ~ .) +  
 theme\_minimal()

