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.

```r
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 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)
```

```r
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_VolumeRatioDaily2In
itial*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]
```

```r
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.

```r
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)])
```

```r
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.

```r
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
```

```r
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.

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

All the columns we now have are:

```r
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.

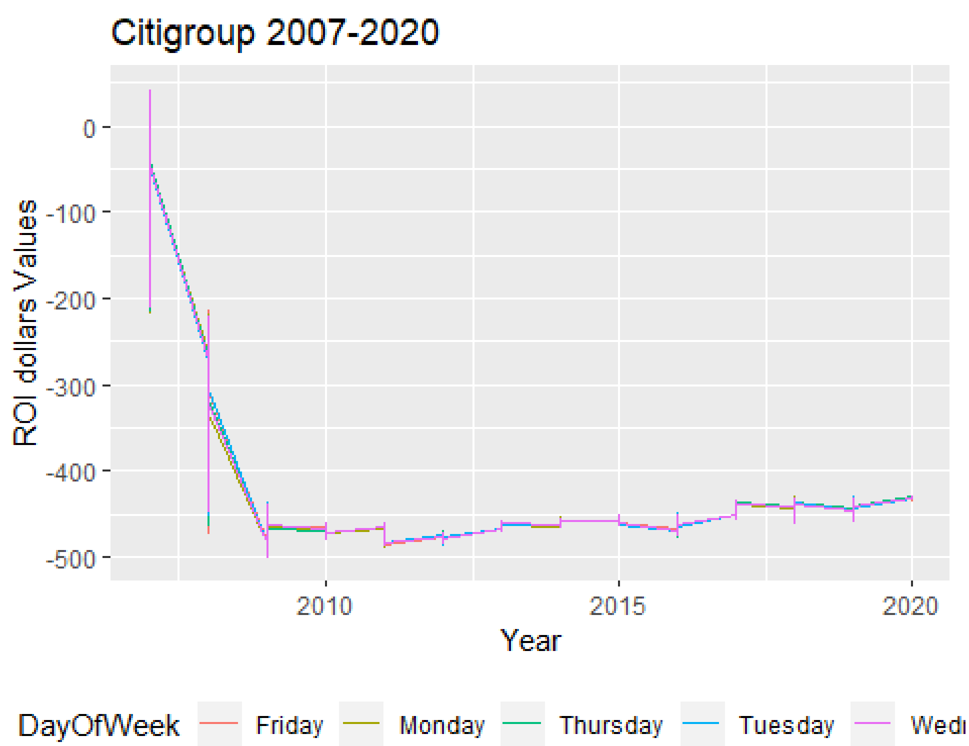
```
m <- StocksSTATS[order(StocksSTATS$Date,
decreasing=FALSE)[length(StocksSTATS$Date)], 107:159]
t <- as.data.frame(t(m))
colnames(t) <- row.names(m)
t$StockROI <- row.names(t)

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.

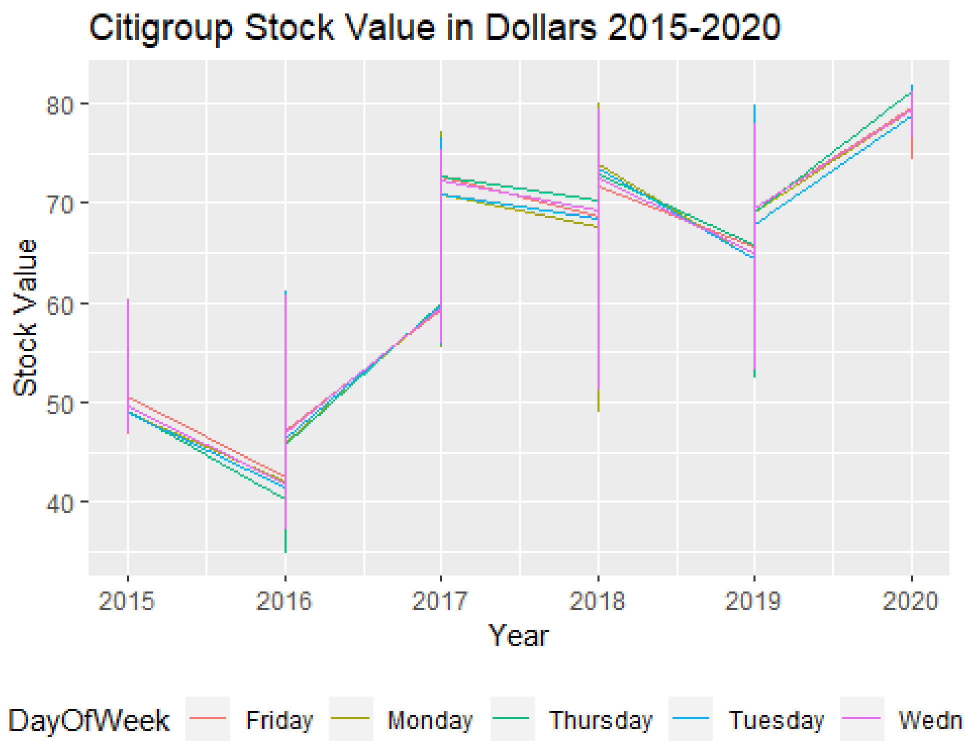
```
ggplot(data = C_stock, aes(x=Year, y=C_ROI_dollars,group=DayOfWeek)) +
  geom_line(aes(color=DayOfWeek))+
  scale_y_continuous()+
  scale_fill_brewer(palette="paired") +
  theme(legend.position="bottom")+
  ggtitle('Citigroup 2007-2020')+
  ylab('ROI dollars Values')

## 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.

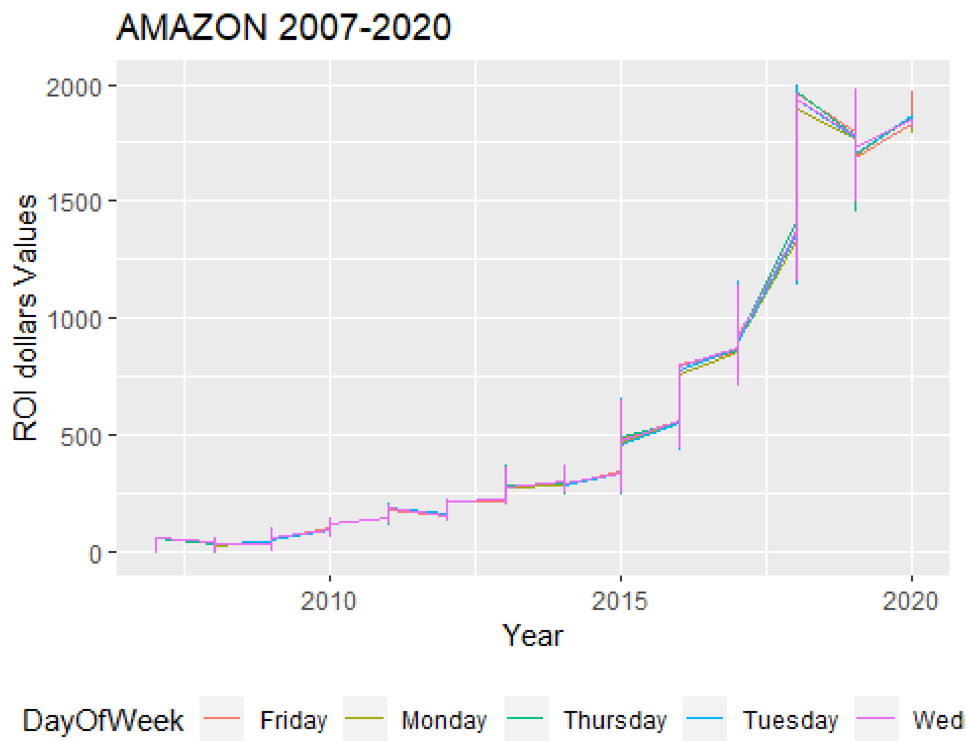
```
y2015plus <- subset(C_stock, C_stock$Year>2014)

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.

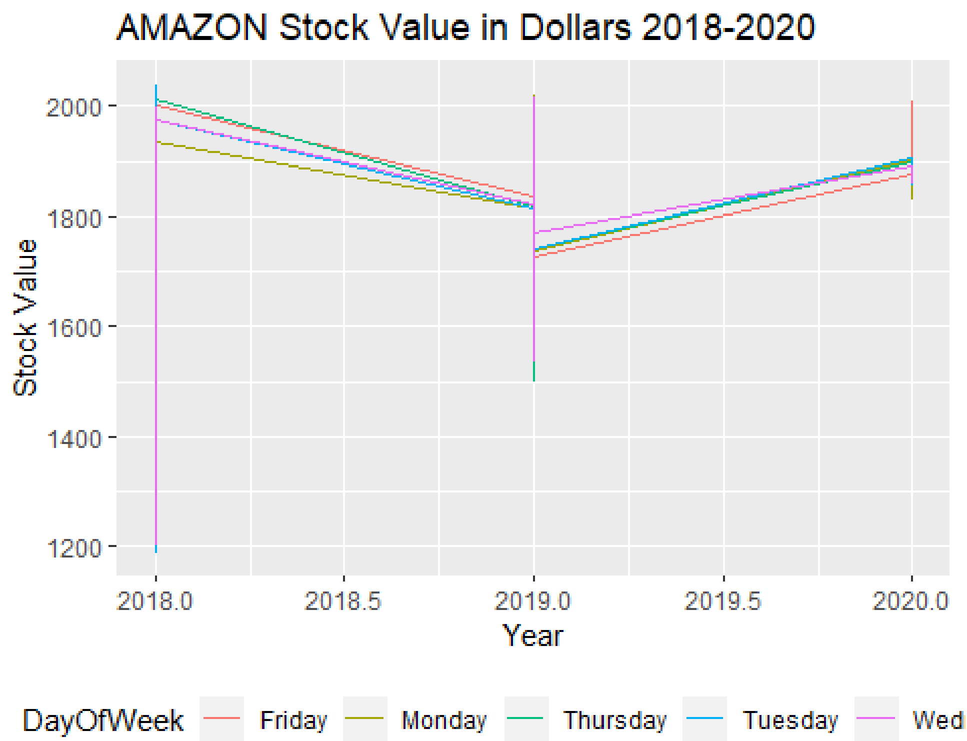
```
ggplot(data = amzn_stock, aes(x=Year, y=AMZN_ROI_dollars,group=DayOfWeek)) +
  geom_line(aes(color=DayOfWeek))+
  scale_y_continuous()+
  scale_fill_brewer(palette="paired") +
  theme(legend.position="bottom")+
  ggtitle('AMAZON 2007-2020')+
  ylab('ROI dollars Values')

## 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
```

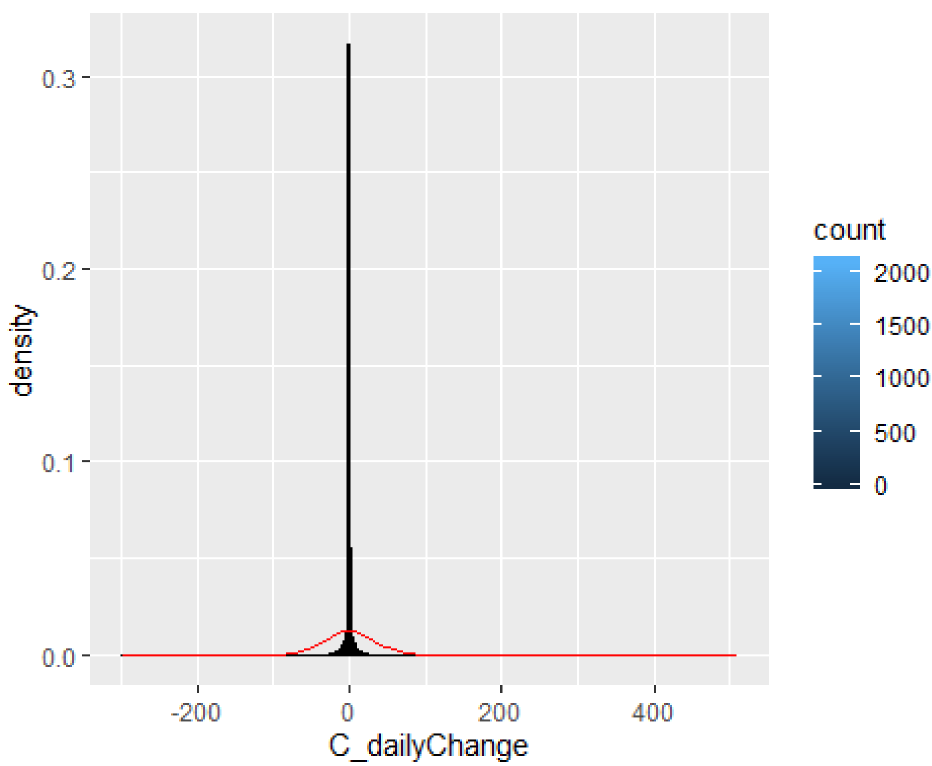
From the above information, Monday is the day of the week with the highest and lowest daily change, as well as the highest volume of trade. Tuesday is the day with the lowest volume of trade. The dates to pull an internet search of news articles about Citigroup to analyze public sentiment on Citi stock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

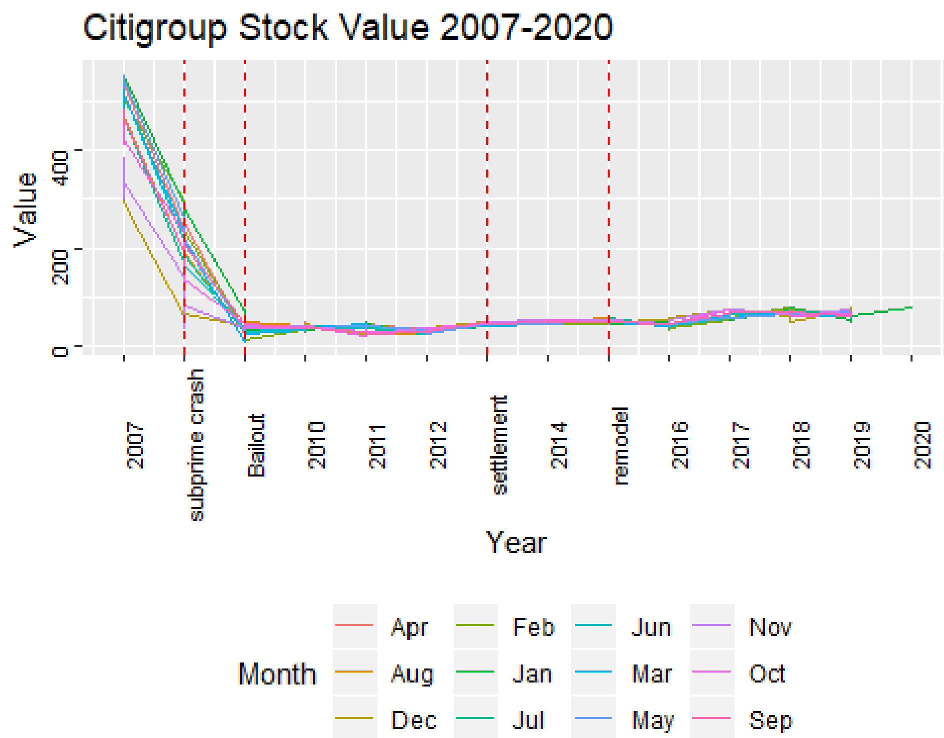
```
## 2015-12-28         28          0.27468
## 2008-06-02          2          0.06828
## 2007-04-021         2          0.25773
## 2007-08-01          1          0.05008
## 2007-12-03          3          0.05723
## 2007-02-01          1          0.04833
## 2007-01-03          3          0.05008
## 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 were about Citi's involvement in the sub-prime mortgage crisis of 2007-2008, and a bailout 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 for covering up bad mortgage loans in August 2012 and a person reported 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',2016,2017,2018,20
19,2020))+
  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?

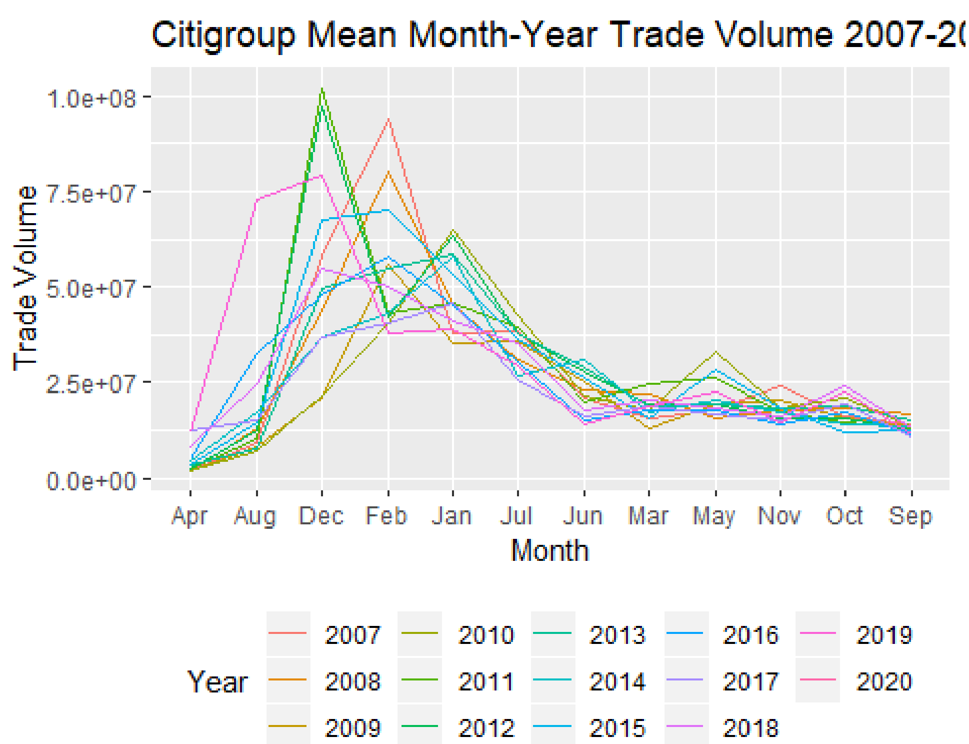
Well, I know that being a student, some people get their student loans around winter quarter in January and that many people expecting tax refunds get their refunds in February. We would have to see if there are any other assumptions about these months. But we would be able to ascertain if students receiving 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.

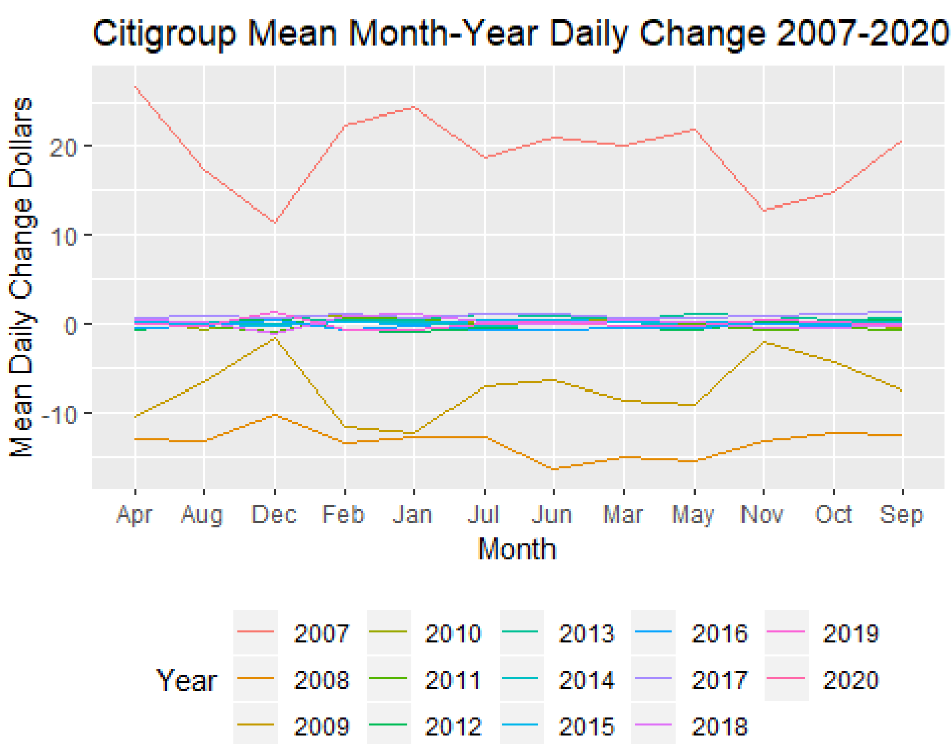
```
Citi_trades_monthYear$Month <- gsub('-[0-
9]{4}','',Citi_trades_monthYear$MonthYear)
Citi_trades_monthYear$Year <- gsub('[a-zA-z]{3}-
','',Citi_trades_monthYear$MonthYear)

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.

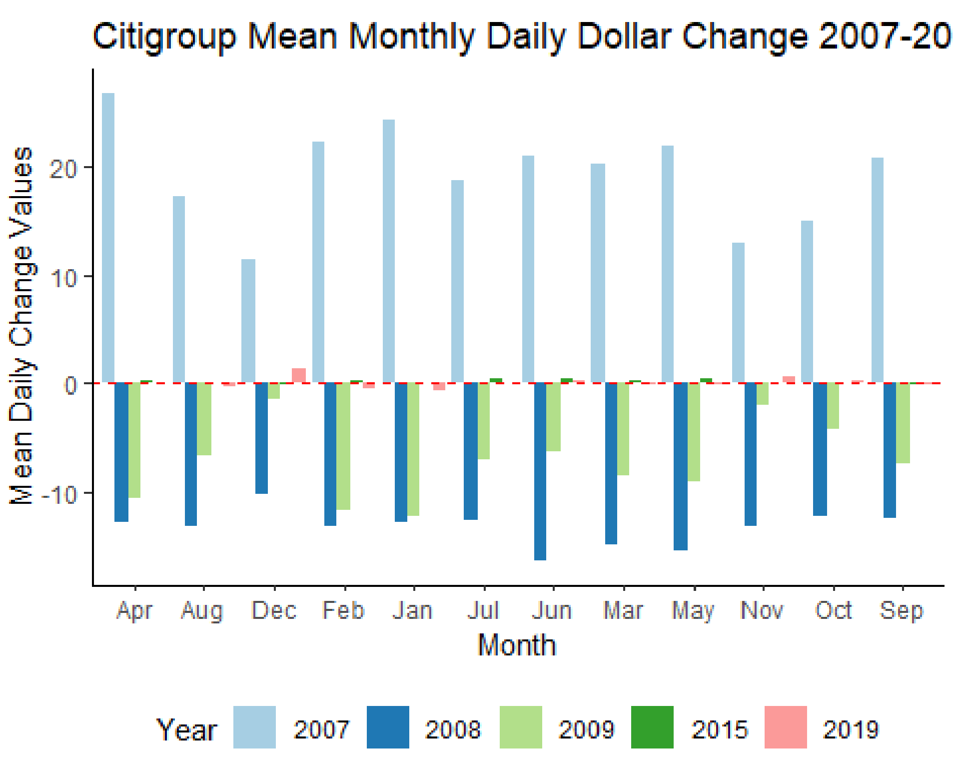
```
Citi_meanMonthly_dailyChange <- C_stock %>% group_by(MonthYear) %>%
  summarise_at(vars(as.vector(colnames(C_stock))[4]), mean)

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.

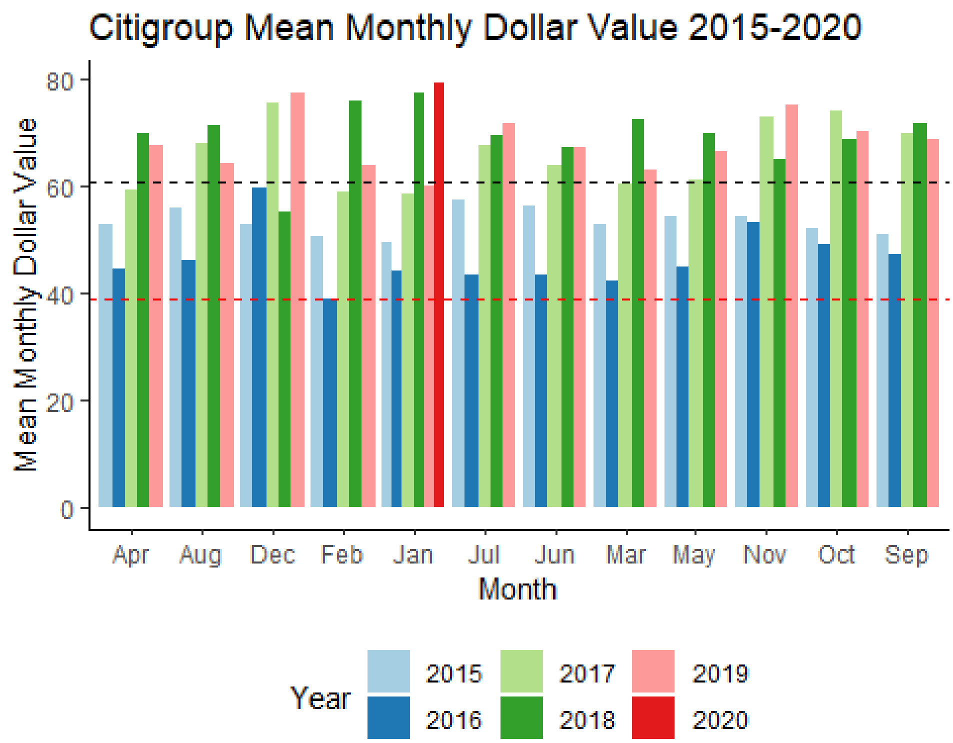
```
y4 <- subset(Citi_meanMonthly_dailyChange,
                            Citi_meanMonthly_dailyChange$Year==2008 |
                            Citi_meanMonthly_dailyChange$Year==2009 |
                            Citi_meanMonthly_dailyChange$Year==2007 |
                            Citi_meanMonthly_dailyChange$Year==2015 |
                            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.

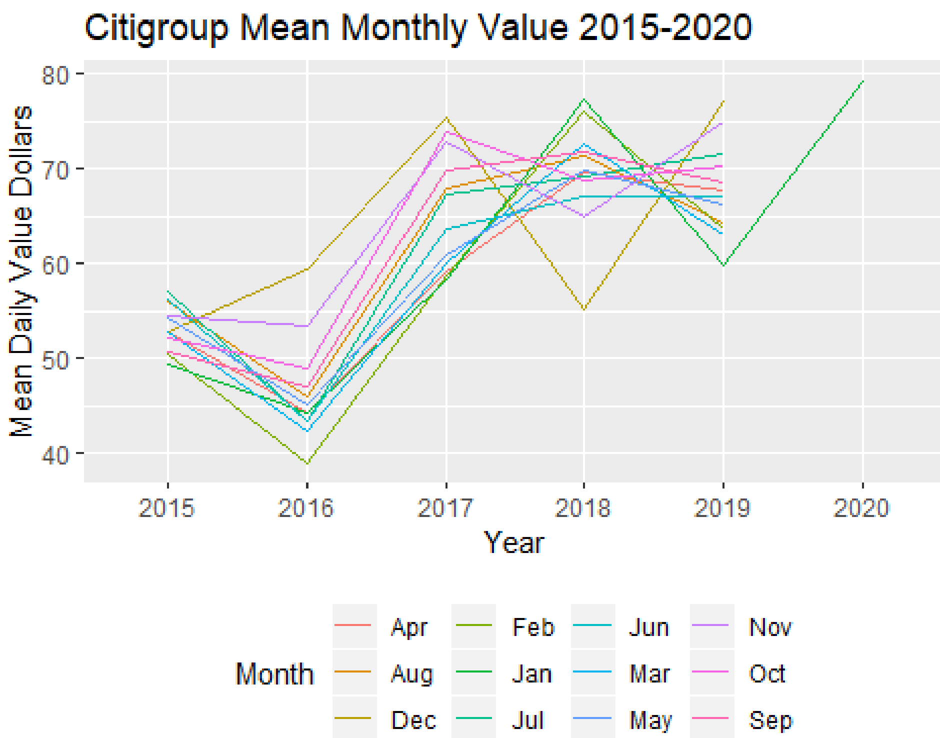
```
y4value <- subset(C_stock, C_stock$Year>2014)
y4valMY <- y4value %>% group_by(MonthYear) %>%
  summarise_at(vars(as.vector(colnames(y4value)[1])), mean)

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
```



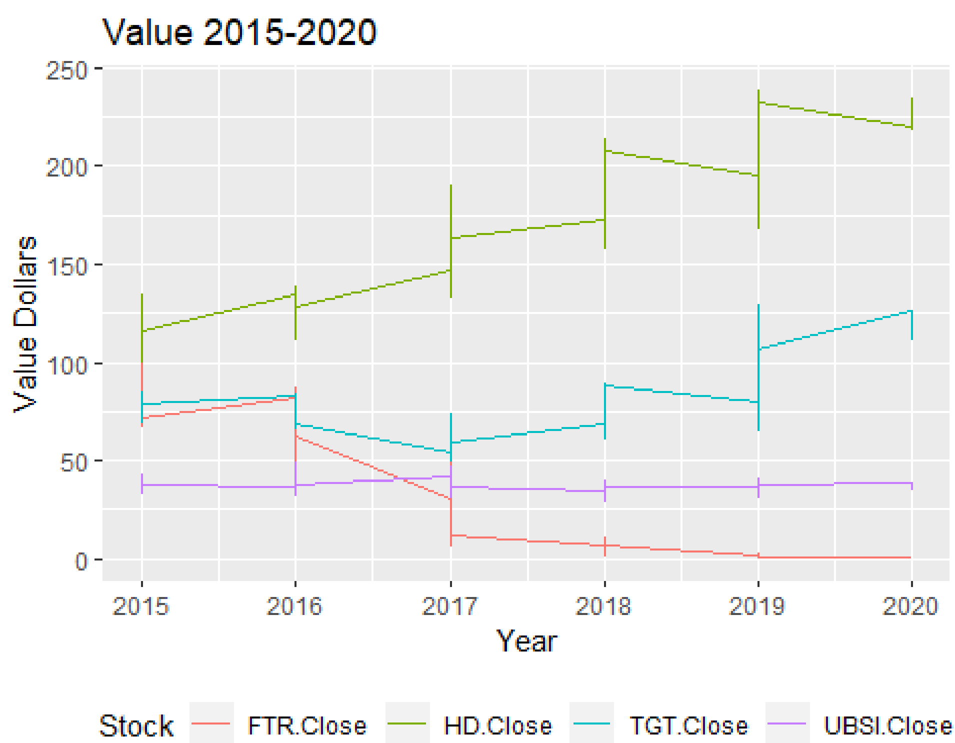
The above line chart of the mean monthly dollar value of the Citigroup stock show that all months move the same direction of decreasing in 2015, increasing in 2016, except for in 2017 and 2018 where 3-6 months decreased and 6-9 months increased monthly mean values. The span of 2019 through 2020 can't 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.

```r
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))+
```

```r
  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.

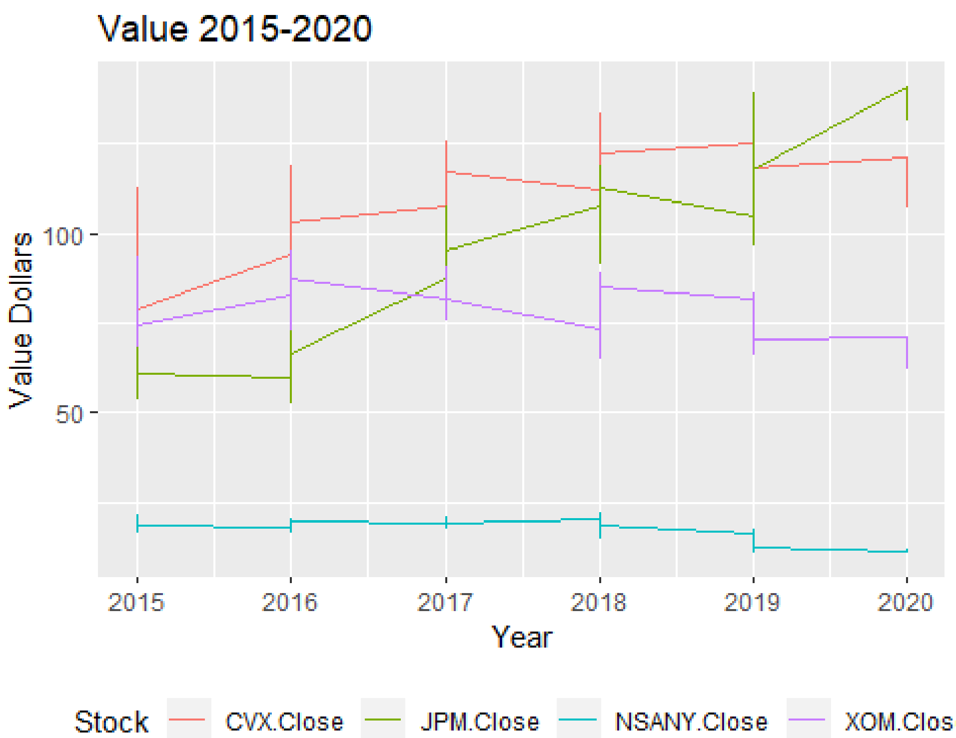
Also, increasing steadily is the blue line for TGT, which started at 75-80 in 2015 and is at 125 in 2020 in value.

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.

```r
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
```

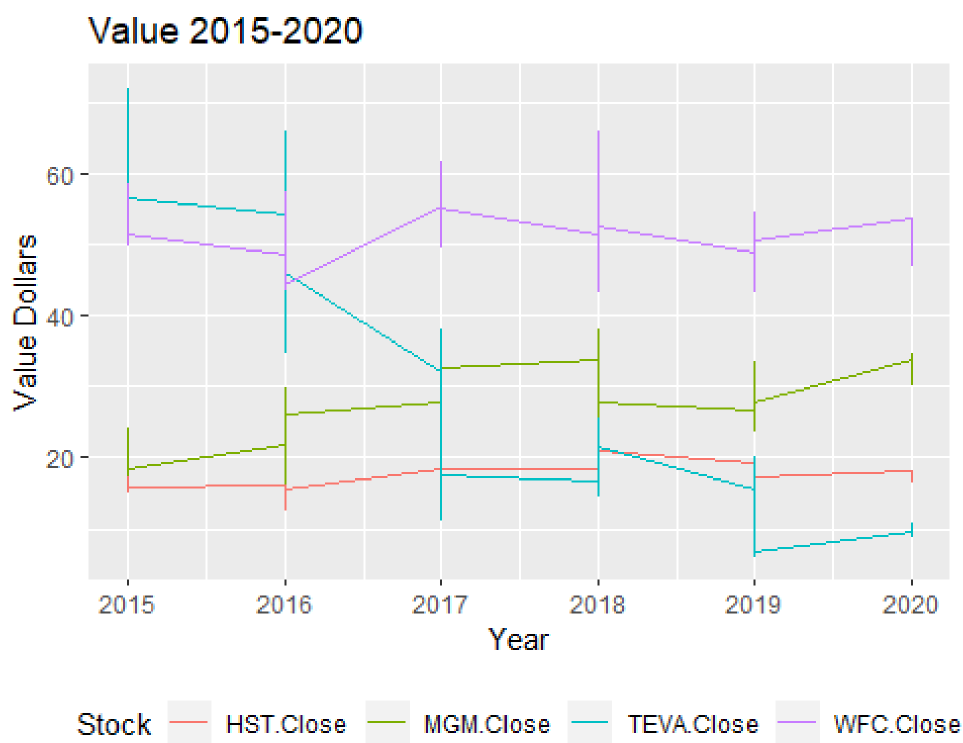


From the above subset of the next four stock in our 53 stocks, we can see that there are two stocks increasing significantly for JPM and CVX. We also note that the XOM and NSANY stocks have decreased over the last five years. ***

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.

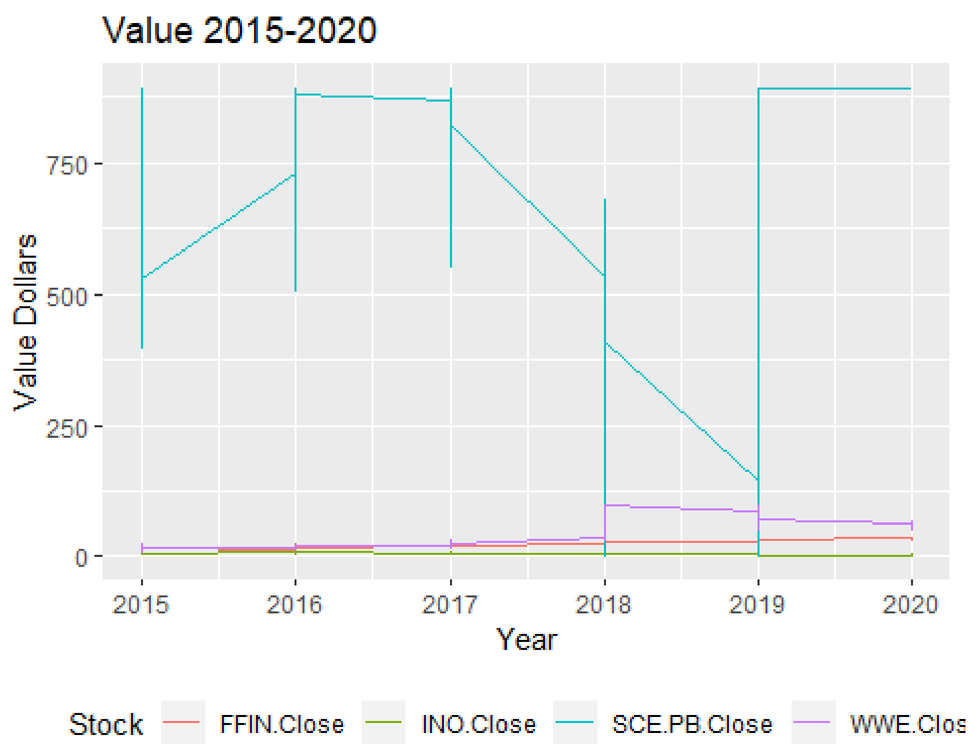
The MGM stock has increased significantly since 2005, and slight increases are shown for WFC and HST though not significantly. There is some cyclical movements in the WFC with 2016 giving a steady increase all year, then declining 2017-2019, and ending with a steady increase in 2019.

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
```



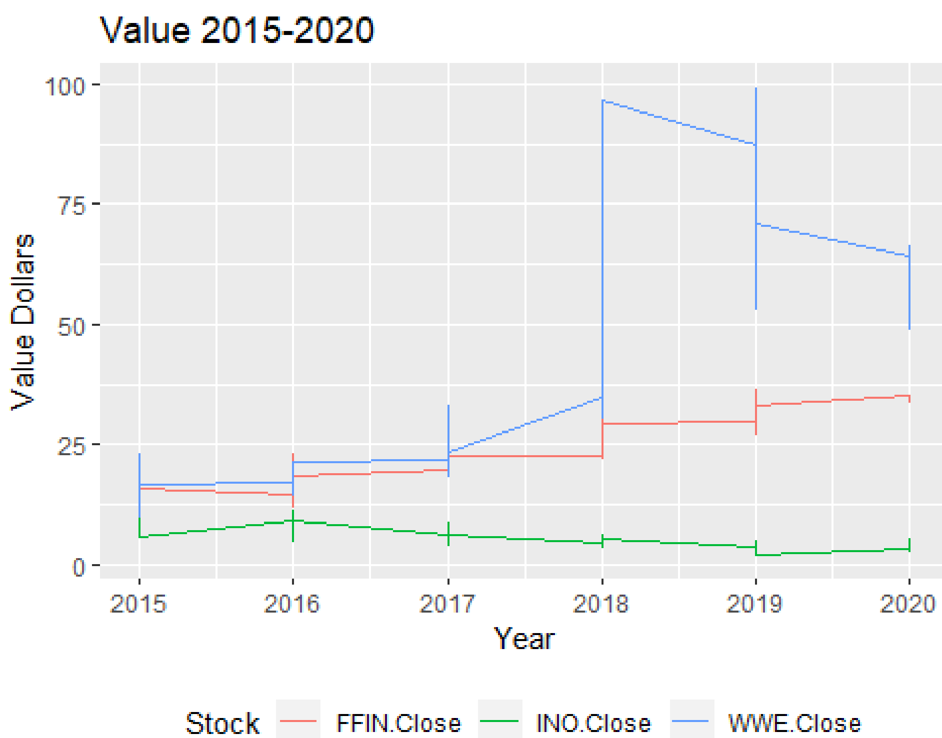
The above line chart shows that SCE.PB is on its own scale that outweighs the scale of the other smaller valued stocks, there is also volatility and cyclical movements in SCE.PB which makes it a good choice to further analyze with timelines of web article events that could have triggered these changes 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.

```r
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
```



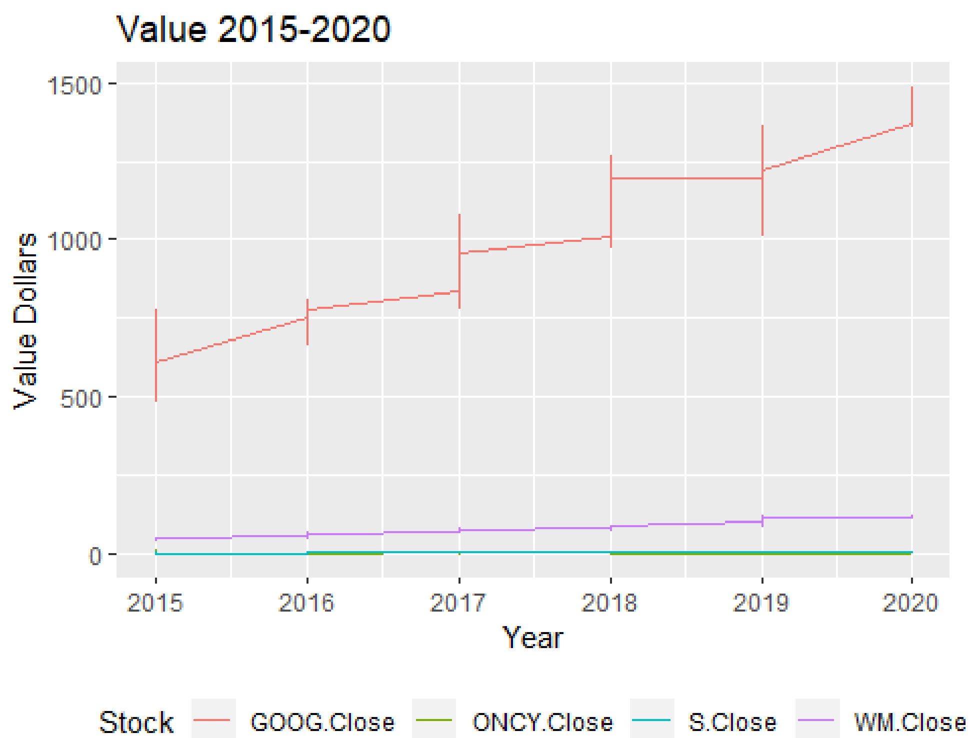
From the above line chart, we see that WWE had a huge jump in 2018 of an increase from the 40 range to the 90 range but then decreased during 2018 and 2019 to a price still much higher at the 60 range than its starting value in 2015 of the 20 range.

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
```

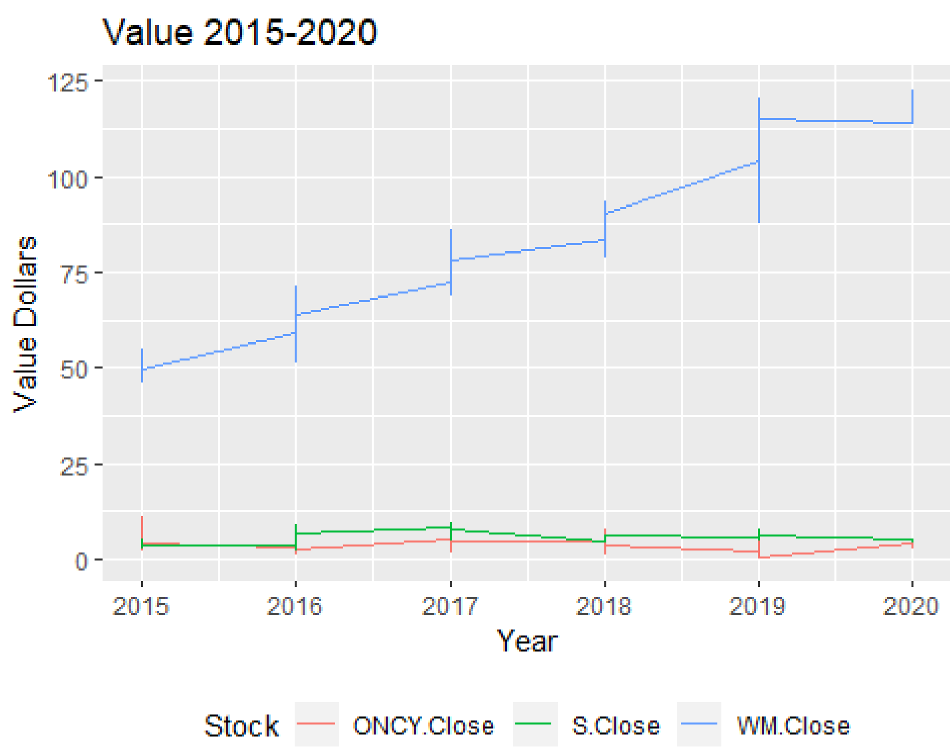


In the above subset of stocks, Google out scales the other three stocks and shows that it has been increasing steadily every year, except 2018 where it is almost the same price all year.

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
```



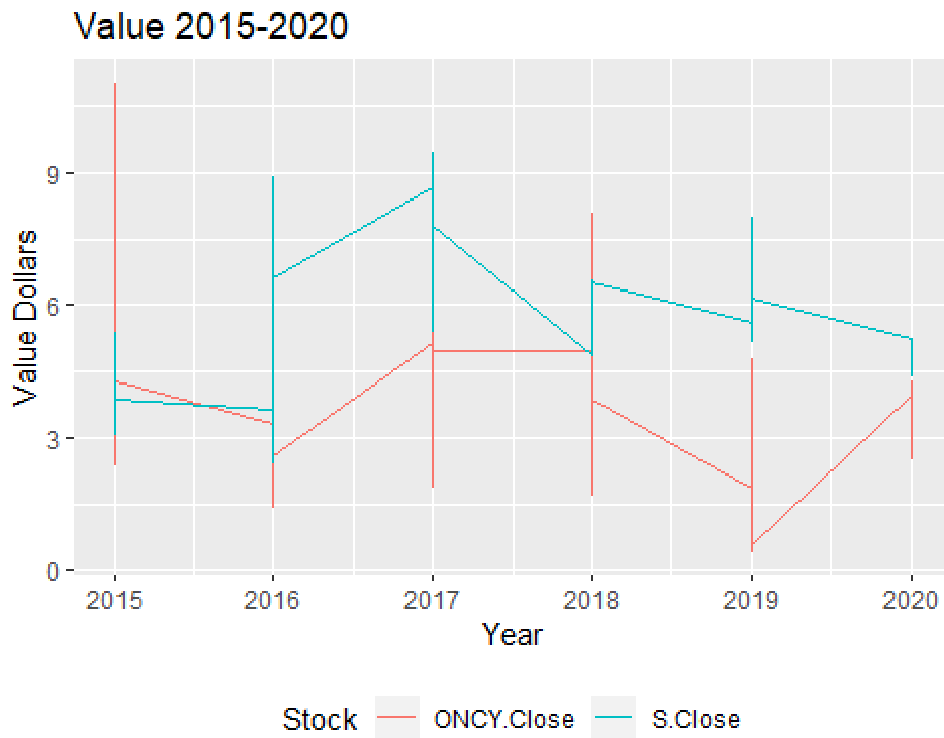
The line chart above shows that WM has increased significantly every year since 2015, with a slight decrease in 2019, but overall has increased from the 50 range in 2015 to the 113 range in 2020.

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')
```

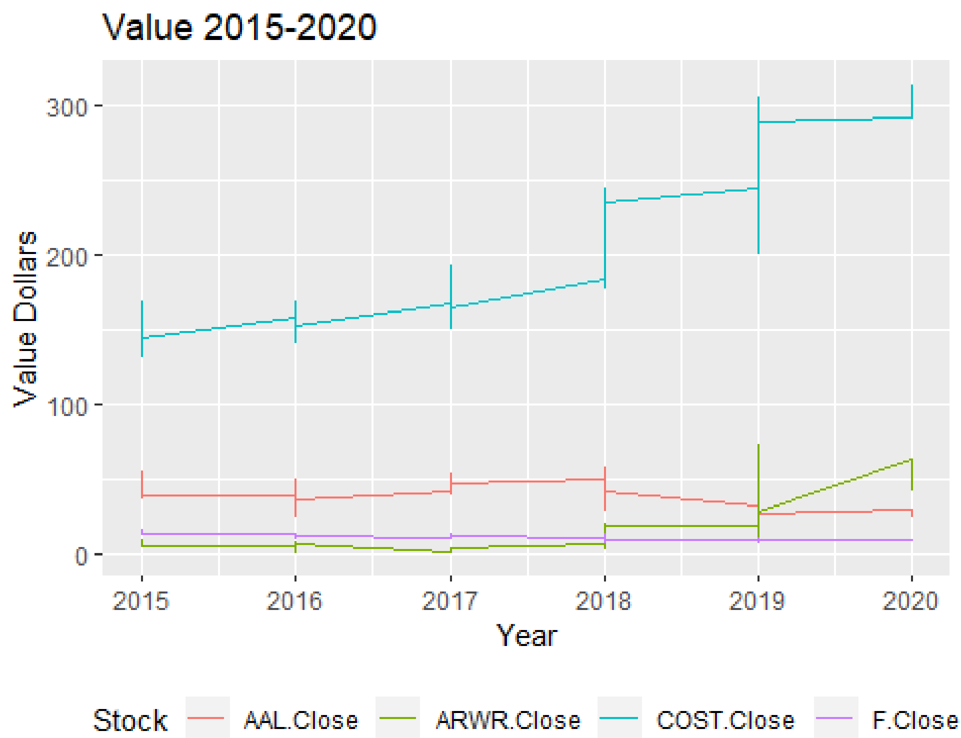
```
## Warning in pal_name(palette, type): Unknown palette paired
```



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.

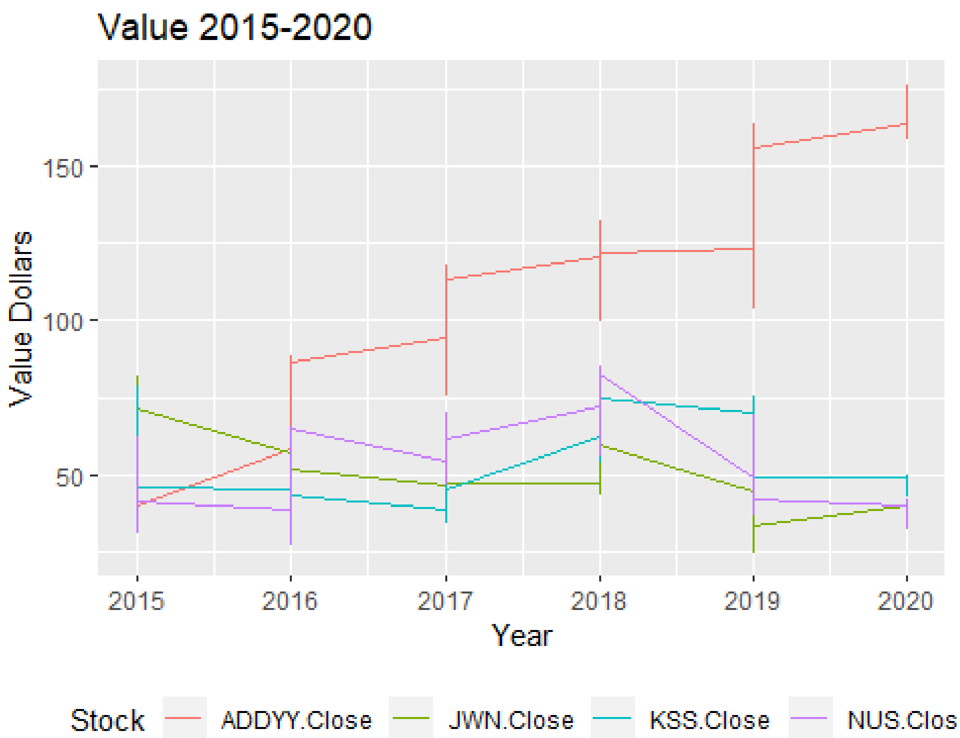
```
sub1 <- Value2[,c(21:24,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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.

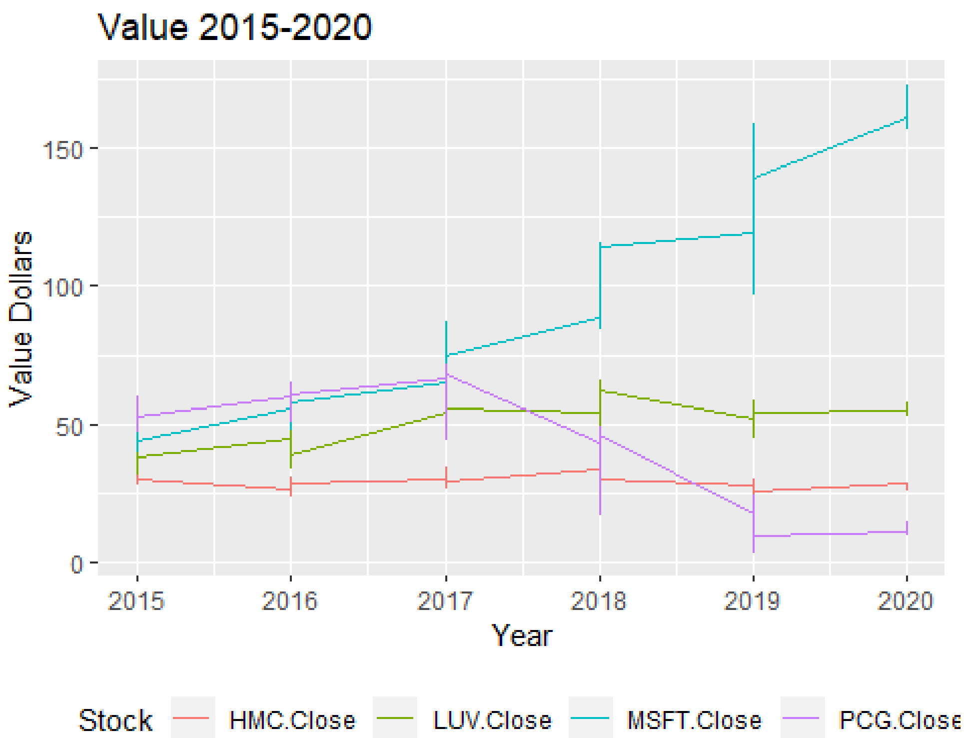
```
sub1 <- Value2[,c(25:28,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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.

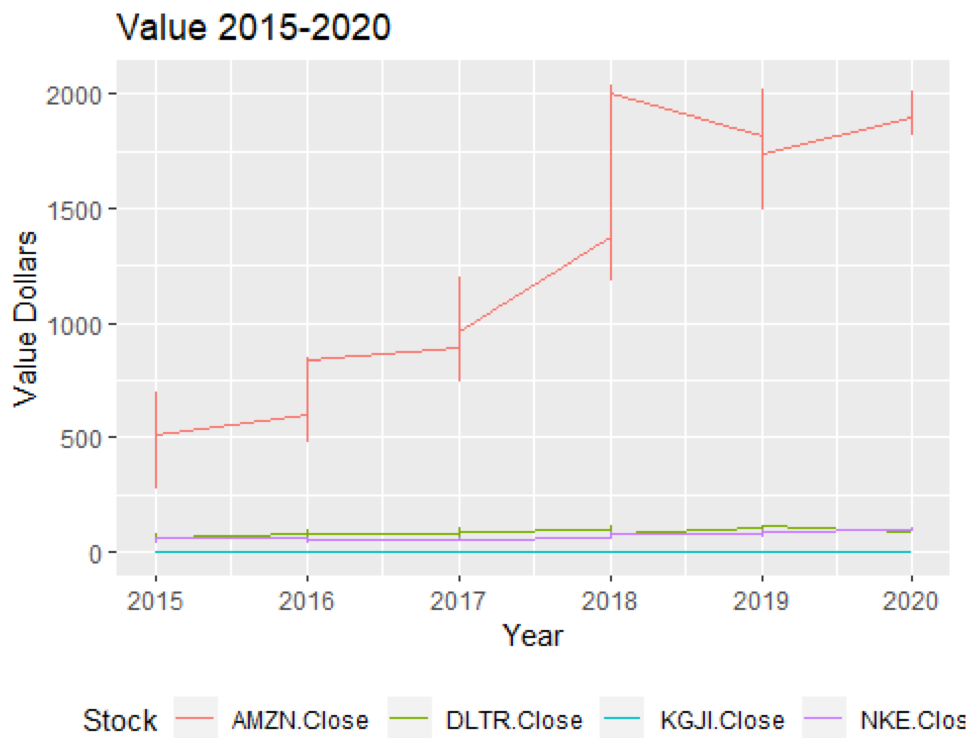
```
sub1 <- Value2[,c(29:32,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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.

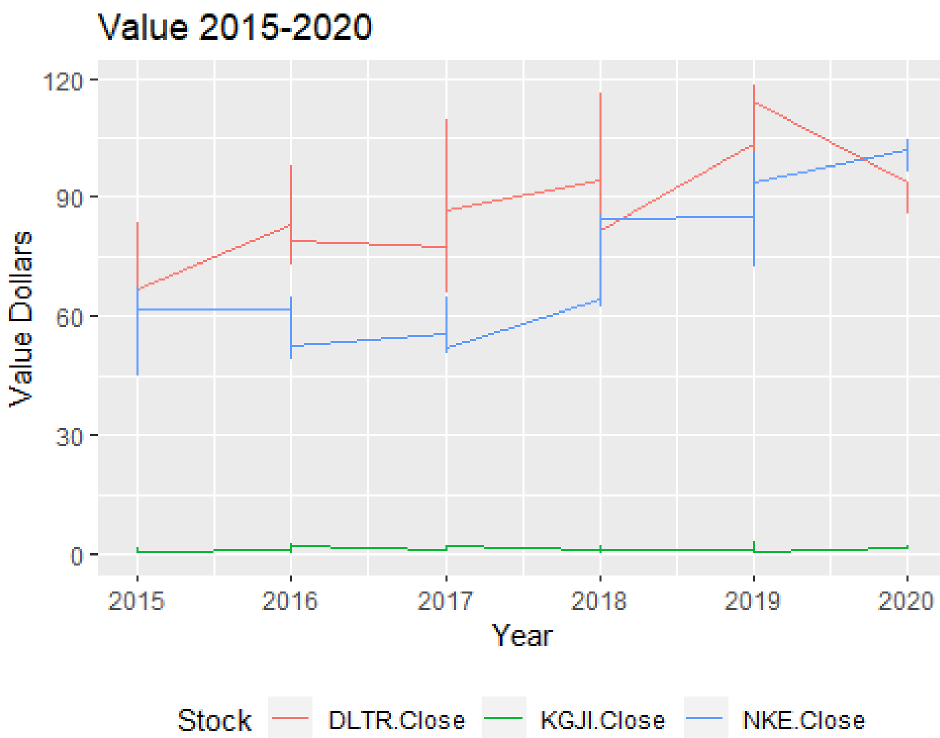
```
sub1 <- Value2[,c(33:36,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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.

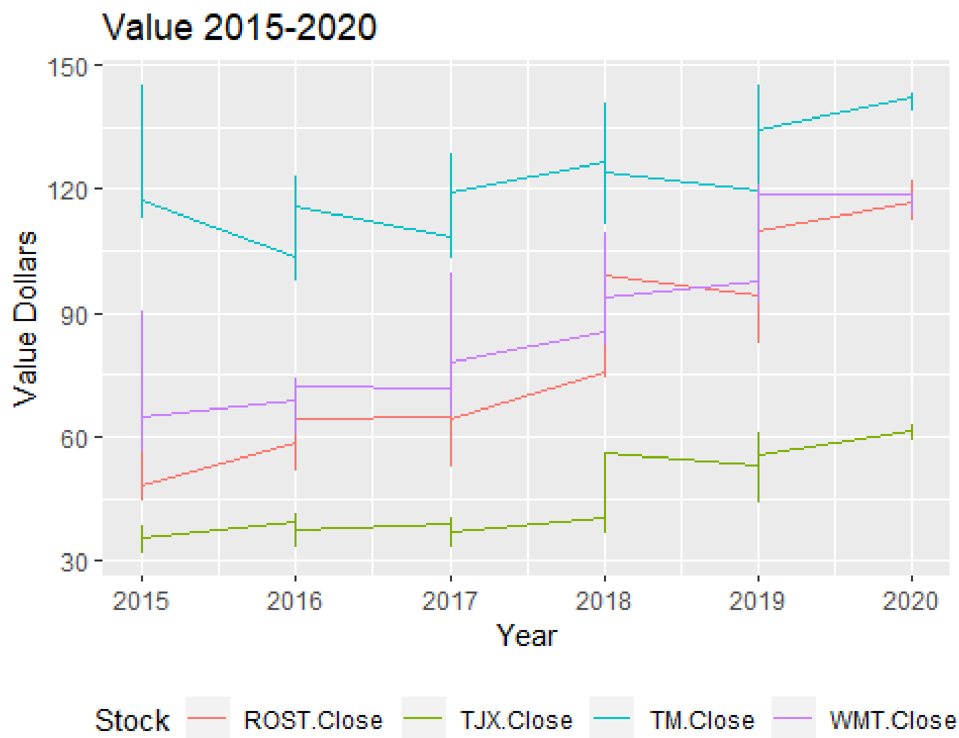
```
sub1 <- Value2[,c(33:35,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:3)

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.

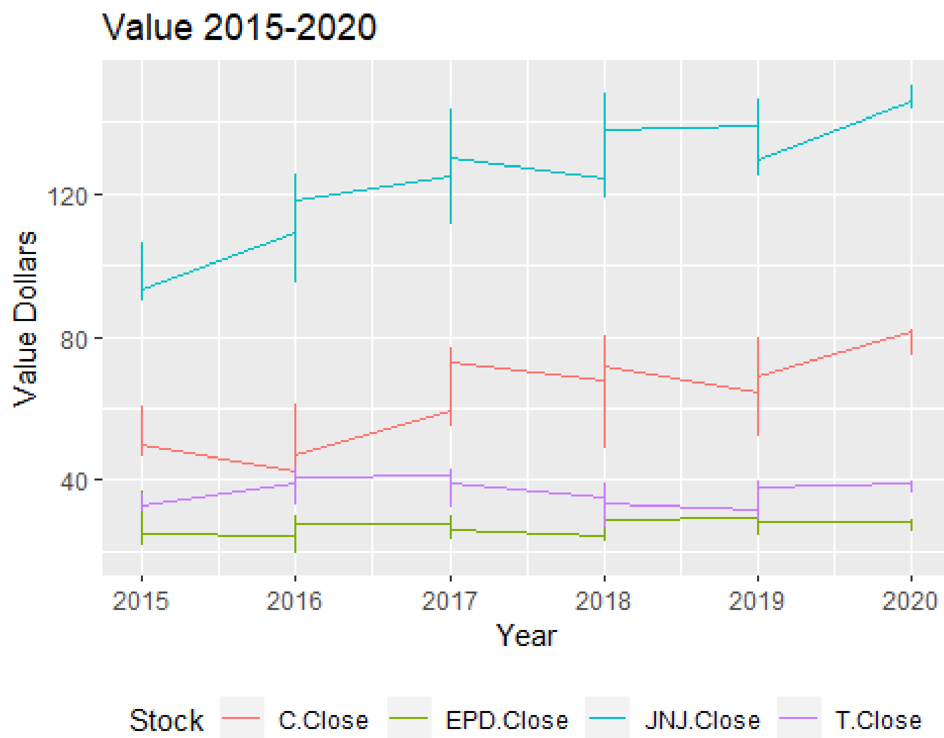
```
sub1 <- Value2[,c(37:40,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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.

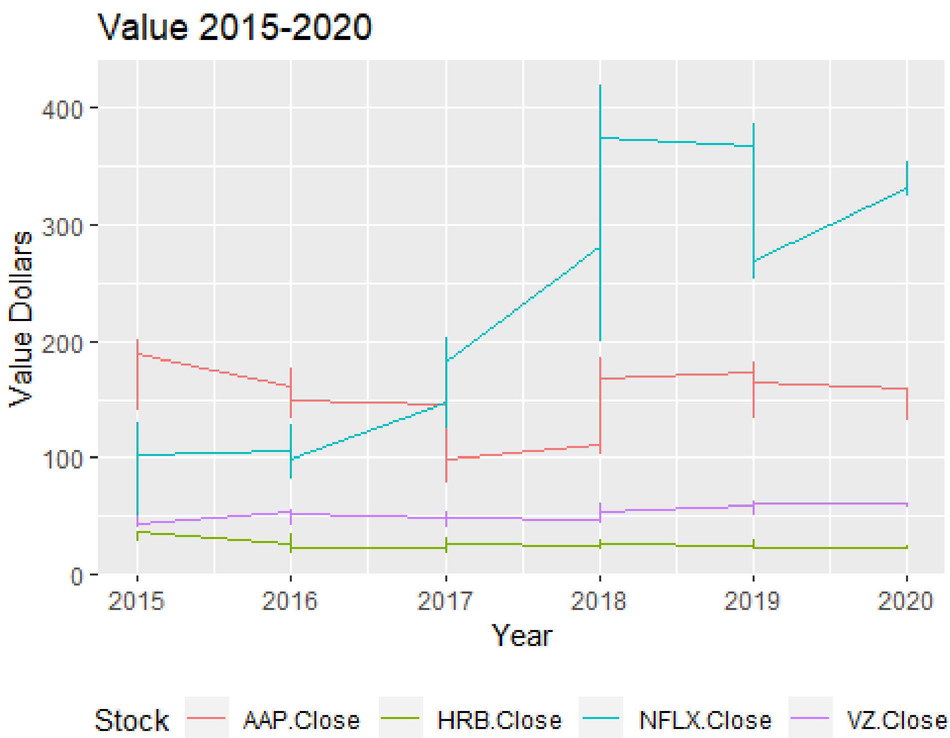
```
sub1 <- Value2[,c(41:44,115)]
sub1tidy <- gather(sub1, 'Stock','Value',1:4)

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
```

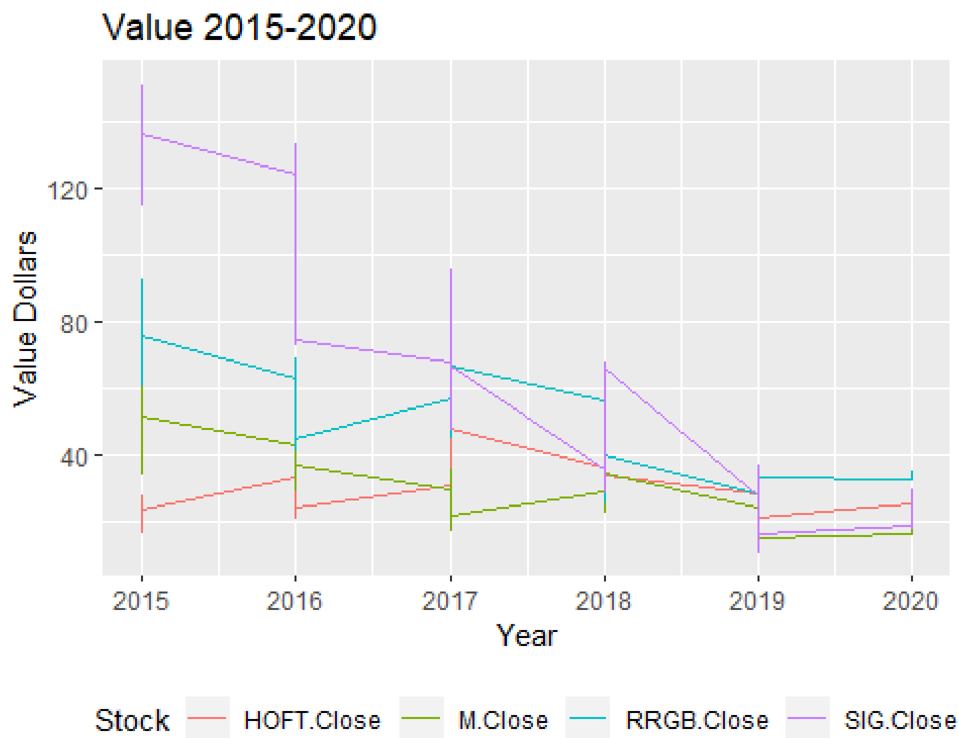


The above line chart shows that NFLX increased significantly while HRB and AAP saw losses over the last five years. VZ stock saw a slight increase in value over the last five years. In 2017 Netflix saw a huge inrease, while in 2018 it stayed somewhat stagnant with a sharp drop in value at the start of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

```r
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.

```r
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$Direction5yr10PercentChange=='',0,StockName
s_STATS_2015_2020$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
```

```r
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)
```

```r
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 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_me
```

```r
dian)),
         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
```

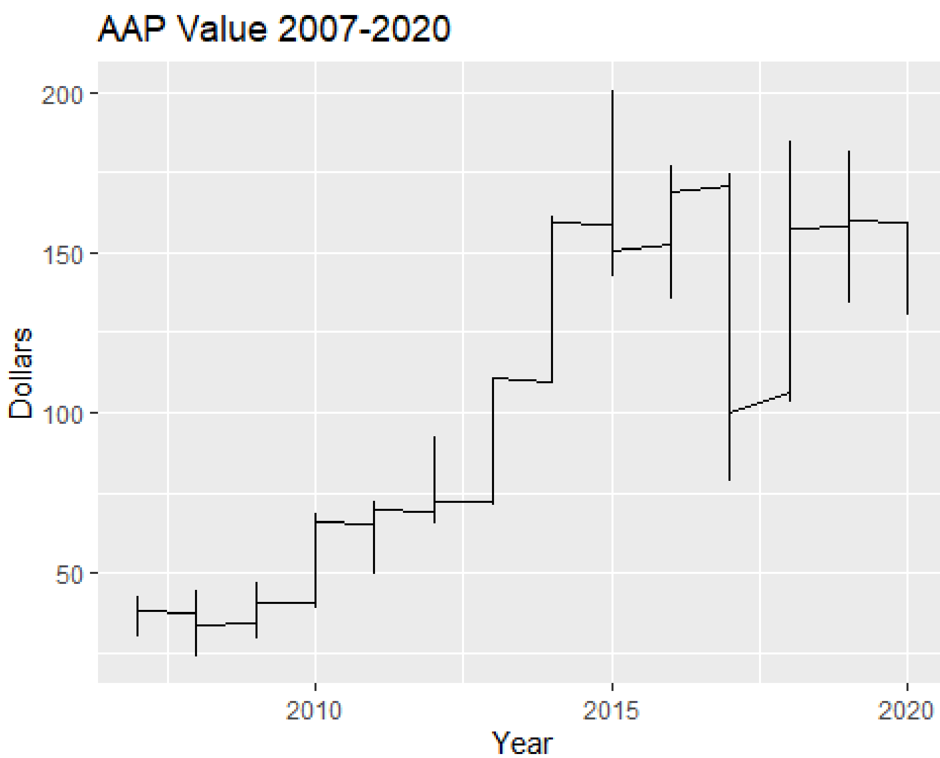
```
##    DOW_down_median
## 1              AAP
## 2            ADDYY
## 3             AMZN
## 4             COST
## 5             DLTR
## 6             FFIN
## 7             GOOG
## 8               HD
## 9              JNJ
## 10            NFLX
## 11             NKE
## 12             PCG
## 13            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)
```

```r
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
```



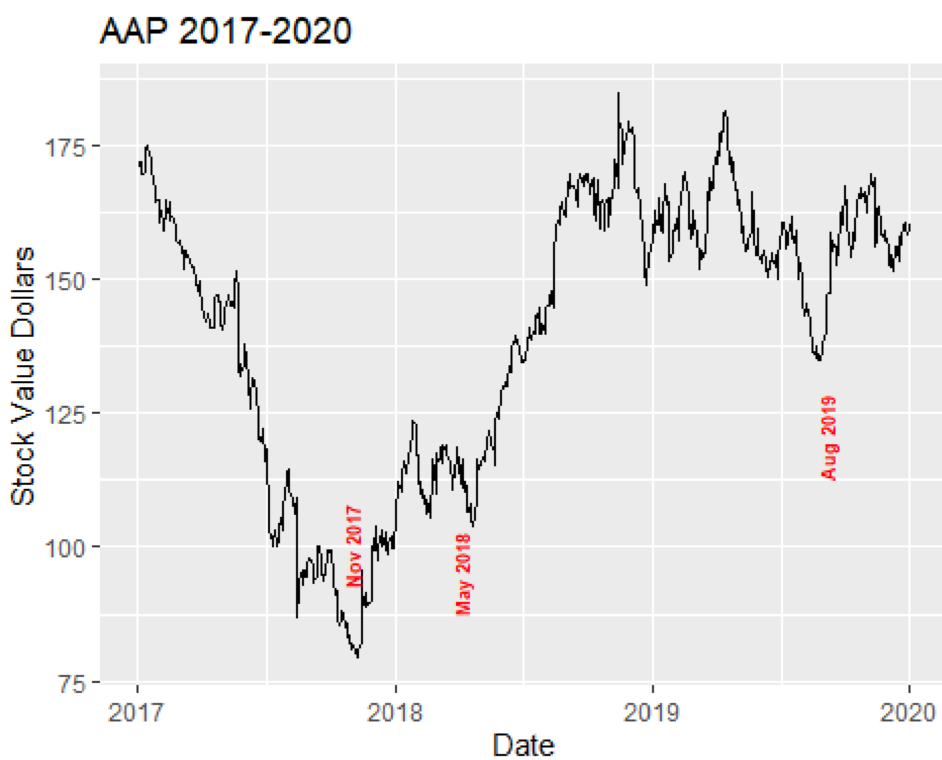
The above shows that AAP had a huge drop in 2017 at a local minimum, the other minimum is in 2008 and is the global minimum for this stock. The next chart shows the years 2017-2020 to zoom in on this loss.

```r
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.

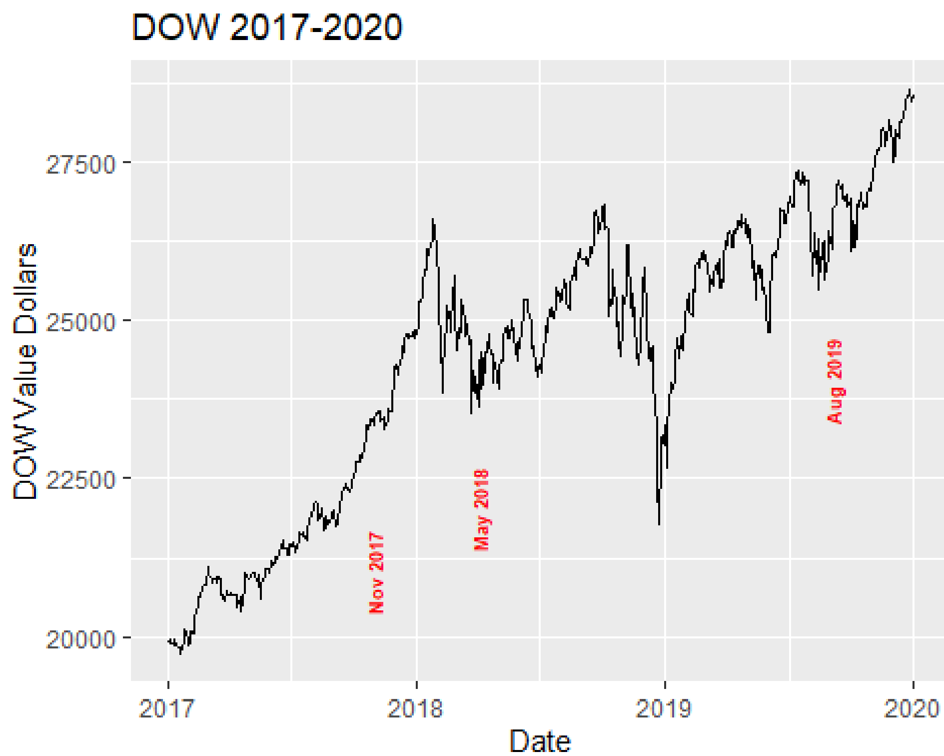
```
summary(aap1)
```

```
##       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.

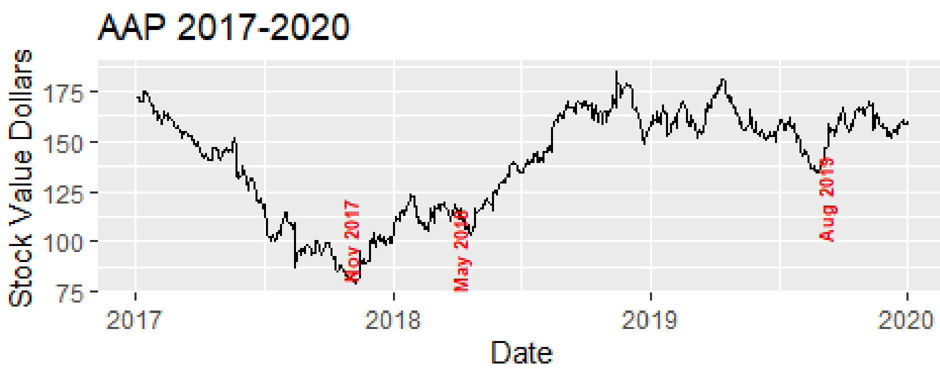
```
annotation2 <- data.frame(
   x = c(as.Date('2017-11-01'),as.Date('2018-04-04'),as.Date('2019-09-04')),
   y = c(21000,22000,24000),
   label = c("Nov 2017", "May 2018","Aug 2019"))

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
```

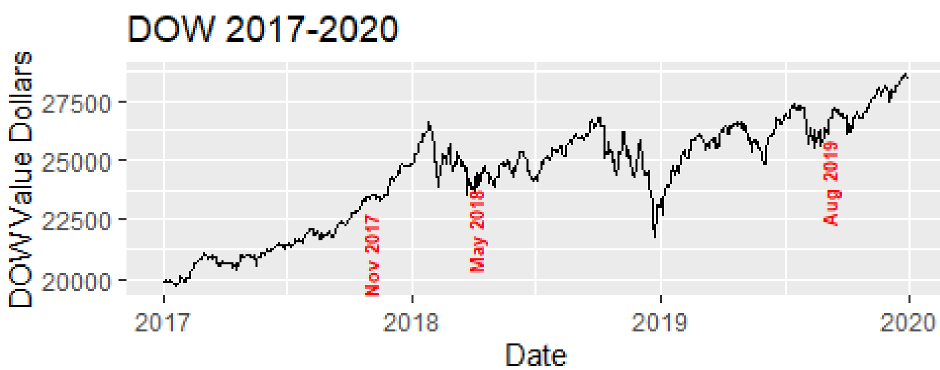


The above chart shows the increasing values of the DOW the same years as the AAP saw a decreasing year in 2017 to its local minimum in the fourth quarter of 2017, and then an increase until 2019 when it stabilized around the same value till 2020. But the DOW increased the time that AAP was decreasing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





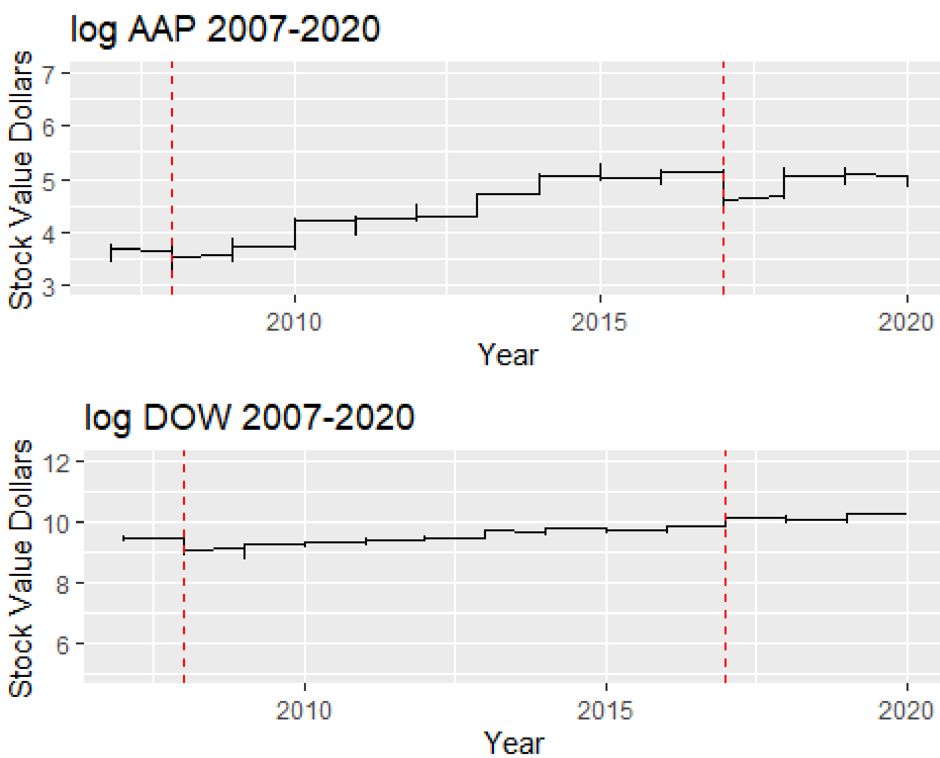
Lets see what happened in 2017 for AAP and in 2018 to cause it to decrease then increase respectively… just declining sales, plans to expand and open more stores, and public outcry on the sales declines of Advance Auto Parts. So, this could mean that because the DOW was doing great and increasing, investors 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, 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
```

```r
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
```

```r
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.

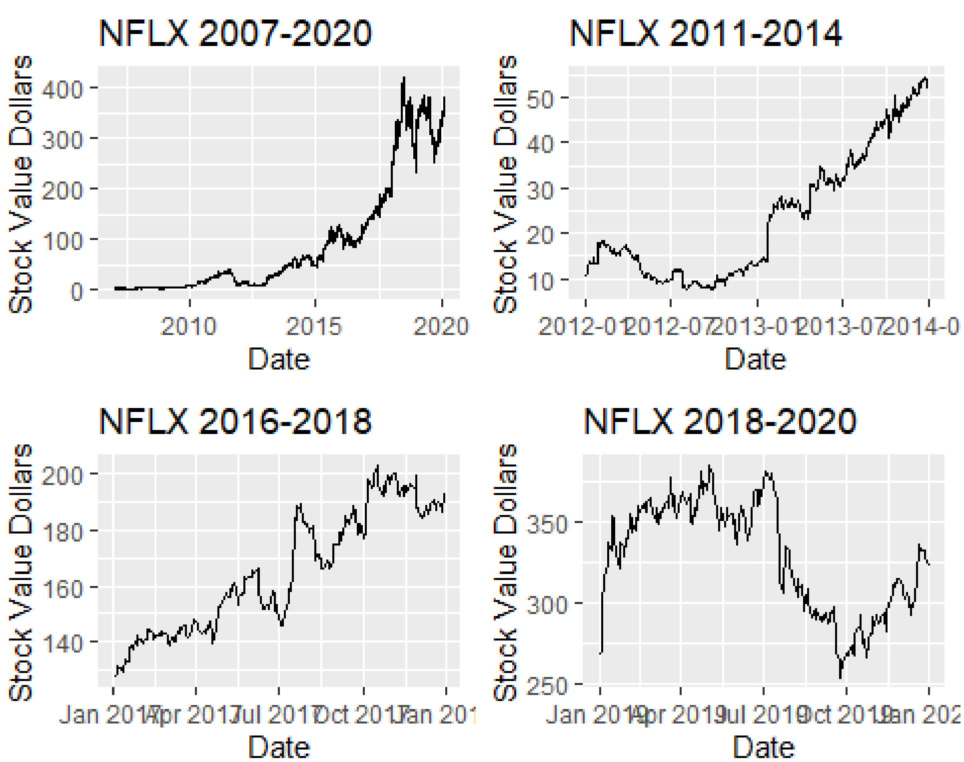
```r
dROI17 <- Close17_dow %>% group_by(DOW_down_median) %>%
  summarise_at(vars(stockValue), mean)
colnames(dROI17)[2] <- 'avgStockValue'
```

```r
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.

```r
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
```

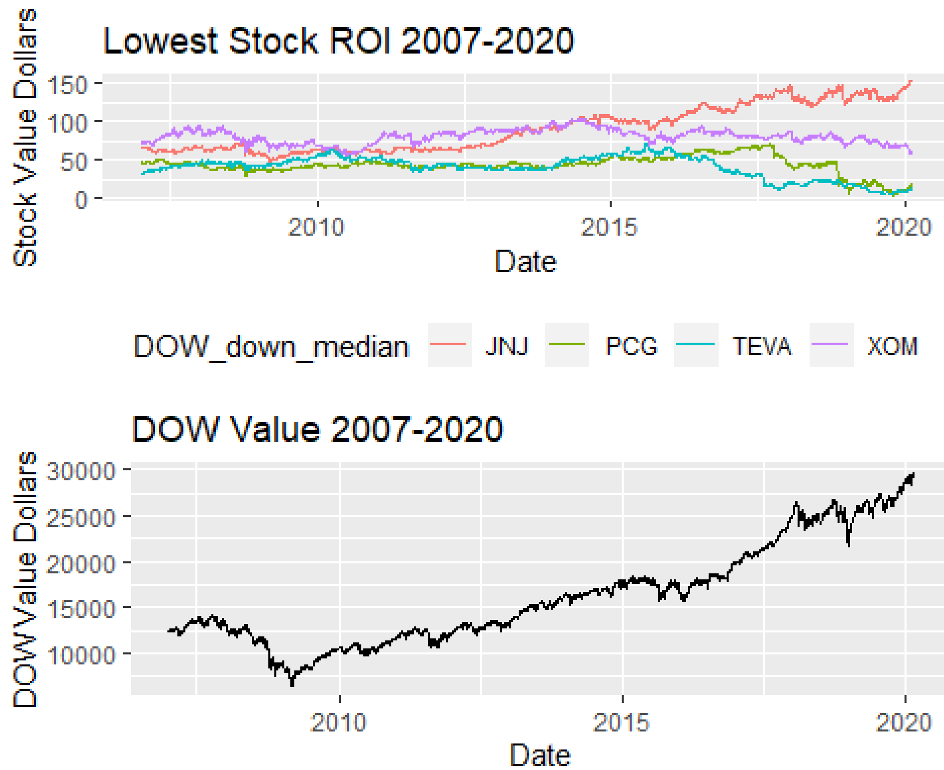
```
##   4 COST            Yes
##   5 DLTR            Yes
##   6 FFIN            Yes
##   7 GOOG            Yes
##   8 HD              Yes
##   9 JNJ             No
##  10 NFLX            Yes
##  11 NKE             Yes
##  12 PCG             No
##  13 ROST            Yes
##  14 TEVA            No
##  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) 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)
```

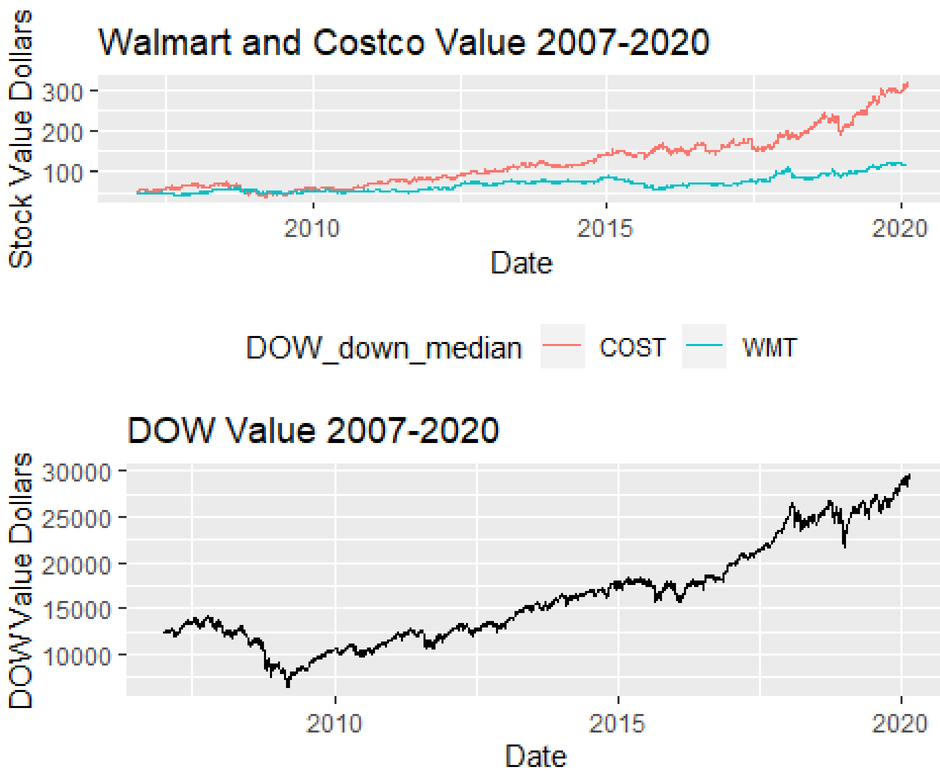


From the above charts, It looks like Johnson & Johnson has started to move upward with the DOW starting around 2013. The TEVA stock seems to be negatively correlated with the DOW and Exxon Mobil was positively correlated with the DOW from 2007 to about 2015, then started moving in the opposite direction 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)
```

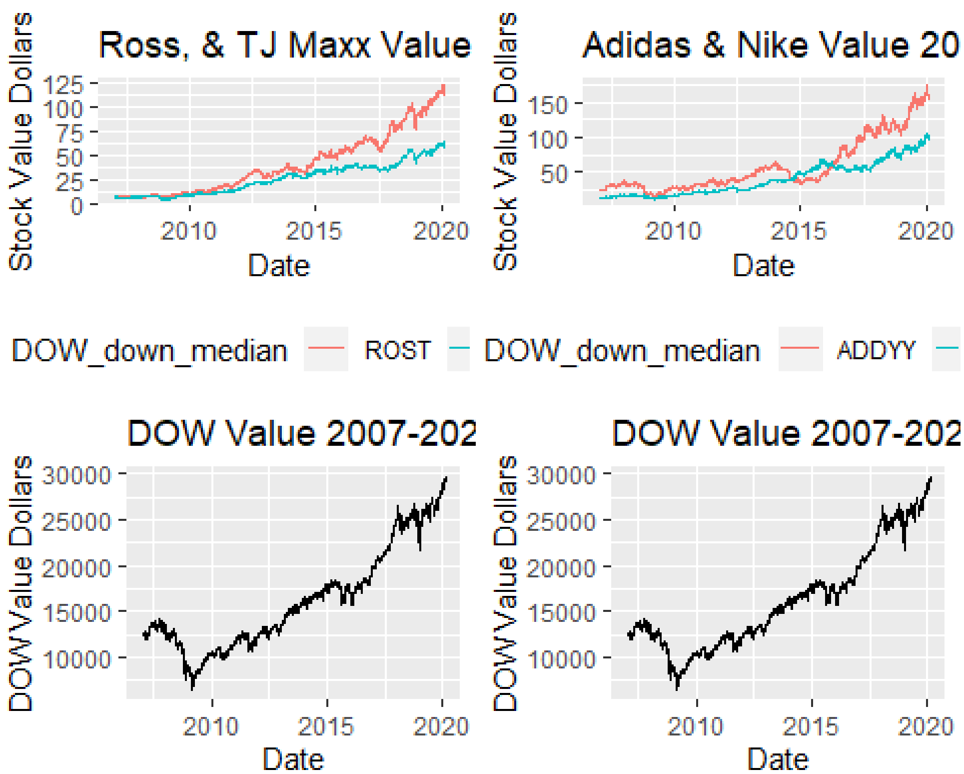


It is interesting to note that Costco and the DOW seem to be identical curves for the direction they move while Walmart seems also be increasing when the DOW does but at a much lower rate over time.

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)
```

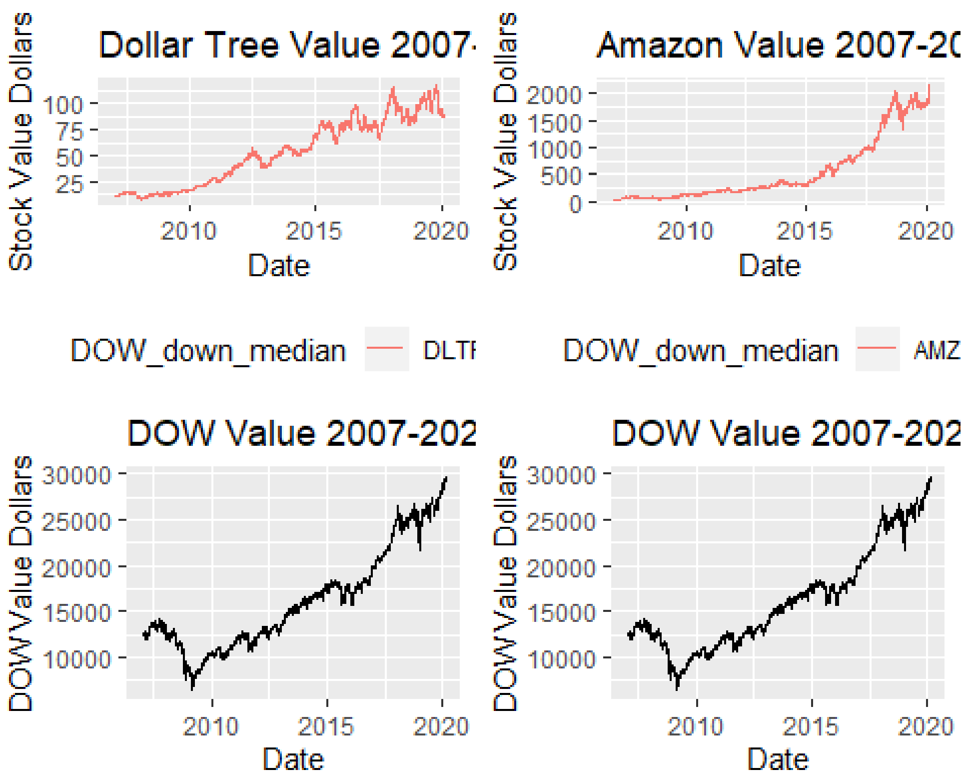


The DOW is plotted below each of the two plots of either Ross & TJ Maxx, or Adidas & Nike. Both Ross and Adidas did better than TJ Maxx and Nike. Except, that Nike did do better than Adidas between 2015 and 2016.

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

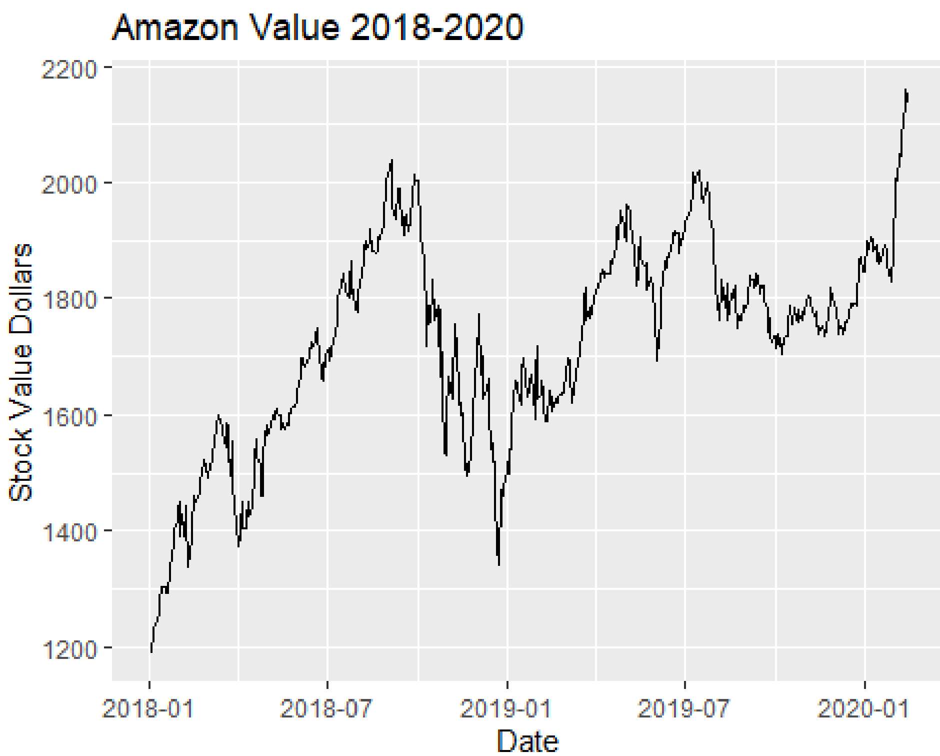
```
DLRv <- subset(Close17_dow, Close17_dow$DOW_down_median == 'DLTR' )
AMZNv <- subset(Close17_dow, Close17_dow$DOW_down_median == 'AMZN' )


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.

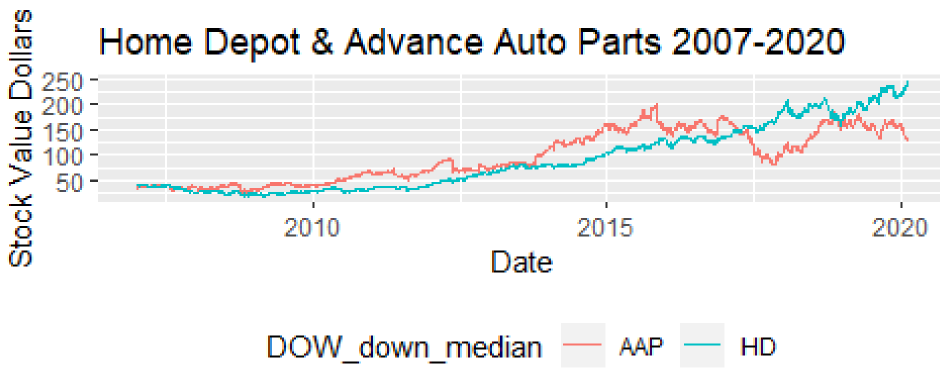
```
AMZNv2 <- subset(Close17_dow, Close17_dow$Year > 2017 &
                   Close17_dow$DOW_down_median == 'AMZN')

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
```

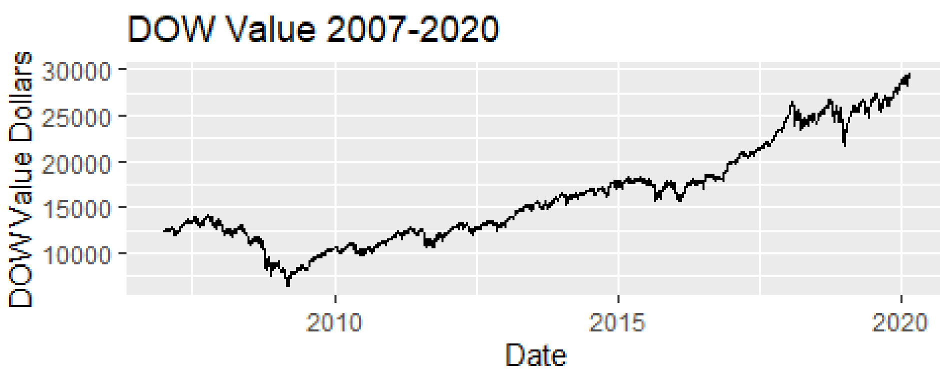


We see a drop in value of Amazon after September 2018 until about January 2019 when it increases until just before the end of the 2nd quarter in 2019 then it drops in the 1st quarter staying low before increasing to a global maximum in January 2020.

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)
```



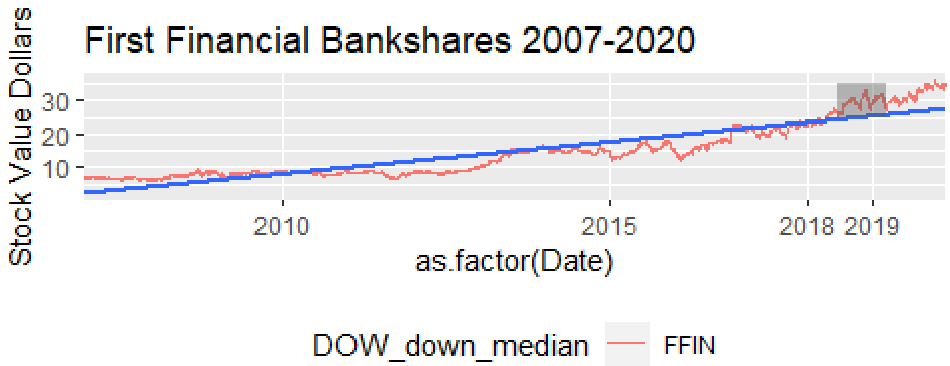


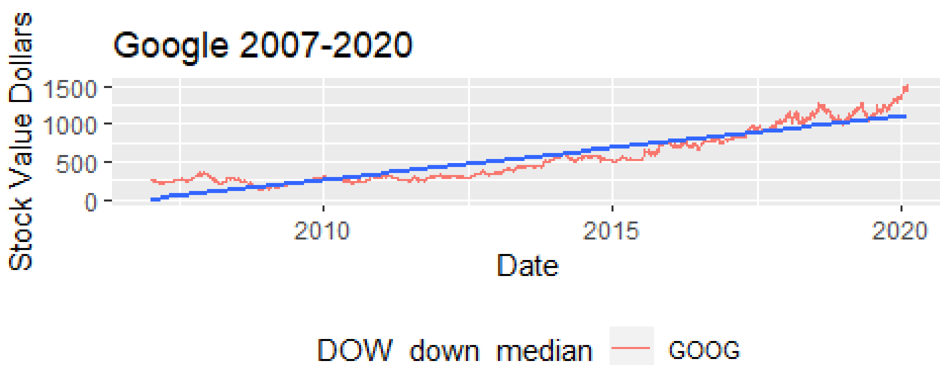
The charts above show that Home Depot and the DOW move together by increasing the time span from 2007-2020. Advance Auto Parts showed it had been increasing from 2007-2016, but then declined to a local minimum in the middle of 2017 where it then increased and remained a steady value up to 2020.

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)
```



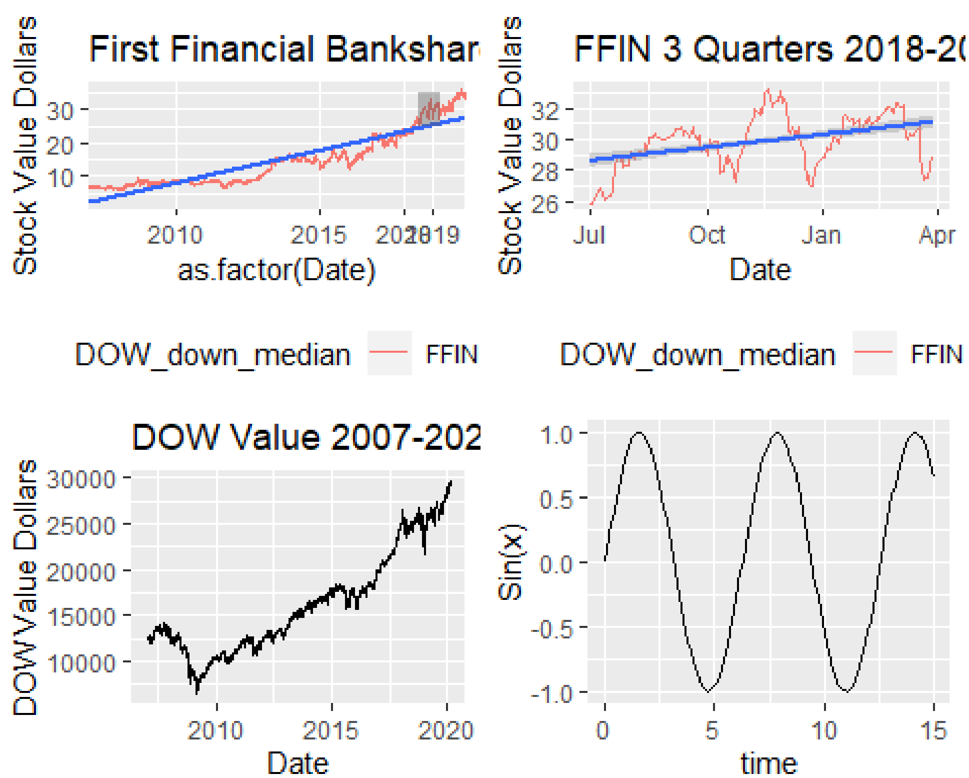


The above chart shows Google and First Financial Bankshares from 2007-2020. First Financial has many highs and lows between 2015-2020 but is overall increasing, while Google is steadily increasing up till 2017 when it has a few highs and lows until 2020. Both have increased overall as indicated by the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

```r
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))
```

```r
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.

```r
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.

```r
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 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$Q3_cSum_incr_L7 <- '3rd Qntl cSum up L7'
dROI17_lag7$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."]))
```

```r
#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.

```r
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:

```r
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'
```

```r
# 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]
```

```r
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:

```r
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())
```

```r
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
```

```r
print('****************************************************************')
```

```
## [1] "****************************************************************"
```

```r
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)
```

```r
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
```

```r
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())
```

```r
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] "*****************************************************************"
```

```r
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:

```r
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)
```

```r
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
```

```r
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
```

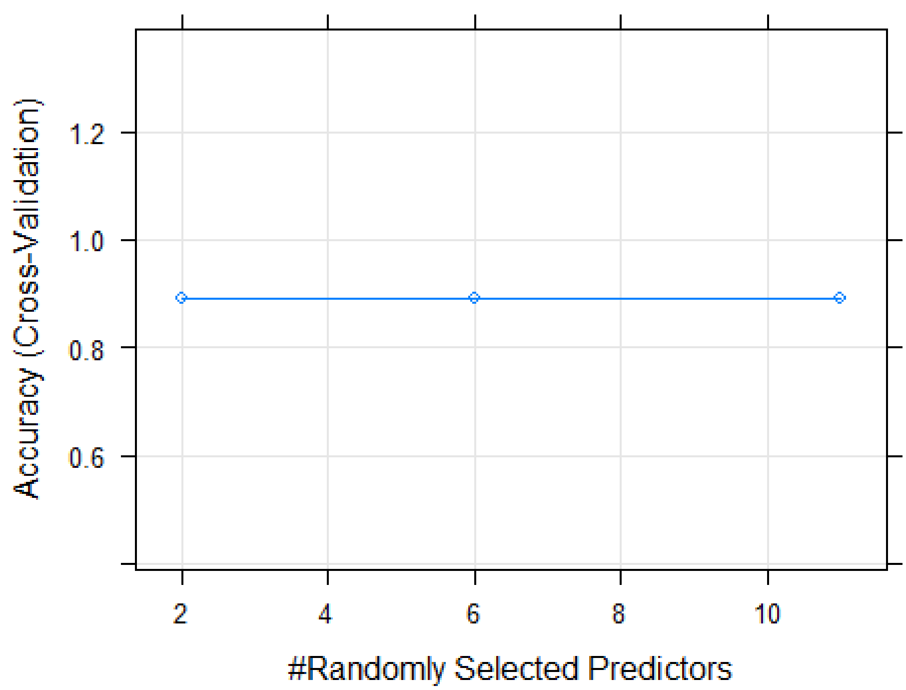
```
## TJX           25.96       7.17       63.38     2.36       8.84
4.0
## WMT           69.78      47.55      117.89     2.36       2.48
4.0
## XOM           81.42      74.11       60.65     2.36       0.82
3.0
##       Q3_cSum_decr_L7 max_cSum_decr_L7 medn_cSum_incr_L7 Q3_cSum_incr_L7
## AAP                 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)
```



```r
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
```

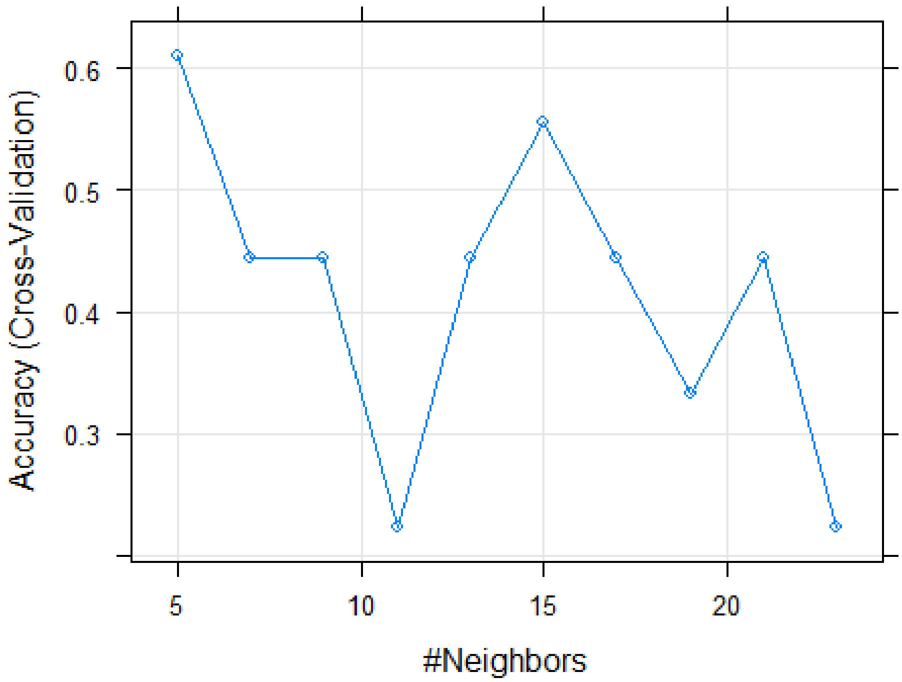
```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
uniqueCut =
## 10, : These variables have zero variances: DOW_ROI

## 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)
```

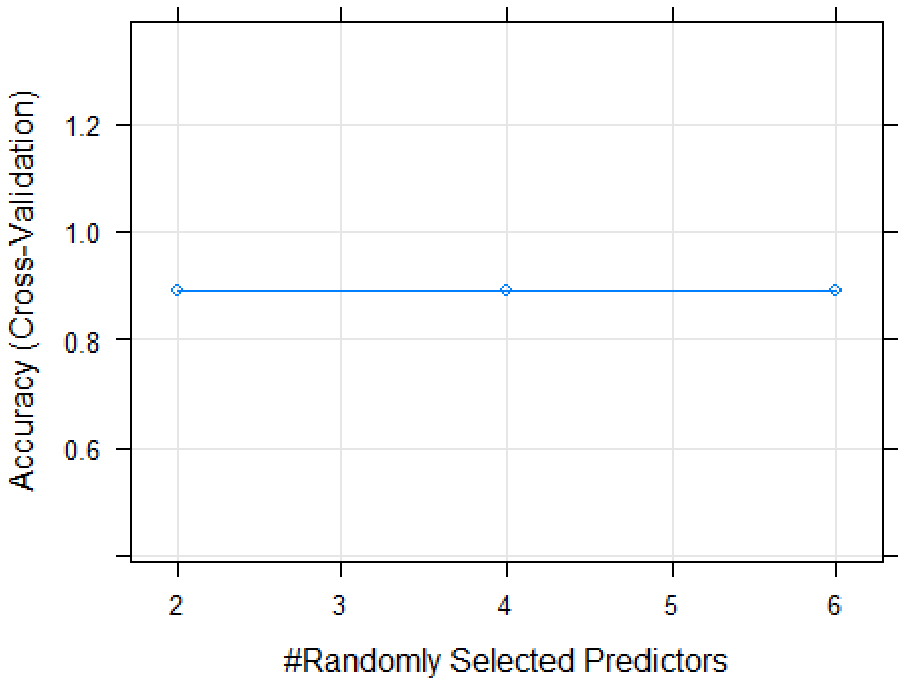



```
rpartMod <- train(ROI_Low_High ~ ., method='rpart', tuneLength=7,
data=trainingSet)

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
```

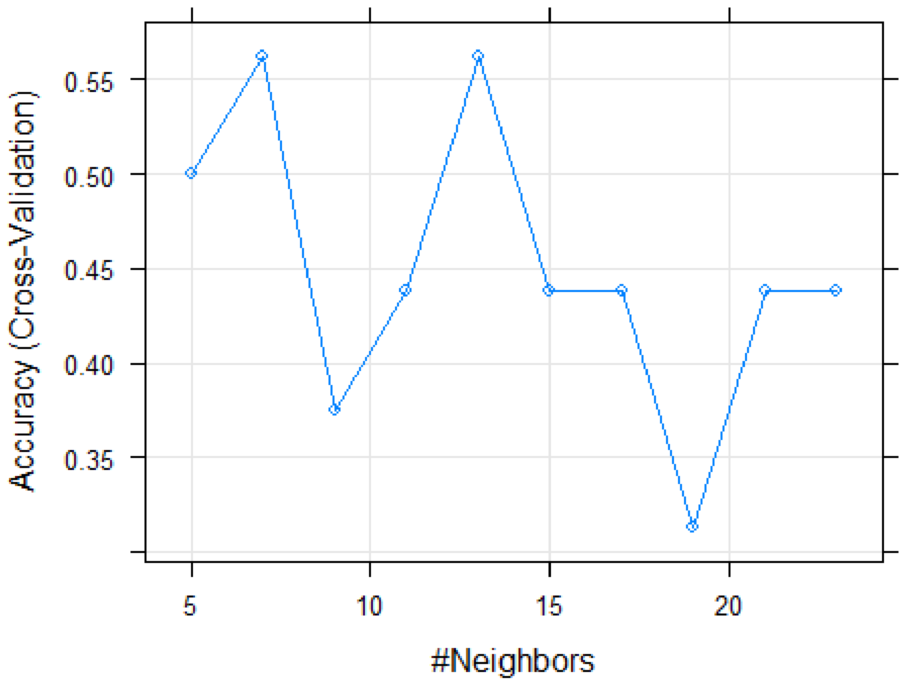
```r
knnMod <- train(ROI_Low_High ~ .,
                method='knn', preProcess=c('center','scale'),
                tuneLength=10, trControl=trainControl(method='cv'),
data=trainingSet)
```

```
## 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
```

```r
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.

```r
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)
```

```r
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
```

```r
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
```

```r
# 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)
```

```r
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
```

```r
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())
```

```r
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
```

```r
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
```

```r
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
```

```r
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
```

```r
median(ftr7_incr_Days_grouped$ftr7_incr_Days)
```

```
## [1] 12
```

```r
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]
```

```r
#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]
```

```r
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
```

```r
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
```

```r
median(hmc7_incr_Days_grouped$hmc7_incr_Days)
```

```
## [1] 14
```

```r
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))
```

```r
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
```

```r
#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
```

```r
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
```

```r
median(hrb7_decr_Days_grouped$hrb7_decr_Days)
```

```
## [1] 14.5
```

```r
#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
```

```r
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
```

```r
median(hrb7_incr_Days_grouped$hrb7_incr_Days)
```

```
## [1] 12
```

```r
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
```

```r
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]
```

```r
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
```

```r
median(jblu7_incr_Days_grouped$jblu7_incr_Days)
```

```
## [1] 12
```

```r
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]
```

```r
# 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
```

```r
median(jwn7_incr_Days_grouped$jwn7_incr_Days)

## [1] 11.5
```

```r
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]
```

```r
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
```

```r
median(kgji7_decr_Days_grouped$kgji7_decr_Days)
```

```
## [1] 14.5
```

```r
#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
```

```r
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
```

```r
median(kgji7_incr_Days_grouped$kgji7_incr_Days)
```

```
## [1] 11.5
```

```r
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
```

```r
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)
```

```r
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
```

```r
# 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
```

```r
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
```

```r
# 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)
```

```r
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
```

```r
NSANY <- subset(remaining36_g, remaining36_g$stockName=='NSANY')

NSANY <- NSANY[complete.cases(NSANY),]

nsany7_a <- ifelse(NSANY$today2lag7>1, 1,0)
```

```r
# 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
```

```r
# 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
```

```r
median(nsany7_decr_Days_grouped$nsany7_decr_Days)
```

```
## [1] 12.5
```

```r
#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
```

```r
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
```

```r
median(nsany7_incr_Days_grouped$nsany7_incr_Days)
```

```
## [1] 12.5
```

```r
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())
```

```r
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))
```

```r
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
```

```r
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,
```

```r
# 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
```

```r
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]
```

```r
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
```

```r
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
```

```r
median(sce7_incr_Days_grouped$sce7_incr_Days)
```

```
## [1] 8
```

```r
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
```

```r
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
```

```r
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)
```

```r
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
```

```r
#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
```

```r
median(ubsi7_decr_Days_grouped$ubsi7_decr_Days)
```

```
## [1] 12
```

```r
#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
```

```r
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
```

```r
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]
```

```r
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
```

```r
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'
```

```r
# 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
```

```r
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
```

```r
median(wm7_incr_Days_grouped$wm7_incr_Days)
```

```
## [1] 11
```

```r
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
```

```r
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)
```

```r
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.

```r
ALL_52 <- read.csv('ALL_52.csv', sep=',', header=TRUE, na.strings=c('',' '))
```

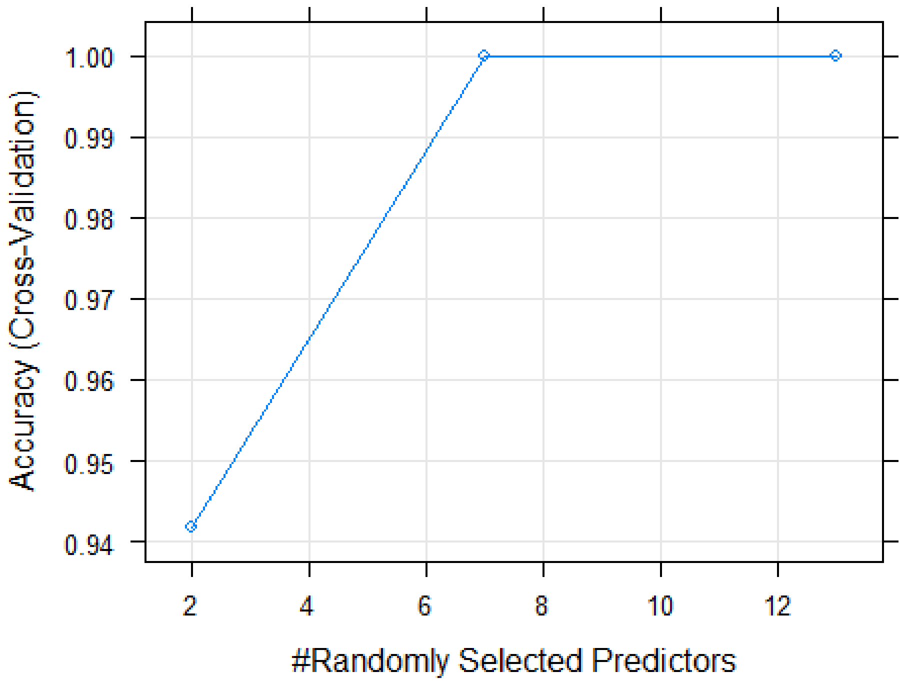
Lets first get the median value for these stocks' ROI.

```r
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.

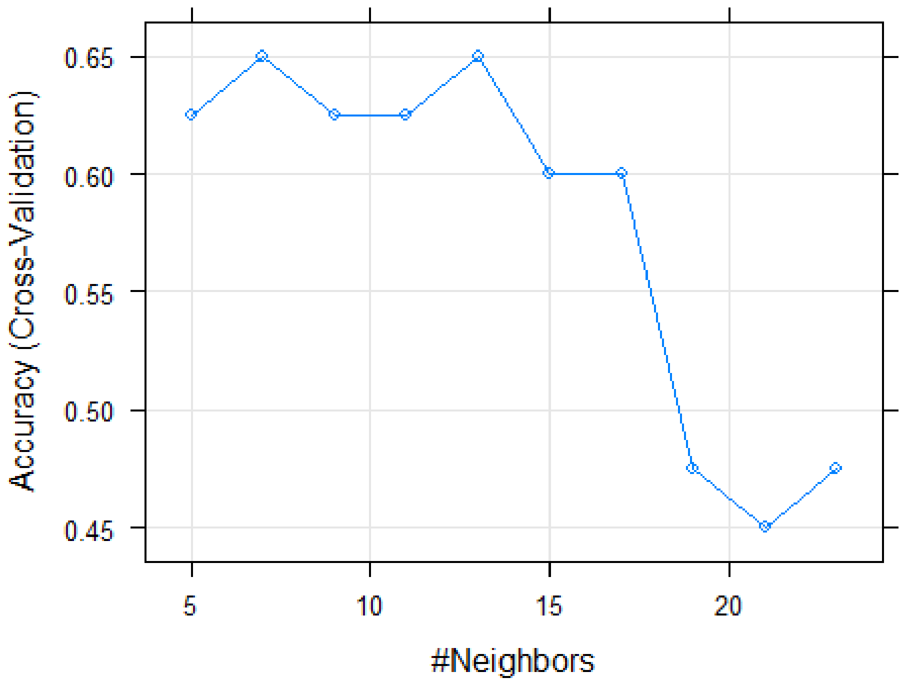
```r
med52ROI <- median(ALL_52$stock_ROI)
med52ROI

## [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)
```

```r
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)
```




```r
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
```

```r
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
```

```r
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
```

```r
#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
```

```r
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

```r
#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
```

```r
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
```

```r
#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)))
```

```r
#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

```r
#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])
```

```r
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

```r
#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
```

```r
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])
```

```r
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

```r
#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
```

```r
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)
```

```r
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)
```