ROI on Hand Picked Stocks 2007-2020

Janis Corona

February and March 2020

This is a project that is for now analyzing some hand picked stock to see if a program can be written based on the analysis of how certain stocks perform from 2007-2020. It looks at cyclical patterns of highs and lows, adds in the DOW highs and lows, the unemployment highs and lows, then mean and median values of daily changes various date fields for day of the week and month. The idea is to get the best performing stocks, analyze them with subsets of the worst performing stocks, get the specific features of each stock to describe it as a profit or loss forecasted stock to invest in based on its current stats, and more.

It will then add in the public sentiments for the lows and highs or local minima and maxima of the stock in the best performing set to predict the best time to buy and best to sell respectively, so that you could buy at a low cost and sell at a high cost and keep trading to increase profits of the portfolio.

portfolio <- read.csv('all\_portfolio\_prices.csv', header=TRUE, na.strings=c('',' '),  
 row.names=1)

portfolio$Date <- row.names(portfolio)

Vol <- grep('Volume', colnames(portfolio))  
close <- grep('Close',colnames(portfolio))  
Close <- portfolio[,close]  
Volume <- portfolio[,Vol]  
colnames(Close)

## [1] "TGT.Close" "FTR.Close" "UBSI.Close" "HD.Close" "JPM.Close"   
## [6] "XOM.Close" "CVX.Close" "NSANY.Close" "GNBT.Close" "MGM.Close"   
## [11] "TEVA.Close" "HST.Close" "FCAU.Close" "WFC.Close" "WWE.Close"   
## [16] "INO.Close" "QSR.Close" "GRPN.Close" "SCE.PB.Close" "FFIN.Close"   
## [21] "GOOG.Close" "WM.Close" "ONCY.Close" "S.Close" "GM.Close"   
## [26] "F.Close" "ASCCY.Close" "ARWR.Close" "COST.Close" "AAL.Close"   
## [31] "JWN.Close" "CSSEP.Close" "NUS.Close" "AMC.Close" "ADDYY.Close"   
## [36] "KSS.Close" "MSFT.Close" "LUV.Close" "HMC.Close" "PCG.Close"   
## [41] "DLTR.Close" "KGJI.Close" "NKE.Close" "AMZN.Close" "ROST.Close"   
## [46] "TMUS.Close" "WMT.Close" "TJX.Close" "TM.Close" "PBYI.Close"   
## [51] "T.Close" "JNJ.Close" "C.Close" "EPD.Close" "VZ.Close"   
## [56] "HRB.Close" "NFLX.Close" "AAP.Close" "HOFT.Close" "SIG.Close"   
## [61] "SDC.Close" "RRGB.Close" "M.Close" "JBLU.Close" "YELP.Close"

Remove NAs from the data. The colSums(is.na(Close)) isn’t returning the columns with NAs, so this must be done manually.

Close\_noNAs <- Close[,-c(9,13,17,18,25,27,32,34,46,50,61,65)]  
Volume\_noNAs <- Volume[,-c(9,13,17,18,25,27,32,34,46,50,61,65)]  
  
Close\_noNAs$SCE.PB.Close <- as.numeric(Close\_noNAs$SCE.PB.Close)  
Volume\_noNAs$SCE.PB.Volume <- as.numeric(Volume\_noNAs$SCE.PB.Volume)

Add in a value of the portfolio column for each day’s closing price of all stock that don’t have NAs.

Close\_noNAs$DailyValue <- rowSums(Close\_noNAs,na.rm=TRUE)

Add in a daily change column of the portfolio closing prices.

dayVal <- as.data.frame(Close\_noNAs$DailyValue)  
colnames(dayVal) <- 'previousDayValue'  
zero <- as.data.frame(as.numeric(dayVal$previousDayValue[1]))  
colnames(zero) <- 'previousDayValue'  
prevDay <- rbind(zero,dayVal)  
Close\_noNAs$prevDay <- prevDay[1:length(prevDay$previousDayValue)-1,1]  
dailyChange <- as.data.frame(Close\_noNAs$DailyValue-Close\_noNAs$prevDay)  
colnames(dailyChange) <- 'dailyValueChange'  
  
Close1 <- cbind(Close\_noNAs,dailyChange)

Add a column that gives the return in dollars on initial dollars invested.

Close1$ROI\_dollars <- Close1$DailyValue-Close1$DailyValue[1]

Add some date fields to look at the values by date, day of the week, month, and year in analyzing this data.

Close1$Date <- as.Date.character(row.names(Close1))

Close1$DayOfWeek <- weekdays(as.Date(Close1$Date))

month <- month(as.Date(Close1$Date))  
Month <- month.abb[month]  
Close1$Month <- Month

Add in the year of the Date column.

Year <- year(as.Date(Close1$Date))  
  
Close1$Year <- Year  
  
Close1$MonthYear <- paste(Close1$Month, Close1$Year, sep='-')  
Close1$MonthYear <- as.factor(Close1$MonthYear)

Add in some [unemployment](https://data.bls.gov/pdq/SurveyOutputServlet) information as a column to see how the portfolio is doing by date.

ue <- read.delim('BLS\_unemploymentRates2007-2020.txt', sep=',',header=TRUE,   
 na.strings=c('',' '))  
UE <- ue[,-14]#remove the empty 'Annual' column

Use tidyr to gather the month fields with their respective unemployment rates per month.

gatherMonths <- gather(UE, 'UE\_Month', 'UE\_monthlyRate',2:13)  
  
gatherMonths$MonthYear <- paste(gatherMonths$UE\_Month, gatherMonths$Year, sep='-')  
gatherMonths$MonthYear <- as.factor(gatherMonths$MonthYear)

UE2 <- gatherMonths[,3:4]  
Close2 <- merge(Close1, UE2, by.x='MonthYear', by.y='MonthYear')  
row.names(Close2) <- Close2$Date  
colnames(Close2)[55:58] <- paste('portfolio',colnames(Close2)[55:58], sep='\_')

write.csv(Close2, 'ROI\_UE\_2007\_2020.csv', row.names=FALSE)

Lets add in the volume of trades per day from the Volume\_noNAs data set. But lets add in some fields for total portfolio trades per day,

Volume1 <- Volume\_noNAs  
Volume1$portfolio\_DailyVolume <- rowSums(Volume1, na.rm=TRUE)  
  
dayVol <- as.data.frame(Volume1$portfolio\_DailyVolume)  
colnames(dayVol) <- 'portfolio\_previousDayVolume'  
zero <- as.data.frame(as.numeric(dayVol$portfolio\_previousDayVolume[1]))  
colnames(zero) <- 'portfolio\_previousDayVolume'  
prevDay1 <- rbind(zero,dayVol)  
Volume1$portfolio\_prevDayVolume <-  
 prevDay1[1:(length(prevDay1$portfolio\_previousDayVolume)-1),1]  
  
dailyVolumeChange <- as.data.frame(Volume1$portfolio\_DailyVolume-Volume1$portfolio\_prevDayVolume)  
colnames(dailyVolumeChange) <- 'portfolio\_dailyVolumeChange'  
  
Volume2 <- cbind(Volume1,dailyVolumeChange)  
Volume2$portfolio\_VolumeRatioDaily2Initial <- Volume2$portfolio\_DailyVolume/Volume2$portfolio\_prevDayVolume[1]  
  
Volume2$Date <- as.Date(row.names(Volume2))

stocks <- cbind(Close2, Volume2)  
  
Stocks <- stocks[,c(2:54,64:116,1,55:63,117:120)]  
colnames(Stocks)

## [1] "TGT.Close" "FTR.Close"   
## [3] "UBSI.Close" "HD.Close"   
## [5] "JPM.Close" "XOM.Close"   
## [7] "CVX.Close" "NSANY.Close"   
## [9] "MGM.Close" "TEVA.Close"   
## [11] "HST.Close" "WFC.Close"   
## [13] "WWE.Close" "INO.Close"   
## [15] "SCE.PB.Close" "FFIN.Close"   
## [17] "GOOG.Close" "WM.Close"   
## [19] "ONCY.Close" "S.Close"   
## [21] "F.Close" "ARWR.Close"   
## [23] "COST.Close" "AAL.Close"   
## [25] "JWN.Close" "NUS.Close"   
## [27] "ADDYY.Close" "KSS.Close"   
## [29] "MSFT.Close" "LUV.Close"   
## [31] "HMC.Close" "PCG.Close"   
## [33] "DLTR.Close" "KGJI.Close"   
## [35] "NKE.Close" "AMZN.Close"   
## [37] "ROST.Close" "WMT.Close"   
## [39] "TJX.Close" "TM.Close"   
## [41] "T.Close" "JNJ.Close"   
## [43] "C.Close" "EPD.Close"   
## [45] "VZ.Close" "HRB.Close"   
## [47] "NFLX.Close" "AAP.Close"   
## [49] "HOFT.Close" "SIG.Close"   
## [51] "RRGB.Close" "M.Close"   
## [53] "JBLU.Close" "TGT.Volume"   
## [55] "FTR.Volume" "UBSI.Volume"   
## [57] "HD.Volume" "JPM.Volume"   
## [59] "XOM.Volume" "CVX.Volume"   
## [61] "NSANY.Volume" "MGM.Volume"   
## [63] "TEVA.Volume" "HST.Volume"   
## [65] "WFC.Volume" "WWE.Volume"   
## [67] "INO.Volume" "SCE.PB.Volume"   
## [69] "FFIN.Volume" "GOOG.Volume"   
## [71] "WM.Volume" "ONCY.Volume"   
## [73] "S.Volume" "F.Volume"   
## [75] "ARWR.Volume" "COST.Volume"   
## [77] "AAL.Volume" "JWN.Volume"   
## [79] "NUS.Volume" "ADDYY.Volume"   
## [81] "KSS.Volume" "MSFT.Volume"   
## [83] "LUV.Volume" "HMC.Volume"   
## [85] "PCG.Volume" "DLTR.Volume"   
## [87] "KGJI.Volume" "NKE.Volume"   
## [89] "AMZN.Volume" "ROST.Volume"   
## [91] "WMT.Volume" "TJX.Volume"   
## [93] "TM.Volume" "T.Volume"   
## [95] "JNJ.Volume" "C.Volume"   
## [97] "EPD.Volume" "VZ.Volume"   
## [99] "HRB.Volume" "NFLX.Volume"   
## [101] "AAP.Volume" "HOFT.Volume"   
## [103] "SIG.Volume" "RRGB.Volume"   
## [105] "M.Volume" "JBLU.Volume"   
## [107] "MonthYear" "portfolio\_DailyValue"   
## [109] "portfolio\_prevDay" "portfolio\_dailyValueChange"   
## [111] "portfolio\_ROI\_dollars" "Date"   
## [113] "DayOfWeek" "Month"   
## [115] "Year" "UE\_monthlyRate"   
## [117] "portfolio\_DailyVolume" "portfolio\_prevDayVolume"   
## [119] "portfolio\_dailyVolumeChange" "portfolio\_VolumeRatioDaily2Initial"

Add a value of stock daily to the initial value as a ratio.

Stocks$portfolio\_ValueRatioDaily2Initial <-  
 Stocks$portfolio\_DailyValue/Stocks$portfolio\_DailyValue[1]

Add a field that multiplies the daily value and daily volume ratios compared to the initial value and volume by the unemployment rate.

Stocks$portfolio\_DailyRatios\_X\_UE <-  
 Stocks$portfolio\_ValueRatioDaily2Initial\*Stocks$portfolio\_VolumeRatioDaily2Initial\*Stocks$UE\_monthlyRate

Add an exponential calculation field based on the unemployment rate for rate, and using t=1/12 for 12 months, and a binary value of 1 or 2 where the daily change is positive is assigned a 1 and a negative is a 2. This will make those values decreasing daily have a lower poisson and those values increasing a higher poisson value. This is a modified poisson used for probability of an outcome occuring with a constant rate. Added to rank daily changes based on unemployment rate of each month.

Stocks <- Stocks[complete.cases(Stocks$UE\_monthlyRate),]  
Stocks$dayOfMonth <- day(Stocks$Date)  
dayOfMonth <- day(Stocks$Date)  
ue1 <- Stocks$UE\_monthlyRate  
  
incrDecr <- ifelse(Stocks$portfolio\_dailyValueChange>0,1,2)  
  
Stocks$portfolio\_poisson <- round((exp(-(ue1\*1/12))\*(ue1\*1/12)^incrDecr)/(factorial(incrDecr)),5)  
  
summary(Stocks$portfolio\_poisson)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.03177 0.07392 0.22652 0.19506 0.29808 0.36217

write.csv(Stocks, 'StocksStats.csv', row.names=TRUE)

Make a daily ROI dollars column for each of the stocks in this set.

stocks1 <- Stocks[,1:53]  
colnames(stocks1)

## [1] "TGT.Close" "FTR.Close" "UBSI.Close" "HD.Close" "JPM.Close"   
## [6] "XOM.Close" "CVX.Close" "NSANY.Close" "MGM.Close" "TEVA.Close"   
## [11] "HST.Close" "WFC.Close" "WWE.Close" "INO.Close" "SCE.PB.Close"  
## [16] "FFIN.Close" "GOOG.Close" "WM.Close" "ONCY.Close" "S.Close"   
## [21] "F.Close" "ARWR.Close" "COST.Close" "AAL.Close" "JWN.Close"   
## [26] "NUS.Close" "ADDYY.Close" "KSS.Close" "MSFT.Close" "LUV.Close"   
## [31] "HMC.Close" "PCG.Close" "DLTR.Close" "KGJI.Close" "NKE.Close"   
## [36] "AMZN.Close" "ROST.Close" "WMT.Close" "TJX.Close" "TM.Close"   
## [41] "T.Close" "JNJ.Close" "C.Close" "EPD.Close" "VZ.Close"   
## [46] "HRB.Close" "NFLX.Close" "AAP.Close" "HOFT.Close" "SIG.Close"   
## [51] "RRGB.Close" "M.Close" "JBLU.Close"

stocks1$TGT\_ROI\_dollars <- stocks1$TGT.Close-stocks1$TGT.Close[1]  
stocks1$FTR\_ROI\_dollars <- stocks1$FTR.Close-stocks1$FTR.Close[1]  
stocks1$UBSI\_ROI\_dollars <- stocks1$UBSI.Close-stocks1$UBSI.Close[1]  
stocks1$HD\_ROI\_dollars <- stocks1$HD.Close-stocks1$HD.Close[1]  
stocks1$JPM\_ROI\_dollars <- stocks1$JPM.Close-stocks1$JPM.Close[1]  
  
stocks1$XOM\_ROI\_dollars <- stocks1$XOM.Close-stocks1$XOM.Close[1]  
stocks1$CVX\_ROI\_dollars <- stocks1$CVX.Close-stocks1$CVX.Close[1]  
stocks1$NSANY\_ROI\_dollars <- stocks1$NSANY.Close-stocks1$NSANY.Close[1]  
stocks1$MGM\_ROI\_dollars <- stocks1$MGM.Close-stocks1$MGM.Close[1]  
stocks1$TEVA\_ROI\_dollars <- stocks1$TEVA.Close-stocks1$TEVA.Close[1]  
  
stocks1$HST\_ROI\_dollars <- stocks1$HST.Close-stocks1$HST.Close[1]  
stocks1$WFC\_ROI\_dollars <- stocks1$WFC.Close-stocks1$WFC.Close[1]  
stocks1$WWE\_ROI\_dollars <- stocks1$WWE.Close-stocks1$WWE.Close[1]  
stocks1$INO\_ROI\_dollars <- stocks1$INO.Close-stocks1$INO.Close[1]  
stocks1$SCE.PB\_ROI\_dollars <- stocks1$SCE.PB.Close-stocks1$SCE.PB.Close[1]  
  
stocks1$FFIN\_ROI\_dollars <- stocks1$FFIN.Close-stocks1$FFIN.Close[1]  
stocks1$GOOG\_ROI\_dollars <- stocks1$GOOG.Close-stocks1$GOOG.Close[1]  
stocks1$WM\_ROI\_dollars <- stocks1$WM.Close-stocks1$WM.Close[1]  
stocks1$ONCY\_ROI\_dollars <- stocks1$ONCY.Close-stocks1$ONCY.Close[1]  
stocks1$S\_ROI\_dollars <- stocks1$S.Close-stocks1$S.Close[1]  
  
stocks1$F\_ROI\_dollars <- stocks1$F.Close-stocks1$F.Close[1]  
stocks1$ARWR\_ROI\_dollars <- stocks1$ARWR.Close-stocks1$ARWR.Close[1]  
stocks1$COST\_ROI\_dollars <- stocks1$COST.Close-stocks1$COST.Close[1]  
stocks1$AAL\_ROI\_dollars <- stocks1$AAL.Close-stocks1$AAL.Close[1]  
stocks1$JWN\_ROI\_dollars <- stocks1$JWN.Close-stocks1$JWN.Close[1]  
  
stocks1$NUS\_ROI\_dollars <- stocks1$NUS.Close-stocks1$NUS.Close[1]  
stocks1$HMC\_ROI\_dollars <- stocks1$HMC.Close-stocks1$HMC.Close[1]  
stocks1$AMZN\_ROI\_dollars <- stocks1$AMZN.Close-stocks1$AMZN.Close[1]  
stocks1$T\_ROI\_dollars <- stocks1$T.Close-stocks1$T.Close[1]  
stocks1$HRB\_ROI\_dollars <- stocks1$HRB.Close-stocks1$HRB.Close[1]  
stocks1$RRGB\_ROI\_dollars <- stocks1$RRGB.Close-stocks1$RRGB.Close[1]  
  
stocks1$ADDYY\_ROI\_dollars <- stocks1$ADDYY.Close-stocks1$ADDYY.Close[1]  
stocks1$PCG\_ROI\_dollars <- stocks1$PCG.Close-stocks1$PCG.Close[1]  
stocks1$ROST\_ROI\_dollars <- stocks1$ROST.Close-stocks1$ROST.Close[1]  
stocks1$JNJ\_ROI\_dollars <- stocks1$JNJ.Close-stocks1$JNJ.Close[1]  
stocks1$NFLX\_ROI\_dollars <- stocks1$NFLX.Close-stocks1$NFLX.Close[1]  
stocks1$M\_ROI\_dollars <- stocks1$M.Close-stocks1$M.Close[1]  
  
stocks1$KSS\_ROI\_dollars <- stocks1$KSS.Close-stocks1$KSS.Close[1]  
stocks1$DLTR\_ROI\_dollars <- stocks1$DLTR.Close-stocks1$DLTR.Close[1]  
stocks1$WMT\_ROI\_dollars <- stocks1$WMT.Close-stocks1$WMT.Close[1]  
stocks1$C\_ROI\_dollars <- stocks1$C.Close-stocks1$C.Close[1]  
stocks1$AAP\_ROI\_dollars <- stocks1$AAP.Close-stocks1$AAP.Close[1]  
stocks1$JBLU\_ROI\_dollars <- stocks1$JBLU.Close-stocks1$JBLU.Close[1]  
  
stocks1$MSFT\_ROI\_dollars <- stocks1$MSFT.Close-stocks1$MSFT.Close[1]  
stocks1$KGJI\_ROI\_dollars <- stocks1$KGJI.Close-stocks1$KGJI.Close[1]  
stocks1$EPD\_ROI\_dollars <- stocks1$EPD.Close-stocks1$EPD.Close[1]  
stocks1$TJX\_ROI\_dollars <- stocks1$TJX.Close-stocks1$TJX.Close[1]  
stocks1$HOFT\_ROI\_dollars <- stocks1$HOFT.Close-stocks1$HOFT.Close[1]  
  
stocks1$LUV\_ROI\_dollars <- stocks1$LUV.Close-stocks1$LUV.Close[1]  
stocks1$NKE\_ROI\_dollars <- stocks1$NKE.Close-stocks1$NKE.Close[1]  
stocks1$TM\_ROI\_dollars <- stocks1$TM.Close-stocks1$TM.Close[1]  
stocks1$VZ\_ROI\_dollars <- stocks1$VZ.Close-stocks1$VZ.Close[1]  
stocks1$SIG\_ROI\_dollars <- stocks1$SIG.Close-stocks1$SIG.Close[1]

These are the values of the stock the previous day that will be subtracted from each day to get the daily change from the day before in dollars.

TGTa <- c(0,stocks1$TGT.Close[1:(length(stocks1$TGT.Close)-1)])  
FTRa <- c(0, stocks1$FTR.Close[1:(length(stocks1$TGT.Close)-1)])  
UBSIa <- c(0,stocks1$UBSI.Close[1:(length(stocks1$TGT.Close)-1)])  
HDa <- c(0,stocks1$HD.Close[1:(length(stocks1$TGT.Close)-1)])  
JPMa <- c(0,stocks1$JPM.Close[1:(length(stocks1$TGT.Close)-1)])  
XOMa <- c(0,stocks1$XOM.Close[1:(length(stocks1$TGT.Close)-1)])  
CVXa <- c(0,stocks1$CVX.Close[1:(length(stocks1$TGT.Close)-1)])  
NSANYa <- c(0,stocks1$NSANY.Close[1:(length(stocks1$TGT.Close)-1)])  
MGMa <- c(0,stocks1$MGM.Close[1:(length(stocks1$TGT.Close)-1)])  
TEVAa <- c(0, stocks1$TEVA.Close[1:(length(stocks1$TGT.Close)-1)])  
HSTa <- c(0, stocks1$HST.Close[1:(length(stocks1$TGT.Close)-1)])  
WFCa <- c(0, stocks1$WFC.Close[1:(length(stocks1$TGT.Close)-1)])  
WWEa <- c(0, stocks1$WWE.Close[1:(length(stocks1$TGT.Close)-1)])  
INOa <- c(0,stocks1$INO.Close[1:(length(stocks1$TGT.Close)-1)])  
SCEa <- c(0,stocks1$SCE.PB.Close[1:(length(stocks1$TGT.Close)-1)])  
FFINa <- c(0,stocks1$FFIN.Close[1:(length(stocks1$TGT.Close)-1)])  
GOOGa <- c(0,stocks1$GOOG.Close[1:(length(stocks1$TGT.Close)-1)])  
WMa <- c(0,stocks1$WM.Close[1:(length(stocks1$TGT.Close)-1)])  
ONCYa <- c(0,stocks1$ONCY.Close[1:(length(stocks1$TGT.Close)-1)])  
Sa <- c(0,stocks1$S.Close[1:(length(stocks1$TGT.Close)-1)])  
Fa <- c(0,stocks1$F.Close[1:(length(stocks1$TGT.Close)-1)])  
ARWRa <- c(0,stocks1$ARWR.Close[1:(length(stocks1$TGT.Close)-1)])  
COSTa <- c(0,stocks1$COST.Close[1:(length(stocks1$TGT.Close)-1)])  
AALa <- c(0,stocks1$AAL.Close[1:(length(stocks1$TGT.Close)-1)])  
JWNa <- c(0,stocks1$JWN.Close[1:(length(stocks1$TGT.Close)-1)])  
NUSa <- c(0,stocks1$NUS.Close[1:(length(stocks1$TGT.Close)-1)])  
ADDYYa <- c(0,stocks1$ADDYY.Close[1:(length(stocks1$TGT.Close)-1)])  
KSSa <- c(0,stocks1$KSS.Close[1:(length(stocks1$TGT.Close)-1)])  
MSFTa <- c(0,stocks1$MSFT.Close[1:(length(stocks1$TGT.Close)-1)])  
LUVa <- c(0,stocks1$LUV.Close[1:(length(stocks1$TGT.Close)-1)])  
HMCa <- c(0,stocks1$HMC.Close[1:(length(stocks1$TGT.Close)-1)])  
PCGa <- c(0,stocks1$PCG.Close[1:(length(stocks1$TGT.Close)-1)])  
DLTRa <- c(0,stocks1$DLTR.Close[1:(length(stocks1$TGT.Close)-1)])  
KGJIa <- c(0,stocks1$KGJI.Close[1:(length(stocks1$TGT.Close)-1)])  
NKEa <- c(0,stocks1$NKE.Close[1:(length(stocks1$TGT.Close)-1)])  
AMZNa <- c(0,stocks1$AMZN.Close[1:(length(stocks1$TGT.Close)-1)])  
ROSTa <- c(0,stocks1$ROST.Close[1:(length(stocks1$TGT.Close)-1)])  
WMTa <- c(0,stocks1$WMT.Close[1:(length(stocks1$TGT.Close)-1)])  
TJXa <- c(0,stocks1$TJX.Close[1:(length(stocks1$TGT.Close)-1)])  
TMa <- c(0,stocks1$TM.Close[1:(length(stocks1$TGT.Close)-1)])  
Ta <- c(0,stocks1$T.Close[1:(length(stocks1$TGT.Close)-1)])  
JNJa <- c(0,stocks1$JNJ.Close[1:(length(stocks1$TGT.Close)-1)])  
Ca <- c(0,stocks1$C.Close[1:(length(stocks1$TGT.Close)-1)])  
EPDa <- c(0,stocks1$EPD.Close[1:(length(stocks1$TGT.Close)-1)])  
VZa <- c(0,stocks1$VZ.Close[1:(length(stocks1$TGT.Close)-1)])  
HRBa <- c(0,stocks1$HRB.Close[1:(length(stocks1$TGT.Close)-1)])  
NFLXa <- c(0,stocks1$NFLX.Close[1:(length(stocks1$TGT.Close)-1)])  
AAPa <- c(0,stocks1$AAP.Close[1:(length(stocks1$TGT.Close)-1)])  
HOFTa <- c(0,stocks1$HOFT.Close[1:(length(stocks1$TGT.Close)-1)])  
SIGa <- c(0,stocks1$SIG.Close[1:(length(stocks1$TGT.Close)-1)])  
RRGBa <- c(0,stocks1$RRGB.Close[1:(length(stocks1$TGT.Close)-1)])  
Ma <- c(0,stocks1$M.Close[1:(length(stocks1$TGT.Close)-1)])  
JBLUa <- c(0,stocks1$JBLU.Close[1:(length(stocks1$TGT.Close)-1)])

This creates the DailyChange per stock columns.

stocks1$TGT\_dailyChange <- stocks1$TGT.Close-TGTa  
stocks1$FTR\_dailyChange <- stocks1$FTR.Close-FTRa  
stocks1$UBSI\_dailyChange <- stocks1$UBSI.Close-UBSIa  
stocks1$HD\_dailyChange <- stocks1$HD.Close-HDa  
stocks1$JPM\_dailyChange <- stocks1$JPM.Close-JPMa  
  
stocks1$XOM\_dailyChange <- stocks1$XOM.Close-XOMa  
stocks1$CVX\_dailyChange <- stocks1$CVX.Close-CVXa  
stocks1$NSANY\_dailyChange <- stocks1$NSANY.Close-NSANYa  
stocks1$MGM\_dailyChange <- stocks1$MGM.Close-MGMa  
stocks1$TEVA\_dailyChange <- stocks1$TEVA.Close-TEVAa  
  
stocks1$HST\_dailyChange <- stocks1$HST.Close-HSTa  
stocks1$WFC\_dailyChange <- stocks1$WFC.Close-WFCa  
stocks1$WWE\_dailyChange <- stocks1$WWE.Close-WWEa  
stocks1$INO\_dailyChange <- stocks1$INO.Close-INOa  
stocks1$SCE.PB\_dailyChange <- stocks1$SCE.PB.Close-SCEa  
  
stocks1$FFIN\_dailyChange <- stocks1$FFIN.Close-FFINa  
stocks1$GOOG\_dailyChange <- stocks1$GOOG.Close-GOOGa  
stocks1$WM\_dailyChange <- stocks1$WM.Close-WMa  
stocks1$ONCY\_dailyChange <- stocks1$ONCY.Close-ONCYa  
stocks1$S\_dailyChange <- stocks1$S.Close-Sa  
  
stocks1$F\_dailyChange <- stocks1$F.Close-Fa  
stocks1$ARWR\_dailyChange <- stocks1$ARWR.Close-ARWRa  
stocks1$COST\_dailyChange <- stocks1$COST.Close-COSTa  
stocks1$AAL\_dailyChange <- stocks1$AAL.Close-AALa  
stocks1$JWN\_dailyChange <- stocks1$JWN.Close-JWNa  
  
stocks1$NUS\_dailyChange <- stocks1$NUS.Close-NUSa  
stocks1$HMC\_dailyChange <- stocks1$HMC.Close-HMCa  
stocks1$AMZN\_dailyChange <- stocks1$AMZN.Close-AMZNa  
stocks1$T\_dailyChange <- stocks1$T.Close-Ta  
stocks1$HRB\_dailyChange <- stocks1$HRB.Close-HRBa  
stocks1$RRGB\_dailyChange <- stocks1$RRGB.Close-RRGBa  
  
stocks1$ADDYY\_dailyChange <- stocks1$ADDYY.Close-ADDYYa  
stocks1$PCG\_dailyChange <- stocks1$PCG.Close-PCGa  
stocks1$ROST\_dailyChange <- stocks1$ROST.Close-ROSTa  
stocks1$JNJ\_dailyChange <- stocks1$JNJ.Close-JNJa  
stocks1$NFLX\_dailyChange <- stocks1$NFLX.Close-NFLXa  
stocks1$M\_dailyChange <- stocks1$M.Close-Ma  
  
stocks1$KSS\_dailyChange <- stocks1$KSS.Close-KSSa  
stocks1$DLTR\_dailyChange <- stocks1$DLTR.Close-DLTRa  
stocks1$WMT\_dailyChange <- stocks1$WMT.Close-WMTa  
stocks1$C\_dailyChange <- stocks1$C.Close-Ca  
stocks1$AAP\_dailyChange <- stocks1$AAP.Close-AAPa  
stocks1$JBLU\_dailyChange <- stocks1$JBLU.Close-JBLUa  
  
stocks1$MSFT\_dailyChange <- stocks1$MSFT.Close-MSFTa  
stocks1$KGJI\_dailyChange <- stocks1$KGJI.Close-KGJIa  
stocks1$EPD\_dailyChange <- stocks1$EPD.Close-EPDa  
stocks1$TJX\_dailyChange <- stocks1$TJX.Close-TJXa  
stocks1$HOFT\_dailyChange <- stocks1$HOFT.Close-HOFTa  
  
stocks1$LUV\_dailyChange <- stocks1$LUV.Close-LUVa  
stocks1$NKE\_dailyChange <- stocks1$NKE.Close-NKEa  
stocks1$TM\_dailyChange <- stocks1$TM.Close-TMa  
stocks1$VZ\_dailyChange <- stocks1$VZ.Close-VZa  
stocks1$SIG\_dailyChange <- stocks1$SIG.Close-SIGa

Combine the stocks1 stats of ROI and daily change in dollars per stock to the stocks stats data table.

stocks2 <- stocks1[,-c(1:53)]  
StocksSTATS <- cbind(Stocks,stocks2)

All the columns we now have are:

StocksSTATS <- StocksSTATS[,c(1:106,125:230,107:124)]  
colnames(StocksSTATS)

## [1] "TGT.Close" "FTR.Close"   
## [3] "UBSI.Close" "HD.Close"   
## [5] "JPM.Close" "XOM.Close"   
## [7] "CVX.Close" "NSANY.Close"   
## [9] "MGM.Close" "TEVA.Close"   
## [11] "HST.Close" "WFC.Close"   
## [13] "WWE.Close" "INO.Close"   
## [15] "SCE.PB.Close" "FFIN.Close"   
## [17] "GOOG.Close" "WM.Close"   
## [19] "ONCY.Close" "S.Close"   
## [21] "F.Close" "ARWR.Close"   
## [23] "COST.Close" "AAL.Close"   
## [25] "JWN.Close" "NUS.Close"   
## [27] "ADDYY.Close" "KSS.Close"   
## [29] "MSFT.Close" "LUV.Close"   
## [31] "HMC.Close" "PCG.Close"   
## [33] "DLTR.Close" "KGJI.Close"   
## [35] "NKE.Close" "AMZN.Close"   
## [37] "ROST.Close" "WMT.Close"   
## [39] "TJX.Close" "TM.Close"   
## [41] "T.Close" "JNJ.Close"   
## [43] "C.Close" "EPD.Close"   
## [45] "VZ.Close" "HRB.Close"   
## [47] "NFLX.Close" "AAP.Close"   
## [49] "HOFT.Close" "SIG.Close"   
## [51] "RRGB.Close" "M.Close"   
## [53] "JBLU.Close" "TGT.Volume"   
## [55] "FTR.Volume" "UBSI.Volume"   
## [57] "HD.Volume" "JPM.Volume"   
## [59] "XOM.Volume" "CVX.Volume"   
## [61] "NSANY.Volume" "MGM.Volume"   
## [63] "TEVA.Volume" "HST.Volume"   
## [65] "WFC.Volume" "WWE.Volume"   
## [67] "INO.Volume" "SCE.PB.Volume"   
## [69] "FFIN.Volume" "GOOG.Volume"   
## [71] "WM.Volume" "ONCY.Volume"   
## [73] "S.Volume" "F.Volume"   
## [75] "ARWR.Volume" "COST.Volume"   
## [77] "AAL.Volume" "JWN.Volume"   
## [79] "NUS.Volume" "ADDYY.Volume"   
## [81] "KSS.Volume" "MSFT.Volume"   
## [83] "LUV.Volume" "HMC.Volume"   
## [85] "PCG.Volume" "DLTR.Volume"   
## [87] "KGJI.Volume" "NKE.Volume"   
## [89] "AMZN.Volume" "ROST.Volume"   
## [91] "WMT.Volume" "TJX.Volume"   
## [93] "TM.Volume" "T.Volume"   
## [95] "JNJ.Volume" "C.Volume"   
## [97] "EPD.Volume" "VZ.Volume"   
## [99] "HRB.Volume" "NFLX.Volume"   
## [101] "AAP.Volume" "HOFT.Volume"   
## [103] "SIG.Volume" "RRGB.Volume"   
## [105] "M.Volume" "JBLU.Volume"   
## [107] "TGT\_ROI\_dollars" "FTR\_ROI\_dollars"   
## [109] "UBSI\_ROI\_dollars" "HD\_ROI\_dollars"   
## [111] "JPM\_ROI\_dollars" "XOM\_ROI\_dollars"   
## [113] "CVX\_ROI\_dollars" "NSANY\_ROI\_dollars"   
## [115] "MGM\_ROI\_dollars" "TEVA\_ROI\_dollars"   
## [117] "HST\_ROI\_dollars" "WFC\_ROI\_dollars"   
## [119] "WWE\_ROI\_dollars" "INO\_ROI\_dollars"   
## [121] "SCE.PB\_ROI\_dollars" "FFIN\_ROI\_dollars"   
## [123] "GOOG\_ROI\_dollars" "WM\_ROI\_dollars"   
## [125] "ONCY\_ROI\_dollars" "S\_ROI\_dollars"   
## [127] "F\_ROI\_dollars" "ARWR\_ROI\_dollars"   
## [129] "COST\_ROI\_dollars" "AAL\_ROI\_dollars"   
## [131] "JWN\_ROI\_dollars" "NUS\_ROI\_dollars"   
## [133] "HMC\_ROI\_dollars" "AMZN\_ROI\_dollars"   
## [135] "T\_ROI\_dollars" "HRB\_ROI\_dollars"   
## [137] "RRGB\_ROI\_dollars" "ADDYY\_ROI\_dollars"   
## [139] "PCG\_ROI\_dollars" "ROST\_ROI\_dollars"   
## [141] "JNJ\_ROI\_dollars" "NFLX\_ROI\_dollars"   
## [143] "M\_ROI\_dollars" "KSS\_ROI\_dollars"   
## [145] "DLTR\_ROI\_dollars" "WMT\_ROI\_dollars"   
## [147] "C\_ROI\_dollars" "AAP\_ROI\_dollars"   
## [149] "JBLU\_ROI\_dollars" "MSFT\_ROI\_dollars"   
## [151] "KGJI\_ROI\_dollars" "EPD\_ROI\_dollars"   
## [153] "TJX\_ROI\_dollars" "HOFT\_ROI\_dollars"   
## [155] "LUV\_ROI\_dollars" "NKE\_ROI\_dollars"   
## [157] "TM\_ROI\_dollars" "VZ\_ROI\_dollars"   
## [159] "SIG\_ROI\_dollars" "TGT\_dailyChange"   
## [161] "FTR\_dailyChange" "UBSI\_dailyChange"   
## [163] "HD\_dailyChange" "JPM\_dailyChange"   
## [165] "XOM\_dailyChange" "CVX\_dailyChange"   
## [167] "NSANY\_dailyChange" "MGM\_dailyChange"   
## [169] "TEVA\_dailyChange" "HST\_dailyChange"   
## [171] "WFC\_dailyChange" "WWE\_dailyChange"   
## [173] "INO\_dailyChange" "SCE.PB\_dailyChange"   
## [175] "FFIN\_dailyChange" "GOOG\_dailyChange"   
## [177] "WM\_dailyChange" "ONCY\_dailyChange"   
## [179] "S\_dailyChange" "F\_dailyChange"   
## [181] "ARWR\_dailyChange" "COST\_dailyChange"   
## [183] "AAL\_dailyChange" "JWN\_dailyChange"   
## [185] "NUS\_dailyChange" "HMC\_dailyChange"   
## [187] "AMZN\_dailyChange" "T\_dailyChange"   
## [189] "HRB\_dailyChange" "RRGB\_dailyChange"   
## [191] "ADDYY\_dailyChange" "PCG\_dailyChange"   
## [193] "ROST\_dailyChange" "JNJ\_dailyChange"   
## [195] "NFLX\_dailyChange" "M\_dailyChange"   
## [197] "KSS\_dailyChange" "DLTR\_dailyChange"   
## [199] "WMT\_dailyChange" "C\_dailyChange"   
## [201] "AAP\_dailyChange" "JBLU\_dailyChange"   
## [203] "MSFT\_dailyChange" "KGJI\_dailyChange"   
## [205] "EPD\_dailyChange" "TJX\_dailyChange"   
## [207] "HOFT\_dailyChange" "LUV\_dailyChange"   
## [209] "NKE\_dailyChange" "TM\_dailyChange"   
## [211] "VZ\_dailyChange" "SIG\_dailyChange"   
## [213] "MonthYear" "portfolio\_DailyValue"   
## [215] "portfolio\_prevDay" "portfolio\_dailyValueChange"   
## [217] "portfolio\_ROI\_dollars" "Date"   
## [219] "DayOfWeek" "Month"   
## [221] "Year" "UE\_monthlyRate"   
## [223] "portfolio\_DailyVolume" "portfolio\_prevDayVolume"   
## [225] "portfolio\_dailyVolumeChange" "portfolio\_VolumeRatioDaily2Initial"  
## [227] "portfolio\_ValueRatioDaily2Initial" "portfolio\_DailyRatios\_X\_UE"   
## [229] "dayOfMonth" "portfolio\_poisson"

write.csv(StocksSTATS, 'STOCKS\_STATS.csv', row.names=TRUE)

Lets us pick one stock, look at the stats we added for that stock and then pull out some googled articles of that stock as a company in the news since 2007 till today’s date of Feb. 18, 2020 to compare the sentiments on the company with words that we will count the number of times the company is in the news, the comments by readers, zoom in on the dates of those articles, and see how the company behaved. Lets choose the highest ROI in dollars out of our stocks and compare it to the lowest ROI in dollars.

m <- StocksSTATS[order(StocksSTATS$Date, decreasing=FALSE)[length(StocksSTATS$Date)], 107:159]  
t <- as.data.frame(t(m))  
colnames(t) <- row.names(m)  
t$StockROI <- row.names(t)  
  
Troi <- t[order(t$'2020-01-31', decreasing=TRUE),]  
  
mostLeast <- rbind(head(Troi,3),tail(Troi,3))  
mostLeast <- na.omit(mostLeast)  
mostLeast

## 2020-01-31 StockROI  
## AMZN\_ROI\_dollars 1968.300 AMZN\_ROI\_dollars  
## GOOG\_ROI\_dollars 1205.821 GOOG\_ROI\_dollars  
## SCE.PB\_ROI\_dollars 679.000 SCE.PB\_ROI\_dollars  
## MGM\_ROI\_dollars -40.520 MGM\_ROI\_dollars  
## FTR\_ROI\_dollars -225.200 FTR\_ROI\_dollars  
## C\_ROI\_dollars -436.090 C\_ROI\_dollars

The above table shows the three highest returns on investment and the three lowest since Jan 3, 2007 to Jan 31, 2020. Lets use the lowest stock for now (C is Citigroup bank), because AMZN (Amazon) is always in the news and it would fluctuate a lot I would think, but we could look at the quartiles for each and get the news releases of each date where the stock was in that quartile range, look at the median ROI, the min and max too, and cross referencing with the other stat fields.

amzn <- grep('AMZN', colnames(StocksSTATS))  
c <- grep('^C[.|\_]', colnames(StocksSTATS))  
C\_stock <- StocksSTATS[,c(c,213:230)]  
amzn\_stock <- StocksSTATS[,c(amzn,213:230)]

Citigroup is our C\_stock table and Amazon is our amzn\_stock table. Lets look at the daily ratios of volume and ROI in dollars times the unemployment rate column and the day of the week and day of the year and poisson columns.

ggplot(data = C\_stock, aes(x=Year, y=C\_ROI\_dollars,group=DayOfWeek)) +  
 geom\_line(aes(color=DayOfWeek))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup 2007-2020')+  
 ylab('ROI dollars Values')

## Warning in pal\_name(palette, type): Unknown palette paired



We can see from the plot above that buying Citigroup stock anywhere before 2010, was a bad idea. But we also see that the stock would have been good to buy around 2010-2016, as it overall increased its return on investment in dollars initially invested.

Lets look at the years from 2016-2020 to see this plotted Citigroup stock.

y2015plus <- subset(C\_stock, C\_stock$Year>2014)  
  
ggplot(data = y2015plus, aes(x=Year, y=C.Close,group=DayOfWeek)) +  
 geom\_line(aes(color=DayOfWeek))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Stock Value in Dollars 2015-2020')+  
 ylab('Stock Value')

## Warning in pal\_name(palette, type): Unknown palette paired



We see from the above plot that Citigroup was good to buy at the start of 2016 or 2019 if you want to see an increase all year long, but in 2017-2018 it decreased.Overall, if investing since 2016, the stock increased from the high $40 to the mid-high $70 range. This would be good to cross reference with unemployment rates and the news articles online text mined for public sentiment on Citigroup.

Lets look at amazon for the same quick plotted analysis as done with Citigroup.

ggplot(data = amzn\_stock, aes(x=Year, y=AMZN\_ROI\_dollars,group=DayOfWeek)) +  
 geom\_line(aes(color=DayOfWeek))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('AMAZON 2007-2020')+  
 ylab('ROI dollars Values')

## Warning in pal\_name(palette, type): Unknown palette paired



We can see from the plot above that buying AMAZON stock anywhere before 2010, was a great idea. But we also see that the stock would have been good to buy around 2010-2018 or 2019 but not in 2018, as it overall increased its return on investment in dollars initially invested.In 2018, you bought high and it decreased the entire year. This would be great to see what happened in 2018 with the value. So we will.

Lets look at the years from 2018-2020 to see this plotted Citigroup stock.

y2015plus <- subset(amzn\_stock, amzn\_stock$Year>2017)  
  
ggplot(data = y2015plus, aes(x=Year, y=AMZN.Close,group=DayOfWeek)) +  
 geom\_line(aes(color=DayOfWeek))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('AMAZON Stock Value in Dollars 2018-2020')+  
 ylab('Stock Value')

## Warning in pal\_name(palette, type): Unknown palette paired



The chart above shows how the value in dollars and day of the week from 2018-2020 decreases in 2018 and increases in 2019. If you bought in 2018, you lost money the entire year, but you gained it back in 2019 plus some additional earnings.

Lets group by the day of the month in this time series of the Citigroup stock and get the median value for the volumne of stocks traded for Citigroup by days 1-31 of the month.

v1 <- as.vector(colnames(C\_stock)[2])  
Citi <- C\_stock %>% group\_by(dayOfMonth) %>% summarise\_at(vars(v1), median,  
 na.rm=T)  
Citi <- as.data.frame(Citi)  
colnames(Citi)[2] <- 'Citi\_Median\_Volume'  
Citi <- Citi[order(Citi$Citi\_Median\_Volume, decreasing=T),]  
headTail\_Citi\_volume <- rbind(head(Citi,3), tail(Citi,3))  
headTail\_Citi\_volume

## dayOfMonth Citi\_Median\_Volume  
## 16 16 22388100  
## 31 31 22302200  
## 3 3 21221500  
## 25 25 17960700  
## 20 20 17548500  
## 2 2 17134600

From the above table we see that the most volume of trades for Citigroup is at the middle and end of the month, and the lowest volume of trades are at the beginning of the new month and the third week of the month.

Lets look at the statistics of citigroup.

summary(C\_stock)

## C.Close C.Volume C\_ROI\_dollars C\_dailyChange   
## Min. : 10.20 Min. : 1005100 Min. :-500.3 Min. :-298.300   
## 1st Qu.: 41.80 1st Qu.: 13019600 1st Qu.:-468.7 1st Qu.: -0.680   
## Median : 51.49 Median : 19493900 Median :-459.0 Median : -0.010   
## Mean : 93.38 Mean : 26987469 Mean :-417.1 Mean : 0.021   
## 3rd Qu.: 69.46 3rd Qu.: 33280800 3rd Qu.:-441.0 3rd Qu.: 0.650   
## Max. :552.50 Max. :377263800 Max. : 42.0 Max. : 510.500   
##   
## MonthYear portfolio\_DailyValue portfolio\_prevDay  
## Aug-2007: 23 Min. :1229 Min. :1229   
## Aug-2011: 23 1st Qu.:2821 1st Qu.:2821   
## Aug-2012: 23 Median :3542 Median :3541   
## Aug-2016: 23 Mean :3988 Mean :3986   
## Aug-2017: 23 3rd Qu.:5104 3rd Qu.:5104   
## Aug-2018: 23 Max. :7910 Max. :7910   
## (Other) :3155   
## portfolio\_dailyValueChange portfolio\_ROI\_dollars Date   
## Min. :-1014.322 Min. :-1748.9 Min. :2007-01-03   
## 1st Qu.: -39.065 1st Qu.: -157.4 1st Qu.:2010-04-12   
## Median : 2.276 Median : 563.9 Median :2013-07-18   
## Mean : 1.475 Mean : 1009.6 Mean :2013-07-16   
## 3rd Qu.: 43.517 3rd Qu.: 2126.4 3rd Qu.:2016-10-21   
## Max. : 1025.453 Max. : 4931.7 Max. :2020-01-31   
##   
## DayOfWeek Month Year UE\_monthlyRate   
## Length:3293 Length:3293 Min. :2007 Min. : 3.500   
## Class :character Class :character 1st Qu.:2010 1st Qu.: 4.600   
## Mode :character Mode :character Median :2013 Median : 5.600   
## Mean :2013 Mean : 6.282   
## 3rd Qu.:2016 3rd Qu.: 8.200   
## Max. :2020 Max. :10.000   
##   
## portfolio\_DailyVolume portfolio\_prevDayVolume portfolio\_dailyVolumeChange  
## Min. :1.133e+08 Min. :1.133e+08 Min. :-714176400   
## 1st Qu.:3.370e+08 1st Qu.:3.370e+08 1st Qu.: -50722061   
## Median :4.194e+08 Median :4.196e+08 Median : 250560   
## Mean :4.752e+08 Mean :4.753e+08 Mean : -55791   
## 3rd Qu.:5.716e+08 3rd Qu.:5.716e+08 3rd Qu.: 50561500   
## Max. :1.611e+09 Max. :1.611e+09 Max. : 620907605   
##   
## portfolio\_VolumeRatioDaily2Initial portfolio\_ValueRatioDaily2Initial  
## Min. :0.1981 Min. :0.4236   
## 1st Qu.:0.5891 1st Qu.:0.9720   
## Median :0.7333 Median :1.2206   
## Mean :0.8307 Mean :1.3742   
## 3rd Qu.:0.9992 3rd Qu.:1.7591   
## Max. :2.8163 Max. :2.7259   
##   
## portfolio\_DailyRatios\_X\_UE dayOfMonth portfolio\_poisson  
## Min. : 0.9658 Min. : 1.00 Min. :0.03177   
## 1st Qu.: 4.4923 1st Qu.: 8.00 1st Qu.:0.07392   
## Median : 5.6528 Median :16.00 Median :0.22652   
## Mean : 6.4285 Mean :15.74 Mean :0.19506   
## 3rd Qu.: 7.8497 3rd Qu.:23.00 3rd Qu.:0.29808   
## Max. :24.2627 Max. :31.00 Max. :0.36217   
##

From the above summary statistics of Citigroup, we see the min, quantiles, median, mean, and max numeric values as well as length and class type for the non-numeric features of this data set.

Some interesting insights into the above table are that considering an initial investment of 510 USD, the return on the initial investment in dollars is almost the entire amount invested but not quite. Definitely about 80% from the quantile and statistics on the ROI column.

The daily changes fluctuated from a loss of 298 USD in one day to a profit of 510 USD on another day. These are good indicators of where to look on these days, to see if the public sentiment on these dates for Citigroup would indicate more people getting rid of their Citi stock or buying up more of it.

Also, the max and min volume of stock is much more and less respectively than the median volume of trades for this Citigroup stock. These dates for information would also be an interesting place to start to find a pattern with buying/selling stock and combining web scraped text from news articles and comments about Citigroup on those dates.

First, we should grab those points of interest in the data and create a table to compare these values.

C\_stock\_minmaxValueChanges <- subset(C\_stock,  
 C\_stock$C\_dailyChange==min(C\_stock$C\_dailyChange) |  
 C\_stock$C\_dailyChange==max(C\_stock$C\_dailyChange) |  
 C\_stock$C.Volume==min(C\_stock$C.Volume) |  
 C\_stock$C.Volume==max(C\_stock$C.Volume))  
C\_stock\_minmaxValueChanges

## C.Close C.Volume C\_ROI\_dollars C\_dailyChange MonthYear  
## 2007-04-02 510.50 2282100 0.00 510.500000 Apr-2007  
## 2013-04-02 44.11 1005100 -466.39 0.320000 Apr-2013  
## 2015-12-28 52.38 377263800 -458.12 -0.329998 Dec-2015  
## 2008-06-02 214.60 15302800 -295.90 -298.300018 Jun-2008  
## portfolio\_DailyValue portfolio\_prevDay portfolio\_dailyValueChange  
## 2007-04-02 2901.650 2891.963 9.686608  
## 2013-04-02 3433.938 3354.901 79.037872  
## 2015-12-28 5005.455 4984.970 20.485009  
## 2008-06-02 3120.541 3144.698 -24.157199  
## portfolio\_ROI\_dollars Date DayOfWeek Month Year UE\_monthlyRate  
## 2007-04-02 -76.28907 2007-04-02 Monday Apr 2007 4.5  
## 2013-04-02 455.99978 2013-04-02 Tuesday Apr 2013 7.6  
## 2015-12-28 2027.51641 2015-12-28 Monday Dec 2015 5.0  
## 2008-06-02 142.60220 2008-06-02 Monday Jun 2008 5.6  
## portfolio\_DailyVolume portfolio\_prevDayVolume  
## 2007-04-02 572035712 572035712  
## 2013-04-02 258084601 330998801  
## 2015-12-28 975152259 752607802  
## 2008-06-02 464823559 265152951  
## portfolio\_dailyVolumeChange portfolio\_VolumeRatioDaily2Initial  
## 2007-04-02 0 1.0000000  
## 2013-04-02 -72914200 0.4511687  
## 2015-12-28 222544457 1.7047052  
## 2008-06-02 199670608 0.8125779  
## portfolio\_ValueRatioDaily2Initial portfolio\_DailyRatios\_X\_UE  
## 2007-04-02 1.000000 4.500000  
## 2013-04-02 1.183444 4.057888  
## 2015-12-28 1.725038 14.703404  
## 2008-06-02 1.075437 4.893707  
## dayOfMonth portfolio\_poisson  
## 2007-04-02 2 0.25773  
## 2013-04-02 2 0.33619  
## 2015-12-28 28 0.27468  
## 2008-06-02 2 0.06828

From the above information, Monday is the day of the week with the highest and lowest daily change, as well as the highest volume of trade. Tuesday is the day with the lowest volume of trade. The dates to pull an internet search of news articles about Citigroup to analyze public sentiment on Citi stock are:

* April 2, 2007
* April 2, 2013
* December 28, 2015
* June 2, 2008

This should be interesting to see what type of articles are available on line with a google search of those dates and citigroup.

Lets see if there are any other outlier dates to examine by looking at the standard deviation of the daily change on Citigroup stock. We want to see if there are any days where the stock has a daily change more than or less than this amount times three then times two. Because most values will be within the standard deviation for the Gaussian curve.

gg <- ggplot(C\_stock, aes(x=C\_dailyChange))  
gg <- gg + geom\_histogram(binwidth=2, colour="black",   
 aes(y=..density.., fill=..count..))  
gg <- gg + stat\_function(fun=dnorm,  
 color="red",  
 args=list(mean=mean(C\_stock$C\_dailyChange),   
 sd=sd(C\_stock$C\_dailyChange)))  
  
gg



sdC <- sd(C\_stock$C\_dailyChange)  
out <- sdC\*3  
sdC;out

## [1] 32.16953

## [1] 96.50858

The standard error for the daily change in dollars is 32.17 USD and our threshold to find dates outside this normal range of daily change dollar values is 96.51 USD.

Lets add another column to this data set called threshold3 for those daily change values inside the threshold and those outside the threshold.

C\_stock$Threshold3 <- ifelse(C\_stock$C\_dailyChange < out, 'inside','outside')  
  
C\_outer\_SD <- subset(C\_stock, C\_stock$Threshold3=='outside')  
summary(C\_outer\_SD)

## C.Close C.Volume C\_ROI\_dollars C\_dailyChange   
## Min. :330.6 Min. : 2282100 Min. :-179.90 Min. :266.2   
## 1st Qu.:471.2 1st Qu.:13456250 1st Qu.: -39.30 1st Qu.:399.6   
## Median :510.6 Median :19551450 Median : 0.15 Median :441.4   
## Mean :488.2 Mean :30425167 Mean : -22.32 Mean :424.4   
## 3rd Qu.:542.8 3rd Qu.:35952375 3rd Qu.: 32.27 3rd Qu.:475.4   
## Max. :552.5 Max. :81343800 Max. : 42.00 Max. :510.5   
##   
## MonthYear portfolio\_DailyValue portfolio\_prevDay portfolio\_dailyValueChange  
## Apr-2007:1 Min. :2724 Min. :2744 Min. :-85.034   
## Aug-2007:1 1st Qu.:2899 1st Qu.:2878 1st Qu.: -4.048   
## Dec-2007:1 Median :2974 Median :2942 Median : -1.393   
## Feb-2007:1 Mean :3104 Mean :3044 Mean : 59.150   
## Jan-2007:1 3rd Qu.:3343 3rd Qu.:3076 3rd Qu.: 20.755   
## Jul-2007:1 Max. :3656 Max. :3619 Max. :734.207   
## (Other) :6   
## portfolio\_ROI\_dollars Date DayOfWeek   
## Min. :-253.961 Min. :2007-01-03 Length:12   
## 1st Qu.: -79.356 1st Qu.:2007-03-25 Class :character   
## Median : -4.371 Median :2007-06-16 Mode :character   
## Mean : 125.597 Mean :2007-06-17   
## 3rd Qu.: 364.923 3rd Qu.:2007-09-10   
## Max. : 677.926 Max. :2007-12-03   
##   
## Month Year UE\_monthlyRate portfolio\_DailyVolume  
## Length:12 Min. :2007 Min. :4.400 Min. :2.160e+08   
## Class :character 1st Qu.:2007 1st Qu.:4.500 1st Qu.:3.962e+08   
## Mode :character Median :2007 Median :4.600 Median :4.644e+08   
## Mean :2007 Mean :4.617 Mean :5.398e+08   
## 3rd Qu.:2007 3rd Qu.:4.700 3rd Qu.:6.314e+08   
## Max. :2007 Max. :5.000 Max. :1.005e+09   
##   
## portfolio\_prevDayVolume portfolio\_dailyVolumeChange  
## Min. :198190500 Min. :-197842207   
## 1st Qu.:387785669 1st Qu.: -23781530   
## Median :564614969 Median : 26069930   
## Mean :528884214 Mean : 10878309   
## 3rd Qu.:594041737 3rd Qu.: 70618878   
## Max. :971072459 Max. : 124348468   
##   
## portfolio\_VolumeRatioDaily2Initial portfolio\_ValueRatioDaily2Initial  
## Min. :0.3776 Min. :0.9388   
## 1st Qu.:0.6926 1st Qu.:0.9989   
## Median :0.8118 Median :1.0248   
## Mean :0.9436 Mean :1.0696   
## 3rd Qu.:1.1038 3rd Qu.:1.1521   
## Max. :1.7576 Max. :1.2599   
##   
## portfolio\_DailyRatios\_X\_UE dayOfMonth portfolio\_poisson Threshold3   
## Min. :1.654 Min. :1.00 Min. :0.04659 Length:12   
## 1st Qu.:3.696 1st Qu.:1.00 1st Qu.:0.05008 Class :character   
## Median :4.400 Median :1.00 Median :0.05454 Mode :character   
## Mean :4.641 Mean :1.75 Mean :0.13836   
## 3rd Qu.:5.116 3rd Qu.:2.25 3rd Qu.:0.25948   
## Max. :8.297 Max. :4.00 Max. :0.26474   
##

We can see from the above statistics on the subset of Citigroup stock that are outside this threshold that there are 12 dates to select in the range of Jan 2007 through Sep 2008. So we will add those dates to our data set of text scraped news articles on Citigroup.

NLP\_dates\_Citi <- rbind(C\_stock\_minmaxValueChanges, C\_outer\_SD[,-23])  
NLP\_dates\_Citi

## C.Close C.Volume C\_ROI\_dollars C\_dailyChange MonthYear  
## 2007-04-02 510.50 2282100 0.000000 510.500000 Apr-2007  
## 2013-04-02 44.11 1005100 -466.389999 0.320000 Apr-2013  
## 2015-12-28 52.38 377263800 -458.119999 -0.329998 Dec-2015  
## 2008-06-02 214.60 15302800 -295.899994 -298.300018 Jun-2008  
## 2007-04-021 510.50 2282100 0.000000 510.500000 Apr-2007  
## 2007-08-01 468.50 13495700 -42.000000 397.800003 Aug-2007  
## 2007-12-03 330.60 81343800 -179.899994 266.250008 Dec-2007  
## 2007-02-01 547.30 80864600 36.799988 467.409989 Feb-2007  
## 2007-01-03 552.50 43508100 42.000000 488.520000 Jan-2007  
## 2007-07-02 516.40 32822200 5.900024 441.990020 Jul-2007  
## 2007-06-01 545.10 23057000 34.599976 473.939972 Jun-2007  
## 2007-03-01 510.80 8981300 0.299988 440.769989 Mar-2007  
## 2007-05-01 542.00 13337900 31.500000 479.779999 May-2007  
## 2007-11-01 385.10 33433800 -125.399994 322.950004 Nov-2007  
## 2007-10-01 477.20 16045900 -33.299988 402.080009 Oct-2007  
## 2007-09-04 472.10 15929600 -38.399994 400.240005 Sep-2007  
## portfolio\_DailyValue portfolio\_prevDay portfolio\_dailyValueChange  
## 2007-04-02 2901.650 2891.963 9.686608  
## 2013-04-02 3433.938 3354.901 79.037872  
## 2015-12-28 5005.455 4984.970 20.485009  
## 2008-06-02 3120.541 3144.698 -24.157199  
## 2007-04-021 2901.650 2891.963 9.686608  
## 2007-08-01 2778.299 2781.133 -2.834138  
## 2007-12-03 2723.978 2743.972 -19.993872  
## 2007-02-01 3279.015 3281.965 -2.949476  
## 2007-01-03 2977.939 2977.939 0.000000  
## 2007-07-02 2969.196 2946.619 22.576765  
## 2007-06-01 3003.989 3006.774 -2.785581  
## 2007-03-01 2889.381 2896.725 -7.344424  
## 2007-05-01 2957.539 2937.392 20.147648  
## 2007-11-01 3534.398 3619.433 -85.034241  
## 2007-10-01 3655.864 3611.738 44.126353  
## 2007-09-04 3571.178 2836.972 734.206543  
## portfolio\_ROI\_dollars Date DayOfWeek Month Year  
## 2007-04-02 -76.289072 2007-04-02 Monday Apr 2007  
## 2013-04-02 455.999776 2013-04-02 Tuesday Apr 2013  
## 2015-12-28 2027.516411 2015-12-28 Monday Dec 2015  
## 2008-06-02 142.602196 2008-06-02 Monday Jun 2008  
## 2007-04-021 -76.289072 2007-04-02 Monday Apr 2007  
## 2007-08-01 -199.639490 2007-08-01 Wednesday Aug 2007  
## 2007-12-03 -253.960930 2007-12-03 Monday Dec 2007  
## 2007-02-01 301.076786 2007-02-01 Thursday Feb 2007  
## 2007-01-03 0.000000 2007-01-03 Wednesday Jan 2007  
## 2007-07-02 -8.742779 2007-07-02 Monday Jul 2007  
## 2007-06-01 26.049900 2007-06-01 Friday Jun 2007  
## 2007-03-01 -88.557542 2007-03-01 Thursday Mar 2007  
## 2007-05-01 -20.399119 2007-05-01 Tuesday May 2007  
## 2007-11-01 556.459753 2007-11-01 Thursday Nov 2007  
## 2007-10-01 677.925528 2007-10-01 Monday Oct 2007  
## 2007-09-04 593.239860 2007-09-04 Tuesday Sep 2007  
## UE\_monthlyRate portfolio\_DailyVolume portfolio\_prevDayVolume  
## 2007-04-02 4.5 572035712 572035712  
## 2013-04-02 7.6 258084601 330998801  
## 2015-12-28 5.0 975152259 752607802  
## 2008-06-02 5.6 464823559 265152951  
## 2007-04-021 4.5 572035712 572035712  
## 2007-08-01 4.6 686001371 572681959  
## 2007-12-03 5.0 1005429691 971072459  
## 2007-02-01 4.5 933350159 809001691  
## 2007-01-03 4.6 613250413 565411759  
## 2007-07-02 4.7 460278863 658121070  
## 2007-06-01 4.6 381151267 397701502  
## 2007-03-01 4.4 215973129 198190500  
## 2007-05-01 4.4 314742689 233827359  
## 2007-11-01 4.7 468477291 563818179  
## 2007-10-01 4.7 401234791 446710205  
## 2007-09-04 4.7 425224899 358038171  
## portfolio\_dailyVolumeChange portfolio\_VolumeRatioDaily2Initial  
## 2007-04-02 0 1.0000000  
## 2013-04-02 -72914200 0.4511687  
## 2015-12-28 222544457 1.7047052  
## 2008-06-02 199670608 0.8125779  
## 2007-04-021 0 1.0000000  
## 2007-08-01 113319412 1.1992282  
## 2007-12-03 34357232 1.7576345  
## 2007-02-01 124348468 1.6316292  
## 2007-01-03 47838654 1.0720492  
## 2007-07-02 -197842207 0.8046331  
## 2007-06-01 -16550235 0.6663068  
## 2007-03-01 17782629 0.3775518  
## 2007-05-01 80915330 0.5502151  
## 2007-11-01 -95340888 0.8189651  
## 2007-10-01 -45475414 0.7014156  
## 2007-09-04 67186728 0.7433538  
## portfolio\_ValueRatioDaily2Initial portfolio\_DailyRatios\_X\_UE  
## 2007-04-02 1.0000000 4.500000  
## 2013-04-02 1.1834435 4.057888  
## 2015-12-28 1.7250378 14.703404  
## 2008-06-02 1.0754368 4.893707  
## 2007-04-021 1.0000000 4.500000  
## 2007-08-01 0.9574896 5.281943  
## 2007-12-03 0.9387687 8.250061  
## 2007-02-01 1.1300522 8.297218  
## 2007-01-03 1.0262916 5.061081  
## 2007-07-02 1.0232786 3.869810  
## 2007-06-01 1.0352692 3.173112  
## 2007-03-01 0.9957719 1.654204  
## 2007-05-01 1.0192614 2.467577  
## 2007-11-01 1.2180652 4.688499  
## 2007-10-01 1.2599262 4.153540  
## 2007-09-04 1.2307408 4.299916  
## dayOfMonth portfolio\_poisson  
## 2007-04-02 2 0.25773  
## 2013-04-02 2 0.33619  
## 2015-12-28 28 0.27468  
## 2008-06-02 2 0.06828  
## 2007-04-021 2 0.25773  
## 2007-08-01 1 0.05008  
## 2007-12-03 3 0.05723  
## 2007-02-01 1 0.04833  
## 2007-01-03 3 0.05008  
## 2007-07-02 2 0.26474  
## 2007-06-01 1 0.05008  
## 2007-03-01 1 0.04659  
## 2007-05-01 1 0.25411  
## 2007-11-01 1 0.05184  
## 2007-10-01 1 0.26474  
## 2007-09-04 4 0.26474

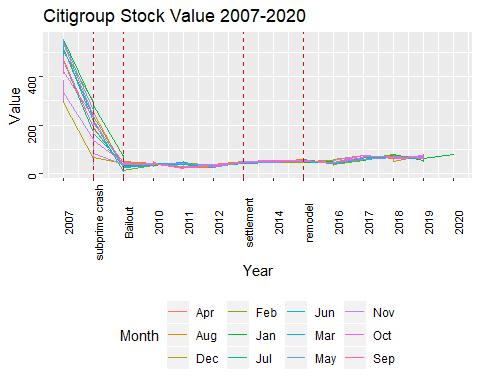
I am going to pull the data from these dates with the Google Search for the specific date on Citigroup stock, put it in a table with the date, the article title, reference, article content, and the comments if available.

Note: when searching the internet, there were limited articles and [most](https://www.nytimes.com/2008/11/23/business/23citi.html) were about Citi’s involvement in the sub-prime mortgage crisis of 2007-2008, and a [bailout](https://www.reuters.com/article/us-citigroup/citigroup-gets-massive-government-bailout-idUSTRE4AJ45G20081124) of Citigroup by the US. For the month and years of the two dates not in or around 2007-2008, there are only two for April 2013 and December 2015. Where Citi settled a [lawsuit](https://www.reuters.com/article/us-citigroup-settlement/citigroup-settles-shareholder-cdo-lawsuit-for-590-million-idUSBRE87S0UA20120829) for covering up bad mortgage loans in August 2012 and a [person reported](https://ficoforums.myfico.com/t5/Credit-Card-Approvals/Citi-Simplicity-Approved-Woohoooooo/td-p/4388074) on a forum about FICO scores how he was approved for a 4600 USD credit card with Citi. There isn’t enough data to rely on the web for NLP on Citigroup for these time frames.

Lets plot this as a simple line chart of the value of the stock over the years.

ggplot(data = C\_stock, aes(x=Year, y=C.Close, group=Month)) +  
 geom\_line(aes(color=Month))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 scale\_x\_continuous(breaks=c(2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020),  
 labels=c(2007,'subprime crash','Bailout',2010,2011,2012,'settlement',2014,'remodel',2016,2017,2018,2019,2020))+  
 theme(axis.text = element\_text(colour = "black", angle=90, size = rel(.75)))+  
 geom\_vline(xintercept=c(2008,2009,2013,2015), linetype='dashed', color='red')+  
 ggtitle('Citigroup Stock Value 2007-2020')+  
 ylab('Value')

## Warning in pal\_name(palette, type): Unknown palette paired



We could pull based on the keywords: ‘settlement’, ‘bail-out’, ‘sub-prime loans’, but we would only get the obvious negative sentiment for these keywords. A New York Times article posted an article in Dec 2015 about the remodeling that Citigroup was doing to their offices, but the full article would have to be purchased. The fact that they spent money on remodeling could have some public sentiment of either they aren’t distributing their profits to shareholders or they are making enough profits to spend money on remodeling, which is also reported at the end of the year in 2015 to write off for that tax year. Although, I was told by an accountant that some corporations and small businesses have a different tax year and a quick search on Google returned the fiscal year is any consecutive 12-month business cycle that usually ends at the end of each quarter.

We can see that the volume of trades is highest in December 2015 from our dates, but we should compare this to which quantile this number is within for the volume of trades of Citi stock.

summary(C\_stock$C.Volume)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1005100 13019600 19493900 26987469 33280800 377263800

We already know that this is the date that the most trades in stock of Citi occured as it is the reason we added this date to our NLP data set of dates to pull information from the web for. The above will refresh the comparisons of the trade volume to this date.

It looks like public sentiment thinks Citi is going back to its old bail-out days of 2007-2008 and not a trust-worthy stock for their personal portfolios. But they are still around, and the fact that people that have a less than trust-worthy credit profile were given a credit card with a high value could indicate some people also consider that they are building a new demographic of people to invest in by earning the trust of those who have sub-par trust worthiness with credit. And, yet some other investors could also think this is a bad move to make as it depends on those same people realizing their mistakes and not making them again. Which really turns into the reason some stocks are volatile to begin with and possibly a reason to understand Game Theory, a class I dropped in my undergrad college. But nonetheless I am a data scientist with other coventional and non-conventional ways of extracting useful information, and this approach uses my math and analytic skills to fully understand the stock market and certain stocks and trends with public sentiment.

On this highest trade day, the daily change in dollars was still within the standard error by only dropping 0.33 USD. Where the standard error is 32.00 USD.

Of note is whether or not those making these trades are doing so to lower their Capital Gains at the end of the year, because there is a slight loss on it to balance out the portfolio. Also, this is the end of the year, possibly the last trading day of the year as it is. Lets look at all monthYear dates equal to Dec-2015 to see if there are any other dates past Dec 28, 2015.

dec2015 <- subset(C\_stock, C\_stock$MonthYear=='Dec-2015')  
tail(dec2015)

## C.Close C.Volume C\_ROI\_dollars C\_dailyChange MonthYear  
## 2015-12-23 52.63 93423000 -457.87 0.620003 Dec-2015  
## 2015-12-24 52.71 119108100 -457.79 0.079998 Dec-2015  
## 2015-12-28 52.38 377263800 -458.12 -0.329998 Dec-2015  
## 2015-12-29 52.98 281369700 -457.52 0.599999 Dec-2015  
## 2015-12-30 52.30 62625000 -458.20 -0.680001 Dec-2015  
## 2015-12-31 51.75 49092600 -458.75 -0.549999 Dec-2015  
## portfolio\_DailyValue portfolio\_prevDay portfolio\_dailyValueChange  
## 2015-12-23 4998.690 4968.045 30.64500  
## 2015-12-24 4984.970 4998.690 -13.72002  
## 2015-12-28 5005.455 4984.970 20.48501  
## 2015-12-29 4738.190 5005.455 -267.26507  
## 2015-12-30 4800.285 4738.190 62.09506  
## 2015-12-31 4707.685 4800.285 -92.59999  
## portfolio\_ROI\_dollars Date DayOfWeek Month Year UE\_monthlyRate  
## 2015-12-23 2020.751 2015-12-23 Wednesday Dec 2015 5  
## 2015-12-24 2007.031 2015-12-24 Thursday Dec 2015 5  
## 2015-12-28 2027.516 2015-12-28 Monday Dec 2015 5  
## 2015-12-29 1760.251 2015-12-29 Tuesday Dec 2015 5  
## 2015-12-30 1822.346 2015-12-30 Wednesday Dec 2015 5  
## 2015-12-31 1729.746 2015-12-31 Thursday Dec 2015 5  
## portfolio\_DailyVolume portfolio\_prevDayVolume  
## 2015-12-23 903674159 619024059  
## 2015-12-24 752607802 903674159  
## 2015-12-28 975152259 752607802  
## 2015-12-29 1248436459 975152259  
## 2015-12-30 534260059 1248436459  
## 2015-12-31 504630159 534260059  
## portfolio\_dailyVolumeChange portfolio\_VolumeRatioDaily2Initial  
## 2015-12-23 284650100 1.5797513  
## 2015-12-24 -151066357 1.3156658  
## 2015-12-28 222544457 1.7047052  
## 2015-12-29 273284200 2.1824450  
## 2015-12-30 -714176400 0.9339628  
## 2015-12-31 -29629900 0.8821655  
## portfolio\_ValueRatioDaily2Initial portfolio\_DailyRatios\_X\_UE  
## 2015-12-23 1.722706 13.607238  
## 2015-12-24 1.717978 11.301424  
## 2015-12-28 1.725038 14.703404  
## 2015-12-29 1.632930 17.818897  
## 2015-12-30 1.654330 7.725412  
## 2015-12-31 1.622417 7.156201  
## dayOfMonth portfolio\_poisson Threshold3  
## 2015-12-23 23 0.27468 inside  
## 2015-12-24 24 0.05723 inside  
## 2015-12-28 28 0.27468 inside  
## 2015-12-29 29 0.05723 inside  
## 2015-12-30 30 0.27468 inside  
## 2015-12-31 31 0.05723 inside

We now know that Dec-28-2015 is not the last trading day of the year, because the 29th through 31st for Tuesday through Thursday are also trading days. There was a fluctuation in dollars earned and lost all under a dollar. Some useful information to add in would be who or where are these trades derived. Are they financial advisors, trust fund managers, independent investors, foreign or national investors, are they hobbyists just playing the stock market on an e-trade, are they educated, experienced, and so on?

To get this information we could first find out how much it costs for a hobbyist to make a trade online from e-trade or similar and whether or not this information is shared on demographics of the stock ownership. We could also look at the American Survey on Census data from the census bureau for numer of financial workers there are and how many people graduated with a BS, MS, or Phd in Finance or Economics. If there is location data on where these stock owners live attach this information gathered to it to make a better inference on this stock and what motivates the trades. Any volunteers?

For now, we will just continue with what we have on hand for Citi. We can answer the question of whether or not, historically there are more trades in December than any other month in our data by grouping by month year and getting the median trades per month and year.

Citi\_trades\_monthYear <- C\_stock %>% group\_by(MonthYear) %>%  
 summarise\_at(vars(colnames(C\_stock[2])), mean)  
Citi\_trades\_monthYear <- Citi\_trades\_monthYear[order(Citi\_trades\_monthYear$C.Volume,decreasing=TRUE),]  
Citi\_trades\_monthYear

## # A tibble: 157 x 2  
## MonthYear C.Volume  
## <fct> <dbl>  
## 1 Dec-2011 102284343.  
## 2 Dec-2012 97253820   
## 3 Feb-2007 94010711.  
## 4 Feb-2008 80151765   
## 5 Dec-2019 79458262.  
## 6 Aug-2019 72849682.  
## 7 Feb-2015 70393405.  
## 8 Dec-2015 67380332.  
## 9 Jan-2010 64943774.  
## 10 Jan-2012 63211745   
## # … with 147 more rows

From the above table ordered from most trades to least trades per month and year by mean number of trades per month, we see that December is in the top 10 month years of high trades in 2011,2012, 2015, and 2019. February has the next highest trades but the years are the same years of the sub-prime mortgage crisis that Citigroup was involved in, but also in 2015. looking at the next top ten months we see that Dec, Jan, and Feb are in the highest mean of the trades per day grouped by month and year. What do we know about Jan and Feb outside of the assumption about December being the last day of the tax year to offset capital gains with capital losses?

Well, I know that being a student, some people get their student loans around winter quarter in January and that many people expecting tax refunds get their refunds in February. We would have to see if there are any other assumptions about these months. But we would be able to ascertain if students receiving an education are investing, and if consumers with tax refunds are using some of that money to invest.There are certainly other assumptions that could be made for why the last month of the year and the first two months of the first quarter are high trade volume days. But for now lets stick with these assumptions.

July starts to show up in the following set of ten top month years from 21-30, as the 30th highest trade month year. Jan and Feb are still in the top 40 high volume trade month years, while June shows up three times in the 30-40 top high volume trade month and years. July could also be the start of the third quarter and the remaining balance on student loans made. Lets see where September/October show up in these top ordered volumes. They are near the end of the top trade months.

So, possibly this indicates no ties to student loan payments, but tax refunds could be likely for February being a high trade month. We definitely know December is a top trade day.

Lets plot this data.

Citi\_trades\_monthYear$Month <- gsub('-[0-9]{4}','',Citi\_trades\_monthYear$MonthYear)  
Citi\_trades\_monthYear$Year <- gsub('[a-zA-z]{3}-','',Citi\_trades\_monthYear$MonthYear)  
  
ggplot(data = Citi\_trades\_monthYear, aes(x=Month, y=C.Volume,group=Year)) +  
 geom\_line(aes(color=Year))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Mean Month-Year Trade Volume 2007-2020')+  
 ylab('Trade Volume')

## Warning in pal\_name(palette, type): Unknown palette paired



We can see that December is definitely the highest trading month, then February as the next highest, and January as the third highest trading month.

Lets look at the daily change mean values per month, by grouping by MonthYear and taking the mean value of the daily change, order by highest to smallest, and plot.

Citi\_meanMonthly\_dailyChange <- C\_stock %>% group\_by(MonthYear) %>%   
 summarise\_at(vars(as.vector(colnames(C\_stock))[4]), mean)

Citi\_meanMonthly\_dailyChange$Year <-  
 gsub('[a-zA-Z]{3}-','',Citi\_meanMonthly\_dailyChange$MonthYear)  
Citi\_meanMonthly\_dailyChange$Month <-  
 gsub('-[0-9]{4}','',Citi\_meanMonthly\_dailyChange$MonthYear)  
  
  
ggplot(data = Citi\_meanMonthly\_dailyChange, aes(x=Month, y=C\_dailyChange,group=Year)) +  
 geom\_line(aes(color=Year))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Mean Month-Year Daily Change 2007-2020')+  
 ylab('Mean Daily Change Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



From the above line chart, it is not obvious what years those years having almost no change are.The year 2007 is at the top with the highest positive mean daily change values fluctuating to around 20 USD per day. While the years 2008 and 2009 have the highest negative mean of daily change values per month with average daily decreases around a daily loss of 5-15 USD.

Lets make a bar chart of 2007, 2008, 2009, 2015, and 2019 of this data on mean daily value changes per month.

y4 <- subset(Citi\_meanMonthly\_dailyChange,  
 Citi\_meanMonthly\_dailyChange$Year==2008 |   
 Citi\_meanMonthly\_dailyChange$Year==2009 |   
 Citi\_meanMonthly\_dailyChange$Year==2007 |  
 Citi\_meanMonthly\_dailyChange$Year==2015 |  
 Citi\_meanMonthly\_dailyChange$Year==2019)  
ggplot(data = y4, aes(x=Month, y=C\_dailyChange,fill=Year)) +  
 geom\_bar(stat='identity', position=position\_dodge())+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette='Paired') +  
 geom\_hline(yintercept=0, linetype="dashed", color = "red")+  
 theme\_classic()+  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Mean Monthly Daily Dollar Change 2007-2019')+  
 ylab('Mean Daily Change Values')



From the above, we can see the Citigroup stock had increases per day in value from the previous day in 2007, but that in 2008 and 2009 those daily increases turned to daily decreases from day to day as the sub-prime loans collapsed that Citigroup held. And in 2015 and 2019 years after Citigroup’s bailout there was a mean monthly daily change value next to nothing as the daily change from day to day fluctuated around zero dollars for the month.

This could mean it is gaining strength and remains as is safe to buy as it increases. But lets look at the years 2015-2019 to see how the value of the Citigroup stock has faired by month year to confirm this assertion just made.

y4value <- subset(C\_stock, C\_stock$Year>2014)  
y4valMY <- y4value %>% group\_by(MonthYear) %>%  
 summarise\_at(vars(as.vector(colnames(y4value)[1])), mean)

y4valMY$Year <- gsub('[a-zA-Z]{3}-','', y4valMY$MonthYear)  
y4valMY$Month <- gsub('-[0-9]{4}','', y4valMY$MonthYear)  
  
ggplot(data = y4valMY, aes(x=Month, y=C.Close,fill=Year)) +  
 geom\_bar(stat='identity', position=position\_dodge())+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette='Paired') +  
 geom\_hline(yintercept=min(y4valMY$C.Close), linetype="dashed", color = "red")+  
 geom\_hline(yintercept=mean(y4valMY$C.Close), linetype="dashed", color = "black")+  
 theme\_classic()+  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Mean Monthly Dollar Value 2015-2020')+  
 ylab('Mean Monthly Dollar Value')



From the above bar chart, we can see that the minimum value is the dashed red line which occured in February 2016. And that every month since 2016 has been above this minimum value. It has almost double from it’s minimum value in January and February 2020.The mean value from 2015-2020 (Jan-Feb) is just above 60 USD which is 1 1/2 times its minimum value.

Lets look at the line chart of this by years 2015-2020.

ggplot(data = y4valMY, aes(x=Year, y=C.Close,group=Month)) +  
 geom\_line(aes(color=Month))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Citigroup Mean Monthly Value 2015-2020')+  
 ylab('Mean Daily Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



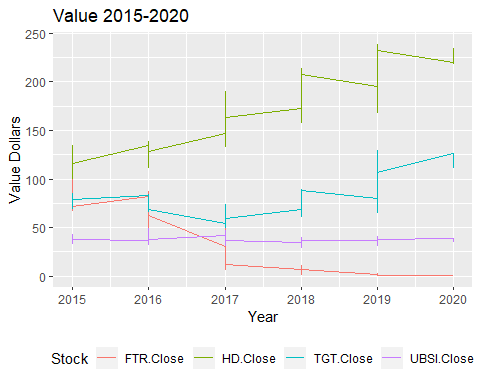
The above line chart of the mean monthly dollar value of the Citigroup stock show that all months move the same direction of decreasing in 2015, increasing in 2016, except for in 2017 and 2018 where 3-6 months decreased and 6-9 months increased monthly mean values. The span of 2019 through 2020 can’t be analyzed yet, but January increased since the year prior. Overall, since 2015 the value has increased from 50-60 USD to between 75-80 USD. This could make it a good stock to have in your portfolio as it has steadily been increasing since it’s historical rough patches of the sub-prime mortgage loan accounts, the public bailout, and the lawsuit settlement payout. But nothing has been in the news about them to discourage investors from dropping this stock from their stock folder.

We saw that Citigroup is maintaining its current value and slightly increasing over the last four years. Lets start subset sampling stocks and look at the changes they have made in value over the last four years. And see if we notice anything we want to further exploit.

Value1 <- StocksSTATS[,c(1:53,160:230)]  
Value2 <- subset(Value1, Year>2014)

sub1 <- Value2[,c(1:4,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



The first four stocks in our set of 53 is shown in the line chart above from 2015-2020.

From the above line chart, it is obvious that over the last five years, the pink line for FTR is a terrible stock as it has been on the decline, but we would have to look at it further to see why it has been decreasing in value since 2015.

The olive color line for HD indicates it has been on a steady increase from the 120-125 USD range in 2015 to the 220-225 USD range in 2020.

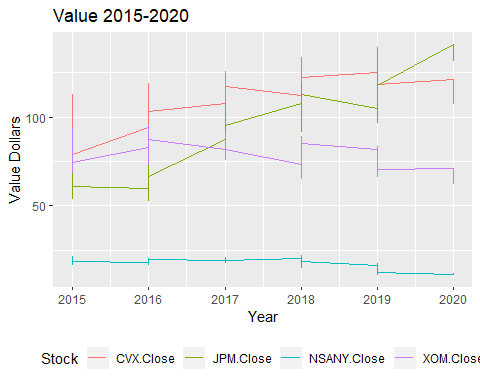
Also, increasing steadily is the blue line for TGT, which started at 75-80 in 2015 and is at 125 in 2020 in value.

The purple line for UBSI has been maintaining steadily from 45 range to 45 range over five years. \*\*\*

Lets look at the next four stocks.

sub1 <- Value2[,c(5:8,115)]  
  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

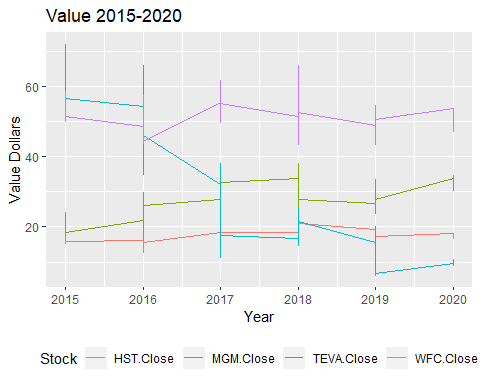


From the above subset of the next four stock in our 53 stocks, we can see that there are two stocks increasing significantly for JPM and CVX. We also note that the XOM and NSANY stocks have decreased over the last five years. \*\*\*

Now for the next four stocks.

sub1 <- Value2[,c(9:12,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



The above line chart shows the third subset of four stocks of our 53 stocks.

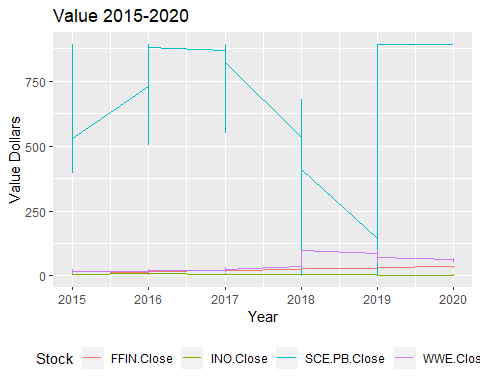
The MGM stock has increased significantly since 2005, and slight increases are shown for WFC and HST though not significantly. There is some cyclical movements in the WFC with 2016 giving a steady increase all year, then declining 2017-2019, and ending with a steady increase in 2019.

The TEVA stock has had a huge loss over the last five years, with the last year showing an an increase slightly. It started at the 55 range in 2015 and is at the 10 range in 2020. This could indicate that it is a good time to buy TEVA, since it is priced low and shows an increase in the last year, where the last four years it has been decreasing annually for each year. This would require further analysis for why it has been decreasing over the last five years. \*\*\*

Now for the next four stocks in our subset four.

sub1 <- Value2[,c(13:16,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



The above line chart shows that SCE.PB is on its own scale that outweighs the scale of the other smaller valued stocks, there is also volatility and cyclical movements in SCE.PB which makes it a good choice to further analyze with timelines of web article events that could have triggered these changes in value of a steady increase in 2015, a high jump increase in 2016, then a steep decline throughout 2017 and 2018, then a huge jump of an increase to the same level at 2016. This is a utility company so government contracts could be involved with all that entails, and possible fires causing damage and settlements in the declining years. But for now it is just speculation and assumptions.

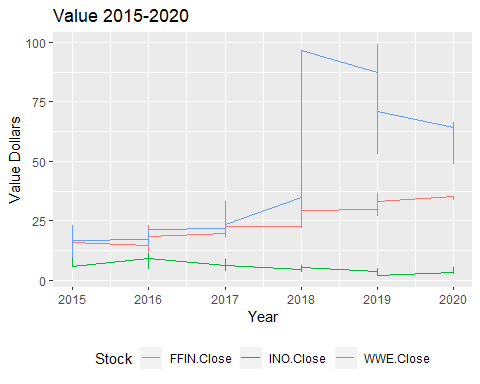
–update– This stock SCE doesn’t belong here. It didn’t get categorized as numeric to begin with, and used factors instead, creating that chart above. I am working on it but many thousands of code have to be altered as they were based on column indexing when rearranging the features. So there are actually only 52 and not 53 stocks in our time series. Sce stock also doesn’t have the same amount of observations as the other stocks do when turned to numeric.–

The other stocks are getting limited spotlight above, and they need their own scale as SCE.PB pushed down their scaled visual line charts.

Now for the next four stocks in our subset four.

sub1 <- Value2[,c(13,14,16,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:3)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



From the above line chart, we see that WWE had a huge jump in 2018 of an increase from the 40 range to the 90 range but then decreased during 2018 and 2019 to a price still much higher at the 60 range than its starting value in 2015 of the 20 range.

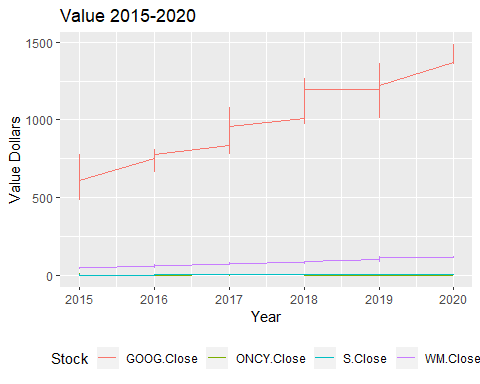
The FFIN stock has been steadily increasing over the last five years with a flat line on the value in 2017 and 2018.

The INO stock has declined since 2016 after an increasing year in 2015, but lost only slightly in value over a five year span returning no profits over that time span.

Now for the next stocks in our subset.

sub1 <- Value2[,c(17:20,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

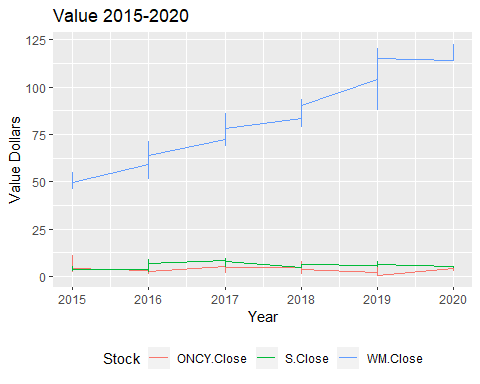


In the above subset of stocks, Google out scales the other three stocks and shows that it has been increasing steadily every year, except 2018 where it is almost the same price all year.

Lets look at the other three stocks that our on a lower scaled value to analyze them.

sub1 <- Value2[,c(18:20,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:3)  
  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 #geom\_hline(yintercept=m15, color='red')  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



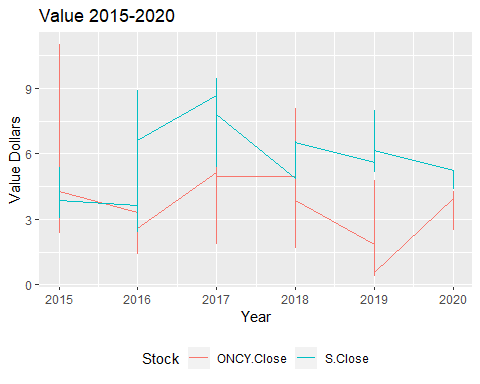
The line chart above shows that WM has increased significantly every year since 2015, with a slight decrease in 2019, but overall has increased from the 50 range in 2015 to the 113 range in 2020.

The ONCY and S stocks have had slight increases and decreases in the last five years but look like they have increased slightly overall from 2015-2020.

Lets look at S and ONCY stocks more closely.

sub1 <- Value2[,c(19:20,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:2)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

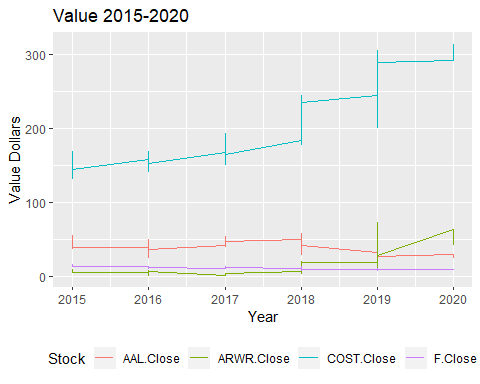


It looks like these two stocks, ONCY and S, have had cyclical patterns in the last five years, and if that is true, then S stock hasn’t reached its cyclical minimum and ONCY stock hasn’t reached it cyclical maximum. And if this is not the case then there are some triggers in the value of this stock in 2016, where they both increased, then steadily decreased in 2017. A global minimum in the last five years is seen in 2019 for ONCY stock, while the global maximums for both stock is in 2017. The start of 2016 showed both stocks had a local minima while S stock had its global minima this year, but only for this last five year period.

Now for the next stocks in our subset.

sub1 <- Value2[,c(21:24,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

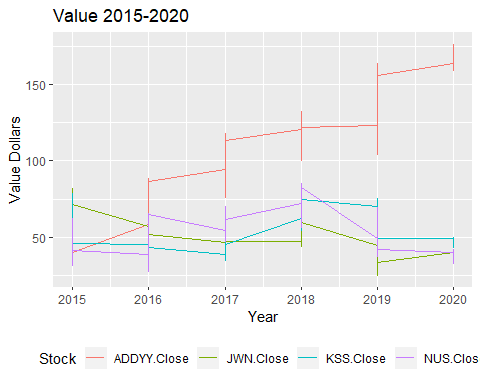


The above subset shows that ARWR and COST stock have been increasing the last two years, but ARWR stock had some near flat changes in value for years 2015, 2016, and 2017. The purple line for Ford is relatively maintaining value, but no increases or decreases of note for Ford in the last five years. The AAL stock had a global maxima in 2018 but overall decreased in value slightly in the last five years. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(25:28,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

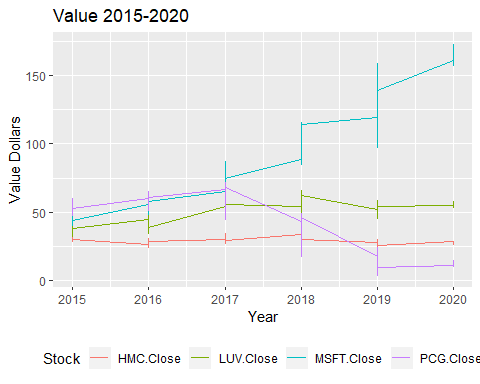


The above line chart shows that ADDYY has been significantly increasing over the last five years it jumped from the 40 USD range to the 165 USD range in 2020. The other three stocks all moved together with slightly different rates of increase and decrease. But the JWN stock lost value over the last five years, while KSS and NUS stocks both increased only marginally after some cyclical rise and falls in value. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(29:32,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

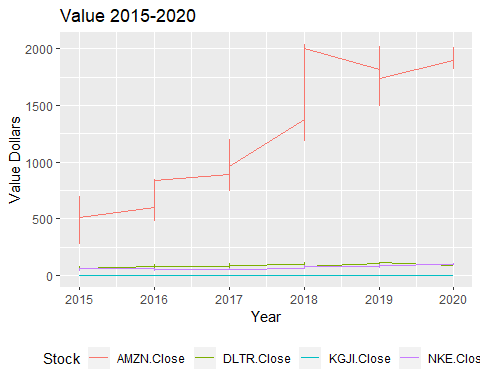


The above line chart shows that MSFT increased steadily the last five years with none of the years having declining values in stock. PCG stock had a local maxima in 2017 but a local minima in 2019 which led to an overall loss in value from 2015-2020. The LUV stock is the olive colored stock that had an increase overall in value by about 10 USD. And the HMC stock slightly stayed the same and may have decreased marginally in the last five years. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(33:36,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

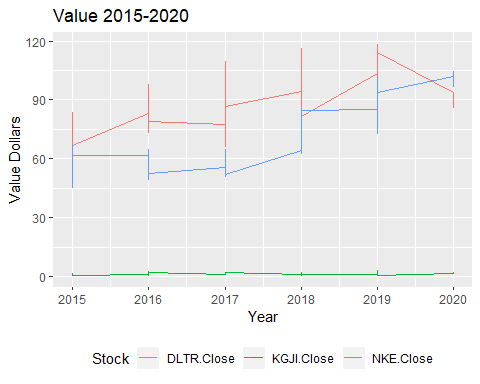


The above line chart shows that AMZN stock is on its own scale and has saw an overall huge jump in value in the last five years, with every year increasing, except in 2018 where it decreased from its local maxima at the start of 2018. Its value in 2015 was in the 500 USD range and at the start of 2020 was in the 1700-1800 USD range.

Lets look at the scale more appropriate for the other three stocks of DLTR, KGJI, and NKE.

sub1 <- Value2[,c(33:35,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:3)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

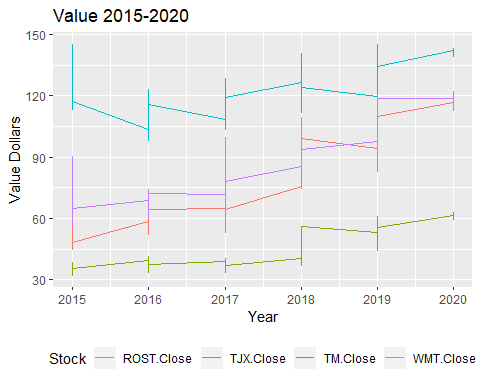


The above line chart shows the smaller scale value changes by year for DLTR, KGJI, and NKE. Both NKE and DLTR stocks have increased in value over the last five years, while DLTR did see a decreasing value throughout the last year of 2019. The KGJI stock showed marginal changes in value over the last five years, with no significant local minimas or local maximas.It does look like a slight increase overall from 2015-2020 for KGJI stock. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(37:40,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

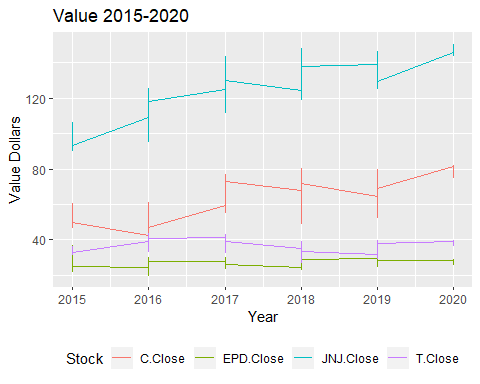


In the above line chart we see that all of the stocks increased noticeably in the last five years. The TM stock had some years that decreased in 2015, 2016, and 2018, but always starts the new year at a higher value than the year before. In 2018 WMT increased, while the other three stocks of TJX, TM, and ROST saw slight decreases. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(41:44,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

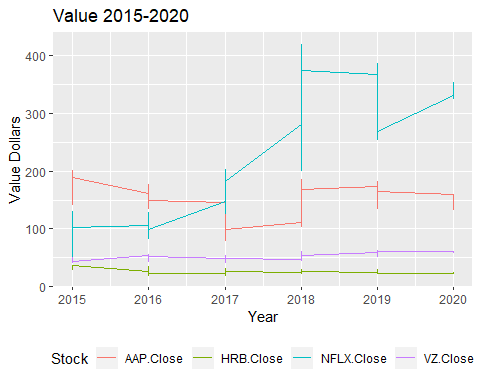


The above line chart also shows an overall increase in value over the last five years with significant jumps in value for C and JNJ stocks. In 2017, there were some decreases in value throughout the year for all these stocks of C, EPD, JNJ, and T stocks, but in two years they all started 2019 at the same values of 2017 and saw increasing values throughout 2019. \*\*\*

Now for the next stocks in our subset.

sub1 <- Value2[,c(45:48,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



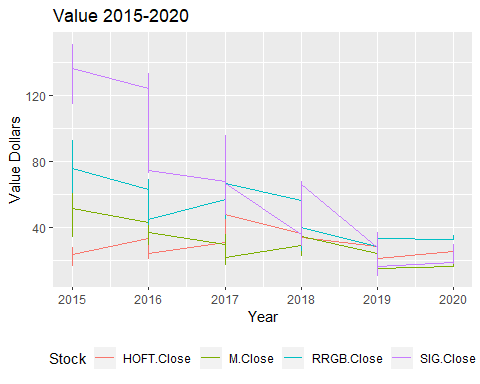
The above line chart shows that NFLX increased significantly while HRB and AAP saw losses over the last five years. VZ stock saw a slight increase in value over the last five years. In 2017 Netflix saw a huge inrease, while in 2018 it stayed somewhat stagnant with a sharp drop in value at the start of 2019 that saw an increasing year throughout 2019.

In 2017, there was a sharp drop in value for AAP, but by the start of 2018 the value increased to a value above the start of 2017.

Now for the last five stocks in our subset.

sub1 <- Value2[,c(49:53,115)]  
sub1tidy <- gather(sub1, 'Stock','Value',1:4)  
  
ggplot(data = sub1tidy, aes(x=Year, y=Value,group=Stock)) +  
 geom\_line(aes(color=Stock))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('Value 2015-2020')+  
 ylab('Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired



Our last set of stock show that RRGB and SIG saw significant losses over the last five years, while M stock showed a smaller loss. HOFT stock saw an increase over the last five years, but only marginally or slightly. In 2017 M stock saw an increasing year for its value after having two years from 2015-2016 see decreasing values throughout those years. M stock and HOFT stock seemed to be negatively correlated for years 2015-2018, with both stocks having different rates of decrease in 2018 and an increase in value of similar rates of increase in 2019. All of these stocks decreased at different rates in 2018, and increased at different rates in 2019. \*\*\*

Lets group by the year and get the mean values over the last five years for each stock value.

Value3 <- Value2[,c(1:53,112,115)]  
  
yearMeans <- Value3 %>% group\_by(Year) %>%  
 summarise\_at(vars(as.vector(colnames(Value3)[1:53])), mean)  
  
yearMeansTidy <- gather(yearMeans,'Stock','YearMeanValue',2:54)  
  
stock5yrMeans <- yearMeansTidy %>% group\_by(Stock) %>%  
 summarise\_at(vars(as.vector(colnames(yearMeansTidy)[3])), mean)  
colnames(stock5yrMeans)[2] <- 'stock5yrMeans'

Stock5year <- merge(stock5yrMeans,yearMeansTidy, by.x='Stock', by.y='Stock')

stock5yrOrdered <- Stock5year[with(Stock5year, order(Stock, Year)),]

Lets add a field that shows if the stock had an increase of 10% during the year and a field that shows if it decreased

ymn <- stock5yrOrdered$YearMeanValue  
YMN <- c(ymn[1],ymn[1:length(ymn)-1])  
  
stc2 <- stock5yrOrdered$Stock  
STC2 <- c('xyz',stc2[1:length(stc2)-1])  
  
STC3 <- ifelse(stc2==STC2, 1,0)  
  
stock5yrOrdered$Direction5yr10PercentChange <- ifelse(STC3==1 & stock5yrOrdered$YearMeanValue-YMN > .10\*YMN,'up10',  
 ifelse(STC3==1 & stock5yrOrdered$YearMeanValue-YMN <= -0.10\*YMN, 'down10',  
 ifelse(STC3==1 & stock5yrOrdered$YearMeanValue-YMN <= 0, 'down', ifelse(STC3==1 & stock5yrOrdered$YearMeanValue-YMN > 0, 'up', ''))))  
  
show1 <- cbind(head(stock5yrOrdered), tail(stock5yrOrdered))  
show1

## Stock stock5yrMeans Year YearMeanValue Direction5yr10PercentChange  
## 2 AAL.Close 38.67371 2015 45.12210   
## 3 AAL.Close 38.67371 2016 38.18385 down10  
## 4 AAL.Close 38.67371 2017 47.49072 up10  
## 1 AAL.Close 38.67371 2018 42.80195 down  
## 5 AAL.Close 38.67371 2019 30.87933 down10  
## 6 AAL.Close 38.67371 2020 27.56429 down10  
## Stock stock5yrMeans Year YearMeanValue Direction5yr10PercentChange  
## 2 XOM.Close 78.737 2015 82.82845   
## 3 XOM.Close 78.737 2016 86.21968 up  
## 4 XOM.Close 78.737 2017 81.86159 down  
## 1 XOM.Close 78.737 2018 79.95570 down  
## 5 XOM.Close 78.737 2019 73.73464 down  
## 6 XOM.Close 78.737 2020 67.82191 down

length(unique(stock5yrOrdered$Stock))

## [1] 53

Lets get these subsets of stocks that within the time span of 2015-2020 increased by more than 10% annually, decreased by 10% or more annually, decreased, or increased.

Stocks10PercentAnnualDecrease2015\_2020 <- subset(stock5yrOrdered, stock5yrOrdered$Direction5yr10PercentChange=='down10')  
  
stocks10Decr <- Stocks10PercentAnnualDecrease2015\_2020 %>% group\_by(Stock) %>% count(n=n())  
colnames(stocks10Decr)[2] <- 'nTimesDecr10\_5yr'  
stocks10Decr <- stocks10Decr[,-3]  
  
Stocks10PercentAnnualIncrease2015\_2020 <- subset(stock5yrOrdered, stock5yrOrdered$Direction5yr10PercentChange=='up10')  
  
stocks10Incr <- Stocks10PercentAnnualIncrease2015\_2020 %>% group\_by(Stock) %>% count(n=n())  
colnames(stocks10Incr)[2] <- 'nTimesIncr10\_5yr'  
stocks10Incr <- stocks10Incr[,-3]  
  
StocksAnnualIncrease2015\_2020 <- subset(stock5yrOrdered, stock5yrOrdered$Direction5yr10PercentChange=='up')  
  
StocksIncrZerobase <- StocksAnnualIncrease2015\_2020 %>% group\_by(Stock) %>% count(n=n())  
colnames(StocksIncrZerobase)[2] <- 'nTimesIncrFromZero\_5yrs'  
StocksIncrZerobase <- StocksIncrZerobase[,-3]  
   
StocksAnnualDecrease2015\_2020 <- subset(stock5yrOrdered, stock5yrOrdered$Direction5yr10PercentChange=='down')  
  
StocksDecrZerobase <- StocksAnnualDecrease2015\_2020 %>% group\_by(Stock) %>% count(n=n())  
colnames(StocksDecrZerobase)[2] <- 'nTimesDecrFromZero\_5yrs'  
StocksDecrZerobase <- StocksDecrZerobase[,-3]

Lets merge these sets together with outer joins.

Stocks5yrChanges\_outerJoin <- merge(stocks10Decr,stocks10Incr, by.x='Stock', by.y='Stock', all=TRUE)  
  
Stocks5yrChanges\_outerJoin1 <- merge(Stocks5yrChanges\_outerJoin,StocksDecrZerobase, by.x='Stock', by.y='Stock', all=TRUE)  
  
Stocks5yrChanges\_outerJoin2 <- merge(Stocks5yrChanges\_outerJoin1,StocksIncrZerobase, by.x='Stock', by.y='Stock', all=TRUE)  
  
stock\_5yr\_stats\_2015\_2020 <- merge(stock5yrOrdered,Stocks5yrChanges\_outerJoin2, by.x='Stock', by.y='Stock', all=TRUE)  
  
length(unique(stock\_5yr\_stats\_2015\_2020$Stock))

## [1] 53

Write this file out to analyze those stocks having decreased and increased the most in the last 5 years.

write.csv(stock\_5yr\_stats\_2015\_2020,'stocks\_STATS\_N\_Changes.csv', row.names=FALSE)

Lets attach the stock name to this data set above by reading in the file with the names on it when hand picking these stocks by searching manually in finance.yahoo.com.

stockNames <- read.csv('yahooStockBasket.csv', header=T, sep=',', na.strings=c('',' '))  
stock\_5yr\_stats\_2015\_2020$Stock <- gsub('[.]Close','', stock\_5yr\_stats\_2015\_2020$Stock)  
stockNames$stock <- gsub('-','.', stockNames$stock)  
  
stock\_5yr\_stats\_2015\_2020$Stock <- as.factor(stock\_5yr\_stats\_2015\_2020$Stock)  
StockNames\_STATS\_2015\_2020 <- merge(stockNames,stock\_5yr\_stats\_2015\_2020,  
 by.x='stock', by.y='Stock')  
  
StockNames\_STATS\_2015\_2020$nTimesDecr10\_5yr <-  
 ifelse(is.na(StockNames\_STATS\_2015\_2020$nTimesDecr10\_5yr==TRUE),  
 0,StockNames\_STATS\_2015\_2020$nTimesDecr10\_5yr)  
  
StockNames\_STATS\_2015\_2020$nTimesIncr10\_5yr <-  
 ifelse(is.na(StockNames\_STATS\_2015\_2020$nTimesIncr10\_5yr==TRUE),  
 0,StockNames\_STATS\_2015\_2020$nTimesIncr10\_5yr)  
  
StockNames\_STATS\_2015\_2020$nTimesDecrFromZero\_5yrs <-  
 ifelse(is.na(StockNames\_STATS\_2015\_2020$nTimesDecrFromZero\_5yrs==TRUE),  
 0,StockNames\_STATS\_2015\_2020$nTimesDecrFromZero\_5yrs)  
  
StockNames\_STATS\_2015\_2020$nTimesIncrFromZero\_5yrs <-  
 ifelse(is.na(StockNames\_STATS\_2015\_2020$nTimesIncrFromZero\_5yrs==TRUE),  
 0,StockNames\_STATS\_2015\_2020$nTimesIncrFromZero\_5yrs)  
  
StockNames\_STATS\_2015\_2020$Direction5yr10PercentChange <-  
 ifelse(StockNames\_STATS\_2015\_2020$Direction5yr10PercentChange=='',0,StockNames\_STATS\_2015\_2020$Direction5yr10PercentChange)  
  
write.csv(StockNames\_STATS\_2015\_2020, 'StockNames\_STATS\_2015\_2020.csv', row.names=FALSE)  
  
show2 <- rbind(head(StockNames\_STATS\_2015\_2020,3),tail(StockNames\_STATS\_2015\_2020,3))  
show2

## stock  
## 1 AAL  
## 2 AAL  
## 3 AAL  
## 316 XOM  
## 317 XOM  
## 318 XOM  
## stockInfo  
## 1 American Airlines Group Inc. (AAL)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 2 American Airlines Group Inc. (AAL)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 3 American Airlines Group Inc. (AAL)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 316 Exxon Mobil Corporation (XOM)\nNYSE - NYSE Delayed Price. Currency in USD  
## 317 Exxon Mobil Corporation (XOM)\nNYSE - NYSE Delayed Price. Currency in USD  
## 318 Exxon Mobil Corporation (XOM)\nNYSE - NYSE Delayed Price. Currency in USD  
## stockExchange stock5yrMeans Year YearMeanValue Direction5yr10PercentChange  
## 1 Nasdaq 38.67371 2018 42.80195 down  
## 2 Nasdaq 38.67371 2017 47.49072 up10  
## 3 Nasdaq 38.67371 2020 27.56429 down10  
## 316 NYSE 78.73700 2017 81.86159 down  
## 317 NYSE 78.73700 2019 73.73464 down  
## 318 NYSE 78.73700 2015 82.82845 0  
## nTimesDecr10\_5yr nTimesIncr10\_5yr nTimesDecrFromZero\_5yrs  
## 1 3 1 1  
## 2 3 1 1  
## 3 3 1 1  
## 316 0 0 4  
## 317 0 0 4  
## 318 0 0 4  
## nTimesIncrFromZero\_5yrs  
## 1 0  
## 2 0  
## 3 0  
## 316 1  
## 317 1  
## 318 1

length(unique(StockNames\_STATS\_2015\_2020$stock))

## [1] 53

Lets the mean annual unemployment rates using the original table to combine with this table of the n times a stock increases/decreases per year in the last five years.

ue$Annual <- round(rowMeans(ue[,2:13], na.rm=T),2)  
ue\_15\_20 <- ue[9:14,c(1,14)]  
colnames(ue\_15\_20)[2] <- 'Annual\_UE'

Now, combine the unemployment and the newest stats with counts table.

stock\_5yrs\_ue <- merge(ue\_15\_20,StockNames\_STATS\_2015\_2020, by.x='Year', by.y='Year')

Add in a boolean field to show if the YearMeanValue is greater than the Stock5yrMeans column as a 1 if true and a 0 if not.

stock\_5yrs\_ue$YearMeanGreaterThan5yrMean <- ifelse(stock\_5yrs\_ue$YearMeanValue >  
 stock\_5yrs\_ue$stock5yrMeans,1,0)

write.csv(stock\_5yrs\_ue,'stock\_2015-2020\_ue.csv',row.names=FALSE)

Make separate portfolios for each of the stocks that increased by more than 10% annually more than at least 1 time, decreased more than 10% annually more than at least 1 time, then get the mean value of the YearMeanValue column. Compare this to the portfolio of the stocks that never decreased more than 10% annually.

sub\_D10 <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$nTimesDecr10\_5yr > 0)  
  
  
D10\_2015 <- subset(sub\_D10, sub\_D10$Year==2015)  
D10\_2020 <- subset(sub\_D10, sub\_D10$Year==2020)  
  
md10\_2015 <- mean(D10\_2015$YearMeanValue)  
md10\_2020 <- mean(D10\_2020$YearMeanValue)  
md10\_2015

## [1] 67.18499

md10\_2020

## [1] 65.57645

ROI\_D10 <- md10\_2020/md10\_2015  
ROI\_D10

## [1] 0.9760581

d10\_startValue <- md10\_2015\*length(unique(D10\_2015$stock))  
d10\_endValue <- md10\_2020\*length(unique(D10\_2020$stock))  
d10\_startValue

## [1] 2149.92

d10\_endValue

## [1] 2098.446

The above values show the 2015 average stock value of those stocks that decreased more than 10 percent in the last five years more than once was 67 USD. And in 2020 those stocks decreased in value to 66 USD giving it a five year ROI in the last five years of 0.976, or a decline of 2.4 percent in value. The 2015 value of this portfolio of stocks was 2150 USD, and in 2020 the portfolio value of the stocks was 2098 USD showing the dollar decrease over five years.

sub\_nvr\_D10 <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$nTimesDecr10\_5yr == 0)  
  
nD10\_2015 <- subset(sub\_nvr\_D10, sub\_nvr\_D10$Year==2015)  
nD10\_2020 <- subset(sub\_nvr\_D10, sub\_nvr\_D10$Year==2020)  
  
mnD10\_2015 <- mean(nD10\_2015$YearMeanValue)  
mnD10\_2020 <- mean(nD10\_2020$YearMeanValue)  
mnD10\_2015

## [1] 109.2018

mnD10\_2020

## [1] 272.5558

ROI\_nD10 <- mnD10\_2020/mnD10\_2015  
ROI\_nD10

## [1] 2.495891

nD10\_startValue <- mnD10\_2015\*length(unique(nD10\_2015$stock))  
nD10\_endValue <- mnD10\_2020\*length(unique(nD10\_2020$stock))  
nD10\_startValue

## [1] 2293.238

nD10\_endValue

## [1] 5723.671

The above numbers show the mean stock value of those stocks that never decreased by more than 10 percent in 2015-2020. The 2015 average stock price of these stocks was 109 USD, and in 2020 the average price was 273 USD. This was a ROI of 2.49 or 249 percent, which means it more than doubled in value over the last five years. The 2015 portfolio price of these specific stock were 2293 USD and in 2020 the portfolio price was 5724 USD. This shows that having a stock that never decreases by more than 10 percent in five years could be a good stock to buy.

Lets now do the reverse and look at those stocks that increased more than 10% at least three times in the last five years of 2015-2020 and compare the means.

sub\_I10 <- subset(StockNames\_STATS\_2015\_2020,   
 StockNames\_STATS\_2015\_2020$nTimesIncr10\_5yr > 3)  
  
sub\_nvr\_I10 <- subset(StockNames\_STATS\_2015\_2020,  
 StockNames\_STATS\_2015\_2020$nTimesIncr10\_5yr == 0)  
  
m2015 <- subset(sub\_I10, sub\_I10$Year==2015)  
m2020 <- subset(sub\_I10, sub\_I10$Year==2020)  
  
pm\_2015 <- mean(m2015$YearMeanValue)  
pm\_2020 <- mean(m2020$YearMeanValue)  
  
ROI\_incr10\_3x <- pm\_2020/pm\_2015  
ROI\_incr10\_3x

## [1] 2.553548

I10\_3\_startValue <- pm\_2015\*length(unique(m2015$stock))  
I10\_3\_endValue <- pm\_2020\*length(unique(m2020$stock))  
I10\_3\_startValue

## [1] 823.0797

I10\_3\_endValue

## [1] 2101.773

mn\_2015 <- subset(sub\_nvr\_I10, sub\_nvr\_I10$Year==2015)  
mn\_2020 <- subset(sub\_nvr\_I10, sub\_nvr\_I10$Year==2020)  
  
pmn\_2015 <- mean(mn\_2015$YearMeanValue)  
pmn\_2020 <- mean(mn\_2020$YearMeanValue)  
  
ROI\_nvr10 <- pmn\_2020/pmn\_2015  
ROI\_nvr10

## [1] 0.5342204

nI10\_startValue <- pmn\_2015\*length(unique(mn\_2015$stock))  
nI10\_endValue <- pmn\_2020\*length(unique(mn\_2020$stock))  
nI10\_startValue

## [1] 502.048

nI10\_endValue

## [1] 268.2043

From the above, we can see that those stocks that never increased by more than 10 percent during the last five years lost almost half their 2015 start value of 502 USD in 2015 and 268 USD in 2020 and having a ROI ratio of 0.53. On the other hand, the stocks that increased by more than 10 percent at least three times during the last five years had a ROI ratio of 2.55, a 2015 portfolio value of 823 USD and a 2020 portfolio value of 2102 USD.

Now lets look at those stocks that increased at least one time in the last five years but never by more than 10 percent.

sub\_Iz <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$nTimesIncrFromZero\_5yr > 0)  
  
Iz\_2015 <- subset(sub\_Iz, sub\_Iz$Year==2015)  
Iz\_2020 <- subset(sub\_Iz, sub\_Iz$Year==2020)  
  
Iz\_2015 <- subset(sub\_Iz, sub\_Iz$Year==2015)  
Iz\_2020 <- subset(sub\_Iz, sub\_Iz$Year==2020)  
  
m\_Iz\_2015 <- mean(Iz\_2015$YearMeanValue)  
m\_Iz\_2020 <- mean(Iz\_2020$YearMeanValue)  
m\_Iz\_2015

## [1] 100.1005

m\_Iz\_2020

## [1] 194.7926

ROI\_Iz <- m\_Iz\_2020/m\_Iz\_2015  
ROI\_Iz

## [1] 1.945971

p\_Iz\_2015 <- m\_Iz\_2015\*length(unique(sub\_Iz$stock))  
p\_Iz\_2020 <- m\_Iz\_2020\*length(unique(sub\_Iz$stock))  
p\_Iz\_2015

## [1] 3503.518

p\_Iz\_2020

## [1] 6817.742

From the data above, the 2015 average stock price of 100 USD for the stock that had an increasing year at least one time in the last five years but not by more than 10 percent of the last year value. The 2020 average stock price increased to 195 USD, with an ROI of 1.95 or 195 percent. The 2015 portfolio value was 3504 USD and in 2020 the portfolio value increased to 6818 USD. This makes sense that those stocks that increase are good to have as they are making you money, but even if they don’t increase by more than 10 percent in any year, when combined with other increasing stock they can nearly double your investment over five years.

Here are the stocks that never increased in the last five years.

sub\_nvr\_Iz <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$nTimesIncrFromZero\_5yr == 0)  
  
nIz\_2015 <- subset(sub\_nvr\_Iz, sub\_nvr\_Iz$Year==2015)  
nIz\_2020 <- subset(sub\_nvr\_Iz, sub\_nvr\_Iz$Year==2020)  
  
m\_nIz\_2015 <- mean(nIz\_2015$YearMeanValue)  
m\_nIz\_2020 <- mean(nIz\_2020$YearMeanValue)  
m\_nIz\_2015

## [1] 52.2022

m\_nIz\_2020

## [1] 55.79861

ROI\_nIz <- m\_nIz\_2020/m\_nIz\_2015  
ROI\_nIz

## [1] 1.068894

nIz\_startValue <- m\_nIz\_2015\*length(unique(nIz\_2015$stock))  
nIz\_endValue <- m\_nIz\_2020\*length(unique(nIz\_2020$stock))  
nIz\_startValue

## [1] 939.6396

nIz\_endValue

## [1] 1004.375

From the above we have a portfolio of stock that never increased from zero, but might have increased by more than 10 percent.This variable was designed to capture exactly those stocks that did not increase by more than 10 percent but did increase some. This portfolio has a 2015 average stock value of 52 USD and this value increases to 56 USD in 2020 with an ROI of 1.07 or a five year interest of 7 percent. The 2015 portfolio value was 940 USD and in 2020 the portfolio value was 1004 USD. \*\*\*

Lets get the entire 53 stock portfolio mean value in 2015 and compare to the same 53 stock portfolio mean value in 2020.

p2015 <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$Year==2015)  
p2020 <- subset(StockNames\_STATS\_2015\_2020, StockNames\_STATS\_2015\_2020$Year==2020)  
  
pm2015 <- mean(p2015$YearMeanValue)  
pm2020 <- mean(p2020$YearMeanValue)  
  
pm2015

## [1] 83.83316

pm2020

## [1] 147.5871

ROI\_all <- pm2020/pm2015  
ROI\_all

## [1] 1.760486

pm2015\*length(unique(StockNames\_STATS\_2015\_2020$stock))

## [1] 4443.157

pm2020\*length(unique(StockNames\_STATS\_2015\_2020$stock))

## [1] 7822.117

The portfolio mean was 84 USD in 2015 and 147 USD in 2020. The 2015 portfolio was valued at 4443 USD and in 2020 at 7822 USD for all stocks. The ROI is 1.76, which is good because you almost doubled the loan with all 53 of these stocks in five years spanning 2015-2020.

76/5

## [1] 15.2

So, with a return of 76% on top of the value invested in 2015 figuratively for this example, that is 76/5 or 15.2% annual interest earned each of five years. This is called pooling, that the wins over compensate for the losses and it is used in health insurance companies as well as financial portfolios like 401k investment tools.

What would the ROI be for all stocks that increased during the last five years by more than 10 per cent?

incr\_10\_2015 <- subset(sub\_I10, sub\_I10$Year==2015)  
mean\_incr\_10\_2015 <- mean(incr\_10\_2015$YearMeanValue)  
  
incr\_10\_2020 <- subset(sub\_I10, sub\_I10$Year==2020)  
mean\_incr\_10\_2020 <- mean(incr\_10\_2020$YearMeanValue)  
  
mean\_incr\_10\_2015

## [1] 117.5828

mean\_incr\_10\_2020

## [1] 300.2533

ROI\_Incr\_10 <- mean\_incr\_10\_2020/mean\_incr\_10\_2015  
ROI\_Incr\_10

## [1] 2.553548

value2015 <- mean\_incr\_10\_2015\*length(unique(sub\_I10$stock))  
value2020 <- mean\_incr\_10\_2020\*length(unique(sub\_I10$stock))  
  
value2015

## [1] 823.0797

value2020

## [1] 2101.773

The **return on investment** is more than doubled to 2102 USD over five years from a value of 823 USD in 2015 by selecting only the stocks in this portfolio of stocks that increased by more than 10% at least once in the last 5 years. The return of the ratio of the mean value in 2020 to 2015 is 2.55, which means the portfolio more than doubled.

But how do we or how can we know what stocks to select now that will increase many times as long as we have the investment? Can machine learning be built from this data set to find the stocks in this sample that produce good indicating features of other stocks that could be profitable to buy? We will develop this as we progress through this portfolio. We would also want indicators that would tell if certain stocks look like a good prospect but are actually going to be on a steady decline that translates to financial loss as long as you own them.

There are four sets of counts for those that increased more than 10%, decreased more than 10%, increased more than zero but less than 10%, and decreased more than zero but less than 10% within the five year span from 2015-2020. Lets see if there is a better subset of choices for a better market portfolio.

Lets add a five year poisson column using lambda=(unemployment rate), time=(nTimesIncr10\_5yr), and k=(YearMeanGreaterThan5yrMean).We will use the best subset so far of the stocks that increased by more than 10% annually in at least 3 out of the last five years.

ue2 <- stock\_5yrs\_ue$Annual\_UE  
t <- stock\_5yrs\_ue$nTimesIncr10\_5yr  
k <- stock\_5yrs\_ue$YearMeanGreaterThan5yrMean  
stock\_5yrs\_ue$poisson5yrUE <- round((exp(-ue2\*t)\*(ue2\*t)^k)/(factorial(k)),5)

Lets get a subset of those stocks that have cyclical patterns within five years, so that we have three years the stock increases more than 10% exactly 3 times, and two years where the stock decreases less than 10% exactly 2 times. Separately, get the stocks it increases greater than 10% exactly 3 times, and decreases more than 10% exactly 2 times. Also get the reverse of these values

cyclical <- subset(stock\_5yrs\_ue, stock\_5yrs\_ue$nTimesIncr10\_5yr==3 & (stock\_5yrs\_ue$nTimesDecr10\_5yr==2 | stock\_5yrs\_ue$nTimesDecrFromZero\_5yrs==2))  
  
cyclical2 <- subset(stock\_5yrs\_ue, stock\_5yrs\_ue$nTimesIncrFromZero\_5yrs >=2 & (stock\_5yrs\_ue$nTimesDecr10\_5yr >= 2 | stock\_5yrs\_ue$nTimesDecrFromZero\_5yrs >= 2))  
  
c1 <- as.character(unique(cyclical$stock))  
c2 <- as.character(unique(cyclical2$stock))  
cycle <- c(c1,c2)  
cycle1 <- as.data.frame(cycle)  
colnames(cycle1) <- 'Stock'  
  
portCycle <- merge(cycle1,stock\_5yrs\_ue, by.x='Stock', by.y='stock')  
portCycle\_2015 <- subset(portCycle, Year==2015)  
portCycle\_2020 <- subset(portCycle, Year==2020)  
  
pc\_mean2015 <- mean(portCycle\_2015$YearMeanValue)  
pc\_mean2020 <- mean(portCycle\_2020$YearMeanValue)  
  
pc\_mean2015

## [1] 36.228

pc\_mean2020

## [1] 37.91673

ROI\_pc <- pc\_mean2020/pc\_mean2015  
ROI\_pc

## [1] 1.046614

startValue <- pc\_mean2015\*length(unique(portCycle\_2015$Stock))  
endValue <- pc\_mean2020\*length(unique(portCycle\_2020$Stock))  
  
startValue

## [1] 434.736

endValue

## [1] 455.0007

The above shows that the **cyclical stocks that have highs and lows the time span of the loan aren’t great investments**, as these stocks started at 435 USD in 2015 but ended with a portfolio value of 455 USD over a five year time span from 2015-2020. The ratio of average stock in 2020 to average stock in 2015 is 1.04, which means it earned 4 perent interest over 5 years or less than 1 percent interest a year. This is equivalent to most bank savings accounts. It is good they at least stayed the same and didn’t cause the portfolio to lose money, and we can assume those stocks that do decrease continuously will be the stocks that lose money. We should look at the columns we added earlier that calculated the ROI dollars for each stock and see the average number of times the stock closes at a decreasing value over the span of the original data. Then use those outcomes to rank the stock a poor, average, good, or great stock to buy.

Lets use the StocksSTATS table with the 230 columns of ROI for each stock from the start in 2007 throughout 2020 and the daily changes for each stock for the same time span. We could add cumulative sum columns to each stock or just plot the daily changes for each of the 53 stocks and see if we notice any patterns and compare the the final recording ROI from the initial investment. Maybe see if some of these stocks are good to jump on, like a wave to increase value of the portfolio, or drop the stock at some point to keep the portfolio from dropping in value.

dailyChange <- grep('dailyChange',colnames(StocksSTATS))  
DailyChanges <- StocksSTATS[,c(dailyChange,218:222,229)]  
summary(DailyChanges)

## TGT\_dailyChange FTR\_dailyChange UBSI\_dailyChange   
## Min. :-66.41001 Min. :-81.00000 Min. :-16.2500   
## 1st Qu.: -0.51000 1st Qu.: -0.90000 1st Qu.: -0.3500   
## Median : 0.03000 Median : 0.00000 Median : 0.0000   
## Mean : 0.03247 Mean : 0.00026 Mean : 0.0115   
## 3rd Qu.: 0.57000 3rd Qu.: 0.75000 3rd Qu.: 0.3300   
## Max. : 59.93000 Max. :235.68000 Max. : 34.8000   
## HD\_dailyChange JPM\_dailyChange XOM\_dailyChange   
## Min. :-199.42000 Min. :-88.18999 Min. :-23.84000   
## 1st Qu.: -0.43999 1st Qu.: -0.51000 1st Qu.: -0.58000   
## Median : 0.06000 Median : 0.02000 Median : 0.00000   
## Mean : 0.07046 Mean : 0.03574 Mean : 0.02144   
## 3rd Qu.: 0.67000 3rd Qu.: 0.58000 3rd Qu.: 0.61000   
## Max. : 57.01999 Max. : 48.24000 Max. : 76.16000   
## CVX\_dailyChange NSANY\_dailyChange MGM\_dailyChange   
## Min. :-56.84000 Min. :-15.639999 Min. :-73.43000   
## 1st Qu.: -0.78000 1st Qu.: -0.170000 1st Qu.: -0.30000   
## Median : 0.08000 Median : 0.000000 Median : 0.01000   
## Mean : 0.03602 Mean : 0.003793 Mean : 0.00842   
## 3rd Qu.: 0.88000 3rd Qu.: 0.170000 3rd Qu.: 0.32000   
## Max. : 74.83000 Max. : 21.540001 Max. : 71.58000   
## TEVA\_dailyChange HST\_dailyChange WFC\_dailyChange   
## Min. :-30.11000 Min. :-13.010280 Min. :-18.84000   
## 1st Qu.: -0.38000 1st Qu.: -0.190000 1st Qu.: -0.37000   
## Median : -0.02000 Median : 0.010000 Median : 0.00000   
## Mean : 0.00209 Mean : 0.005251 Mean : 0.01532   
## 3rd Qu.: 0.34000 3rd Qu.: 0.200000 3rd Qu.: 0.38000   
## Max. : 38.18000 Max. : 26.206558 Max. : 34.01000   
## WWE\_dailyChange INO\_dailyChange SCE.PB\_dailyChange   
## Min. :-69.75000 Min. :-11.520000 Min. :-890.0000   
## 1st Qu.: -0.18000 1st Qu.: -0.120000 1st Qu.: -19.0000   
## Median : 0.01000 Median : 0.000000 Median : 0.0000   
## Mean : 0.02161 Mean : 0.000623 Mean : 0.2712   
## 3rd Qu.: 0.21000 3rd Qu.: 0.080000 3rd Qu.: 22.0000   
## Max. : 67.69000 Max. : 13.320000 Max. : 886.0000   
## FFIN\_dailyChange GOOG\_dailyChange WM\_dailyChange   
## Min. :-28.191665 Min. :-1170.0303 Min. :-84.98000   
## 1st Qu.: -0.120000 1st Qu.: -3.1314 1st Qu.: -0.23000   
## Median : 0.006667 Median : 0.2989 Median : 0.05000   
## Mean : 0.010121 Mean : 0.3702 Mean : 0.03492   
## 3rd Qu.: 0.155000 3rd Qu.: 4.4200 3rd Qu.: 0.34000   
## Max. : 9.840002 Max. : 344.4899 Max. : 34.65000   
## ONCY\_dailyChange S\_dailyChange F\_dailyChange   
## Min. :-25.08000 Min. :-14.830000 Min. :-6.200001   
## 1st Qu.: -0.28500 1st Qu.: -0.100000 1st Qu.:-0.120000   
## Median : 0.00000 Median : 0.000000 Median : 0.000000   
## Mean : 0.00017 Mean : 0.001874 Mean : 0.002782   
## 3rd Qu.: 0.19000 3rd Qu.: 0.090000 3rd Qu.: 0.120000   
## Max. : 41.23000 Max. : 19.299999 Max. : 9.780000   
## ARWR\_dailyChange COST\_dailyChange AAL\_dailyChange   
## Min. :-49.50000 Min. :-245.26999 Min. :-42.66000   
## 1st Qu.: -0.20000 1st Qu.: -0.60000 1st Qu.: -0.36000   
## Median : 0.00000 Median : 0.10000 Median : 0.00000   
## Mean : 0.00856 Mean : 0.08749 Mean : 0.00819   
## 3rd Qu.: 0.20000 3rd Qu.: 0.90000 3rd Qu.: 0.37000   
## Max. : 50.91000 Max. : 93.68001 Max. : 45.05000   
## JWN\_dailyChange NUS\_dailyChange HMC\_dailyChange   
## Min. :-38.08000 Min. :-96.59000 Min. :-13.20000   
## 1st Qu.: -0.54000 1st Qu.: -0.41000 1st Qu.: -0.31000   
## Median : 0.03000 Median : 0.02000 Median : 0.00000   
## Mean : 0.01022 Mean : 0.01292 Mean : 0.00792   
## 3rd Qu.: 0.58000 3rd Qu.: 0.52000 3rd Qu.: 0.30000   
## Max. : 53.74000 Max. : 95.71001 Max. : 34.69000   
## AMZN\_dailyChange T\_dailyChange HRB\_dailyChange   
## Min. :-1939.1100 Min. :-15.26000 Min. :-14.630001   
## 1st Qu.: -2.4300 1st Qu.: -0.22000 1st Qu.: -0.220001   
## Median : 0.3200 Median : 0.02000 Median : 0.010000   
## Mean : 0.5272 Mean : 0.01149 Mean : 0.007173   
## 3rd Qu.: 4.1200 3rd Qu.: 0.24000 3rd Qu.: 0.230001   
## Max. : 1078.1600 Max. : 39.46000 Max. : 21.260000   
## RRGB\_dailyChange ADDYY\_dailyChange PCG\_dailyChange   
## Min. :-43.2600 Min. :-138.60000 Min. :-36.91000   
## 1st Qu.: -0.5200 1st Qu.: -0.45000 1st Qu.: -0.34000   
## Median : 0.0000 Median : 0.04000 Median : 0.03000   
## Mean : 0.0101 Mean : 0.04728 Mean : 0.00304   
## 3rd Qu.: 0.5600 3rd Qu.: 0.57001 3rd Qu.: 0.35000   
## Max. : 42.7500 Max. : 48.06000 Max. : 49.44000   
## ROST\_dailyChange JNJ\_dailyChange NFLX\_dailyChange   
## Min. :-109.62500 Min. :-87.02000 Min. :-368.0886   
## 1st Qu.: -0.19000 1st Qu.: -0.38000 1st Qu.: -0.4286   
## Median : 0.02750 Median : 0.03000 Median : 0.0143   
## Mean : 0.03336 Mean : 0.03929 Mean : 0.0813   
## 3rd Qu.: 0.32500 3rd Qu.: 0.50000 3rd Qu.: 0.6243   
## Max. : 35.09000 Max. : 60.10000 Max. : 216.5200   
## M\_dailyChange KSS\_dailyChange DLTR\_dailyChange   
## Min. :-35.46000 Min. :-31.74000 Min. :-98.37333   
## 1st Qu.: -0.40000 1st Qu.: -0.59000 1st Qu.: -0.32667   
## Median : 0.02000 Median : 0.01000 Median : 0.02333   
## Mean : 0.00472 Mean : 0.01508 Mean : 0.03467   
## 3rd Qu.: 0.44000 3rd Qu.: 0.61000 3rd Qu.: 0.46000   
## Max. : 45.06000 Max. : 77.49000 Max. : 37.19000   
## WMT\_dailyChange C\_dailyChange AAP\_dailyChange   
## Min. :-74.62000 Min. :-298.300 Min. :-131.92001   
## 1st Qu.: -0.39000 1st Qu.: -0.680 1st Qu.: -0.74000   
## Median : 0.04000 Median : -0.010 Median : 0.04000   
## Mean : 0.03604 Mean : 0.021 Mean : 0.05023   
## 3rd Qu.: 0.46000 3rd Qu.: 0.650 3rd Qu.: 0.85001   
## Max. : 47.40000 Max. : 510.500 Max. : 87.86001   
## JBLU\_dailyChange MSFT\_dailyChange KGJI\_dailyChange   
## Min. :-10.390000 Min. :-140.49000 Min. :-7.820000   
## 1st Qu.: -0.139999 1st Qu.: -0.29000 1st Qu.:-0.020000   
## Median : 0.000000 Median : 0.03000 Median : 0.000000   
## Mean : 0.005087 Mean : 0.04222 Mean : 0.000182   
## 3rd Qu.: 0.140000 3rd Qu.: 0.39000 3rd Qu.: 0.020000   
## Max. : 14.690000 Max. : 56.19000 Max. : 9.090000   
## EPD\_dailyChange TJX\_dailyChange HOFT\_dailyChange   
## Min. :-14.510001 Min. :-53.93750 Min. :-23.229999   
## 1st Qu.: -0.170000 1st Qu.: -0.14250 1st Qu.: -0.240000   
## Median : 0.015000 Median : 0.02000 Median : 0.000000   
## Mean : 0.008679 Mean : 0.01693 Mean : 0.006511   
## 3rd Qu.: 0.195000 3rd Qu.: 0.21250 3rd Qu.: 0.250000   
## Max. : 15.875000 Max. : 21.21500 Max. : 22.400002   
## LUV\_dailyChange NKE\_dailyChange TM\_dailyChange VZ\_dailyChange   
## Min. :-42.8500 Min. :-88.72500 Min. :-47.61000 Min. :-25.92523   
## 1st Qu.: -0.2100 1st Qu.: -0.23250 1st Qu.: -0.84000 1st Qu.: -0.29000   
## Median : 0.0100 Median : 0.02750 Median : -0.02000 Median : 0.02814   
## Mean : 0.0164 Mean : 0.02852 Mean : 0.04084 Mean : 0.01833   
## 3rd Qu.: 0.2500 3rd Qu.: 0.31750 3rd Qu.: 0.85999 3rd Qu.: 0.31888   
## Max. : 25.0600 Max. : 29.47000 Max. :126.92000 Max. : 35.31607   
## SIG\_dailyChange Date DayOfWeek Month   
## Min. :-75.22000 Min. :2007-01-03 Length:3293 Length:3293   
## 1st Qu.: -0.61000 1st Qu.:2010-04-12 Class :character Class :character   
## Median : 0.03000 Median :2013-07-18 Mode :character Mode :character   
## Mean : 0.00509 Mean :2013-07-16   
## 3rd Qu.: 0.61000 3rd Qu.:2016-10-21   
## Max. : 50.72000 Max. :2020-01-31   
## Year UE\_monthlyRate dayOfMonth   
## Min. :2007 Min. : 3.500 Min. : 1.00   
## 1st Qu.:2010 1st Qu.: 4.600 1st Qu.: 8.00   
## Median :2013 Median : 5.600 Median :16.00   
## Mean :2013 Mean : 6.282 Mean :15.74   
## 3rd Qu.:2016 3rd Qu.: 8.200 3rd Qu.:23.00   
## Max. :2020 Max. :10.000 Max. :31.00

dailyChangesColSums <- as.data.frame(colSums(DailyChanges[1:53]))  
colnames(dailyChangesColSums) <- 'avgDailyChange\_2007\_2020'  
row.names(dailyChangesColSums) <- gsub('\_dailyChange','',row.names(dailyChangesColSums))  
head(dailyChangesColSums,5)

## avgDailyChange\_2007\_2020  
## TGT 106.91  
## FTR 0.87  
## UBSI 37.87  
## HD 232.02  
## JPM 117.69

The DOW Industrial Jones average was also downloaded from [Yahoo Finance](https://finance.yahoo.com/quote/%5EDJI/history/) to see a bigger picture of these daily changes by adding in the change in the DOW. We will upload it to our data and put the daily change values into a new column with the Close of the DOW daily.

dow <- read.csv('DOW.csv', sep=',', header=T, na.strings=c('',' '))  
head(dow)

## Date Open High Low Close Adj.Close Volume  
## 1 2007-01-03 12459.54 12580.35 12404.82 12474.52 12474.52 327200000  
## 2 2007-01-04 12473.16 12510.41 12403.86 12480.69 12480.69 259060000  
## 3 2007-01-05 12480.05 12480.13 12365.41 12398.01 12398.01 235220000  
## 4 2007-01-08 12392.01 12445.92 12337.37 12423.49 12423.49 223500000  
## 5 2007-01-09 12424.77 12466.43 12369.17 12416.60 12416.60 225190000  
## 6 2007-01-10 12417.00 12451.61 12355.63 12442.16 12442.16 226570000

Lets keep the date, close, and volume columns.

dow1 <- dow[,c(1,5,7)]  
colnames(dow1) <- c('Date','DOW\_Daily\_Close','DOW\_Daily\_Volume')  
head(dow1)

## Date DOW\_Daily\_Close DOW\_Daily\_Volume  
## 1 2007-01-03 12474.52 327200000  
## 2 2007-01-04 12480.69 259060000  
## 3 2007-01-05 12398.01 235220000  
## 4 2007-01-08 12423.49 223500000  
## 5 2007-01-09 12416.60 225190000  
## 6 2007-01-10 12442.16 226570000

Now add in a daily change column to the dow1 table.

dow\_a <- dow1$DOW\_Daily\_Close  
dow\_b <- c(0,dow\_a)  
dow\_c <- dow\_b[1:(length(dow\_b)-1)]  
dow1$DOW\_Daily\_Change <- dow1$DOW\_Daily\_Close-dow\_c  
head(dow1)

## Date DOW\_Daily\_Close DOW\_Daily\_Volume DOW\_Daily\_Change  
## 1 2007-01-03 12474.52 327200000 12474.519531  
## 2 2007-01-04 12480.69 259060000 6.170899  
## 3 2007-01-05 12398.01 235220000 -82.680664  
## 4 2007-01-08 12423.49 223500000 25.480468  
## 5 2007-01-09 12416.60 225190000 -6.890625  
## 6 2007-01-10 12442.16 226570000 25.560547

Lets attach the daily change of the DOW to the table of daily changes per stock we made earlier and compare.

dow1$Date <- as.Date(dow1$Date)  
DailyChanges2 <- merge(DailyChanges, dow1, by.x='Date', by.y='Date')  
colnames(DailyChanges2)

## [1] "Date" "TGT\_dailyChange" "FTR\_dailyChange"   
## [4] "UBSI\_dailyChange" "HD\_dailyChange" "JPM\_dailyChange"   
## [7] "XOM\_dailyChange" "CVX\_dailyChange" "NSANY\_dailyChange"   
## [10] "MGM\_dailyChange" "TEVA\_dailyChange" "HST\_dailyChange"   
## [13] "WFC\_dailyChange" "WWE\_dailyChange" "INO\_dailyChange"   
## [16] "SCE.PB\_dailyChange" "FFIN\_dailyChange" "GOOG\_dailyChange"   
## [19] "WM\_dailyChange" "ONCY\_dailyChange" "S\_dailyChange"   
## [22] "F\_dailyChange" "ARWR\_dailyChange" "COST\_dailyChange"   
## [25] "AAL\_dailyChange" "JWN\_dailyChange" "NUS\_dailyChange"   
## [28] "HMC\_dailyChange" "AMZN\_dailyChange" "T\_dailyChange"   
## [31] "HRB\_dailyChange" "RRGB\_dailyChange" "ADDYY\_dailyChange"   
## [34] "PCG\_dailyChange" "ROST\_dailyChange" "JNJ\_dailyChange"   
## [37] "NFLX\_dailyChange" "M\_dailyChange" "KSS\_dailyChange"   
## [40] "DLTR\_dailyChange" "WMT\_dailyChange" "C\_dailyChange"   
## [43] "AAP\_dailyChange" "JBLU\_dailyChange" "MSFT\_dailyChange"   
## [46] "KGJI\_dailyChange" "EPD\_dailyChange" "TJX\_dailyChange"   
## [49] "HOFT\_dailyChange" "LUV\_dailyChange" "NKE\_dailyChange"   
## [52] "TM\_dailyChange" "VZ\_dailyChange" "SIG\_dailyChange"   
## [55] "DayOfWeek" "Month" "Year"   
## [58] "UE\_monthlyRate" "dayOfMonth" "DOW\_Daily\_Close"   
## [61] "DOW\_Daily\_Volume" "DOW\_Daily\_Change"

Lets add an indicator for increasing or decreasing unemployment rate per month.

DailyChanges2$lastMonth\_UE\_rate <-  
 c(DailyChanges2$UE\_monthlyRate[1],  
 DailyChanges2$UE\_monthlyRate[1:length(DailyChanges2$UE\_monthlyRate)-1])  
  
DailyChanges2$increasingMonthly\_UE\_rate <- ifelse((DailyChanges2$UE\_monthlyRate-DailyChanges2$lastMonth\_UE\_rate) > 0, 1, 0)

Save this file to csv.

write.csv(DailyChanges2, 'DailyChanges\_UE\_DOW\_07\_20.csv', row.names=FALSE)

Lets see a summary of our date with summaries when the unemployment rate increased the next month and the DOW daily changes increased the next day and separately a subset of the DOW decreasing daily.This will see if the DOW is affected by the increasing unemployment rate or not. And also show which stocks are increasing when the DOW is decreasing and unemployment rate increasing to indicate great public sentiment for those stocks during poor public sentiment about the state of the economy.

dow\_up\_ue\_up <- subset(DailyChanges2, DailyChanges2$increasingMonthly\_UE\_rate==1 &   
 DailyChanges2$DOW\_Daily\_Change >= 0)  
  
dow\_down\_ue\_up <- subset(DailyChanges2, DailyChanges2$increasingMonthly\_UE\_rate==1 &   
 DailyChanges2$DOW\_Daily\_Change < 0)

Summary of the DOW up and unemployment up:

summary(dow\_up\_ue\_up)

## Date TGT\_dailyChange FTR\_dailyChange UBSI\_dailyChange   
## Min. :2007-04-02 Min. :-46.9000 Min. :-81.00 Min. :-10.700   
## 1st Qu.:2009-04-16 1st Qu.:-13.2650 1st Qu.:-40.95 1st Qu.: -4.000   
## Median :2013-01-02 Median : -6.5100 Median : -3.15 Median : -1.120   
## Mean :2013-03-11 Mean : -0.7819 Mean : 13.59 Mean : 0.489   
## 3rd Qu.:2017-02-15 3rd Qu.: 7.5500 3rd Qu.: 10.20 3rd Qu.: 3.860   
## Max. :2020-01-02 Max. : 59.9300 Max. :235.68 Max. : 34.800   
## HD\_dailyChange JPM\_dailyChange XOM\_dailyChange CVX\_dailyChange   
## Min. :-198.220 Min. :-83.210 Min. :-23.840 Min. :-40.880   
## 1st Qu.: -6.020 1st Qu.: -7.530 1st Qu.: -7.665 1st Qu.:-15.290   
## Median : 2.790 Median : 1.840 Median : -2.420 Median : 3.410   
## Mean : -8.685 Mean : -2.735 Mean : 1.085 Mean : -1.455   
## 3rd Qu.: 22.855 3rd Qu.: 7.700 3rd Qu.: 6.285 3rd Qu.: 8.180   
## Max. : 41.820 Max. : 48.240 Max. : 76.160 Max. : 74.830   
## NSANY\_dailyChange MGM\_dailyChange TEVA\_dailyChange HST\_dailyChange   
## Min. :-11.7900 Min. :-57.280 Min. :-23.680 Min. :-13.0103   
## 1st Qu.: -4.2400 1st Qu.: -8.315 1st Qu.: -6.195 1st Qu.: -4.3800   
## Median : 0.2600 Median : -0.780 Median : -1.280 Median : 0.1700   
## Mean : 0.1368 Mean : -1.060 Mean : 3.095 Mean : 0.1865   
## 3rd Qu.: 2.1000 3rd Qu.: 5.685 3rd Qu.: 9.995 3rd Qu.: 4.6449   
## Max. : 21.5400 Max. : 71.580 Max. : 37.410 Max. : 26.2066   
## WFC\_dailyChange WWE\_dailyChange INO\_dailyChange SCE.PB\_dailyChange  
## Min. :-18.840 Min. :-55.090 Min. :-10.8800 Min. :-878.000   
## 1st Qu.: -7.195 1st Qu.: -4.690 1st Qu.: -1.8000 1st Qu.:-414.000   
## Median : -2.110 Median : 0.290 Median : -1.0000 Median : 16.000   
## Mean : -1.618 Mean : 1.587 Mean : 0.3177 Mean : 6.774   
## 3rd Qu.: 3.285 3rd Qu.: 4.820 3rd Qu.: 3.2150 3rd Qu.: 362.500   
## Max. : 34.010 Max. : 67.690 Max. : 13.3200 Max. : 857.000   
## FFIN\_dailyChange GOOG\_dailyChange WM\_dailyChange ONCY\_dailyChange   
## Min. :-27.0050 Min. :-1170.03 Min. :-81.960 Min. :-15.105   
## 1st Qu.: -0.1725 1st Qu.: -72.16 1st Qu.: -1.530 1st Qu.: -4.037   
## Median : 0.8500 Median : 32.93 Median : 2.970 Median : -0.760   
## Mean : -0.6908 Mean : -41.75 Mean : -2.227 Mean : 2.146   
## 3rd Qu.: 3.0767 3rd Qu.: 168.05 3rd Qu.: 10.125 3rd Qu.: 5.035   
## Max. : 9.8400 Max. : 267.46 Max. : 34.650 Max. : 28.690   
## S\_dailyChange F\_dailyChange ARWR\_dailyChange COST\_dailyChange   
## Min. :-14.8300 Min. :-5.520 Min. :-49.500 Min. :-245.270   
## 1st Qu.: -4.0250 1st Qu.:-1.775 1st Qu.: -8.215 1st Qu.: -8.480   
## Median : -0.1800 Median :-0.600 Median : -0.470 Median : 12.580   
## Mean : 0.3581 Mean : 0.129 Mean : 1.236 Mean : -8.776   
## 3rd Qu.: 1.8500 3rd Qu.: 0.800 3rd Qu.: 9.550 3rd Qu.: 24.215   
## Max. : 19.3000 Max. : 9.780 Max. : 49.000 Max. : 76.860   
## AAL\_dailyChange JWN\_dailyChange NUS\_dailyChange HMC\_dailyChange   
## Min. :-28.6000 Min. :-36.350 Min. :-42.910 Min. :-9.7300   
## 1st Qu.: -6.9450 1st Qu.:-16.185 1st Qu.:-13.845 1st Qu.:-4.3500   
## Median : 0.0500 Median : -1.520 Median : -2.010 Median :-0.9600   
## Mean : -0.2045 Mean : -1.158 Mean : -4.774 Mean : 0.7997   
## 3rd Qu.: 5.3500 3rd Qu.: 7.755 3rd Qu.: 4.405 3rd Qu.: 3.1850   
## Max. : 45.0500 Max. : 53.740 Max. : 20.330 Max. :34.6900   
## AMZN\_dailyChange T\_dailyChange HRB\_dailyChange RRGB\_dailyChange  
## Min. :-1939.11 Min. :-13.89000 Min. :-14.6300 Min. :-33.04   
## 1st Qu.: -3.79 1st Qu.: -5.51000 1st Qu.: -3.9850 1st Qu.:-19.39   
## Median : 40.42 Median : -0.20000 Median : -0.1100 Median : 1.02   
## Mean : -33.22 Mean : 0.02839 Mean : -0.5958 Mean : -4.03   
## 3rd Qu.: 216.99 3rd Qu.: 3.72000 3rd Qu.: 2.0500 3rd Qu.: 7.43   
## Max. : 899.08 Max. : 39.46000 Max. : 21.2600 Max. : 39.10   
## ADDYY\_dailyChange PCG\_dailyChange ROST\_dailyChange JNJ\_dailyChange   
## Min. :-127.610 Min. :-24.670 Min. :-104.5000 Min. :-87.020   
## 1st Qu.: -12.300 1st Qu.: -2.170 1st Qu.: 0.4012 1st Qu.: -8.945   
## Median : 3.770 Median : 0.700 Median : 1.9225 Median : 0.850   
## Mean : -5.184 Mean : 3.108 Mean : -4.3563 Mean : -2.412   
## 3rd Qu.: 16.255 3rd Qu.: 6.950 3rd Qu.: 6.1050 3rd Qu.: 11.580   
## Max. : 44.310 Max. : 49.440 Max. : 35.0900 Max. : 60.100   
## NFLX\_dailyChange M\_dailyChange KSS\_dailyChange DLTR\_dailyChange   
## Min. :-342.307 Min. :-34.4400 Min. :-31.740 Min. :-95.987   
## 1st Qu.: -3.334 1st Qu.:-14.3050 1st Qu.: -9.950 1st Qu.: -3.310   
## Median : 2.124 Median : 1.1500 Median : -1.500 Median : 2.647   
## Mean : -4.954 Mean : -0.9623 Mean : 1.922 Mean : -5.528   
## 3rd Qu.: 26.558 3rd Qu.: 10.6000 3rd Qu.: 10.290 3rd Qu.: 8.771   
## Max. : 210.520 Max. : 45.0600 Max. : 77.490 Max. : 24.700   
## WMT\_dailyChange C\_dailyChange AAP\_dailyChange JBLU\_dailyChange   
## Min. :-73.960 Min. :-294.400 Min. :-127.1000 Min. :-9.9600   
## 1st Qu.: -6.650 1st Qu.: -17.600 1st Qu.: -4.7650 1st Qu.:-3.5800   
## Median : 0.540 Median : -1.570 Median : 4.3000 Median :-0.0200   
## Mean : -3.403 Mean : 11.267 Mean : 0.7365 Mean :-0.1635   
## 3rd Qu.: 8.530 3rd Qu.: 9.565 3rd Qu.: 23.3150 3rd Qu.: 1.6400   
## Max. : 47.400 Max. : 510.500 Max. : 87.8600 Max. :11.9000   
## MSFT\_dailyChange KGJI\_dailyChange EPD\_dailyChange TJX\_dailyChange   
## Min. :-140.490 Min. :-1.9700 Min. :-14.5100 Min. :-52.1600   
## 1st Qu.: -4.260 1st Qu.:-0.6050 1st Qu.: -3.1300 1st Qu.: -0.6175   
## Median : 6.110 Median :-0.3200 Median : -0.1400 Median : 1.8675   
## Mean : -3.038 Mean : 0.2361 Mean : -0.5506 Mean : -1.7903   
## 3rd Qu.: 14.225 3rd Qu.: 0.5200 3rd Qu.: 2.9700 3rd Qu.: 4.8825   
## Max. : 56.190 Max. : 7.5600 Max. : 15.8750 Max. : 21.2150   
## HOFT\_dailyChange LUV\_dailyChange NKE\_dailyChange TM\_dailyChange   
## Min. :-19.6400 Min. :-40.830 Min. :-81.700 Min. :-41.950   
## 1st Qu.: -4.4550 1st Qu.: -6.485 1st Qu.: -1.359 1st Qu.:-19.645   
## Median : -2.1300 Median : -1.450 Median : 2.765 Median : -6.110   
## Mean : -0.5397 Mean : -2.059 Mean : -2.018 Mean : -2.171   
## 3rd Qu.: 3.1850 3rd Qu.: 6.880 3rd Qu.: 11.512 3rd Qu.: 4.820   
## Max. : 20.9100 Max. : 21.740 Max. : 29.470 Max. :126.920   
## VZ\_dailyChange SIG\_dailyChange DayOfWeek Month   
## Min. :-20.6356 Min. :-55.5000 Length:31 Length:31   
## 1st Qu.: -3.7856 1st Qu.:-19.9650 Class :character Class :character   
## Median : -0.1100 Median : -1.1300 Mode :character Mode :character   
## Mean : -0.1209 Mean : -0.4039   
## 3rd Qu.: 4.5250 3rd Qu.: 17.5700   
## Max. : 35.3161 Max. : 50.7200   
## Year UE\_monthlyRate dayOfMonth DOW\_Daily\_Close  
## Min. :2007 Min. :3.600 Min. :1.000 Min. : 7762   
## 1st Qu.:2009 1st Qu.:4.450 1st Qu.:1.000 1st Qu.:11253   
## Median :2013 Median :5.100 Median :1.000 Median :13668   
## Mean :2013 Mean :6.142 Mean :1.581 Mean :16184   
## 3rd Qu.:2016 3rd Qu.:8.500 3rd Qu.:2.000 3rd Qu.:20192   
## Max. :2020 Max. :9.900 Max. :4.000 Max. :28869   
## DOW\_Daily\_Volume DOW\_Daily\_Change lastMonth\_UE\_rate  
## Min. : 74050000 Min. : 2.471 Min. :3.500   
## 1st Qu.:148865000 1st Qu.: 36.360 1st Qu.:4.350   
## Median :213700000 Median :114.949 Median :5.000   
## Mean :217756129 Mean :129.052 Mean :5.977   
## 3rd Qu.:307650000 3rd Qu.:200.670 3rd Qu.:8.300   
## Max. :388480000 Max. :348.580 Max. :9.800   
## increasingMonthly\_UE\_rate  
## Min. :1   
## 1st Qu.:1   
## Median :1   
## Mean :1   
## 3rd Qu.:1   
## Max. :1

Summary of the DOW down and unemployment down:

summary(dow\_down\_ue\_up)

## Date TGT\_dailyChange FTR\_dailyChange UBSI\_dailyChange   
## Min. :2007-12-03 Min. :-47.520 Min. :-72.150 Min. :-16.220   
## 1st Qu.:2008-10-17 1st Qu.:-20.555 1st Qu.:-53.700 1st Qu.: -5.790   
## Median :2009-09-01 Median : -7.550 Median : -5.250 Median : -0.850   
## Mean :2011-04-15 Mean : -8.380 Mean :-11.453 Mean : -3.033   
## 3rd Qu.:2013-09-02 3rd Qu.: -1.405 3rd Qu.: 2.175 3rd Qu.: 1.535   
## Max. :2019-10-01 Max. : 22.310 Max. :192.700 Max. : 4.450   
## HD\_dailyChange JPM\_dailyChange XOM\_dailyChange CVX\_dailyChange   
## Min. :-199.420 Min. :-64.650 Min. :-19.6700 Min. :-29.880   
## 1st Qu.: -7.195 1st Qu.: -7.430 1st Qu.:-10.5250 1st Qu.:-14.040   
## Median : 0.670 Median : -0.300 Median : 0.5000 Median : -5.660   
## Mean : -6.023 Mean : -5.855 Mean : -0.9374 Mean : -2.875   
## 3rd Qu.: 11.375 3rd Qu.: 4.255 3rd Qu.: 7.5250 3rd Qu.: 8.705   
## Max. : 55.200 Max. : 7.560 Max. : 24.9600 Max. : 34.630   
## NSANY\_dailyChange MGM\_dailyChange TEVA\_dailyChange HST\_dailyChange   
## Min. :-15.640 Min. :-73.43 Min. :-13.220 Min. :-12.247   
## 1st Qu.: -7.670 1st Qu.:-53.83 1st Qu.: -3.925 1st Qu.: -9.090   
## Median : -3.450 Median : -5.75 Median : 2.760 Median : -2.130   
## Mean : -3.421 Mean :-18.56 Mean : 3.083 Mean : -3.406   
## 3rd Qu.: 1.000 3rd Qu.: 3.68 3rd Qu.: 6.490 3rd Qu.: 1.055   
## Max. : 10.350 Max. : 56.48 Max. : 38.180 Max. : 4.810   
## WFC\_dailyChange WWE\_dailyChange INO\_dailyChange SCE.PB\_dailyChange  
## Min. :-18.260 Min. :-55.790 Min. :-8.8000 Min. :-879.0   
## 1st Qu.: -7.755 1st Qu.: -3.205 1st Qu.:-2.6800 1st Qu.:-608.5   
## Median : -5.520 Median : -1.320 Median :-1.0000 Median : -58.0   
## Mean : -2.947 Mean : -3.239 Mean :-0.1611 Mean :-103.1   
## 3rd Qu.: 3.100 3rd Qu.: 0.610 3rd Qu.: 3.0900 3rd Qu.: 285.0   
## Max. : 9.350 Max. : 15.740 Max. : 7.2000 Max. : 729.0   
## FFIN\_dailyChange GOOG\_dailyChange WM\_dailyChange ONCY\_dailyChange   
## Min. :-24.2067 Min. :-848.61 Min. :-84.980 Min. :-19.950   
## 1st Qu.: -0.1592 1st Qu.: -60.75 1st Qu.: -3.010 1st Qu.: -8.645   
## Median : 1.0133 Median : 25.88 Median : -1.300 Median : -5.700   
## Mean : -0.1642 Mean : -35.57 Mean : -2.921 Mean : -1.166   
## 3rd Qu.: 1.8629 3rd Qu.: 79.80 3rd Qu.: 3.305 3rd Qu.: 2.708   
## Max. : 3.6650 Max. : 165.49 Max. : 25.400 Max. : 34.010   
## S\_dailyChange F\_dailyChange ARWR\_dailyChange COST\_dailyChange   
## Min. :-11.980 Min. :-5.380 Min. :-31.200 Min. :-227.820   
## 1st Qu.:-10.695 1st Qu.:-3.500 1st Qu.:-25.150 1st Qu.: -14.105   
## Median : -2.190 Median :-1.710 Median : -5.500 Median : 5.930   
## Mean : -3.012 Mean :-1.140 Mean : -9.518 Mean : -6.886   
## 3rd Qu.: 1.510 3rd Qu.: 1.135 3rd Qu.: -0.890 3rd Qu.: 14.720   
## Max. : 9.050 Max. : 4.780 Max. : 15.420 Max. : 55.300   
## AAL\_dailyChange JWN\_dailyChange NUS\_dailyChange HMC\_dailyChange   
## Min. :-33.490 Min. :-32.290 Min. :-75.570 Min. :-13.200   
## 1st Qu.: -9.750 1st Qu.:-19.460 1st Qu.: -8.070 1st Qu.: -6.905   
## Median : -6.380 Median :-12.070 Median : -1.180 Median : -2.430   
## Mean : -4.933 Mean : -8.785 Mean : -3.667 Mean : -2.399   
## 3rd Qu.: 3.420 3rd Qu.: 3.345 3rd Qu.: 3.035 3rd Qu.: 1.020   
## Max. : 20.530 Max. : 12.880 Max. : 41.580 Max. : 9.830   
## AMZN\_dailyChange T\_dailyChange HRB\_dailyChange RRGB\_dailyChange   
## Min. :-1685.38 Min. :-15.260 Min. :-10.1000 Min. :-29.840   
## 1st Qu.: -14.44 1st Qu.: -9.550 1st Qu.: -2.5000 1st Qu.:-21.235   
## Median : 11.82 Median : -2.150 Median : -0.0300 Median : -7.640   
## Mean : -58.88 Mean : -2.689 Mean : 0.1426 Mean : -6.002   
## 3rd Qu.: 57.96 3rd Qu.: 3.460 3rd Qu.: 2.2100 3rd Qu.: 4.840   
## Max. : 254.85 Max. : 9.980 Max. : 14.5700 Max. : 20.720   
## ADDYY\_dailyChange PCG\_dailyChange ROST\_dailyChange JNJ\_dailyChange   
## Min. :-114.700 Min. :-36.9100 Min. :-99.39750 Min. :-60.6500   
## 1st Qu.: -14.550 1st Qu.: -6.8350 1st Qu.: 0.06875 1st Qu.: -7.9600   
## Median : 0.950 Median : 0.5500 Median : 2.10500 Median : 2.6200   
## Mean : -4.267 Mean : -0.7774 Mean : -2.43053 Mean : -0.2811   
## 3rd Qu.: 9.335 3rd Qu.: 2.4800 3rd Qu.: 3.72000 3rd Qu.: 9.0450   
## Max. : 45.600 Max. : 36.1400 Max. : 12.96000 Max. : 31.4200   
## NFLX\_dailyChange M\_dailyChange KSS\_dailyChange DLTR\_dailyChange   
## Min. :-290.35286 Min. :-23.290 Min. :-31.300 Min. :-91.7233   
## 1st Qu.: -0.00286 1st Qu.:-17.480 1st Qu.:-16.285 1st Qu.: -1.3917   
## Median : 1.10714 Median :-13.550 Median : -7.260 Median : 3.6000   
## Mean : -12.92707 Mean : -6.327 Mean : -7.433 Mean : -0.2938   
## 3rd Qu.: 3.03357 3rd Qu.: 5.530 3rd Qu.: 2.655 3rd Qu.: 4.8650   
## Max. : 30.90000 Max. : 14.980 Max. : 20.100 Max. : 28.9500   
## WMT\_dailyChange C\_dailyChange AAP\_dailyChange JBLU\_dailyChange   
## Min. :-66.390 Min. :-298.30 Min. :-102.160 Min. :-10.390   
## 1st Qu.: -3.830 1st Qu.:-201.40 1st Qu.: -5.170 1st Qu.: -4.100   
## Median : 5.480 Median : -91.20 Median : 2.260 Median : -0.200   
## Mean : 1.438 Mean : -97.17 Mean : 1.378 Mean : -1.680   
## 3rd Qu.: 10.335 3rd Qu.: 2.37 3rd Qu.: 7.135 3rd Qu.: 0.445   
## Max. : 17.570 Max. : 266.25 Max. : 46.750 Max. : 2.850   
## MSFT\_dailyChange KGJI\_dailyChange EPD\_dailyChange TJX\_dailyChange   
## Min. :-104.940 Min. :-3.7200 Min. :-12.7250 Min. :-47.8125   
## 1st Qu.: -9.850 1st Qu.:-0.6300 1st Qu.: -3.2000 1st Qu.: -0.2062   
## Median : -1.670 Median :-0.3400 Median : -0.0800 Median : 1.0200   
## Mean : -5.594 Mean :-0.4168 Mean : -0.3142 Mean : -1.1578   
## 3rd Qu.: 3.825 3rd Qu.:-0.1200 3rd Qu.: 2.5650 3rd Qu.: 2.5537   
## Max. : 30.260 Max. : 1.4100 Max. : 6.8200 Max. : 5.8875   
## HOFT\_dailyChange LUV\_dailyChange NKE\_dailyChange TM\_dailyChange   
## Min. :-15.410 Min. :-38.580 Min. :-67.750 Min. :-47.61   
## 1st Qu.: -5.810 1st Qu.: -3.530 1st Qu.: -2.794 1st Qu.:-30.57   
## Median : -1.770 Median : -0.010 Median : 0.765 Median : -7.82   
## Mean : -3.428 Mean : -1.265 Mean : -2.084 Mean :-12.20   
## 3rd Qu.: 0.075 3rd Qu.: 2.575 3rd Qu.: 2.931 3rd Qu.: 5.45   
## Max. : 4.060 Max. : 16.720 Max. : 17.240 Max. : 19.53   
## VZ\_dailyChange SIG\_dailyChange DayOfWeek Month   
## Min. :-17.933 Min. :-51.310 Length:19 Length:19   
## 1st Qu.: -7.713 1st Qu.:-18.580 Class :character Class :character   
## Median : -1.522 Median :-15.490 Mode :character Mode :character   
## Mean : -2.523 Mean : -3.959   
## 3rd Qu.: 1.765 3rd Qu.: 15.620   
## Max. : 9.990 Max. : 42.810   
## Year UE\_monthlyRate dayOfMonth DOW\_Daily\_Close  
## Min. :2007 Min. : 3.6 Min. :1.000 Min. : 6763   
## 1st Qu.:2008 1st Qu.: 5.7 1st Qu.:1.000 1st Qu.: 9415   
## Median :2009 Median : 6.7 Median :1.000 Median :12290   
## Mean :2011 Mean : 6.9 Mean :1.789 Mean :12912   
## 3rd Qu.:2014 3rd Qu.: 8.1 3rd Qu.:3.000 3rd Qu.:14973   
## Max. :2019 Max. :10.0 Max. :3.000 Max. :26573   
## DOW\_Daily\_Volume DOW\_Daily\_Change lastMonth\_UE\_rate  
## Min. : 75630000 Min. :-679.95 Min. :3.500   
## 1st Qu.:131975000 1st Qu.:-241.33 1st Qu.:5.550   
## Median :199090000 Median : -59.98 Median :6.500   
## Mean :213377895 Mean :-147.22 Mean :6.668   
## 3rd Qu.:263550000 3rd Qu.: -23.12 3rd Qu.:7.750   
## Max. :568670000 Max. : -5.18 Max. :9.800   
## increasingMonthly\_UE\_rate  
## Min. :1   
## 1st Qu.:1   
## Median :1   
## Mean :1   
## 3rd Qu.:1   
## Max. :1

From the above subset of stock daily changes during a time of increasing monthly unemployment rate and decreasing DOW daily value, there are only three stocks that all had increasing daily median and mean values for those time periods: TEVA, WMT, and AAP. There are some stocks that only had median increasing values: HD, XOM, FFIN, GOOG, COST, AMZN, ADDY, PCG, ROST, JNJ, NFLX, DLTR, TJX, and NKE. One stock only had an increasing daily change mean value but not median value: HRB.

Lets look at these stocks that increased during decreasing public outlook on economy assumed from decreasing DOW value (losses in investments/future/retirement) and increasing unemployment (more people not working) from month before.

stocksGood <- subset(stockNames, stockNames$stock == 'TEVA' |  
 stockNames$stock == 'WMT'|  
 stockNames$stock == 'AAP'|  
 stockNames$stock == 'HD'|  
 stockNames$stock == 'XOM'|  
 stockNames$stock == 'FFIN'|  
 stockNames$stock == 'GOOG'|  
 stockNames$stock == 'COST'|  
 stockNames$stock == 'AMZN'|  
 stockNames$stock == 'ADDY'|  
 stockNames$stock == 'PCG'|  
 stockNames$stock == 'ROST'|  
 stockNames$stock == 'JNJ'|  
 stockNames$stock == 'NFLX'|  
 stockNames$stock == 'DLTR'|  
 stockNames$stock == 'TJX'|  
 stockNames$stock == 'NKE'|  
 stockNames$stock == 'HRB')  
stocksGood$stockInfo

## [1] The Home Depot, Inc. (HD)\nNYSE - NYSE Delayed Price. Currency in USD   
## [2] Exxon Mobil Corporation (XOM)\nNYSE - NYSE Delayed Price. Currency in USD   
## [3] Teva Pharmaceutical Industries Limited (TEVA)\nNYSE - NYSE Delayed Price. Currency in USD   
## [4] First Financial Bankshares, Inc. (FFIN)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## [5] Alphabet Inc. (GOOG)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [6] Costco Wholesale Corporation (COST)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [7] PG&E Corporation (PCG)\nNYSE - NYSE Delayed Price. Currency in USD   
## [8] Dollar Tree, Inc. (DLTR)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [9] NIKE, Inc. (NKE)\nNYSE - NYSE Delayed Price. Currency in USD   
## [10] Amazon.com, Inc. (AMZN)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [11] Ross Stores, Inc. (ROST)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [12] Walmart Inc. (WMT)\nNYSE - NYSE Delayed Price. Currency in USD   
## [13] The TJX Companies, Inc. (TJX)\nNYSE - NYSE Delayed Price. Currency in USD   
## [14] Johnson & Johnson (JNJ)\nNYSE - NYSE Delayed Price. Currency in USD   
## [15] H&R Block, Inc. (HRB)\nNYSE - NYSE Delayed Price. Currency in USD   
## [16] Netflix, Inc. (NFLX)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD   
## [17] Advance Auto Parts, Inc. (AAP)\nNYSE - NYSE Delayed Price. Currency in USD   
## 65 Levels: adidas AG (ADDYY)\nOther OTC - Other OTC Delayed Price. Currency in USD ...

From the above, the stocks of auto parts, cheap department and goods, health and beauty products, Nike sports shoes for people wanting to workout and not spend money to occupy time or to predict an increase in low crime robberies (assumptions made by real person not AI), Google for job searches, Amazon because ever expanding and employing many workers, costco for middle class workers and families, Ross and TJ Maxx for low cost business/dress attire and goods, electric company, fuel, home improvement/repair stores, and low cost movie entertainment at home.

Split the summaries of each table to show those that have mean positive values.

S <- as.data.frame(summary(dow\_up\_ue\_up))  
S1 <- as.data.frame(summary(dow\_down\_ue\_up))  
S <- S[-(1:6),-1]  
S1 <- S1[-(1:6),-1]  
  
S$Freq <- as.character(S$Freq)  
S1$Freq <- as.character(S1$Freq)  
  
s\_a <- strsplit(S$Freq, ':')  
s\_b <- strsplit(S1$Freq, ':')  
  
S$Stat <- lapply(s\_a, '[',1)  
S1$Stat <- lapply(s\_b, '[',1)  
  
S$Stat <- as.vector(S$Stat)  
  
S$StatValue <- as.numeric(lapply(s\_a, '[',2))  
S1$StatValue <- as.numeric(lapply(s\_b, '[',2))  
  
S\_mean <- S[grep('Mean', S$Stat),]  
S1\_mean <- S1[grep('Mean', S1$Stat),]  
  
Dow\_up\_ue\_up\_meanPos <- subset(S\_mean, S\_mean$StatValue >= 0)  
Dow\_down\_ue\_up\_meanPos <- subset(S1\_mean, S1\_mean$StatValue >= 0)  
  
Dow\_up\_ue\_up\_meanPos <- Dow\_up\_ue\_up\_meanPos[grep('dailyChange', Dow\_up\_ue\_up\_meanPos$Var2),]  
Dow\_down\_ue\_up\_meanPos <- Dow\_down\_ue\_up\_meanPos[grep('dailyChange',  
 Dow\_down\_ue\_up\_meanPos$Var2),]  
colnames(Dow\_up\_ue\_up\_meanPos)[1] <- 'DOW\_up\_UE\_up'  
colnames(Dow\_down\_ue\_up\_meanPos)[1] <- 'DOW\_down\_UE\_down'  
  
S\_Median <- S[grep('Median', S$Stat),]  
S1\_Median <- S1[grep('Median', S1$Stat),]  
  
Dow\_up\_ue\_up\_MedianPos <- subset(S\_Median, S\_Median$StatValue >= 0)  
Dow\_down\_ue\_up\_MedianPos <- subset(S1\_Median, S1\_Median$StatValue >= 0)  
  
Dow\_up\_ue\_up\_MedianPos <- Dow\_up\_ue\_up\_MedianPos[grep('dailyChange', Dow\_up\_ue\_up\_MedianPos$Var2),]  
Dow\_down\_ue\_up\_MedianPos <- Dow\_down\_ue\_up\_MedianPos[grep('dailyChange',  
 Dow\_down\_ue\_up\_MedianPos$Var2),]  
colnames(Dow\_up\_ue\_up\_MedianPos)[1] <- 'DOW\_up\_UE\_up'  
colnames(Dow\_down\_ue\_up\_MedianPos)[1] <- 'DOW\_down\_UE\_down'

Write those tables to csv to use as needed. We should test how well the same amount invested in the original 53 stocks over the span from 2007-2020 did to the same amount of money using different weights on those stocks whose value of daily changes was positive for the median and mean values separately when the DOW was up and unemployment was up and also when the DOW was down and unemployment was up. The Stat column is a list and won’t print to csv without removing it, and the Freq column also has the statistic being evaluated.

write.csv(Dow\_up\_ue\_up\_meanPos[,-3], 'Dow\_up\_ue\_up\_meanPos.csv', row.names=FALSE)  
write.csv(Dow\_down\_ue\_up\_meanPos[,-3],'Dow\_down\_ue\_up\_meanPos.csv', row.names=FALSE)  
write.csv(Dow\_up\_ue\_up\_MedianPos[,-3],'Dow\_up\_ue\_up\_MedianPos.csv', row.names=FALSE)  
write.csv(Dow\_down\_ue\_up\_MedianPos[,-3],'Dow\_down\_ue\_up\_MedianPos.csv', row.names=FALSE)

Now, will make a vector of those stocks that have positive medians and means when the DOW is up or down when unemployment is up.

Dow\_up\_med <- as.data.frame(Dow\_up\_ue\_up\_MedianPos$DOW\_up\_UE\_up)  
DOW\_up\_mean <- as.data.frame(Dow\_up\_ue\_up\_meanPos$DOW\_up\_UE\_up)  
DOW\_down\_med <- as.data.frame(Dow\_down\_ue\_up\_MedianPos$DOW\_down\_UE\_down)  
DOW\_down\_mean <- as.data.frame(Dow\_down\_ue\_up\_meanPos$DOW\_down\_UE\_down)  
  
colnames(Dow\_up\_med) <- 'Dow\_up\_median'  
colnames(DOW\_up\_mean) <- 'DOW\_up\_mean'  
colnames(DOW\_down\_med) <- 'DOW\_down\_median'  
colnames(DOW\_down\_mean) <- 'DOW\_down\_mean'  
  
DOW\_up\_mean$DOW\_up\_mean <- gsub('\_dailyChange','', DOW\_up\_mean$DOW\_up\_mean)  
DOW\_down\_mean$DOW\_down\_mean <- gsub('\_dailyChange', '', DOW\_down\_mean$DOW\_down\_mean)  
Dow\_up\_med$Dow\_up\_median <- gsub('\_dailyChange', '', Dow\_up\_med$Dow\_up\_median)  
DOW\_down\_med$DOW\_down\_median <- gsub('\_dailyChange', '', DOW\_down\_med$DOW\_down\_median)

Lets add the values to these subsets of all 53 original stocks.

StockValues <- Close2  
StockValues <- StockValues[,-c(1,55:58,60:63)]  
  
colnames(StockValues) <- gsub('.Close','', colnames(StockValues))  
StockValues$total <- rowSums(StockValues[1:53])  
  
portfolio53 <- StockValues[order(StockValues$Date,decreasing=FALSE),]  
  
portfolio53 <- portfolio53[c(1,3303),]  
portfolio53

## TGT FTR UBSI HD JPM XOM CVX NSANY MGM TEVA  
## 2007-01-03 57.18 215.40 39.05 41.07 48.07 74.11 70.97 24.16 57.57 31.26  
## 2020-02-14 116.63 0.57 34.36 245.03 137.46 60.65 110.08 9.46 31.52 12.22  
## HST WFC WWE INO SCE.PB FFIN GOOG WM ONCY  
## 2007-01-03 24.04307 35.74 16.18 13.16 280 6.996666 232.922 37.03 21.945  
## 2020-02-14 16.91000 48.22 44.93 4.15 893 34.580002 1520.740 125.75 2.590  
## S F ARWR COST AAL JWN NUS ADDYY KSS MSFT LUV  
## 2007-01-03 19.04 7.51 44.00 52.84 56.3 51.39 18.54 25.00 67.08 29.86 15.52  
## 2020-02-14 8.69 8.10 41.27 318.31 29.2 40.28 30.45 156.11 44.47 185.35 57.97  
## HMC PCG DLTR KGJI NKE AMZN ROST WMT TJX  
## 2007-01-03 39.71 47.27 10.23 1.10 12.20875 38.70 7.6225 47.55 7.1675  
## 2020-02-14 26.78 16.20 88.68 1.19 103.54000 2134.87 121.7800 117.89 63.3800  
## TM T JNJ C EPD VZ HRB NFLX AAP  
## 2007-01-03 135.30 34.95 66.40 552.50 14.485 35.30673 23.20 3.801429 35.58  
## 2020-02-14 140.15 38.25 150.13 78.79 26.270 58.51000 22.38 380.399994 133.59  
## HOFT SIG RRGB M JBLU Date total  
## 2007-01-03 15.47 47.74 35.00 37.51 15.20 2007-01-03 2977.939  
## 2020-02-14 22.25 26.07 35.21 16.67 21.27 2020-02-14 8193.300

profit\_all <- 8193-2978  
profit\_all

## [1] 5215

With all 53 stocks from January 1, 2007 throughout February 14, 2020, the portfoliio initially cost 2978 USD and was valued at 8193 USD at the end of that time span. Lets see how much the portfolio is worth when using only the stocks in our subsets of stocks that had positive values when the DOW was up or down but unemployment was increasing. The profit earned was 5215 USD with this portfolio.

p53 <- gather(StockValues, 'stock','stockValue', 1:53)

The positive median value stock when the DOW was up and unemployment was up.

p1 <- merge(Dow\_up\_med,p53, by.x='Dow\_up\_median','stock')  
P1 <- p1 %>% group\_by(Date) %>% summarise\_at(vars(stockValue), sum)  
P1[c(1,3303),]

## # A tibble: 2 x 2  
## Date stockValue  
## <date> <dbl>  
## 1 2007-01-03 1294.  
## 2 2020-02-14 7189.

The initial stock value for the portfolio of stock that had a positive median value when the DOW was up and unemployment was up started at 1294 USD and ended with a value of 7189 USD. Lets weight this portfolio so that we can see the profits in dollars if the initial investment was the same amount as the investment of all 53 stocks.

P1\_i <- P1$stockValue[1]  
P1\_l <- P1$stockValue[3303]  
P1\_i

## [1] 1294.482

P1\_l

## [1] 7189.25

profit1 <- P1\_l-P1\_i  
profit1

## [1] 5894.768

r1 <- P1\_l/P1\_i  
r1

## [1] 5.553767

p53i <- portfolio53$total[1]  
p53i

## [1] 2977.939

finalValue\_P1 <- p53i\*r1  
finalValue\_P1

## [1] 16538.78

total\_P1\_profit <- finalValue\_P1 - p53i  
total\_P1\_profit

## [1] 13560.84

unique(p1$Dow\_up\_median)

## [1] "AAL" "AAP" "ADDYY" "AMZN" "COST" "CVX" "DLTR" "FFIN"   
## [9] "GOOG" "HD" "HST" "JNJ" "JPM" "M" "MSFT" "NFLX"   
## [17] "NKE" "NSANY" "PCG" "ROST" "RRGB" "SCE.PB" "TJX" "WM"   
## [25] "WMT" "WWE"

From the above values, the initial investment was 1294 USD, and the final value of these stock were 7189 USD in our time series. The profit in dollars earned was 5895 USD. The ratio of final value to initial value was 5.55. The total profit if the same investment amount made as with all 53 stocks was made on these stocks (2978 USD) that had a median positive value when the DOW was up and unemployment was increasing took the initial amount times the ratio of final/initial value added to the difference in the initial price invested in all stocks times the ratio of final/initial stock in this portfolio of 26 stocks. The final value of this portfolio is 16539 USD with profits earned of 13561 USD. \*\*\*

The positive median value stock when the DOW was down and unemployment was up.

p2 <- merge(DOW\_down\_med,p53, by.x='DOW\_down\_median','stock')  
P2 <- p2 %>% group\_by(Date) %>% summarise\_at(vars(stockValue), sum)  
P2[c(1,3303),]

## # A tibble: 2 x 2  
## Date stockValue  
## <date> <dbl>  
## 1 2007-01-03 741.  
## 2 2020-02-14 5658.

P2\_i <- P2$stockValue[1]  
P2\_l <- P2$stockValue[3303]  
P2\_i

## [1] 740.7288

P2\_l

## [1] 5658.1

profit2 <- P2\_l-P2\_i  
profit2

## [1] 4917.371

r2 <- P2\_l/P2\_i  
r2

## [1] 7.638558

p53i <- portfolio53$total[1]  
p53i

## [1] 2977.939

total\_P2\_Value <- p53i\*r2   
total\_P2\_Value

## [1] 22747.16

total\_P2\_profit <- total\_P2\_Value - p53i  
total\_P2\_profit

## [1] 19769.22

unique(p2$DOW\_down\_median)

## [1] "AAP" "ADDYY" "AMZN" "COST" "DLTR" "FFIN" "GOOG" "HD" "JNJ"   
## [10] "NFLX" "NKE" "PCG" "ROST" "TEVA" "TJX" "WMT" "XOM"

The portfolio of stock that had positive median values of daily change when the DOW was down and unemployment was higher than the month before is shown above. There are 17 stocks in this portfolio. The initial value was 741 USD with a final value of 5658 USD and a profit of 4917 USD earned as is. If the same amount was invested in just these stocks as was invested in the entire portfolio of 53 stocks of 2978 USD, then the ratio of final/initial value would be used on adding additional stocks in this portfolio at a ratio of 7.64. The total end value would be 22747 USD with profits earned of 19769 USD.

The positive mean value stock when the DOW was up and unemployment was up.

p3 <- merge(DOW\_down\_mean,p53, by.x='DOW\_down\_mean','stock')  
P3 <- p3 %>% group\_by(Date) %>% summarise\_at(vars(stockValue), sum)  
P3[c(1,3303),]

## # A tibble: 2 x 2  
## Date stockValue  
## <date> <dbl>  
## 1 2007-01-03 138.  
## 2 2020-02-14 286.

P3\_i <- P3$stockValue[1]  
P3\_l <- P3$stockValue[3303]  
P3\_i

## [1] 137.59

P3\_l

## [1] 286.08

profit3 <- P3\_l-P3\_i  
profit3

## [1] 148.49

r3 <- P3\_l/P3\_i  
r3

## [1] 2.079221

p53i <- portfolio53$total[1]  
p53i

## [1] 2977.939

total\_P3\_Value <- (p53i)\*r3   
total\_P3\_Value

## [1] 6191.792

total\_P3\_profit <- total\_P3\_Value - p53i  
total\_P3\_profit

## [1] 3213.853

unique(p3$DOW\_down\_mean)

## [1] "AAP" "HRB" "TEVA" "WMT"

The above portfolio shows those stocks who had a positive mean value of daily changes when the DOW was down and unemployment was increased more than the month before. There are only four stock in this portfolio. The initial value was 138 USD and the final value was 286 USD. The profits in dollars was 149 USD as is. The ratio of final/initial value is 2.08. When investing the same amount of 2978 USD as was used in the portfolio of 53 stock, the dollars earned were 6192 USD, with profit earned in dollars of 3213 USD.

The positive mean value stock when the DOW was down and unemployment was up.

p4 <- merge(DOW\_up\_mean,p53, by.x='DOW\_up\_mean','stock')  
P4 <- p4 %>% group\_by(Date) %>% summarise\_at(vars(stockValue), sum)  
P4[c(1,3303),]

## # A tibble: 2 x 2  
## Date stockValue  
## <date> <dbl>  
## 1 2007-01-03 1588.  
## 2 2020-02-14 1476.

P4\_i <- P4$stockValue[1]  
P4\_l <- P4$stockValue[3303]  
P4\_i

## [1] 1588.048

P4\_l

## [1] 1476.17

profit4 <- P4\_l-P4\_i  
profit4

## [1] -111.8781

r4 <- P4\_l/P4\_i  
r4

## [1] 0.9295499

p53i <- portfolio53$total[1]  
p53i

## [1] 2977.939

total\_P4\_Value <- p53i\*r4   
total\_P4\_Value

## [1] 2768.143

total\_P4\_profit <- total\_P4\_Value - p53i  
total\_P4\_profit

## [1] -209.7959

unique(p4$DOW\_up\_mean)

## [1] "AAP" "ARWR" "C" "F" "FTR" "HMC" "HST" "INO"   
## [9] "KGJI" "KSS" "NSANY" "ONCY" "PCG" "S" "SCE.PB" "T"   
## [17] "TEVA" "UBSI" "WWE" "XOM"

The above shows those stock in the portfolio that had a positive mean daily change when the DOW was up and unemployment was up. There are 20 stocks in this portfolio. The initial value of this portfolio was 1588 USD and the final value was less at 1476 USD. The loss was 112 USD with a final/initial ratio of 0.93. When investing the same amount as the initial portfolio of 2978 USD, the final portfolio value is a loss of 210 USD.

Lets make a data table of this information.

du1 <- as.data.frame( c(P1\_i, P2\_i,P3\_i,P4\_i))  
du2 <- as.data.frame( c(P1\_l,P2\_l,P3\_l,P4\_l))   
du3 <- as.data.frame(c(length(unique(p1$Dow\_up\_median)),length(unique(p2$DOW\_down\_median)),  
 length(unique(p3$DOW\_down\_mean)), length(unique(p4$DOW\_up\_mean))))  
du4 <- as.data.frame( c(profit1, profit2, profit3, profit4))  
  
colnames(du1) <- 'initialValue'  
colnames(du2) <- 'finalValue'  
colnames(du3) <- 'numberStocksInPortfolio'  
colnames(du4) <- 'profitInitialValue'  
  
du5 <- as.data.frame(c(p53i,p53i,p53i,p53i))  
colnames(du5) <- 'ifInitialInvestmentAsAll53Made'  
  
du6 <- as.data.frame(c(finalValue\_P1, total\_P2\_Value,total\_P3\_Value,total\_P4\_Value))  
colnames(du6) <- 'finalValueIfSame53StockInvestment'  
  
du7 <- as.data.frame(c(total\_P1\_profit, total\_P2\_profit, total\_P3\_profit, total\_P4\_profit))  
colnames(du7) <- 'totalProfitSame53StockInvestment'  
  
du8 <- as.data.frame(c(r1,r2,r3,r4))  
colnames(du8) <- 'ratioFinal\_2\_Initial'  
  
DOW\_UE <- cbind(du1,du2,du8,du4,du3,du5,du6,du7)  
row.names(DOW\_UE) <- c('Dow\_up\_median','DOW\_down\_median','DOW\_down\_mean','DOW\_up\_mean')  
  
write.csv(DOW\_UE, 'DOW\_UE.csv', row.names=TRUE)  
  
DOW\_UE

## initialValue finalValue ratioFinal\_2\_Initial profitInitialValue  
## Dow\_up\_median 1294.4819 7189.25 5.5537665 5894.7682  
## DOW\_down\_median 740.7288 5658.10 7.6385583 4917.3713  
## DOW\_down\_mean 137.5900 286.08 2.0792208 148.4900  
## DOW\_up\_mean 1588.0481 1476.17 0.9295499 -111.8781  
## numberStocksInPortfolio ifInitialInvestmentAsAll53Made  
## Dow\_up\_median 26 2977.939  
## DOW\_down\_median 17 2977.939  
## DOW\_down\_mean 4 2977.939  
## DOW\_up\_mean 20 2977.939  
## finalValueIfSame53StockInvestment  
## Dow\_up\_median 16538.776  
## DOW\_down\_median 22747.158  
## DOW\_down\_mean 6191.792  
## DOW\_up\_mean 2768.143  
## totalProfitSame53StockInvestment  
## Dow\_up\_median 13560.8372  
## DOW\_down\_median 19769.2190  
## DOW\_down\_mean 3213.8533  
## DOW\_up\_mean -209.7959

In summary of evaluating the stocks that had positive mean and median values when the unemployment rate was more than the previous month, but the DOW was either increasing or decreasing on that day from the previous day, the best portfolio of stocks was the one with the highest profit. The return on investment ratio was 7.63 for this portfolio, with an initial investment of 741 USD it returned 5658 USD, but when the same investment amount was distributed to this portfolio as the entire portfolio of 53 stocks of 2978 USD, the profits made were 19769 USD from 2007 through 2020.

The portfolio of stock that performed the worst with a loss of 111 USD on an initial investment of 1588 USD and a final to initial value ratio of 0.93 was the portfolio of stock that had a positive daily change mean value when the DOW was up and unemployment rate up. When this portfolio had the 2978 USD invested in it as the original portfolio of 53 stocks it saw a loss of 210 USD from 2007-2020.

The portfolio of stocks all having a positive median value of daily changes did much better than the positive mean value stock portfolios when the DOW was up or down and unemployment rate increased from the previous month from 2007-2020.

The question arises when asked on how to distribute the remaining dollars of the initial investment of the original portfolio of all 53 stocks, when that value you want to invest is 2978 USD but the portfolio of single stocks have a set value of 138-1588 USD for the four portfolios.

We would use an even weighted distribution if we could buy partial stocks, but it is likely we will not be able to. If it is the case that we could distribute the weights of the remaining balance to buy partial stocks then you would take that remaining balance and divide by the number of stock in the portfolio. We could do that now to see how much each of the weights are in investment dollars of each stock.

DOW\_UE$EvenRemainingWeightsUSD <- (DOW\_UE$ifInitialInvestmentAsAll53Made-DOW\_UE$initialValue)/  
 (DOW\_UE$numberStocksInPortfolio)  
DOW\_UE$EvenWeightsUSD <- (DOW\_UE$ifInitialInvestmentAsAll53Made)/(DOW\_UE$numberStocksInPortfolio)  
DOW\_UE[,c(1,5,9,10)]

## initialValue numberStocksInPortfolio EvenRemainingWeightsUSD  
## Dow\_up\_median 1294.4819 26 64.74834  
## DOW\_down\_median 740.7288 17 131.60058  
## DOW\_down\_mean 137.5900 4 710.08715  
## DOW\_up\_mean 1588.0481 20 69.49453  
## EvenWeightsUSD  
## Dow\_up\_median 114.5361  
## DOW\_down\_median 175.1729  
## DOW\_down\_mean 744.4847  
## DOW\_up\_mean 148.8969

The above table shows even weights after one stock of each is bought and the remaining money from 2978 USD is dispursed equally to each stock in the portfolio in the ‘EvenRemainingWeightsUSD’ column. The value of the even weights on the amount of dollars to invest in each stock from the total 2978 USD is in the ‘EvenWeightUSD’ column.

If you are not allowed to buy partial stocks, then you would have to rank the stocks in each portfolio so that more money is spent on the forecasted higher yielding stock.

So, we found a subset of stock in the portfolio that did outstanding, and we want to buy those stocks to make a profit, but we also want to look at the characteristics of those stocks and see what features they have or properties in the data that could make any other stock fit a description of a ‘good stock to buy’ category. Some features that come to mind are, are they all increasing, are they cyclical, how many local maxima and local minima each of these stocks have, what the sentiment in the internet search engines provide for these stocks, do they market, are they politically motivated such as Nike with the footbal player protesting police abuse of black males, are they part of larger business mergers such as talks of Tmobile getting bought out by Sprint, or how Frontier bought a portion of Verizon, and so on.

Also, we want to look at this as a careless surfer looking for intervals of small waves to buy close to the local minima on these stocks, ride it out and sell it close to its local maxima to simulate how exploiting the stocks in the short run can lead to more profits. We could all do this and just like this line of code is in 1920s, a depression could follow if we all did this. Like a huge crash. But we’re all blind, arrogant data scientists in charge of our own way of thinking and we want to see what happens. So lets do it. Did I lose you on the analogy? Which one? Its ok, you’ll find yourself for the next part of this data exploration.

Lets look at our 17 stocks belonging to the best subset and see what qualities each stock has by first adding a feature column on the stock brand with an internet search and looking at the local minima and maxima of each stock.

set17 <- merge(DOW\_down\_med, stockNames, by.x='DOW\_down\_median', by.y='stock')  
set17

## DOW\_down\_median  
## 1 AAP  
## 2 ADDYY  
## 3 AMZN  
## 4 COST  
## 5 DLTR  
## 6 FFIN  
## 7 GOOG  
## 8 HD  
## 9 JNJ  
## 10 NFLX  
## 11 NKE  
## 12 PCG  
## 13 ROST  
## 14 TEVA  
## 15 TJX  
## 16 WMT  
## 17 XOM  
## stockInfo  
## 1 Advance Auto Parts, Inc. (AAP)\nNYSE - NYSE Delayed Price. Currency in USD  
## 2 adidas AG (ADDYY)\nOther OTC - Other OTC Delayed Price. Currency in USD  
## 3 Amazon.com, Inc. (AMZN)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 4 Costco Wholesale Corporation (COST)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 5 Dollar Tree, Inc. (DLTR)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 6 First Financial Bankshares, Inc. (FFIN)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 7 Alphabet Inc. (GOOG)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 8 The Home Depot, Inc. (HD)\nNYSE - NYSE Delayed Price. Currency in USD  
## 9 Johnson & Johnson (JNJ)\nNYSE - NYSE Delayed Price. Currency in USD  
## 10 Netflix, Inc. (NFLX)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 11 NIKE, Inc. (NKE)\nNYSE - NYSE Delayed Price. Currency in USD  
## 12 PG&E Corporation (PCG)\nNYSE - NYSE Delayed Price. Currency in USD  
## 13 Ross Stores, Inc. (ROST)\nNasdaqGS - NasdaqGS Real Time Price. Currency in USD  
## 14 Teva Pharmaceutical Industries Limited (TEVA)\nNYSE - NYSE Delayed Price. Currency in USD  
## 15 The TJX Companies, Inc. (TJX)\nNYSE - NYSE Delayed Price. Currency in USD  
## 16 Walmart Inc. (WMT)\nNYSE - NYSE Delayed Price. Currency in USD  
## 17 Exxon Mobil Corporation (XOM)\nNYSE - NYSE Delayed Price. Currency in USD  
## stockExchange  
## 1 NYSE  
## 2 Other OTC  
## 3 Nasdaq  
## 4 Nasdaq  
## 5 Nasdaq  
## 6 Nasdaq  
## 7 Nasdaq  
## 8 NYSE  
## 9 NYSE  
## 10 Nasdaq  
## 11 NYSE  
## 12 NYSE  
## 13 Nasdaq  
## 14 NYSE  
## 15 NYSE  
## 16 NYSE  
## 17 NYSE

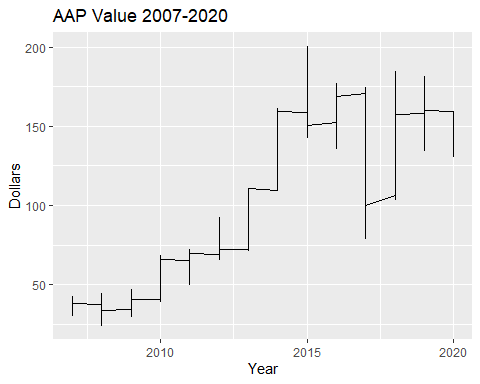
Lets search these companies and add in a feature that gives the number of results for each company.

set17$numberSearchReturnMillions <- round(c(0.446, 1.99,541, 0.780, 0.379, 0.0465, 3.51, 0.410, 45.1,   
 4.46, 4.28, 0.00417, 4.11, 0.00392, 2.59, 1.7, 1.14),4)

closing17 <- Close2[,-c(1,55:58,60)]  
colnames(closing17) <- gsub('.Close','',colnames(closing17))  
colnames(closing17) <- gsub('.PB','', colnames(closing17))  
close17 <- gather(closing17, 'stock','stockValue',1:53)  
Close17 <- merge(DOW\_down\_med, close17, by.x='DOW\_down\_median', by.y='stock')  
Close17 <- Close17[order(Close17$Date),]  
  
dow17 <- dow[,-c(2:4,6,7)]  
dow17$Date <- as.Date(dow17$Date)  
colnames(dow17)[2] <- 'DOW\_Close'  
Close17\_dow <- merge(Close17, dow17, by.x='Date', by.y='Date')

aap <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='AAP' )  
aap1 <- subset(aap, aap$Year==2017|  
 aap$Year==2018|  
 aap$Year==2019)  
  
ggplot(data = aap, aes(x=Year, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('AAP Value 2007-2020')+  
 ylab('Dollars')

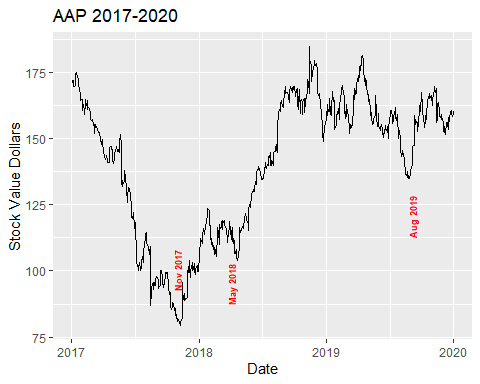
## Warning in pal\_name(palette, type): Unknown palette paired

 The above shows that AAP had a huge drop in 2017 at a local minimum, the other minimum is in 2008 and is the global minimum for this stock. The next chart shows the years 2017-2020 to zoom in on this loss.

annotation1 <- data.frame(  
 x = c(as.Date('2017-11-01'),as.Date('2018-04-04'),as.Date('2019-09-04')),  
 y = c(100,95,120),  
 label = c("Nov 2017", "May 2018","Aug 2019"))  
  
gg1 <- ggplot(data = aap1, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('AAP 2017-2020')+  
 geom\_text(data=annotation1, aes( x=x, y=y, label=label), ,   
 color="red",   
 size=2.5 , angle=90 , fontface="bold")+  
 ylab('Stock Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

gg1



The chart above shows that AAP had a decreasing year in 2017 down to its minimum value in the fourth quarter of the year, then increased throughout 2018 until 2019. There was also another local minima in the third quarter of 2019 for AAP before it began increasing. Something could have happened in the first quarter of 2017 to cause it to decrease and another thing in 2018. We could check the internet for articles around that time. Lets look at the summary stats for this table.

summary(aap1)

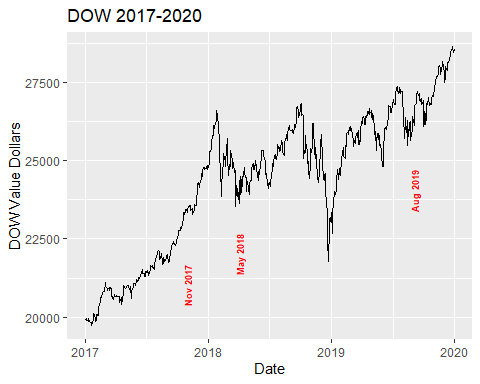
## Date DOW\_down\_median Month Year   
## Min. :2017-01-03 Length:754 Length:754 Min. :2017   
## 1st Qu.:2017-10-02 Class :character Class :character 1st Qu.:2017   
## Median :2018-07-02 Mode :character Mode :character Median :2018   
## Mean :2018-07-02 Mean :2018   
## 3rd Qu.:2019-04-02 3rd Qu.:2019   
## Max. :2019-12-31 Max. :2019   
## UE\_monthlyRate stockValue DOW\_Close   
## Min. :3.500 Min. : 79.38 Min. :19732   
## 1st Qu.:3.700 1st Qu.:116.19 1st Qu.:22421   
## Median :3.900 Median :149.94 Median :24879   
## Mean :3.965 Mean :140.15 Mean :24397   
## 3rd Qu.:4.200 3rd Qu.:161.47 3rd Qu.:26061   
## Max. :4.700 Max. :184.72 Max. :28645

The above summary statistics show a low unemployment rate for this period of time spanning three years from 2017 to 2020 with unemployment ranging from 3.5 to 4.7. The DOW had closing values ranging from 19732 USD to 28645 USD. Lets plot this date range for the DOW and see if they move together.

annotation2 <- data.frame(  
 x = c(as.Date('2017-11-01'),as.Date('2018-04-04'),as.Date('2019-09-04')),  
 y = c(21000,22000,24000),  
 label = c("Nov 2017", "May 2018","Aug 2019"))  
  
gg2 <- ggplot(data = aap1, aes(x=Date, y=DOW\_Close)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 ggtitle('DOW 2017-2020')+  
 ylab('DOW Value Dollars')+  
 geom\_text(data=annotation2, aes( x=x, y=y, label=label), ,   
 color="red",   
 size=2.5 , angle=90 , fontface="bold")

## Warning in pal\_name(palette, type): Unknown palette paired

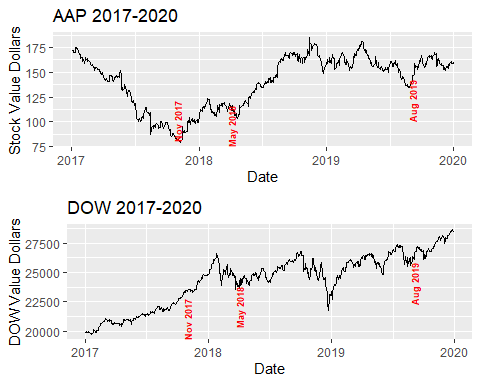
gg2



The above chart shows the increasing values of the DOW the same years as the AAP saw a decreasing year in 2017 to its local minimum in the fourth quarter of 2017, and then an increase until 2019 when it stabilized around the same value till 2020. But the DOW increased the time that AAP was decreasing and saw a local minima in the end of 2018 right when AAP reached a smaller local minima and also in the third quarter when AAP also had a local minima.

Lets look at these charts on top of each other.

grid.arrange(gg1, gg2, nrow = 2)



Lets see what happened in 2017 for AAP and in 2018 to cause it to decrease then increase respectively… just declining sales, plans to expand and open more stores, and public outcry on the sales declines of Advance Auto Parts. So, this could mean that because the DOW was doing great and increasing, investors thought to take their money out of the after market car parts stores, or maybe they thought they were hurting because of Amazon Prime taking their business. Those are some possibilities. When looking at stocks in the [DOW Jones industrial average](https://www.buyupside.com/sample_portfolios/djindustrialsstocks.php), AAP isn’t listed as one of these stocks, so it could be that people took their money out of the after markets car parts stock as it was declining and put it into any of the stocks that belong to the DOW, because it increased in value in 2017 while AAP decreased, then they both moved together around and after 2018.

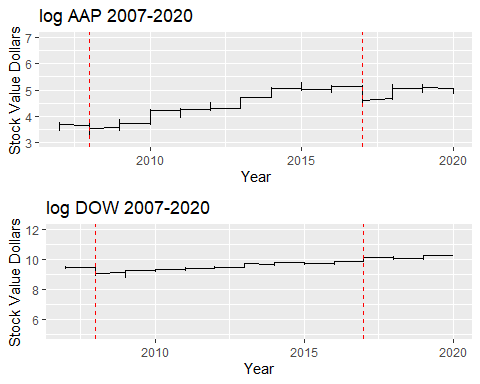
aaplog <- aap  
aaplog$logAAP <- log1p(aaplog$stockValue)  
aaplog$logDOW <- log1p(aaplog$DOW\_Close)  
  
g1 <- ggplot(data = aaplog, aes(x=Year, y=logAAP)) +  
 geom\_line()+  
 scale\_y\_continuous(limits=c(3,7))+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 geom\_vline(xintercept=c(2008,2017), linetype='dashed', color='red')+  
 ggtitle('log AAP 2007-2020')+  
 ylab('Stock Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

g2 <- ggplot(data = aaplog, aes(x=Year, y=logDOW)) +  
 geom\_line()+  
 scale\_y\_continuous(limits=c(5,12))+  
 scale\_fill\_brewer(palette="paired") +  
 theme(legend.position="bottom")+  
 geom\_vline(xintercept=c(2008,2017), linetype='dashed', color='red')+  
 ggtitle('log DOW 2007-2020')+  
 ylab('Stock Value Dollars')

## Warning in pal\_name(palette, type): Unknown palette paired

grid.arrange(g1, g2, nrow = 2)



The above chart shows the log scale of the value + 1 so that there aren’t any natural log errors in scaling. There is also an added few lines to show the years of 2008 and 2017 when AAP had decreasing years in stock value.

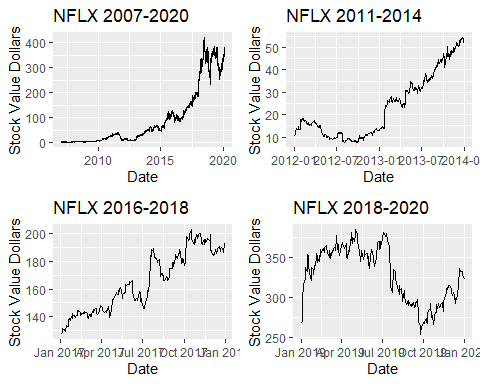
Lets look at the DOW over the years with the other 16 stocks in this portfolio of stocks that proved most profitable when the DOW was down and unemployment was up using the set of stocks with positive median values under those constraints.

dROI17 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(stockValue), mean)  
colnames(dROI17)[2] <- 'avgStockValue'  
  
start17 <- subset(Close17\_dow, Close17\_dow$Date=='2007-01-03')  
final17 <- subset(Close17\_dow, Close17\_dow$Date=='2020-02-14')  
  
start17 <- start17[order(start17$DOW\_down\_median),]  
final17 <- final17[order(final17$DOW\_down\_median),]  
dROI17 <- dROI17[order(dROI17$DOW\_down\_median),]  
  
DOW\_ROI <- as.data.frame(final17$DOW\_Close/start17$DOW\_Close)  
colnames(DOW\_ROI) <- 'DOW\_ROI'  
  
colnames(start17)[6] <- 'startValue'  
colnames(final17)[6] <- 'finalValue'  
  
dROI17$startValue <- start17$startValue  
dROI17$finalValue <- final17$finalValue  
dROI17$DOW\_ROI <- DOW\_ROI$DOW\_ROI  
dROI17$stock\_ROI <- dROI17$finalValue/dROI17$startValue  
dROI17

## # A tibble: 17 x 6  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 AAP 97.3 35.6 134. 2.36 3.75   
## 2 ADDYY 58.4 25 156. 2.36 6.24   
## 3 AMZN 551. 38.7 2135. 2.36 55.2   
## 4 COST 123. 52.8 318. 2.36 6.02   
## 5 DLTR 52.8 10.2 88.7 2.36 8.67   
## 6 FFIN 14.7 7.00 34.6 2.36 4.94   
## 7 GOOG 559. 233. 1521. 2.36 6.53   
## 8 HD 90.9 41.1 245. 2.36 5.97   
## 9 JNJ 91.1 66.4 150. 2.36 2.26   
## 10 NFLX 92.4 3.80 380. 2.36 100.   
## 11 NKE 39.7 12.2 104. 2.36 8.48   
## 12 PCG 44.1 47.3 16.2 2.36 0.343  
## 13 ROST 40.8 7.62 122. 2.36 16.0   
## 14 TEVA 41.1 31.3 12.2 2.36 0.391  
## 15 TJX 26.0 7.17 63.4 2.36 8.84   
## 16 WMT 69.8 47.5 118. 2.36 2.48   
## 17 XOM 81.4 74.1 60.7 2.36 0.818

Netflix killed the return on investment with more than 100 fold profits. Lets look at Netflix to see the highs and lows of this stock since 2007 and through till 2020.

nflx <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='NFLX')  
nflx1 <- subset(nflx, nflx$Year > 2011 & nflx$Year < 2014)  
nflx2 <- subset(nflx, nflx$Year > 2016 & nflx$Year < 2018)  
nflx3 <- subset(nflx, nflx$Year > 2018 & nflx$Year < 2020)  
  
gg3 <- ggplot(data = nflx, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('NFLX 2007-2020')+  
 ylab('Stock Value Dollars')  
gg4 <- ggplot(data = nflx1, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('NFLX 2011-2014')+  
 ylab('Stock Value Dollars')  
gg5 <- ggplot(data = nflx2, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('NFLX 2016-2018')+  
 ylab('Stock Value Dollars')  
gg6 <- ggplot(data = nflx3, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('NFLX 2018-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg3, gg4, gg5,gg6, nrow = 2)



Netflix was certainly a great stock to invest in at around 3 USD in 2007 and at around 325 USD in 2020. We see that Netflix was on the up and up almost its entire course growing to more than 100 times its initial starting value in 2007. There were some lows such as in 2012 there was small dip in the curve, then in July and August-September 2017, and also in the third quarter of 2019. But it still performed amazingly. The stock dreams of riches are made of and conartists use to get more money from people on risky start up penny stocks. But lets put out all we know about Netflix.

* Netflix was first heard from the author of this tutorial in 2003 when some roommate of a guy the author dated bragged about how awesome Netflix is to cost $7/month and you can rent new movies mailed to your home for no additional charge. This roommate also bought a flat screen tv for 7000 USD before they became ubiquitously priced from 200-500 USD five years later.
* The minimum wage for workers around CA in this time period was also about the cost of the Netflix monthly membership. Many tv shows started being options to rent from sources such as premium cable tv shows like Dexter around 2007 or so.
* I pulled the cord on cable due to high costs and got a Netflix membership for around 8-10 USD in about 2014. Which was also around the price of minimum wage at that time.
* cell phones became very great and needed personal items with fast wifi and internet streamings still at a cost that beat cable tv and home phone lines at this same time period.
* Netflix got more innovative, they started adding more Netflix produced shows in 2016 that made memes on instagram and facebook, the top social media platforms of the time in the 2010s like with ‘Orange is the New Black’ ( I couldn’t watch, but saw many memes on).

So, given what you know and what you scanned above, isn’t it no surprise that a stock that out competes alternative forms of entertainment, is low cost in price and able to be taken mobile or use anywhere and at any time for next to nothing in cost as an hour of a consumer’s 160-200 hour work month if working minimum wage and full time. Maybe its time to add another feature, like federal minimum wage rates to this data. We will in fact do this later, but for now we will add another feature that compares the ROI ration to that of the DOW Jones industrial average, and try to pull the most striking features out of that group.

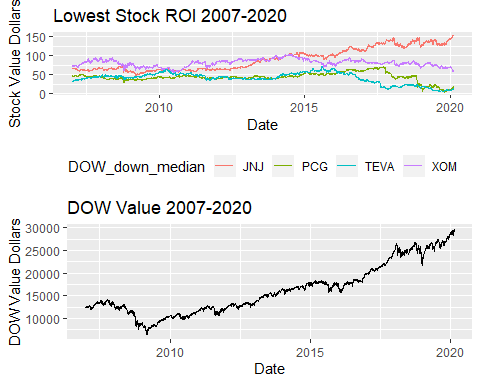
dROI17$stockBeatsDOW <- ifelse(dROI17$stock\_ROI > dROI17$DOW\_ROI, 'Yes', 'No')  
dROI17[,c(1,7)]

## # A tibble: 17 x 2  
## DOW\_down\_median stockBeatsDOW  
## <chr> <chr>   
## 1 AAP Yes   
## 2 ADDYY Yes   
## 3 AMZN Yes   
## 4 COST Yes   
## 5 DLTR Yes   
## 6 FFIN Yes   
## 7 GOOG Yes   
## 8 HD Yes   
## 9 JNJ No   
## 10 NFLX Yes   
## 11 NKE Yes   
## 12 PCG No   
## 13 ROST Yes   
## 14 TEVA No   
## 15 TJX Yes   
## 16 WMT Yes   
## 17 XOM No

Looking at the above chart of the stocks that beat the DOW in return on investment ratios of final stock value to initial stock value (2007-2020), Exxon Mobil, Johnson & Johnson, Teva Pharmaceuticals Industries, and Pacific Gas and Electric lossed money or had lower returns than that of the DOW. Of note, the pharmaceuticals might make you go on a wild tangent to know which company is supplying us with flu vaccines annually. If so, [Sanofi (SNY)](https://finance.yahoo.com/quote/SNY/) is the US’s largest supplier and they aren’t in this analysis. But it would be interesting to see when they make the most money, considering we have a CoV-19 flu contagion globally as of Feb. 2020. While, it is safe to say the companies that low income consumers love or live by did well, such as: Walmart, Adidas, Nike, Home Depot, Netflix, Amazon, Advance Auto Parts, TJ Maxx, Ross, Dollar Tree, and Costco. Costco is more of a middle class or small business store because you have to have cash, or used to have cash or money in your checking account to buy their goods and services with your atm card. They may have changed this. First Financial bank also did well, and it was selected because it was one of the banks available when hand picking these stocks. I don’t use it and don’t know anyone who does, and that is because I am on the West Coast, and this bank originates out of the East Coast. This could be an indicator that the East Coast is picking up in business and getting more home loans, business loans, etc. Than the more West Coast known banks like Citi, Chase, Bank of America, JP Morgan Chase. As these other banks did not perform well for median positive daily changes in stock prices during increasing unemployment and decreasing DOW values.

Lets plot those four stocks in this portfolio that did worse than the DOW.

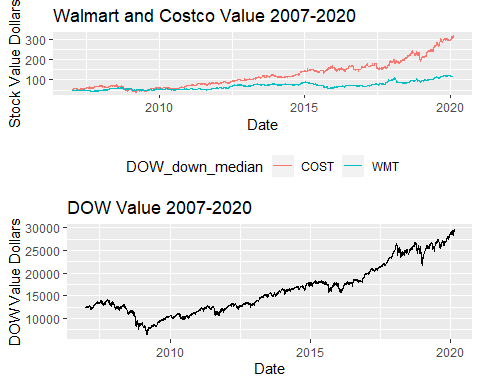
low4 <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median == 'JNJ'|  
 Close17\_dow$DOW\_down\_median == 'PCG'|  
 Close17\_dow$DOW\_down\_median == 'TEVA'|  
 Close17\_dow$DOW\_down\_median == 'XOM')  
gg7 <- ggplot(data = low4, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 scale\_fill\_brewer(palette="Spectral") +  
  
 theme(legend.position="bottom")+  
 ggtitle('Lowest Stock ROI 2007-2020')+  
 ylab('Stock Value Dollars')  
gg8 <- ggplot(data = low4, aes(x=Date, y=DOW\_Close)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('DOW Value 2007-2020')+  
 ylab('DOW Value Dollars')  
  
grid.arrange(gg7,gg8, nrow = 2)



From the above charts, It looks like Johnson & Johnson has started to move upward with the DOW starting around 2013. The TEVA stock seems to be negatively correlated with the DOW and Exxon Mobil was positively correlated with the DOW from 2007 to about 2015, then started moving in the opposite direction after 2015. Exxon supplies fuels to automobiles, while PCG supplies electricity to hybrid and electric vehicles in certain US regions. Yet, both started moving opposite directions with the DOW after 2015.

Lets now compare Costco and Walmart to each other and the DOW.

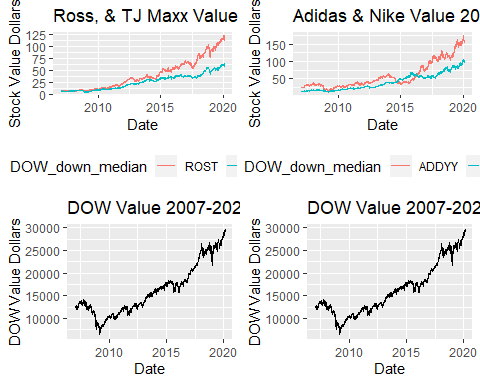
wal\_Cost <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='WMT' |   
 Close17\_dow$DOW\_down\_median=='COST')  
  
gg9 <- ggplot(data = wal\_Cost, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Walmart and Costco Value 2007-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg9,gg8, nrow = 2)



It is interesting to note that Costco and the DOW seem to be identical curves for the direction they move while Walmart seems also be increasing when the DOW does but at a much lower rate over time.

Lets now compare TJ Maxx, Ross, Nike, and Adidas to the DOW.

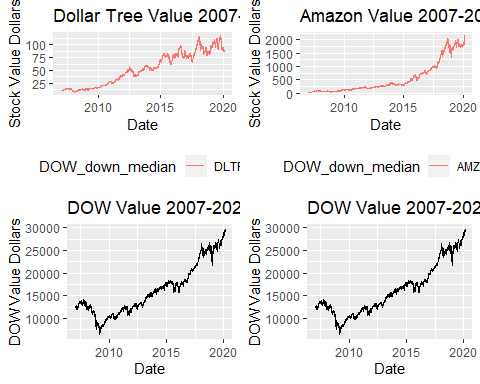
Retail <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median == 'TJX' |  
 Close17\_dow$DOW\_down\_median == 'ROST' )  
  
Shoes <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median == 'NKE' |  
 Close17\_dow$DOW\_down\_median == 'ADDYY' )  
  
  
gg10 <- ggplot(data = Retail, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Ross, & TJ Maxx Value 2007-2020')+  
 ylab('Stock Value Dollars')  
  
gg11 <- ggplot(data = Shoes, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Adidas & Nike Value 2007-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg10, gg11, gg8, gg8, nrow =2)



The DOW is plotted below each of the two plots of either Ross & TJ Maxx, or Adidas & Nike. Both Ross and Adidas did better than TJ Maxx and Nike. Except, that Nike did do better than Adidas between 2015 and 2016.

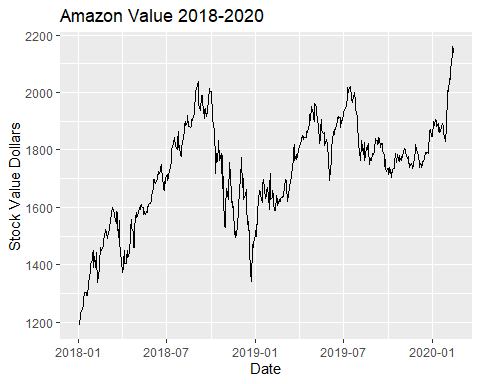
Now, lets look at Dollar Tree and Amazon compared to each other and the DOW in this time span.

DLRv <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median == 'DLTR' )  
AMZNv <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median == 'AMZN' )  
  
  
gg12 <- ggplot(data = DLRv, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Dollar Tree Value 2007-2020')+  
 ylab('Stock Value Dollars')  
  
gg13 <- ggplot(data = AMZNv, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Amazon Value 2007-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg12, gg13, gg8, gg8, nrow =2)



The above charts show Dollar Tree compared to Amazon and both compared to the DOW between 2007 and 2020. The Dollar Tree seems to be cyclical but overall increasing, while Amazon was a steady increase over the years except in 2018 where it had a decrease.

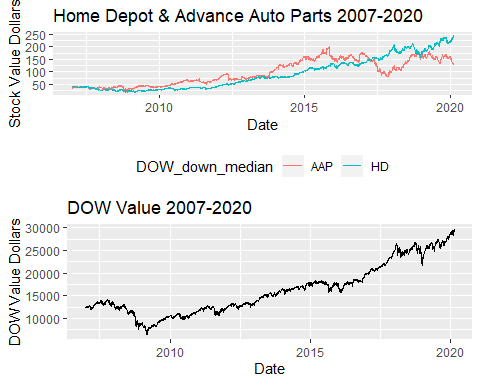
AMZNv2 <- subset(Close17\_dow, Close17\_dow$Year > 2017 &  
 Close17\_dow$DOW\_down\_median == 'AMZN')  
  
gg14 <- ggplot(data = AMZNv2, aes(x=Date, y=stockValue)) +  
 geom\_line()+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Amazon Value 2018-2020')+  
 ylab('Stock Value Dollars')  
gg14



We see a drop in value of Amazon after September 2018 until about January 2019 when it increases until just before the end of the 2nd quarter in 2019 then it drops in the 1st quarter staying low before increasing to a global maximum in January 2020.

Lets look at Home Depot and Advance Auto parts now.

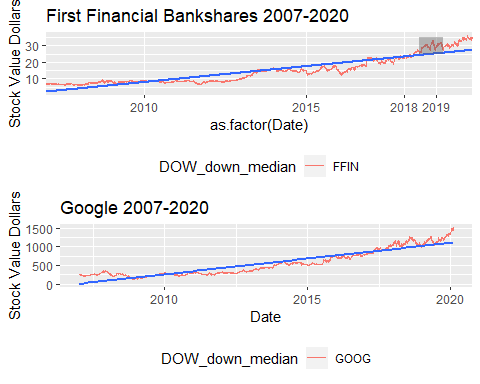
HDv <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='HD' |  
 Close17\_dow$DOW\_down\_median=='AAP')  
  
gg15 <- ggplot(data = HDv, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Home Depot & Advance Auto Parts 2007-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg15, gg8, nrow =2)



The charts above show that Home Depot and the DOW move together by increasing the time span from 2007-2020. Advance Auto Parts showed it had been increasing from 2007-2016, but then declined to a local minimum in the middle of 2017 where it then increased and remained a steady value up to 2020.

Lets also look at Google and First Financial Bankshares.

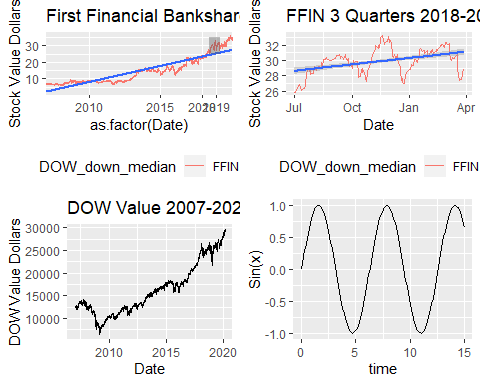
ffin <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='FFIN')  
goog <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='GOOG')  
  
gg16 <- ggplot(data=ffin, aes(x=as.factor(Date), y=stockValue, group=DOW\_down\_median))+  
 geom\_line(aes(color=DOW\_down\_median))+  
 geom\_smooth(method = "lm")+  
 annotate("rect", xmin = "2018-07-01", xmax = "2019-03-31", ymin = 25, ymax = 35,  
 alpha = .4)+  
 scale\_x\_discrete(breaks=c("2010-01-04","2015-01-02","2018-01-05",  
 "2019-01-03"),  
 labels=c("2010","2015", "2018","2019"))+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('First Financial Bankshares 2007-2020')+  
 ylab('Stock Value Dollars')  
   
gg17 <- ggplot(data = goog, aes(x=Date, y=stockValue, group=DOW\_down\_median)) +  
 geom\_line(aes(color=DOW\_down\_median))+  
 geom\_smooth(method = "lm")+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('Google 2007-2020')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg16, gg17,nrow =2)



The above chart shows Google and First Financial Bankshares from 2007-2020. First Financial has many highs and lows between 2015-2020 but is overall increasing, while Google is steadily increasing up till 2017 when it has a few highs and lows until 2020. Both have increased overall as indicated by the linear trendlines added to the two linear plots above for Google and First Financial Bankshares.

We know that some of these plots have time periods that are cyclical, but so far we know that overall, if we want to earn the most on our portfolios we would like to buy low and sell high. These stocks are increasing to rates higher than when they start a cyclical pattern. Lets examine the First Financial Bankshares stock when it sees these cyclical patterns further for some strategy development. There are three peaks or highs for FFIN after about 2017, so lets plot this.

ffin2 <- subset(ffin, ffin$Year==2018 & ffin$Month=='Jul'|  
 ffin$Year==2018 & ffin$Month=='Aug'|  
 ffin$Year==2018 & ffin$Month=='Sep'|  
 ffin$Year==2018 & ffin$Month=='Oct'|  
 ffin$Year==2018 & ffin$Month=='Nov'|  
 ffin$Year==2018 & ffin$Month=='Dec'|  
 ffin$Year==2019 & ffin$Month=='Jan'|  
 ffin$Year==2019 & ffin$Month=='Feb'|  
 ffin$Year==2019 & ffin$Month=='Mar')  
   
   
  
t <- seq(0,15,0.1)  
y <- sin(t)  
ty <- qplot(t,y, geom='path', xlab='time', ylab='Sin(x)')  
  
gg18 <- ggplot(data=ffin2, aes(x=Date, y=stockValue, group=DOW\_down\_median))+  
 geom\_line(aes(color=DOW\_down\_median))+  
 geom\_smooth(method = "lm")+  
 scale\_y\_continuous()+  
 theme(legend.position="bottom")+  
 ggtitle('FFIN 3 Quarters 2018-2019')+  
 ylab('Stock Value Dollars')  
  
grid.arrange(gg16,gg18,gg8, ty,nrow=2)



The above chart shows the sine curve to illustrate the highs and lows of the daily value changes for the First Financial Bankshares linear plot. If the curve was distorted at certain time intervals startind in the 3rd quarter of 2018 and ending before the 2nd quarter of 2019, this sine curve would show the three waves this stock saw. We are interested in buying at the beginning of the 3rd quarter when the price is low, then selling before the start of the 4th quarter when the stock is high, then buying again when it sees a dip in price after about the 1st month of the 4th quarter, then selling in the middle of the 4th quarter when high, buying at the end of the 4th quarter when low, and selling when high in the middle of the 1st quarter of 2019, and buying when low at the end of the 1st quarter, to maximize profits.

These quick analysis of the stock that did well when the DOW was decreasing and unemployment was increasing was interesting to look at. But now, lets look at these stocks and find out if we can indicate when a stock is good by setting a threshold for the number of minimums the stock has and if it decreases by more than its value in the last quarter and also the last two quarters then compare this to the local maxima where it increases to more than a set threshold than it was valued at in the last quarter and also the last two quarters. Count the number of times this happens and compare to the data on the stock ROI. This will require adding in more columns to calculate the daily change compared to a median value of the stock in each quarter. We have the years, the months, and the stock values in the table we are currently using.

We can do this by using time lags with the dplyr package. We should already have this package loaded. We have been using for median and mean calculations when grouping by stock. The data set we should use could be this one, Close17\_dow, or we could spread out the stock names back to being columns and add the stock lags for each stock for 7 days, 30 days, 90 days, and 180 days. Don’t get too excited, we have 53 stocks to do this for, or we could just do it with these 17 stocks that made the most profit from having positive median values when unemployment was increasing and the DOW was decreasing. I vote we do the 17. \*\*\*

We are going to create time lags of 7,30,60,90, 120, 150, and 180 days to see if there are any rolling stock values that could indicate when to buy or sell and pin point are threshold values and possibly work this into an automated program later with continuous uploads of these stocks in monitoring our portfolio of stock.

Close17\_dow <- Close17\_dow[with(Close17\_dow, order(DOW\_down\_median, Date)),]  
  
Laap <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='AAP')  
Laddyy <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='ADDYY')  
Lamzn <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='AMZN')  
Lcost <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='COST')  
Ldltr <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='DLTR')  
Lffin <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='FFIN')  
Lgoog <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='GOOG')  
Lhd <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='HD')  
Ljnj <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='JNJ')  
Lnflx <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='NFLX')  
Lnke <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='NKE')  
Lpcg <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='PCG')  
Lrost <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='ROST')  
Lteva <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='TEVA')  
Ltjx <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='TJX')  
Lwmt <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='WMT')  
Lxom <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='XOM')  
  
aapL7 <- lag(Laap$stockValue,7)  
addyyL7 <- lag(Laddyy$stockValue,7)  
amznL7 <- lag(Lamzn$stockValue,7)  
costL7 <- lag(Lcost$stockValue,7)  
dltrL7 <- lag(Ldltr$stockValue,7)  
ffinL7 <- lag(Lffin$stockValue,7)  
googL7 <- lag(Lgoog$stockValue,7)  
hdL7 <- lag(Lhd$stockValue,7)  
jnjL7 <- lag(Ljnj$stockValue,7)  
nflxL7 <- lag(Lnflx$stockValue,7)  
nkeL7 <- lag(Lnke$stockValue,7)  
pcgL7 <- lag(Lpcg$stockValue,7)  
rostL7 <- lag(Lrost$stockValue,7)  
tevaL7 <- lag(Lteva$stockValue,7)  
tjxL7 <- lag(Ltjx$stockValue,7)  
wmtL7 <- lag(Lwmt$stockValue,7)  
xomL7 <- lag(Lxom$stockValue,7)  
  
Close17\_dow$lag7 <- c(aapL7,addyyL7,amznL7,costL7,dltrL7,ffinL7,googL7,  
 hdL7,jnjL7,nflxL7,nkeL7,pcgL7,rostL7,tevaL7,  
 tjxL7,wmtL7,xomL7)  
  
aapL30 <- lag(Laap$stockValue,30)  
addyyL30 <- lag(Laddyy$stockValue,30)  
amznL30 <- lag(Lamzn$stockValue,30)  
costL30 <- lag(Lcost$stockValue,30)  
dltrL30 <- lag(Ldltr$stockValue,30)  
ffinL30 <- lag(Lffin$stockValue,30)  
googL30 <- lag(Lgoog$stockValue,30)  
hdL30 <- lag(Lhd$stockValue,30)  
jnjL30 <- lag(Ljnj$stockValue,30)  
nflxL30 <- lag(Lnflx$stockValue,30)  
nkeL30 <- lag(Lnke$stockValue,30)  
pcgL30 <- lag(Lpcg$stockValue,30)  
rostL30 <- lag(Lrost$stockValue,30)  
tevaL30 <- lag(Lteva$stockValue,30)  
tjxL30 <- lag(Ltjx$stockValue,30)  
wmtL30 <- lag(Lwmt$stockValue,30)  
xomL30 <- lag(Lxom$stockValue,30)  
  
Close17\_dow$lag30 <- c(aapL30,addyyL30,amznL30,costL30,dltrL30,ffinL30,googL30,  
 hdL30,jnjL30,nflxL30,nkeL30,pcgL30,rostL30,tevaL30,  
 tjxL30,wmtL30,xomL30)  
  
  
aapL60 <- lag(Laap$stockValue,60)  
addyyL60 <- lag(Laddyy$stockValue,60)  
amznL60 <- lag(Lamzn$stockValue,60)  
costL60 <- lag(Lcost$stockValue,60)  
dltrL60 <- lag(Ldltr$stockValue,60)  
ffinL60 <- lag(Lffin$stockValue,60)  
googL60 <- lag(Lgoog$stockValue,60)  
hdL60 <- lag(Lhd$stockValue,60)  
jnjL60 <- lag(Ljnj$stockValue,60)  
nflxL60 <- lag(Lnflx$stockValue,60)  
nkeL60 <- lag(Lnke$stockValue,60)  
pcgL60 <- lag(Lpcg$stockValue,60)  
rostL60 <- lag(Lrost$stockValue,60)  
tevaL60 <- lag(Lteva$stockValue,60)  
tjxL60 <- lag(Ltjx$stockValue,60)  
wmtL60 <- lag(Lwmt$stockValue,60)  
xomL60 <- lag(Lxom$stockValue,60)  
  
Close17\_dow$lag60 <- c(aapL60,addyyL60,amznL60,costL60,dltrL60,ffinL60,googL60,  
 hdL60,jnjL60,nflxL60,nkeL60,pcgL60,rostL60,tevaL60,  
 tjxL60,wmtL60,xomL60)  
  
aapL90 <- lag(Laap$stockValue,90)  
addyyL90 <- lag(Laddyy$stockValue,90)  
amznL90 <- lag(Lamzn$stockValue,90)  
costL90 <- lag(Lcost$stockValue,90)  
dltrL90 <- lag(Ldltr$stockValue,90)  
ffinL90 <- lag(Lffin$stockValue,90)  
googL90 <- lag(Lgoog$stockValue,90)  
hdL90 <- lag(Lhd$stockValue,90)  
jnjL90 <- lag(Ljnj$stockValue,90)  
nflxL90 <- lag(Lnflx$stockValue,90)  
nkeL90 <- lag(Lnke$stockValue,90)  
pcgL90 <- lag(Lpcg$stockValue,90)  
rostL90 <- lag(Lrost$stockValue,90)  
tevaL90 <- lag(Lteva$stockValue,90)  
tjxL90 <- lag(Ltjx$stockValue,90)  
wmtL90 <- lag(Lwmt$stockValue,90)  
xomL90 <- lag(Lxom$stockValue,90)  
  
Close17\_dow$lag90 <- c(aapL90,addyyL90,amznL90,costL90,dltrL90,ffinL90,googL90,  
 hdL90,jnjL90,nflxL90,nkeL90,pcgL90,rostL90,tevaL90,  
 tjxL90,wmtL90,xomL90)  
  
aapL120 <- lag(Laap$stockValue,120)  
addyyL120 <- lag(Laddyy$stockValue,120)  
amznL120 <- lag(Lamzn$stockValue,120)  
costL120 <- lag(Lcost$stockValue,120)  
dltrL120 <- lag(Ldltr$stockValue,120)  
ffinL120 <- lag(Lffin$stockValue,120)  
googL120 <- lag(Lgoog$stockValue,120)  
hdL120 <- lag(Lhd$stockValue,120)  
jnjL120 <- lag(Ljnj$stockValue,120)  
nflxL120 <- lag(Lnflx$stockValue,120)  
nkeL120 <- lag(Lnke$stockValue,120)  
pcgL120 <- lag(Lpcg$stockValue,120)  
rostL120 <- lag(Lrost$stockValue,120)  
tevaL120 <- lag(Lteva$stockValue,120)  
tjxL120 <- lag(Ltjx$stockValue,120)  
wmtL120 <- lag(Lwmt$stockValue,120)  
xomL120 <- lag(Lxom$stockValue,120)  
  
Close17\_dow$lag120 <- c(aapL120,addyyL120,amznL120,costL120,dltrL120,ffinL120,googL120,  
 hdL120,jnjL120,nflxL120,nkeL120,pcgL120,rostL120,tevaL120,  
 tjxL120,wmtL120,xomL120)  
  
aapL150 <- lag(Laap$stockValue,150)  
addyyL150 <- lag(Laddyy$stockValue,150)  
amznL150 <- lag(Lamzn$stockValue,150)  
costL150 <- lag(Lcost$stockValue,150)  
dltrL150 <- lag(Ldltr$stockValue,150)  
ffinL150 <- lag(Lffin$stockValue,150)  
googL150 <- lag(Lgoog$stockValue,150)  
hdL150 <- lag(Lhd$stockValue,150)  
jnjL150 <- lag(Ljnj$stockValue,150)  
nflxL150 <- lag(Lnflx$stockValue,150)  
nkeL150 <- lag(Lnke$stockValue,150)  
pcgL150 <- lag(Lpcg$stockValue,150)  
rostL150 <- lag(Lrost$stockValue,150)  
tevaL150 <- lag(Lteva$stockValue,150)  
tjxL150 <- lag(Ltjx$stockValue,150)  
wmtL150 <- lag(Lwmt$stockValue,150)  
xomL150 <- lag(Lxom$stockValue,150)  
  
Close17\_dow$lag150 <- c(aapL150,addyyL150,amznL150,costL150,dltrL150,ffinL150,googL150,  
 hdL150,jnjL150,nflxL150,nkeL150,pcgL150,rostL150,tevaL150,  
 tjxL150,wmtL150,xomL150)  
  
aapL180 <- lag(Laap$stockValue,180)  
addyyL180 <- lag(Laddyy$stockValue,180)  
amznL180 <- lag(Lamzn$stockValue,180)  
costL180 <- lag(Lcost$stockValue,180)  
dltrL180 <- lag(Ldltr$stockValue,180)  
ffinL180 <- lag(Lffin$stockValue,180)  
googL180 <- lag(Lgoog$stockValue,180)  
hdL180 <- lag(Lhd$stockValue,180)  
jnjL180 <- lag(Ljnj$stockValue,180)  
nflxL180 <- lag(Lnflx$stockValue,180)  
nkeL180 <- lag(Lnke$stockValue,180)  
pcgL180 <- lag(Lpcg$stockValue,180)  
rostL180 <- lag(Lrost$stockValue,180)  
tevaL180 <- lag(Lteva$stockValue,180)  
tjxL180 <- lag(Ltjx$stockValue,180)  
wmtL180 <- lag(Lwmt$stockValue,180)  
xomL180 <- lag(Lxom$stockValue,180)  
  
Close17\_dow$lag180 <- c(aapL180,addyyL180,amznL180,costL180,dltrL180,ffinL180,googL180,  
 hdL180,jnjL180,nflxL180,nkeL180,pcgL180,rostL180,tevaL180,  
 tjxL180,wmtL180,xomL180)

Save this new table by writing it to csv file. Then we will see how many times each stock is lower than 7,30,60,90,120,150, and 180 days prior in stock value prices for each stock. See if we can use this to automate a data set that selects the stock as good or bad to buy, or good or bad to buy/sell. We could also use this information to create a machine learning data set that will use this information for those stocks that are good/bad at certain points in time to predict what its price will be or if it will return a profit. We already know four of these stocks didn’t return a profit, but they are in this portfolio of 17 stocks whose median values were positive when the DOW was decreasing and unemployment was increasing. The other 13 stocks returned a profit, and some substantially such as Netflix with 100 fold increased value, and Amazon with 55 fold increased value.

write.csv(Close17\_dow, 'Close17\_dow\_lags.csv', row.names=FALSE)

head(Close17\_dow,10)

## Date DOW\_down\_median Month Year UE\_monthlyRate stockValue DOW\_Close  
## 1 2007-01-03 AAP Jan 2007 4.6 35.58 12474.52  
## 18 2007-01-04 AAP Jan 2007 4.6 35.81 12480.69  
## 35 2007-01-05 AAP Jan 2007 4.6 35.02 12398.01  
## 52 2007-01-08 AAP Jan 2007 4.6 35.14 12423.49  
## 69 2007-01-09 AAP Jan 2007 4.6 35.44 12416.60  
## 86 2007-01-10 AAP Jan 2007 4.6 35.49 12442.16  
## 103 2007-01-11 AAP Jan 2007 4.6 36.40 12514.98  
## 120 2007-01-12 AAP Jan 2007 4.6 36.20 12556.08  
## 137 2007-01-16 AAP Jan 2007 4.6 36.20 12582.59  
## 154 2007-01-17 AAP Jan 2007 4.6 36.35 12577.15  
## lag7 lag30 lag60 lag90 lag120 lag150 lag180  
## 1 NA NA NA NA NA NA NA  
## 18 NA NA NA NA NA NA NA  
## 35 NA NA NA NA NA NA NA  
## 52 NA NA NA NA NA NA NA  
## 69 NA NA NA NA NA NA NA  
## 86 NA NA NA NA NA NA NA  
## 103 NA NA NA NA NA NA NA  
## 120 35.58 NA NA NA NA NA NA  
## 137 35.81 NA NA NA NA NA NA  
## 154 35.02 NA NA NA NA NA NA

tail(Close17\_dow,10)

## Date DOW\_down\_median Month Year UE\_monthlyRate stockValue DOW\_Close  
## 55998 2020-02-03 XOM Feb 2020 NA 60.73 28399.81  
## 56015 2020-02-04 XOM Feb 2020 NA 59.97 28807.63  
## 56032 2020-02-05 XOM Feb 2020 NA 62.73 29290.85  
## 56049 2020-02-06 XOM Feb 2020 NA 61.88 29379.77  
## 56066 2020-02-07 XOM Feb 2020 NA 61.47 29102.51  
## 56083 2020-02-10 XOM Feb 2020 NA 59.96 29276.82  
## 56100 2020-02-11 XOM Feb 2020 NA 60.53 29276.34  
## 56117 2020-02-12 XOM Feb 2020 NA 61.27 29551.42  
## 56134 2020-02-13 XOM Feb 2020 NA 60.93 29423.31  
## 56151 2020-02-14 XOM Feb 2020 NA 60.65 29398.08  
## lag7 lag30 lag60 lag90 lag120 lag150 lag180  
## 55998 66.77 69.87 73.09 71.14 69.63 76.63 76.36  
## 56015 66.32 69.39 71.49 71.35 70.49 76.56 75.91  
## 56032 64.74 69.94 73.01 70.97 67.65 75.72 75.90  
## 56049 64.65 70.29 70.77 71.48 67.25 76.44 76.25  
## 56066 64.11 70.02 70.34 70.61 68.30 76.13 75.56  
## 56083 64.79 70.13 69.37 68.95 69.45 76.48 73.79  
## 56100 62.12 69.89 68.80 67.15 69.03 76.43 74.10  
## 56117 60.73 69.48 68.50 67.98 69.72 77.51 72.61  
## 56134 59.97 69.78 69.19 68.97 69.57 77.57 72.16  
## 56151 62.73 70.90 68.52 68.02 67.49 77.63 71.97

Now create the ratios of current day’s stock value to the lag day stock value. This will allow us to see how these stocks compare to previous days, and get median and average fold change values or ratios.

Close17\_dow$today2lag7 <- Close17\_dow$lag7/Close17\_dow$stockValue  
Close17\_dow$today2lag30 <- Close17\_dow$lag30/Close17\_dow$stockValue  
Close17\_dow$today2lag60 <- Close17\_dow$lag60/Close17\_dow$stockValue  
Close17\_dow$today2lag90 <- Close17\_dow$lag90/Close17\_dow$stockValue  
Close17\_dow$today2lag120 <- Close17\_dow$lag120/Close17\_dow$stockValue  
Close17\_dow$today2lag150 <- Close17\_dow$lag150/Close17\_dow$stockValue  
Close17\_dow$today2lag180 <- Close17\_dow$lag180/Close17\_dow$stockValue

Group by the stock names in this table and add a median value for each lag value.

median17today2lag7 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag7), median, na.rm=T)  
median17today2lag30 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag30), median, na.rm=T)  
median17today2lag60 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag60), median, na.rm=T)  
median17today2lag90 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag90), median, na.rm=T)  
median17today2lag120 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag120), median, na.rm=T)  
median17today2lag150 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag150), median, na.rm=T)  
median17today2lag180 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag180), median, na.rm=T)  
  
mediantoday2lags <- cbind(median17today2lag7, median17today2lag30[2], median17today2lag60[2], median17today2lag90[2],  
 median17today2lag120[2], median17today2lag150[2], median17today2lag180[2])  
colnames(mediantoday2lags)[2:8] <- paste('median',colnames(mediantoday2lags)[2:8], sep='')  
Close17\_dow1 <- merge(Close17\_dow, mediantoday2lags, by.x='DOW\_down\_median', by.y='DOW\_down\_median')

Group by the stock name and now add a mean value. This will be to gather our threshold values to possibly indicate when its good to buy or sell. We will compare this later to the true outcome of each stock.

meantoday2lag7 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag7),mean, na.rm=T)  
meantoday2lag30 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag30),mean, na.rm=T)  
meantoday2lag60 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag60),mean, na.rm=T)  
meantoday2lag90 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag90),mean, na.rm=T)  
meantoday2lag120 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag120),mean, na.rm=T)  
meantoday2lag150 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag150),mean, na.rm=T)  
meantoday2lag180 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag180),mean, na.rm=T)  
  
meantoday2lags <- cbind(meantoday2lag7, meantoday2lag30[2], meantoday2lag60[2], meantoday2lag90[2],  
 meantoday2lag120[2], meantoday2lag150[2], meantoday2lag180[2])  
colnames(meantoday2lags)[2:8] <- paste('mean',colnames(meantoday2lags)[2:8], sep='')  
Close17\_dow2 <- merge(Close17\_dow1, meantoday2lags, by.x='DOW\_down\_median', by.y='DOW\_down\_median')

We should also get the min to know our local minimas in each today2lag.

minimumtoday2lag7 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag7),min, na.rm=T)  
minimumtoday2lag30 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag30),min, na.rm=T)  
minimumtoday2lag60 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag60),min, na.rm=T)  
minimumtoday2lag90 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag90),min, na.rm=T)  
minimumtoday2lag120 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag120),min, na.rm=T)  
minimumtoday2lag150 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag150),min, na.rm=T)  
minimumtoday2lag180 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag180),min, na.rm=T)  
  
minimumtoday2lags <- cbind(minimumtoday2lag7, minimumtoday2lag30[2], minimumtoday2lag60[2], minimumtoday2lag90[2],  
 minimumtoday2lag120[2], minimumtoday2lag150[2], minimumtoday2lag180[2])  
colnames(minimumtoday2lags)[2:8] <- paste('minimum',colnames(minimumtoday2lags)[2:8], sep='')  
Close17\_dow3 <- merge(Close17\_dow2, minimumtoday2lags, by.x='DOW\_down\_median', by.y='DOW\_down\_median')

Lets also add our local maximas by getting the max values for each of these today2lags.

maximumtoday2lag7 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag7),max, na.rm=T)  
maximumtoday2lag30 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag30),max, na.rm=T)  
maximumtoday2lag60 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag60),max, na.rm=T)  
maximumtoday2lag90 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag90),max, na.rm=T)  
maximumtoday2lag120 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag120),max, na.rm=T)  
maximumtoday2lag150 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag150),max, na.rm=T)  
maximumtoday2lag180 <- Close17\_dow %>% group\_by(DOW\_down\_median) %>%  
 summarise\_at(vars(today2lag180),max, na.rm=T)  
  
maximumtoday2lags <- cbind(maximumtoday2lag7, maximumtoday2lag30[2], maximumtoday2lag60[2], maximumtoday2lag90[2],  
 maximumtoday2lag120[2], maximumtoday2lag150[2], maximumtoday2lag180[2])  
colnames(maximumtoday2lags)[2:8] <- paste('maximum',colnames(maximumtoday2lags)[2:8], sep='')  
Close17\_dow4 <- merge(Close17\_dow3, maximumtoday2lags, by.x='DOW\_down\_median', by.y='DOW\_down\_median')

Write to file

write.csv(Close17\_dow4, 'Close17\_dow4\_lagStats.csv', row.names=FALSE)

Smaller file of the above, you can rbind() the two together, after importing them if needed.

Close17\_dow4\_a <- Close17\_dow4[1:28500,]  
Close17\_dow4\_b <- Close17\_dow4[28501:56151,]  
write.csv(Close17\_dow4\_a, 'Close17\_dow4\_lagStats\_parta.csv', row.names=TRUE)  
write.csv(Close17\_dow4\_b, 'Close17\_dow4\_lagStats\_partb.csv', row.names=TRUE)

The fields for the median and mean ratios of todays value to the value 7,30,60,90,120,150, and 180 days prior give some useful information. The minimum and maximum fields give the minimum and maximum ratio values for each group and could also provide some useful information.

Lets get the median values of the 17 stock in terms of each median lag and mean lag ratio. Then compare. But lets get a summary of those fields first to see the quantiles, that could help us decide which stocks to categorize based on its behavior in time.

summary(Close17\_dow4[22:35])

## mediantoday2lag7 mediantoday2lag30 mediantoday2lag60 mediantoday2lag90  
## Min. :0.9897 Min. :0.9452 Min. :0.8916 Min. :0.8493   
## 1st Qu.:0.9937 1st Qu.:0.9742 1st Qu.:0.9511 1st Qu.:0.9348   
## Median :0.9947 Median :0.9787 Median :0.9570 Median :0.9412   
## Mean :0.9951 Mean :0.9791 Mean :0.9599 Mean :0.9454   
## 3rd Qu.:0.9971 3rd Qu.:0.9893 3rd Qu.:0.9810 3rd Qu.:0.9681   
## Max. :1.0001 Max. :0.9996 Max. :1.0061 Max. :1.0089   
## mediantoday2lag120 mediantoday2lag150 mediantoday2lag180 meantoday2lag7   
## Min. :0.8105 Min. :0.7771 Min. :0.7388 Min. :0.9932   
## 1st Qu.:0.9091 1st Qu.:0.8921 1st Qu.:0.8667 1st Qu.:0.9963   
## Median :0.9241 Median :0.9079 Median :0.8928 Median :0.9971   
## Mean :0.9312 Mean :0.9184 Mean :0.9023 Mean :0.9979   
## 3rd Qu.:0.9630 3rd Qu.:0.9549 3rd Qu.:0.9455 3rd Qu.:0.9985   
## Max. :1.0158 Max. :1.0208 Max. :1.0235 Max. :1.0063   
## meantoday2lag30 meantoday2lag60 meantoday2lag90 meantoday2lag120  
## Min. :0.9715 Min. :0.9446 Min. :0.9185 Min. :0.8938   
## 1st Qu.:0.9837 1st Qu.:0.9676 1st Qu.:0.9515 1st Qu.:0.9363   
## Median :0.9879 Median :0.9755 Median :0.9624 Median :0.9493   
## Mean :0.9909 Mean :0.9827 Mean :0.9748 Mean :0.9663   
## 3rd Qu.:0.9935 3rd Qu.:0.9866 3rd Qu.:0.9798 3rd Qu.:0.9728   
## Max. :1.0250 Max. :1.0588 Max. :1.0879 Max. :1.1073   
## meantoday2lag150 meantoday2lag180  
## Min. :0.8680 Min. :0.8424   
## 1st Qu.:0.9220 1st Qu.:0.9075   
## Median :0.9358 Median :0.9233   
## Mean :0.9571 Mean :0.9477   
## 3rd Qu.:0.9657 3rd Qu.:0.9578   
## Max. :1.1225 Max. :1.1520

Lets recall which stocks were not returning profits and those that were. There were three stocks that didn’t return profits(XOM, TEVA, and PCG), and 14 that did. But only those three and one other stock didn’t beat the DOW (JNJ). The table with this information is dROI17.

dROI17

## # A tibble: 17 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 AAP 97.3 35.6 134. 2.36 3.75   
## 2 ADDYY 58.4 25 156. 2.36 6.24   
## 3 AMZN 551. 38.7 2135. 2.36 55.2   
## 4 COST 123. 52.8 318. 2.36 6.02   
## 5 DLTR 52.8 10.2 88.7 2.36 8.67   
## 6 FFIN 14.7 7.00 34.6 2.36 4.94   
## 7 GOOG 559. 233. 1521. 2.36 6.53   
## 8 HD 90.9 41.1 245. 2.36 5.97   
## 9 JNJ 91.1 66.4 150. 2.36 2.26   
## 10 NFLX 92.4 3.80 380. 2.36 100.   
## 11 NKE 39.7 12.2 104. 2.36 8.48   
## 12 PCG 44.1 47.3 16.2 2.36 0.343  
## 13 ROST 40.8 7.62 122. 2.36 16.0   
## 14 TEVA 41.1 31.3 12.2 2.36 0.391  
## 15 TJX 26.0 7.17 63.4 2.36 8.84   
## 16 WMT 69.8 47.5 118. 2.36 2.48   
## 17 XOM 81.4 74.1 60.7 2.36 0.818  
## # … with 1 more variable: stockBeatsDOW <chr>

Lets get the median value of the stock\_ROI from the dROI17 table.

medianStockROI <- median(dROI17$stock\_ROI)  
medianStockROI

## [1] 6.024035

listROI\_middle <- dROI17[order(dROI17$stock\_ROI)[7:11],]  
listROI\_middle$DOW\_down\_median

## [1] "FFIN" "HD" "COST" "ADDYY" "GOOG"

The above shows us the median value of the return on investment on these 17 stocks is 6.02 or 600% from 2007-2020. The stocks in the middle of this list closest to the median value are the two before the middle and after the middle: FFIN, HD, COST, ADDYY, and GOOG.

listROI\_middle

## # A tibble: 5 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 FFIN 14.7 7.00 34.6 2.36 4.94  
## 2 HD 90.9 41.1 245. 2.36 5.97  
## 3 COST 123. 52.8 318. 2.36 6.02  
## 4 ADDYY 58.4 25 156. 2.36 6.24  
## 5 GOOG 559. 233. 1521. 2.36 6.53  
## # … with 1 more variable: stockBeatsDOW <chr>

Lets make count variables of these stock and get the values and counts for each time the stock went above or below its price 7, 30, 60, 90, 120, 150, and 180 days prior, and the same for the counts each stock was above those prices. Lets just use the Close17\_dow, instead of the added median, mean, min, and max lag value ratios.

FFIN <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='FFIN')  
  
ffin7 <- ifelse(FFIN$today2lag7>1, 1,0)  
ffin30 <- ifelse(FFIN$today2lag30>1,1,0)  
ffin60 <- ifelse(FFIN$today2lag60>1,1,0)  
ffin90 <- ifelse(FFIN$today2lag90>1,1,0)  
ffin120 <- ifelse(FFIN$today2lag120>1,1,0)  
ffin150 <- ifelse(FFIN$today2lag150>1,1,0)  
ffin180 <- ifelse(FFIN$today2lag180>1,1,0)  
  
neg7 <- sum(ffin7==0, na.rm=T)  
pos7 <- sum(ffin7==1, na.rm=T)  
  
neg30 <- sum(ffin30==0, na.rm=T)  
pos30 <- sum(ffin30==1, na.rm=T)  
  
neg60 <- sum(ffin60==0, na.rm=T)  
pos60 <- sum(ffin60==1, na.rm=T)  
  
neg90 <- sum(ffin90==0, na.rm=T)  
pos90 <- sum(ffin90==1, na.rm=T)  
  
neg120 <- sum(ffin120==0, na.rm=T)  
pos120 <- sum(ffin120==1, na.rm=T)  
  
neg150 <- sum(ffin150==0, na.rm=T)  
pos150 <- sum(ffin150==1, na.rm=T)  
  
neg180 <- sum(ffin180==0, na.rm=T)  
pos180 <- sum(ffin180==1, na.rm=T)  
  
ffin\_counts0 <- as.data.frame(c(neg7,neg30,neg60,neg90,neg120,neg150,neg180))  
ffin\_counts1 <- as.data.frame(c(pos7,pos30,pos60,pos90,pos120,pos150,pos180))  
ffin\_counts <- cbind(ffin\_counts0, ffin\_counts1)  
row.names(ffin\_counts) <- c('lag7','lag30','lag60','lag90','lag120','lag150','lag180')  
colnames(ffin\_counts) <- c('sum\_neg\_lags','sum\_pos\_lags')  
  
ffin\_counts

## sum\_neg\_lags sum\_pos\_lags  
## lag7 1786 1510  
## lag30 1939 1334  
## lag60 2069 1174  
## lag90 2133 1080  
## lag120 2096 1087  
## lag150 2210 943  
## lag180 2223 900

From the above chart on the number of days the current value was lower than 7, 30, 60, 90, 120, 150, or 180 days prior to todays’s stock value, there were more in each lag than in those days the stocks was higher. There are the fewest number of days the stock increases from 180 days prior’s stock value for FFIN stock shown above. This means you have more days to buy low and fewer to sell high.

We should add a cumulative field or make a cumulative sum variable for each of these seven groups of lags in stock value. This way we can know how many cumulative days there are for each positive or negative value. Are there more frequent days that the stock stays lower in value to its lag price before increasing and how many days does the stock increase before starting to decrease.

length(ffin7)

## [1] 3303

#rm nas or all nas as output from the today2lag7 stock ratio values  
ffin7\_a <- ffin7[-c(1:7)]  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for ffin7 <- <- ifelse(FFIN$today2lag7>1, 1,0)  
ffin7\_ab <- cumsum(ffin7\_a)  
  
ffin7\_abc <- as.data.frame(as.factor(ffin7\_ab))  
colnames(ffin7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countFFIN <- ffin7\_abc %>% group\_by(cSum) %>% count(n=n())  
countFFIN <- as.data.frame(countFFIN)  
countFFIN <- countFFIN[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countFFIN$decr\_Days <- countFFIN$n-1  
countFFIN <- countFFIN[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countFFIN2 <- subset(countFFIN, countFFIN$decr\_Days>0)  
summary(countFFIN2$decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.347 8.000 38.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
decr\_Days\_grouped <- countFFIN2 %>% group\_by(decr\_Days) %>% count(n=n())  
decr\_Days\_grouped <- decr\_Days\_grouped[,-3]

The above table countFFIN2 gives the number of days in a row the stock value decreased starting at the cSum variable value.

The decr\_Days\_grouped table shows the number of days in a row the stock decreased from 2007-2020, and how many times the stock decreased that many days in a row.

decr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: decr\_Days [23]  
## decr\_Days n  
## <dbl> <int>  
## 1 1 96  
## 2 2 54  
## 3 3 28  
## 4 4 22  
## 5 5 11  
## 6 6 13  
## 7 7 20  
## 8 8 21  
## 9 9 10  
## 10 10 12  
## # … with 13 more rows

From the above chart on the FFIN stock from 2007-2020, it decreased for one day before increasing, exactly 96 times, and it had 10 consecutive days of decreasing exactly 12 times.

median(decr\_Days\_grouped$decr\_Days)

## [1] 12

The median number of days it decreased consecutively was 12. And the chart above shows it did this exactly three times from 2007-2020. Meaning there were three intervals where this stock, FFIN, decreased for 12 days before increasing. This could be a feature for evaluating whether to buy the stock once, it has decreased for 12 days. We will test it later.

Now, it would be nice to wrap a way around getting the number of consecutive days the FFIN stock value of each instance compared to the value seven days prior increased. So that we could compare number of days and sets of increasing days to that of the decreasing days. We should be able to do this by changing the original variable that assigned 1 to increasing and instead assigned 1 to decreasing to calculate the days increasing with the same method we used above.

#assign a 1 to decreasing values  
ffin7\_b <- ifelse(FFIN$today2lag7>1, 0,1)  
  
#rm NAs  
ffin7\_b1 <- ffin7\_b[-c(1:7)]   
  
#counts the   
ffin7\_b2 <- cumsum(ffin7\_b1)  
  
ffin7\_b3 <- as.data.frame(as.factor(ffin7\_b2))  
colnames(ffin7\_b3) <- 'cSum'  
  
countFFIN1 <- ffin7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countFFIN1 <- as.data.frame(countFFIN1)  
countFFIN1 <- countFFIN1[,-3]  
  
countFFIN1$incr\_Days <- countFFIN1$n-1  
countFFIN1 <- countFFIN1[,-2]  
  
countFFIN3 <- subset(countFFIN1, countFFIN1$incr\_Days>0)  
summary(countFFIN3$incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.504 7.000 26.000

incr\_Days\_grouped <- countFFIN3 %>% group\_by(incr\_Days) %>% count(n=n())  
incr\_Days\_grouped <- incr\_Days\_grouped[,-3]

There were 26 consecutive days that the stock compared to seven days prior increased, and the median number of consecutive days the stock increased was 3 days in a row. The number of groups or times in this times series from 2007-2020 for FFIN that the stock increased consecutively is shown next.

incr\_Days\_grouped

## # A tibble: 21 x 2  
## # Groups: incr\_Days [21]  
## incr\_Days n  
## <dbl> <int>  
## 1 1 112  
## 2 2 50  
## 3 3 32  
## 4 4 28  
## 5 5 13  
## 6 6 12  
## 7 7 22  
## 8 8 10  
## 9 9 12  
## 10 10 10  
## # … with 11 more rows

The above chart shows that the stock had 21 sets of consecutive days it increased. FFIN had 10 consecutive days it increased compared to the value 7 days ago, and this happened ten times from 2007-2020.

median(incr\_Days\_grouped$incr\_Days)

## [1] 11

The median number of consecutive days the FFIN stock increased is 11 days, and from the previous chart we know this happened six times from 2007-2020.

We can now do this for the other time lags and see if we gain any information by manually making rules to decide a stock’s predictability as selling, buying, short and long term returns on investment as profitable, not profitable, or break even. This is just for this stock FFIN, we have to examine the other stock from this table of fold change in today/lag7 stock value for HD, GOOG, ADDYY, and COST stocks. This will grab the useful features of the median return stocks in this portfolio, then we will use the same method on the lowest return stocks and then again on the highest return stock.

Eventually, we will answer:

* how many times did this stock increase compared to the last 7, 30, 60, 90, 120, 150, and 180 days?
* how many times did this stock decrease compared to the last 7, 30, 60, 90, 120, 150, and 180 days?
* what is the median stock value fold change of todays/lagN (where N is one of above values)?
* how many times was this stock below their median value for each lag, and above it?
* what threshold appears to be the absolute minimum or range of decrease or increase in fold change before the stock increases or decreases?
* how many times was it below the DOW in fold change by lag? Does the DOW being down or up seem to affect how many consecutive days this stock is up or down?
* was the DOW high or low when the stock had their max number of consecutive days increasing or decreasing for each lag? Does the DOW seem to impact the number of consecutive days of increase or decrease of the stock?
* what was the [federal minimum](https://bebusinessed.com/history/history-of-minimum-wage/) wage of this time period? Does it affect the number of consecutive days the stock increases/decreases? This will be added later if available for our time period.Since 2009 the above link says the minimum wage has been 7.25 USD, although we know some states have higher minimum wages. It was 5.85 USD in 2007 and 6.55 USD in 2008
* was unemployment increasing or decreasing when the stock increased, decreased, or had more consecutive days of increasing or decreasing value compared to each lag?
* more questions as they develop.

Lets look again at our 17 stock that were in the best performing subset with return on investment in dollars filtered by those that had positive median daily change stock values when the DOW was down and unemployment higher than the last month. This was in our dROI17 table.

listROI <- dROI17[order(dROI17$stock\_ROI),]  
  
listROI\_low <- listROI[1:6,]  
listROI\_middle <- listROI[7:11,]  
listROI\_high <- listROI[12:17,]  
  
listROI

## # A tibble: 17 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 PCG 44.1 47.3 16.2 2.36 0.343  
## 2 TEVA 41.1 31.3 12.2 2.36 0.391  
## 3 XOM 81.4 74.1 60.7 2.36 0.818  
## 4 JNJ 91.1 66.4 150. 2.36 2.26   
## 5 WMT 69.8 47.5 118. 2.36 2.48   
## 6 AAP 97.3 35.6 134. 2.36 3.75   
## 7 FFIN 14.7 7.00 34.6 2.36 4.94   
## 8 HD 90.9 41.1 245. 2.36 5.97   
## 9 COST 123. 52.8 318. 2.36 6.02   
## 10 ADDYY 58.4 25 156. 2.36 6.24   
## 11 GOOG 559. 233. 1521. 2.36 6.53   
## 12 NKE 39.7 12.2 104. 2.36 8.48   
## 13 DLTR 52.8 10.2 88.7 2.36 8.67   
## 14 TJX 26.0 7.17 63.4 2.36 8.84   
## 15 ROST 40.8 7.62 122. 2.36 16.0   
## 16 AMZN 551. 38.7 2135. 2.36 55.2   
## 17 NFLX 92.4 3.80 380. 2.36 100.   
## # … with 1 more variable: stockBeatsDOW <chr>

The low median return on investment ratios of final/start values:

listROI\_low

## # A tibble: 6 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 PCG 44.1 47.3 16.2 2.36 0.343  
## 2 TEVA 41.1 31.3 12.2 2.36 0.391  
## 3 XOM 81.4 74.1 60.7 2.36 0.818  
## 4 JNJ 91.1 66.4 150. 2.36 2.26   
## 5 WMT 69.8 47.5 118. 2.36 2.48   
## 6 AAP 97.3 35.6 134. 2.36 3.75   
## # … with 1 more variable: stockBeatsDOW <chr>

The middle median return on investment ratios of final/start values that we only evaluated a seven day lag with for number of increasing and decreasing days in this time span of 2007-2020:

listROI\_middle

## # A tibble: 5 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 FFIN 14.7 7.00 34.6 2.36 4.94  
## 2 HD 90.9 41.1 245. 2.36 5.97  
## 3 COST 123. 52.8 318. 2.36 6.02  
## 4 ADDYY 58.4 25 156. 2.36 6.24  
## 5 GOOG 559. 233. 1521. 2.36 6.53  
## # … with 1 more variable: stockBeatsDOW <chr>

The high median return on investment ratios of final/start values:

listROI\_high

## # A tibble: 6 x 7  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 NKE 39.7 12.2 104. 2.36 8.48  
## 2 DLTR 52.8 10.2 88.7 2.36 8.67  
## 3 TJX 26.0 7.17 63.4 2.36 8.84  
## 4 ROST 40.8 7.62 122. 2.36 16.0   
## 5 AMZN 551. 38.7 2135. 2.36 55.2   
## 6 NFLX 92.4 3.80 380. 2.36 100.   
## # … with 1 more variable: stockBeatsDOW <chr>

Now we can add to these tables the number of increasing and decreasing days in this life cycle for each stock by their median number of increasing and decreasing days by lag 7. Lets do this only for lag 7. Later we will do this for lags 30-180 for each stock, to fill in some of those questions we wanted answers to for predicting the return on a stock based on those feature characteristics for each stock. We will know the return on investment and use the remaining stock in our 53 total stocks to categorize the return as low if at least the value of our median ROI of 6.02 as a ratio of final/start values. The other stocks above that value will be high. This will give two classes to predict if the stock will be a low or high stock return buy. For now, lets massage the data enough to get this additional data, then we can wrestle with it into our machine learning data model.

We need to now get the number of days that each of the other stock increased and decreased as a median value and for the 7 day lag.Lets make this table with default values set before adding each result as we get it for each stock. Lets also add the 3rd quantile to these stock for the top 75% of cumulative sums value in comparing lag 7 fold change of today’s stock value to 7 days ago.

dROI17\_lag7 <- dROI17  
dROI17\_lag7$medn\_cSum\_decr\_L7 <- 'median cSum down L7'  
dROI17\_lag7$Q3\_cSum\_decr\_L7 <- '3rd Qntl cSum down L7'  
dROI17\_lag7$max\_cSum\_decr\_L7 <- 'max cSum down L7'  
dROI17\_lag7$medn\_cSum\_incr\_L7 <- 'median cSum up L7'  
dROI17\_lag7$Q3\_cSum\_incr\_L7 <- '3rd Qntl cSun up L7'  
dROI17\_lag7$max\_cSum\_incr\_L7 <- 'max cSum up L7'

Lets run the FFIN (First Financial Bankshares) stock again, save the variables, grep it from the last table dROI17\_lag7, fill in with our values, and find and replace the stock name with the new stock moving forward.

FFIN <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='FFIN')  
  
#assign a 1 to increasing values  
ffin7 <- ifelse(FFIN$today2lag7>1, 1,0)  
  
ffin7\_a <- na.omit(ffin7)  
  
ffin7\_ab <- cumsum(ffin7\_a)  
  
ffin7\_abc <- as.data.frame(as.factor(ffin7\_ab))  
colnames(ffin7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countFFIN <- ffin7\_abc %>% group\_by(cSum) %>% count(n=n())  
countFFIN <- as.data.frame(countFFIN)  
countFFIN <- countFFIN[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countFFIN$decr\_Days <- countFFIN$n-1  
countFFIN <- countFFIN[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countFFIN1 <- subset(countFFIN, countFFIN$decr\_Days>0)  
summary(countFFIN1$decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.347 8.000 38.000

mxffin\_d <- max(countFFIN1$decr\_Days)  
mdnffin\_d <- median(countFFIN1$decr\_Days)  
q3ffin\_d <- as.numeric(as.character(summary(countFFIN1$decr\_Days)["3rd Qu."]))

#assign a 1 to decreasing values  
ffin7\_b <- ifelse(FFIN$today2lag7>1, 0,1)  
  
#rm NAs  
ffin7\_b1 <- na.omit(ffin7\_b)   
  
#counts the   
ffin7\_b2 <- cumsum(ffin7\_b1)  
  
ffin7\_b3 <- as.data.frame(as.factor(ffin7\_b2))  
colnames(ffin7\_b3) <- 'cSum'  
  
countFFIN2 <- ffin7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countFFIN2 <- as.data.frame(countFFIN2)  
countFFIN2 <- countFFIN2[,-3]  
  
countFFIN2$incr\_Days <- countFFIN2$n-1  
countFFIN2 <- countFFIN2[,-2]  
  
countFFIN3 <- subset(countFFIN2, countFFIN2$incr\_Days>0)  
summary(countFFIN3$incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.504 7.000 26.000

mxffin\_i <- max(countFFIN3$incr\_Days)  
mdnffin\_i <- median(countFFIN3$incr\_Days)  
q3ffin\_i <- as.numeric(as.character(summary(countFFIN3$incr\_Days)["3rd Qu."]))

Add these statistics to their respective stock instance in the dROI17\_lag7 table.

ffin\_g <- grep('FFIN',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[ffin\_g,8:13] <- c(mdnffin\_d,q3ffin\_d,mxffin\_d,mdnffin\_i,q3ffin\_i,mxffin\_i)

GOOG (Google) stats:

GOOG <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='GOOG')  
  
#assign a 1 to increasing values  
goog7<- ifelse(GOOG$today2lag7>1, 1,0)  
  
goog7\_a <- na.omit(goog7)  
  
goog7\_ab <- cumsum(goog7\_a)  
  
goog7\_abc <- as.data.frame(as.factor(goog7\_ab))  
colnames(goog7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countGOOG <- goog7\_abc %>% group\_by(cSum) %>% count(n=n())  
countGOOG <- as.data.frame(countGOOG)  
countGOOG <- countGOOG[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countGOOG$decr\_Days <- countGOOG$n-1  
countGOOG <- countGOOG[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countGOOG1 <- subset(countGOOG, countGOOG$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countGOOG1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 6.500 7.576 11.000 39.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxgoog\_d <- max(countGOOG1$decr\_Days)  
mdngoog\_d <- median(countGOOG1$decr\_Days)  
q3goog\_d <- as.numeric(as.character(summary(countGOOG1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
goog7\_b <- ifelse(GOOG$today2lag7>1, 0,1)  
  
goog7\_b1 <- na.omit(goog7\_b)  
  
#counts the   
goog7\_b2 <- cumsum(goog7\_b1)  
  
goog7\_b3 <- as.data.frame(as.factor(goog7\_b2))  
colnames(goog7\_b3) <- 'cSum'  
  
countGOOG2 <- goog7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countGOOG2 <- as.data.frame(countGOOG2)  
countGOOG2 <- countGOOG2[,-3]  
  
countGOOG2$incr\_Days <- countGOOG2$n-1  
countGOOG2 <- countGOOG2[,-2]  
  
countGOOG3 <- subset(countGOOG2, countGOOG2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countGOOG3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 4.000 5.627 8.000 28.000

mxgoog\_i <- max(countGOOG3$incr\_Days)  
mdngoog\_i <- median(countGOOG3$incr\_Days)  
q3goog\_i <- as.numeric(as.character(summary(countGOOG3$incr\_Days)["3rd Qu."]))  
  
  
goog\_g <- grep('GOOG',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[goog\_g,8:13] <- c(mdngoog\_d,q3goog\_d,mxgoog\_d,mdngoog\_i,q3goog\_i,mxgoog\_i)

HD-Home Depot:

HD <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='HD')  
  
#assign a 1 to increasing values  
hd7<- ifelse(HD$today2lag7>1, 1,0)  
  
hd7\_a <- na.omit(hd7)  
  
hd7\_ab <- cumsum(hd7\_a)  
  
hd7\_abc <- as.data.frame(as.factor(hd7\_ab))  
colnames(hd7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countHD <- hd7\_abc %>% group\_by(cSum) %>% count(n=n())  
countHD <- as.data.frame(countHD)  
countHD <- countHD[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countHD$decr\_Days <- countHD$n-1  
countHD <- countHD[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countHD1 <- subset(countHD, countHD$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countHD1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.486 10.000 30.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxhd\_d <- max(countHD1$decr\_Days)  
mdnhd\_d <- median(countHD1$decr\_Days)  
q3hd\_d <- as.numeric(as.character(summary(countHD1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
hd7\_b <- ifelse(HD$today2lag7>1, 0,1)  
  
hd7\_b1 <- na.omit(hd7\_b)  
  
#counts the   
hd7\_b2 <- cumsum(hd7\_b1)  
  
hd7\_b3 <- as.data.frame(as.factor(hd7\_b2))  
colnames(hd7\_b3) <- 'cSum'  
  
countHD2 <- hd7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countHD2 <- as.data.frame(countHD2)  
countHD2 <- countHD2[,-3]  
  
countHD2$incr\_Days <- countHD2$n-1  
countHD2 <- countHD2[,-2]  
  
countHD3 <- subset(countHD2, countHD2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countHD3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.035 8.000 28.000

mxhd\_i <- max(countHD3$incr\_Days)  
mdnhd\_i <- median(countHD3$incr\_Days)  
q3hd\_i <- as.numeric(as.character(summary(countHD3$incr\_Days)["3rd Qu."]))  
  
  
hd\_g <- grep('HD',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[hd\_g,8:13] <- c(mdnhd\_d,q3hd\_d,mxhd\_d,mdnhd\_i,q3hd\_i,mxhd\_i)

JNJ, Johnson & Johnson:

JNJ <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='JNJ')  
  
#assign a 1 to increasing values  
jnj7<- ifelse(JNJ$today2lag7>1, 1,0)  
  
jnj7\_a <- na.omit(jnj7)  
  
jnj7\_ab <- cumsum(jnj7\_a)  
  
jnj7\_abc <- as.data.frame(as.factor(jnj7\_ab))  
colnames(jnj7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countJNJ <- jnj7\_abc %>% group\_by(cSum) %>% count(n=n())  
countJNJ <- as.data.frame(countJNJ)  
countJNJ <- countJNJ[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countJNJ$decr\_Days <- countJNJ$n-1  
countJNJ <- countJNJ[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countJNJ1 <- subset(countJNJ, countJNJ$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countJNJ1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.139 8.000 36.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxjnj\_d <- max(countJNJ1$decr\_Days)  
mdnjnj\_d <- median(countJNJ1$decr\_Days)  
q3jnj\_d <- as.numeric(as.character(summary(countJNJ1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
jnj7\_b <- ifelse(JNJ$today2lag7>1, 0,1)  
  
jnj7\_b1 <- na.omit(jnj7\_b)  
  
#counts the   
jnj7\_b2 <- cumsum(jnj7\_b1)  
  
jnj7\_b3 <- as.data.frame(as.factor(jnj7\_b2))  
colnames(jnj7\_b3) <- 'cSum'  
  
countJNJ2 <- jnj7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countJNJ2 <- as.data.frame(countJNJ2)  
countJNJ2 <- countJNJ2[,-3]  
  
countJNJ2$incr\_Days <- countJNJ2$n-1  
countJNJ2 <- countJNJ2[,-2]  
  
countJNJ3 <- subset(countJNJ2, countJNJ2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countJNJ3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.051 7.000 33.000

mxjnj\_i <- max(countJNJ3$incr\_Days)  
mdnjnj\_i <- median(countJNJ3$incr\_Days)  
q3jnj\_i <- as.numeric(as.character(summary(countJNJ3$incr\_Days)["3rd Qu."]))  
  
  
jnj\_g <- grep('JNJ',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[jnj\_g,8:13] <- c(mdnjnj\_d,q3jnj\_d,mxjnj\_d,mdnjnj\_i,q3jnj\_i,mxjnj\_i)

NFLX, Netflix stats:

NFLX <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='NFLX')  
  
#assign a 1 to increasing values  
nflx7<- ifelse(NFLX$today2lag7>1, 1,0)  
  
nflx7\_a <- na.omit(nflx7)  
  
nflx7\_ab <- cumsum(nflx7\_a)  
  
nflx7\_abc <- as.data.frame(as.factor(nflx7\_ab))  
colnames(nflx7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countNFLX <- nflx7\_abc %>% group\_by(cSum) %>% count(n=n())  
countNFLX <- as.data.frame(countNFLX)  
countNFLX <- countNFLX[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countNFLX$decr\_Days <- countNFLX$n-1  
countNFLX <- countNFLX[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countNFLX1 <- subset(countNFLX, countNFLX$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countNFLX1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 5.500 6.896 10.000 37.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxnflx\_d <- max(countNFLX1$decr\_Days)  
mdnnflx\_d <- median(countNFLX1$decr\_Days)  
q3nflx\_d <- as.numeric(as.character(summary(countNFLX1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
nflx7\_b <- ifelse(NFLX$today2lag7>1, 0,1)  
  
nflx7\_b1 <- na.omit(nflx7\_b)  
  
#counts the   
nflx7\_b2 <- cumsum(nflx7\_b1)  
  
nflx7\_b3 <- as.data.frame(as.factor(nflx7\_b2))  
colnames(nflx7\_b3) <- 'cSum'  
  
countNFLX2 <- nflx7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countNFLX2 <- as.data.frame(countNFLX2)  
countNFLX2 <- countNFLX2[,-3]  
  
countNFLX2$incr\_Days <- countNFLX2$n-1  
countNFLX2 <- countNFLX2[,-2]  
  
countNFLX3 <- subset(countNFLX2, countNFLX2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countNFLX3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.307 7.750 25.000

mxnflx\_i <- max(countNFLX3$incr\_Days)  
mdnnflx\_i <- median(countNFLX3$incr\_Days)  
q3nflx\_i <- as.numeric(as.character(summary(countNFLX3$incr\_Days)["3rd Qu."]))  
  
  
nflx\_g <- grep('NFLX',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[nflx\_g,8:13] <- c(mdnnflx\_d,q3nflx\_d,mxnflx\_d,mdnnflx\_i,q3nflx\_i,mxnflx\_i)

NKE, Nike stats:

NKE <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='NKE')  
  
#assign a 1 to increasing values  
nke7<- ifelse(NKE$today2lag7>1, 1,0)  
  
nke7\_a <- na.omit(nke7)  
  
nke7\_ab <- cumsum(nke7\_a)  
  
nke7\_abc <- as.data.frame(as.factor(nke7\_ab))  
colnames(nke7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countNKE <- nke7\_abc %>% group\_by(cSum) %>% count(n=n())  
countNKE <- as.data.frame(countNKE)  
countNKE <- countNKE[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countNKE$decr\_Days <- countNKE$n-1  
countNKE <- countNKE[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countNKE1 <- subset(countNKE, countNKE$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countNKE1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.375 9.000 40.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxnke\_d <- max(countNKE1$decr\_Days)  
mdnnke\_d <- median(countNKE1$decr\_Days)  
q3nke\_d <- as.numeric(as.character(summary(countNKE1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
nke7\_b <- ifelse(NKE$today2lag7>1, 0,1)  
  
nke7\_b1 <- na.omit(nke7\_b)  
  
#counts the   
nke7\_b2 <- cumsum(nke7\_b1)  
  
nke7\_b3 <- as.data.frame(as.factor(nke7\_b2))  
colnames(nke7\_b3) <- 'cSum'  
  
countNKE2 <- nke7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countNKE2 <- as.data.frame(countNKE2)  
countNKE2 <- countNKE2[,-3]  
  
countNKE2$incr\_Days <- countNKE2$n-1  
countNKE2 <- countNKE2[,-2]  
  
countNKE3 <- subset(countNKE2, countNKE2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countNKE3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.773 7.000 22.000

mxnke\_i <- max(countNKE3$incr\_Days)  
mdnnke\_i <- median(countNKE3$incr\_Days)  
q3nke\_i <- as.numeric(as.character(summary(countNKE3$incr\_Days)["3rd Qu."]))  
  
  
nke\_g <- grep('NKE',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[nke\_g,8:13] <- c(mdnnke\_d,q3nke\_d,mxnke\_d,mdnnke\_i,q3nke\_i,mxnke\_i)

PCG, Pacific Gas and Electric stats:

PCG <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='PCG')  
  
#assign a 1 to increasing values  
pcg7<- ifelse(PCG$today2lag7>1, 1,0)  
  
pcg7\_a <- na.omit(pcg7)  
  
pcg7\_ab <- cumsum(pcg7\_a)  
  
pcg7\_abc <- as.data.frame(as.factor(pcg7\_ab))  
colnames(pcg7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countPCG <- pcg7\_abc %>% group\_by(cSum) %>% count(n=n())  
countPCG <- as.data.frame(countPCG)  
countPCG <- countPCG[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countPCG$decr\_Days <- countPCG$n-1  
countPCG <- countPCG[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countPCG1 <- subset(countPCG, countPCG$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countPCG1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.488 7.000 34.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxpcg\_d <- max(countPCG1$decr\_Days)  
mdnpcg\_d <- median(countPCG1$decr\_Days)  
q3pcg\_d <- as.numeric(as.character(summary(countPCG1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
pcg7\_b <- ifelse(PCG$today2lag7>1, 0,1)  
  
pcg7\_b1 <- na.omit(pcg7\_b)  
  
#counts the   
pcg7\_b2 <- cumsum(pcg7\_b1)  
  
pcg7\_b3 <- as.data.frame(as.factor(pcg7\_b2))  
colnames(pcg7\_b3) <- 'cSum'  
  
countPCG2 <- pcg7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countPCG2 <- as.data.frame(countPCG2)  
countPCG2 <- countPCG2[,-3]  
  
countPCG2$incr\_Days <- countPCG2$n-1  
countPCG2 <- countPCG2[,-2]  
  
countPCG3 <- subset(countPCG2, countPCG2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countPCG3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.731 7.000 27.000

mxpcg\_i <- max(countPCG3$incr\_Days)  
mdnpcg\_i <- median(countPCG3$incr\_Days)  
q3pcg\_i <- as.numeric(as.character(summary(countPCG3$incr\_Days)["3rd Qu."]))  
  
  
pcg\_g <- grep('PCG',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[pcg\_g,8:13] <- c(mdnpcg\_d,q3pcg\_d,mxpcg\_d,mdnpcg\_i,q3pcg\_i,mxpcg\_i)

ROST, Ross stores stats:

ROST <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='ROST')  
  
#assign a 1 to increasing values  
rost7<- ifelse(ROST$today2lag7>1, 1,0)  
  
rost7\_a <- na.omit(rost7)  
  
rost7\_ab <- cumsum(rost7\_a)  
  
rost7\_abc <- as.data.frame(as.factor(rost7\_ab))  
colnames(rost7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countROST <- rost7\_abc %>% group\_by(cSum) %>% count(n=n())  
countROST <- as.data.frame(countROST)  
countROST <- countROST[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countROST$decr\_Days <- countROST$n-1  
countROST <- countROST[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countROST1 <- subset(countROST, countROST$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countROST1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.047 9.000 32.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxrost\_d <- max(countROST1$decr\_Days)  
mdnrost\_d <- median(countROST1$decr\_Days)  
q3rost\_d <- as.numeric(as.character(summary(countROST1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
rost7\_b <- ifelse(ROST$today2lag7>1, 0,1)  
  
rost7\_b1 <- na.omit(rost7\_b)  
  
#counts the   
rost7\_b2 <- cumsum(rost7\_b1)  
  
rost7\_b3 <- as.data.frame(as.factor(rost7\_b2))  
colnames(rost7\_b3) <- 'cSum'  
  
countROST2 <- rost7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countROST2 <- as.data.frame(countROST2)  
countROST2 <- countROST2[,-3]  
  
countROST2$incr\_Days <- countROST2$n-1  
countROST2 <- countROST2[,-2]  
  
countROST3 <- subset(countROST2, countROST2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countROST3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 4.263 7.000 28.000

mxrost\_i <- max(countROST3$incr\_Days)  
mdnrost\_i <- median(countROST3$incr\_Days)  
q3rost\_i <- as.numeric(as.character(summary(countROST3$incr\_Days)["3rd Qu."]))  
  
  
rost\_g <- grep('ROST',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[rost\_g,8:13] <- c(mdnrost\_d,q3rost\_d,mxrost\_d,mdnrost\_i,q3rost\_i,mxrost\_i)

TEVA, TEVA Pharmaceuticals stats:

TEVA <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='TEVA')  
  
#assign a 1 to increasing values  
teva7<- ifelse(TEVA$today2lag7>1, 1,0)  
  
teva7\_a <- na.omit(teva7)  
  
teva7\_ab <- cumsum(teva7\_a)  
  
teva7\_abc <- as.data.frame(as.factor(teva7\_ab))  
colnames(teva7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countTEVA <- teva7\_abc %>% group\_by(cSum) %>% count(n=n())  
countTEVA <- as.data.frame(countTEVA)  
countTEVA <- countTEVA[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countTEVA$decr\_Days <- countTEVA$n-1  
countTEVA <- countTEVA[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countTEVA1 <- subset(countTEVA, countTEVA$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countTEVA1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.583 8.000 30.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxteva\_d <- max(countTEVA1$decr\_Days)  
mdnteva\_d <- median(countTEVA1$decr\_Days)  
q3teva\_d <- as.numeric(as.character(summary(countTEVA1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
teva7\_b <- ifelse(TEVA$today2lag7>1, 0,1)  
  
teva7\_b1 <- na.omit(teva7\_b)  
  
#counts the   
teva7\_b2 <- cumsum(teva7\_b1)  
  
teva7\_b3 <- as.data.frame(as.factor(teva7\_b2))  
colnames(teva7\_b3) <- 'cSum'  
  
countTEVA2 <- teva7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countTEVA2 <- as.data.frame(countTEVA2)  
countTEVA2 <- countTEVA2[,-3]  
  
countTEVA2$incr\_Days <- countTEVA2$n-1  
countTEVA2 <- countTEVA2[,-2]  
  
countTEVA3 <- subset(countTEVA2, countTEVA2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countTEVA3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.605 8.000 32.000

mxteva\_i <- max(countTEVA3$incr\_Days)  
mdnteva\_i <- median(countTEVA3$incr\_Days)  
q3teva\_i <- as.numeric(as.character(summary(countTEVA3$incr\_Days)["3rd Qu."]))  
  
  
teva\_g <- grep('TEVA',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[teva\_g,8:13] <- c(mdnteva\_d,q3teva\_d,mxteva\_d,mdnteva\_i,q3teva\_i,mxteva\_i)

TJX, TJ Maxx stores stats:

TJX <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='TJX')  
  
#assign a 1 to increasing values  
tjx7<- ifelse(TJX$today2lag7>1, 1,0)  
  
tjx7\_a <- na.omit(tjx7)  
  
tjx7\_ab <- cumsum(tjx7\_a)  
  
tjx7\_abc <- as.data.frame(as.factor(tjx7\_ab))  
colnames(tjx7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countTJX <- tjx7\_abc %>% group\_by(cSum) %>% count(n=n())  
countTJX <- as.data.frame(countTJX)  
countTJX <- countTJX[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countTJX$decr\_Days <- countTJX$n-1  
countTJX <- countTJX[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countTJX1 <- subset(countTJX, countTJX$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countTJX1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 4.00 5.96 8.00 35.00

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxtjx\_d <- max(countTJX1$decr\_Days)  
mdntjx\_d <- median(countTJX1$decr\_Days)  
q3tjx\_d <- as.numeric(as.character(summary(countTJX1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
tjx7\_b <- ifelse(TJX$today2lag7>1, 0,1)  
  
tjx7\_b1 <- na.omit(tjx7\_b)  
  
#counts the   
tjx7\_b2 <- cumsum(tjx7\_b1)  
  
tjx7\_b3 <- as.data.frame(as.factor(tjx7\_b2))  
colnames(tjx7\_b3) <- 'cSum'  
  
countTJX2 <- tjx7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countTJX2 <- as.data.frame(countTJX2)  
countTJX2 <- countTJX2[,-3]  
  
countTJX2$incr\_Days <- countTJX2$n-1  
countTJX2 <- countTJX2[,-2]  
  
countTJX3 <- subset(countTJX2, countTJX2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countTJX3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 4.319 7.000 32.000

mxtjx\_i <- max(countTJX3$incr\_Days)  
mdntjx\_i <- median(countTJX3$incr\_Days)  
q3tjx\_i <- as.numeric(as.character(summary(countTJX3$incr\_Days)["3rd Qu."]))  
  
  
tjx\_g <- grep('TJX',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[tjx\_g,8:13] <- c(mdntjx\_d,q3tjx\_d,mxtjx\_d,mdntjx\_i,q3tjx\_i,mxtjx\_i)

WMT, Walmart Stores:

WMT <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='WMT')  
  
#assign a 1 to increasing values  
wmt7<- ifelse(WMT$today2lag7>1, 1,0)  
  
wmt7\_a <- na.omit(wmt7)  
  
wmt7\_ab <- cumsum(wmt7\_a)  
  
wmt7\_abc <- as.data.frame(as.factor(wmt7\_ab))  
colnames(wmt7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countWMT <- wmt7\_abc %>% group\_by(cSum) %>% count(n=n())  
countWMT <- as.data.frame(countWMT)  
countWMT <- countWMT[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countWMT$decr\_Days <- countWMT$n-1  
countWMT <- countWMT[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countWMT1 <- subset(countWMT, countWMT$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countWMT1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.163 8.000 33.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxwmt\_d <- max(countWMT1$decr\_Days)  
mdnwmt\_d <- median(countWMT1$decr\_Days)  
q3wmt\_d <- as.numeric(as.character(summary(countWMT1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
wmt7\_b <- ifelse(WMT$today2lag7>1, 0,1)  
  
wmt7\_b1 <- na.omit(wmt7\_b)  
  
#counts the   
wmt7\_b2 <- cumsum(wmt7\_b1)  
  
wmt7\_b3 <- as.data.frame(as.factor(wmt7\_b2))  
colnames(wmt7\_b3) <- 'cSum'  
  
countWMT2 <- wmt7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countWMT2 <- as.data.frame(countWMT2)  
countWMT2 <- countWMT2[,-3]  
  
countWMT2$incr\_Days <- countWMT2$n-1  
countWMT2 <- countWMT2[,-2]  
  
countWMT3 <- subset(countWMT2, countWMT2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countWMT3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.061 7.000 31.000

mxwmt\_i <- max(countWMT3$incr\_Days)  
mdnwmt\_i <- median(countWMT3$incr\_Days)  
q3wmt\_i <- as.numeric(as.character(summary(countWMT3$incr\_Days)["3rd Qu."]))  
  
  
wmt\_g <- grep('WMT',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[wmt\_g,8:13] <- c(mdnwmt\_d,q3wmt\_d,mxwmt\_d,mdnwmt\_i,q3wmt\_i,mxwmt\_i)

XOM, Exxon Mobile gas:

XOM <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='XOM')  
  
#assign a 1 to increasing values  
xom7<- ifelse(XOM$today2lag7>1, 1,0)  
  
xom7\_a <- na.omit(xom7)  
  
xom7\_ab <- cumsum(xom7\_a)  
  
xom7\_abc <- as.data.frame(as.factor(xom7\_ab))  
colnames(xom7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countXOM <- xom7\_abc %>% group\_by(cSum) %>% count(n=n())  
countXOM <- as.data.frame(countXOM)  
countXOM <- countXOM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countXOM$decr\_Days <- countXOM$n-1  
countXOM <- countXOM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countXOM1 <- subset(countXOM, countXOM$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countXOM1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.839 8.000 49.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxxom\_d <- max(countXOM1$decr\_Days)  
mdnxom\_d <- median(countXOM1$decr\_Days)  
q3xom\_d <- as.numeric(as.character(summary(countXOM1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
xom7\_b <- ifelse(XOM$today2lag7>1, 0,1)  
  
xom7\_b1 <- na.omit(xom7\_b)  
  
#counts the   
xom7\_b2 <- cumsum(xom7\_b1)  
  
xom7\_b3 <- as.data.frame(as.factor(xom7\_b2))  
colnames(xom7\_b3) <- 'cSum'  
  
countXOM2 <- xom7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countXOM2 <- as.data.frame(countXOM2)  
countXOM2 <- countXOM2[,-3]  
  
countXOM2$incr\_Days <- countXOM2$n-1  
countXOM2 <- countXOM2[,-2]  
  
countXOM3 <- subset(countXOM2, countXOM2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countXOM3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.984 7.000 28.000

mxxom\_i <- max(countXOM3$incr\_Days)  
mdnxom\_i <- median(countXOM3$incr\_Days)  
q3xom\_i <- as.numeric(as.character(summary(countXOM3$incr\_Days)["3rd Qu."]))  
  
  
xom\_g <- grep('XOM',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[xom\_g,8:13] <- c(mdnxom\_d,q3xom\_d,mxxom\_d,mdnxom\_i,q3xom\_i,mxxom\_i)

AAP, Advance Auto Parts:

AAP <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='AAP')  
  
#assign a 1 to increasing values  
aap7<- ifelse(AAP$today2lag7>1, 1,0)  
  
aap7\_a <- na.omit(aap7)  
  
aap7\_ab <- cumsum(aap7\_a)  
  
aap7\_abc <- as.data.frame(as.factor(aap7\_ab))  
colnames(aap7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countAAP <- aap7\_abc %>% group\_by(cSum) %>% count(n=n())  
countAAP <- as.data.frame(countAAP)  
countAAP <- countAAP[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countAAP$decr\_Days <- countAAP$n-1  
countAAP <- countAAP[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countAAP1 <- subset(countAAP, countAAP$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countAAP1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 6.092 9.000 44.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxaap\_d <- max(countAAP1$decr\_Days)  
mdnaap\_d <- median(countAAP1$decr\_Days)  
q3aap\_d <- as.numeric(as.character(summary(countAAP1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
aap7\_b <- ifelse(AAP$today2lag7>1, 0,1)  
  
aap7\_b1 <- na.omit(aap7\_b)  
  
#counts the   
aap7\_b2 <- cumsum(aap7\_b1)  
  
aap7\_b3 <- as.data.frame(as.factor(aap7\_b2))  
colnames(aap7\_b3) <- 'cSum'  
  
countAAP2 <- aap7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countAAP2 <- as.data.frame(countAAP2)  
countAAP2 <- countAAP2[,-3]  
  
countAAP2$incr\_Days <- countAAP2$n-1  
countAAP2 <- countAAP2[,-2]  
  
countAAP3 <- subset(countAAP2, countAAP2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countAAP3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.154 7.000 30.000

mxaap\_i <- max(countAAP3$incr\_Days)  
mdnaap\_i <- median(countAAP3$incr\_Days)  
q3aap\_i <- as.numeric(as.character(summary(countAAP3$incr\_Days)["3rd Qu."]))  
  
  
aap\_g <- grep('AAP',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[aap\_g,8:13] <- c(mdnaap\_d,q3aap\_d,mxaap\_d,mdnaap\_i,q3aap\_i,mxaap\_i)

ADDYY, Adidas Sports Attire and shoes:

ADDYY <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='ADDYY')  
  
#assign a 1 to increasing values  
addyy7<- ifelse(ADDYY$today2lag7>1, 1,0)  
  
addyy7\_a <- na.omit(addyy7)  
  
addyy7\_ab <- cumsum(addyy7\_a)  
  
addyy7\_abc <- as.data.frame(as.factor(addyy7\_ab))  
colnames(addyy7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countADDYY <- addyy7\_abc %>% group\_by(cSum) %>% count(n=n())  
countADDYY <- as.data.frame(countADDYY)  
countADDYY <- countADDYY[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countADDYY$decr\_Days <- countADDYY$n-1  
countADDYY <- countADDYY[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countADDYY1 <- subset(countADDYY, countADDYY$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countADDYY1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.75 5.00 6.67 10.00 35.00

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxaddyy\_d <- max(countADDYY1$decr\_Days)  
mdnaddyy\_d <- median(countADDYY1$decr\_Days)  
q3addyy\_d <- as.numeric(as.character(summary(countADDYY1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
addyy7\_b <- ifelse(ADDYY$today2lag7>1, 0,1)  
  
addyy7\_b1 <- na.omit(addyy7\_b)  
  
#counts the   
addyy7\_b2 <- cumsum(addyy7\_b1)  
  
addyy7\_b3 <- as.data.frame(as.factor(addyy7\_b2))  
colnames(addyy7\_b3) <- 'cSum'  
  
countADDYY2 <- addyy7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countADDYY2 <- as.data.frame(countADDYY2)  
countADDYY2 <- countADDYY2[,-3]  
  
countADDYY2$incr\_Days <- countADDYY2$n-1  
countADDYY2 <- countADDYY2[,-2]  
  
countADDYY3 <- subset(countADDYY2, countADDYY2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countADDYY3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.249 8.000 25.000

mxaddyy\_i <- max(countADDYY3$incr\_Days)  
mdnaddyy\_i <- median(countADDYY3$incr\_Days)  
q3addyy\_i <- as.numeric(as.character(summary(countADDYY3$incr\_Days)["3rd Qu."]))  
  
  
addyy\_g <- grep('ADDYY',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[addyy\_g,8:13] <- c(mdnaddyy\_d,q3addyy\_d,mxaddyy\_d,mdnaddyy\_i,q3addyy\_i,mxaddyy\_i)

AMZN, Amazon:

AMZN <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='AMZN')  
  
#assign a 1 to increasing values  
amzn7<- ifelse(AMZN$today2lag7>1, 1,0)  
  
amzn7\_a <- na.omit(amzn7)  
  
amzn7\_ab <- cumsum(amzn7\_a)  
  
amzn7\_abc <- as.data.frame(as.factor(amzn7\_ab))  
colnames(amzn7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countAMZN <- amzn7\_abc %>% group\_by(cSum) %>% count(n=n())  
countAMZN <- as.data.frame(countAMZN)  
countAMZN <- countAMZN[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countAMZN$decr\_Days <- countAMZN$n-1  
countAMZN <- countAMZN[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countAMZN1 <- subset(countAMZN, countAMZN$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countAMZN1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 5.000 6.989 11.000 31.000

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxamzn\_d <- max(countAMZN1$decr\_Days)  
mdnamzn\_d <- median(countAMZN1$decr\_Days)  
q3amzn\_d <- as.numeric(as.character(summary(countAMZN1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
amzn7\_b <- ifelse(AMZN$today2lag7>1, 0,1)  
  
amzn7\_b1 <- na.omit(amzn7\_b)  
  
#counts the   
amzn7\_b2 <- cumsum(amzn7\_b1)  
  
amzn7\_b3 <- as.data.frame(as.factor(amzn7\_b2))  
colnames(amzn7\_b3) <- 'cSum'  
  
countAMZN2 <- amzn7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countAMZN2 <- as.data.frame(countAMZN2)  
countAMZN2 <- countAMZN2[,-3]  
  
countAMZN2$incr\_Days <- countAMZN2$n-1  
countAMZN2 <- countAMZN2[,-2]  
  
countAMZN3 <- subset(countAMZN2, countAMZN2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countAMZN3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.306 8.000 26.000

mxamzn\_i <- max(countAMZN3$incr\_Days)  
mdnamzn\_i <- median(countAMZN3$incr\_Days)  
q3amzn\_i <- as.numeric(as.character(summary(countAMZN3$incr\_Days)["3rd Qu."]))  
  
  
amzn\_g <- grep('AMZN',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[amzn\_g,8:13] <- c(mdnamzn\_d,q3amzn\_d,mxamzn\_d,mdnamzn\_i,q3amzn\_i,mxamzn\_i)

COST, Costco Stores:

COST <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='COST')  
  
#assign a 1 to increasing values  
cost7<- ifelse(COST$today2lag7>1, 1,0)  
  
cost7\_a <- na.omit(cost7)  
  
cost7\_ab <- cumsum(cost7\_a)  
  
cost7\_abc <- as.data.frame(as.factor(cost7\_ab))  
colnames(cost7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countCOST <- cost7\_abc %>% group\_by(cSum) %>% count(n=n())  
countCOST <- as.data.frame(countCOST)  
countCOST <- countCOST[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countCOST$decr\_Days <- countCOST$n-1  
countCOST <- countCOST[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countCOST1 <- subset(countCOST, countCOST$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countCOST1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 4.00 6.41 9.00 40.00

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxcost\_d <- max(countCOST1$decr\_Days)  
mdncost\_d <- median(countCOST1$decr\_Days)  
q3cost\_d <- as.numeric(as.character(summary(countCOST1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
cost7\_b <- ifelse(COST$today2lag7>1, 0,1)  
  
cost7\_b1 <- na.omit(cost7\_b)  
  
#counts the   
cost7\_b2 <- cumsum(cost7\_b1)  
  
cost7\_b3 <- as.data.frame(as.factor(cost7\_b2))  
colnames(cost7\_b3) <- 'cSum'  
  
countCOST2 <- cost7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countCOST2 <- as.data.frame(countCOST2)  
countCOST2 <- countCOST2[,-3]  
  
countCOST2$incr\_Days <- countCOST2$n-1  
countCOST2 <- countCOST2[,-2]  
  
countCOST3 <- subset(countCOST2, countCOST2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countCOST3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.776 7.000 45.000

mxcost\_i <- max(countCOST3$incr\_Days)  
mdncost\_i <- median(countCOST3$incr\_Days)  
q3cost\_i <- as.numeric(as.character(summary(countCOST3$incr\_Days)["3rd Qu."]))  
  
  
cost\_g <- grep('COST',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[cost\_g,8:13] <- c(mdncost\_d,q3cost\_d,mxcost\_d,mdncost\_i,q3cost\_i,mxcost\_i)

DLTR, Dollar Tree Stores:

DLTR <- subset(Close17\_dow, Close17\_dow$DOW\_down\_median=='DLTR')  
  
#assign a 1 to increasing values  
dltr7<- ifelse(DLTR$today2lag7>1, 1,0)  
  
dltr7\_a <- na.omit(dltr7)  
  
dltr7\_ab <- cumsum(dltr7\_a)  
  
dltr7\_abc <- as.data.frame(as.factor(dltr7\_ab))  
colnames(dltr7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countDLTR <- dltr7\_abc %>% group\_by(cSum) %>% count(n=n())  
countDLTR <- as.data.frame(countDLTR)  
countDLTR <- countDLTR[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countDLTR$decr\_Days <- countDLTR$n-1  
countDLTR <- countDLTR[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countDLTR1 <- subset(countDLTR, countDLTR$decr\_Days>0)  
print('Decreasing Consecutive Days');summary(countDLTR1$decr\_Days)

## [1] "Decreasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 1.0 4.0 6.5 9.0 33.0

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

mxdltr\_d <- max(countDLTR1$decr\_Days)  
mdndltr\_d <- median(countDLTR1$decr\_Days)  
q3dltr\_d <- as.numeric(as.character(summary(countDLTR1$decr\_Days)["3rd Qu."]))  
  
#assign a 1 to decreasing values  
dltr7\_b <- ifelse(DLTR$today2lag7>1, 0,1)  
  
dltr7\_b1 <- na.omit(dltr7\_b)  
  
#counts the   
dltr7\_b2 <- cumsum(dltr7\_b1)  
  
dltr7\_b3 <- as.data.frame(as.factor(dltr7\_b2))  
colnames(dltr7\_b3) <- 'cSum'  
  
countDLTR2 <- dltr7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countDLTR2 <- as.data.frame(countDLTR2)  
countDLTR2 <- countDLTR2[,-3]  
  
countDLTR2$incr\_Days <- countDLTR2$n-1  
countDLTR2 <- countDLTR2[,-2]  
  
countDLTR3 <- subset(countDLTR2, countDLTR2$incr\_Days>0)  
print('Increasing Consecutive Days');summary(countDLTR3$incr\_Days)

## [1] "Increasing Consecutive Days"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.941 7.000 38.000

mxdltr\_i <- max(countDLTR3$incr\_Days)  
mdndltr\_i <- median(countDLTR3$incr\_Days)  
q3dltr\_i <- as.numeric(as.character(summary(countDLTR3$incr\_Days)["3rd Qu."]))  
  
  
dltr\_g <- grep('DLTR',dROI17\_lag7$DOW\_down\_median)  
  
dROI17\_lag7[dltr\_g,8:13] <- c(mdndltr\_d,q3dltr\_d,mxdltr\_d,mdndltr\_i,q3dltr\_i,mxdltr\_i)

Now that we have our cumulative number of days in lag7 stock value prices that the stock decreases or increases in our best performing stock portfolio of 17 stocks that had positive median stock values when the DOW was down and unemployment up, lets add a field to classify whether the stock is low or high based on the Costco stock ROI of 6.02, which was the median stock ROI of this portfolio.

dROI17\_lag7$ROI\_Low\_High <- ifelse(dROI17\_lag7$stock\_ROI <= 6.02, 'Low', 'High')  
dROI17\_lag7

## # A tibble: 17 x 14  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 AAP 97.3 35.6 134. 2.36 3.75   
## 2 ADDYY 58.4 25 156. 2.36 6.24   
## 3 AMZN 551. 38.7 2135. 2.36 55.2   
## 4 COST 123. 52.8 318. 2.36 6.02   
## 5 DLTR 52.8 10.2 88.7 2.36 8.67   
## 6 FFIN 14.7 7.00 34.6 2.36 4.94   
## 7 GOOG 559. 233. 1521. 2.36 6.53   
## 8 HD 90.9 41.1 245. 2.36 5.97   
## 9 JNJ 91.1 66.4 150. 2.36 2.26   
## 10 NFLX 92.4 3.80 380. 2.36 100.   
## 11 NKE 39.7 12.2 104. 2.36 8.48   
## 12 PCG 44.1 47.3 16.2 2.36 0.343  
## 13 ROST 40.8 7.62 122. 2.36 16.0   
## 14 TEVA 41.1 31.3 12.2 2.36 0.391  
## 15 TJX 26.0 7.17 63.4 2.36 8.84   
## 16 WMT 69.8 47.5 118. 2.36 2.48   
## 17 XOM 81.4 74.1 60.7 2.36 0.818  
## # … with 8 more variables: stockBeatsDOW <chr>, medn\_cSum\_decr\_L7 <chr>,  
## # Q3\_cSum\_decr\_L7 <chr>, max\_cSum\_decr\_L7 <chr>, medn\_cSum\_incr\_L7 <chr>,  
## # Q3\_cSum\_incr\_L7 <chr>, max\_cSum\_incr\_L7 <chr>, ROI\_Low\_High <chr>

The table we extracted these stock is from the Close2 table.

colnames(Close2)

## [1] "MonthYear" "TGT.Close"   
## [3] "FTR.Close" "UBSI.Close"   
## [5] "HD.Close" "JPM.Close"   
## [7] "XOM.Close" "CVX.Close"   
## [9] "NSANY.Close" "MGM.Close"   
## [11] "TEVA.Close" "HST.Close"   
## [13] "WFC.Close" "WWE.Close"   
## [15] "INO.Close" "SCE.PB.Close"   
## [17] "FFIN.Close" "GOOG.Close"   
## [19] "WM.Close" "ONCY.Close"   
## [21] "S.Close" "F.Close"   
## [23] "ARWR.Close" "COST.Close"   
## [25] "AAL.Close" "JWN.Close"   
## [27] "NUS.Close" "ADDYY.Close"   
## [29] "KSS.Close" "MSFT.Close"   
## [31] "LUV.Close" "HMC.Close"   
## [33] "PCG.Close" "DLTR.Close"   
## [35] "KGJI.Close" "NKE.Close"   
## [37] "AMZN.Close" "ROST.Close"   
## [39] "WMT.Close" "TJX.Close"   
## [41] "TM.Close" "T.Close"   
## [43] "JNJ.Close" "C.Close"   
## [45] "EPD.Close" "VZ.Close"   
## [47] "HRB.Close" "NFLX.Close"   
## [49] "AAP.Close" "HOFT.Close"   
## [51] "SIG.Close" "RRGB.Close"   
## [53] "M.Close" "JBLU.Close"   
## [55] "portfolio\_DailyValue" "portfolio\_prevDay"   
## [57] "portfolio\_dailyValueChange" "portfolio\_ROI\_dollars"   
## [59] "Date" "DayOfWeek"   
## [61] "Month" "Year"   
## [63] "UE\_monthlyRate"

There are 36 more stock in our entire portfolio of stocks with values from 2007-2020 that we could also get these feature characteristics for. But for now lets just add a field to the table we derived these 17 stock from by stock closing value, take the overall return on investment for each of the remaining 36 stocks, and add in the ROI class of ‘Low’ or ‘High’ to that table. Then separate into training and testing sets and use machine learning to predict the outcome of the stock being low or high in ROI ratio of start value to final value of the stock according to the initial and final prices of each stock in our time series.

Lets keep only the date field and all the stocks in our Close2 table.

Close53 <- Close2[,-c(1,55:58,60:63)]  
  
colnames(Close53) <- gsub('.Close','', colnames(Close53))  
colnames(Close53) <- gsub('.PB', '', colnames(Close53))  
colnames(Close53)

## [1] "TGT" "FTR" "UBSI" "HD" "JPM" "XOM" "CVX" "NSANY" "MGM"   
## [10] "TEVA" "HST" "WFC" "WWE" "INO" "SCE" "FFIN" "GOOG" "WM"   
## [19] "ONCY" "S" "F" "ARWR" "COST" "AAL" "JWN" "NUS" "ADDYY"  
## [28] "KSS" "MSFT" "LUV" "HMC" "PCG" "DLTR" "KGJI" "NKE" "AMZN"   
## [37] "ROST" "WMT" "TJX" "TM" "T" "JNJ" "C" "EPD" "VZ"   
## [46] "HRB" "NFLX" "AAP" "HOFT" "SIG" "RRGB" "M" "JBLU" "Date"

Make a mini table of start and final values to get our ROI for each stock.

Close53\_roi <- subset(Close53, Close53$Date=='2007-01-03' |  
 Close53$Date=='2020-02-14')  
Close53\_roi\_t <- as.data.frame(t(Close53\_roi[1:53]))  
Close53\_roi\_t$stock\_ROI\_2007\_2020 <- round(Close53\_roi\_t$`2020-02-14`/Close53\_roi\_t$`2007-01-03`,2)  
Close53\_roi\_t$ROI\_Low\_High\_2007\_2020 <- ifelse(Close53\_roi\_t$stock\_ROI\_2007\_2020 <= 6.02, 'Low', 'High')  
Close53\_roi\_t$stockName <- as.factor(row.names(Close53\_roi\_t))  
Close53\_b <- Close53\_roi\_t[,-c(1:2)]  
Close53\_g <- gather(Close53, 'stockName','stockDayValue',1:53)  
Close53\_g$stockName <- as.factor(Close53\_g$stockName)  
  
Close53\_r <- merge(Close53\_g, Close53\_b, by.x='stockName', by.y='stockName')

We now have a table of the stock from 2007-2020, with the outcome class on the stock based on its daily value in this time period as ‘Low’ or ‘High’ ratio of start to final stock value. This measure or threshold was selected by picking the median ROI ratio of the best portfolio of stocks that included 17 of the stocks in our set. That threshold was 6.02 as the ROI of Costco from 2007-2020. Lets put together our data and make two separate data tables for the portfolio of best performing stock in the dROI17\_lag7 and the Close53\_r data tables. We can subset the Close53\_r table by date to see how well any given year of stock information can predict the stock ROI as being low or high.

Change the data types from char to numeric for the stat fields, as this wasn’t noticed earlier, and is easier to do this now, then go back to 17 separate coding areas of this script by backtracking.

dROI17\_lag7$medn\_cSum\_decr\_L7 <- as.numeric(dROI17\_lag7$medn\_cSum\_decr\_L7)  
dROI17\_lag7$Q3\_cSum\_decr\_L7 <- as.numeric(dROI17\_lag7$Q3\_cSum\_decr\_L7)  
dROI17\_lag7$max\_cSum\_decr\_L7 <- as.numeric(dROI17\_lag7$max\_cSum\_decr\_L7)  
dROI17\_lag7$medn\_cSum\_incr\_L7 <- as.numeric(dROI17\_lag7$medn\_cSum\_incr\_L7)  
dROI17\_lag7$Q3\_cSum\_incr\_L7 <- as.numeric(dROI17\_lag7$Q3\_cSum\_incr\_L7)  
dROI17\_lag7$max\_cSum\_incr\_L7 <- as.numeric(dROI17\_lag7$max\_cSum\_incr\_L7)  
  
dROI17\_lag7$DOW\_down\_median <- as.factor(dROI17\_lag7$DOW\_down\_median)  
dROI17\_lag7$ROI\_Low\_High <- as.factor(dROI17\_lag7$ROI\_Low\_High)

This is a mixed data table of numeric and factor features. Lets remove the stockBeatsDow feature.

dROI17\_lag7\_b <- dROI17\_lag7[,-7]  
dROI17\_lag7\_b

## # A tibble: 17 x 13  
## DOW\_down\_median avgStockValue startValue finalValue DOW\_ROI stock\_ROI  
## <fct> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 AAP 97.3 35.6 134. 2.36 3.75   
## 2 ADDYY 58.4 25 156. 2.36 6.24   
## 3 AMZN 551. 38.7 2135. 2.36 55.2   
## 4 COST 123. 52.8 318. 2.36 6.02   
## 5 DLTR 52.8 10.2 88.7 2.36 8.67   
## 6 FFIN 14.7 7.00 34.6 2.36 4.94   
## 7 GOOG 559. 233. 1521. 2.36 6.53   
## 8 HD 90.9 41.1 245. 2.36 5.97   
## 9 JNJ 91.1 66.4 150. 2.36 2.26   
## 10 NFLX 92.4 3.80 380. 2.36 100.   
## 11 NKE 39.7 12.2 104. 2.36 8.48   
## 12 PCG 44.1 47.3 16.2 2.36 0.343  
## 13 ROST 40.8 7.62 122. 2.36 16.0   
## 14 TEVA 41.1 31.3 12.2 2.36 0.391  
## 15 TJX 26.0 7.17 63.4 2.36 8.84   
## 16 WMT 69.8 47.5 118. 2.36 2.48   
## 17 XOM 81.4 74.1 60.7 2.36 0.818  
## # … with 7 more variables: medn\_cSum\_decr\_L7 <dbl>, Q3\_cSum\_decr\_L7 <dbl>,  
## # max\_cSum\_decr\_L7 <dbl>, medn\_cSum\_incr\_L7 <dbl>, Q3\_cSum\_incr\_L7 <dbl>,  
## # max\_cSum\_incr\_L7 <dbl>, ROI\_Low\_High <fct>

Lets make the row names of the dROI17\_lag7\_b table the DOW\_down\_median factor type feature, then remove that field from the table.

names <- as.character(dROI17\_lag7\_b$DOW\_down\_median)  
dROI17\_lag7\_c <- dROI17\_lag7\_b[,-1]  
row.names(dROI17\_lag7\_c) <- names   
  
dROI17\_lag7\_d <- round(dROI17\_lag7\_c[1:11],2)  
row.names(dROI17\_lag7\_d) <- names  
ROI\_17 <- cbind(dROI17\_lag7\_d, dROI17\_lag7\_c[12])  
ROI\_17

## avgStockValue startValue finalValue DOW\_ROI stock\_ROI medn\_cSum\_decr\_L7  
## AAP 97.31 35.58 133.59 2.36 3.75 3.0  
## ADDYY 58.44 25.00 156.11 2.36 6.24 5.0  
## AMZN 550.53 38.70 2134.87 2.36 55.16 5.0  
## COST 123.21 52.84 318.31 2.36 6.02 4.0  
## DLTR 52.83 10.23 88.68 2.36 8.67 4.0  
## FFIN 14.71 7.00 34.58 2.36 4.94 3.0  
## GOOG 558.98 232.92 1520.74 2.36 6.53 6.5  
## HD 90.93 41.07 245.03 2.36 5.97 4.0  
## JNJ 91.07 66.40 150.13 2.36 2.26 4.0  
## NFLX 92.39 3.80 380.40 2.36 100.07 5.5  
## NKE 39.66 12.21 103.54 2.36 8.48 4.0  
## PCG 44.14 47.27 16.20 2.36 0.34 3.0  
## ROST 40.77 7.62 121.78 2.36 15.98 4.0  
## TEVA 41.06 31.26 12.22 2.36 0.39 4.0  
## TJX 25.96 7.17 63.38 2.36 8.84 4.0  
## WMT 69.78 47.55 117.89 2.36 2.48 4.0  
## XOM 81.42 74.11 60.65 2.36 0.82 3.0  
## Q3\_cSum\_decr\_L7 max\_cSum\_decr\_L7 medn\_cSum\_incr\_L7 Q3\_cSum\_incr\_L7  
## AAP 9 44 3 7.00  
## ADDYY 10 35 3 8.00  
## AMZN 11 31 3 8.00  
## COST 9 40 3 7.00  
## DLTR 9 33 3 7.00  
## FFIN 8 38 3 7.00  
## GOOG 11 39 4 8.00  
## HD 10 30 3 8.00  
## JNJ 8 36 3 7.00  
## NFLX 10 37 4 7.75  
## NKE 9 40 3 7.00  
## PCG 7 34 3 7.00  
## ROST 9 32 2 7.00  
## TEVA 8 30 4 8.00  
## TJX 8 35 2 7.00  
## WMT 8 33 3 7.00  
## XOM 8 49 3 7.00  
## max\_cSum\_incr\_L7 ROI\_Low\_High  
## AAP 30 Low  
## ADDYY 25 High  
## AMZN 26 High  
## COST 45 High  
## DLTR 38 High  
## FFIN 26 Low  
## GOOG 28 High  
## HD 28 Low  
## JNJ 33 Low  
## NFLX 25 High  
## NKE 22 High  
## PCG 27 Low  
## ROST 28 High  
## TEVA 32 Low  
## TJX 32 High  
## WMT 31 Low  
## XOM 28 Low

write.csv(ROI\_17, 'ROI\_17\_ML.csv', row.names=TRUE)

We have the machine learning ready table from above for the 17 stocks belonging to the best portfolio of stocks. The other table of the low or high return values on table Close53\_r wouldn’t be much use to use machine learning on, since the name of the stock is already tied to the class, with the return on investment ratio. The individual stock lags, ratios of today’s stock value to the lag value, and the additional stock statistics for cumulative days of increasing and decreasing counts as well as the median, max, and 3rd quantile of each of those cumulative days by lag period to get a bigger picture. This will take twice as many lines of code as already has been produced for these 17, with the 36 other stock in the portfolio of stocks with values between 2007-2020.

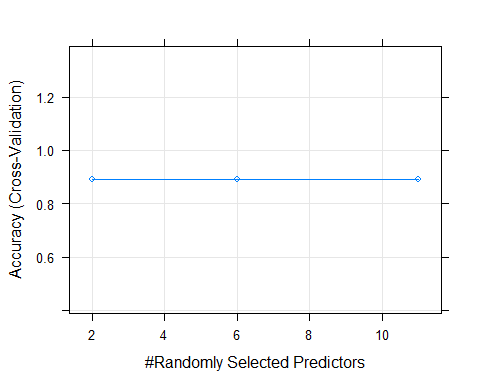
Lets get started with testing out some machine learning on this small subset of stocks to see how well it can predict a high or low return on investment from 70% training on the remaining 30% testing set of these stocks as samples. We will keep all the variables even if they are redundant and make it easier for the algorithm to predict the class.

set.seed(12356789)  
  
inTrain <- createDataPartition(y=ROI\_17$ROI\_Low\_High, p=0.7, list=FALSE)  
  
trainingSet <- ROI\_17[inTrain,]  
testingSet <- ROI\_17[-inTrain,]

rfMod <- train(ROI\_Low\_High~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

plot(rfMod)



predRF <- predict(rfMod, testingSet)  
  
predDF <- data.frame(predRF, type=testingSet$ROI\_Low\_High)  
predDF

## predRF type  
## 1 High High  
## 2 High Low  
## 3 High Low  
## 4 High High

sum <- sum(predRF==testingSet$ROI\_Low\_High)   
length <- length(testingSet$ROI\_Low\_High)  
accuracy\_rfMod <- (sum/length)   
accuracy\_rfMod

## [1] 0.5

results <- c(round(accuracy\_rfMod,2), round(100,2))  
results <- as.factor(results)  
results <- t(data.frame(results))  
  
colnames(results) <- colnames(predDF)  
Results <- rbind(predDF, results)   
Results

## predRF type  
## 1 High High  
## 2 High Low  
## 3 High Low  
## 4 High High  
## results 0.5 100

knnMod <- train(ROI\_Low\_High ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet)

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 13 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 15 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 17 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 19 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 21 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0990255204590788,  
## -0.369915578302799, : k = 23 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 13 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 15 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 17 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 19 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 21 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.145288423436488,  
## -0.407568225879971, : k = 23 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 13 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 15 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 17 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 19 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 21 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0577834480326509,  
## -0.32695863529271, : k = 23 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 13 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 15 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 17 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 19 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 21 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.10929086214704,  
## 0.0629363994586444, : k = 23 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 13 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 15 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 17 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 19 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 21 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.0954524073703742,  
## -0.352977557669156, : k = 23 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 13 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 15 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 17 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 19 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 21 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.106399664371088,  
## -0.378235450366873, : k = 23 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 13 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 15 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 17 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 19 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 21 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(0.928385821477331,  
## -0.492387741245252, : k = 23 exceeds number 11 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 13 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 15 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 17 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 19 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 21 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.076329367893291,  
## -0.345499023598641, : k = 23 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 13 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 15 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 17 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 19 exceeds number 12 of patterns

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 21 exceeds number 12 of patterns

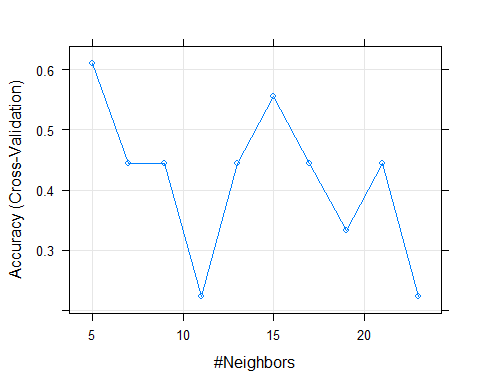
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

## Warning in knn3Train(train = structure(c(-0.341769049046496,  
## 0.106585425746993, : k = 23 exceeds number 12 of patterns

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut =  
## 10, : These variables have zero variances: DOW\_ROI

plot(knnMod)



rpartMod <- train(ROI\_Low\_High ~ ., method='rpart', tuneLength=7, data=trainingSet)

glmMod <- train(ROI\_Low\_High ~ .,   
 method='glm', data=trainingSet)

predKNN <- predict(knnMod, testingSet)  
predRPART <- predict(rpartMod, testingSet)  
predGLM <- predict(glmMod, testingSet)

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

length=length(testingSet$ROI\_Low\_High)  
  
sumKNN <- sum(predKNN==testingSet$ROI\_Low\_High)  
sumRPart <- sum(predRPART==testingSet$ROI\_Low\_High)  
sumGLM <- sum(predGLM==testingSet$ROI\_Low\_High)

accuracy\_KNN <- sumKNN/length   
accuracy\_RPART <- sumRPart/length   
accuracy\_GLM <- sumGLM/length

predDF2 <- data.frame(predRF,predKNN,predRPART,predGLM,   
 TYPE=testingSet$ROI\_Low\_High)  
colnames(predDF2) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
  
results <- c(round(accuracy\_rfMod,2),   
 round(accuracy\_KNN,2),   
 round(accuracy\_RPART,2),  
 round(accuracy\_GLM,2),   
 round(100,2))  
  
results <- as.factor(results)  
results <- t(data.frame(results))  
colnames(results) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
Results <- rbind(predDF2, results)   
Results

## RandomForest KNN Rpart GLM TrueValue  
## 1 High High High High High  
## 2 High Low High High Low  
## 3 High High High High Low  
## 4 High Low High Low High  
## results 0.5 0.5 0.5 0.25 100

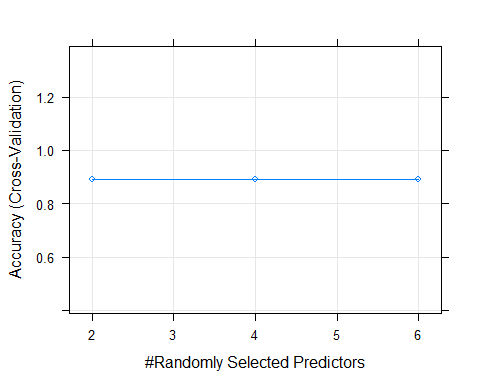
The above algorithms of machine learning using the default settings didn’t give very accurate predictions. There were only 13 samples to train on, and 4 to test on and the data fluctuates. There are just as many features as samples in the training set and more so in the testing set. What if we remove some of the features? Lets do this and remove the first five features.

set.seed(12356789)  
  
ROI\_17b <- ROI\_17[,-c(1:5)]  
inTrain <- createDataPartition(y=ROI\_17b$ROI\_Low\_High, p=0.7, list=FALSE)  
  
trainingSet <- ROI\_17b[inTrain,]  
testingSet <- ROI\_17b[-inTrain,]

rfMod <- train(ROI\_Low\_High~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5)

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

plot(rfMod)



predRF <- predict(rfMod, testingSet)  
  
predDF <- data.frame(predRF, type=testingSet$ROI\_Low\_High)  
predDF

## predRF type  
## 1 High High  
## 2 Low Low  
## 3 High Low  
## 4 Low High

sum <- sum(predRF==testingSet$ROI\_Low\_High)   
length <- length(testingSet$ROI\_Low\_High)  
accuracy\_rfMod <- (sum/length)   
accuracy\_rfMod

## [1] 0.5

results <- c(round(accuracy\_rfMod,2), round(100,2))  
results <- as.factor(results)  
results <- t(data.frame(results))  
  
colnames(results) <- colnames(predDF)  
Results <- rbind(predDF, results)   
Results

## predRF type  
## 1 High High  
## 2 Low Low  
## 3 High Low  
## 4 Low High  
## results 0.5 100

knnMod <- train(ROI\_Low\_High ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet)

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 13 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 15 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 17 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 19 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 21 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.19481902729849, 0.908062460746855, :  
## k = 23 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 13 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 15 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 17 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 19 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 21 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 23 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 13 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 15 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 17 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 19 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 21 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 23 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 13 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 15 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 17 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 19 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 21 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.10932123720135, 0.792372312286679, :  
## k = 23 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 13 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 15 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 17 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 19 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 21 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.23143859197392, 0.703679195413668, :  
## k = 23 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 13 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 15 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 17 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 19 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 21 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-1.2734971031749, 1.38423598171185, : k  
## = 23 exceeds number 12 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 13 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 15 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 17 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 19 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 21 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-0.180591048459026,  
## -0.180591048459026, : k = 23 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 13 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 15 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 17 exceeds number 11 of patterns

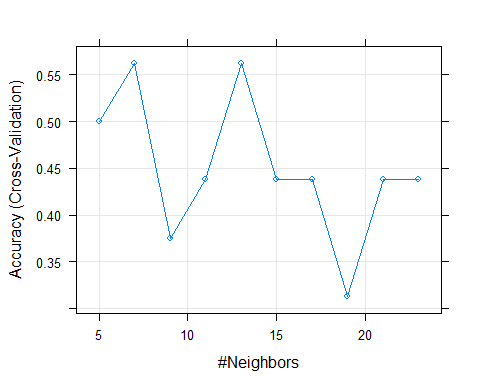
## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 19 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 21 exceeds number 11 of patterns

## Warning in knn3Train(train = structure(c(-1.07276939951792, 0.742686507358564, :  
## k = 23 exceeds number 11 of patterns

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
## There were missing values in resampled performance measures.

plot(knnMod)



rpartMod <- train(ROI\_Low\_High ~ ., method='rpart', tuneLength=7, data=trainingSet)

glmMod <- train(ROI\_Low\_High ~ .,   
 method='glm', data=trainingSet)

predKNN <- predict(knnMod, testingSet)  
predRPART <- predict(rpartMod, testingSet)  
predGLM <- predict(glmMod, testingSet)

length=length(testingSet$ROI\_Low\_High)  
  
sumKNN <- sum(predKNN==testingSet$ROI\_Low\_High)  
sumRPart <- sum(predRPART==testingSet$ROI\_Low\_High)  
sumGLM <- sum(predGLM==testingSet$ROI\_Low\_High)

accuracy\_KNN <- sumKNN/length   
accuracy\_RPART <- sumRPart/length   
accuracy\_GLM <- sumGLM/length

predDF2 <- data.frame(predRF,predKNN,predRPART,predGLM,   
 TYPE=testingSet$ROI\_Low\_High)  
colnames(predDF2) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
  
results <- c(round(accuracy\_rfMod,2),   
 round(accuracy\_KNN,2),   
 round(accuracy\_RPART,2),  
 round(accuracy\_GLM,2),   
 round(100,2))  
  
results <- as.factor(results)  
results <- t(data.frame(results))  
colnames(results) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
Results <- rbind(predDF2, results)   
Results

## RandomForest KNN Rpart GLM TrueValue  
## 1 High High High High High  
## 2 Low High High Low Low  
## 3 High High High Low Low  
## 4 Low High High High High  
## results 0.5 0.5 0.5 1 100

The feature removal was better in accuracy than the original data. Based on the number of times the ratio of today’s stock values by date to the value seven days ago as increasing, decreasing, the added statistics of max, median, and 3rd quantile all added more information and got rid of noise so that there was one model that scored 100% accuracy and the others only got half right. In the other set, with the average stock values, DOW ROI, and start and final values for the time span 2007-2020, there was one that scored 1/4 and the others got half right. In this case, the GLM or generalized linear model based on linear and logistic regression performed better than the others and got all predictions of the four samples right.

This could mean having these statistics is powerful in predicting the value of the stock as high or low, or that the GLM notices that there is information in the statistical lag information that was used to classify the stock as a high or low return, or both.

These questions can be answered if making an app where a person asks what stock to predict, the app pulls the data from the internet, for the specified time. The app then gathers the seven day lag information of the stock, gets the fold change of the daily stock values of the stock to the seven day lag, counts the number of times the stock has cumulative days of increasing and decreasing, gets the median value of that range, compares the number of days the stock was increasing last, compared with the last few days ratios to indicate what the set of increase or decrease count is, then returns whether to buy or sell, based on whether the set of days is reaching the median value of days(or 3rd quartile and/or beyond the previous max cumulative days increasing/decreasing) that increase or decrease for a short run profit, figuratively.

A way to see of Naive Bayes, a GLM model, algorithm to calculate the probabilities of the stock continuing to increase or decrease based on the previous days increase, possibly the day of the month, quarter of the year, and so on, that could return the best probability for continuing to decrease, or continuing to increase. The features would again have to be limited for the best results, but there could be algorithms for all of these questions inside a stock forecast app.

What remains is to add in the other 36 stock lag information and test this same series of machine learning default setting algorithms on that data to see if results improve. And do this for all the lag information of the 30-60-90-120-150-180 lags. Bunch more coding but the methods are above, with time consuming find and replace carefully done.

To test how well this works for any given quarter year, month, year, etc., these same steps would have to be done to smaller subsets. \*\*\*

Lets start by making a list of those stocks we already have lag values for.

colnames(Close53)

## [1] "TGT" "FTR" "UBSI" "HD" "JPM" "XOM" "CVX" "NSANY" "MGM"   
## [10] "TEVA" "HST" "WFC" "WWE" "INO" "SCE" "FFIN" "GOOG" "WM"   
## [19] "ONCY" "S" "F" "ARWR" "COST" "AAL" "JWN" "NUS" "ADDYY"  
## [28] "KSS" "MSFT" "LUV" "HMC" "PCG" "DLTR" "KGJI" "NKE" "AMZN"   
## [37] "ROST" "WMT" "TJX" "TM" "T" "JNJ" "C" "EPD" "VZ"   
## [46] "HRB" "NFLX" "AAP" "HOFT" "SIG" "RRGB" "M" "JBLU" "Date"

row.names(dROI17\_lag7\_d)

## [1] "AAP" "ADDYY" "AMZN" "COST" "DLTR" "FFIN" "GOOG" "HD" "JNJ"   
## [10] "NFLX" "NKE" "PCG" "ROST" "TEVA" "TJX" "WMT" "XOM"

The remaining stocks we need to add lagfields to, and then get the lag7 information for each stock as we did above on our machine learning data set of the 17 stocks in the best performing portfolio of stocks that had positive median daily value changes when the DOW was down and unemployment was up from the previous month.

remaining <- colnames(Close53)[c(1:3,5,7:9,11:15,18:22,24:26,28:31,34,40,41,43:46,49:53)]  
remaining

## [1] "TGT" "FTR" "UBSI" "JPM" "CVX" "NSANY" "MGM" "HST" "WFC"   
## [10] "WWE" "INO" "SCE" "WM" "ONCY" "S" "F" "ARWR" "AAL"   
## [19] "JWN" "NUS" "KSS" "MSFT" "LUV" "HMC" "KGJI" "TM" "T"   
## [28] "C" "EPD" "VZ" "HRB" "HOFT" "SIG" "RRGB" "M" "JBLU"

#this is the list of stocks that we need, and we need to drop the 17 we already analyzed  
Remaining <- as.data.frame(remaining)  
colnames(Remaining) <- 'stockName'  
  
#make all the stock name columns one field of stock names  
df36 <- gather(Close53, 'stockName','stockDayValue',1:53)  
  
#merge the names of the remaining stock and the total stocks to keep only the ones needed  
DF\_36 <- merge(Remaining,df36, by.x='stockName', by.y='stockName')  
  
#spread out the stock names into columns after getting only the stocks needed for adding stats  
#from the 53 total, and 17 completed.  
remaining36 <- spread(DF\_36,'stockName','stockDayValue')  
  
#order by date  
remaining36 <- remaining36[order(remaining36$Date),]

write.csv(remaining36,'remaining36.csv', row.names=FALSE)

Add the 7 day lag then add the ratio of each day’s stock value to the lag. Then get the counts of cumulative days the ratio increased and decreased. Then add the max, median, and 3rd quantile stats of those cumulative day counts for increased and decreased.

remaining36\_g <- gather(remaining36, 'stockName','stockDayValue',2:37)

remaining36\_g <- remaining36\_g[with(remaining36\_g, order(stockName, Date)),]  
  
Laal <- subset(remaining36\_g, remaining36\_g$stockName=='AAL')  
Larwr <- subset(remaining36\_g, remaining36\_g$stockName=='ARWR')  
Lc <- subset(remaining36\_g, remaining36\_g$stockName=='C')  
Lcvx <- subset(remaining36\_g, remaining36\_g$stockName=='CVX')  
Lepd <- subset(remaining36\_g, remaining36\_g$stockName=='EPD')  
Lf <- subset(remaining36\_g, remaining36\_g$stockName=='F')  
Lftr <- subset(remaining36\_g, remaining36\_g$stockName=='FTR')  
Lhmc <- subset(remaining36\_g, remaining36\_g$stockName=='HMC')  
Lhoft <- subset(remaining36\_g, remaining36\_g$stockName=='HOFT')  
Lhrb <- subset(remaining36\_g, remaining36\_g$stockName=='HRB')  
Lhst <- subset(remaining36\_g, remaining36\_g$stockName=='HST')  
Lino <- subset(remaining36\_g, remaining36\_g$stockName=='INO')  
Ljblu <- subset(remaining36\_g, remaining36\_g$stockName=='JBLU')  
Ljpm <- subset(remaining36\_g, remaining36\_g$stockName=='JPM')  
Ljwn <- subset(remaining36\_g, remaining36\_g$stockName=='JWN')  
Lkgji <- subset(remaining36\_g, remaining36\_g$stockName=='KGJI')  
Lkss <- subset(remaining36\_g, remaining36\_g$stockName=='KSS')  
Lluv <- subset(remaining36\_g, remaining36\_g$stockName=='LUV')  
Lm <- subset(remaining36\_g, remaining36\_g$stockName=='M')  
Lmgm <- subset(remaining36\_g, remaining36\_g$stockName=='MGM')  
Lmsft <- subset(remaining36\_g, remaining36\_g$stockName=='MSFT')  
Lnsany <- subset(remaining36\_g, remaining36\_g$stockName=='NSANY')  
Lnus <- subset(remaining36\_g, remaining36\_g$stockName=='NUS')  
Loncy <- subset(remaining36\_g, remaining36\_g$stockName=='ONCY')  
Lrrgb <- subset(remaining36\_g, remaining36\_g$stockName=='RRGB')  
Ls <- subset(remaining36\_g, remaining36\_g$stockName=='S')  
Lsce <- subset(remaining36\_g, remaining36\_g$stockName=='SCE')  
Lsig <- subset(remaining36\_g, remaining36\_g$stockName=='SIG')  
Lt <- subset(remaining36\_g, remaining36\_g$stockName=='T')  
Ltgt <- subset(remaining36\_g, remaining36\_g$stockName=='TGT')  
Ltm <- subset(remaining36\_g, remaining36\_g$stockName=='TM')  
Lubsi <- subset(remaining36\_g, remaining36\_g$stockName=='UBSI')  
Lvz <- subset(remaining36\_g, remaining36\_g$stockName=='VZ')  
Lwfc <- subset(remaining36\_g, remaining36\_g$stockName=='WFC')  
Lwm <- subset(remaining36\_g, remaining36\_g$stockName=='WM')  
Lwwe <- subset(remaining36\_g, remaining36\_g$stockName=='WWE')  
  
aalL7 <- lag(Laal$stockDayValue,7)  
arwrL7 <- lag(Larwr$stockDayValue,7)  
cL7 <- lag(Lc$stockDayValue,7)  
cvxL7 <- lag(Lcvx$stockDayValue,7)  
epdL7 <- lag(Lepd$stockDayValue,7)  
fL7 <- lag(Lf$stockDayValue,7)  
ftrL7 <- lag(Lftr$stockDayValue,7)  
hmcL7 <- lag(Lhmc$stockDayValue,7)  
hoftL7 <- lag(Lhoft$stockDayValue,7)  
hrbL7 <- lag(Lhrb$stockDayValue,7)  
hstL7 <- lag(Lhst$stockDayValue,7)  
inoL7 <- lag(Lino$stockDayValue,7)  
jbluL7 <- lag(Ljblu$stockDayValue,7)  
jpmL7 <- lag(Ljpm$stockDayValue,7)  
jwnL7 <- lag(Ljwn$stockDayValue,7)  
kgjiL7 <- lag(Lkgji$stockDayValue,7)  
kssL7 <- lag(Lkss$stockDayValue,7)  
luvL7 <- lag(Lluv$stockDayValue,7)  
mL7 <- lag(Lm$stockDayValue,7)  
mgmL7 <- lag(Lmgm$stockDayValue,7)  
msftL7 <- lag(Lmsft$stockDayValue,7)  
nsanyL7 <- lag(Lnsany$stockDayValue,7)  
nusL7 <- lag(Lnus$stockDayValue,7)  
oncyL7 <- lag(Loncy$stockDayValue,7)  
rrgbL7 <- lag(Lrrgb$stockDayValue,7)  
sL7 <- lag(Ls$stockDayValue,7)  
sceL7 <- lag(Lsce$stockDayValue,7)  
sigL7 <- lag(Lsig$stockDayValue,7)  
tL7 <- lag(Lt$stockDayValue,7)  
tgtL7 <- lag(Ltgt$stockDayValue,7)  
tmL7 <- lag(Ltm$stockDayValue,7)  
ubsiL7 <- lag(Lubsi$stockDayValue,7)  
vzL7 <- lag(Lvz$stockDayValue,7)  
wfcL7 <- lag(Lwfc$stockDayValue,7)  
wmL7 <- lag(Lwm$stockDayValue,7)  
wweL7 <- lag(Lwwe$stockDayValue,7)  
  
remaining36\_g$lag7 <- c(aalL7,arwrL7,cL7,cvxL7,epdL7,fL7,ftrL7,  
 hmcL7,hoftL7,hrbL7,hstL7,inoL7,jbluL7,jpmL7,  
 jwnL7,kgjiL7,kssL7,luvL7,mL7,mgmL7,msftL7,nsanyL7,nusL7,oncyL7,  
 rrgbL7,sL7,sceL7,sigL7,tL7,tgtL7,tmL7,  
 ubsiL7,vzL7,wfcL7,wmL7,wweL7)  
#the dataset is ordered by stock then by date, so this works with matching,  
#NAs are present when no data in dataset before that date, and the today2lag7  
#will also have NAs in those instances  
remaining36\_g$today2lag7 <- remaining36\_g$lag7/remaining36\_g$stockDayValue

AAL <- subset(remaining36\_g, remaining36\_g$stockName=='AAL')  
  
AAL <- AAL[complete.cases(AAL),]  
  
aal7\_a <- ifelse(AAL$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for aal7 <- <- ifelse(AAL$today2lag7>1, 1,0)  
aal7\_ab <- cumsum(aal7\_a)  
  
aal7\_abc <- as.data.frame(as.factor(aal7\_ab))  
colnames(aal7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countAAL <- aal7\_abc %>% group\_by(cSum) %>% count(n=n())  
countAAL <- as.data.frame(countAAL)  
countAAL <- countAAL[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countAAL$aal7\_decr\_Days <- countAAL$n-1  
countAAL <- countAAL[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countAAL2 <- subset(countAAL, countAAL$aal7\_decr\_Days>0)  
summary(countAAL2$aal7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.622 10.000 32.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
aal7\_decr\_Days\_grouped <- countAAL2 %>% group\_by(aal7\_decr\_Days) %>% count(n=n())  
aal7\_decr\_Days\_grouped <- aal7\_decr\_Days\_grouped[,-3]

aal7\_decr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: aal7\_decr\_Days [28]  
## aal7\_decr\_Days n  
## <dbl> <int>  
## 1 1 71  
## 2 2 28  
## 3 3 20  
## 4 4 12  
## 5 5 9  
## 6 6 8  
## 7 7 21  
## 8 8 9  
## 9 9 9  
## 10 10 8  
## # … with 18 more rows

median(aal7\_decr\_Days\_grouped$aal7\_decr\_Days)

## [1] 14.5

#assign a 1 to decreasing values  
aal7\_b1 <- ifelse(AAL$today2lag7>1, 0,1)  
  
  
aal7\_b2 <- cumsum(aal7\_b1)  
  
aal7\_b3 <- as.data.frame(as.factor(aal7\_b2))  
colnames(aal7\_b3) <- 'cSum'  
  
countAAL1 <- aal7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countAAL1 <- as.data.frame(countAAL1)  
countAAL1 <- countAAL1[,-3]  
  
countAAL1$aal7\_incr\_Days <- countAAL1$n-1  
countAAL1 <- countAAL1[,-2]  
  
countAAL3 <- subset(countAAL1, countAAL1$aal7\_incr\_Days>0)  
summary(countAAL3$aal7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 5.000 6.375 9.000 30.000

aal7\_incr\_Days\_grouped <- countAAL3 %>% group\_by(aal7\_incr\_Days) %>% count(n=n())  
aal7\_incr\_Days\_grouped <- aal7\_incr\_Days\_grouped[,-3]

aal7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: aal7\_incr\_Days [23]  
## aal7\_incr\_Days n  
## <dbl> <int>  
## 1 1 62  
## 2 2 31  
## 3 3 23  
## 4 4 7  
## 5 5 13  
## 6 6 7  
## 7 7 25  
## 8 8 12  
## 9 9 14  
## 10 10 14  
## # … with 13 more rows

median(aal7\_incr\_Days\_grouped$aal7\_incr\_Days)

## [1] 12

ARWR <- subset(remaining36\_g, remaining36\_g$stockName=='ARWR')  
  
ARWR <- ARWR[complete.cases(ARWR),]  
  
arwr7\_a <- ifelse(ARWR$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for arwr7 <- <- ifelse(ARWR$today2lag7>1, 1,0)  
arwr7\_ab <- cumsum(arwr7\_a)  
  
arwr7\_abc <- as.data.frame(as.factor(arwr7\_ab))  
colnames(arwr7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countARWR <- arwr7\_abc %>% group\_by(cSum) %>% count(n=n())  
countARWR <- as.data.frame(countARWR)  
countARWR <- countARWR[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countARWR$arwr7\_decr\_Days <- countARWR$n-1  
countARWR <- countARWR[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countARWR2 <- subset(countARWR, countARWR$arwr7\_decr\_Days>0)  
summary(countARWR2$arwr7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.974 8.000 33.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
arwr7\_decr\_Days\_grouped <- countARWR2 %>% group\_by(arwr7\_decr\_Days) %>% count(n=n())  
arwr7\_decr\_Days\_grouped <- arwr7\_decr\_Days\_grouped[,-3]

arwr7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: arwr7\_decr\_Days [24]  
## arwr7\_decr\_Days n  
## <dbl> <int>  
## 1 1 74  
## 2 2 35  
## 3 3 25  
## 4 4 19  
## 5 5 7  
## 6 6 12  
## 7 7 19  
## 8 8 14  
## 9 9 14  
## 10 10 7  
## # … with 14 more rows

median(arwr7\_decr\_Days\_grouped$arwr7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
arwr7\_b1 <- ifelse(ARWR$today2lag7>1, 0,1)  
  
  
arwr7\_b2 <- cumsum(arwr7\_b1)  
  
arwr7\_b3 <- as.data.frame(as.factor(arwr7\_b2))  
colnames(arwr7\_b3) <- 'cSum'  
  
countARWR1 <- arwr7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countARWR1 <- as.data.frame(countARWR1)  
countARWR1 <- countARWR1[,-3]  
  
countARWR1$arwr7\_incr\_Days <- countARWR1$n-1  
countARWR1 <- countARWR1[,-2]  
  
countARWR3 <- subset(countARWR1, countARWR1$arwr7\_incr\_Days>0)  
summary(countARWR3$arwr7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.117 9.000 40.000

arwr7\_incr\_Days\_grouped <- countARWR3 %>% group\_by(arwr7\_incr\_Days) %>% count(n=n())  
arwr7\_incr\_Days\_grouped <- arwr7\_incr\_Days\_grouped[,-3]

arwr7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: arwr7\_incr\_Days [24]  
## arwr7\_incr\_Days n  
## <dbl> <int>  
## 1 1 70  
## 2 2 32  
## 3 3 24  
## 4 4 20  
## 5 5 14  
## 6 6 6  
## 7 7 17  
## 8 8 17  
## 9 9 17  
## 10 10 4  
## # … with 14 more rows

median(arwr7\_incr\_Days\_grouped$arwr7\_incr\_Days)

## [1] 12.5

C <- subset(remaining36\_g, remaining36\_g$stockName=='C')  
  
C <- C[complete.cases(C),]  
  
c7\_a <- ifelse(C$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for c7 <- <- ifelse(C$today2lag7>1, 1,0)  
c7\_ab <- cumsum(c7\_a)  
  
c7\_abc <- as.data.frame(as.factor(c7\_ab))  
colnames(c7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countC <- c7\_abc %>% group\_by(cSum) %>% count(n=n())  
countC <- as.data.frame(countC)  
countC <- countC[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countC$c7\_decr\_Days <- countC$n-1  
countC <- countC[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countC2 <- subset(countC, countC$c7\_decr\_Days>0)  
summary(countC2$c7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.916 8.000 26.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
c7\_decr\_Days\_grouped <- countC2 %>% group\_by(c7\_decr\_Days) %>% count(n=n())  
c7\_decr\_Days\_grouped <- c7\_decr\_Days\_grouped[,-3]

c7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: c7\_decr\_Days [24]  
## c7\_decr\_Days n  
## <dbl> <int>  
## 1 1 87  
## 2 2 28  
## 3 3 15  
## 4 4 19  
## 5 5 12  
## 6 6 11  
## 7 7 25  
## 8 8 17  
## 9 9 11  
## 10 10 5  
## # … with 14 more rows

median(c7\_decr\_Days\_grouped$c7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
c7\_b1 <- ifelse(C$today2lag7>1, 0,1)  
  
  
c7\_b2 <- cumsum(c7\_b1)  
  
c7\_b3 <- as.data.frame(as.factor(c7\_b2))  
colnames(c7\_b3) <- 'cSum'  
  
countC1 <- c7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countC1 <- as.data.frame(countC1)  
countC1 <- countC1[,-3]  
  
countC1$c7\_incr\_Days <- countC1$n-1  
countC1 <- countC1[,-2]  
  
countC3 <- subset(countC1, countC1$c7\_incr\_Days>0)  
summary(countC3$c7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.626 8.000 27.000

c7\_incr\_Days\_grouped <- countC3 %>% group\_by(c7\_incr\_Days) %>% count(n=n())  
c7\_incr\_Days\_grouped <- c7\_incr\_Days\_grouped[,-3]

c7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: c7\_incr\_Days [23]  
## c7\_incr\_Days n  
## <dbl> <int>  
## 1 1 88  
## 2 2 36  
## 3 3 18  
## 4 4 21  
## 5 5 10  
## 6 6 8  
## 7 7 29  
## 8 8 13  
## 9 9 8  
## 10 10 7  
## # … with 13 more rows

median(c7\_incr\_Days\_grouped$c7\_incr\_Days)

## [1] 12

CVX <- subset(remaining36\_g, remaining36\_g$stockName=='CVX')  
  
CVX <- CVX[complete.cases(CVX),]  
  
cvx7\_a <- ifelse(CVX$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for cvx7 <- <- ifelse(CVX$today2lag7>1, 1,0)  
cvx7\_ab <- cumsum(cvx7\_a)  
  
cvx7\_abc <- as.data.frame(as.factor(cvx7\_ab))  
colnames(cvx7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countCVX <- cvx7\_abc %>% group\_by(cSum) %>% count(n=n())  
countCVX <- as.data.frame(countCVX)  
countCVX <- countCVX[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countCVX$cvx7\_decr\_Days <- countCVX$n-1  
countCVX <- countCVX[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countCVX2 <- subset(countCVX, countCVX$cvx7\_decr\_Days>0)  
summary(countCVX2$cvx7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.867 8.000 42.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
cvx7\_decr\_Days\_grouped <- countCVX2 %>% group\_by(cvx7\_decr\_Days) %>% count(n=n())  
cvx7\_decr\_Days\_grouped <- cvx7\_decr\_Days\_grouped[,-3]

cvx7\_decr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: cvx7\_decr\_Days [27]  
## cvx7\_decr\_Days n  
## <dbl> <int>  
## 1 1 100  
## 2 2 36  
## 3 3 23  
## 4 4 17  
## 5 5 10  
## 6 6 12  
## 7 7 17  
## 8 8 17  
## 9 9 7  
## 10 10 9  
## # … with 17 more rows

median(cvx7\_decr\_Days\_grouped$cvx7\_decr\_Days)

## [1] 14

#assign a 1 to decreasing values  
cvx7\_b1 <- ifelse(CVX$today2lag7>1, 0,1)  
  
  
cvx7\_b2 <- cumsum(cvx7\_b1)  
  
cvx7\_b3 <- as.data.frame(as.factor(cvx7\_b2))  
colnames(cvx7\_b3) <- 'cSum'  
  
countCVX1 <- cvx7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countCVX1 <- as.data.frame(countCVX1)  
countCVX1 <- countCVX1[,-3]  
  
countCVX1$cvx7\_incr\_Days <- countCVX1$n-1  
countCVX1 <- countCVX1[,-2]  
  
countCVX3 <- subset(countCVX1, countCVX1$cvx7\_incr\_Days>0)  
summary(countCVX3$cvx7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.063 8.000 25.000

cvx7\_incr\_Days\_grouped <- countCVX3 %>% group\_by(cvx7\_incr\_Days) %>% count(n=n())  
cvx7\_incr\_Days\_grouped <- cvx7\_incr\_Days\_grouped[,-3]

cvx7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: cvx7\_incr\_Days [23]  
## cvx7\_incr\_Days n  
## <dbl> <int>  
## 1 1 101  
## 2 2 43  
## 3 3 25  
## 4 4 18  
## 5 5 10  
## 6 6 7  
## 7 7 17  
## 8 8 12  
## 9 9 17  
## 10 10 10  
## # … with 13 more rows

median(cvx7\_incr\_Days\_grouped$cvx7\_incr\_Days)

## [1] 12

EPD <- subset(remaining36\_g, remaining36\_g$stockName=='EPD')  
  
EPD <- EPD[complete.cases(EPD),]  
  
epd7\_a <- ifelse(EPD$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for epd7 <- <- ifelse(EPD$today2lag7>1, 1,0)  
epd7\_ab <- cumsum(epd7\_a)  
  
epd7\_abc <- as.data.frame(as.factor(epd7\_ab))  
colnames(epd7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countEPD <- epd7\_abc %>% group\_by(cSum) %>% count(n=n())  
countEPD <- as.data.frame(countEPD)  
countEPD <- countEPD[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countEPD$epd7\_decr\_Days <- countEPD$n-1  
countEPD <- countEPD[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countEPD2 <- subset(countEPD, countEPD$epd7\_decr\_Days>0)  
summary(countEPD2$epd7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 1.0 5.0 6.2 8.0 42.0

# this table shows how many sets of cumulative days decreasing there were in lag 7  
epd7\_decr\_Days\_grouped <- countEPD2 %>% group\_by(epd7\_decr\_Days) %>% count(n=n())  
epd7\_decr\_Days\_grouped <- epd7\_decr\_Days\_grouped[,-3]

epd7\_decr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: epd7\_decr\_Days [27]  
## epd7\_decr\_Days n  
## <dbl> <int>  
## 1 1 79  
## 2 2 29  
## 3 3 20  
## 4 4 13  
## 5 5 16  
## 6 6 18  
## 7 7 24  
## 8 8 15  
## 9 9 10  
## 10 10 15  
## # … with 17 more rows

median(epd7\_decr\_Days\_grouped$epd7\_decr\_Days)

## [1] 14

#assign a 1 to decreasing values  
epd7\_b1 <- ifelse(EPD$today2lag7>1, 0,1)  
  
  
epd7\_b2 <- cumsum(epd7\_b1)  
  
epd7\_b3 <- as.data.frame(as.factor(epd7\_b2))  
colnames(epd7\_b3) <- 'cSum'  
  
countEPD1 <- epd7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countEPD1 <- as.data.frame(countEPD1)  
countEPD1 <- countEPD1[,-3]  
  
countEPD1$epd7\_incr\_Days <- countEPD1$n-1  
countEPD1 <- countEPD1[,-2]  
  
countEPD3 <- subset(countEPD1, countEPD1$epd7\_incr\_Days>0)  
summary(countEPD3$epd7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.361 8.000 26.000

epd7\_incr\_Days\_grouped <- countEPD3 %>% group\_by(epd7\_incr\_Days) %>% count(n=n())  
epd7\_incr\_Days\_grouped <- epd7\_incr\_Days\_grouped[,-3]

epd7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: epd7\_incr\_Days [22]  
## epd7\_incr\_Days n  
## <dbl> <int>  
## 1 1 86  
## 2 2 42  
## 3 3 13  
## 4 4 15  
## 5 5 20  
## 6 6 14  
## 7 7 21  
## 8 8 11  
## 9 9 14  
## 10 10 5  
## # … with 12 more rows

median(epd7\_incr\_Days\_grouped$epd7\_incr\_Days)

## [1] 11.5

F <- subset(remaining36\_g, remaining36\_g$stockName=='F')  
  
F <- F[complete.cases(F),]  
  
f7\_a <- ifelse(F$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for f7 <- <- ifelse(F$today2lag7>1, 1,0)  
f7\_ab <- cumsum(f7\_a)  
  
f7\_abc <- as.data.frame(as.factor(f7\_ab))  
colnames(f7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countF <- f7\_abc %>% group\_by(cSum) %>% count(n=n())  
countF <- as.data.frame(countF)  
countF <- countF[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countF$f7\_decr\_Days <- countF$n-1  
countF <- countF[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countF2 <- subset(countF, countF$f7\_decr\_Days>0)  
summary(countF2$f7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.139 9.000 33.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
f7\_decr\_Days\_grouped <- countF2 %>% group\_by(f7\_decr\_Days) %>% count(n=n())  
f7\_decr\_Days\_grouped <- f7\_decr\_Days\_grouped[,-3]

f7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: f7\_decr\_Days [26]  
## f7\_decr\_Days n  
## <dbl> <int>  
## 1 1 73  
## 2 2 39  
## 3 3 17  
## 4 4 15  
## 5 5 13  
## 6 6 8  
## 7 7 21  
## 8 8 10  
## 9 9 12  
## 10 10 8  
## # … with 16 more rows

median(f7\_decr\_Days\_grouped$f7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
f7\_b1 <- ifelse(F$today2lag7>1, 0,1)  
  
  
f7\_b2 <- cumsum(f7\_b1)  
  
f7\_b3 <- as.data.frame(as.factor(f7\_b2))  
colnames(f7\_b3) <- 'cSum'  
  
countF1 <- f7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countF1 <- as.data.frame(countF1)  
countF1 <- countF1[,-3]  
  
countF1$f7\_incr\_Days <- countF1$n-1  
countF1 <- countF1[,-2]  
  
countF3 <- subset(countF1, countF1$f7\_incr\_Days>0)  
summary(countF3$f7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.500 5.000 6.202 9.000 32.000

f7\_incr\_Days\_grouped <- countF3 %>% group\_by(f7\_incr\_Days) %>% count(n=n())  
f7\_incr\_Days\_grouped <- f7\_incr\_Days\_grouped[,-3]

f7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: f7\_incr\_Days [24]  
## f7\_incr\_Days n  
## <dbl> <int>  
## 1 1 67  
## 2 2 27  
## 3 3 18  
## 4 4 13  
## 5 5 13  
## 6 6 10  
## 7 7 25  
## 8 8 21  
## 9 9 14  
## 10 10 10  
## # … with 14 more rows

median(f7\_incr\_Days\_grouped$f7\_incr\_Days)

## [1] 12.5

FTR <- subset(remaining36\_g, remaining36\_g$stockName=='FTR')  
  
FTR <- FTR[complete.cases(FTR),]  
  
ftr7\_a <- ifelse(FTR$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for ftr7 <- <- ifelse(FTR$today2lag7>1, 1,0)  
ftr7\_ab <- cumsum(ftr7\_a)  
  
ftr7\_abc <- as.data.frame(as.factor(ftr7\_ab))  
colnames(ftr7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countFTR <- ftr7\_abc %>% group\_by(cSum) %>% count(n=n())  
countFTR <- as.data.frame(countFTR)  
countFTR <- countFTR[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countFTR$ftr7\_decr\_Days <- countFTR$n-1  
countFTR <- countFTR[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countFTR2 <- subset(countFTR, countFTR$ftr7\_decr\_Days>0)  
summary(countFTR2$ftr7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 1.0 4.0 5.4 8.0 28.0

# this table shows how many sets of cumulative days decreasing there were in lag 7  
ftr7\_decr\_Days\_grouped <- countFTR2 %>% group\_by(ftr7\_decr\_Days) %>% count(n=n())  
ftr7\_decr\_Days\_grouped <- ftr7\_decr\_Days\_grouped[,-3]

ftr7\_decr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: ftr7\_decr\_Days [22]  
## ftr7\_decr\_Days n  
## <dbl> <int>  
## 1 1 79  
## 2 2 37  
## 3 3 27  
## 4 4 22  
## 5 5 14  
## 6 6 9  
## 7 7 22  
## 8 8 15  
## 9 9 11  
## 10 10 10  
## # … with 12 more rows

median(ftr7\_decr\_Days\_grouped$ftr7\_decr\_Days)

## [1] 11.5

#assign a 1 to decreasing values  
ftr7\_b1 <- ifelse(FTR$today2lag7>1, 0,1)  
  
  
ftr7\_b2 <- cumsum(ftr7\_b1)  
  
ftr7\_b3 <- as.data.frame(as.factor(ftr7\_b2))  
colnames(ftr7\_b3) <- 'cSum'  
  
countFTR1 <- ftr7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countFTR1 <- as.data.frame(countFTR1)  
countFTR1 <- countFTR1[,-3]  
  
countFTR1$ftr7\_incr\_Days <- countFTR1$n-1  
countFTR1 <- countFTR1[,-2]  
  
countFTR3 <- subset(countFTR1, countFTR1$ftr7\_incr\_Days>0)  
summary(countFTR3$ftr7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.983 9.000 25.000

ftr7\_incr\_Days\_grouped <- countFTR3 %>% group\_by(ftr7\_incr\_Days) %>% count(n=n())  
ftr7\_incr\_Days\_grouped <- ftr7\_incr\_Days\_grouped[,-3]

ftr7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: ftr7\_incr\_Days [23]  
## ftr7\_incr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 42  
## 3 3 18  
## 4 4 11  
## 5 5 13  
## 6 6 7  
## 7 7 15  
## 8 8 14  
## 9 9 19  
## 10 10 9  
## # … with 13 more rows

median(ftr7\_incr\_Days\_grouped$ftr7\_incr\_Days)

## [1] 12

HMC <- subset(remaining36\_g, remaining36\_g$stockName=='HMC')  
  
HMC <- HMC[complete.cases(HMC),]  
  
hmc7\_a <- ifelse(HMC$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for hmc7 <- <- ifelse(HMC$today2lag7>1, 1,0)  
hmc7\_ab <- cumsum(hmc7\_a)  
  
hmc7\_abc <- as.data.frame(as.factor(hmc7\_ab))  
colnames(hmc7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countHMC <- hmc7\_abc %>% group\_by(cSum) %>% count(n=n())  
countHMC <- as.data.frame(countHMC)  
countHMC <- countHMC[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countHMC$hmc7\_decr\_Days <- countHMC$n-1  
countHMC <- countHMC[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countHMC2 <- subset(countHMC, countHMC$hmc7\_decr\_Days>0)  
summary(countHMC2$hmc7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.819 8.000 37.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
hmc7\_decr\_Days\_grouped <- countHMC2 %>% group\_by(hmc7\_decr\_Days) %>% count(n=n())  
hmc7\_decr\_Days\_grouped <- hmc7\_decr\_Days\_grouped[,-3]

hmc7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: hmc7\_decr\_Days [26]  
## hmc7\_decr\_Days n  
## <dbl> <int>  
## 1 1 73  
## 2 2 38  
## 3 3 17  
## 4 4 18  
## 5 5 17  
## 6 6 11  
## 7 7 27  
## 8 8 14  
## 9 9 14  
## 10 10 16  
## # … with 16 more rows

median(hmc7\_decr\_Days\_grouped$hmc7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
hmc7\_b1 <- ifelse(HMC$today2lag7>1, 0,1)  
  
  
hmc7\_b2 <- cumsum(hmc7\_b1)  
  
hmc7\_b3 <- as.data.frame(as.factor(hmc7\_b2))  
colnames(hmc7\_b3) <- 'cSum'  
  
countHMC1 <- hmc7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countHMC1 <- as.data.frame(countHMC1)  
countHMC1 <- countHMC1[,-3]  
  
countHMC1$hmc7\_incr\_Days <- countHMC1$n-1  
countHMC1 <- countHMC1[,-2]  
  
countHMC3 <- subset(countHMC1, countHMC1$hmc7\_incr\_Days>0)  
summary(countHMC3$hmc7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.865 8.750 31.000

hmc7\_incr\_Days\_grouped <- countHMC3 %>% group\_by(hmc7\_incr\_Days) %>% count(n=n())  
hmc7\_incr\_Days\_grouped <- hmc7\_incr\_Days\_grouped[,-3]

hmc7\_incr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: hmc7\_incr\_Days [27]  
## hmc7\_incr\_Days n  
## <dbl> <int>  
## 1 1 86  
## 2 2 28  
## 3 3 25  
## 4 4 16  
## 5 5 10  
## 6 6 7  
## 7 7 27  
## 8 8 12  
## 9 9 11  
## 10 10 11  
## # … with 17 more rows

median(hmc7\_incr\_Days\_grouped$hmc7\_incr\_Days)

## [1] 14

HOFT <- subset(remaining36\_g, remaining36\_g$stockName=='HOFT')  
  
HOFT <- HOFT[complete.cases(HOFT),]  
  
hoft7\_a <- ifelse(HOFT$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for hoft7 <- <- ifelse(HOFT$today2lag7>1, 1,0)  
hoft7\_ab <- cumsum(hoft7\_a)  
  
hoft7\_abc <- as.data.frame(as.factor(hoft7\_ab))  
colnames(hoft7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countHOFT <- hoft7\_abc %>% group\_by(cSum) %>% count(n=n())  
countHOFT <- as.data.frame(countHOFT)  
countHOFT <- countHOFT[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countHOFT$hoft7\_decr\_Days <- countHOFT$n-1  
countHOFT <- countHOFT[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countHOFT2 <- subset(countHOFT, countHOFT$hoft7\_decr\_Days>0)  
summary(countHOFT2$hoft7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.408 8.000 33.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
hoft7\_decr\_Days\_grouped <- countHOFT2 %>% group\_by(hoft7\_decr\_Days) %>% count(n=n())  
hoft7\_decr\_Days\_grouped <- hoft7\_decr\_Days\_grouped[,-3]

hoft7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: hoft7\_decr\_Days [24]  
## hoft7\_decr\_Days n  
## <dbl> <int>  
## 1 1 93  
## 2 2 55  
## 3 3 20  
## 4 4 21  
## 5 5 13  
## 6 6 12  
## 7 7 24  
## 8 8 16  
## 9 9 9  
## 10 10 9  
## # … with 14 more rows

median(hoft7\_decr\_Days\_grouped$hoft7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
hoft7\_b1 <- ifelse(HOFT$today2lag7>1, 0,1)  
  
  
hoft7\_b2 <- cumsum(hoft7\_b1)  
  
hoft7\_b3 <- as.data.frame(as.factor(hoft7\_b2))  
colnames(hoft7\_b3) <- 'cSum'  
  
countHOFT1 <- hoft7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countHOFT1 <- as.data.frame(countHOFT1)  
countHOFT1 <- countHOFT1[,-3]  
  
countHOFT1$hoft7\_incr\_Days <- countHOFT1$n-1  
countHOFT1 <- countHOFT1[,-2]  
  
countHOFT3 <- subset(countHOFT1, countHOFT1$hoft7\_incr\_Days>0)  
summary(countHOFT3$hoft7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.857 7.000 28.000

hoft7\_incr\_Days\_grouped <- countHOFT3 %>% group\_by(hoft7\_incr\_Days) %>% count(n=n())  
hoft7\_incr\_Days\_grouped <- hoft7\_incr\_Days\_grouped[,-3]

hoft7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: hoft7\_incr\_Days [23]  
## hoft7\_incr\_Days n  
## <dbl> <int>  
## 1 1 106  
## 2 2 41  
## 3 3 25  
## 4 4 22  
## 5 5 14  
## 6 6 11  
## 7 7 31  
## 8 8 12  
## 9 9 14  
## 10 10 9  
## # … with 13 more rows

median(hoft7\_incr\_Days\_grouped$hoft7\_incr\_Days)

## [1] 12

HRB <- subset(remaining36\_g, remaining36\_g$stockName=='HRB')  
  
HRB <- HRB[complete.cases(HRB),]  
  
hrb7\_a <- ifelse(HRB$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for hrb7 <- <- ifelse(HRB$today2lag7>1, 1,0)  
hrb7\_ab <- cumsum(hrb7\_a)  
  
hrb7\_abc <- as.data.frame(as.factor(hrb7\_ab))  
colnames(hrb7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countHRB <- hrb7\_abc %>% group\_by(cSum) %>% count(n=n())  
countHRB <- as.data.frame(countHRB)  
countHRB <- countHRB[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countHRB$hrb7\_decr\_Days <- countHRB$n-1  
countHRB <- countHRB[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countHRB2 <- subset(countHRB, countHRB$hrb7\_decr\_Days>0)  
summary(countHRB2$hrb7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 6.052 8.000 39.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
hrb7\_decr\_Days\_grouped <- countHRB2 %>% group\_by(hrb7\_decr\_Days) %>% count(n=n())  
hrb7\_decr\_Days\_grouped <- hrb7\_decr\_Days\_grouped[,-3]

hrb7\_decr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: hrb7\_decr\_Days [28]  
## hrb7\_decr\_Days n  
## <dbl> <int>  
## 1 1 89  
## 2 2 36  
## 3 3 22  
## 4 4 12  
## 5 5 17  
## 6 6 16  
## 7 7 13  
## 8 8 13  
## 9 9 9  
## 10 10 6  
## # … with 18 more rows

median(hrb7\_decr\_Days\_grouped$hrb7\_decr\_Days)

## [1] 14.5

#assign a 1 to decreasing values  
hrb7\_b1 <- ifelse(HRB$today2lag7>1, 0,1)  
  
  
hrb7\_b2 <- cumsum(hrb7\_b1)  
  
hrb7\_b3 <- as.data.frame(as.factor(hrb7\_b2))  
colnames(hrb7\_b3) <- 'cSum'  
  
countHRB1 <- hrb7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countHRB1 <- as.data.frame(countHRB1)  
countHRB1 <- countHRB1[,-3]  
  
countHRB1$hrb7\_incr\_Days <- countHRB1$n-1  
countHRB1 <- countHRB1[,-2]  
  
countHRB3 <- subset(countHRB1, countHRB1$hrb7\_incr\_Days>0)  
summary(countHRB3$hrb7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 3.00 5.31 7.00 42.00

hrb7\_incr\_Days\_grouped <- countHRB3 %>% group\_by(hrb7\_incr\_Days) %>% count(n=n())  
hrb7\_incr\_Days\_grouped <- hrb7\_incr\_Days\_grouped[,-3]

hrb7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: hrb7\_incr\_Days [23]  
## hrb7\_incr\_Days n  
## <dbl> <int>  
## 1 1 86  
## 2 2 43  
## 3 3 19  
## 4 4 15  
## 5 5 15  
## 6 6 13  
## 7 7 27  
## 8 8 11  
## 9 9 13  
## 10 10 8  
## # … with 13 more rows

median(hrb7\_incr\_Days\_grouped$hrb7\_incr\_Days)

## [1] 12

HST <- subset(remaining36\_g, remaining36\_g$stockName=='HST')  
  
HST <- HST[complete.cases(HST),]  
  
hst7\_a <- ifelse(HST$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for hst7 <- <- ifelse(HST$today2lag7>1, 1,0)  
hst7\_ab <- cumsum(hst7\_a)  
  
hst7\_abc <- as.data.frame(as.factor(hst7\_ab))  
colnames(hst7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countHST <- hst7\_abc %>% group\_by(cSum) %>% count(n=n())  
countHST <- as.data.frame(countHST)  
countHST <- countHST[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countHST$hst7\_decr\_Days <- countHST$n-1  
countHST <- countHST[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countHST2 <- subset(countHST, countHST$hst7\_decr\_Days>0)  
summary(countHST2$hst7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.799 8.000 32.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
hst7\_decr\_Days\_grouped <- countHST2 %>% group\_by(hst7\_decr\_Days) %>% count(n=n())  
hst7\_decr\_Days\_grouped <- hst7\_decr\_Days\_grouped[,-3]

hst7\_decr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: hst7\_decr\_Days [25]  
## hst7\_decr\_Days n  
## <dbl> <int>  
## 1 1 87  
## 2 2 38  
## 3 3 17  
## 4 4 14  
## 5 5 13  
## 6 6 17  
## 7 7 21  
## 8 8 16  
## 9 9 14  
## 10 10 7  
## # … with 15 more rows

median(hst7\_decr\_Days\_grouped$hst7\_decr\_Days)

## [1] 13

#assign a 1 to decreasing values  
hst7\_b1 <- ifelse(HST$today2lag7>1, 0,1)  
  
  
hst7\_b2 <- cumsum(hst7\_b1)  
  
hst7\_b3 <- as.data.frame(as.factor(hst7\_b2))  
colnames(hst7\_b3) <- 'cSum'  
  
countHST1 <- hst7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countHST1 <- as.data.frame(countHST1)  
countHST1 <- countHST1[,-3]  
  
countHST1$hst7\_incr\_Days <- countHST1$n-1  
countHST1 <- countHST1[,-2]  
  
countHST3 <- subset(countHST1, countHST1$hst7\_incr\_Days>0)  
summary(countHST3$hst7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.447 8.000 40.000

hst7\_incr\_Days\_grouped <- countHST3 %>% group\_by(hst7\_incr\_Days) %>% count(n=n())  
hst7\_incr\_Days\_grouped <- hst7\_incr\_Days\_grouped[,-3]

hst7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: hst7\_incr\_Days [24]  
## hst7\_incr\_Days n  
## <dbl> <int>  
## 1 1 93  
## 2 2 30  
## 3 3 34  
## 4 4 18  
## 5 5 13  
## 6 6 9  
## 7 7 20  
## 8 8 13  
## 9 9 9  
## 10 10 8  
## # … with 14 more rows

median(hst7\_incr\_Days\_grouped$hst7\_incr\_Days)

## [1] 12.5

INO <- subset(remaining36\_g, remaining36\_g$stockName=='INO')  
  
INO <- INO[complete.cases(INO),]  
  
ino7\_a <- ifelse(INO$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for ino7 <- <- ifelse(INO$today2lag7>1, 1,0)  
ino7\_ab <- cumsum(ino7\_a)  
  
ino7\_abc <- as.data.frame(as.factor(ino7\_ab))  
colnames(ino7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countINO <- ino7\_abc %>% group\_by(cSum) %>% count(n=n())  
countINO <- as.data.frame(countINO)  
countINO <- countINO[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countINO$ino7\_decr\_Days <- countINO$n-1  
countINO <- countINO[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countINO2 <- subset(countINO, countINO$ino7\_decr\_Days>0)  
summary(countINO2$ino7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.342 8.000 24.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
ino7\_decr\_Days\_grouped <- countINO2 %>% group\_by(ino7\_decr\_Days) %>% count(n=n())  
ino7\_decr\_Days\_grouped <- ino7\_decr\_Days\_grouped[,-3]

ino7\_decr\_Days\_grouped

## # A tibble: 20 x 2  
## # Groups: ino7\_decr\_Days [20]  
## ino7\_decr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 41  
## 3 3 25  
## 4 4 18  
## 5 5 7  
## 6 6 5  
## 7 7 29  
## 8 8 19  
## 9 9 17  
## 10 10 10  
## 11 11 5  
## 12 12 10  
## 13 13 7  
## 14 14 7  
## 15 15 4  
## 16 16 1  
## 17 17 3  
## 18 18 3  
## 19 23 1  
## 20 24 1

median(ino7\_decr\_Days\_grouped$ino7\_decr\_Days)

## [1] 10.5

#assign a 1 to decreasing values  
ino7\_b1 <- ifelse(INO$today2lag7>1, 0,1)  
  
  
ino7\_b2 <- cumsum(ino7\_b1)  
  
ino7\_b3 <- as.data.frame(as.factor(ino7\_b2))  
colnames(ino7\_b3) <- 'cSum'  
  
countINO1 <- ino7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countINO1 <- as.data.frame(countINO1)  
countINO1 <- countINO1[,-3]  
  
countINO1$ino7\_incr\_Days <- countINO1$n-1  
countINO1 <- countINO1[,-2]  
  
countINO3 <- subset(countINO1, countINO1$ino7\_incr\_Days>0)  
summary(countINO3$ino7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.847 8.750 34.000

ino7\_incr\_Days\_grouped <- countINO3 %>% group\_by(ino7\_incr\_Days) %>% count(n=n())  
ino7\_incr\_Days\_grouped <- ino7\_incr\_Days\_grouped[,-3]

ino7\_incr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: ino7\_incr\_Days [28]  
## ino7\_incr\_Days n  
## <dbl> <int>  
## 1 1 85  
## 2 2 38  
## 3 3 22  
## 4 4 17  
## 5 5 14  
## 6 6 12  
## 7 7 15  
## 8 8 17  
## 9 9 14  
## 10 10 11  
## # … with 18 more rows

median(ino7\_incr\_Days\_grouped$ino7\_incr\_Days)

## [1] 14.5

JBLU <- subset(remaining36\_g, remaining36\_g$stockName=='JBLU')  
  
JBLU <- JBLU[complete.cases(JBLU),]  
  
jblu7\_a <- ifelse(JBLU$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for jblu7 <- <- ifelse(JBLU$today2lag7>1, 1,0)  
jblu7\_ab <- cumsum(jblu7\_a)  
  
jblu7\_abc <- as.data.frame(as.factor(jblu7\_ab))  
colnames(jblu7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countJBLU <- jblu7\_abc %>% group\_by(cSum) %>% count(n=n())  
countJBLU <- as.data.frame(countJBLU)  
countJBLU <- countJBLU[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countJBLU$jblu7\_decr\_Days <- countJBLU$n-1  
countJBLU <- countJBLU[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countJBLU2 <- subset(countJBLU, countJBLU$jblu7\_decr\_Days>0)  
summary(countJBLU2$jblu7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 4.00 5.67 8.00 30.00

# this table shows how many sets of cumulative days decreasing there were in lag 7  
jblu7\_decr\_Days\_grouped <- countJBLU2 %>% group\_by(jblu7\_decr\_Days) %>% count(n=n())  
jblu7\_decr\_Days\_grouped <- jblu7\_decr\_Days\_grouped[,-3]

jblu7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: jblu7\_decr\_Days [24]  
## jblu7\_decr\_Days n  
## <dbl> <int>  
## 1 1 84  
## 2 2 44  
## 3 3 17  
## 4 4 22  
## 5 5 11  
## 6 6 9  
## 7 7 29  
## 8 8 11  
## 9 9 12  
## 10 10 12  
## # … with 14 more rows

median(jblu7\_decr\_Days\_grouped$jblu7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
jblu7\_b1 <- ifelse(JBLU$today2lag7>1, 0,1)  
  
  
jblu7\_b2 <- cumsum(jblu7\_b1)  
  
jblu7\_b3 <- as.data.frame(as.factor(jblu7\_b2))  
colnames(jblu7\_b3) <- 'cSum'  
  
countJBLU1 <- jblu7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countJBLU1 <- as.data.frame(countJBLU1)  
countJBLU1 <- countJBLU1[,-3]  
  
countJBLU1$jblu7\_incr\_Days <- countJBLU1$n-1  
countJBLU1 <- countJBLU1[,-2]  
  
countJBLU3 <- subset(countJBLU1, countJBLU1$jblu7\_incr\_Days>0)  
summary(countJBLU3$jblu7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.331 8.000 29.000

jblu7\_incr\_Days\_grouped <- countJBLU3 %>% group\_by(jblu7\_incr\_Days) %>% count(n=n())  
jblu7\_incr\_Days\_grouped <- jblu7\_incr\_Days\_grouped[,-3]

jblu7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: jblu7\_incr\_Days [23]  
## jblu7\_incr\_Days n  
## <dbl> <int>  
## 1 1 88  
## 2 2 43  
## 3 3 29  
## 4 4 17  
## 5 5 17  
## 6 6 7  
## 7 7 17  
## 8 8 15  
## 9 9 10  
## 10 10 9  
## # … with 13 more rows

median(jblu7\_incr\_Days\_grouped$jblu7\_incr\_Days)

## [1] 12

JPM <- subset(remaining36\_g, remaining36\_g$stockName=='JPM')  
  
JPM <- JPM[complete.cases(JPM),]  
  
jpm7\_a <- ifelse(JPM$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for jpm7 <- <- ifelse(JPM$today2lag7>1, 1,0)  
jpm7\_ab <- cumsum(jpm7\_a)  
  
jpm7\_abc <- as.data.frame(as.factor(jpm7\_ab))  
colnames(jpm7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countJPM <- jpm7\_abc %>% group\_by(cSum) %>% count(n=n())  
countJPM <- as.data.frame(countJPM)  
countJPM <- countJPM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countJPM$jpm7\_decr\_Days <- countJPM$n-1  
countJPM <- countJPM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countJPM2 <- subset(countJPM, countJPM$jpm7\_decr\_Days>0)  
summary(countJPM2$jpm7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.464 9.000 43.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
jpm7\_decr\_Days\_grouped <- countJPM2 %>% group\_by(jpm7\_decr\_Days) %>% count(n=n())  
jpm7\_decr\_Days\_grouped <- jpm7\_decr\_Days\_grouped[,-3]

jpm7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: jpm7\_decr\_Days [26]  
## jpm7\_decr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 37  
## 3 3 20  
## 4 4 13  
## 5 5 9  
## 6 6 11  
## 7 7 22  
## 8 8 15  
## 9 9 11  
## 10 10 9  
## # … with 16 more rows

median(jpm7\_decr\_Days\_grouped$jpm7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
jpm7\_b1 <- ifelse(JPM$today2lag7>1, 0,1)  
  
  
jpm7\_b2 <- cumsum(jpm7\_b1)  
  
jpm7\_b3 <- as.data.frame(as.factor(jpm7\_b2))  
colnames(jpm7\_b3) <- 'cSum'  
  
countJPM1 <- jpm7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countJPM1 <- as.data.frame(countJPM1)  
countJPM1 <- countJPM1[,-3]  
  
countJPM1$jpm7\_incr\_Days <- countJPM1$n-1  
countJPM1 <- countJPM1[,-2]  
  
countJPM3 <- subset(countJPM1, countJPM1$jpm7\_incr\_Days>0)  
summary(countJPM3$jpm7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.304 8.000 29.000

jpm7\_incr\_Days\_grouped <- countJPM3 %>% group\_by(jpm7\_incr\_Days) %>% count(n=n())  
jpm7\_incr\_Days\_grouped <- jpm7\_incr\_Days\_grouped[,-3]

jpm7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: jpm7\_incr\_Days [23]  
## jpm7\_incr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 38  
## 3 3 17  
## 4 4 17  
## 5 5 16  
## 6 6 13  
## 7 7 23  
## 8 8 15  
## 9 9 13  
## 10 10 11  
## # … with 13 more rows

median(jpm7\_incr\_Days\_grouped$jpm7\_incr\_Days)

## [1] 12

JWN <- subset(remaining36\_g, remaining36\_g$stockName=='JWN')  
  
JWN <- JWN[complete.cases(JWN),]  
  
jwn7\_a <- ifelse(JWN$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for jwn7 <- <- ifelse(JWN$today2lag7>1, 1,0)  
jwn7\_ab <- cumsum(jwn7\_a)  
  
jwn7\_abc <- as.data.frame(as.factor(jwn7\_ab))  
colnames(jwn7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countJWN <- jwn7\_abc %>% group\_by(cSum) %>% count(n=n())  
countJWN <- as.data.frame(countJWN)  
countJWN <- countJWN[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countJWN$jwn7\_decr\_Days <- countJWN$n-1  
countJWN <- countJWN[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countJWN2 <- subset(countJWN, countJWN$jwn7\_decr\_Days>0)  
summary(countJWN2$jwn7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 4.000 5.779 8.000 32.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
jwn7\_decr\_Days\_grouped <- countJWN2 %>% group\_by(jwn7\_decr\_Days) %>% count(n=n())  
jwn7\_decr\_Days\_grouped <- jwn7\_decr\_Days\_grouped[,-3]

jwn7\_decr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: jwn7\_decr\_Days [27]  
## jwn7\_decr\_Days n  
## <dbl> <int>  
## 1 1 65  
## 2 2 37  
## 3 3 36  
## 4 4 19  
## 5 5 22  
## 6 6 11  
## 7 7 24  
## 8 8 21  
## 9 9 11  
## 10 10 8  
## # … with 17 more rows

median(jwn7\_decr\_Days\_grouped$jwn7\_decr\_Days)

## [1] 14

#assign a 1 to decreasing values  
jwn7\_b1 <- ifelse(JWN$today2lag7>1, 0,1)  
  
  
jwn7\_b2 <- cumsum(jwn7\_b1)  
  
jwn7\_b3 <- as.data.frame(as.factor(jwn7\_b2))  
colnames(jwn7\_b3) <- 'cSum'  
  
countJWN1 <- jwn7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countJWN1 <- as.data.frame(countJWN1)  
countJWN1 <- countJWN1[,-3]  
  
countJWN1$jwn7\_incr\_Days <- countJWN1$n-1  
countJWN1 <- countJWN1[,-2]  
  
countJWN3 <- subset(countJWN1, countJWN1$jwn7\_incr\_Days>0)  
summary(countJWN3$jwn7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.241 8.000 40.000

jwn7\_incr\_Days\_grouped <- countJWN3 %>% group\_by(jwn7\_incr\_Days) %>% count(n=n())  
jwn7\_incr\_Days\_grouped <- jwn7\_incr\_Days\_grouped[,-3]

jwn7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: jwn7\_incr\_Days [22]  
## jwn7\_incr\_Days n  
## <dbl> <int>  
## 1 1 91  
## 2 2 37  
## 3 3 25  
## 4 4 13  
## 5 5 15  
## 6 6 13  
## 7 7 28  
## 8 8 14  
## 9 9 18  
## 10 10 5  
## # … with 12 more rows

median(jwn7\_incr\_Days\_grouped$jwn7\_incr\_Days)

## [1] 11.5

KGJI <- subset(remaining36\_g, remaining36\_g$stockName=='KGJI')  
  
KGJI <- KGJI[complete.cases(KGJI),]  
  
kgji7\_a <- ifelse(KGJI$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for kgji7 <- <- ifelse(KGJI$today2lag7>1, 1,0)  
kgji7\_ab <- cumsum(kgji7\_a)  
  
kgji7\_abc <- as.data.frame(as.factor(kgji7\_ab))  
colnames(kgji7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countKGJI <- kgji7\_abc %>% group\_by(cSum) %>% count(n=n())  
countKGJI <- as.data.frame(countKGJI)  
countKGJI <- countKGJI[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countKGJI$kgji7\_decr\_Days <- countKGJI$n-1  
countKGJI <- countKGJI[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countKGJI2 <- subset(countKGJI, countKGJI$kgji7\_decr\_Days>0)  
summary(countKGJI2$kgji7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 6.179 7.250 83.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
kgji7\_decr\_Days\_grouped <- countKGJI2 %>% group\_by(kgji7\_decr\_Days) %>% count(n=n())  
kgji7\_decr\_Days\_grouped <- kgji7\_decr\_Days\_grouped[,-3]

kgji7\_decr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: kgji7\_decr\_Days [28]  
## kgji7\_decr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 43  
## 3 3 36  
## 4 4 21  
## 5 5 11  
## 6 6 6  
## 7 7 29  
## 8 8 12  
## 9 9 10  
## 10 10 10  
## # … with 18 more rows

median(kgji7\_decr\_Days\_grouped$kgji7\_decr\_Days)

## [1] 14.5

#assign a 1 to decreasing values  
kgji7\_b1 <- ifelse(KGJI$today2lag7>1, 0,1)  
  
  
kgji7\_b2 <- cumsum(kgji7\_b1)  
  
kgji7\_b3 <- as.data.frame(as.factor(kgji7\_b2))  
colnames(kgji7\_b3) <- 'cSum'  
  
countKGJI1 <- kgji7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countKGJI1 <- as.data.frame(countKGJI1)  
countKGJI1 <- countKGJI1[,-3]  
  
countKGJI1$kgji7\_incr\_Days <- countKGJI1$n-1  
countKGJI1 <- countKGJI1[,-2]  
  
countKGJI3 <- subset(countKGJI1, countKGJI1$kgji7\_incr\_Days>0)  
summary(countKGJI3$kgji7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.953 7.000 27.000

kgji7\_incr\_Days\_grouped <- countKGJI3 %>% group\_by(kgji7\_incr\_Days) %>% count(n=n())  
kgji7\_incr\_Days\_grouped <- kgji7\_incr\_Days\_grouped[,-3]

kgji7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: kgji7\_incr\_Days [22]  
## kgji7\_incr\_Days n  
## <dbl> <int>  
## 1 1 93  
## 2 2 41  
## 3 3 30  
## 4 4 18  
## 5 5 11  
## 6 6 7  
## 7 7 33  
## 8 8 9  
## 9 9 6  
## 10 10 10  
## # … with 12 more rows

median(kgji7\_incr\_Days\_grouped$kgji7\_incr\_Days)

## [1] 11.5

KSS <- subset(remaining36\_g, remaining36\_g$stockName=='KSS')  
  
KSS <- KSS[complete.cases(KSS),]  
  
kss7\_a <- ifelse(KSS$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for kss7 <- <- ifelse(KSS$today2lag7>1, 1,0)  
kss7\_ab <- cumsum(kss7\_a)  
  
kss7\_abc <- as.data.frame(as.factor(kss7\_ab))  
colnames(kss7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countKSS <- kss7\_abc %>% group\_by(cSum) %>% count(n=n())  
countKSS <- as.data.frame(countKSS)  
countKSS <- countKSS[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countKSS$kss7\_decr\_Days <- countKSS$n-1  
countKSS <- countKSS[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countKSS2 <- subset(countKSS, countKSS$kss7\_decr\_Days>0)  
summary(countKSS2$kss7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.528 8.000 33.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
kss7\_decr\_Days\_grouped <- countKSS2 %>% group\_by(kss7\_decr\_Days) %>% count(n=n())  
kss7\_decr\_Days\_grouped <- kss7\_decr\_Days\_grouped[,-3]

kss7\_decr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: kss7\_decr\_Days [25]  
## kss7\_decr\_Days n  
## <dbl> <int>  
## 1 1 90  
## 2 2 36  
## 3 3 23  
## 4 4 20  
## 5 5 15  
## 6 6 16  
## 7 7 18  
## 8 8 20  
## 9 9 13  
## 10 10 8  
## # … with 15 more rows

median(kss7\_decr\_Days\_grouped$kss7\_decr\_Days)

## [1] 13

#assign a 1 to decreasing values  
kss7\_b1 <- ifelse(KSS$today2lag7>1, 0,1)  
  
  
kss7\_b2 <- cumsum(kss7\_b1)  
  
kss7\_b3 <- as.data.frame(as.factor(kss7\_b2))  
colnames(kss7\_b3) <- 'cSum'  
  
countKSS1 <- kss7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countKSS1 <- as.data.frame(countKSS1)  
countKSS1 <- countKSS1[,-3]  
  
countKSS1$kss7\_incr\_Days <- countKSS1$n-1  
countKSS1 <- countKSS1[,-2]  
  
countKSS3 <- subset(countKSS1, countKSS1$kss7\_incr\_Days>0)  
summary(countKSS3$kss7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.347 8.000 40.000

kss7\_incr\_Days\_grouped <- countKSS3 %>% group\_by(kss7\_incr\_Days) %>% count(n=n())  
kss7\_incr\_Days\_grouped <- kss7\_incr\_Days\_grouped[,-3]

kss7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: kss7\_incr\_Days [22]  
## kss7\_incr\_Days n  
## <dbl> <int>  
## 1 1 84  
## 2 2 45  
## 3 3 26  
## 4 4 20  
## 5 5 11  
## 6 6 10  
## 7 7 22  
## 8 8 23  
## 9 9 7  
## 10 10 9  
## # … with 12 more rows

median(kss7\_incr\_Days\_grouped$kss7\_incr\_Days)

## [1] 11.5

LUV <- subset(remaining36\_g, remaining36\_g$stockName=='LUV')  
  
LUV <- LUV[complete.cases(LUV),]  
  
luv7\_a <- ifelse(LUV$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for luv7 <- <- ifelse(LUV$today2lag7>1, 1,0)  
luv7\_ab <- cumsum(luv7\_a)  
  
luv7\_abc <- as.data.frame(as.factor(luv7\_ab))  
colnames(luv7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countLUV <- luv7\_abc %>% group\_by(cSum) %>% count(n=n())  
countLUV <- as.data.frame(countLUV)  
countLUV <- countLUV[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countLUV$luv7\_decr\_Days <- countLUV$n-1  
countLUV <- countLUV[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countLUV2 <- subset(countLUV, countLUV$luv7\_decr\_Days>0)  
summary(countLUV2$luv7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.498 9.000 61.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
luv7\_decr\_Days\_grouped <- countLUV2 %>% group\_by(luv7\_decr\_Days) %>% count(n=n())  
luv7\_decr\_Days\_grouped <- luv7\_decr\_Days\_grouped[,-3]

luv7\_decr\_Days\_grouped

## # A tibble: 32 x 2  
## # Groups: luv7\_decr\_Days [32]  
## luv7\_decr\_Days n  
## <dbl> <int>  
## 1 1 83  
## 2 2 33  
## 3 3 20  
## 4 4 17  
## 5 5 16  
## 6 6 8  
## 7 7 21  
## 8 8 8  
## 9 9 18  
## 10 10 8  
## # … with 22 more rows

median(luv7\_decr\_Days\_grouped$luv7\_decr\_Days)

## [1] 16.5

#assign a 1 to decreasing values  
luv7\_b1 <- ifelse(LUV$today2lag7>1, 0,1)  
  
  
luv7\_b2 <- cumsum(luv7\_b1)  
  
luv7\_b3 <- as.data.frame(as.factor(luv7\_b2))  
colnames(luv7\_b3) <- 'cSum'  
  
countLUV1 <- luv7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countLUV1 <- as.data.frame(countLUV1)  
countLUV1 <- countLUV1[,-3]  
  
countLUV1$luv7\_incr\_Days <- countLUV1$n-1  
countLUV1 <- countLUV1[,-2]  
  
countLUV3 <- subset(countLUV1, countLUV1$luv7\_incr\_Days>0)  
summary(countLUV3$luv7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.246 7.000 37.000

luv7\_incr\_Days\_grouped <- countLUV3 %>% group\_by(luv7\_incr\_Days) %>% count(n=n())  
luv7\_incr\_Days\_grouped <- luv7\_incr\_Days\_grouped[,-3]

luv7\_incr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: luv7\_incr\_Days [25]  
## luv7\_incr\_Days n  
## <dbl> <int>  
## 1 1 101  
## 2 2 34  
## 3 3 21  
## 4 4 14  
## 5 5 8  
## 6 6 17  
## 7 7 21  
## 8 8 11  
## 9 9 4  
## 10 10 8  
## # … with 15 more rows

median(luv7\_incr\_Days\_grouped$luv7\_incr\_Days)

## [1] 13

M <- subset(remaining36\_g, remaining36\_g$stockName=='M')  
  
M <- M[complete.cases(M),]  
  
m7\_a <- ifelse(M$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for m7 <- <- ifelse(M$today2lag7>1, 1,0)  
m7\_ab <- cumsum(m7\_a)  
  
m7\_abc <- as.data.frame(as.factor(m7\_ab))  
colnames(m7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countM <- m7\_abc %>% group\_by(cSum) %>% count(n=n())  
countM <- as.data.frame(countM)  
countM <- countM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countM$m7\_decr\_Days <- countM$n-1  
countM <- countM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countM2 <- subset(countM, countM$m7\_decr\_Days>0)  
summary(countM2$m7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 2.00 4.00 6.21 8.00 38.00

# this table shows how many sets of cumulative days decreasing there were in lag 7  
m7\_decr\_Days\_grouped <- countM2 %>% group\_by(m7\_decr\_Days) %>% count(n=n())  
m7\_decr\_Days\_grouped <- m7\_decr\_Days\_grouped[,-3]

m7\_decr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: m7\_decr\_Days [28]  
## m7\_decr\_Days n  
## <dbl> <int>  
## 1 1 65  
## 2 2 39  
## 3 3 28  
## 4 4 16  
## 5 5 12  
## 6 6 12  
## 7 7 26  
## 8 8 11  
## 9 9 7  
## 10 10 5  
## # … with 18 more rows

median(m7\_decr\_Days\_grouped$m7\_decr\_Days)

## [1] 14.5

#assign a 1 to decreasing values  
m7\_b1 <- ifelse(M$today2lag7>1, 0,1)  
  
  
m7\_b2 <- cumsum(m7\_b1)  
  
m7\_b3 <- as.data.frame(as.factor(m7\_b2))  
colnames(m7\_b3) <- 'cSum'  
  
countM1 <- m7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countM1 <- as.data.frame(countM1)  
countM1 <- countM1[,-3]  
  
countM1$m7\_incr\_Days <- countM1$n-1  
countM1 <- countM1[,-2]  
  
countM3 <- subset(countM1, countM1$m7\_incr\_Days>0)  
summary(countM3$m7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.904 8.000 28.000

m7\_incr\_Days\_grouped <- countM3 %>% group\_by(m7\_incr\_Days) %>% count(n=n())  
m7\_incr\_Days\_grouped <- m7\_incr\_Days\_grouped[,-3]

m7\_incr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: m7\_incr\_Days [25]  
## m7\_incr\_Days n  
## <dbl> <int>  
## 1 1 69  
## 2 2 35  
## 3 3 27  
## 4 4 13  
## 5 5 12  
## 6 6 17  
## 7 7 27  
## 8 8 9  
## 9 9 9  
## 10 10 9  
## # … with 15 more rows

median(m7\_incr\_Days\_grouped$m7\_incr\_Days)

## [1] 13

MGM <- subset(remaining36\_g, remaining36\_g$stockName=='MGM')  
  
MGM <- MGM[complete.cases(MGM),]  
  
mgm7\_a <- ifelse(MGM$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for mgm7 <- <- ifelse(MGM$today2lag7>1, 1,0)  
mgm7\_ab <- cumsum(mgm7\_a)  
  
mgm7\_abc <- as.data.frame(as.factor(mgm7\_ab))  
colnames(mgm7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countMGM <- mgm7\_abc %>% group\_by(cSum) %>% count(n=n())  
countMGM <- as.data.frame(countMGM)  
countMGM <- countMGM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countMGM$mgm7\_decr\_Days <- countMGM$n-1  
countMGM <- countMGM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countMGM2 <- subset(countMGM, countMGM$mgm7\_decr\_Days>0)  
summary(countMGM2$mgm7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.083 9.000 38.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
mgm7\_decr\_Days\_grouped <- countMGM2 %>% group\_by(mgm7\_decr\_Days) %>% count(n=n())  
mgm7\_decr\_Days\_grouped <- mgm7\_decr\_Days\_grouped[,-3]

mgm7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: mgm7\_decr\_Days [26]  
## mgm7\_decr\_Days n  
## <dbl> <int>  
## 1 1 79  
## 2 2 36  
## 3 3 17  
## 4 4 16  
## 5 5 11  
## 6 6 5  
## 7 7 27  
## 8 8 14  
## 9 9 5  
## 10 10 13  
## # … with 16 more rows

median(mgm7\_decr\_Days\_grouped$mgm7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
mgm7\_b1 <- ifelse(MGM$today2lag7>1, 0,1)  
  
  
mgm7\_b2 <- cumsum(mgm7\_b1)  
  
mgm7\_b3 <- as.data.frame(as.factor(mgm7\_b2))  
colnames(mgm7\_b3) <- 'cSum'  
  
countMGM1 <- mgm7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countMGM1 <- as.data.frame(countMGM1)  
countMGM1 <- countMGM1[,-3]  
  
countMGM1$mgm7\_incr\_Days <- countMGM1$n-1  
countMGM1 <- countMGM1[,-2]  
  
countMGM3 <- subset(countMGM1, countMGM1$mgm7\_incr\_Days>0)  
summary(countMGM3$mgm7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 4.00 5.77 8.00 41.00

mgm7\_incr\_Days\_grouped <- countMGM3 %>% group\_by(mgm7\_incr\_Days) %>% count(n=n())  
mgm7\_incr\_Days\_grouped <- mgm7\_incr\_Days\_grouped[,-3]

mgm7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: mgm7\_incr\_Days [24]  
## mgm7\_incr\_Days n  
## <dbl> <int>  
## 1 1 74  
## 2 2 36  
## 3 3 25  
## 4 4 18  
## 5 5 10  
## 6 6 9  
## 7 7 23  
## 8 8 16  
## 9 9 11  
## 10 10 13  
## # … with 14 more rows

median(mgm7\_incr\_Days\_grouped$mgm7\_incr\_Days)

## [1] 12.5

MSFT <- subset(remaining36\_g, remaining36\_g$stockName=='MSFT')  
  
MSFT <- MSFT[complete.cases(MSFT),]  
  
msft7\_a <- ifelse(MSFT$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for msft7 <- <- ifelse(MSFT$today2lag7>1, 1,0)  
msft7\_ab <- cumsum(msft7\_a)  
  
msft7\_abc <- as.data.frame(as.factor(msft7\_ab))  
colnames(msft7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countMSFT <- msft7\_abc %>% group\_by(cSum) %>% count(n=n())  
countMSFT <- as.data.frame(countMSFT)  
countMSFT <- countMSFT[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countMSFT$msft7\_decr\_Days <- countMSFT$n-1  
countMSFT <- countMSFT[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countMSFT2 <- subset(countMSFT, countMSFT$msft7\_decr\_Days>0)  
summary(countMSFT2$msft7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 4.000 6.273 9.000 33.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
msft7\_decr\_Days\_grouped <- countMSFT2 %>% group\_by(msft7\_decr\_Days) %>% count(n=n())  
msft7\_decr\_Days\_grouped <- msft7\_decr\_Days\_grouped[,-3]

msft7\_decr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: msft7\_decr\_Days [28]  
## msft7\_decr\_Days n  
## <dbl> <int>  
## 1 1 74  
## 2 2 33  
## 3 3 24  
## 4 4 30  
## 5 5 6  
## 6 6 13  
## 7 7 28  
## 8 8 13  
## 9 9 10  
## 10 10 13  
## # … with 18 more rows

median(msft7\_decr\_Days\_grouped$msft7\_decr\_Days)

## [1] 14.5

#assign a 1 to decreasing values  
msft7\_b1 <- ifelse(MSFT$today2lag7>1, 0,1)  
  
  
msft7\_b2 <- cumsum(msft7\_b1)  
  
msft7\_b3 <- as.data.frame(as.factor(msft7\_b2))  
colnames(msft7\_b3) <- 'cSum'  
  
countMSFT1 <- msft7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countMSFT1 <- as.data.frame(countMSFT1)  
countMSFT1 <- countMSFT1[,-3]  
  
countMSFT1$msft7\_incr\_Days <- countMSFT1$n-1  
countMSFT1 <- countMSFT1[,-2]  
  
countMSFT3 <- subset(countMSFT1, countMSFT1$msft7\_incr\_Days>0)  
summary(countMSFT3$msft7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.726 7.000 29.000

msft7\_incr\_Days\_grouped <- countMSFT3 %>% group\_by(msft7\_incr\_Days) %>% count(n=n())  
msft7\_incr\_Days\_grouped <- msft7\_incr\_Days\_grouped[,-3]

msft7\_incr\_Days\_grouped

## # A tibble: 21 x 2  
## # Groups: msft7\_incr\_Days [21]  
## msft7\_incr\_Days n  
## <dbl> <int>  
## 1 1 92  
## 2 2 42  
## 3 3 25  
## 4 4 18  
## 5 5 18  
## 6 6 11  
## 7 7 32  
## 8 8 16  
## 9 9 9  
## 10 10 9  
## # … with 11 more rows

median(msft7\_incr\_Days\_grouped$msft7\_incr\_Days)

## [1] 11

NSANY <- subset(remaining36\_g, remaining36\_g$stockName=='NSANY')  
  
NSANY <- NSANY[complete.cases(NSANY),]  
  
nsany7\_a <- ifelse(NSANY$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for nsany7 <- <- ifelse(NSANY$today2lag7>1, 1,0)  
nsany7\_ab <- cumsum(nsany7\_a)  
  
nsany7\_abc <- as.data.frame(as.factor(nsany7\_ab))  
colnames(nsany7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countNSANY <- nsany7\_abc %>% group\_by(cSum) %>% count(n=n())  
countNSANY <- as.data.frame(countNSANY)  
countNSANY <- countNSANY[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countNSANY$nsany7\_decr\_Days <- countNSANY$n-1  
countNSANY <- countNSANY[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countNSANY2 <- subset(countNSANY, countNSANY$nsany7\_decr\_Days>0)  
summary(countNSANY2$nsany7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.579 8.750 35.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
nsany7\_decr\_Days\_grouped <- countNSANY2 %>% group\_by(nsany7\_decr\_Days) %>% count(n=n())  
nsany7\_decr\_Days\_grouped <- nsany7\_decr\_Days\_grouped[,-3]

nsany7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: nsany7\_decr\_Days [24]  
## nsany7\_decr\_Days n  
## <dbl> <int>  
## 1 1 87  
## 2 2 39  
## 3 3 20  
## 4 4 16  
## 5 5 13  
## 6 6 11  
## 7 7 15  
## 8 8 16  
## 9 9 19  
## 10 10 10  
## # … with 14 more rows

median(nsany7\_decr\_Days\_grouped$nsany7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
nsany7\_b1 <- ifelse(NSANY$today2lag7>1, 0,1)  
  
  
nsany7\_b2 <- cumsum(nsany7\_b1)  
  
nsany7\_b3 <- as.data.frame(as.factor(nsany7\_b2))  
colnames(nsany7\_b3) <- 'cSum'  
  
countNSANY1 <- nsany7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countNSANY1 <- as.data.frame(countNSANY1)  
countNSANY1 <- countNSANY1[,-3]  
  
countNSANY1$nsany7\_incr\_Days <- countNSANY1$n-1  
countNSANY1 <- countNSANY1[,-2]  
  
countNSANY3 <- subset(countNSANY1, countNSANY1$nsany7\_incr\_Days>0)  
summary(countNSANY3$nsany7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.763 8.000 29.000

nsany7\_incr\_Days\_grouped <- countNSANY3 %>% group\_by(nsany7\_incr\_Days) %>% count(n=n())  
nsany7\_incr\_Days\_grouped <- nsany7\_incr\_Days\_grouped[,-3]

nsany7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: nsany7\_incr\_Days [24]  
## nsany7\_incr\_Days n  
## <dbl> <int>  
## 1 1 89  
## 2 2 26  
## 3 3 15  
## 4 4 19  
## 5 5 20  
## 6 6 16  
## 7 7 25  
## 8 8 12  
## 9 9 13  
## 10 10 12  
## # … with 14 more rows

median(nsany7\_incr\_Days\_grouped$nsany7\_incr\_Days)

## [1] 12.5

NUS <- subset(remaining36\_g, remaining36\_g$stockName=='NUS')  
  
NUS <- NUS[complete.cases(NUS),]  
  
nus7\_a <- ifelse(NUS$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for nus7 <- <- ifelse(NUS$today2lag7>1, 1,0)  
nus7\_ab <- cumsum(nus7\_a)  
  
nus7\_abc <- as.data.frame(as.factor(nus7\_ab))  
colnames(nus7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countNUS <- nus7\_abc %>% group\_by(cSum) %>% count(n=n())  
countNUS <- as.data.frame(countNUS)  
countNUS <- countNUS[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countNUS$nus7\_decr\_Days <- countNUS$n-1  
countNUS <- countNUS[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countNUS2 <- subset(countNUS, countNUS$nus7\_decr\_Days>0)  
summary(countNUS2$nus7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.128 9.000 34.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
nus7\_decr\_Days\_grouped <- countNUS2 %>% group\_by(nus7\_decr\_Days) %>% count(n=n())  
nus7\_decr\_Days\_grouped <- nus7\_decr\_Days\_grouped[,-3]

nus7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: nus7\_decr\_Days [26]  
## nus7\_decr\_Days n  
## <dbl> <int>  
## 1 1 84  
## 2 2 35  
## 3 3 20  
## 4 4 15  
## 5 5 13  
## 6 6 9  
## 7 7 20  
## 8 8 20  
## 9 9 10  
## 10 10 10  
## # … with 16 more rows

median(nus7\_decr\_Days\_grouped$nus7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
nus7\_b1 <- ifelse(NUS$today2lag7>1, 0,1)  
  
  
nus7\_b2 <- cumsum(nus7\_b1)  
  
nus7\_b3 <- as.data.frame(as.factor(nus7\_b2))  
colnames(nus7\_b3) <- 'cSum'  
  
countNUS1 <- nus7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countNUS1 <- as.data.frame(countNUS1)  
countNUS1 <- countNUS1[,-3]  
  
countNUS1$nus7\_incr\_Days <- countNUS1$n-1  
countNUS1 <- countNUS1[,-2]  
  
countNUS3 <- subset(countNUS1, countNUS1$nus7\_incr\_Days>0)  
summary(countNUS3$nus7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.255 7.000 28.000

nus7\_incr\_Days\_grouped <- countNUS3 %>% group\_by(nus7\_incr\_Days) %>% count(n=n())  
nus7\_incr\_Days\_grouped <- nus7\_incr\_Days\_grouped[,-3]

nus7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: nus7\_incr\_Days [24]  
## nus7\_incr\_Days n  
## <dbl> <int>  
## 1 1 89  
## 2 2 38  
## 3 3 26  
## 4 4 17  
## 5 5 13  
## 6 6 14  
## 7 7 26  
## 8 8 7  
## 9 9 13  
## 10 10 8  
## # … with 14 more rows

median(nus7\_incr\_Days\_grouped$nus7\_incr\_Days)

## [1] 12.5

ONCY <- subset(remaining36\_g, remaining36\_g$stockName=='ONCY')  
  
ONCY <- ONCY[complete.cases(ONCY),]  
  
oncy7\_a <- ifelse(ONCY$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for oncy7 <- <- ifelse(ONCY$today2lag7>1, 1,0)  
oncy7\_ab <- cumsum(oncy7\_a)  
  
oncy7\_abc <- as.data.frame(as.factor(oncy7\_ab))  
colnames(oncy7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countONCY <- oncy7\_abc %>% group\_by(cSum) %>% count(n=n())  
countONCY <- as.data.frame(countONCY)  
countONCY <- countONCY[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countONCY$oncy7\_decr\_Days <- countONCY$n-1  
countONCY <- countONCY[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countONCY2 <- subset(countONCY, countONCY$oncy7\_decr\_Days>0)  
summary(countONCY2$oncy7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.094 7.000 23.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
oncy7\_decr\_Days\_grouped <- countONCY2 %>% group\_by(oncy7\_decr\_Days) %>% count(n=n())  
oncy7\_decr\_Days\_grouped <- oncy7\_decr\_Days\_grouped[,-3]

oncy7\_decr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: oncy7\_decr\_Days [23]  
## oncy7\_decr\_Days n  
## <dbl> <int>  
## 1 1 91  
## 2 2 33  
## 3 3 33  
## 4 4 23  
## 5 5 11  
## 6 6 11  
## 7 7 23  
## 8 8 18  
## 9 9 7  
## 10 10 7  
## # … with 13 more rows

median(oncy7\_decr\_Days\_grouped$oncy7\_decr\_Days)

## [1] 12

#assign a 1 to decreasing values  
oncy7\_b1 <- ifelse(ONCY$today2lag7>1, 0,1)  
  
  
oncy7\_b2 <- cumsum(oncy7\_b1)  
  
oncy7\_b3 <- as.data.frame(as.factor(oncy7\_b2))  
colnames(oncy7\_b3) <- 'cSum'  
  
countONCY1 <- oncy7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countONCY1 <- as.data.frame(countONCY1)  
countONCY1 <- countONCY1[,-3]  
  
countONCY1$oncy7\_incr\_Days <- countONCY1$n-1  
countONCY1 <- countONCY1[,-2]  
  
countONCY3 <- subset(countONCY1, countONCY1$oncy7\_incr\_Days>0)  
summary(countONCY3$oncy7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.963 9.000 28.000

oncy7\_incr\_Days\_grouped <- countONCY3 %>% group\_by(oncy7\_incr\_Days) %>% count(n=n())  
oncy7\_incr\_Days\_grouped <- oncy7\_incr\_Days\_grouped[,-3]

oncy7\_incr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: oncy7\_incr\_Days [25]  
## oncy7\_incr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 38  
## 3 3 25  
## 4 4 20  
## 5 5 19  
## 6 6 11  
## 7 7 20  
## 8 8 11  
## 9 9 5  
## 10 10 10  
## # … with 15 more rows

median(oncy7\_incr\_Days\_grouped$oncy7\_incr\_Days)

## [1] 13

RRGB <- subset(remaining36\_g, remaining36\_g$stockName=='RRGB')  
  
RRGB <- RRGB[complete.cases(RRGB),]  
  
rrgb7\_a <- ifelse(RRGB$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for rrgb7 <- <- ifelse(RRGB$today2lag7>1, 1,0)  
rrgb7\_ab <- cumsum(rrgb7\_a)  
  
rrgb7\_abc <- as.data.frame(as.factor(rrgb7\_ab))  
colnames(rrgb7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countRRGB <- rrgb7\_abc %>% group\_by(cSum) %>% count(n=n())  
countRRGB <- as.data.frame(countRRGB)  
countRRGB <- countRRGB[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countRRGB$rrgb7\_decr\_Days <- countRRGB$n-1  
countRRGB <- countRRGB[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countRRGB2 <- subset(countRRGB, countRRGB$rrgb7\_decr\_Days>0)  
summary(countRRGB2$rrgb7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.207 9.250 26.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
rrgb7\_decr\_Days\_grouped <- countRRGB2 %>% group\_by(rrgb7\_decr\_Days) %>% count(n=n())  
rrgb7\_decr\_Days\_grouped <- rrgb7\_decr\_Days\_grouped[,-3]

rrgb7\_decr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: rrgb7\_decr\_Days [22]  
## rrgb7\_decr\_Days n  
## <dbl> <int>  
## 1 1 72  
## 2 2 31  
## 3 3 26  
## 4 4 14  
## 5 5 10  
## 6 6 3  
## 7 7 26  
## 8 8 12  
## 9 9 13  
## 10 10 9  
## # … with 12 more rows

median(rrgb7\_decr\_Days\_grouped$rrgb7\_decr\_Days)

## [1] 11.5

#assign a 1 to decreasing values  
rrgb7\_b1 <- ifelse(RRGB$today2lag7>1, 0,1)  
  
  
rrgb7\_b2 <- cumsum(rrgb7\_b1)  
  
rrgb7\_b3 <- as.data.frame(as.factor(rrgb7\_b2))  
colnames(rrgb7\_b3) <- 'cSum'  
  
countRRGB1 <- rrgb7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countRRGB1 <- as.data.frame(countRRGB1)  
countRRGB1 <- countRRGB1[,-3]  
  
countRRGB1$rrgb7\_incr\_Days <- countRRGB1$n-1  
countRRGB1 <- countRRGB1[,-2]  
  
countRRGB3 <- subset(countRRGB1, countRRGB1$rrgb7\_incr\_Days>0)  
summary(countRRGB3$rrgb7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.500 5.732 9.000 28.000

rrgb7\_incr\_Days\_grouped <- countRRGB3 %>% group\_by(rrgb7\_incr\_Days) %>% count(n=n())  
rrgb7\_incr\_Days\_grouped <- rrgb7\_incr\_Days\_grouped[,-3]

rrgb7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: rrgb7\_incr\_Days [24]  
## rrgb7\_incr\_Days n  
## <dbl> <int>  
## 1 1 73  
## 2 2 45  
## 3 3 20  
## 4 4 15  
## 5 5 12  
## 6 6 7  
## 7 7 21  
## 8 8 11  
## 9 9 16  
## 10 10 7  
## # … with 14 more rows

median(rrgb7\_incr\_Days\_grouped$rrgb7\_incr\_Days)

## [1] 12.5

S <- subset(remaining36\_g, remaining36\_g$stockName=='S')  
  
S <- S[complete.cases(S),]  
  
s7\_a <- ifelse(S$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for s7 <- <- ifelse(S$today2lag7>1, 1,0)  
s7\_ab <- cumsum(s7\_a)  
  
s7\_abc <- as.data.frame(as.factor(s7\_ab))  
colnames(s7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countS <- s7\_abc %>% group\_by(cSum) %>% count(n=n())  
countS <- as.data.frame(countS)  
countS <- countS[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countS$s7\_decr\_Days <- countS$n-1  
countS <- countS[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countS2 <- subset(countS, countS$s7\_decr\_Days>0)  
summary(countS2$s7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 1.00 4.00 5.75 8.00 27.00

# this table shows how many sets of cumulative days decreasing there were in lag 7  
s7\_decr\_Days\_grouped <- countS2 %>% group\_by(s7\_decr\_Days) %>% count(n=n())  
s7\_decr\_Days\_grouped <- s7\_decr\_Days\_grouped[,-3]

s7\_decr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: s7\_decr\_Days [24]  
## s7\_decr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 36  
## 3 3 16  
## 4 4 21  
## 5 5 14  
## 6 6 16  
## 7 7 20  
## 8 8 18  
## 9 9 13  
## 10 10 8  
## # … with 14 more rows

median(s7\_decr\_Days\_grouped$s7\_decr\_Days)

## [1] 12.5

#assign a 1 to decreasing values  
s7\_b1 <- ifelse(S$today2lag7>1, 0,1)  
  
  
s7\_b2 <- cumsum(s7\_b1)  
  
s7\_b3 <- as.data.frame(as.factor(s7\_b2))  
colnames(s7\_b3) <- 'cSum'  
  
countS1 <- s7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countS1 <- as.data.frame(countS1)  
countS1 <- countS1[,-3]  
  
countS1$s7\_incr\_Days <- countS1$n-1  
countS1 <- countS1[,-2]  
  
countS3 <- subset(countS1, countS1$s7\_incr\_Days>0)  
summary(countS3$s7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.852 8.000 37.000

s7\_incr\_Days\_grouped <- countS3 %>% group\_by(s7\_incr\_Days) %>% count(n=n())  
s7\_incr\_Days\_grouped <- s7\_incr\_Days\_grouped[,-3]

s7\_incr\_Days\_grouped

## # A tibble: 28 x 2  
## # Groups: s7\_incr\_Days [28]  
## s7\_incr\_Days n  
## <dbl> <int>  
## 1 1 81  
## 2 2 33  
## 3 3 22  
## 4 4 28  
## 5 5 11  
## 6 6 12  
## 7 7 22  
## 8 8 14  
## 9 9 8  
## 10 10 4  
## # … with 18 more rows

median(s7\_incr\_Days\_grouped$s7\_incr\_Days)

## [1] 14.5

SCE <- subset(remaining36\_g, remaining36\_g$stockName=='SCE')  
  
SCE <- SCE[complete.cases(SCE),]  
  
sce7\_a <- ifelse(SCE$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for sce7 <- <- ifelse(SCE$today2lag7>1, 1,0)  
sce7\_ab <- cumsum(sce7\_a)  
  
sce7\_abc <- as.data.frame(as.factor(sce7\_ab))  
colnames(sce7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countSCE <- sce7\_abc %>% group\_by(cSum) %>% count(n=n())  
countSCE <- as.data.frame(countSCE)  
countSCE <- countSCE[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countSCE$sce7\_decr\_Days <- countSCE$n-1  
countSCE <- countSCE[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countSCE2 <- subset(countSCE, countSCE$sce7\_decr\_Days>0)  
summary(countSCE2$sce7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 3.559 4.000 189.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
sce7\_decr\_Days\_grouped <- countSCE2 %>% group\_by(sce7\_decr\_Days) %>% count(n=n())  
sce7\_decr\_Days\_grouped <- sce7\_decr\_Days\_grouped[,-3]

sce7\_decr\_Days\_grouped

## # A tibble: 21 x 2  
## # Groups: sce7\_decr\_Days [21]  
## sce7\_decr\_Days n  
## <dbl> <int>  
## 1 1 211  
## 2 2 126  
## 3 3 67  
## 4 4 43  
## 5 5 30  
## 6 6 14  
## 7 7 26  
## 8 8 21  
## 9 9 3  
## 10 10 5  
## # … with 11 more rows

median(sce7\_decr\_Days\_grouped$sce7\_decr\_Days)

## [1] 11

#assign a 1 to decreasing values  
sce7\_b1 <- ifelse(SCE$today2lag7>1, 0,1)  
  
  
sce7\_b2 <- cumsum(sce7\_b1)  
  
sce7\_b3 <- as.data.frame(as.factor(sce7\_b2))  
colnames(sce7\_b3) <- 'cSum'  
  
countSCE1 <- sce7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countSCE1 <- as.data.frame(countSCE1)  
countSCE1 <- countSCE1[,-3]  
  
countSCE1$sce7\_incr\_Days <- countSCE1$n-1  
countSCE1 <- countSCE1[,-2]  
  
countSCE3 <- subset(countSCE1, countSCE1$sce7\_incr\_Days>0)  
summary(countSCE3$sce7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 2.236 3.000 15.000

sce7\_incr\_Days\_grouped <- countSCE3 %>% group\_by(sce7\_incr\_Days) %>% count(n=n())  
sce7\_incr\_Days\_grouped <- sce7\_incr\_Days\_grouped[,-3]

sce7\_incr\_Days\_grouped

## # A tibble: 15 x 2  
## # Groups: sce7\_incr\_Days [15]  
## sce7\_incr\_Days n  
## <dbl> <int>  
## 1 1 313  
## 2 2 108  
## 3 3 66  
## 4 4 25  
## 5 5 9  
## 6 6 9  
## 7 7 14  
## 8 8 5  
## 9 9 7  
## 10 10 3  
## 11 11 3  
## 12 12 1  
## 13 13 3  
## 14 14 1  
## 15 15 1

median(sce7\_incr\_Days\_grouped$sce7\_incr\_Days)

## [1] 8

SIG <- subset(remaining36\_g, remaining36\_g$stockName=='SIG')  
  
SIG <- SIG[complete.cases(SIG),]  
  
sig7\_a <- ifelse(SIG$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for sig7 <- <- ifelse(SIG$today2lag7>1, 1,0)  
sig7\_ab <- cumsum(sig7\_a)  
  
sig7\_abc <- as.data.frame(as.factor(sig7\_ab))  
colnames(sig7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countSIG <- sig7\_abc %>% group\_by(cSum) %>% count(n=n())  
countSIG <- as.data.frame(countSIG)  
countSIG <- countSIG[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countSIG$sig7\_decr\_Days <- countSIG$n-1  
countSIG <- countSIG[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countSIG2 <- subset(countSIG, countSIG$sig7\_decr\_Days>0)  
summary(countSIG2$sig7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.175 9.000 34.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
sig7\_decr\_Days\_grouped <- countSIG2 %>% group\_by(sig7\_decr\_Days) %>% count(n=n())  
sig7\_decr\_Days\_grouped <- sig7\_decr\_Days\_grouped[,-3]

sig7\_decr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: sig7\_decr\_Days [25]  
## sig7\_decr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 31  
## 3 3 25  
## 4 4 9  
## 5 5 15  
## 6 6 10  
## 7 7 19  
## 8 8 15  
## 9 9 8  
## 10 10 17  
## # … with 15 more rows

median(sig7\_decr\_Days\_grouped$sig7\_decr\_Days)

## [1] 13

#assign a 1 to decreasing values  
sig7\_b1 <- ifelse(SIG$today2lag7>1, 0,1)  
  
  
sig7\_b2 <- cumsum(sig7\_b1)  
  
sig7\_b3 <- as.data.frame(as.factor(sig7\_b2))  
colnames(sig7\_b3) <- 'cSum'  
  
countSIG1 <- sig7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countSIG1 <- as.data.frame(countSIG1)  
countSIG1 <- countSIG1[,-3]  
  
countSIG1$sig7\_incr\_Days <- countSIG1$n-1  
countSIG1 <- countSIG1[,-2]  
  
countSIG3 <- subset(countSIG1, countSIG1$sig7\_incr\_Days>0)  
summary(countSIG3$sig7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 4.000 5.829 8.000 42.000

sig7\_incr\_Days\_grouped <- countSIG3 %>% group\_by(sig7\_incr\_Days) %>% count(n=n())  
sig7\_incr\_Days\_grouped <- sig7\_incr\_Days\_grouped[,-3]

sig7\_incr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: sig7\_incr\_Days [23]  
## sig7\_incr\_Days n  
## <dbl> <int>  
## 1 1 66  
## 2 2 40  
## 3 3 27  
## 4 4 20  
## 5 5 11  
## 6 6 9  
## 7 7 23  
## 8 8 17  
## 9 9 11  
## 10 10 7  
## # … with 13 more rows

median(sig7\_incr\_Days\_grouped$sig7\_incr\_Days)

## [1] 12

T <- subset(remaining36\_g, remaining36\_g$stockName=='T')  
  
T <- T[complete.cases(T),]  
  
t7\_a <- ifelse(T$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for t7 <- <- ifelse(T$today2lag7>1, 1,0)  
t7\_ab <- cumsum(t7\_a)  
  
t7\_abc <- as.data.frame(as.factor(t7\_ab))  
colnames(t7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countT <- t7\_abc %>% group\_by(cSum) %>% count(n=n())  
countT <- as.data.frame(countT)  
countT <- countT[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countT$t7\_decr\_Days <- countT$n-1  
countT <- countT[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countT2 <- subset(countT, countT$t7\_decr\_Days>0)  
summary(countT2$t7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 5.723 8.000 35.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
t7\_decr\_Days\_grouped <- countT2 %>% group\_by(t7\_decr\_Days) %>% count(n=n())  
t7\_decr\_Days\_grouped <- t7\_decr\_Days\_grouped[,-3]

t7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: t7\_decr\_Days [26]  
## t7\_decr\_Days n  
## <dbl> <int>  
## 1 1 73  
## 2 2 46  
## 3 3 33  
## 4 4 19  
## 5 5 17  
## 6 6 9  
## 7 7 23  
## 8 8 15  
## 9 9 4  
## 10 10 16  
## # … with 16 more rows

median(t7\_decr\_Days\_grouped$t7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
t7\_b1 <- ifelse(T$today2lag7>1, 0,1)  
  
  
t7\_b2 <- cumsum(t7\_b1)  
  
t7\_b3 <- as.data.frame(as.factor(t7\_b2))  
colnames(t7\_b3) <- 'cSum'  
  
countT1 <- t7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countT1 <- as.data.frame(countT1)  
countT1 <- countT1[,-3]  
  
countT1$t7\_incr\_Days <- countT1$n-1  
countT1 <- countT1[,-2]  
  
countT3 <- subset(countT1, countT1$t7\_incr\_Days>0)  
summary(countT3$t7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.169 8.000 29.000

t7\_incr\_Days\_grouped <- countT3 %>% group\_by(t7\_incr\_Days) %>% count(n=n())  
t7\_incr\_Days\_grouped <- t7\_incr\_Days\_grouped[,-3]

t7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: t7\_incr\_Days [22]  
## t7\_incr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 38  
## 3 3 35  
## 4 4 17  
## 5 5 10  
## 6 6 11  
## 7 7 27  
## 8 8 16  
## 9 9 24  
## 10 10 10  
## # … with 12 more rows

median(t7\_incr\_Days\_grouped$t7\_incr\_Days)

## [1] 11.5

TGT <- subset(remaining36\_g, remaining36\_g$stockName=='TGT')  
  
TGT <- TGT[complete.cases(TGT),]  
  
tgt7\_a <- ifelse(TGT$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for tgt7 <- <- ifelse(TGT$today2lag7>1, 1,0)  
tgt7\_ab <- cumsum(tgt7\_a)  
  
tgt7\_abc <- as.data.frame(as.factor(tgt7\_ab))  
colnames(tgt7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countTGT <- tgt7\_abc %>% group\_by(cSum) %>% count(n=n())  
countTGT <- as.data.frame(countTGT)  
countTGT <- countTGT[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countTGT$tgt7\_decr\_Days <- countTGT$n-1  
countTGT <- countTGT[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countTGT2 <- subset(countTGT, countTGT$tgt7\_decr\_Days>0)  
summary(countTGT2$tgt7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.135 9.000 29.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
tgt7\_decr\_Days\_grouped <- countTGT2 %>% group\_by(tgt7\_decr\_Days) %>% count(n=n())  
tgt7\_decr\_Days\_grouped <- tgt7\_decr\_Days\_grouped[,-3]

tgt7\_decr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: tgt7\_decr\_Days [25]  
## tgt7\_decr\_Days n  
## <dbl> <int>  
## 1 1 75  
## 2 2 36  
## 3 3 20  
## 4 4 23  
## 5 5 18  
## 6 6 10  
## 7 7 17  
## 8 8 12  
## 9 9 12  
## 10 10 9  
## # … with 15 more rows

median(tgt7\_decr\_Days\_grouped$tgt7\_decr\_Days)

## [1] 13

#assign a 1 to decreasing values  
tgt7\_b1 <- ifelse(TGT$today2lag7>1, 0,1)  
  
  
tgt7\_b2 <- cumsum(tgt7\_b1)  
  
tgt7\_b3 <- as.data.frame(as.factor(tgt7\_b2))  
colnames(tgt7\_b3) <- 'cSum'  
  
countTGT1 <- tgt7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countTGT1 <- as.data.frame(countTGT1)  
countTGT1 <- countTGT1[,-3]  
  
countTGT1$tgt7\_incr\_Days <- countTGT1$n-1  
countTGT1 <- countTGT1[,-2]  
  
countTGT3 <- subset(countTGT1, countTGT1$tgt7\_incr\_Days>0)  
summary(countTGT3$tgt7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.324 8.000 23.000

tgt7\_incr\_Days\_grouped <- countTGT3 %>% group\_by(tgt7\_incr\_Days) %>% count(n=n())  
tgt7\_incr\_Days\_grouped <- tgt7\_incr\_Days\_grouped[,-3]

tgt7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: tgt7\_incr\_Days [22]  
## tgt7\_incr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 34  
## 3 3 24  
## 4 4 17  
## 5 5 13  
## 6 6 9  
## 7 7 28  
## 8 8 21  
## 9 9 14  
## 10 10 6  
## # … with 12 more rows

median(tgt7\_incr\_Days\_grouped$tgt7\_incr\_Days)

## [1] 11.5

TM <- subset(remaining36\_g, remaining36\_g$stockName=='TM')  
  
TM <- TM[complete.cases(TM),]  
  
tm7\_a <- ifelse(TM$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for tm7 <- <- ifelse(TM$today2lag7>1, 1,0)  
tm7\_ab <- cumsum(tm7\_a)  
  
tm7\_abc <- as.data.frame(as.factor(tm7\_ab))  
colnames(tm7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countTM <- tm7\_abc %>% group\_by(cSum) %>% count(n=n())  
countTM <- as.data.frame(countTM)  
countTM <- countTM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countTM$tm7\_decr\_Days <- countTM$n-1  
countTM <- countTM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countTM2 <- subset(countTM, countTM$tm7\_decr\_Days>0)  
summary(countTM2$tm7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.535 8.000 36.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
tm7\_decr\_Days\_grouped <- countTM2 %>% group\_by(tm7\_decr\_Days) %>% count(n=n())  
tm7\_decr\_Days\_grouped <- tm7\_decr\_Days\_grouped[,-3]

tm7\_decr\_Days\_grouped

## # A tibble: 29 x 2  
## # Groups: tm7\_decr\_Days [29]  
## tm7\_decr\_Days n  
## <dbl> <int>  
## 1 1 99  
## 2 2 34  
## 3 3 35  
## 4 4 6  
## 5 5 15  
## 6 6 10  
## 7 7 27  
## 8 8 15  
## 9 9 10  
## 10 10 5  
## # … with 19 more rows

median(tm7\_decr\_Days\_grouped$tm7\_decr\_Days)

## [1] 15

#assign a 1 to decreasing values  
tm7\_b1 <- ifelse(TM$today2lag7>1, 0,1)  
  
  
tm7\_b2 <- cumsum(tm7\_b1)  
  
tm7\_b3 <- as.data.frame(as.factor(tm7\_b2))  
colnames(tm7\_b3) <- 'cSum'  
  
countTM1 <- tm7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countTM1 <- as.data.frame(countTM1)  
countTM1 <- countTM1[,-3]  
  
countTM1$tm7\_incr\_Days <- countTM1$n-1  
countTM1 <- countTM1[,-2]  
  
countTM3 <- subset(countTM1, countTM1$tm7\_incr\_Days>0)  
summary(countTM3$tm7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.322 8.000 29.000

tm7\_incr\_Days\_grouped <- countTM3 %>% group\_by(tm7\_incr\_Days) %>% count(n=n())  
tm7\_incr\_Days\_grouped <- tm7\_incr\_Days\_grouped[,-3]

tm7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: tm7\_incr\_Days [22]  
## tm7\_incr\_Days n  
## <dbl> <int>  
## 1 1 87  
## 2 2 44  
## 3 3 22  
## 4 4 17  
## 5 5 16  
## 6 6 11  
## 7 7 23  
## 8 8 18  
## 9 9 13  
## 10 10 7  
## # … with 12 more rows

median(tm7\_incr\_Days\_grouped$tm7\_incr\_Days)

## [1] 11.5

UBSI <- subset(remaining36\_g, remaining36\_g$stockName=='UBSI')  
  
UBSI <- UBSI[complete.cases(UBSI),]  
  
ubsi7\_a <- ifelse(UBSI$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for ubsi7 <- <- ifelse(UBSI$today2lag7>1, 1,0)  
ubsi7\_ab <- cumsum(ubsi7\_a)  
  
ubsi7\_abc <- as.data.frame(as.factor(ubsi7\_ab))  
colnames(ubsi7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countUBSI <- ubsi7\_abc %>% group\_by(cSum) %>% count(n=n())  
countUBSI <- as.data.frame(countUBSI)  
countUBSI <- countUBSI[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countUBSI$ubsi7\_decr\_Days <- countUBSI$n-1  
countUBSI <- countUBSI[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countUBSI2 <- subset(countUBSI, countUBSI$ubsi7\_decr\_Days>0)  
summary(countUBSI2$ubsi7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.253 7.000 41.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
ubsi7\_decr\_Days\_grouped <- countUBSI2 %>% group\_by(ubsi7\_decr\_Days) %>% count(n=n())  
ubsi7\_decr\_Days\_grouped <- ubsi7\_decr\_Days\_grouped[,-3]

ubsi7\_decr\_Days\_grouped

## # A tibble: 23 x 2  
## # Groups: ubsi7\_decr\_Days [23]  
## ubsi7\_decr\_Days n  
## <dbl> <int>  
## 1 1 102  
## 2 2 45  
## 3 3 15  
## 4 4 24  
## 5 5 20  
## 6 6 15  
## 7 7 16  
## 8 8 9  
## 9 9 9  
## 10 10 7  
## # … with 13 more rows

median(ubsi7\_decr\_Days\_grouped$ubsi7\_decr\_Days)

## [1] 12

#assign a 1 to decreasing values  
ubsi7\_b1 <- ifelse(UBSI$today2lag7>1, 0,1)  
  
  
ubsi7\_b2 <- cumsum(ubsi7\_b1)  
  
ubsi7\_b3 <- as.data.frame(as.factor(ubsi7\_b2))  
colnames(ubsi7\_b3) <- 'cSum'  
  
countUBSI1 <- ubsi7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countUBSI1 <- as.data.frame(countUBSI1)  
countUBSI1 <- countUBSI1[,-3]  
  
countUBSI1$ubsi7\_incr\_Days <- countUBSI1$n-1  
countUBSI1 <- countUBSI1[,-2]  
  
countUBSI3 <- subset(countUBSI1, countUBSI1$ubsi7\_incr\_Days>0)  
summary(countUBSI3$ubsi7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.291 7.000 35.000

ubsi7\_incr\_Days\_grouped <- countUBSI3 %>% group\_by(ubsi7\_incr\_Days) %>% count(n=n())  
ubsi7\_incr\_Days\_grouped <- ubsi7\_incr\_Days\_grouped[,-3]

ubsi7\_incr\_Days\_grouped

## # A tibble: 25 x 2  
## # Groups: ubsi7\_incr\_Days [25]  
## ubsi7\_incr\_Days n  
## <dbl> <int>  
## 1 1 97  
## 2 2 45  
## 3 3 21  
## 4 4 20  
## 5 5 13  
## 6 6 12  
## 7 7 28  
## 8 8 11  
## 9 9 17  
## 10 10 2  
## # … with 15 more rows

median(ubsi7\_incr\_Days\_grouped$ubsi7\_incr\_Days)

## [1] 13

VZ <- subset(remaining36\_g, remaining36\_g$stockName=='VZ')  
  
VZ <- VZ[complete.cases(VZ),]  
  
vz7\_a <- ifelse(VZ$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for vz7 <- <- ifelse(VZ$today2lag7>1, 1,0)  
vz7\_ab <- cumsum(vz7\_a)  
  
vz7\_abc <- as.data.frame(as.factor(vz7\_ab))  
colnames(vz7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countVZ <- vz7\_abc %>% group\_by(cSum) %>% count(n=n())  
countVZ <- as.data.frame(countVZ)  
countVZ <- countVZ[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countVZ$vz7\_decr\_Days <- countVZ$n-1  
countVZ <- countVZ[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countVZ2 <- subset(countVZ, countVZ$vz7\_decr\_Days>0)  
summary(countVZ2$vz7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.500 6.096 8.000 30.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
vz7\_decr\_Days\_grouped <- countVZ2 %>% group\_by(vz7\_decr\_Days) %>% count(n=n())  
vz7\_decr\_Days\_grouped <- vz7\_decr\_Days\_grouped[,-3]

vz7\_decr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: vz7\_decr\_Days [27]  
## vz7\_decr\_Days n  
## <dbl> <int>  
## 1 1 76  
## 2 2 47  
## 3 3 18  
## 4 4 24  
## 5 5 14  
## 6 6 3  
## 7 7 18  
## 8 8 13  
## 9 9 6  
## 10 10 9  
## # … with 17 more rows

median(vz7\_decr\_Days\_grouped$vz7\_decr\_Days)

## [1] 14

#assign a 1 to decreasing values  
vz7\_b1 <- ifelse(VZ$today2lag7>1, 0,1)  
  
  
vz7\_b2 <- cumsum(vz7\_b1)  
  
vz7\_b3 <- as.data.frame(as.factor(vz7\_b2))  
colnames(vz7\_b3) <- 'cSum'  
  
countVZ1 <- vz7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countVZ1 <- as.data.frame(countVZ1)  
countVZ1 <- countVZ1[,-3]  
  
countVZ1$vz7\_incr\_Days <- countVZ1$n-1  
countVZ1 <- countVZ1[,-2]  
  
countVZ3 <- subset(countVZ1, countVZ1$vz7\_incr\_Days>0)  
summary(countVZ3$vz7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.569 8.000 32.000

vz7\_incr\_Days\_grouped <- countVZ3 %>% group\_by(vz7\_incr\_Days) %>% count(n=n())  
vz7\_incr\_Days\_grouped <- vz7\_incr\_Days\_grouped[,-3]

vz7\_incr\_Days\_grouped

## # A tibble: 24 x 2  
## # Groups: vz7\_incr\_Days [24]  
## vz7\_incr\_Days n  
## <dbl> <int>  
## 1 1 82  
## 2 2 28  
## 3 3 29  
## 4 4 14  
## 5 5 9  
## 6 6 15  
## 7 7 26  
## 8 8 18  
## 9 9 12  
## 10 10 13  
## # … with 14 more rows

median(vz7\_incr\_Days\_grouped$vz7\_incr\_Days)

## [1] 12.5

WFC <- subset(remaining36\_g, remaining36\_g$stockName=='WFC')  
  
WFC <- WFC[complete.cases(WFC),]  
  
wfc7\_a <- ifelse(WFC$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for wfc7 <- <- ifelse(WFC$today2lag7>1, 1,0)  
wfc7\_ab <- cumsum(wfc7\_a)  
  
wfc7\_abc <- as.data.frame(as.factor(wfc7\_ab))  
colnames(wfc7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countWFC <- wfc7\_abc %>% group\_by(cSum) %>% count(n=n())  
countWFC <- as.data.frame(countWFC)  
countWFC <- countWFC[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countWFC$wfc7\_decr\_Days <- countWFC$n-1  
countWFC <- countWFC[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countWFC2 <- subset(countWFC, countWFC$wfc7\_decr\_Days>0)  
summary(countWFC2$wfc7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 5.536 8.000 46.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
wfc7\_decr\_Days\_grouped <- countWFC2 %>% group\_by(wfc7\_decr\_Days) %>% count(n=n())  
wfc7\_decr\_Days\_grouped <- wfc7\_decr\_Days\_grouped[,-3]

wfc7\_decr\_Days\_grouped

## # A tibble: 26 x 2  
## # Groups: wfc7\_decr\_Days [26]  
## wfc7\_decr\_Days n  
## <dbl> <int>  
## 1 1 85  
## 2 2 40  
## 3 3 30  
## 4 4 21  
## 5 5 13  
## 6 6 12  
## 7 7 31  
## 8 8 18  
## 9 9 14  
## 10 10 10  
## # … with 16 more rows

median(wfc7\_decr\_Days\_grouped$wfc7\_decr\_Days)

## [1] 13.5

#assign a 1 to decreasing values  
wfc7\_b1 <- ifelse(WFC$today2lag7>1, 0,1)  
  
  
wfc7\_b2 <- cumsum(wfc7\_b1)  
  
wfc7\_b3 <- as.data.frame(as.factor(wfc7\_b2))  
colnames(wfc7\_b3) <- 'cSum'  
  
countWFC1 <- wfc7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countWFC1 <- as.data.frame(countWFC1)  
countWFC1 <- countWFC1[,-3]  
  
countWFC1$wfc7\_incr\_Days <- countWFC1$n-1  
countWFC1 <- countWFC1[,-2]  
  
countWFC3 <- subset(countWFC1, countWFC1$wfc7\_incr\_Days>0)  
summary(countWFC3$wfc7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.858 7.000 29.000

wfc7\_incr\_Days\_grouped <- countWFC3 %>% group\_by(wfc7\_incr\_Days) %>% count(n=n())  
wfc7\_incr\_Days\_grouped <- wfc7\_incr\_Days\_grouped[,-3]

wfc7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: wfc7\_incr\_Days [22]  
## wfc7\_incr\_Days n  
## <dbl> <int>  
## 1 1 94  
## 2 2 46  
## 3 3 31  
## 4 4 24  
## 5 5 15  
## 6 6 15  
## 7 7 20  
## 8 8 15  
## 9 9 9  
## 10 10 9  
## # … with 12 more rows

median(wfc7\_incr\_Days\_grouped$wfc7\_incr\_Days)

## [1] 11.5

WM <- subset(remaining36\_g, remaining36\_g$stockName=='WM')  
  
WM <- WM[complete.cases(WM),]  
  
wm7\_a <- ifelse(WM$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for wm7 <- <- ifelse(WM$today2lag7>1, 1,0)  
wm7\_ab <- cumsum(wm7\_a)  
  
wm7\_abc <- as.data.frame(as.factor(wm7\_ab))  
colnames(wm7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countWM <- wm7\_abc %>% group\_by(cSum) %>% count(n=n())  
countWM <- as.data.frame(countWM)  
countWM <- countWM[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countWM$wm7\_decr\_Days <- countWM$n-1  
countWM <- countWM[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countWM2 <- subset(countWM, countWM$wm7\_decr\_Days>0)  
summary(countWM2$wm7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 4.000 6.498 9.000 43.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
wm7\_decr\_Days\_grouped <- countWM2 %>% group\_by(wm7\_decr\_Days) %>% count(n=n())  
wm7\_decr\_Days\_grouped <- wm7\_decr\_Days\_grouped[,-3]

wm7\_decr\_Days\_grouped

## # A tibble: 30 x 2  
## # Groups: wm7\_decr\_Days [30]  
## wm7\_decr\_Days n  
## <dbl> <int>  
## 1 1 75  
## 2 2 40  
## 3 3 24  
## 4 4 14  
## 5 5 14  
## 6 6 8  
## 7 7 27  
## 8 8 13  
## 9 9 11  
## 10 10 11  
## # … with 20 more rows

median(wm7\_decr\_Days\_grouped$wm7\_decr\_Days)

## [1] 15.5

#assign a 1 to decreasing values  
wm7\_b1 <- ifelse(WM$today2lag7>1, 0,1)  
  
  
wm7\_b2 <- cumsum(wm7\_b1)  
  
wm7\_b3 <- as.data.frame(as.factor(wm7\_b2))  
colnames(wm7\_b3) <- 'cSum'  
  
countWM1 <- wm7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countWM1 <- as.data.frame(countWM1)  
countWM1 <- countWM1[,-3]  
  
countWM1$wm7\_incr\_Days <- countWM1$n-1  
countWM1 <- countWM1[,-2]  
  
countWM3 <- subset(countWM1, countWM1$wm7\_incr\_Days>0)  
summary(countWM3$wm7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 4.687 7.000 24.000

wm7\_incr\_Days\_grouped <- countWM3 %>% group\_by(wm7\_incr\_Days) %>% count(n=n())  
wm7\_incr\_Days\_grouped <- wm7\_incr\_Days\_grouped[,-3]

wm7\_incr\_Days\_grouped

## # A tibble: 21 x 2  
## # Groups: wm7\_incr\_Days [21]  
## wm7\_incr\_Days n  
## <dbl> <int>  
## 1 1 107  
## 2 2 38  
## 3 3 20  
## 4 4 19  
## 5 5 14  
## 6 6 11  
## 7 7 27  
## 8 8 12  
## 9 9 8  
## 10 10 8  
## # … with 11 more rows

median(wm7\_incr\_Days\_grouped$wm7\_incr\_Days)

## [1] 11

WWE <- subset(remaining36\_g, remaining36\_g$stockName=='WWE')  
  
WWE <- WWE[complete.cases(WWE),]  
  
wwe7\_a <- ifelse(WWE$today2lag7>1, 1,0)  
  
# get cumulative sum of the number of times today's stock increased from 7 days ago,  
# the day will be repeated if it didn't increase. For example '10' is repeated 21 times, this   
# counts the number of times the value decreased when setting the variable in the previous   
# code block for wwe7 <- <- ifelse(WWE$today2lag7>1, 1,0)  
wwe7\_ab <- cumsum(wwe7\_a)  
  
wwe7\_abc <- as.data.frame(as.factor(wwe7\_ab))  
colnames(wwe7\_abc) <- 'cSum'  
  
# get the count of how many instances or days there are, the more counts, the more   
# days that were decreasing stock values for today's value to 7 days prior value.  
countWWE <- wwe7\_abc %>% group\_by(cSum) %>% count(n=n())  
countWWE <- as.data.frame(countWWE)  
countWWE <- countWWE[,-3]  
  
# remove this additional day, so that the number of days in a row decreasing is measured  
countWWE$wwe7\_decr\_Days <- countWWE$n-1  
countWWE <- countWWE[,-2]  
  
#this is a set of only those days decreasing at least one day in the lag 7 comparison  
countWWE2 <- subset(countWWE, countWWE$wwe7\_decr\_Days>0)  
summary(countWWE2$wwe7\_decr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.933 9.000 54.000

# this table shows how many sets of cumulative days decreasing there were in lag 7  
wwe7\_decr\_Days\_grouped <- countWWE2 %>% group\_by(wwe7\_decr\_Days) %>% count(n=n())  
wwe7\_decr\_Days\_grouped <- wwe7\_decr\_Days\_grouped[,-3]

wwe7\_decr\_Days\_grouped

## # A tibble: 27 x 2  
## # Groups: wwe7\_decr\_Days [27]  
## wwe7\_decr\_Days n  
## <dbl> <int>  
## 1 1 83  
## 2 2 47  
## 3 3 25  
## 4 4 16  
## 5 5 12  
## 6 6 7  
## 7 7 18  
## 8 8 14  
## 9 9 12  
## 10 10 7  
## # … with 17 more rows

median(wwe7\_decr\_Days\_grouped$wwe7\_decr\_Days)

## [1] 14

#assign a 1 to decreasing values  
wwe7\_b1 <- ifelse(WWE$today2lag7>1, 0,1)  
  
  
wwe7\_b2 <- cumsum(wwe7\_b1)  
  
wwe7\_b3 <- as.data.frame(as.factor(wwe7\_b2))  
colnames(wwe7\_b3) <- 'cSum'  
  
countWWE1 <- wwe7\_b3 %>% group\_by(cSum) %>% count(n=n())  
countWWE1 <- as.data.frame(countWWE1)  
countWWE1 <- countWWE1[,-3]  
  
countWWE1$wwe7\_incr\_Days <- countWWE1$n-1  
countWWE1 <- countWWE1[,-2]  
  
countWWE3 <- subset(countWWE1, countWWE1$wwe7\_incr\_Days>0)  
summary(countWWE3$wwe7\_incr\_Days)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 5.124 7.000 26.000

wwe7\_incr\_Days\_grouped <- countWWE3 %>% group\_by(wwe7\_incr\_Days) %>% count(n=n())  
wwe7\_incr\_Days\_grouped <- wwe7\_incr\_Days\_grouped[,-3]

wwe7\_incr\_Days\_grouped

## # A tibble: 22 x 2  
## # Groups: wwe7\_incr\_Days [22]  
## wwe7\_incr\_Days n  
## <dbl> <int>  
## 1 1 95  
## 2 2 36  
## 3 3 20  
## 4 4 20  
## 5 5 15  
## 6 6 14  
## 7 7 26  
## 8 8 9  
## 9 9 12  
## 10 10 10  
## # … with 12 more rows

median(wwe7\_incr\_Days\_grouped$wwe7\_incr\_Days)

## [1] 11.5

Start making the table of statistics for the remaining 36 stocks, using the first set of 17 that were in the best performing portfolio compared to the DOW.

This gets the average and the median stock values from the start and end of this time series from Jan 3, 2007 to Feb 14, 2020.

close53\_avg <- Close53\_g %>% group\_by(stockName) %>% summarise\_at(vars(stockDayValue),  
 mean)  
close53\_median <- Close53\_g %>% group\_by(stockName) %>% summarise\_at(vars(stockDayValue),  
 median)  
colnames(close53\_avg)[2] <- 'avgStockValue'  
colnames(close53\_median)[2] <- 'medianStockValue'

first17 <- dROI17\_lag7\_d[,-c(1,4)]  
row.names(first17) <- row.names(dROI17\_lag7\_d)  
  
Remaining36 <- remaining36[order(remaining36$Date),]  
  
remaining36\_st\_fi <- Remaining36[c(1,length(remaining36$Date)),]  
remaining36\_st\_fi\_t <- as.data.frame(t(remaining36\_st\_fi))  
colnames(remaining36\_st\_fi\_t) <- c('startValue', 'finalValue')  
  
remaining36\_st\_fi\_t <- remaining36\_st\_fi\_t[-1,]  
last36 <- remaining36\_st\_fi\_t  
  
last36$startValue <- as.numeric(as.character(last36$startValue))  
last36$finalValue <- as.numeric(as.character(last36$finalValue))  
  
last36$stock\_ROI <- last36$finalValue/last36$startValue  
  
last36$medn\_cSum\_decr\_L7 <- 'median cSum down L7'  
last36$Q3\_cSum\_decr\_L7 <- '3rd Qntl cSum down L7'  
last36$max\_cSum\_decr\_L7 <- 'max cSum down L7'  
last36$medn\_cSum\_incr\_L7 <- 'median cSum up L7'  
last36$Q3\_cSum\_incr\_L7 <- '3rd Qntl cSun up L7'  
last36$max\_cSum\_incr\_L7 <- 'max cSum up L7'  
  
all53 <- rbind(first17, last36)  
  
all53$stockName <- row.names(all53)  
  
all\_53 <- merge(close53\_avg, all53, by.x='stockName', by.y='stockName')  
All53 <- merge(close53\_median, all\_53, by.x='stockName', by.y='stockName')  
head(All53)

## stockName medianStockValue avgStockValue startValue finalValue stock\_ROI  
## 1 AAL 28.08 25.81191 56.30 29.20 0.5186501  
## 2 AAP 84.59 97.31300 35.58 133.59 3.7500000  
## 3 ADDYY 39.99 58.44379 25.00 156.11 6.2400000  
## 4 AMZN 288.80 550.53223 38.70 2134.87 55.1600000  
## 5 ARWR 6.87 13.18195 44.00 41.27 0.9379545  
## 6 C 51.59 93.33454 552.50 78.79 0.1426063  
## medn\_cSum\_decr\_L7 Q3\_cSum\_decr\_L7 max\_cSum\_decr\_L7 medn\_cSum\_incr\_L7  
## 1 median cSum down L7 3rd Qntl cSum down L7 max cSum down L7 median cSum up L7  
## 2 3 9 44 3  
## 3 5 10 35 3  
## 4 5 11 31 3  
## 5 median cSum down L7 3rd Qntl cSum down L7 max cSum down L7 median cSum up L7  
## 6 median cSum down L7 3rd Qntl cSum down L7 max cSum down L7 median cSum up L7  
## Q3\_cSum\_incr\_L7 max\_cSum\_incr\_L7  
## 1 3rd Qntl cSun up L7 max cSum up L7  
## 2 7 30  
## 3 8 25  
## 4 8 26  
## 5 3rd Qntl cSun up L7 max cSum up L7  
## 6 3rd Qntl cSun up L7 max cSum up L7

Add these statistics to their respective stock instance in the All53 table now that we have the table, there are 36 stock statistics on the number of times the stock increases and decreases in this time span that need to be filled in. Carefully find and replace the names of each of the remaining 36 stock for the lower and upper instances in the block below. Then after running each chunk of code, the All53 table will be filled in.

AAL-American Airlines:

mdnaal\_d <- median(countAAL2$aal7\_decr\_Days)  
q3aal\_d <- as.numeric(as.character(summary(countAAL2$aal7\_decr\_Days)["3rd Qu."]))  
mxaal\_d <- max(countAAL2$aal7\_decr\_Days)  
  
mdnaal\_i <- median(countAAL3$aal7\_incr\_Days)  
q3aal\_i <- as.numeric(as.character(summary(countAAL3$aal7\_incr\_Days)["3rd Qu."]))  
mxaal\_i <- max(countAAL3$aal7\_incr\_Days)  
  
aal\_g <- grep('AAL', All53$stockName )  
  
All53[aal\_g,7:12] <- c(mdnaal\_d,q3aal\_d,mxaal\_d,mdnaal\_i,q3aal\_i,mxaal\_i)

The above code chunk filled in the AAL stock statistical information.

head(All53)

## stockName medianStockValue avgStockValue startValue finalValue stock\_ROI  
## 1 AAL 28.08 25.81191 56.30 29.20 0.5186501  
## 2 AAP 84.59 97.31300 35.58 133.59 3.7500000  
## 3 ADDYY 39.99 58.44379 25.00 156.11 6.2400000  
## 4 AMZN 288.80 550.53223 38.70 2134.87 55.1600000  
## 5 ARWR 6.87 13.18195 44.00 41.27 0.9379545  
## 6 C 51.59 93.33454 552.50 78.79 0.1426063  
## medn\_cSum\_decr\_L7 Q3\_cSum\_decr\_L7 max\_cSum\_decr\_L7 medn\_cSum\_incr\_L7  
## 1 4 10 32 5  
## 2 3 9 44 3  
## 3 5 10 35 3  
## 4 5 11 31 3  
## 5 median cSum down L7 3rd Qntl cSum down L7 max cSum down L7 median cSum up L7  
## 6 median cSum down L7 3rd Qntl cSum down L7 max cSum down L7 median cSum up L7  
## Q3\_cSum\_incr\_L7 max\_cSum\_incr\_L7  
## 1 9 30  
## 2 7 30  
## 3 8 25  
## 4 8 26  
## 5 3rd Qntl cSun up L7 max cSum up L7  
## 6 3rd Qntl cSun up L7 max cSum up L7

Now to fill in the other 35 stocks for our completed ML table of stats (for now).

ARWR-Arrowhead Pharmaceuticals:

mdnarwr\_d <- median(countARWR2$arwr7\_decr\_Days)  
q3arwr\_d <- as.numeric(as.character(summary(countARWR2$arwr7\_decr\_Days)["3rd Qu."]))  
mxarwr\_d <- max(countARWR2$arwr7\_decr\_Days)  
  
mdnarwr\_i <- median(countARWR3$arwr7\_incr\_Days)  
q3arwr\_i <- as.numeric(as.character(summary(countARWR3$arwr7\_incr\_Days)["3rd Qu."]))  
mxarwr\_i <- max(countARWR3$arwr7\_incr\_Days)  
  
arwr\_g <- grep('ARWR', All53$stockName )  
  
All53[arwr\_g,7:12] <- c(mdnarwr\_d,q3arwr\_d,mxarwr\_d,mdnarwr\_i,q3arwr\_i,mxarwr\_i)

C-Citigroup:

mdnc\_d <- median(countC2$c7\_decr\_Days)  
q3c\_d <- as.numeric(as.character(summary(countC2$c7\_decr\_Days)["3rd Qu."]))  
mxc\_d <- max(countC2$c7\_decr\_Days)  
  
mdnc\_i <- median(countC3$c7\_incr\_Days)  
q3c\_i <- as.numeric(as.character(summary(countC3$c7\_incr\_Days)["3rd Qu."]))  
mxc\_i <- max(countC3$c7\_incr\_Days)  
  
c\_g <- grep('C', All53$stockName )  
  
All53[c\_g,7:12] <- c(mdnc\_d,q3c\_d,mxc\_d,mdnc\_i,q3c\_i,mxc\_i)

EPD-Enterprise Products Partners:

mdnepd\_d <- median(countEPD2$epd7\_decr\_Days)  
q3epd\_d <- as.numeric(as.character(summary(countEPD2$epd7\_decr\_Days)["3rd Qu."]))  
mxepd\_d <- max(countEPD2$epd7\_decr\_Days)  
  
mdnepd\_i <- median(countEPD3$epd7\_incr\_Days)  
q3epd\_i <- as.numeric(as.character(summary(countEPD3$epd7\_incr\_Days)["3rd Qu."]))  
mxepd\_i <- max(countEPD3$epd7\_incr\_Days)  
  
epd\_g <- grep('EPD', All53$stockName )  
  
All53[epd\_g,7:12] <- c(mdnepd\_d,q3epd\_d,mxepd\_d,mdnepd\_i,q3epd\_i,mxepd\_i)

F-Ford:

mdnf\_d <- median(countF2$f7\_decr\_Days)  
q3f\_d <- as.numeric(as.character(summary(countF2$f7\_decr\_Days)["3rd Qu."]))  
mxf\_d <- max(countF2$f7\_decr\_Days)  
  
mdnf\_i <- median(countF3$f7\_incr\_Days)  
q3f\_i <- as.numeric(as.character(summary(countF3$f7\_incr\_Days)["3rd Qu."]))  
mxf\_i <- max(countF3$f7\_incr\_Days)  
  
f\_g <- grep('F', All53$stockName )  
  
All53[f\_g,7:12] <- c(mdnf\_d,q3f\_d,mxf\_d,mdnf\_i,q3f\_i,mxf\_i)

HRB-H&R Block:

mdnhrb\_d <- median(countHRB2$hrb7\_decr\_Days)  
q3hrb\_d <- as.numeric(as.character(summary(countHRB2$hrb7\_decr\_Days)["3rd Qu."]))  
mxhrb\_d <- max(countHRB2$hrb7\_decr\_Days)  
  
mdnhrb\_i <- median(countHRB3$hrb7\_incr\_Days)  
q3hrb\_i <- as.numeric(as.character(summary(countHRB3$hrb7\_incr\_Days)["3rd Qu."]))  
mxhrb\_i <- max(countHRB3$hrb7\_incr\_Days)  
  
hrb\_g <- grep('HRB', All53$stockName )  
  
All53[hrb\_g,7:12] <- c(mdnhrb\_d,q3hrb\_d,mxhrb\_d,mdnhrb\_i,q3hrb\_i,mxhrb\_i)

HST-Host Hotels & Resorts:

mdnhst\_d <- median(countHST2$hst7\_decr\_Days)  
q3hst\_d <- as.numeric(as.character(summary(countHST2$hst7\_decr\_Days)["3rd Qu."]))  
mxhst\_d <- max(countHST2$hst7\_decr\_Days)  
  
mdnhst\_i <- median(countHST3$hst7\_incr\_Days)  
q3hst\_i <- as.numeric(as.character(summary(countHST3$hst7\_incr\_Days)["3rd Qu."]))  
mxhst\_i <- max(countHST3$hst7\_incr\_Days)  
  
hst\_g <- grep('HST', All53$stockName )  
  
All53[hst\_g,7:12] <- c(mdnhst\_d,q3hst\_d,mxhst\_d,mdnhst\_i,q3hst\_i,mxhst\_i)

INO-Inovio Pharmaceuticals:

mdnino\_d <- median(countINO2$ino7\_decr\_Days)  
q3ino\_d <- as.numeric(as.character(summary(countINO2$ino7\_decr\_Days)["3rd Qu."]))  
mxino\_d <- max(countINO2$ino7\_decr\_Days)  
  
mdnino\_i <- median(countINO3$ino7\_incr\_Days)  
q3ino\_i <- as.numeric(as.character(summary(countINO3$ino7\_incr\_Days)["3rd Qu."]))  
mxino\_i <- max(countINO3$ino7\_incr\_Days)  
  
ino\_g <- grep('INO', All53$stockName )  
  
All53[ino\_g,7:12] <- c(mdnino\_d,q3ino\_d,mxino\_d,mdnino\_i,q3ino\_i,mxino\_i)

JBLU-Jet Blue Airways:

mdnjblu\_d <- median(countJBLU2$jblu7\_decr\_Days)  
q3jblu\_d <- as.numeric(as.character(summary(countJBLU2$jblu7\_decr\_Days)["3rd Qu."]))  
mxjblu\_d <- max(countJBLU2$jblu7\_decr\_Days)  
  
mdnjblu\_i <- median(countJBLU3$jblu7\_incr\_Days)  
q3jblu\_i <- as.numeric(as.character(summary(countJBLU3$jblu7\_incr\_Days)["3rd Qu."]))  
mxjblu\_i <- max(countJBLU3$jblu7\_incr\_Days)  
  
jblu\_g <- grep('JBLU', All53$stockName )  
  
All53[jblu\_g,7:12] <- c(mdnjblu\_d,q3jblu\_d,mxjblu\_d,mdnjblu\_i,q3jblu\_i,mxjblu\_i)

JPM-JP Morgan Chase & Co.:

mdnjpm\_d <- median(countJPM2$jpm7\_decr\_Days)  
q3jpm\_d <- as.numeric(as.character(summary(countJPM2$jpm7\_decr\_Days)["3rd Qu."]))  
mxjpm\_d <- max(countJPM2$jpm7\_decr\_Days)  
  
mdnjpm\_i <- median(countJPM3$jpm7\_incr\_Days)  
q3jpm\_i <- as.numeric(as.character(summary(countJPM3$jpm7\_incr\_Days)["3rd Qu."]))  
mxjpm\_i <- max(countJPM3$jpm7\_incr\_Days)  
  
jpm\_g <- grep('JPM', All53$stockName )  
  
All53[jpm\_g,7:12] <- c(mdnjpm\_d,q3jpm\_d,mxjpm\_d,mdnjpm\_i,q3jpm\_i,mxjpm\_i)

JWN-Nordstrom:

mdnjwn\_d <- median(countJWN2$jwn7\_decr\_Days)  
q3jwn\_d <- as.numeric(as.character(summary(countJWN2$jwn7\_decr\_Days)["3rd Qu."]))  
mxjwn\_d <- max(countJWN2$jwn7\_decr\_Days)  
  
mdnjwn\_i <- median(countJWN3$jwn7\_incr\_Days)  
q3jwn\_i <- as.numeric(as.character(summary(countJWN3$jwn7\_incr\_Days)["3rd Qu."]))  
mxjwn\_i <- max(countJWN3$jwn7\_incr\_Days)  
  
jwn\_g <- grep('JWN', All53$stockName )  
  
All53[jwn\_g,7:12] <- c(mdnjwn\_d,q3jwn\_d,mxjwn\_d,mdnjwn\_i,q3jwn\_i,mxjwn\_i)

KGJI-Kingold Jewelry,Inc:

mdnkgji\_d <- median(countKGJI2$kgji7\_decr\_Days)  
q3kgji\_d <- as.numeric(as.character(summary(countKGJI2$kgji7\_decr\_Days)["3rd Qu."]))  
mxkgji\_d <- max(countKGJI2$kgji7\_decr\_Days)  
  
mdnkgji\_i <- median(countKGJI3$kgji7\_incr\_Days)  
q3kgji\_i <- as.numeric(as.character(summary(countKGJI3$kgji7\_incr\_Days)["3rd Qu."]))  
mxkgji\_i <- max(countKGJI3$kgji7\_incr\_Days)  
  
kgji\_g <- grep('KGJI', All53$stockName )  
  
All53[kgji\_g,7:12] <- c(mdnkgji\_d,q3kgji\_d,mxkgji\_d,mdnkgji\_i,q3kgji\_i,mxkgji\_i)

KSS-Kohls Corporation:

mdnkss\_d <- median(countKSS2$kss7\_decr\_Days)  
q3kss\_d <- as.numeric(as.character(summary(countKSS2$kss7\_decr\_Days)["3rd Qu."]))  
mxkss\_d <- max(countKSS2$kss7\_decr\_Days)  
  
mdnkss\_i <- median(countKSS3$kss7\_incr\_Days)  
q3kss\_i <- as.numeric(as.character(summary(countKSS3$kss7\_incr\_Days)["3rd Qu."]))  
mxkss\_i <- max(countKSS3$kss7\_incr\_Days)  
  
kss\_g <- grep('KSS', All53$stockName )  
  
All53[kss\_g,7:12] <- c(mdnkss\_d,q3kss\_d,mxkss\_d,mdnkss\_i,q3kss\_i,mxkss\_i)

LUV-Southwest Airlines:

mdnluv\_d <- median(countLUV2$luv7\_decr\_Days)  
q3luv\_d <- as.numeric(as.character(summary(countLUV2$luv7\_decr\_Days)["3rd Qu."]))  
mxluv\_d <- max(countLUV2$luv7\_decr\_Days)  
  
mdnluv\_i <- median(countLUV3$luv7\_incr\_Days)  
q3luv\_i <- as.numeric(as.character(summary(countLUV3$luv7\_incr\_Days)["3rd Qu."]))  
mxluv\_i <- max(countLUV3$luv7\_incr\_Days)  
  
luv\_g <- grep('LUV', All53$stockName )  
  
All53[luv\_g,7:12] <- c(mdnluv\_d,q3luv\_d,mxluv\_d,mdnluv\_i,q3luv\_i,mxluv\_i)

M-Macys:

mdnm\_d <- median(countM2$m7\_decr\_Days)  
q3m\_d <- as.numeric(as.character(summary(countM2$m7\_decr\_Days)["3rd Qu."]))  
mxm\_d <- max(countM2$m7\_decr\_Days)  
  
mdnm\_i <- median(countM3$m7\_incr\_Days)  
q3m\_i <- as.numeric(as.character(summary(countM3$m7\_incr\_Days)["3rd Qu."]))  
mxm\_i <- max(countM3$m7\_incr\_Days)  
  
m\_g <- grep('M', All53$stockName )  
  
All53[m\_g,7:12] <- c(mdnm\_d,q3m\_d,mxm\_d,mdnm\_i,q3m\_i,mxm\_i)

NSANY-Nissan Motor Co.:

mdnnsany\_d <- median(countNSANY2$nsany7\_decr\_Days)  
q3nsany\_d <- as.numeric(as.character(summary(countNSANY2$nsany7\_decr\_Days)["3rd Qu."]))  
mxnsany\_d <- max(countNSANY2$nsany7\_decr\_Days)  
  
mdnnsany\_i <- median(countNSANY3$nsany7\_incr\_Days)  
q3nsany\_i <- as.numeric(as.character(summary(countNSANY3$nsany7\_incr\_Days)["3rd Qu."]))  
mxnsany\_i <- max(countNSANY3$nsany7\_incr\_Days)  
  
nsany\_g <- grep('NSANY', All53$stockName )  
  
All53[nsany\_g,7:12] <- c(mdnnsany\_d,q3nsany\_d,mxnsany\_d,mdnnsany\_i,q3nsany\_i,mxnsany\_i)

NUS-Nu Skin Enterprises:

mdnnus\_d <- median(countNUS2$nus7\_decr\_Days)  
q3nus\_d <- as.numeric(as.character(summary(countNUS2$nus7\_decr\_Days)["3rd Qu."]))  
mxnus\_d <- max(countNUS2$nus7\_decr\_Days)  
  
mdnnus\_i <- median(countNUS3$nus7\_incr\_Days)  
q3nus\_i <- as.numeric(as.character(summary(countNUS3$nus7\_incr\_Days)["3rd Qu."]))  
mxnus\_i <- max(countNUS3$nus7\_incr\_Days)  
  
nus\_g <- grep('NUS', All53$stockName )  
  
All53[nus\_g,7:12] <- c(mdnnus\_d,q3nus\_d,mxnus\_d,mdnnus\_i,q3nus\_i,mxnus\_i)

RRGB-Red Robin Gourmet Burgers:

mdnrrgb\_d <- median(countRRGB2$rrgb7\_decr\_Days)  
q3rrgb\_d <- as.numeric(as.character(summary(countRRGB2$rrgb7\_decr\_Days)["3rd Qu."]))  
mxrrgb\_d <- max(countRRGB2$rrgb7\_decr\_Days)  
  
mdnrrgb\_i <- median(countRRGB3$rrgb7\_incr\_Days)  
q3rrgb\_i <- as.numeric(as.character(summary(countRRGB3$rrgb7\_incr\_Days)["3rd Qu."]))  
mxrrgb\_i <- max(countRRGB3$rrgb7\_incr\_Days)  
  
rrgb\_g <- grep('RRGB', All53$stockName )  
  
All53[rrgb\_g,7:12] <- c(mdnrrgb\_d,q3rrgb\_d,mxrrgb\_d,mdnrrgb\_i,q3rrgb\_i,mxrrgb\_i)

S-Sprint Corporation:

mdns\_d <- median(countS2$s7\_decr\_Days)  
q3s\_d <- as.numeric(as.character(summary(countS2$s7\_decr\_Days)["3rd Qu."]))  
mxs\_d <- max(countS2$s7\_decr\_Days)  
  
mdns\_i <- median(countS3$s7\_incr\_Days)  
q3s\_i <- as.numeric(as.character(summary(countS3$s7\_incr\_Days)["3rd Qu."]))  
mxs\_i <- max(countS3$s7\_incr\_Days)  
  
s\_g <- grep('S', All53$stockName )  
  
All53[s\_g,7:12] <- c(mdns\_d,q3s\_d,mxs\_d,mdns\_i,q3s\_i,mxs\_i)

T-AT&T Inc:

mdnt\_d <- median(countT2$t7\_decr\_Days)  
q3t\_d <- as.numeric(as.character(summary(countT2$t7\_decr\_Days)["3rd Qu."]))  
mxt\_d <- max(countT2$t7\_decr\_Days)  
  
mdnt\_i <- median(countT3$t7\_incr\_Days)  
q3t\_i <- as.numeric(as.character(summary(countT3$t7\_incr\_Days)["3rd Qu."]))  
mxt\_i <- max(countT3$t7\_incr\_Days)  
  
t\_g <- grep('T', All53$stockName )  
  
All53[t\_g,7:12] <- c(mdnt\_d,q3t\_d,mxt\_d,mdnt\_i,q3t\_i,mxt\_i)

VZ-Verizon:

mdnvz\_d <- median(countVZ2$vz7\_decr\_Days)  
q3vz\_d <- as.numeric(as.character(summary(countVZ2$vz7\_decr\_Days)["3rd Qu."]))  
mxvz\_d <- max(countVZ2$vz7\_decr\_Days)  
  
mdnvz\_i <- median(countVZ3$vz7\_incr\_Days)  
q3vz\_i <- as.numeric(as.character(summary(countVZ3$vz7\_incr\_Days)["3rd Qu."]))  
mxvz\_i <- max(countVZ3$vz7\_incr\_Days)  
  
vz\_g <- grep('VZ', All53$stockName )  
  
All53[vz\_g,7:12] <- c(mdnvz\_d,q3vz\_d,mxvz\_d,mdnvz\_i,q3vz\_i,mxvz\_i)

WWE-World Wrestling Entertainment:

mdnwwe\_d <- median(countWWE2$wwe7\_decr\_Days)  
q3wwe\_d <- as.numeric(as.character(summary(countWWE2$wwe7\_decr\_Days)["3rd Qu."]))  
mxwwe\_d <- max(countWWE2$wwe7\_decr\_Days)  
  
mdnwwe\_i <- median(countWWE3$wwe7\_incr\_Days)  
q3wwe\_i <- as.numeric(as.character(summary(countWWE3$wwe7\_incr\_Days)["3rd Qu."]))  
mxwwe\_i <- max(countWWE3$wwe7\_incr\_Days)  
  
wwe\_g <- grep('WWE', All53$stockName )  
  
All53[wwe\_g,7:12] <- c(mdnwwe\_d,q3wwe\_d,mxwwe\_d,mdnwwe\_i,q3wwe\_i,mxwwe\_i)

Now, lets look at the columns to see they are the right data types.

str(All53)

## 'data.frame': 53 obs. of 12 variables:  
## $ stockName : Factor w/ 53 levels "AAL","AAP","ADDYY",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ medianStockValue : num 28.08 84.59 39.99 288.8 6.87 ...  
## $ avgStockValue : num 25.8 97.3 58.4 550.5 13.2 ...  
## $ startValue : num 56.3 35.6 25 38.7 44 ...  
## $ finalValue : num 29.2 133.6 156.1 2134.9 41.3 ...  
## $ stock\_ROI : num 0.519 3.75 6.24 55.16 0.938 ...  
## $ medn\_cSum\_decr\_L7: chr "4" "3" "5" "4" ...  
## $ Q3\_cSum\_decr\_L7 : chr "10" "9" "10" "8" ...  
## $ max\_cSum\_decr\_L7 : chr "32" "44" "35" "38" ...  
## $ medn\_cSum\_incr\_L7: chr "5" "3" "3" "4" ...  
## $ Q3\_cSum\_incr\_L7 : chr "9" "7" "8" "8" ...  
## $ max\_cSum\_incr\_L7 : chr "30" "30" "25" "38" ...

We need to change the stat columns in the All53 table from char to numeric. So lets do that.

All53$medn\_cSum\_decr\_L7 <- as.numeric(All53$medn\_cSum\_decr\_L7)  
All53$Q3\_cSum\_decr\_L7 <- as.numeric(All53$Q3\_cSum\_decr\_L7)  
All53$max\_cSum\_decr\_L7 <- as.numeric(All53$max\_cSum\_decr\_L7)  
All53$medn\_cSum\_incr\_L7 <- as.numeric(All53$medn\_cSum\_incr\_L7)  
All53$Q3\_cSum\_incr\_L7 <- as.numeric(All53$Q3\_cSum\_incr\_L7)  
All53$max\_cSum\_incr\_L7 <- as.numeric(All53$max\_cSum\_incr\_L7)  
  
str(All53)

## 'data.frame': 53 obs. of 12 variables:  
## $ stockName : Factor w/ 53 levels "AAL","AAP","ADDYY",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ medianStockValue : num 28.08 84.59 39.99 288.8 6.87 ...  
## $ avgStockValue : num 25.8 97.3 58.4 550.5 13.2 ...  
## $ startValue : num 56.3 35.6 25 38.7 44 ...  
## $ finalValue : num 29.2 133.6 156.1 2134.9 41.3 ...  
## $ stock\_ROI : num 0.519 3.75 6.24 55.16 0.938 ...  
## $ medn\_cSum\_decr\_L7: num 4 3 5 4 4 4 3 26 8 5 ...  
## $ Q3\_cSum\_decr\_L7 : num 10 9 10 8 8 26 8 8 35 8 ...  
## $ max\_cSum\_decr\_L7 : num 32 44 35 38 33 8 35 4 3 42 ...  
## $ medn\_cSum\_incr\_L7: num 5 3 3 4 4 4 3 26 8 4 ...  
## $ Q3\_cSum\_incr\_L7 : num 9 7 8 8 9 26 8 8 29 8 ...  
## $ max\_cSum\_incr\_L7 : num 30 30 25 38 40 8 29 4 3 26 ...

Lets also add the name of the stock to the stock name. We will also make a table of the total stocks in the hand picked portfolio of 65 that had many missing observations in the yahoo finance grab of stock information from 2007-2020 to start with.

names53 <- stockNames[,1:2]  
names53$stock <- gsub('.PB','', names53$stock)  
names53$stock <- as.factor(names53$stock)  
  
ALL53 <- merge(names53, All53, by.x='stock', by.y='stockName')  
ALL65 <- merge(names53, All53, by.x='stock', by.y='stockName', all.x=TRUE)  
  
ALL65 <- ALL65[order(ALL65$medn\_cSum\_decr\_L7, na.last=TRUE),]  
na12stocks <- ALL65[54:65,]

Before we write these to file, lets also categorize these stocks by type of stock. Such as cell phone, airlines, restaurants, etc. To get a category measure of each stock stats within their category or business category.

businessType <- as.data.frame(c('airTravel','autoParts','athleticShoes','cloudComputing',  
 'genotypePharmaceuticals',  
 'bank','retail','fuel','discountRetail','fuel','automotive','bank',  
 'communications','cloudComputing','homeRepair','automotive',  
 'furniture','taxService', 'hotelTravel','immunoPharmaceuticals', 'airTravel',  
 'healthCarePharmaceuticals', 'bank','retail', 'jewelry',   
 'retail', 'airTravel', 'retail',  
 'hotelTravel', 'CloudComputing', 'subsriptionEntertainment', 'athleticShoes',  
 'automotive','beautyProducts','cancerPharmaceuticals', 'energyUtility',  
 'discountRetail','restaurant','mobileCommunications','energyUtility',  
 'jewelry', 'communications','genericPharmaceuticals','retail','discountRetail',  
 'automotive','bank','mobileCommunications', 'bank','wasteUtility',  
 'discountRetail', 'sportsEntertainment','fuel'))  
colnames(businessType) <- 'businessType'  
  
businessType2 <- as.data.frame(c('movieEntertainment','athleticShoes','movieEntertainment',  
 'automotive','automotive','immunoPharmaceuticals',  
 'couponDeals','cancerPharmaceuticals',  
 'fastFoodRestaurant','dentalBioTech','mobileCommunications',  
 'businessSocialMedia'))  
colnames(businessType2) <- 'businessType'  
other12 <- cbind(businessType2,na12stocks)  
other12 <- other12[,c(2,3,1,4:14)]

ALL\_53 <- cbind(businessType,ALL53)  
ALL\_53 <- ALL\_53[,c(2,3,1,4:14)]  
  
ALL\_65 <- rbind(ALL\_53,other12)  
  
write.csv(ALL\_53, 'ALL\_53.csv', row.names=FALSE)  
write.csv(ALL\_65, 'ALL\_65.csv', row.names=FALSE)

I noticed early in this script that SCE belongs to the other data set because of a class type that converted to numeric using the factors. That was why the plot early on of SCE was so choppy. Lets make a new data set to corresond to this new information.

sce1 <- grep('SCE', ALL\_53$stock)  
ALL\_52 <- ALL\_53[-sce1,]  
ALL\_65\_b <- ALL\_65  
ALL\_65\_b[sce1,3:13] <- NA  
ALL\_65\_b <- ALL\_65\_b[order(ALL\_65\_b$stock,na.last=TRUE),]

write.csv(ALL\_65\_b, 'ALL\_65\_b.csv', row.names=FALSE)  
write.csv(ALL\_52, 'ALL\_52.csv', row.names=FALSE)

This R markdown file shows all the work to gather statistical information on counts and other behind the scenes stock information on 52 hand picked stocks with time series information from Jan 2007- Feb 2020 compared to the 17 stock information of the same in the ROI\_HandPickedStocks.Rmd file.

Lets use the data set we created in the ROI-HandPickedStocks.Rmd file, ALL52 data set, in the ‘ALL\_52.csv’ file to see how well the machine learning does on this data frame.

ALL\_52 <- read.csv('ALL\_52.csv', sep=',', header=TRUE, na.strings=c('',' '))

Lets first get the median value for these stocks’ ROI.

colnames(ALL\_52)

## [1] "stock" "stockInfo" "businessType"   
## [4] "medianStockValue" "avgStockValue" "startValue"   
## [7] "finalValue" "stock\_ROI" "medn\_cSum\_decr\_L7"  
## [10] "Q3\_cSum\_decr\_L7" "max\_cSum\_decr\_L7" "medn\_cSum\_incr\_L7"  
## [13] "Q3\_cSum\_incr\_L7" "max\_cSum\_incr\_L7"

Lets add in some columns features for classifying this data. One to show if the stock has a low or high ROI based on the median ROI for these 52 stocks, one to show if the stock decreases more or less than the median number of times the stock has decreased from 2007-2020, and one to show if the stock increases more or less than the median number of times all stocks increased.

med52ROI <- median(ALL\_52$stock\_ROI)  
med52ROI

## [1] 1.374265

ALL\_52$ROI\_Low\_High <- ifelse(ALL\_52$stock\_ROI > med52ROI,'High',  
 'Low')  
med52Decr <- median(ALL\_52$medn\_cSum\_decr\_L7)  
med52Decr

## [1] 4

ALL\_52$MedCountsDecreasing <- ifelse(ALL\_52$medn\_cSum\_decr\_L7 > med52Decr,  
 'High Decreasing Counts',  
 'Low Decreasing Counts')  
  
med52Incr <- median(ALL\_52$medn\_cSum\_incr\_L7)  
med52Incr

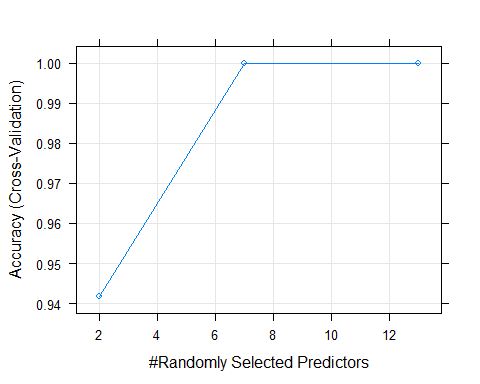
## [1] 4

ALL\_52$MedCountsIncreasing <- ifelse(ALL\_52$medn\_cSum\_incr\_L7 > med52Incr,   
 'High Increasing Counts',   
 'Low Increase Counts')

row.names(ALL\_52) <- ALL\_52$stock  
ALL\_52\_ML <- ALL\_52[,-c(1:3)]  
  
  
write.csv(ALL\_52, 'ALL\_52\_ml.csv', row.names=TRUE)  
write.csv(ALL\_52\_ML, 'ALL\_52\_ML1.csv', row.names=TRUE)

set.seed(12356789)  
  
inTrain <- createDataPartition(y=ALL\_52\_ML$ROI\_Low\_High, p=0.7, list=FALSE)  
  
trainingSet <- ALL\_52\_ML[inTrain,]  
testingSet <- ALL\_52\_ML[-inTrain,]

rfMod <- train(ROI\_Low\_High~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5)  
plot(rfMod)



predRF <- predict(rfMod, testingSet)  
  
predDF <- data.frame(predRF, type=testingSet$ROI\_Low\_High)  
predDF

## predRF type  
## 1 High High  
## 2 High High  
## 3 Low Low  
## 4 Low Low  
## 5 High High  
## 6 High High  
## 7 High High  
## 8 Low Low  
## 9 Low Low  
## 10 High High  
## 11 Low Low  
## 12 Low Low  
## 13 High High  
## 14 High Low

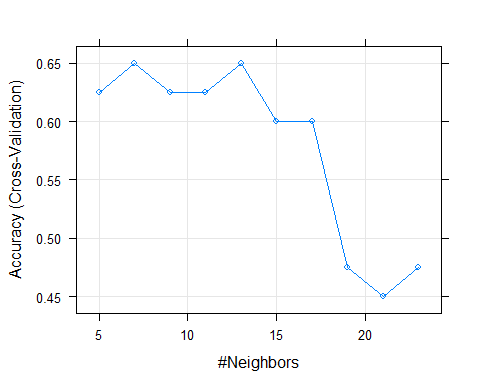
sum <- sum(predRF==testingSet$ROI\_Low\_High)   
length <- length(testingSet$ROI\_Low\_High)  
accuracy\_rfMod <- (sum/length)   
accuracy\_rfMod

## [1] 0.9285714

results <- c(round(accuracy\_rfMod,2), round(100,2))  
results <- as.factor(results)  
results <- t(data.frame(results))  
  
colnames(results) <- colnames(predDF)  
Results <- rbind(predDF, results)   
Results

## predRF type  
## 1 High High  
## 2 High High  
## 3 Low Low  
## 4 Low Low  
## 5 High High  
## 6 High High  
## 7 High High  
## 8 Low Low  
## 9 Low Low  
## 10 High High  
## 11 Low Low  
## 12 Low Low  
## 13 High High  
## 14 High Low  
## results 0.93 100

knnMod <- train(ROI\_Low\_High ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet)  
plot(knnMod)



rpartMod <- train(ROI\_Low\_High ~ ., method='rpart', tuneLength=7, data=trainingSet)

glmMod <- train(ROI\_Low\_High ~ .,   
 method='glm', data=trainingSet)

predKNN <- predict(knnMod, testingSet)  
predRPART <- predict(rpartMod, testingSet)  
predGLM <- predict(glmMod, testingSet)

length=length(testingSet$ROI\_Low\_High)  
  
sumKNN <- sum(predKNN==testingSet$ROI\_Low\_High)  
sumRPart <- sum(predRPART==testingSet$ROI\_Low\_High)  
sumGLM <- sum(predGLM==testingSet$ROI\_Low\_High)

accuracy\_KNN <- sumKNN/length   
accuracy\_RPART <- sumRPart/length   
accuracy\_GLM <- sumGLM/length

predDF2 <- data.frame(predRF,predKNN,predRPART,predGLM,   
 TYPE=testingSet$ROI\_Low\_High)  
colnames(predDF2) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
  
results <- c(round(accuracy\_rfMod,2),   
 round(accuracy\_KNN,2),   
 round(accuracy\_RPART,2),  
 round(accuracy\_GLM,2),   
 round(100,2))  
  
results <- as.factor(results)  
results <- t(data.frame(results))  
colnames(results) <- c('RandomForest','KNN','Rpart','GLM','TrueValue')  
Results <- rbind(predDF2, results)   
Results

## RandomForest KNN Rpart GLM TrueValue  
## 1 High Low High Low High  
## 2 High Low High High High  
## 3 Low Low Low Low Low  
## 4 Low Low Low Low Low  
## 5 High High High High High  
## 6 High High High High High  
## 7 High High High Low High  
## 8 Low High Low Low Low  
## 9 Low Low Low Low Low  
## 10 High High High High High  
## 11 Low Low Low Low Low  
## 12 Low High Low Low Low  
## 13 High High High High High  
## 14 High Low High Low Low  
## results 0.93 0.71 0.93 0.86 100

Lets add in columns for each stock from the number of times the stock increased or decreased in this time span of 2007-2020, and also how many times the stock had that many days of increasing or decreasing. So that, lets say the stock increased 23 days, but from 2007-2020, the stock actually had time periods it increased 23 days 8 times. This could also be a useful feature to have about the stock. We will go back to our ROI-HandPickedStocks.Rmd file to grab these values we made for the last 36 stocks not in the set of 17 best stocks that had positive median values of daily changes when the DOW was down and unemployment was higher than the month before.

aal7\_gd <- aal7\_decr\_Days\_grouped  
aal7\_gi <- aal7\_incr\_Days\_grouped  
  
  
arwr7\_gd <- arwr7\_decr\_Days\_grouped  
arwr7\_gi <- arwr7\_incr\_Days\_grouped  
  
c7\_gd <- c7\_decr\_Days\_grouped  
c7\_gi <- c7\_incr\_Days\_grouped  
  
  
cvx7\_gd <- cvx7\_decr\_Days\_grouped  
cvx7\_gi <- cvx7\_incr\_Days\_grouped  
  
epd7\_gd <- epd7\_decr\_Days\_grouped  
epd7\_gi <- epd7\_incr\_Days\_grouped  
  
  
f7\_gd <- f7\_decr\_Days\_grouped  
f7\_gi <- f7\_incr\_Days\_grouped  
  
ftr7\_gd <- ftr7\_decr\_Days\_grouped  
ftr7\_gi <- ftr7\_incr\_Days\_grouped  
  
  
hmc7\_gd <- hmc7\_decr\_Days\_grouped  
hmc7\_gi <- hmc7\_incr\_Days\_grouped  
  
hoft7\_gd <- hoft7\_decr\_Days\_grouped  
hoft7\_gi <- hoft7\_incr\_Days\_grouped  
  
  
hrb7\_gd <- hrb7\_decr\_Days\_grouped  
hrb7\_gi <- hrb7\_incr\_Days\_grouped  
  
hst7\_gd <- hst7\_decr\_Days\_grouped  
hst7\_gi <- hst7\_incr\_Days\_grouped  
  
  
ino7\_gd <- ino7\_decr\_Days\_grouped  
ino7\_gi <- ino7\_incr\_Days\_grouped  
  
jblu7\_gd <- jblu7\_decr\_Days\_grouped  
jblu7\_gi <- jblu7\_incr\_Days\_grouped  
  
  
kgji7\_gd <- kgji7\_decr\_Days\_grouped  
kgji7\_gi <- kgji7\_incr\_Days\_grouped  
  
kss7\_gd <- kss7\_decr\_Days\_grouped  
kss7\_gi <- kss7\_incr\_Days\_grouped  
  
  
luv7\_gd <- luv7\_decr\_Days\_grouped  
luv7\_gi <- luv7\_incr\_Days\_grouped  
  
m7\_gd <- m7\_decr\_Days\_grouped  
m7\_gi <- m7\_incr\_Days\_grouped  
  
  
mgm7\_gd <- mgm7\_decr\_Days\_grouped  
mgm7\_gi <- mgm7\_incr\_Days\_grouped  
  
msft7\_gd <- msft7\_decr\_Days\_grouped  
msft7\_gi <- msft7\_incr\_Days\_grouped  
  
  
nsany7\_gd <- nsany7\_decr\_Days\_grouped  
nsany7\_gi <- nsany7\_incr\_Days\_grouped  
  
nus7\_gd <- nus7\_decr\_Days\_grouped  
nus7\_gi <- nus7\_incr\_Days\_grouped  
  
  
oncy7\_gd <- oncy7\_decr\_Days\_grouped  
oncy7\_gi <- oncy7\_incr\_Days\_grouped  
  
rrgb7\_gd <- rrgb7\_decr\_Days\_grouped  
rrgb7\_gi <- rrgb7\_incr\_Days\_grouped  
  
  
s7\_gd <- s7\_decr\_Days\_grouped  
s7\_gi <- s7\_incr\_Days\_grouped  
  
sig7\_gd <- sig7\_decr\_Days\_grouped  
sig7\_gi <- sig7\_incr\_Days\_grouped  
  
  
t7\_gd <- t7\_decr\_Days\_grouped  
t7\_gi <- t7\_incr\_Days\_grouped  
  
tg7\_gd <- tgt7\_decr\_Days\_grouped  
tgt7\_gi <- tgt7\_incr\_Days\_grouped  
  
  
tm7\_gd <- tm7\_decr\_Days\_grouped  
tm7\_gi <- tm7\_incr\_Days\_grouped  
  
ubsi7\_gd <- ubsi7\_decr\_Days\_grouped  
ubsi7\_gi <- ubsi7\_incr\_Days\_grouped  
  
  
vz7\_gd <- vz7\_decr\_Days\_grouped  
vz7\_gi <- vz7\_incr\_Days\_grouped  
  
wfc7\_gd <- wfc7\_decr\_Days\_grouped  
wfc7\_gi <- wfc7\_incr\_Days\_grouped  
  
  
wm7\_gd <- wm7\_decr\_Days\_grouped  
wm7\_gi <- wm7\_incr\_Days\_grouped  
  
wwe7\_gd <- wwe7\_decr\_Days\_grouped  
wwe7\_gi <- wwe7\_incr\_Days\_grouped

The groups by each count of cumulative sums of days that increased or decreased and counts the number of times there were that number of days that increased or decreased before changing to the opposite direction in increasing or decreasing. These values were obtained by taking the date of each instance’s stock value to the stock value 7 days earlier using the lag of the stock value with a value of 7 from the ROI-HandPickedStocks.Rmd file.

ffin7\_decr\_Days\_grouped <- countFFIN1 %>% group\_by(decr\_Days) %>% count(n=n())  
ffin7\_decr\_Days\_grouped <- ffin7\_decr\_Days\_grouped[-3]  
colnames(ffin7\_decr\_Days\_grouped)[1] <- c('ffin\_decr\_Days')  
  
ffin7\_incr\_Days\_grouped <- countFFIN2 %>% group\_by(incr\_Days) %>% count(n=n())  
ffin7\_incr\_Days\_grouped <- ffin7\_incr\_Days\_grouped[-3]  
colnames(ffin7\_incr\_Days\_grouped)[1] <- c('ffin\_incr\_Days')  
  
ffin7\_gd <- ffin7\_decr\_Days\_grouped  
ffin7\_gi <- ffin7\_incr\_Days\_grouped

aap7\_decr\_Days\_grouped <- countAAP1 %>% group\_by(decr\_Days) %>% count(n=n())  
aap7\_decr\_Days\_grouped <- aap7\_decr\_Days\_grouped[-3]  
colnames(aap7\_decr\_Days\_grouped)[1] <- c('aap\_decr\_Days')  
  
aap7\_incr\_Days\_grouped <- countAAP2 %>% group\_by(incr\_Days) %>% count(n=n())  
aap7\_incr\_Days\_grouped <- aap7\_incr\_Days\_grouped[-3]  
colnames(aap7\_incr\_Days\_grouped)[1] <- c('aap\_incr\_Days')  
  
aap7\_gd <- aap7\_decr\_Days\_grouped  
aap7\_gi <- aap7\_incr\_Days\_grouped

addy7\_decr\_Days\_grouped <- countADDYY1 %>% group\_by(decr\_Days) %>% count(n=n())  
addy7\_decr\_Days\_grouped <- addy7\_decr\_Days\_grouped[-3]  
colnames(addy7\_decr\_Days\_grouped)[1] <- c('addy\_decr\_Days')  
  
addy7\_incr\_Days\_grouped <- countADDYY2 %>% group\_by(incr\_Days) %>% count(n=n())  
addy7\_incr\_Days\_grouped <- addy7\_incr\_Days\_grouped[-3]  
colnames(addy7\_incr\_Days\_grouped)[1] <- c('addy\_incr\_Days')  
  
addy7\_gd <- addy7\_decr\_Days\_grouped  
addy7\_gi <- addy7\_incr\_Days\_grouped

amzn7\_decr\_Days\_grouped <- countAMZN1 %>% group\_by(decr\_Days) %>% count(n=n())  
amzn7\_decr\_Days\_grouped <- amzn7\_decr\_Days\_grouped[-3]  
colnames(amzn7\_decr\_Days\_grouped)[1] <- c('amzn\_decr\_Days')  
  
amzn7\_incr\_Days\_grouped <- countAMZN2 %>% group\_by(incr\_Days) %>% count(n=n())  
amzn7\_incr\_Days\_grouped <- amzn7\_incr\_Days\_grouped[-3]  
colnames(amzn7\_incr\_Days\_grouped)[1] <- c('amzn\_incr\_Days')  
  
amzn7\_gd <- amzn7\_decr\_Days\_grouped  
amzn7\_gi <- amzn7\_incr\_Days\_grouped

cost7\_decr\_Days\_grouped <- countCOST1 %>% group\_by(decr\_Days) %>% count(n=n())  
cost7\_decr\_Days\_grouped <- cost7\_decr\_Days\_grouped[-3]  
colnames(cost7\_decr\_Days\_grouped)[1] <- c('cost\_decr\_Days')  
  
cost7\_incr\_Days\_grouped <- countCOST2 %>% group\_by(incr\_Days) %>% count(n=n())  
cost7\_incr\_Days\_grouped <- cost7\_incr\_Days\_grouped[-3]  
colnames(cost7\_incr\_Days\_grouped)[1] <- c('cost\_incr\_Days')  
  
cost7\_gd <- cost7\_decr\_Days\_grouped  
cost7\_gi <- cost7\_incr\_Days\_grouped

dltr7\_decr\_Days\_grouped <- countDLTR1 %>% group\_by(decr\_Days) %>% count(n=n())  
dltr7\_decr\_Days\_grouped <- dltr7\_decr\_Days\_grouped[-3]  
colnames(dltr7\_decr\_Days\_grouped)[1] <- c('dltr\_decr\_Days')  
  
dltr7\_incr\_Days\_grouped <- countDLTR2 %>% group\_by(incr\_Days) %>% count(n=n())  
dltr7\_incr\_Days\_grouped <- dltr7\_incr\_Days\_grouped[-3]  
colnames(dltr7\_incr\_Days\_grouped)[1] <- c('dltr\_incr\_Days')  
  
dltr7\_gd <- dltr7\_decr\_Days\_grouped  
dltr7\_gi <- dltr7\_incr\_Days\_grouped

goog7\_decr\_Days\_grouped <- countGOOG1 %>% group\_by(decr\_Days) %>% count(n=n())  
goog7\_decr\_Days\_grouped <- goog7\_decr\_Days\_grouped[-3]  
colnames(goog7\_decr\_Days\_grouped)[1] <- c('goog\_decr\_Days')  
  
goog7\_incr\_Days\_grouped <- countGOOG2 %>% group\_by(incr\_Days) %>% count(n=n())  
goog7\_incr\_Days\_grouped <- goog7\_incr\_Days\_grouped[-3]  
colnames(goog7\_incr\_Days\_grouped)[1] <- c('goog\_incr\_Days')  
  
goog7\_gd <- goog7\_decr\_Days\_grouped  
goog7\_gi <- goog7\_incr\_Days\_grouped

hd7\_decr\_Days\_grouped <- countHD1 %>% group\_by(decr\_Days) %>% count(n=n())  
hd7\_decr\_Days\_grouped <- hd7\_decr\_Days\_grouped[-3]  
colnames(hd7\_decr\_Days\_grouped)[1] <- c('hd\_decr\_Days')  
  
hd7\_incr\_Days\_grouped <- countHD2 %>% group\_by(incr\_Days) %>% count(n=n())  
hd7\_incr\_Days\_grouped <- hd7\_incr\_Days\_grouped[-3]  
colnames(hd7\_incr\_Days\_grouped)[1] <- c('hd\_incr\_Days')  
  
hd7\_gd <- hd7\_decr\_Days\_grouped  
hd7\_gi <- hd7\_incr\_Days\_grouped

jnj7\_decr\_Days\_grouped <- countJNJ1 %>% group\_by(decr\_Days) %>% count(n=n())  
jnj7\_decr\_Days\_grouped <- jnj7\_decr\_Days\_grouped[-3]  
colnames(jnj7\_decr\_Days\_grouped)[1] <- c('jnj\_decr\_Days')  
  
jnj7\_incr\_Days\_grouped <- countJNJ2 %>% group\_by(incr\_Days) %>% count(n=n())  
jnj7\_incr\_Days\_grouped <- jnj7\_incr\_Days\_grouped[-3]  
colnames(jnj7\_incr\_Days\_grouped)[1] <- c('jnj\_incr\_Days')  
  
jnj7\_gd <- jnj7\_decr\_Days\_grouped  
jnj7\_gi <- jnj7\_incr\_Days\_grouped

nflx7\_decr\_Days\_grouped <- countNFLX1 %>% group\_by(decr\_Days) %>% count(n=n())  
nflx7\_decr\_Days\_grouped <- nflx7\_decr\_Days\_grouped[-3]  
colnames(nflx7\_decr\_Days\_grouped)[1] <- c('nflx\_decr\_Days')  
  
nflx7\_incr\_Days\_grouped <- countNFLX2 %>% group\_by(incr\_Days) %>% count(n=n())  
nflx7\_incr\_Days\_grouped <- nflx7\_incr\_Days\_grouped[-3]  
colnames(nflx7\_incr\_Days\_grouped)[1] <- c('nflx\_incr\_Days')  
  
nflx7\_gd <- nflx7\_decr\_Days\_grouped  
nflx7\_gi <- nflx7\_incr\_Days\_grouped

nke7\_decr\_Days\_grouped <- countNKE1 %>% group\_by(decr\_Days) %>% count(n=n())  
nke7\_decr\_Days\_grouped <- nke7\_decr\_Days\_grouped[-3]  
colnames(nke7\_decr\_Days\_grouped)[1] <- c('nke\_decr\_Days')  
  
nke7\_incr\_Days\_grouped <- countNKE2 %>% group\_by(incr\_Days) %>% count(n=n())  
nke7\_incr\_Days\_grouped <- nke7\_incr\_Days\_grouped[-3]  
colnames(nke7\_incr\_Days\_grouped)[1] <- c('nke\_incr\_Days')  
  
nke7\_gd <- nke7\_decr\_Days\_grouped  
nke7\_gi <- nke7\_incr\_Days\_grouped

pcg7\_decr\_Days\_grouped <- countPCG1 %>% group\_by(decr\_Days) %>% count(n=n())  
pcg7\_decr\_Days\_grouped <- pcg7\_decr\_Days\_grouped[-3]  
colnames(pcg7\_decr\_Days\_grouped)[1] <- c('pcg\_decr\_Days')  
  
pcg7\_incr\_Days\_grouped <- countPCG2 %>% group\_by(incr\_Days) %>% count(n=n())  
pcg7\_incr\_Days\_grouped <- pcg7\_incr\_Days\_grouped[-3]  
colnames(pcg7\_incr\_Days\_grouped)[1] <- c('pcg\_incr\_Days')  
  
pcg7\_gd <- pcg7\_decr\_Days\_grouped  
pcg7\_gi <- pcg7\_incr\_Days\_grouped

rost7\_decr\_Days\_grouped <- countROST1 %>% group\_by(decr\_Days) %>% count(n=n())  
rost7\_decr\_Days\_grouped <- rost7\_decr\_Days\_grouped[-3]  
colnames(rost7\_decr\_Days\_grouped)[1] <- c('rost\_decr\_Days')  
  
rost7\_incr\_Days\_grouped <- countROST2 %>% group\_by(incr\_Days) %>% count(n=n())  
rost7\_incr\_Days\_grouped <- rost7\_incr\_Days\_grouped[-3]  
colnames(rost7\_incr\_Days\_grouped)[1] <- c('rost\_incr\_Days')  
  
rost7\_gd <- rost7\_decr\_Days\_grouped  
rost7\_gi <- rost7\_incr\_Days\_grouped

teva7\_decr\_Days\_grouped <- countTEVA1 %>% group\_by(decr\_Days) %>% count(n=n())  
teva7\_decr\_Days\_grouped <- teva7\_decr\_Days\_grouped[-3]  
colnames(teva7\_decr\_Days\_grouped)[1] <- c('teva\_decr\_Days')  
  
teva7\_incr\_Days\_grouped <- countTEVA2 %>% group\_by(incr\_Days) %>% count(n=n())  
teva7\_incr\_Days\_grouped <- teva7\_incr\_Days\_grouped[-3]  
colnames(teva7\_incr\_Days\_grouped)[1] <- c('teva\_incr\_Days')  
  
teva7\_gd <- teva7\_decr\_Days\_grouped  
teva7\_gi <- teva7\_incr\_Days\_grouped

tjx7\_decr\_Days\_grouped <- countTJX1 %>% group\_by(decr\_Days) %>% count(n=n())  
tjx7\_decr\_Days\_grouped <- tjx7\_decr\_Days\_grouped[-3]  
colnames(tjx7\_decr\_Days\_grouped)[1] <- c('tjx\_decr\_Days')  
  
tjx7\_incr\_Days\_grouped <- countTJX2 %>% group\_by(incr\_Days) %>% count(n=n())  
tjx7\_incr\_Days\_grouped <- tjx7\_incr\_Days\_grouped[-3]  
colnames(tjx7\_incr\_Days\_grouped)[1] <- c('tjx\_incr\_Days')  
  
tjx7\_gd <- tjx7\_decr\_Days\_grouped  
tjx7\_gi <- tjx7\_incr\_Days\_grouped

wmt7\_decr\_Days\_grouped <- countWMT1 %>% group\_by(decr\_Days) %>% count(n=n())  
wmt7\_decr\_Days\_grouped <- wmt7\_decr\_Days\_grouped[-3]  
colnames(wmt7\_decr\_Days\_grouped)[1] <- c('wmt\_decr\_Days')  
  
wmt7\_incr\_Days\_grouped <- countWMT2 %>% group\_by(incr\_Days) %>% count(n=n())  
wmt7\_incr\_Days\_grouped <- wmt7\_incr\_Days\_grouped[-3]  
colnames(wmt7\_incr\_Days\_grouped)[1] <- c('wmt\_incr\_Days')  
  
wmt7\_gd <- wmt7\_decr\_Days\_grouped  
wmt7\_gi <- wmt7\_incr\_Days\_grouped

xom7\_decr\_Days\_grouped <- countXOM1 %>% group\_by(decr\_Days) %>% count(n=n())  
xom7\_decr\_Days\_grouped <- xom7\_decr\_Days\_grouped[-3]  
colnames(xom7\_decr\_Days\_grouped)[1] <- c('xom\_decr\_Days')  
  
xom7\_incr\_Days\_grouped <- countXOM2 %>% group\_by(incr\_Days) %>% count(n=n())  
xom7\_incr\_Days\_grouped <- xom7\_incr\_Days\_grouped[-3]  
colnames(xom7\_incr\_Days\_grouped)[1] <- c('xom\_incr\_Days')  
  
xom7\_gd <- xom7\_decr\_Days\_grouped  
xom7\_gi <- xom7\_incr\_Days\_grouped

Now that we have the grouped counts of how many times a stock in this set increased or decreased consecutively from 2007-2020. We can add these values to our dataset of ALL\_52.

ALL\_52$MedianDecrDayGroup <- 'Median Decreasing Group'  
ALL\_52$MedianDecrDayCounts <- 'median N times Group Decr'  
  
ALL\_52$Q3DecrDayGroup <- '3rd Quantile Decreasing Group'  
ALL\_52$Q3DecrDayCounts <- '3rd Quantile N times Group Decr'  
  
ALL\_52$MaxDecrDayGroup <- 'Max Decreasing Group'  
ALL\_52$MaxDecrDayCounts <- 'Max N times Group Decr'  
  
ALL\_52$MedianIncrDayGroup <- 'group incr median'  
ALL\_52$MedianIncrDayCounts <- 'median N times Group Incr'  
  
ALL\_52$Q3IncrDayGroup <- 'group incr 3rd quantile'  
ALL\_52$Q3IncrDayCounts <- '3rd Quantile N times Group Incr'  
  
ALL\_52$MaxIncrDayGroup <- 'group incr max'  
ALL\_52$MaxIncrDayCounts <- 'Max N times Group Incr'

AAL

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
aal7\_gd <- aal7\_decr\_Days\_grouped  
aal7\_gi <- aal7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(aal7\_gd) <- NULL  
row.names(aal7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
aal7\_gd <- aal7\_gd[order(aal7\_gd$n),]  
aal7\_gi <- aal7\_gi[order(aal7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnaal\_d <- as.numeric(floor(quantile(aal7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(aal7\_gd$n)/2)  
mdnaal\_d1 <- as.numeric(aal7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3aal\_d <- as.numeric(floor(quantile(aal7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(aal7\_gd$n)\*.75)  
q3aal\_d1 <- as.numeric(aal7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxaal\_d <- as.numeric(floor(quantile(aal7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxaal\_d1 <- as.numeric(aal7\_gd[length(aal7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnaal\_i <- as.numeric(floor(quantile(aal7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(aal7\_gi$n)/2)  
mdnaal\_i1 <- as.numeric(aal7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3aal\_i <- as.numeric(floor(quantile(aal7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(aal7\_gi$n)\*.75)  
q3aal\_i1 <- as.numeric(aal7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxaal\_i <- as.numeric(floor(quantile(aal7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxaal\_i1 <- as.numeric(aal7\_gi[length(aal7\_gi$n),1])  
  
  
aal\_g <- grep('AAL', ALL\_52$stock)  
  
ALL\_52[aal\_g,18:29] <- c(mdnaal\_d1, mdnaal\_d, q3aal\_d1, q3aal\_d,  
 mxaal\_d1, mxaal\_d, mdnaal\_i1, mdnaal\_i,  
 q3aal\_i1, q3aal\_i, mxaal\_i1, mxaal\_i)

AAP

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
aap7\_gd <- aap7\_decr\_Days\_grouped  
aap7\_gi <- aap7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(aap7\_gd) <- NULL  
row.names(aap7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
aap7\_gd <- aap7\_gd[order(aap7\_gd$n),]  
aap7\_gi <- aap7\_gi[order(aap7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnaap\_d <- as.numeric(floor(quantile(aap7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(aap7\_gd$n)/2)  
mdnaap\_d1 <- as.numeric(aap7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3aap\_d <- as.numeric(floor(quantile(aap7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(aap7\_gd$n)\*.75)  
q3aap\_d1 <- as.numeric(aap7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxaap\_d <- as.numeric(floor(quantile(aap7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxaap\_d1 <- as.numeric(aap7\_gd[length(aap7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnaap\_i <- as.numeric(floor(quantile(aap7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(aap7\_gi$n)/2)  
mdnaap\_i1 <- as.numeric(aap7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3aap\_i <- as.numeric(floor(quantile(aap7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(aap7\_gi$n)\*.75)  
q3aap\_i1 <- as.numeric(aap7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxaap\_i <- as.numeric(floor(quantile(aap7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxaap\_i1 <- as.numeric(aap7\_gi[length(aap7\_gi$n),1])  
  
  
aap\_g <- grep('AAP', ALL\_52$stock)  
  
ALL\_52[aap\_g,18:29] <- c(mdnaap\_d1, mdnaap\_d, q3aap\_d1, q3aap\_d,  
 mxaap\_d1, mxaap\_d, mdnaap\_i1, mdnaap\_i,  
 q3aap\_i1, q3aap\_i, mxaap\_i1, mxaap\_i)

ADDY

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
addy7\_gd <- addy7\_decr\_Days\_grouped  
addy7\_gi <- addy7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(addy7\_gd) <- NULL  
row.names(addy7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
addy7\_gd <- addy7\_gd[order(addy7\_gd$n),]  
addy7\_gi <- addy7\_gi[order(addy7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnaddy\_d <- as.numeric(floor(quantile(addy7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(addy7\_gd$n)/2)  
mdnaddy\_d1 <- as.numeric(addy7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3addy\_d <- as.numeric(floor(quantile(addy7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(addy7\_gd$n)\*.75)  
q3addy\_d1 <- as.numeric(addy7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxaddy\_d <- as.numeric(floor(quantile(addy7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxaddy\_d1 <- as.numeric(addy7\_gd[length(addy7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnaddy\_i <- as.numeric(floor(quantile(addy7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(addy7\_gi$n)/2)  
mdnaddy\_i1 <- as.numeric(addy7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3addy\_i <- as.numeric(floor(quantile(addy7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(addy7\_gi$n)\*.75)  
q3addy\_i1 <- as.numeric(addy7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxaddy\_i <- as.numeric(floor(quantile(addy7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxaddy\_i1 <- as.numeric(addy7\_gi[length(addy7\_gi$n),1])  
  
  
addy\_g <- grep('ADDY', ALL\_52$stock)  
  
ALL\_52[addy\_g,18:29] <- c(mdnaddy\_d1, mdnaddy\_d, q3addy\_d1, q3addy\_d,  
 mxaddy\_d1, mxaddy\_d, mdnaddy\_i1, mdnaddy\_i,  
 q3addy\_i1, q3addy\_i, mxaddy\_i1, mxaddy\_i)

AMZN

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
amzn7\_gd <- amzn7\_decr\_Days\_grouped  
amzn7\_gi <- amzn7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(amzn7\_gd) <- NULL  
row.names(amzn7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
amzn7\_gd <- amzn7\_gd[order(amzn7\_gd$n),]  
amzn7\_gi <- amzn7\_gi[order(amzn7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnamzn\_d <- as.numeric(floor(quantile(amzn7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(amzn7\_gd$n)/2)  
mdnamzn\_d1 <- as.numeric(amzn7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3amzn\_d <- as.numeric(floor(quantile(amzn7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(amzn7\_gd$n)\*.75)  
q3amzn\_d1 <- as.numeric(amzn7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxamzn\_d <- as.numeric(floor(quantile(amzn7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxamzn\_d1 <- as.numeric(amzn7\_gd[length(amzn7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnamzn\_i <- as.numeric(floor(quantile(amzn7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(amzn7\_gi$n)/2)  
mdnamzn\_i1 <- as.numeric(amzn7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3amzn\_i <- as.numeric(floor(quantile(amzn7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(amzn7\_gi$n)\*.75)  
q3amzn\_i1 <- as.numeric(amzn7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxamzn\_i <- as.numeric(floor(quantile(amzn7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxamzn\_i1 <- as.numeric(amzn7\_gi[length(amzn7\_gi$n),1])  
  
  
amzn\_g <- grep('AMZN', ALL\_52$stock)  
  
ALL\_52[amzn\_g,18:29] <- c(mdnamzn\_d1, mdnamzn\_d, q3amzn\_d1, q3amzn\_d,  
 mxamzn\_d1, mxamzn\_d, mdnamzn\_i1, mdnamzn\_i,  
 q3amzn\_i1, q3amzn\_i, mxamzn\_i1, mxamzn\_i)

ARWR

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
arwr7\_gd <- arwr7\_decr\_Days\_grouped  
arwr7\_gi <- arwr7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(arwr7\_gd) <- NULL  
row.names(arwr7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
arwr7\_gd <- arwr7\_gd[order(arwr7\_gd$n),]  
arwr7\_gi <- arwr7\_gi[order(arwr7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnarwr\_d <- as.numeric(floor(quantile(arwr7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(arwr7\_gd$n)/2)  
mdnarwr\_d1 <- as.numeric(arwr7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3arwr\_d <- as.numeric(floor(quantile(arwr7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(arwr7\_gd$n)\*.75)  
q3arwr\_d1 <- as.numeric(arwr7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxarwr\_d <- as.numeric(floor(quantile(arwr7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxarwr\_d1 <- as.numeric(arwr7\_gd[length(arwr7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnarwr\_i <- as.numeric(floor(quantile(arwr7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(arwr7\_gi$n)/2)  
mdnarwr\_i1 <- as.numeric(arwr7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3arwr\_i <- as.numeric(floor(quantile(arwr7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(arwr7\_gi$n)\*.75)  
q3arwr\_i1 <- as.numeric(arwr7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxarwr\_i <- as.numeric(floor(quantile(arwr7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxarwr\_i1 <- as.numeric(arwr7\_gi[length(arwr7\_gi$n),1])  
  
  
arwr\_g <- grep('ARWR', ALL\_52$stock)  
  
ALL\_52[arwr\_g,18:29] <- c(mdnarwr\_d1, mdnarwr\_d, q3arwr\_d1, q3arwr\_d,  
 mxarwr\_d1, mxarwr\_d, mdnarwr\_i1, mdnarwr\_i,  
 q3arwr\_i1, q3arwr\_i, mxarwr\_i1, mxarwr\_i)

C

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
c7\_gd <- c7\_decr\_Days\_grouped  
c7\_gi <- c7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(c7\_gd) <- NULL  
row.names(c7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
c7\_gd <- c7\_gd[order(c7\_gd$n),]  
c7\_gi <- c7\_gi[order(c7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnc\_d <- as.numeric(floor(quantile(c7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(c7\_gd$n)/2)  
mdnc\_d1 <- as.numeric(c7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3c\_d <- as.numeric(floor(quantile(c7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(c7\_gd$n)\*.75)  
q3c\_d1 <- as.numeric(c7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxc\_d <- as.numeric(floor(quantile(c7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxc\_d1 <- as.numeric(c7\_gd[length(c7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnc\_i <- as.numeric(floor(quantile(c7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(c7\_gi$n)/2)  
mdnc\_i1 <- as.numeric(c7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3c\_i <- as.numeric(floor(quantile(c7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(c7\_gi$n)\*.75)  
q3c\_i1 <- as.numeric(c7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxc\_i <- as.numeric(floor(quantile(c7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxc\_i1 <- as.numeric(c7\_gi[length(c7\_gi$n),1])  
  
  
c\_g <- grep('^C$', ALL\_52$stock)  
  
ALL\_52[c\_g,18:29] <- c(mdnc\_d1, mdnc\_d, q3c\_d1, q3c\_d,  
 mxc\_d1, mxc\_d, mdnc\_i1, mdnc\_i,  
 q3c\_i1, q3c\_i, mxc\_i1, mxc\_i)

COST

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
cost7\_gd <- cost7\_decr\_Days\_grouped  
cost7\_gi <- cost7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(cost7\_gd) <- NULL  
row.names(cost7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
cost7\_gd <- cost7\_gd[order(cost7\_gd$n),]  
cost7\_gi <- cost7\_gi[order(cost7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdncost\_d <- as.numeric(floor(quantile(cost7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(cost7\_gd$n)/2)  
mdncost\_d1 <- as.numeric(cost7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3cost\_d <- as.numeric(floor(quantile(cost7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(cost7\_gd$n)\*.75)  
q3cost\_d1 <- as.numeric(cost7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxcost\_d <- as.numeric(floor(quantile(cost7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxcost\_d1 <- as.numeric(cost7\_gd[length(cost7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdncost\_i <- as.numeric(floor(quantile(cost7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(cost7\_gi$n)/2)  
mdncost\_i1 <- as.numeric(cost7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3cost\_i <- as.numeric(floor(quantile(cost7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(cost7\_gi$n)\*.75)  
q3cost\_i1 <- as.numeric(cost7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxcost\_i <- as.numeric(floor(quantile(cost7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxcost\_i1 <- as.numeric(cost7\_gi[length(cost7\_gi$n),1])  
  
  
cost\_g <- grep('COST', ALL\_52$stock)  
  
ALL\_52[cost\_g,18:29] <- c(mdncost\_d1, mdncost\_d, q3cost\_d1, q3cost\_d,  
 mxcost\_d1, mxcost\_d, mdncost\_i1, mdncost\_i,  
 q3cost\_i1, q3cost\_i, mxcost\_i1, mxcost\_i)

CVX

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
cvx7\_gd <- cvx7\_decr\_Days\_grouped  
cvx7\_gi <- cvx7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(cvx7\_gd) <- NULL  
row.names(cvx7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
cvx7\_gd <- cvx7\_gd[order(cvx7\_gd$n),]  
cvx7\_gi <- cvx7\_gi[order(cvx7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdncvx\_d <- as.numeric(floor(quantile(cvx7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(cvx7\_gd$n)/2)  
mdncvx\_d1 <- as.numeric(cvx7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3cvx\_d <- as.numeric(floor(quantile(cvx7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(cvx7\_gd$n)\*.75)  
q3cvx\_d1 <- as.numeric(cvx7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxcvx\_d <- as.numeric(floor(quantile(cvx7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxcvx\_d1 <- as.numeric(cvx7\_gd[length(cvx7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdncvx\_i <- as.numeric(floor(quantile(cvx7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(cvx7\_gi$n)/2)  
mdncvx\_i1 <- as.numeric(cvx7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3cvx\_i <- as.numeric(floor(quantile(cvx7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(cvx7\_gi$n)\*.75)  
q3cvx\_i1 <- as.numeric(cvx7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxcvx\_i <- as.numeric(floor(quantile(cvx7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxcvx\_i1 <- as.numeric(cvx7\_gi[length(cvx7\_gi$n),1])  
  
  
cvx\_g <- grep('CVX', ALL\_52$stock)  
  
ALL\_52[cvx\_g,18:29] <- c(mdncvx\_d1, mdncvx\_d, q3cvx\_d1, q3cvx\_d,  
 mxcvx\_d1, mxcvx\_d, mdncvx\_i1, mdncvx\_i,  
 q3cvx\_i1, q3cvx\_i, mxcvx\_i1, mxcvx\_i)

DLTR

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
dltr7\_gd <- dltr7\_decr\_Days\_grouped  
dltr7\_gi <- dltr7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(dltr7\_gd) <- NULL  
row.names(dltr7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
dltr7\_gd <- dltr7\_gd[order(dltr7\_gd$n),]  
dltr7\_gi <- dltr7\_gi[order(dltr7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdndltr\_d <- as.numeric(floor(quantile(dltr7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(dltr7\_gd$n)/2)  
mdndltr\_d1 <- as.numeric(dltr7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3dltr\_d <- as.numeric(floor(quantile(dltr7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(dltr7\_gd$n)\*.75)  
q3dltr\_d1 <- as.numeric(dltr7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxdltr\_d <- as.numeric(floor(quantile(dltr7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxdltr\_d1 <- as.numeric(dltr7\_gd[length(dltr7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdndltr\_i <- as.numeric(floor(quantile(dltr7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(dltr7\_gi$n)/2)  
mdndltr\_i1 <- as.numeric(dltr7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3dltr\_i <- as.numeric(floor(quantile(dltr7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(dltr7\_gi$n)\*.75)  
q3dltr\_i1 <- as.numeric(dltr7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxdltr\_i <- as.numeric(floor(quantile(dltr7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxdltr\_i1 <- as.numeric(dltr7\_gi[length(dltr7\_gi$n),1])  
  
  
dltr\_g <- grep('DLTR', ALL\_52$stock)  
  
ALL\_52[dltr\_g,18:29] <- c(mdndltr\_d1, mdndltr\_d, q3dltr\_d1, q3dltr\_d,  
 mxdltr\_d1, mxdltr\_d, mdndltr\_i1, mdndltr\_i,  
 q3dltr\_i1, q3dltr\_i, mxdltr\_i1, mxdltr\_i)

EPD

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
epd7\_gd <- epd7\_decr\_Days\_grouped  
epd7\_gi <- epd7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(epd7\_gd) <- NULL  
row.names(epd7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
epd7\_gd <- epd7\_gd[order(epd7\_gd$n),]  
epd7\_gi <- epd7\_gi[order(epd7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnepd\_d <- as.numeric(floor(quantile(epd7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(epd7\_gd$n)/2)  
mdnepd\_d1 <- as.numeric(epd7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3epd\_d <- as.numeric(floor(quantile(epd7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(epd7\_gd$n)\*.75)  
q3epd\_d1 <- as.numeric(epd7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxepd\_d <- as.numeric(floor(quantile(epd7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxepd\_d1 <- as.numeric(epd7\_gd[length(epd7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnepd\_i <- as.numeric(floor(quantile(epd7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(epd7\_gi$n)/2)  
mdnepd\_i1 <- as.numeric(epd7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3epd\_i <- as.numeric(floor(quantile(epd7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(epd7\_gi$n)\*.75)  
q3epd\_i1 <- as.numeric(epd7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxepd\_i <- as.numeric(floor(quantile(epd7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxepd\_i1 <- as.numeric(epd7\_gi[length(epd7\_gi$n),1])  
  
  
epd\_g <- grep('EPD', ALL\_52$stock)  
  
ALL\_52[epd\_g,18:29] <- c(mdnepd\_d1, mdnepd\_d, q3epd\_d1, q3epd\_d,  
 mxepd\_d1, mxepd\_d, mdnepd\_i1, mdnepd\_i,  
 q3epd\_i1, q3epd\_i, mxepd\_i1, mxepd\_i)

F

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
f7\_gd <- f7\_decr\_Days\_grouped  
f7\_gi <- f7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(f7\_gd) <- NULL  
row.names(f7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
f7\_gd <- f7\_gd[order(f7\_gd$n),]  
f7\_gi <- f7\_gi[order(f7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnf\_d <- as.numeric(floor(quantile(f7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(f7\_gd$n)/2)  
mdnf\_d1 <- as.numeric(f7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3f\_d <- as.numeric(floor(quantile(f7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(f7\_gd$n)\*.75)  
q3f\_d1 <- as.numeric(f7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxf\_d <- as.numeric(floor(quantile(f7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxf\_d1 <- as.numeric(f7\_gd[length(f7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnf\_i <- as.numeric(floor(quantile(f7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(f7\_gi$n)/2)  
mdnf\_i1 <- as.numeric(f7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3f\_i <- as.numeric(floor(quantile(f7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(f7\_gi$n)\*.75)  
q3f\_i1 <- as.numeric(f7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxf\_i <- as.numeric(floor(quantile(f7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxf\_i1 <- as.numeric(f7\_gi[length(f7\_gi$n),1])  
  
  
f\_g <- grep('^F$', ALL\_52$stock)  
  
ALL\_52[f\_g,18:29] <- c(mdnf\_d1, mdnf\_d, q3f\_d1, q3f\_d,  
 mxf\_d1, mxf\_d, mdnf\_i1, mdnf\_i,  
 q3f\_i1, q3f\_i, mxf\_i1, mxf\_i)

FFIN

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
ffin7\_gd <- ffin7\_decr\_Days\_grouped  
ffin7\_gi <- ffin7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(ffin7\_gd) <- NULL  
row.names(ffin7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
ffin7\_gd <- ffin7\_gd[order(ffin7\_gd$n),]  
ffin7\_gi <- ffin7\_gi[order(ffin7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnffin\_d <- as.numeric(floor(quantile(ffin7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(ffin7\_gd$n)/2)  
mdnffin\_d1 <- as.numeric(ffin7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ffin\_d <- as.numeric(floor(quantile(ffin7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ffin7\_gd$n)\*.75)  
q3ffin\_d1 <- as.numeric(ffin7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxffin\_d <- as.numeric(floor(quantile(ffin7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxffin\_d1 <- as.numeric(ffin7\_gd[length(ffin7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnffin\_i <- as.numeric(floor(quantile(ffin7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(ffin7\_gi$n)/2)  
mdnffin\_i1 <- as.numeric(ffin7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ffin\_i <- as.numeric(floor(quantile(ffin7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ffin7\_gi$n)\*.75)  
q3ffin\_i1 <- as.numeric(ffin7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxffin\_i <- as.numeric(floor(quantile(ffin7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxffin\_i1 <- as.numeric(ffin7\_gi[length(ffin7\_gi$n),1])  
  
  
ffin\_g <- grep('FFIN', ALL\_52$stock)  
  
ALL\_52[ffin\_g,18:29] <- c(mdnffin\_d1, mdnffin\_d, q3ffin\_d1, q3ffin\_d,  
 mxffin\_d1, mxffin\_d, mdnffin\_i1, mdnffin\_i,  
 q3ffin\_i1, q3ffin\_i, mxffin\_i1, mxffin\_i)

FTR

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
ftr7\_gd <- ftr7\_decr\_Days\_grouped  
ftr7\_gi <- ftr7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(ftr7\_gd) <- NULL  
row.names(ftr7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
ftr7\_gd <- ftr7\_gd[order(ftr7\_gd$n),]  
ftr7\_gi <- ftr7\_gi[order(ftr7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnftr\_d <- as.numeric(floor(quantile(ftr7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(ftr7\_gd$n)/2)  
mdnftr\_d1 <- as.numeric(ftr7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ftr\_d <- as.numeric(floor(quantile(ftr7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ftr7\_gd$n)\*.75)  
q3ftr\_d1 <- as.numeric(ftr7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxftr\_d <- as.numeric(floor(quantile(ftr7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxftr\_d1 <- as.numeric(ftr7\_gd[length(ftr7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnftr\_i <- as.numeric(floor(quantile(ftr7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(ftr7\_gi$n)/2)  
mdnftr\_i1 <- as.numeric(ftr7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ftr\_i <- as.numeric(floor(quantile(ftr7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ftr7\_gi$n)\*.75)  
q3ftr\_i1 <- as.numeric(ftr7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxftr\_i <- as.numeric(floor(quantile(ftr7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxftr\_i1 <- as.numeric(ftr7\_gi[length(ftr7\_gi$n),1])  
  
  
ftr\_g <- grep('FTR', ALL\_52$stock)  
  
ALL\_52[ftr\_g,18:29] <- c(mdnftr\_d1, mdnftr\_d, q3ftr\_d1, q3ftr\_d,  
 mxftr\_d1, mxftr\_d, mdnftr\_i1, mdnftr\_i,  
 q3ftr\_i1, q3ftr\_i, mxftr\_i1, mxftr\_i)

GOOG

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
goog7\_gd <- goog7\_decr\_Days\_grouped  
goog7\_gi <- goog7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(goog7\_gd) <- NULL  
row.names(goog7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
goog7\_gd <- goog7\_gd[order(goog7\_gd$n),]  
goog7\_gi <- goog7\_gi[order(goog7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdngoog\_d <- as.numeric(floor(quantile(goog7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(goog7\_gd$n)/2)  
mdngoog\_d1 <- as.numeric(goog7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3goog\_d <- as.numeric(floor(quantile(goog7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(goog7\_gd$n)\*.75)  
q3goog\_d1 <- as.numeric(goog7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxgoog\_d <- as.numeric(floor(quantile(goog7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxgoog\_d1 <- as.numeric(goog7\_gd[length(goog7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdngoog\_i <- as.numeric(floor(quantile(goog7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(goog7\_gi$n)/2)  
mdngoog\_i1 <- as.numeric(goog7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3goog\_i <- as.numeric(floor(quantile(goog7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(goog7\_gi$n)\*.75)  
q3goog\_i1 <- as.numeric(goog7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxgoog\_i <- as.numeric(floor(quantile(goog7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxgoog\_i1 <- as.numeric(goog7\_gi[length(goog7\_gi$n),1])  
  
  
goog\_g <- grep('GOOG', ALL\_52$stock)  
  
ALL\_52[goog\_g,18:29] <- c(mdngoog\_d1, mdngoog\_d, q3goog\_d1, q3goog\_d,  
 mxgoog\_d1, mxgoog\_d, mdngoog\_i1, mdngoog\_i,  
 q3goog\_i1, q3goog\_i, mxgoog\_i1, mxgoog\_i)

HD

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
hd7\_gd <- hd7\_decr\_Days\_grouped  
hd7\_gi <- hd7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(hd7\_gd) <- NULL  
row.names(hd7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
hd7\_gd <- hd7\_gd[order(hd7\_gd$n),]  
hd7\_gi <- hd7\_gi[order(hd7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnhd\_d <- as.numeric(floor(quantile(hd7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(hd7\_gd$n)/2)  
mdnhd\_d1 <- as.numeric(hd7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hd\_d <- as.numeric(floor(quantile(hd7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hd7\_gd$n)\*.75)  
q3hd\_d1 <- as.numeric(hd7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxhd\_d <- as.numeric(floor(quantile(hd7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxhd\_d1 <- as.numeric(hd7\_gd[length(hd7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnhd\_i <- as.numeric(floor(quantile(hd7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(hd7\_gi$n)/2)  
mdnhd\_i1 <- as.numeric(hd7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hd\_i <- as.numeric(floor(quantile(hd7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hd7\_gi$n)\*.75)  
q3hd\_i1 <- as.numeric(hd7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxhd\_i <- as.numeric(floor(quantile(hd7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxhd\_i1 <- as.numeric(hd7\_gi[length(hd7\_gi$n),1])  
  
  
hd\_g <- grep('HD', ALL\_52$stock)  
  
ALL\_52[hd\_g,18:29] <- c(mdnhd\_d1, mdnhd\_d, q3hd\_d1, q3hd\_d,  
 mxhd\_d1, mxhd\_d, mdnhd\_i1, mdnhd\_i,  
 q3hd\_i1, q3hd\_i, mxhd\_i1, mxhd\_i)

HMC

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
hmc7\_gd <- hmc7\_decr\_Days\_grouped  
hmc7\_gi <- hmc7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(hmc7\_gd) <- NULL  
row.names(hmc7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
hmc7\_gd <- hmc7\_gd[order(hmc7\_gd$n),]  
hmc7\_gi <- hmc7\_gi[order(hmc7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnhmc\_d <- as.numeric(floor(quantile(hmc7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(hmc7\_gd$n)/2)  
mdnhmc\_d1 <- as.numeric(hmc7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hmc\_d <- as.numeric(floor(quantile(hmc7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hmc7\_gd$n)\*.75)  
q3hmc\_d1 <- as.numeric(hmc7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxhmc\_d <- as.numeric(floor(quantile(hmc7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxhmc\_d1 <- as.numeric(hmc7\_gd[length(hmc7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnhmc\_i <- as.numeric(floor(quantile(hmc7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(hmc7\_gi$n)/2)  
mdnhmc\_i1 <- as.numeric(hmc7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hmc\_i <- as.numeric(floor(quantile(hmc7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hmc7\_gi$n)\*.75)  
q3hmc\_i1 <- as.numeric(hmc7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxhmc\_i <- as.numeric(floor(quantile(hmc7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxhmc\_i1 <- as.numeric(hmc7\_gi[length(hmc7\_gi$n),1])  
  
  
hmc\_g <- grep('HMC', ALL\_52$stock)  
  
ALL\_52[hmc\_g,18:29] <- c(mdnhmc\_d1, mdnhmc\_d, q3hmc\_d1, q3hmc\_d,  
 mxhmc\_d1, mxhmc\_d, mdnhmc\_i1, mdnhmc\_i,  
 q3hmc\_i1, q3hmc\_i, mxhmc\_i1, mxhmc\_i)

HOFT

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
hoft7\_gd <- hoft7\_decr\_Days\_grouped  
hoft7\_gi <- hoft7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(hoft7\_gd) <- NULL  
row.names(hoft7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
hoft7\_gd <- hoft7\_gd[order(hoft7\_gd$n),]  
hoft7\_gi <- hoft7\_gi[order(hoft7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnhoft\_d <- as.numeric(floor(quantile(hoft7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(hoft7\_gd$n)/2)  
mdnhoft\_d1 <- as.numeric(hoft7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hoft\_d <- as.numeric(floor(quantile(hoft7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hoft7\_gd$n)\*.75)  
q3hoft\_d1 <- as.numeric(hoft7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxhoft\_d <- as.numeric(floor(quantile(hoft7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxhoft\_d1 <- as.numeric(hoft7\_gd[length(hoft7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnhoft\_i <- as.numeric(floor(quantile(hoft7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(hoft7\_gi$n)/2)  
mdnhoft\_i1 <- as.numeric(hoft7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hoft\_i <- as.numeric(floor(quantile(hoft7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hoft7\_gi$n)\*.75)  
q3hoft\_i1 <- as.numeric(hoft7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxhoft\_i <- as.numeric(floor(quantile(hoft7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxhoft\_i1 <- as.numeric(hoft7\_gi[length(hoft7\_gi$n),1])  
  
  
hoft\_g <- grep('HOFT', ALL\_52$stock)  
  
ALL\_52[hoft\_g,18:29] <- c(mdnhoft\_d1, mdnhoft\_d, q3hoft\_d1, q3hoft\_d,  
 mxhoft\_d1, mxhoft\_d, mdnhoft\_i1, mdnhoft\_i,  
 q3hoft\_i1, q3hoft\_i, mxhoft\_i1, mxhoft\_i)

HRB

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
hrb7\_gd <- hrb7\_decr\_Days\_grouped  
hrb7\_gi <- hrb7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(hrb7\_gd) <- NULL  
row.names(hrb7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
hrb7\_gd <- hrb7\_gd[order(hrb7\_gd$n),]  
hrb7\_gi <- hrb7\_gi[order(hrb7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnhrb\_d <- as.numeric(floor(quantile(hrb7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(hrb7\_gd$n)/2)  
mdnhrb\_d1 <- as.numeric(hrb7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hrb\_d <- as.numeric(floor(quantile(hrb7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hrb7\_gd$n)\*.75)  
q3hrb\_d1 <- as.numeric(hrb7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxhrb\_d <- as.numeric(floor(quantile(hrb7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxhrb\_d1 <- as.numeric(hrb7\_gd[length(hrb7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnhrb\_i <- as.numeric(floor(quantile(hrb7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(hrb7\_gi$n)/2)  
mdnhrb\_i1 <- as.numeric(hrb7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hrb\_i <- as.numeric(floor(quantile(hrb7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hrb7\_gi$n)\*.75)  
q3hrb\_i1 <- as.numeric(hrb7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxhrb\_i <- as.numeric(floor(quantile(hrb7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxhrb\_i1 <- as.numeric(hrb7\_gi[length(hrb7\_gi$n),1])  
  
  
hrb\_g <- grep('HRB', ALL\_52$stock)  
  
ALL\_52[hrb\_g,18:29] <- c(mdnhrb\_d1, mdnhrb\_d, q3hrb\_d1, q3hrb\_d,  
 mxhrb\_d1, mxhrb\_d, mdnhrb\_i1, mdnhrb\_i,  
 q3hrb\_i1, q3hrb\_i, mxhrb\_i1, mxhrb\_i)

HST

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
hst7\_gd <- hst7\_decr\_Days\_grouped  
hst7\_gi <- hst7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(hst7\_gd) <- NULL  
row.names(hst7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
hst7\_gd <- hst7\_gd[order(hst7\_gd$n),]  
hst7\_gi <- hst7\_gi[order(hst7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnhst\_d <- as.numeric(floor(quantile(hst7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(hst7\_gd$n)/2)  
mdnhst\_d1 <- as.numeric(hst7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hst\_d <- as.numeric(floor(quantile(hst7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hst7\_gd$n)\*.75)  
q3hst\_d1 <- as.numeric(hst7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxhst\_d <- as.numeric(floor(quantile(hst7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxhst\_d1 <- as.numeric(hst7\_gd[length(hst7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnhst\_i <- as.numeric(floor(quantile(hst7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(hst7\_gi$n)/2)  
mdnhst\_i1 <- as.numeric(hst7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3hst\_i <- as.numeric(floor(quantile(hst7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(hst7\_gi$n)\*.75)  
q3hst\_i1 <- as.numeric(hst7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxhst\_i <- as.numeric(floor(quantile(hst7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxhst\_i1 <- as.numeric(hst7\_gi[length(hst7\_gi$n),1])  
  
  
hst\_g <- grep('HST', ALL\_52$stock)  
  
ALL\_52[hst\_g,18:29] <- c(mdnhst\_d1, mdnhst\_d, q3hst\_d1, q3hst\_d,  
 mxhst\_d1, mxhst\_d, mdnhst\_i1, mdnhst\_i,  
 q3hst\_i1, q3hst\_i, mxhst\_i1, mxhst\_i)

INO

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
ino7\_gd <- ino7\_decr\_Days\_grouped  
ino7\_gi <- ino7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(ino7\_gd) <- NULL  
row.names(ino7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
ino7\_gd <- ino7\_gd[order(ino7\_gd$n),]  
ino7\_gi <- ino7\_gi[order(ino7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnino\_d <- as.numeric(floor(quantile(ino7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(ino7\_gd$n)/2)  
mdnino\_d1 <- as.numeric(ino7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ino\_d <- as.numeric(floor(quantile(ino7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ino7\_gd$n)\*.75)  
q3ino\_d1 <- as.numeric(ino7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxino\_d <- as.numeric(floor(quantile(ino7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxino\_d1 <- as.numeric(ino7\_gd[length(ino7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnino\_i <- as.numeric(floor(quantile(ino7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(ino7\_gi$n)/2)  
mdnino\_i1 <- as.numeric(ino7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ino\_i <- as.numeric(floor(quantile(ino7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ino7\_gi$n)\*.75)  
q3ino\_i1 <- as.numeric(ino7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxino\_i <- as.numeric(floor(quantile(ino7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxino\_i1 <- as.numeric(ino7\_gi[length(ino7\_gi$n),1])  
  
  
ino\_g <- grep('INO', ALL\_52$stock)  
  
ALL\_52[ino\_g,18:29] <- c(mdnino\_d1, mdnino\_d, q3ino\_d1, q3ino\_d,  
 mxino\_d1, mxino\_d, mdnino\_i1, mdnino\_i,  
 q3ino\_i1, q3ino\_i, mxino\_i1, mxino\_i)

JBLU

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
jblu7\_gd <- jblu7\_decr\_Days\_grouped  
jblu7\_gi <- jblu7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(jblu7\_gd) <- NULL  
row.names(jblu7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
jblu7\_gd <- jblu7\_gd[order(jblu7\_gd$n),]  
jblu7\_gi <- jblu7\_gi[order(jblu7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnjblu\_d <- as.numeric(floor(quantile(jblu7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(jblu7\_gd$n)/2)  
mdnjblu\_d1 <- as.numeric(jblu7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jblu\_d <- as.numeric(floor(quantile(jblu7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jblu7\_gd$n)\*.75)  
q3jblu\_d1 <- as.numeric(jblu7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxjblu\_d <- as.numeric(floor(quantile(jblu7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxjblu\_d1 <- as.numeric(jblu7\_gd[length(jblu7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnjblu\_i <- as.numeric(floor(quantile(jblu7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(jblu7\_gi$n)/2)  
mdnjblu\_i1 <- as.numeric(jblu7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jblu\_i <- as.numeric(floor(quantile(jblu7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jblu7\_gi$n)\*.75)  
q3jblu\_i1 <- as.numeric(jblu7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxjblu\_i <- as.numeric(floor(quantile(jblu7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxjblu\_i1 <- as.numeric(jblu7\_gi[length(jblu7\_gi$n),1])  
  
  
jblu\_g <- grep('JBLU', ALL\_52$stock)  
  
ALL\_52[jblu\_g,18:29] <- c(mdnjblu\_d1, mdnjblu\_d, q3jblu\_d1, q3jblu\_d,  
 mxjblu\_d1, mxjblu\_d, mdnjblu\_i1, mdnjblu\_i,  
 q3jblu\_i1, q3jblu\_i, mxjblu\_i1, mxjblu\_i)

JNJ

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
jnj7\_gd <- jnj7\_decr\_Days\_grouped  
jnj7\_gi <- jnj7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(jnj7\_gd) <- NULL  
row.names(jnj7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
jnj7\_gd <- jnj7\_gd[order(jnj7\_gd$n),]  
jnj7\_gi <- jnj7\_gi[order(jnj7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnjnj\_d <- as.numeric(floor(quantile(jnj7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(jnj7\_gd$n)/2)  
mdnjnj\_d1 <- as.numeric(jnj7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jnj\_d <- as.numeric(floor(quantile(jnj7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jnj7\_gd$n)\*.75)  
q3jnj\_d1 <- as.numeric(jnj7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxjnj\_d <- as.numeric(floor(quantile(jnj7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxjnj\_d1 <- as.numeric(jnj7\_gd[length(jnj7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnjnj\_i <- as.numeric(floor(quantile(jnj7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(jnj7\_gi$n)/2)  
mdnjnj\_i1 <- as.numeric(jnj7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jnj\_i <- as.numeric(floor(quantile(jnj7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jnj7\_gi$n)\*.75)  
q3jnj\_i1 <- as.numeric(jnj7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxjnj\_i <- as.numeric(floor(quantile(jnj7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxjnj\_i1 <- as.numeric(jnj7\_gi[length(jnj7\_gi$n),1])  
  
  
jnj\_g <- grep('JNJ', ALL\_52$stock)  
  
ALL\_52[jnj\_g,18:29] <- c(mdnjnj\_d1, mdnjnj\_d, q3jnj\_d1, q3jnj\_d,  
 mxjnj\_d1, mxjnj\_d, mdnjnj\_i1, mdnjnj\_i,  
 q3jnj\_i1, q3jnj\_i, mxjnj\_i1, mxjnj\_i)

JPM

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
jpm7\_gd <- jpm7\_decr\_Days\_grouped  
jpm7\_gi <- jpm7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(jpm7\_gd) <- NULL  
row.names(jpm7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
jpm7\_gd <- jpm7\_gd[order(jpm7\_gd$n),]  
jpm7\_gi <- jpm7\_gi[order(jpm7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnjpm\_d <- as.numeric(floor(quantile(jpm7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(jpm7\_gd$n)/2)  
mdnjpm\_d1 <- as.numeric(jpm7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jpm\_d <- as.numeric(floor(quantile(jpm7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jpm7\_gd$n)\*.75)  
q3jpm\_d1 <- as.numeric(jpm7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxjpm\_d <- as.numeric(floor(quantile(jpm7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxjpm\_d1 <- as.numeric(jpm7\_gd[length(jpm7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnjpm\_i <- as.numeric(floor(quantile(jpm7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(jpm7\_gi$n)/2)  
mdnjpm\_i1 <- as.numeric(jpm7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jpm\_i <- as.numeric(floor(quantile(jpm7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jpm7\_gi$n)\*.75)  
q3jpm\_i1 <- as.numeric(jpm7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxjpm\_i <- as.numeric(floor(quantile(jpm7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxjpm\_i1 <- as.numeric(jpm7\_gi[length(jpm7\_gi$n),1])  
  
  
jpm\_g <- grep('JPM', ALL\_52$stock)  
  
ALL\_52[jpm\_g,18:29] <- c(mdnjpm\_d1, mdnjpm\_d, q3jpm\_d1, q3jpm\_d,  
 mxjpm\_d1, mxjpm\_d, mdnjpm\_i1, mdnjpm\_i,  
 q3jpm\_i1, q3jpm\_i, mxjpm\_i1, mxjpm\_i)

JWN

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
jwn7\_gd <- jwn7\_decr\_Days\_grouped  
jwn7\_gi <- jwn7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(jwn7\_gd) <- NULL  
row.names(jwn7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
jwn7\_gd <- jwn7\_gd[order(jwn7\_gd$n),]  
jwn7\_gi <- jwn7\_gi[order(jwn7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnjwn\_d <- as.numeric(floor(quantile(jwn7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(jwn7\_gd$n)/2)  
mdnjwn\_d1 <- as.numeric(jwn7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jwn\_d <- as.numeric(floor(quantile(jwn7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jwn7\_gd$n)\*.75)  
q3jwn\_d1 <- as.numeric(jwn7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxjwn\_d <- as.numeric(floor(quantile(jwn7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxjwn\_d1 <- as.numeric(jwn7\_gd[length(jwn7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnjwn\_i <- as.numeric(floor(quantile(jwn7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(jwn7\_gi$n)/2)  
mdnjwn\_i1 <- as.numeric(jwn7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3jwn\_i <- as.numeric(floor(quantile(jwn7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(jwn7\_gi$n)\*.75)  
q3jwn\_i1 <- as.numeric(jwn7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxjwn\_i <- as.numeric(floor(quantile(jwn7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxjwn\_i1 <- as.numeric(jwn7\_gi[length(jwn7\_gi$n),1])  
  
  
jwn\_g <- grep('JWN', ALL\_52$stock)  
  
ALL\_52[jwn\_g,18:29] <- c(mdnjwn\_d1, mdnjwn\_d, q3jwn\_d1, q3jwn\_d,  
 mxjwn\_d1, mxjwn\_d, mdnjwn\_i1, mdnjwn\_i,  
 q3jwn\_i1, q3jwn\_i, mxjwn\_i1, mxjwn\_i)

KGJI

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
kgji7\_gd <- kgji7\_decr\_Days\_grouped  
kgji7\_gi <- kgji7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(kgji7\_gd) <- NULL  
row.names(kgji7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
kgji7\_gd <- kgji7\_gd[order(kgji7\_gd$n),]  
kgji7\_gi <- kgji7\_gi[order(kgji7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnkgji\_d <- as.numeric(floor(quantile(kgji7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(kgji7\_gd$n)/2)  
mdnkgji\_d1 <- as.numeric(kgji7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3kgji\_d <- as.numeric(floor(quantile(kgji7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(kgji7\_gd$n)\*.75)  
q3kgji\_d1 <- as.numeric(kgji7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxkgji\_d <- as.numeric(floor(quantile(kgji7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxkgji\_d1 <- as.numeric(kgji7\_gd[length(kgji7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnkgji\_i <- as.numeric(floor(quantile(kgji7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(kgji7\_gi$n)/2)  
mdnkgji\_i1 <- as.numeric(kgji7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3kgji\_i <- as.numeric(floor(quantile(kgji7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(kgji7\_gi$n)\*.75)  
q3kgji\_i1 <- as.numeric(kgji7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxkgji\_i <- as.numeric(floor(quantile(kgji7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxkgji\_i1 <- as.numeric(kgji7\_gi[length(kgji7\_gi$n),1])  
  
  
kgji\_g <- grep('KGJI', ALL\_52$stock)  
  
ALL\_52[kgji\_g,18:29] <- c(mdnkgji\_d1, mdnkgji\_d, q3kgji\_d1, q3kgji\_d,  
 mxkgji\_d1, mxkgji\_d, mdnkgji\_i1, mdnkgji\_i,  
 q3kgji\_i1, q3kgji\_i, mxkgji\_i1, mxkgji\_i)

KSS

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
kss7\_gd <- kss7\_decr\_Days\_grouped  
kss7\_gi <- kss7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(kss7\_gd) <- NULL  
row.names(kss7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
kss7\_gd <- kss7\_gd[order(kss7\_gd$n),]  
kss7\_gi <- kss7\_gi[order(kss7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnkss\_d <- as.numeric(floor(quantile(kss7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(kss7\_gd$n)/2)  
mdnkss\_d1 <- as.numeric(kss7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3kss\_d <- as.numeric(floor(quantile(kss7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(kss7\_gd$n)\*.75)  
q3kss\_d1 <- as.numeric(kss7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxkss\_d <- as.numeric(floor(quantile(kss7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxkss\_d1 <- as.numeric(kss7\_gd[length(kss7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnkss\_i <- as.numeric(floor(quantile(kss7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(kss7\_gi$n)/2)  
mdnkss\_i1 <- as.numeric(kss7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3kss\_i <- as.numeric(floor(quantile(kss7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(kss7\_gi$n)\*.75)  
q3kss\_i1 <- as.numeric(kss7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxkss\_i <- as.numeric(floor(quantile(kss7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxkss\_i1 <- as.numeric(kss7\_gi[length(kss7\_gi$n),1])  
  
  
kss\_g <- grep('KSS', ALL\_52$stock)  
  
ALL\_52[kss\_g,18:29] <- c(mdnkss\_d1, mdnkss\_d, q3kss\_d1, q3kss\_d,  
 mxkss\_d1, mxkss\_d, mdnkss\_i1, mdnkss\_i,  
 q3kss\_i1, q3kss\_i, mxkss\_i1, mxkss\_i)

LUV

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
luv7\_gd <- luv7\_decr\_Days\_grouped  
luv7\_gi <- luv7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(luv7\_gd) <- NULL  
row.names(luv7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
luv7\_gd <- luv7\_gd[order(luv7\_gd$n),]  
luv7\_gi <- luv7\_gi[order(luv7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnluv\_d <- as.numeric(floor(quantile(luv7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(luv7\_gd$n)/2)  
mdnluv\_d1 <- as.numeric(luv7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3luv\_d <- as.numeric(floor(quantile(luv7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(luv7\_gd$n)\*.75)  
q3luv\_d1 <- as.numeric(luv7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxluv\_d <- as.numeric(floor(quantile(luv7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxluv\_d1 <- as.numeric(luv7\_gd[length(luv7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnluv\_i <- as.numeric(floor(quantile(luv7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(luv7\_gi$n)/2)  
mdnluv\_i1 <- as.numeric(luv7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3luv\_i <- as.numeric(floor(quantile(luv7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(luv7\_gi$n)\*.75)  
q3luv\_i1 <- as.numeric(luv7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxluv\_i <- as.numeric(floor(quantile(luv7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxluv\_i1 <- as.numeric(luv7\_gi[length(luv7\_gi$n),1])  
  
  
luv\_g <- grep('LUV', ALL\_52$stock)  
  
ALL\_52[luv\_g,18:29] <- c(mdnluv\_d1, mdnluv\_d, q3luv\_d1, q3luv\_d,  
 mxluv\_d1, mxluv\_d, mdnluv\_i1, mdnluv\_i,  
 q3luv\_i1, q3luv\_i, mxluv\_i1, mxluv\_i)

M

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
m7\_gd <- m7\_decr\_Days\_grouped  
m7\_gi <- m7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(m7\_gd) <- NULL  
row.names(m7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
m7\_gd <- m7\_gd[order(m7\_gd$n),]  
m7\_gi <- m7\_gi[order(m7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnm\_d <- as.numeric(floor(quantile(m7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(m7\_gd$n)/2)  
mdnm\_d1 <- as.numeric(m7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3m\_d <- as.numeric(floor(quantile(m7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(m7\_gd$n)\*.75)  
q3m\_d1 <- as.numeric(m7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxm\_d <- as.numeric(floor(quantile(m7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxm\_d1 <- as.numeric(m7\_gd[length(m7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnm\_i <- as.numeric(floor(quantile(m7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(m7\_gi$n)/2)  
mdnm\_i1 <- as.numeric(m7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3m\_i <- as.numeric(floor(quantile(m7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(m7\_gi$n)\*.75)  
q3m\_i1 <- as.numeric(m7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxm\_i <- as.numeric(floor(quantile(m7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxm\_i1 <- as.numeric(m7\_gi[length(m7\_gi$n),1])  
  
  
m\_g <- grep('^M$', ALL\_52$stock)  
  
ALL\_52[m\_g,18:29] <- c(mdnm\_d1, mdnm\_d, q3m\_d1, q3m\_d,  
 mxm\_d1, mxm\_d, mdnm\_i1, mdnm\_i,  
 q3m\_i1, q3m\_i, mxm\_i1, mxm\_i)

MGM

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
mgm7\_gd <- mgm7\_decr\_Days\_grouped  
mgm7\_gi <- mgm7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(mgm7\_gd) <- NULL  
row.names(mgm7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
mgm7\_gd <- mgm7\_gd[order(mgm7\_gd$n),]  
mgm7\_gi <- mgm7\_gi[order(mgm7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnmgm\_d <- as.numeric(floor(quantile(mgm7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(mgm7\_gd$n)/2)  
mdnmgm\_d1 <- as.numeric(mgm7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3mgm\_d <- as.numeric(floor(quantile(mgm7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(mgm7\_gd$n)\*.75)  
q3mgm\_d1 <- as.numeric(mgm7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxmgm\_d <- as.numeric(floor(quantile(mgm7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxmgm\_d1 <- as.numeric(mgm7\_gd[length(mgm7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnmgm\_i <- as.numeric(floor(quantile(mgm7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(mgm7\_gi$n)/2)  
mdnmgm\_i1 <- as.numeric(mgm7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3mgm\_i <- as.numeric(floor(quantile(mgm7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(mgm7\_gi$n)\*.75)  
q3mgm\_i1 <- as.numeric(mgm7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxmgm\_i <- as.numeric(floor(quantile(mgm7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxmgm\_i1 <- as.numeric(mgm7\_gi[length(mgm7\_gi$n),1])  
  
  
mgm\_g <- grep('MGM', ALL\_52$stock)  
  
ALL\_52[mgm\_g,18:29] <- c(mdnmgm\_d1, mdnmgm\_d, q3mgm\_d1, q3mgm\_d,  
 mxmgm\_d1, mxmgm\_d, mdnmgm\_i1, mdnmgm\_i,  
 q3mgm\_i1, q3mgm\_i, mxmgm\_i1, mxmgm\_i)

MSFT

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
msft7\_gd <- msft7\_decr\_Days\_grouped  
msft7\_gi <- msft7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(msft7\_gd) <- NULL  
row.names(msft7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
msft7\_gd <- msft7\_gd[order(msft7\_gd$n),]  
msft7\_gi <- msft7\_gi[order(msft7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnmsft\_d <- as.numeric(floor(quantile(msft7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(msft7\_gd$n)/2)  
mdnmsft\_d1 <- as.numeric(msft7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3msft\_d <- as.numeric(floor(quantile(msft7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(msft7\_gd$n)\*.75)  
q3msft\_d1 <- as.numeric(msft7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxmsft\_d <- as.numeric(floor(quantile(msft7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxmsft\_d1 <- as.numeric(msft7\_gd[length(msft7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnmsft\_i <- as.numeric(floor(quantile(msft7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(msft7\_gi$n)/2)  
mdnmsft\_i1 <- as.numeric(msft7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3msft\_i <- as.numeric(floor(quantile(msft7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(msft7\_gi$n)\*.75)  
q3msft\_i1 <- as.numeric(msft7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxmsft\_i <- as.numeric(floor(quantile(msft7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxmsft\_i1 <- as.numeric(msft7\_gi[length(msft7\_gi$n),1])  
  
  
msft\_g <- grep('MSFT', ALL\_52$stock)  
  
ALL\_52[msft\_g,18:29] <- c(mdnmsft\_d1, mdnmsft\_d, q3msft\_d1, q3msft\_d,  
 mxmsft\_d1, mxmsft\_d, mdnmsft\_i1, mdnmsft\_i,  
 q3msft\_i1, q3msft\_i, mxmsft\_i1, mxmsft\_i)

NFLX

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
nflx7\_gd <- nflx7\_decr\_Days\_grouped  
nflx7\_gi <- nflx7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(nflx7\_gd) <- NULL  
row.names(nflx7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
nflx7\_gd <- nflx7\_gd[order(nflx7\_gd$n),]  
nflx7\_gi <- nflx7\_gi[order(nflx7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnnflx\_d <- as.numeric(floor(quantile(nflx7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(nflx7\_gd$n)/2)  
mdnnflx\_d1 <- as.numeric(nflx7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nflx\_d <- as.numeric(floor(quantile(nflx7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nflx7\_gd$n)\*.75)  
q3nflx\_d1 <- as.numeric(nflx7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxnflx\_d <- as.numeric(floor(quantile(nflx7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxnflx\_d1 <- as.numeric(nflx7\_gd[length(nflx7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnnflx\_i <- as.numeric(floor(quantile(nflx7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(nflx7\_gi$n)/2)  
mdnnflx\_i1 <- as.numeric(nflx7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nflx\_i <- as.numeric(floor(quantile(nflx7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nflx7\_gi$n)\*.75)  
q3nflx\_i1 <- as.numeric(nflx7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxnflx\_i <- as.numeric(floor(quantile(nflx7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxnflx\_i1 <- as.numeric(nflx7\_gi[length(nflx7\_gi$n),1])  
  
  
nflx\_g <- grep('NFLX', ALL\_52$stock)  
  
ALL\_52[nflx\_g,18:29] <- c(mdnnflx\_d1, mdnnflx\_d, q3nflx\_d1, q3nflx\_d,  
 mxnflx\_d1, mxnflx\_d, mdnnflx\_i1, mdnnflx\_i,  
 q3nflx\_i1, q3nflx\_i, mxnflx\_i1, mxnflx\_i)

NKE

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
nke7\_gd <- nke7\_decr\_Days\_grouped  
nke7\_gi <- nke7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(nke7\_gd) <- NULL  
row.names(nke7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
nke7\_gd <- nke7\_gd[order(nke7\_gd$n),]  
nke7\_gi <- nke7\_gi[order(nke7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnnke\_d <- as.numeric(floor(quantile(nke7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(nke7\_gd$n)/2)  
mdnnke\_d1 <- as.numeric(nke7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nke\_d <- as.numeric(floor(quantile(nke7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nke7\_gd$n)\*.75)  
q3nke\_d1 <- as.numeric(nke7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxnke\_d <- as.numeric(floor(quantile(nke7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxnke\_d1 <- as.numeric(nke7\_gd[length(nke7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnnke\_i <- as.numeric(floor(quantile(nke7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(nke7\_gi$n)/2)  
mdnnke\_i1 <- as.numeric(nke7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nke\_i <- as.numeric(floor(quantile(nke7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nke7\_gi$n)\*.75)  
q3nke\_i1 <- as.numeric(nke7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxnke\_i <- as.numeric(floor(quantile(nke7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxnke\_i1 <- as.numeric(nke7\_gi[length(nke7\_gi$n),1])  
  
  
nke\_g <- grep('NKE', ALL\_52$stock)  
  
ALL\_52[nke\_g,18:29] <- c(mdnnke\_d1, mdnnke\_d, q3nke\_d1, q3nke\_d,  
 mxnke\_d1, mxnke\_d, mdnnke\_i1, mdnnke\_i,  
 q3nke\_i1, q3nke\_i, mxnke\_i1, mxnke\_i)

NSANY

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
nsany7\_gd <- nsany7\_decr\_Days\_grouped  
nsany7\_gi <- nsany7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(nsany7\_gd) <- NULL  
row.names(nsany7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
nsany7\_gd <- nsany7\_gd[order(nsany7\_gd$n),]  
nsany7\_gi <- nsany7\_gi[order(nsany7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnnsany\_d <- as.numeric(floor(quantile(nsany7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(nsany7\_gd$n)/2)  
mdnnsany\_d1 <- as.numeric(nsany7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nsany\_d <- as.numeric(floor(quantile(nsany7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nsany7\_gd$n)\*.75)  
q3nsany\_d1 <- as.numeric(nsany7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxnsany\_d <- as.numeric(floor(quantile(nsany7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxnsany\_d1 <- as.numeric(nsany7\_gd[length(nsany7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnnsany\_i <- as.numeric(floor(quantile(nsany7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(nsany7\_gi$n)/2)  
mdnnsany\_i1 <- as.numeric(nsany7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nsany\_i <- as.numeric(floor(quantile(nsany7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nsany7\_gi$n)\*.75)  
q3nsany\_i1 <- as.numeric(nsany7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxnsany\_i <- as.numeric(floor(quantile(nsany7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxnsany\_i1 <- as.numeric(nsany7\_gi[length(nsany7\_gi$n),1])  
  
  
nsany\_g <- grep('NSANY', ALL\_52$stock)  
  
ALL\_52[nsany\_g,18:29] <- c(mdnnsany\_d1, mdnnsany\_d, q3nsany\_d1, q3nsany\_d,  
 mxnsany\_d1, mxnsany\_d, mdnnsany\_i1, mdnnsany\_i,  
 q3nsany\_i1, q3nsany\_i, mxnsany\_i1, mxnsany\_i)

NUS

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
nus7\_gd <- nus7\_decr\_Days\_grouped  
nus7\_gi <- nus7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(nus7\_gd) <- NULL  
row.names(nus7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
nus7\_gd <- nus7\_gd[order(nus7\_gd$n),]  
nus7\_gi <- nus7\_gi[order(nus7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnnus\_d <- as.numeric(floor(quantile(nus7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(nus7\_gd$n)/2)  
mdnnus\_d1 <- as.numeric(nus7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nus\_d <- as.numeric(floor(quantile(nus7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nus7\_gd$n)\*.75)  
q3nus\_d1 <- as.numeric(nus7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxnus\_d <- as.numeric(floor(quantile(nus7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxnus\_d1 <- as.numeric(nus7\_gd[length(nus7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnnus\_i <- as.numeric(floor(quantile(nus7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(nus7\_gi$n)/2)  
mdnnus\_i1 <- as.numeric(nus7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3nus\_i <- as.numeric(floor(quantile(nus7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(nus7\_gi$n)\*.75)  
q3nus\_i1 <- as.numeric(nus7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxnus\_i <- as.numeric(floor(quantile(nus7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxnus\_i1 <- as.numeric(nus7\_gi[length(nus7\_gi$n),1])  
  
  
nus\_g <- grep('NUS', ALL\_52$stock)  
  
ALL\_52[nus\_g,18:29] <- c(mdnnus\_d1, mdnnus\_d, q3nus\_d1, q3nus\_d,  
 mxnus\_d1, mxnus\_d, mdnnus\_i1, mdnnus\_i,  
 q3nus\_i1, q3nus\_i, mxnus\_i1, mxnus\_i)

ONCY

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
oncy7\_gd <- oncy7\_decr\_Days\_grouped  
oncy7\_gi <- oncy7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(oncy7\_gd) <- NULL  
row.names(oncy7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
oncy7\_gd <- oncy7\_gd[order(oncy7\_gd$n),]  
oncy7\_gi <- oncy7\_gi[order(oncy7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnoncy\_d <- as.numeric(floor(quantile(oncy7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(oncy7\_gd$n)/2)  
mdnoncy\_d1 <- as.numeric(oncy7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3oncy\_d <- as.numeric(floor(quantile(oncy7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(oncy7\_gd$n)\*.75)  
q3oncy\_d1 <- as.numeric(oncy7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxoncy\_d <- as.numeric(floor(quantile(oncy7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxoncy\_d1 <- as.numeric(oncy7\_gd[length(oncy7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnoncy\_i <- as.numeric(floor(quantile(oncy7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(oncy7\_gi$n)/2)  
mdnoncy\_i1 <- as.numeric(oncy7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3oncy\_i <- as.numeric(floor(quantile(oncy7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(oncy7\_gi$n)\*.75)  
q3oncy\_i1 <- as.numeric(oncy7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxoncy\_i <- as.numeric(floor(quantile(oncy7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxoncy\_i1 <- as.numeric(oncy7\_gi[length(oncy7\_gi$n),1])  
  
  
oncy\_g <- grep('ONCY', ALL\_52$stock)  
  
ALL\_52[oncy\_g,18:29] <- c(mdnoncy\_d1, mdnoncy\_d, q3oncy\_d1, q3oncy\_d,  
 mxoncy\_d1, mxoncy\_d, mdnoncy\_i1, mdnoncy\_i,  
 q3oncy\_i1, q3oncy\_i, mxoncy\_i1, mxoncy\_i)

PCG

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
pcg7\_gd <- pcg7\_decr\_Days\_grouped  
pcg7\_gi <- pcg7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(pcg7\_gd) <- NULL  
row.names(pcg7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
pcg7\_gd <- pcg7\_gd[order(pcg7\_gd$n),]  
pcg7\_gi <- pcg7\_gi[order(pcg7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnpcg\_d <- as.numeric(floor(quantile(pcg7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(pcg7\_gd$n)/2)  
mdnpcg\_d1 <- as.numeric(pcg7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3pcg\_d <- as.numeric(floor(quantile(pcg7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(pcg7\_gd$n)\*.75)  
q3pcg\_d1 <- as.numeric(pcg7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxpcg\_d <- as.numeric(floor(quantile(pcg7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxpcg\_d1 <- as.numeric(pcg7\_gd[length(pcg7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnpcg\_i <- as.numeric(floor(quantile(pcg7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(pcg7\_gi$n)/2)  
mdnpcg\_i1 <- as.numeric(pcg7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3pcg\_i <- as.numeric(floor(quantile(pcg7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(pcg7\_gi$n)\*.75)  
q3pcg\_i1 <- as.numeric(pcg7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxpcg\_i <- as.numeric(floor(quantile(pcg7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxpcg\_i1 <- as.numeric(pcg7\_gi[length(pcg7\_gi$n),1])  
  
  
pcg\_g <- grep('PCG', ALL\_52$stock)  
  
ALL\_52[pcg\_g,18:29] <- c(mdnpcg\_d1, mdnpcg\_d, q3pcg\_d1, q3pcg\_d,  
 mxpcg\_d1, mxpcg\_d, mdnpcg\_i1, mdnpcg\_i,  
 q3pcg\_i1, q3pcg\_i, mxpcg\_i1, mxpcg\_i)

ROST

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
rost7\_gd <- rost7\_decr\_Days\_grouped  
rost7\_gi <- rost7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(rost7\_gd) <- NULL  
row.names(rost7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
rost7\_gd <- rost7\_gd[order(rost7\_gd$n),]  
rost7\_gi <- rost7\_gi[order(rost7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnrost\_d <- as.numeric(floor(quantile(rost7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(rost7\_gd$n)/2)  
mdnrost\_d1 <- as.numeric(rost7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3rost\_d <- as.numeric(floor(quantile(rost7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(rost7\_gd$n)\*.75)  
q3rost\_d1 <- as.numeric(rost7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxrost\_d <- as.numeric(floor(quantile(rost7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxrost\_d1 <- as.numeric(rost7\_gd[length(rost7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnrost\_i <- as.numeric(floor(quantile(rost7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(rost7\_gi$n)/2)  
mdnrost\_i1 <- as.numeric(rost7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3rost\_i <- as.numeric(floor(quantile(rost7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(rost7\_gi$n)\*.75)  
q3rost\_i1 <- as.numeric(rost7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxrost\_i <- as.numeric(floor(quantile(rost7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxrost\_i1 <- as.numeric(rost7\_gi[length(rost7\_gi$n),1])  
  
  
rost\_g <- grep('ROST', ALL\_52$stock)  
  
ALL\_52[rost\_g,18:29] <- c(mdnrost\_d1, mdnrost\_d, q3rost\_d1, q3rost\_d,  
 mxrost\_d1, mxrost\_d, mdnrost\_i1, mdnrost\_i,  
 q3rost\_i1, q3rost\_i, mxrost\_i1, mxrost\_i)

RRGB

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
rrgb7\_gd <- rrgb7\_decr\_Days\_grouped  
rrgb7\_gi <- rrgb7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(rrgb7\_gd) <- NULL  
row.names(rrgb7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
rrgb7\_gd <- rrgb7\_gd[order(rrgb7\_gd$n),]  
rrgb7\_gi <- rrgb7\_gi[order(rrgb7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnrrgb\_d <- as.numeric(floor(quantile(rrgb7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(rrgb7\_gd$n)/2)  
mdnrrgb\_d1 <- as.numeric(rrgb7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3rrgb\_d <- as.numeric(floor(quantile(rrgb7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(rrgb7\_gd$n)\*.75)  
q3rrgb\_d1 <- as.numeric(rrgb7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxrrgb\_d <- as.numeric(floor(quantile(rrgb7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxrrgb\_d1 <- as.numeric(rrgb7\_gd[length(rrgb7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnrrgb\_i <- as.numeric(floor(quantile(rrgb7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(rrgb7\_gi$n)/2)  
mdnrrgb\_i1 <- as.numeric(rrgb7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3rrgb\_i <- as.numeric(floor(quantile(rrgb7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(rrgb7\_gi$n)\*.75)  
q3rrgb\_i1 <- as.numeric(rrgb7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxrrgb\_i <- as.numeric(floor(quantile(rrgb7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxrrgb\_i1 <- as.numeric(rrgb7\_gi[length(rrgb7\_gi$n),1])  
  
  
rrgb\_g <- grep('RRGB', ALL\_52$stock)  
  
ALL\_52[rrgb\_g,18:29] <- c(mdnrrgb\_d1, mdnrrgb\_d, q3rrgb\_d1, q3rrgb\_d,  
 mxrrgb\_d1, mxrrgb\_d, mdnrrgb\_i1, mdnrrgb\_i,  
 q3rrgb\_i1, q3rrgb\_i, mxrrgb\_i1, mxrrgb\_i)

S

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
s7\_gd <- s7\_decr\_Days\_grouped  
s7\_gi <- s7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(s7\_gd) <- NULL  
row.names(s7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
s7\_gd <- s7\_gd[order(s7\_gd$n),]  
s7\_gi <- s7\_gi[order(s7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdns\_d <- as.numeric(floor(quantile(s7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(s7\_gd$n)/2)  
mdns\_d1 <- as.numeric(s7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3s\_d <- as.numeric(floor(quantile(s7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(s7\_gd$n)\*.75)  
q3s\_d1 <- as.numeric(s7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxs\_d <- as.numeric(floor(quantile(s7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxs\_d1 <- as.numeric(s7\_gd[length(s7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdns\_i <- as.numeric(floor(quantile(s7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(s7\_gi$n)/2)  
mdns\_i1 <- as.numeric(s7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3s\_i <- as.numeric(floor(quantile(s7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(s7\_gi$n)\*.75)  
q3s\_i1 <- as.numeric(s7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxs\_i <- as.numeric(floor(quantile(s7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxs\_i1 <- as.numeric(s7\_gi[length(s7\_gi$n),1])  
  
  
s\_g <- grep('S$', ALL\_52$stock)  
  
ALL\_52[s\_g,18:29] <- c(mdns\_d1, mdns\_d, q3s\_d1, q3s\_d,  
 mxs\_d1, mxs\_d, mdns\_i1, mdns\_i,  
 q3s\_i1, q3s\_i, mxs\_i1, mxs\_i)

SIG

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
sig7\_gd <- sig7\_decr\_Days\_grouped  
sig7\_gi <- sig7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(sig7\_gd) <- NULL  
row.names(sig7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
sig7\_gd <- sig7\_gd[order(sig7\_gd$n),]  
sig7\_gi <- sig7\_gi[order(sig7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnsig\_d <- as.numeric(floor(quantile(sig7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(sig7\_gd$n)/2)  
mdnsig\_d1 <- as.numeric(sig7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3sig\_d <- as.numeric(floor(quantile(sig7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(sig7\_gd$n)\*.75)  
q3sig\_d1 <- as.numeric(sig7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxsig\_d <- as.numeric(floor(quantile(sig7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxsig\_d1 <- as.numeric(sig7\_gd[length(sig7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnsig\_i <- as.numeric(floor(quantile(sig7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(sig7\_gi$n)/2)  
mdnsig\_i1 <- as.numeric(sig7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3sig\_i <- as.numeric(floor(quantile(sig7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(sig7\_gi$n)\*.75)  
q3sig\_i1 <- as.numeric(sig7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxsig\_i <- as.numeric(floor(quantile(sig7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxsig\_i1 <- as.numeric(sig7\_gi[length(sig7\_gi$n),1])  
  
  
sig\_g <- grep('SIG', ALL\_52$stock)  
  
ALL\_52[sig\_g,18:29] <- c(mdnsig\_d1, mdnsig\_d, q3sig\_d1, q3sig\_d,  
 mxsig\_d1, mxsig\_d, mdnsig\_i1, mdnsig\_i,  
 q3sig\_i1, q3sig\_i, mxsig\_i1, mxsig\_i)

T

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
t7\_gd <- t7\_decr\_Days\_grouped  
t7\_gi <- t7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(t7\_gd) <- NULL  
row.names(t7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
t7\_gd <- t7\_gd[order(t7\_gd$n),]  
t7\_gi <- t7\_gi[order(t7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnt\_d <- as.numeric(floor(quantile(t7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(t7\_gd$n)/2)  
mdnt\_d1 <- as.numeric(t7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3t\_d <- as.numeric(floor(quantile(t7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(t7\_gd$n)\*.75)  
q3t\_d1 <- as.numeric(t7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxt\_d <- as.numeric(floor(quantile(t7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxt\_d1 <- as.numeric(t7\_gd[length(t7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnt\_i <- as.numeric(floor(quantile(t7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(t7\_gi$n)/2)  
mdnt\_i1 <- as.numeric(t7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3t\_i <- as.numeric(floor(quantile(t7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(t7\_gi$n)\*.75)  
q3t\_i1 <- as.numeric(t7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxt\_i <- as.numeric(floor(quantile(t7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxt\_i1 <- as.numeric(t7\_gi[length(t7\_gi$n),1])  
  
  
t\_g <- grep('^T$', ALL\_52$stock)  
  
ALL\_52[t\_g,18:29] <- c(mdnt\_d1, mdnt\_d, q3t\_d1, q3t\_d,  
 mxt\_d1, mxt\_d, mdnt\_i1, mdnt\_i,  
 q3t\_i1, q3t\_i, mxt\_i1, mxt\_i)

TEVA

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
teva7\_gd <- teva7\_decr\_Days\_grouped  
teva7\_gi <- teva7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(teva7\_gd) <- NULL  
row.names(teva7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
teva7\_gd <- teva7\_gd[order(teva7\_gd$n),]  
teva7\_gi <- teva7\_gi[order(teva7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnteva\_d <- as.numeric(floor(quantile(teva7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(teva7\_gd$n)/2)  
mdnteva\_d1 <- as.numeric(teva7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3teva\_d <- as.numeric(floor(quantile(teva7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(teva7\_gd$n)\*.75)  
q3teva\_d1 <- as.numeric(teva7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxteva\_d <- as.numeric(floor(quantile(teva7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxteva\_d1 <- as.numeric(teva7\_gd[length(teva7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnteva\_i <- as.numeric(floor(quantile(teva7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(teva7\_gi$n)/2)  
mdnteva\_i1 <- as.numeric(teva7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3teva\_i <- as.numeric(floor(quantile(teva7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(teva7\_gi$n)\*.75)  
q3teva\_i1 <- as.numeric(teva7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxteva\_i <- as.numeric(floor(quantile(teva7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxteva\_i1 <- as.numeric(teva7\_gi[length(teva7\_gi$n),1])  
  
  
teva\_g <- grep('TEVA', ALL\_52$stock)  
  
ALL\_52[teva\_g,18:29] <- c(mdnteva\_d1, mdnteva\_d, q3teva\_d1, q3teva\_d,  
 mxteva\_d1, mxteva\_d, mdnteva\_i1, mdnteva\_i,  
 q3teva\_i1, q3teva\_i, mxteva\_i1, mxteva\_i)

TGT

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
tgt7\_gd <- tgt7\_decr\_Days\_grouped  
tgt7\_gi <- tgt7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(tgt7\_gd) <- NULL  
row.names(tgt7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
tgt7\_gd <- tgt7\_gd[order(tgt7\_gd$n),]  
tgt7\_gi <- tgt7\_gi[order(tgt7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdntgt\_d <- as.numeric(floor(quantile(tgt7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(tgt7\_gd$n)/2)  
mdntgt\_d1 <- as.numeric(tgt7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tgt\_d <- as.numeric(floor(quantile(tgt7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tgt7\_gd$n)\*.75)  
q3tgt\_d1 <- as.numeric(tgt7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxtgt\_d <- as.numeric(floor(quantile(tgt7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxtgt\_d1 <- as.numeric(tgt7\_gd[length(tgt7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdntgt\_i <- as.numeric(floor(quantile(tgt7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(tgt7\_gi$n)/2)  
mdntgt\_i1 <- as.numeric(tgt7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tgt\_i <- as.numeric(floor(quantile(tgt7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tgt7\_gi$n)\*.75)  
q3tgt\_i1 <- as.numeric(tgt7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxtgt\_i <- as.numeric(floor(quantile(tgt7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxtgt\_i1 <- as.numeric(tgt7\_gi[length(tgt7\_gi$n),1])  
  
  
tgt\_g <- grep('TGT', ALL\_52$stock)  
  
ALL\_52[tgt\_g,18:29] <- c(mdntgt\_d1, mdntgt\_d, q3tgt\_d1, q3tgt\_d,  
 mxtgt\_d1, mxtgt\_d, mdntgt\_i1, mdntgt\_i,  
 q3tgt\_i1, q3tgt\_i, mxtgt\_i1, mxtgt\_i)

TJX

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
tjx7\_gd <- tjx7\_decr\_Days\_grouped  
tjx7\_gi <- tjx7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(tjx7\_gd) <- NULL  
row.names(tjx7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
tjx7\_gd <- tjx7\_gd[order(tjx7\_gd$n),]  
tjx7\_gi <- tjx7\_gi[order(tjx7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdntjx\_d <- as.numeric(floor(quantile(tjx7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(tjx7\_gd$n)/2)  
mdntjx\_d1 <- as.numeric(tjx7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tjx\_d <- as.numeric(floor(quantile(tjx7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tjx7\_gd$n)\*.75)  
q3tjx\_d1 <- as.numeric(tjx7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxtjx\_d <- as.numeric(floor(quantile(tjx7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxtjx\_d1 <- as.numeric(tjx7\_gd[length(tjx7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdntjx\_i <- as.numeric(floor(quantile(tjx7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(tjx7\_gi$n)/2)  
mdntjx\_i1 <- as.numeric(tjx7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tjx\_i <- as.numeric(floor(quantile(tjx7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tjx7\_gi$n)\*.75)  
q3tjx\_i1 <- as.numeric(tjx7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxtjx\_i <- as.numeric(floor(quantile(tjx7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxtjx\_i1 <- as.numeric(tjx7\_gi[length(tjx7\_gi$n),1])  
  
  
tjx\_g <- grep('TJX', ALL\_52$stock)  
  
ALL\_52[tjx\_g,18:29] <- c(mdntjx\_d1, mdntjx\_d, q3tjx\_d1, q3tjx\_d,  
 mxtjx\_d1, mxtjx\_d, mdntjx\_i1, mdntjx\_i,  
 q3tjx\_i1, q3tjx\_i, mxtjx\_i1, mxtjx\_i)

TM

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
tm7\_gd <- tm7\_decr\_Days\_grouped  
tm7\_gi <- tm7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(tm7\_gd) <- NULL  
row.names(tm7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
tm7\_gd <- tm7\_gd[order(tm7\_gd$n),]  
tm7\_gi <- tm7\_gi[order(tm7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdntm\_d <- as.numeric(floor(quantile(tm7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(tm7\_gd$n)/2)  
mdntm\_d1 <- as.numeric(tm7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tm\_d <- as.numeric(floor(quantile(tm7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tm7\_gd$n)\*.75)  
q3tm\_d1 <- as.numeric(tm7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxtm\_d <- as.numeric(floor(quantile(tm7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxtm\_d1 <- as.numeric(tm7\_gd[length(tm7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdntm\_i <- as.numeric(floor(quantile(tm7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(tm7\_gi$n)/2)  
mdntm\_i1 <- as.numeric(tm7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3tm\_i <- as.numeric(floor(quantile(tm7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(tm7\_gi$n)\*.75)  
q3tm\_i1 <- as.numeric(tm7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxtm\_i <- as.numeric(floor(quantile(tm7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxtm\_i1 <- as.numeric(tm7\_gi[length(tm7\_gi$n),1])  
  
  
tm\_g <- grep('^TM$', ALL\_52$stock)  
  
ALL\_52[tm\_g,18:29] <- c(mdntm\_d1, mdntm\_d, q3tm\_d1, q3tm\_d,  
 mxtm\_d1, mxtm\_d, mdntm\_i1, mdntm\_i,  
 q3tm\_i1, q3tm\_i, mxtm\_i1, mxtm\_i)

UBSI

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
ubsi7\_gd <- ubsi7\_decr\_Days\_grouped  
ubsi7\_gi <- ubsi7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(ubsi7\_gd) <- NULL  
row.names(ubsi7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
ubsi7\_gd <- ubsi7\_gd[order(ubsi7\_gd$n),]  
ubsi7\_gi <- ubsi7\_gi[order(ubsi7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnubsi\_d <- as.numeric(floor(quantile(ubsi7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(ubsi7\_gd$n)/2)  
mdnubsi\_d1 <- as.numeric(ubsi7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ubsi\_d <- as.numeric(floor(quantile(ubsi7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ubsi7\_gd$n)\*.75)  
q3ubsi\_d1 <- as.numeric(ubsi7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxubsi\_d <- as.numeric(floor(quantile(ubsi7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxubsi\_d1 <- as.numeric(ubsi7\_gd[length(ubsi7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnubsi\_i <- as.numeric(floor(quantile(ubsi7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(ubsi7\_gi$n)/2)  
mdnubsi\_i1 <- as.numeric(ubsi7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3ubsi\_i <- as.numeric(floor(quantile(ubsi7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(ubsi7\_gi$n)\*.75)  
q3ubsi\_i1 <- as.numeric(ubsi7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxubsi\_i <- as.numeric(floor(quantile(ubsi7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxubsi\_i1 <- as.numeric(ubsi7\_gi[length(ubsi7\_gi$n),1])  
  
  
ubsi\_g <- grep('UBSI', ALL\_52$stock)  
  
ALL\_52[ubsi\_g,18:29] <- c(mdnubsi\_d1, mdnubsi\_d, q3ubsi\_d1, q3ubsi\_d,  
 mxubsi\_d1, mxubsi\_d, mdnubsi\_i1, mdnubsi\_i,  
 q3ubsi\_i1, q3ubsi\_i, mxubsi\_i1, mxubsi\_i)

VZ

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
vz7\_gd <- vz7\_decr\_Days\_grouped  
vz7\_gi <- vz7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(vz7\_gd) <- NULL  
row.names(vz7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
vz7\_gd <- vz7\_gd[order(vz7\_gd$n),]  
vz7\_gi <- vz7\_gi[order(vz7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnvz\_d <- as.numeric(floor(quantile(vz7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(vz7\_gd$n)/2)  
mdnvz\_d1 <- as.numeric(vz7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3vz\_d <- as.numeric(floor(quantile(vz7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(vz7\_gd$n)\*.75)  
q3vz\_d1 <- as.numeric(vz7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxvz\_d <- as.numeric(floor(quantile(vz7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxvz\_d1 <- as.numeric(vz7\_gd[length(vz7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnvz\_i <- as.numeric(floor(quantile(vz7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(vz7\_gi$n)/2)  
mdnvz\_i1 <- as.numeric(vz7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3vz\_i <- as.numeric(floor(quantile(vz7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(vz7\_gi$n)\*.75)  
q3vz\_i1 <- as.numeric(vz7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxvz\_i <- as.numeric(floor(quantile(vz7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxvz\_i1 <- as.numeric(vz7\_gi[length(vz7\_gi$n),1])  
  
  
vz\_g <- grep('VZ', ALL\_52$stock)  
  
ALL\_52[vz\_g,18:29] <- c(mdnvz\_d1, mdnvz\_d, q3vz\_d1, q3vz\_d,  
 mxvz\_d1, mxvz\_d, mdnvz\_i1, mdnvz\_i,  
 q3vz\_i1, q3vz\_i, mxvz\_i1, mxvz\_i)

WFC

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
wfc7\_gd <- wfc7\_decr\_Days\_grouped  
wfc7\_gi <- wfc7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(wfc7\_gd) <- NULL  
row.names(wfc7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
wfc7\_gd <- wfc7\_gd[order(wfc7\_gd$n),]  
wfc7\_gi <- wfc7\_gi[order(wfc7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnwfc\_d <- as.numeric(floor(quantile(wfc7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(wfc7\_gd$n)/2)  
mdnwfc\_d1 <- as.numeric(wfc7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wfc\_d <- as.numeric(floor(quantile(wfc7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wfc7\_gd$n)\*.75)  
q3wfc\_d1 <- as.numeric(wfc7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxwfc\_d <- as.numeric(floor(quantile(wfc7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxwfc\_d1 <- as.numeric(wfc7\_gd[length(wfc7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnwfc\_i <- as.numeric(floor(quantile(wfc7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(wfc7\_gi$n)/2)  
mdnwfc\_i1 <- as.numeric(wfc7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wfc\_i <- as.numeric(floor(quantile(wfc7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wfc7\_gi$n)\*.75)  
q3wfc\_i1 <- as.numeric(wfc7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxwfc\_i <- as.numeric(floor(quantile(wfc7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxwfc\_i1 <- as.numeric(wfc7\_gi[length(wfc7\_gi$n),1])  
  
  
wfc\_g <- grep('WFC', ALL\_52$stock)  
  
ALL\_52[wfc\_g,18:29] <- c(mdnwfc\_d1, mdnwfc\_d, q3wfc\_d1, q3wfc\_d,  
 mxwfc\_d1, mxwfc\_d, mdnwfc\_i1, mdnwfc\_i,  
 q3wfc\_i1, q3wfc\_i, mxwfc\_i1, mxwfc\_i)

WM

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
wm7\_gd <- wm7\_decr\_Days\_grouped  
wm7\_gi <- wm7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(wm7\_gd) <- NULL  
row.names(wm7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
wm7\_gd <- wm7\_gd[order(wm7\_gd$n),]  
wm7\_gi <- wm7\_gi[order(wm7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnwm\_d <- as.numeric(floor(quantile(wm7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(wm7\_gd$n)/2)  
mdnwm\_d1 <- as.numeric(wm7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wm\_d <- as.numeric(floor(quantile(wm7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wm7\_gd$n)\*.75)  
q3wm\_d1 <- as.numeric(wm7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxwm\_d <- as.numeric(floor(quantile(wm7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxwm\_d1 <- as.numeric(wm7\_gd[length(wm7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnwm\_i <- as.numeric(floor(quantile(wm7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(wm7\_gi$n)/2)  
mdnwm\_i1 <- as.numeric(wm7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wm\_i <- as.numeric(floor(quantile(wm7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wm7\_gi$n)\*.75)  
q3wm\_i1 <- as.numeric(wm7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxwm\_i <- as.numeric(floor(quantile(wm7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxwm\_i1 <- as.numeric(wm7\_gi[length(wm7\_gi$n),1])  
  
  
wm\_g <- grep('^WM$', ALL\_52$stock)  
  
ALL\_52[wm\_g,18:29] <- c(mdnwm\_d1, mdnwm\_d, q3wm\_d1, q3wm\_d,  
 mxwm\_d1, mxwm\_d, mdnwm\_i1, mdnwm\_i,  
 q3wm\_i1, q3wm\_i, mxwm\_i1, mxwm\_i)

WMT

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
wmt7\_gd <- wmt7\_decr\_Days\_grouped  
wmt7\_gi <- wmt7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(wmt7\_gd) <- NULL  
row.names(wmt7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
wmt7\_gd <- wmt7\_gd[order(wmt7\_gd$n),]  
wmt7\_gi <- wmt7\_gi[order(wmt7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnwmt\_d <- as.numeric(floor(quantile(wmt7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(wmt7\_gd$n)/2)  
mdnwmt\_d1 <- as.numeric(wmt7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wmt\_d <- as.numeric(floor(quantile(wmt7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wmt7\_gd$n)\*.75)  
q3wmt\_d1 <- as.numeric(wmt7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxwmt\_d <- as.numeric(floor(quantile(wmt7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxwmt\_d1 <- as.numeric(wmt7\_gd[length(wmt7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnwmt\_i <- as.numeric(floor(quantile(wmt7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(wmt7\_gi$n)/2)  
mdnwmt\_i1 <- as.numeric(wmt7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wmt\_i <- as.numeric(floor(quantile(wmt7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wmt7\_gi$n)\*.75)  
q3wmt\_i1 <- as.numeric(wmt7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxwmt\_i <- as.numeric(floor(quantile(wmt7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxwmt\_i1 <- as.numeric(wmt7\_gi[length(wmt7\_gi$n),1])  
  
  
wmt\_g <- grep('WMT$', ALL\_52$stock)  
  
ALL\_52[wmt\_g,18:29] <- c(mdnwmt\_d1, mdnwmt\_d, q3wmt\_d1, q3wmt\_d,  
 mxwmt\_d1, mxwmt\_d, mdnwmt\_i1, mdnwmt\_i,  
 q3wmt\_i1, q3wmt\_i, mxwmt\_i1, mxwmt\_i)

WWE

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
wwe7\_gd <- wwe7\_decr\_Days\_grouped  
wwe7\_gi <- wwe7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(wwe7\_gd) <- NULL  
row.names(wwe7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
wwe7\_gd <- wwe7\_gd[order(wwe7\_gd$n),]  
wwe7\_gi <- wwe7\_gi[order(wwe7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnwwe\_d <- as.numeric(floor(quantile(wwe7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(wwe7\_gd$n)/2)  
mdnwwe\_d1 <- as.numeric(wwe7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wwe\_d <- as.numeric(floor(quantile(wwe7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wwe7\_gd$n)\*.75)  
q3wwe\_d1 <- as.numeric(wwe7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxwwe\_d <- as.numeric(floor(quantile(wwe7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxwwe\_d1 <- as.numeric(wwe7\_gd[length(wwe7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnwwe\_i <- as.numeric(floor(quantile(wwe7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(wwe7\_gi$n)/2)  
mdnwwe\_i1 <- as.numeric(wwe7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3wwe\_i <- as.numeric(floor(quantile(wwe7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(wwe7\_gi$n)\*.75)  
q3wwe\_i1 <- as.numeric(wwe7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxwwe\_i <- as.numeric(floor(quantile(wwe7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxwwe\_i1 <- as.numeric(wwe7\_gi[length(wwe7\_gi$n),1])  
  
  
wwe\_g <- grep('WWE', ALL\_52$stock)  
  
ALL\_52[wwe\_g,18:29] <- c(mdnwwe\_d1, mdnwwe\_d, q3wwe\_d1, q3wwe\_d,  
 mxwwe\_d1, mxwwe\_d, mdnwwe\_i1, mdnwwe\_i,  
 q3wwe\_i1, q3wwe\_i, mxwwe\_i1, mxwwe\_i)

XOM

#the grouped tables for the decreasing and increasing counts as sets  
#where each set of counts happened one or more times  
xom7\_gd <- xom7\_decr\_Days\_grouped  
xom7\_gi <- xom7\_incr\_Days\_grouped  
  
#removes the row names   
row.names(xom7\_gd) <- NULL  
row.names(xom7\_gi) <- NULL  
  
#orders the tables by n counts of decreasing days, so indexing can be done  
xom7\_gd <- xom7\_gd[order(xom7\_gd$n),]  
xom7\_gi <- xom7\_gi[order(xom7\_gi$n),]  
  
#the median value of counts for the number of times a stock decr in this time span  
mdnxom\_d <- as.numeric(floor(quantile(xom7\_gd$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of decr days occurred in this time series  
c1 <- ceiling(length(xom7\_gd$n)/2)  
mdnxom\_d1 <- as.numeric(xom7\_gd[c1,1])  
  
#the 3rd quantile of counts of sets decr days in this time span from a quantile tibble  
#has to be set to numeric  
q3xom\_d <- as.numeric(floor(quantile(xom7\_gd$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the decr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(xom7\_gd$n)\*.75)  
q3xom\_d1 <- as.numeric(xom7\_gd[c2,1])  
  
#the max value of times a set's decr days occurred in this time span  
mxxom\_d <- as.numeric(floor(quantile(xom7\_gd$n, probs=1, na.rm=TRUE)))  
  
#the set of decreasing days in the max counts  
mxxom\_d1 <- as.numeric(xom7\_gd[length(xom7\_gd$n),1])  
  
  
#the median value of counts for the number of times a stock incr in this time span  
mdnxom\_i <- as.numeric(floor(quantile(xom7\_gi$n, probs=0.5, na.rm=TRUE)))  
  
#the index of the median number of times a sum of incr days occurred in this time series  
c1 <- ceiling(length(xom7\_gi$n)/2)  
mdnxom\_i1 <- as.numeric(xom7\_gi[c1,1])  
  
#the 3rd quantile of counts of sets incr days in this time span from a quantile tibble  
#has to be set to numeric  
q3xom\_i <- as.numeric(floor(quantile(xom7\_gi$n, probs=0.75, na.rm=TRUE)))  
  
#the index of the 3rd quantile, to get the incr set that occured 75% of the time in counts for this time span  
c2 <- floor(length(xom7\_gi$n)\*.75)  
q3xom\_i1 <- as.numeric(xom7\_gi[c2,1])  
  
#the max value of times a set's incr days occurred in this time span  
mxxom\_i <- as.numeric(floor(quantile(xom7\_gi$n, probs=1, na.rm=TRUE)))  
  
#the set of increasing days in the max counts  
mxxom\_i1 <- as.numeric(xom7\_gi[length(xom7\_gi$n),1])  
  
  
xom\_g <- grep('XOM', ALL\_52$stock)  
  
ALL\_52[xom\_g,18:29] <- c(mdnxom\_d1, mdnxom\_d, q3xom\_d1, q3xom\_d,  
 mxxom\_d1, mxxom\_d, mdnxom\_i1, mdnxom\_i,  
 q3xom\_i1, q3xom\_i, mxxom\_i1, mxxom\_i)

Write this file out to csv.

write.csv(ALL\_52, 'ALL\_52\_L7\_grouped\_counts.csv', row.names=TRUE)