10 Years of Historical Price 01/09/2008 ~ 01/09/2018

SECTION 0 - OBTAIN AND TRANSFORMING DATA

3M CSV dataset

dataset\_MMM <- read.csv("MMM.csv")  
head(dataset\_MMM)

## Date Open High Low Close Adj.Close Volume  
## 1 2008-01-08 81.35 82.07 80.10 80.21 60.62685 5711200  
## 2 2008-01-09 80.20 80.82 78.49 80.19 60.61175 6422300  
## 3 2008-01-10 79.75 80.71 79.02 80.21 60.62685 6835800  
## 4 2008-01-11 79.63 81.06 76.97 77.58 58.63898 8414500  
## 5 2008-01-14 78.13 78.64 77.50 78.50 59.33435 2927600  
## 6 2008-01-15 78.00 78.70 77.03 77.18 58.33664 4206500

summary(dataset\_MMM)

## Date Open High Low   
## 2008-01-08: 1 Min. : 41.31 Min. : 42.45 Min. : 40.87   
## 2008-01-09: 1 1st Qu.: 81.26 1st Qu.: 81.83 1st Qu.: 80.50   
## 2008-01-10: 1 Median : 96.92 Median : 97.50 Median : 96.57   
## 2008-01-11: 1 Mean :118.35 Mean :119.20 Mean :117.52   
## 2008-01-14: 1 3rd Qu.:157.59 3rd Qu.:158.46 3rd Qu.:156.73   
## 2008-01-15: 1 Max. :243.19 Max. :244.23 Max. :239.18   
## (Other) :2513   
## Close Adj.Close Volume   
## Min. : 41.83 Min. : 32.86 Min. : 651000   
## 1st Qu.: 81.22 1st Qu.: 66.41 1st Qu.: 2026100   
## Median : 97.21 Median : 82.92 Median : 2782500   
## Mean :118.42 Mean :105.61 Mean : 3288103   
## 3rd Qu.:157.65 3rd Qu.:145.32 3rd Qu.: 3943750   
## Max. :243.14 Max. :238.47 Max. :16317000   
##

C CSV dataset

dataset\_C <- read.csv("C.csv")  
head(dataset\_C)

## Date Open High Low Close Adj.Close Volume  
## 1 2008-09-02 196.3 198.2 188.8 191.1 182.3512 7979000  
## 2 2008-09-03 191.2 196.6 187.7 196.1 187.1223 6362700  
## 3 2008-09-04 193.0 194.7 182.1 183.0 174.6220 8057300  
## 4 2008-09-05 180.5 191.8 179.3 190.7 181.9695 8401400  
## 5 2008-09-08 208.4 209.5 195.0 203.2 193.8972 17043000  
## 6 2008-09-09 200.2 203.8 188.1 188.8 180.1565 11634800

summary(dataset\_C)

## Date Open High Low   
## 2008-09-02: 1 Min. : 10.20 Min. : 10.70 Min. : 9.70   
## 2008-09-03: 1 1st Qu.: 39.34 1st Qu.: 39.90 1st Qu.: 38.59   
## 2008-09-04: 1 Median : 47.60 Median : 48.00 Median : 47.07   
## 2008-09-05: 1 Mean : 50.17 Mean : 50.91 Mean : 49.28   
## 2008-09-08: 1 3rd Qu.: 54.46 3rd Qu.: 54.95 3rd Qu.: 54.01   
## 2008-09-09: 1 Max. :231.60 Max. :235.00 Max. :220.00   
## (Other) :2513   
## Close Adj.Close Volume   
## Min. : 10.20 Min. : 9.747 Min. : 3998000   
## 1st Qu.: 39.10 1st Qu.: 37.393 1st Qu.: 16221050   
## Median : 47.56 Median : 45.603 Median : 24164500   
## Mean : 50.04 Mean : 48.246 Mean : 32159178   
## 3rd Qu.: 54.48 3rd Qu.: 52.304 3rd Qu.: 39243400   
## Max. :230.00 Max. :219.470 Max. :377263800   
##

PG CSV dataset

dataset\_PG <- read.csv("PG.csv")  
head(dataset\_PG)

## Date Open High Low Close Adj.Close Volume  
## 1 2008-09-02 70.35 71.43 70.15 70.47 51.39056 11255200  
## 2 2008-09-03 70.42 71.50 70.18 71.42 52.08335 10433900  
## 3 2008-09-04 71.16 71.52 70.37 70.44 51.36869 12298000  
## 4 2008-09-05 70.39 71.07 69.73 70.78 51.61662 11578800  
## 5 2008-09-08 71.33 72.48 71.31 72.37 52.77613 16521200  
## 6 2008-09-09 71.66 72.46 71.47 71.89 52.42612 18542200

summary(dataset\_PG)

## Date Open High Low   
## 2008-09-02: 1 Min. :44.51 Min. :45.19 Min. :39.37   
## 2008-09-03: 1 1st Qu.:63.40 1st Qu.:63.80 1st Qu.:62.97   
## 2008-09-04: 1 Median :76.39 Median :76.99 Median :75.81   
## 2008-09-05: 1 Mean :73.44 Mean :73.93 Mean :72.96   
## 2008-09-08: 1 3rd Qu.:82.56 3rd Qu.:83.08 3rd Qu.:82.10   
## 2008-09-09: 1 Max. :94.17 Max. :94.67 Max. :93.83   
## (Other) :2513   
## Close Adj.Close Volume   
## Min. :44.18 Min. :32.65 Min. : 2196700   
## 1st Qu.:63.33 1st Qu.:49.97 1st Qu.: 7058250   
## Median :76.41 Median :66.62 Median : 8970100   
## Mean :73.46 Mean :63.62 Mean : 10524215   
## 3rd Qu.:82.62 3rd Qu.:76.19 3rd Qu.: 11791750   
## Max. :94.40 Max. :91.31 Max. :123735700   
##

GOOG CSV dataset

dataset\_GOOG <- read.csv("GOOG.csv")  
head(dataset\_GOOG)

## Date Open High Low Close Adj.Close Volume  
## 1 2008-09-02 236.8446 239.5321 229.2191 231.1218 231.1218 12302400  
## 2 2008-09-03 232.8505 235.6126 228.3051 230.7045 230.7045 8685300  
## 3 2008-09-04 228.5137 230.1233 223.2480 223.6752 223.6752 9759900  
## 4 2008-09-05 221.3056 224.7681 218.6181 220.6896 220.6896 9127500  
## 5 2008-09-08 224.5495 225.0065 207.4259 208.6181 208.6181 18153000  
## 6 2008-09-09 210.2177 214.7930 206.1591 207.9773 207.9773 14553100

summary(dataset\_GOOG)

## Date Open High Low   
## 2008-09-02: 1 Min. : 130.4 Min. : 133.8 Min. : 122.9   
## 2008-09-03: 1 1st Qu.: 285.6 1st Qu.: 287.6 1st Qu.: 282.7   
## 2008-09-04: 1 Median : 444.0 Median : 446.0 Median : 440.2   
## 2008-09-05: 1 Mean : 515.8 Mean : 520.1 Mean : 511.2   
## 2008-09-08: 1 3rd Qu.: 722.8 3rd Qu.: 727.3 3rd Qu.: 716.5   
## 2008-09-09: 1 Max. :1271.0 Max. :1273.9 Max. :1249.0   
## (Other) :2513   
## Close Adj.Close Volume   
## Min. : 127.9 Min. : 127.9 Min. : 7900   
## 1st Qu.: 286.0 1st Qu.: 286.0 1st Qu.: 1657600   
## Median : 442.6 Median : 442.6 Median : 3445800   
## Mean : 515.8 Mean : 515.8 Mean : 4289452   
## 3rd Qu.: 720.8 3rd Qu.: 720.8 3rd Qu.: 5423650   
## Max. :1268.3 Max. :1268.3 Max. :32690500   
##

MYL CSV dataset

dataset\_MYL <- read.csv("MYL.csv")  
head(dataset\_MYL)

## Date Open High Low Close Adj.Close Volume  
## 1 2008-09-02 12.89 13.14 12.68 12.83 12.83 5319000  
## 2 2008-09-03 12.84 12.90 12.60 12.87 12.87 5851400  
## 3 2008-09-04 12.82 13.01 12.66 12.84 12.84 5865200  
## 4 2008-09-05 12.72 12.84 12.46 12.50 12.50 3865900  
## 5 2008-09-08 12.69 12.80 12.33 12.42 12.42 3591700  
## 6 2008-09-09 11.80 12.45 10.93 11.10 11.10 28487100

summary(dataset\_MYL)

## Date Open High Low   
## 2008-09-02: 1 Min. : 5.98 Min. : 6.25 Min. : 5.75   
## 2008-09-03: 1 1st Qu.:20.92 1st Qu.:21.12 1st Qu.:20.60   
## 2008-09-04: 1 Median :32.10 Median :32.64 Median :31.80   
## 2008-09-05: 1 Mean :33.14 Mean :33.58 Mean :32.66   
## 2008-09-08: 1 3rd Qu.:43.97 3rd Qu.:44.57 3rd Qu.:43.02   
## 2008-09-09: 1 Max. :75.40 Max. :76.69 Max. :73.63   
## (Other) :2513   
## Close Adj.Close Volume   
## Min. : 5.77 Min. : 5.77 Min. : 856200   
## 1st Qu.:20.89 1st Qu.:20.89 1st Qu.: 3795500   
## Median :32.13 Median :32.13 Median : 5237900   
## Mean :33.12 Mean :33.12 Mean : 6130334   
## 3rd Qu.:43.80 3rd Qu.:43.80 3rd Qu.: 7200400   
## Max. :76.06 Max. :76.06 Max. :50917600   
##

Retrieve the daily closing price

# Initialize date column for analyze purpose  
close\_price = NULL  
close\_price$date = dataset\_MMM$Date  
  
close\_price$MMM = dataset\_MMM$Adj.Close  
close\_price$C = dataset\_C$Adj.Close  
close\_price$PG = dataset\_PG$Adj.Close  
close\_price$GOOG = dataset\_GOOG$Adj.Close  
close\_price$MYL = dataset\_MYL$Adj.Close  
  
#Show several example of each asset's close price  
head(close\_price$MMM)

## [1] 60.62685 60.61175 60.62685 58.63898 59.33435 58.33664

head(close\_price$C)

## [1] 182.3512 187.1223 174.6220 181.9695 193.8972 180.1565

head(close\_price$PG)

## [1] 51.39056 52.08335 51.36869 51.61662 52.77613 52.42612

head(close\_price$GOOG)

## [1] 231.1218 230.7045 223.6752 220.6896 208.6181 207.9773

head(close\_price$MYL)

## [1] 12.83 12.87 12.84 12.50 12.42 11.10

Construct the log daily returns of each asset

# Initialize date column for analyze purpose  
returns = NULL  
returns$MMM = diff(log(close\_price$MMM), lag = 1)  
returns$C = diff(log(close\_price$C), lag = 1)  
returns$PG = diff(log(close\_price$PG), lag = 1)  
returns$GOOG = diff(log(close\_price$GOOG), lag = 1)  
returns$MYL = diff(log(close\_price$MYL), lag = 1)  
  
#Show several example of each asset's close price  
head(returns$MMM)

## [1] -0.0002491946 0.0002491946 -0.0333383182 0.0117888012 -0.0169581124  
## [6] 0.0015540409

head(returns$C)

## [1] 0.02582777 -0.06913856 0.04121547 0.06348920 -0.07350256 -0.01064978

head(returns$PG)

## [1] 0.013390919 -0.013816555 0.004814910 0.022215387 -0.006654121  
## [6] 0.006930648

head(returns$GOOG)

## [1] -0.001807161 -0.030942593 -0.013437720 -0.056251814 -0.003076484  
## [6] -0.010806750

head(returns$MYL)

## [1] 0.003112843 -0.002333723 -0.026836654 -0.006420568 -0.112362968  
## [6] 0.032789823

Insert the result into data frame format

transformed\_data.close <- data.frame(close\_price$date, close\_price$MMM, close\_price$C, close\_price$PG, close\_price$GOOG, close\_price$MYL)  
str(transformed\_data.close)

## 'data.frame': 2519 obs. of 6 variables:  
## $ close\_price.date: Factor w/ 2519 levels "2008-01-08","2008-01-09",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ close\_price.MMM : num 60.6 60.6 60.6 58.6 59.3 ...  
## $ close\_price.C : num 182 187 175 182 194 ...  
## $ close\_price.PG : num 51.4 52.1 51.4 51.6 52.8 ...  
## $ close\_price.GOOG: num 231 231 224 221 209 ...  
## $ close\_price.MYL : num 12.8 12.9 12.8 12.5 12.4 ...

n <- nrow(transformed\_data.close)  
  
returns$date <- close\_price$date[2:n]  
  
transformed\_data.returns <- data.frame(returns$date, returns$MMM, returns$C, returns$PG, returns$GOOG, returns$MYL)  
str(transformed\_data.returns)

## 'data.frame': 2518 obs. of 6 variables:  
## $ returns.date: Factor w/ 2519 levels "2008-01-08","2008-01-09",..: 2 3 4 5 6 7 8 9 10 11 ...  
## $ returns.MMM : num -0.000249 0.000249 -0.033338 0.011789 -0.016958 ...  
## $ returns.C : num 0.0258 -0.0691 0.0412 0.0635 -0.0735 ...  
## $ returns.PG : num 0.01339 -0.01382 0.00481 0.02222 -0.00665 ...  
## $ returns.GOOG: num -0.00181 -0.03094 -0.01344 -0.05625 -0.00308 ...  
## $ returns.MYL : num 0.00311 -0.00233 -0.02684 -0.00642 -0.11236 ...

SECTION 1 - EXPLORATORY DATA ANALYSIS

Generate a time plot series of closing price on the same graph

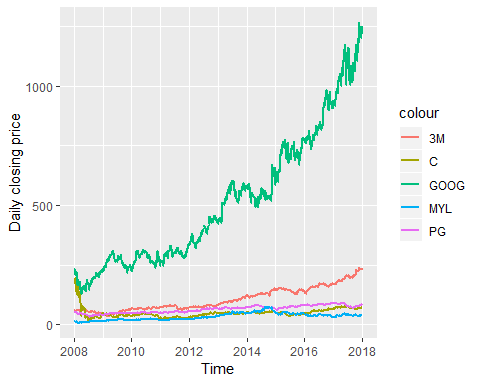
# Load necessary packages  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

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

library(ggplot2)  
  
  
# View time plot of the closing price  
 ggplot(data = transformed\_data.close,   
 aes(x=as.Date(close\_price$date), y=value, group=1) ) +   
 geom\_line(aes(y=close\_price$MMM, col='3M'), size = 1) +   
 geom\_line(aes(y=close\_price$C, col='C'), size = 1) +   
 geom\_line(aes(y=close\_price$PG, col='PG'), size = 1) +   
 geom\_line(aes(y=close\_price$GOOG, col='GOOG'), size = 1) +   
 geom\_line(aes(y=close\_price$MYL, col='MYL'), size = 1) +   
 labs(x = "Time", y = "Daily closing price")



Discussion: (Akmal)

(Mark)

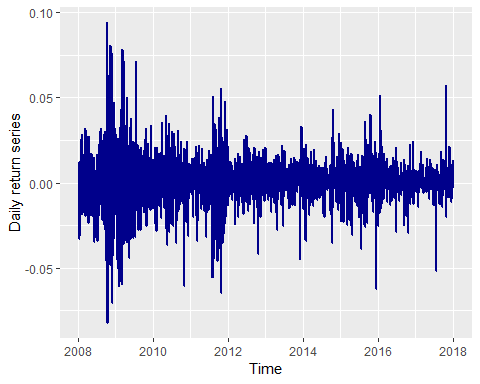
(Sudheer)

(Surya)

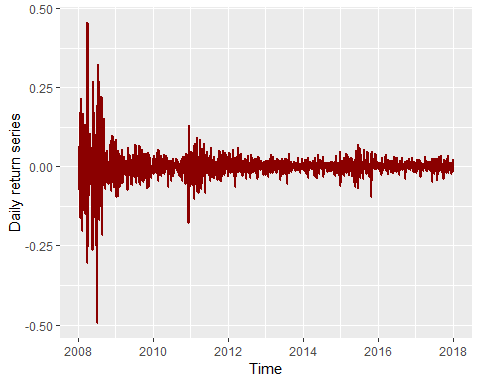
By looking at the daily closing price in 10 years from 2008 to 2018, I could see that the growth of the close price for GOOG (Alphabet Inc.) was positively growth over time. Initially, C (Citigroup Inc.) has a similar close price in 2008. However, their got a negative trend from early 2008 to mid 2008 which made them stuck for years ahead. Interestingly, even though the MMM (3M Company)’s performance was overshadowed with GOOG, they enjoyed quite a positive trend. The 3 others assets have a similar stagnant closing price over time.

Generate a time plot series of return series

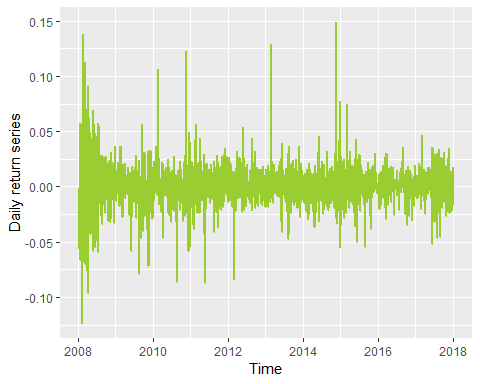
# View time plot of the 3M return price  
 ggplot(data = transformed\_data.returns,   
 aes(x=as.Date(returns$date), y=value, group=1) ) +   
 geom\_line(aes(y=returns$MMM, col='3M'), size = 1, color='darkblue') +   
 labs(x = "Time", y = "Daily return series")



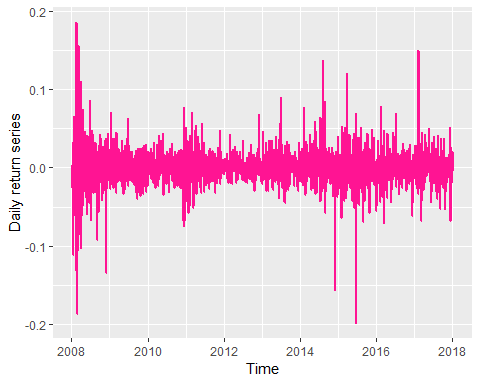
# View time plot of the C return price  
 ggplot(data = transformed\_data.returns,   
 aes(x=as.Date(returns$date), y=value, group=1) ) +   
 geom\_line(aes(y=returns$C, col='C'), size = 1, color='darkred') +   
 labs(x = "Time", y = "Daily return series")



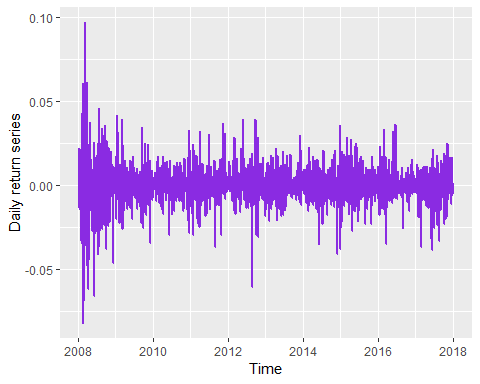
# View time plot of the GOOG return price  
 ggplot(data = transformed\_data.returns,   
 aes(x=as.Date(returns$date), y=value, group=1) ) +   
 geom\_line(aes(y=returns$GOOG, col='GOOG'), size = 1, color='yellowgreen') +   
 labs(x = "Time", y = "Daily return series")



# View time plot of the MYL return price  
 ggplot(data = transformed\_data.returns,   
 aes(x=as.Date(returns$date), y=value, group=1) ) +   
 geom\_line(aes(y=returns$MYL, col='MYL'), size = 1, color='deeppink') +   
 labs(x = "Time", y = "Daily return series")



# View time plot of the PG return price  
 ggplot(data = transformed\_data.returns,   
 aes(x=as.Date(returns$date), y=value, group=1) ) +   
 geom\_line(aes(y=returns$PG, col='PG'), size = 1, color='blueviolet') +   
 labs(x = "Time", y = "Daily return series")



Discussion: Similarities

The average value of all assets over 10 years was quite stable where the mean of log return is almost zero.

Difference - MMM (Surya)

The average value of 3M asset over 10 years was quite stable where the mean of log return is almost zero. However, the volatility becomes larger for quite a long time at the end of 2008 and at the end of 2011; for a short time at the end of 2015 and mid 2017.

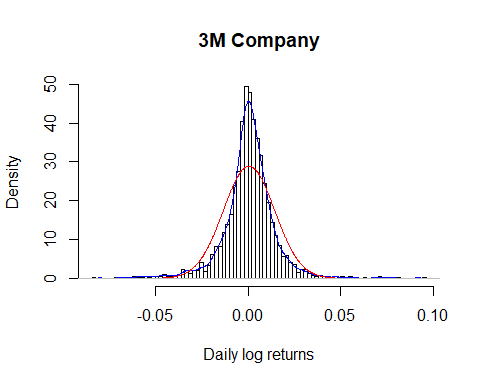
* C
* GOOG
* MYL
* PG

Generate a kernel density plot of return series

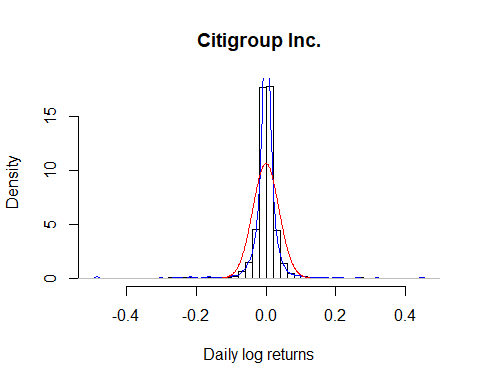
library(kdensity)  
  
# MMM  
hist(transformed\_data.returns$returns.MMM, breaks = 64, freq = FALSE, main = "3M Company", xlab = "Daily log returns")  
kde\_mmm = kdensity(transformed\_data.returns$returns.MMM, start = "normal")

## Warning: namespace 'extraDistr' is not available and has been replaced  
## by .GlobalEnv when processing object ''

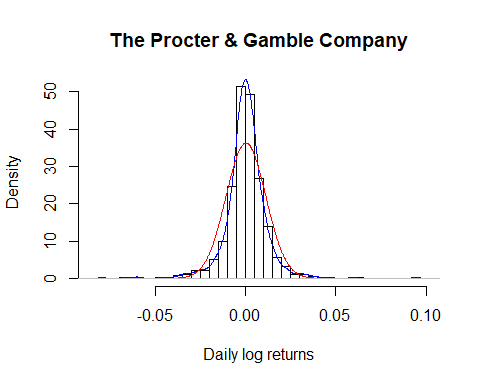
lines(kde\_mmm, col="blue")  
lines(kde\_mmm, plot\_start = TRUE, col="red")



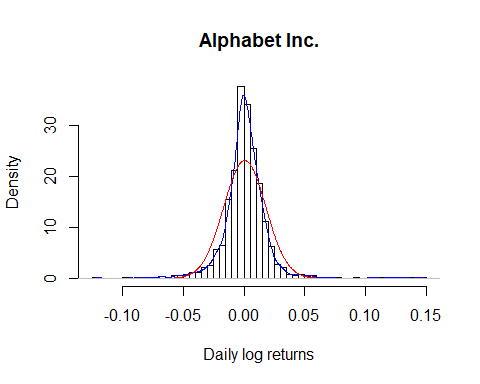
# C  
hist(transformed\_data.returns$returns.C, breaks = 64, freq = FALSE, main = "Citigroup Inc.", xlab = "Daily log returns")  
kde\_c = kdensity(transformed\_data.returns$returns.C, start = "normal")  
lines(kde\_c, col="blue")  
lines(kde\_c, plot\_start = TRUE, col="red")



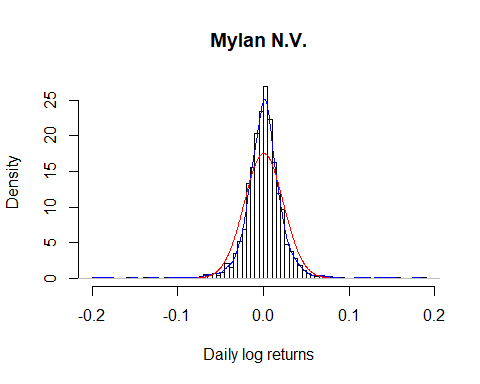
# PG  
hist(transformed\_data.returns$returns.PG, breaks = 64, freq = FALSE, main = "The Procter & Gamble Company", xlab = "Daily log returns")  
kde\_pg = kdensity(transformed\_data.returns$returns.PG, start = "normal")  
lines(kde\_pg, col="blue")  
lines(kde\_pg, plot\_start = TRUE, col="red")



# GOOG  
hist(transformed\_data.returns$returns.GOOG, breaks = 64, freq = FALSE, main = "Alphabet Inc.", xlab = "Daily log returns")  
kde\_goog = kdensity(transformed\_data.returns$returns.GOOG, start = "normal")  
lines(kde\_goog, col="blue")  
lines(kde\_goog, plot\_start = TRUE, col="red")



# MYL  
hist(transformed\_data.returns$returns.MYL, breaks = 64, freq = FALSE, main = "Mylan N.V.", xlab = "Daily log returns")  
kde\_myl = kdensity(transformed\_data.returns$returns.MYL, start = "normal")  
lines(kde\_myl, col="blue")  
lines(kde\_myl, plot\_start = TRUE, col="red")



Discussion: Similarities

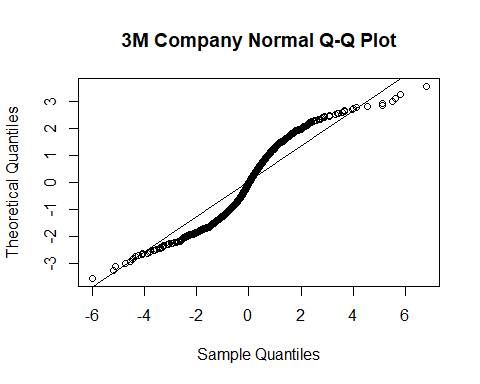
Difference - MMM (Surya)

There is a slight different in the left and left side which the density is quite lower than the right side. Thus, we could say that the data doesn’t normally distributed. The kernel density has a higher kurtosis which result in much longer than normal tails of the density.

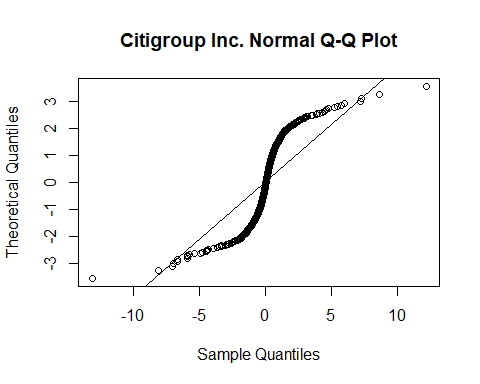
* C
* GOOG
* MYL
* PG

Generate a normal Q-Q plot of return series

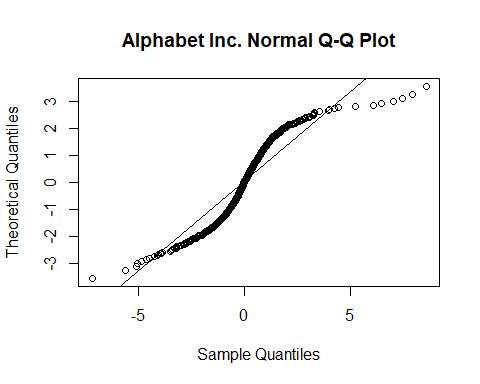
#MMM  
mean\_mmm = mean(transformed\_data.returns$returns.MMM)  
sd\_mmm = sd(transformed\_data.returns$returns.MMM)  
mmm\_std = (transformed\_data.returns$returns.MMM - mean\_mmm) / sd\_mmm  
qqnorm(mmm\_std, main="3M Company Normal Q-Q Plot", plot.it = TRUE, datax = TRUE)  
qqline(mmm\_std, datax = FALSE, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7)



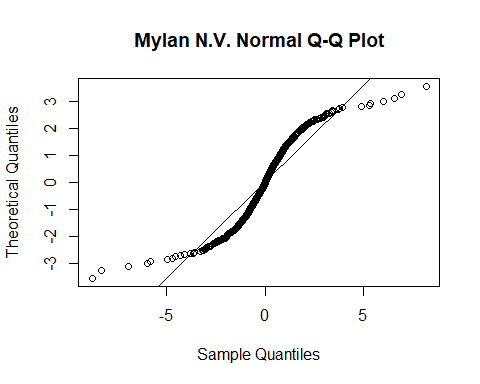
#C  
mean\_c = mean(transformed\_data.returns$returns.C)  
sd\_c = sd(transformed\_data.returns$returns.C)  
c\_std = (transformed\_data.returns$returns.C - mean\_c) / sd\_c  
qqnorm(c\_std, main="Citigroup Inc. Normal Q-Q Plot", plot.it = TRUE, datax = TRUE)  
qqline(c\_std, datax = FALSE, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7)



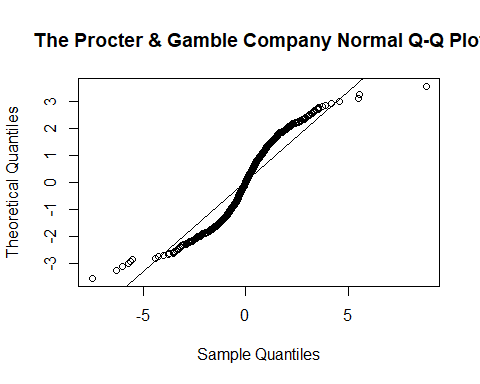
#GOOG  
mean\_goog = mean(transformed\_data.returns$returns.GOOG)  
sd\_goog = sd(transformed\_data.returns$returns.GOOG)  
goog\_std = (transformed\_data.returns$returns.GOOG - mean\_goog) / sd\_goog  
qqnorm(goog\_std, main="Alphabet Inc. Normal Q-Q Plot", plot.it = TRUE, datax = TRUE)  
qqline(goog\_std, datax = FALSE, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7)



#MYL  
mean\_myl = mean(transformed\_data.returns$returns.MYL)  
sd\_myl = sd(transformed\_data.returns$returns.MYL)  
myl\_std = (transformed\_data.returns$returns.MYL - mean\_myl) / sd\_myl  
qqnorm(myl\_std, main="Mylan N.V. Normal Q-Q Plot", plot.it = TRUE, datax = TRUE)  
qqline(myl\_std, datax = FALSE, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7)



#PG  
mean\_pg = mean(transformed\_data.returns$returns.PG)  
sd\_pg = sd(transformed\_data.returns$returns.PG)  
pg\_std = (transformed\_data.returns$returns.PG - mean\_pg) / sd\_pg  
qqnorm(pg\_std, main="The Procter & Gamble Company Normal Q-Q Plot", plot.it = TRUE, datax = TRUE)  
qqline(pg\_std, datax = FALSE, distribution = qnorm, probs = c(0.25, 0.75), qtype = 7)



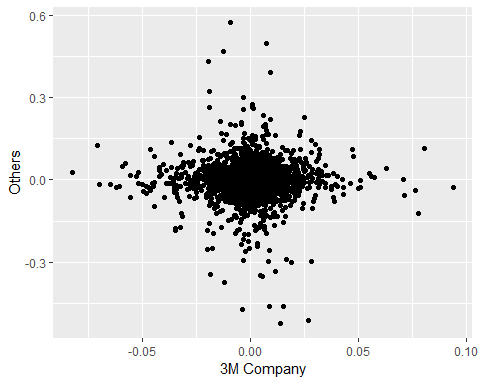
Discussion: Similarities

Difference - MMM (Surya) The 3M asset has a thick tails which mean that, compared to the normal distribution, there is more data located at the extremes of the distribution and less data in the center of the distribution. In terms of quantiles this means that the first quantile is much less than the first theoretical quantile and the last quantile is greater than the last theoretical quantile.

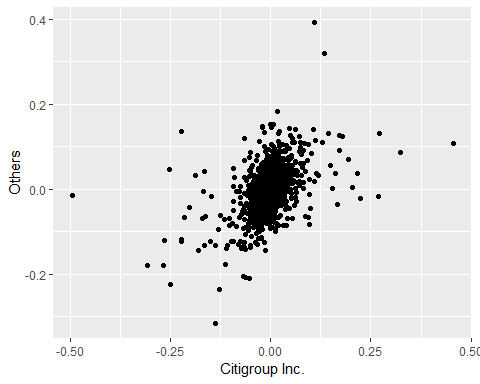
* C
* GOOG
* MYL
* PG

Generate a scatter plot of return series for one asset againts the others

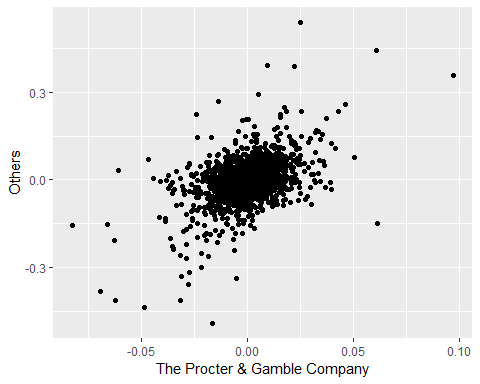
ggplot(data = transformed\_data.returns, aes(returns$MMM, (returns$C + returns$PG + returns$GOOG + returns$MYL))) +  
 geom\_point() +   
 labs(x = "3M Company", y = "Others")



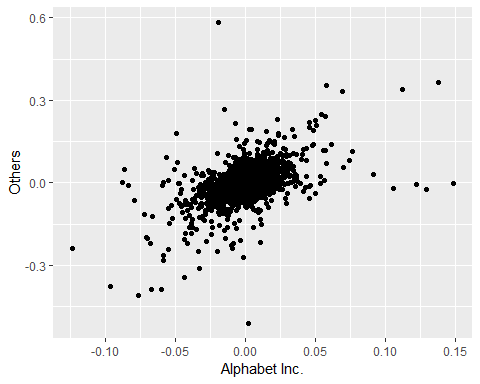
ggplot(data = transformed\_data.returns, aes(returns$C, (returns$MMM + returns$PG + returns$GOOG + returns$MYL))) +  
 geom\_point() +   
 labs(x = "Citigroup Inc.", y = "Others")



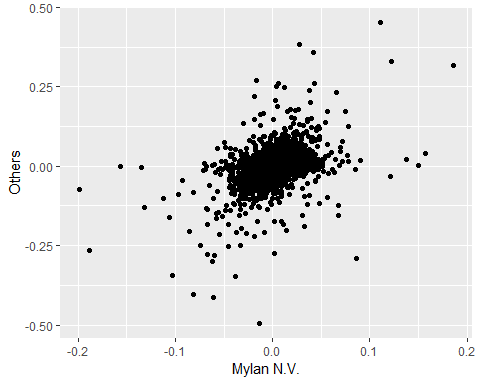
ggplot(data = transformed\_data.returns, aes(returns$PG, (returns$MMM + returns$C + returns$GOOG + returns$MYL))) +  
 geom\_point() +   
 labs(x = "The Procter & Gamble Company", y = "Others")



ggplot(data = transformed\_data.returns, aes(returns$GOOG, (returns$MMM + returns$C + returns$PG + returns$MYL))) +  
 geom\_point() +   
 labs(x = "Alphabet Inc.", y = "Others")



ggplot(data = transformed\_data.returns, aes(returns$MYL, (returns$MMM + returns$C + returns$GOOG + returns$PG))) +  
 geom\_point() +   
 labs(x = "Mylan N.V.", y = "Others")



Compute the correlation matrix between all assets

return\_assets <- data.frame(transformed\_data.returns$returns.MMM, transformed\_data.returns$returns.C, transformed\_data.returns$returns.GOOG, transformed\_data.returns$returns.MYL, transformed\_data.returns$returns.PG)  
  
cor(return\_assets)

## transformed\_data.returns.returns.MMM  
## transformed\_data.returns.returns.MMM 1.00000000  
## transformed\_data.returns.returns.C -0.01547224  
## transformed\_data.returns.returns.GOOG 0.01443115  
## transformed\_data.returns.returns.MYL -0.01088348  
## transformed\_data.returns.returns.PG -0.01881660  
## transformed\_data.returns.returns.C  
## transformed\_data.returns.returns.MMM -0.01547224  
## transformed\_data.returns.returns.C 1.00000000  
## transformed\_data.returns.returns.GOOG 0.41930042  
## transformed\_data.returns.returns.MYL 0.34499815  
## transformed\_data.returns.returns.PG 0.35257870  
## transformed\_data.returns.returns.GOOG  
## transformed\_data.returns.returns.MMM 0.01443115  
## transformed\_data.returns.returns.C 0.41930042  
## transformed\_data.returns.returns.GOOG 1.00000000  
## transformed\_data.returns.returns.MYL 0.38438938  
## transformed\_data.returns.returns.PG 0.40576726  
## transformed\_data.returns.returns.MYL  
## transformed\_data.returns.returns.MMM -0.01088348  
## transformed\_data.returns.returns.C 0.34499815  
## transformed\_data.returns.returns.GOOG 0.38438938  
## transformed\_data.returns.returns.MYL 1.00000000  
## transformed\_data.returns.returns.PG 0.37663423  
## transformed\_data.returns.returns.PG  
## transformed\_data.returns.returns.MMM -0.0188166  
## transformed\_data.returns.returns.C 0.3525787  
## transformed\_data.returns.returns.GOOG 0.4057673  
## transformed\_data.returns.returns.MYL 0.3766342  
## transformed\_data.returns.returns.PG 1.0000000

Discussion: Difference - MMM (Surya) Based on the result, the correlation between MMM with the others is quite small, which is either -0.01 or 0.01. Thus, I could conclude that MMM asset has quite a strong correlation with the other assets

* C
* GOOG
* MYL
* PG

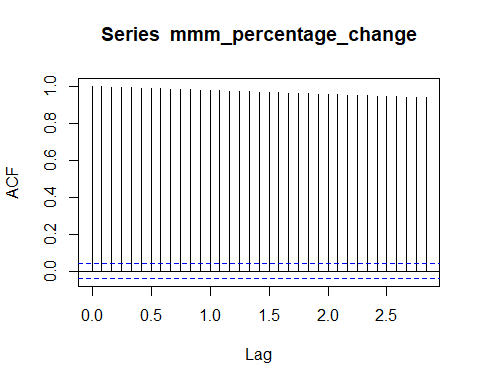
SECTION 2 - UNIVARIATE TIME SERIES ANALYSIS

Load required libraries

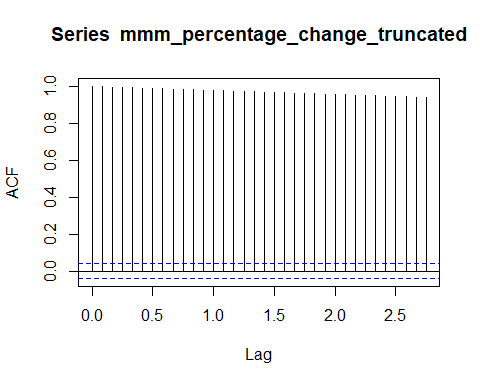
library(stats)  
library(tseries)  
library(forecast)

SECTION 2.1 - AUTOCORRELATION Stationary analysis of the price series

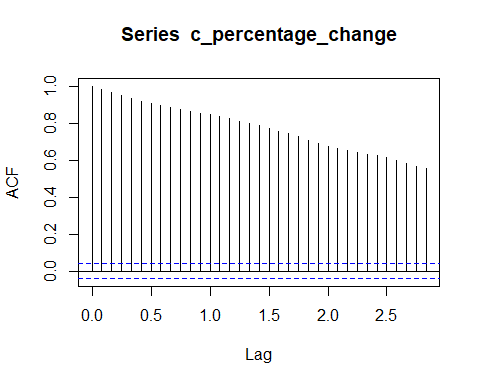
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.close$close\_price.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
acf(mmm\_percentage\_change)



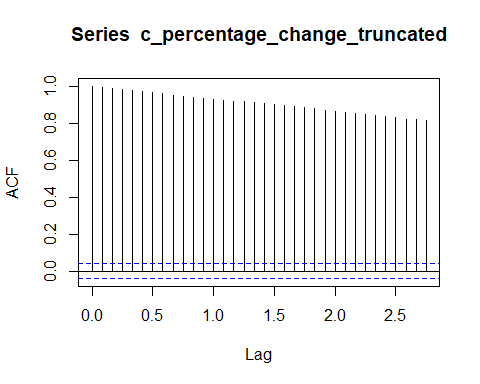
acf(mmm\_percentage\_change\_truncated)



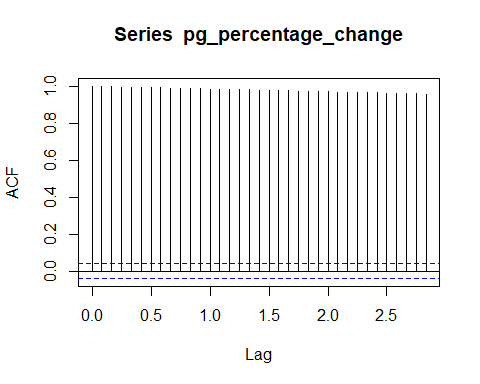
# C  
c\_percentage\_change <- ts(transformed\_data.close$close\_price.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
acf(c\_percentage\_change)



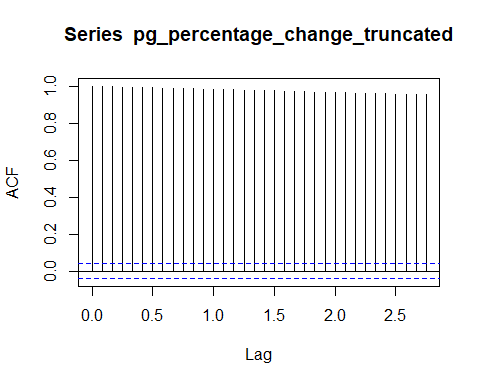
acf(c\_percentage\_change\_truncated)



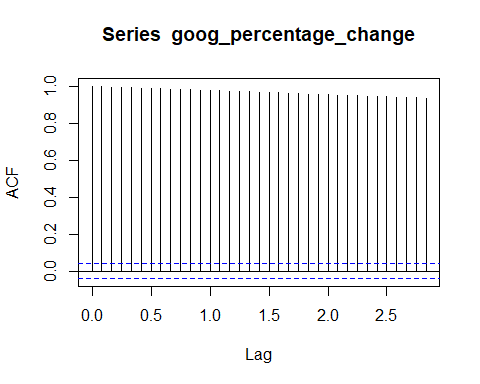
# PG  
pg\_percentage\_change <- ts(transformed\_data.close$close\_price.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
acf(pg\_percentage\_change)



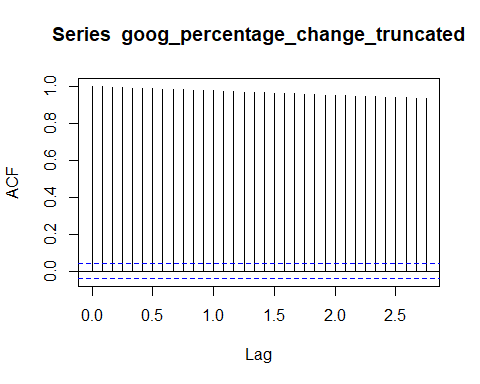
acf(pg\_percentage\_change\_truncated)



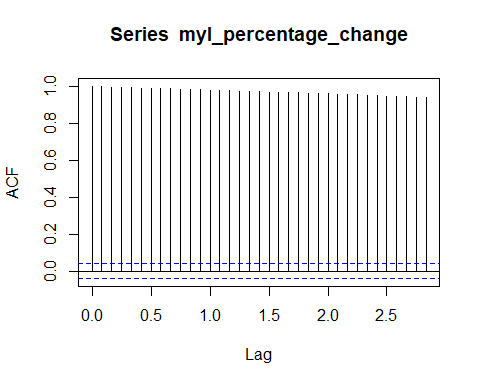
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.close$close\_price.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
acf(goog\_percentage\_change)



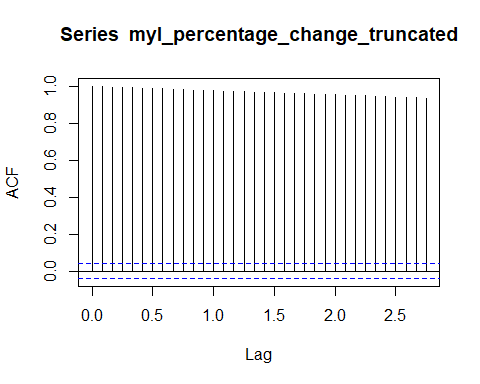
acf(goog\_percentage\_change\_truncated)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.close$close\_price.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
acf(myl\_percentage\_change)



acf(myl\_percentage\_change\_truncated)



Discussion: Similarities

Difference - MMM (Surya)

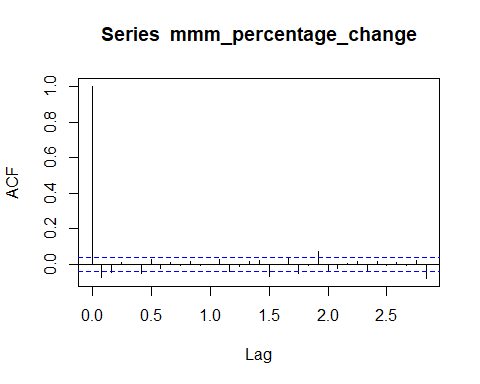
In this full autocorrelation, the spike was quite stable on each lag, where it barely decreased. Because the spike has a value that is significantly different from zero, it becomes the evidence of autocorrelation. It also means when the close price of MMM stock rises, it tends to continue rising and vise versa. It also means that this is a non-stationary series because it is decaying.

The truncated autocorrelation also shows quite a similar result with the full autocorrelation which reinforced my argument that the MMM asset is quite stable over time.

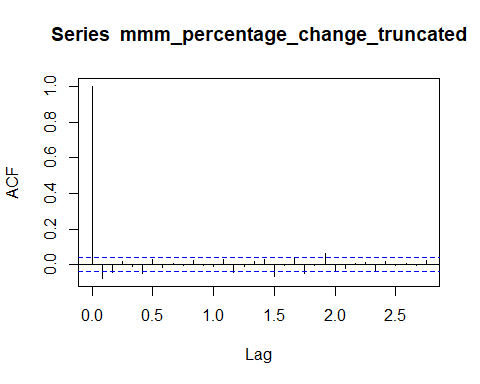
* C
* GOOG
* MYL
* PG

Stationary analysis of the returns series

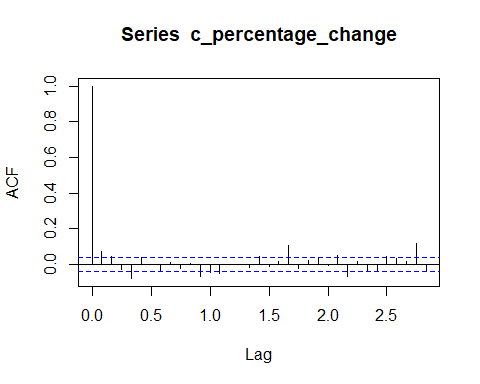
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
acf(mmm\_percentage\_change)



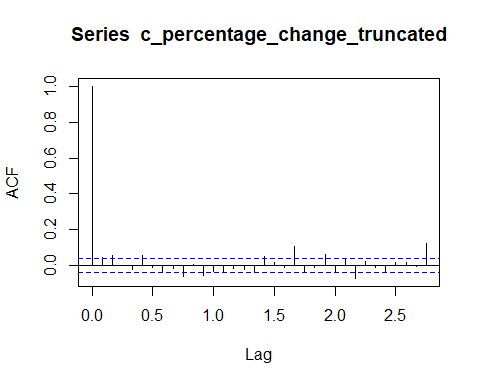
acf(mmm\_percentage\_change\_truncated)



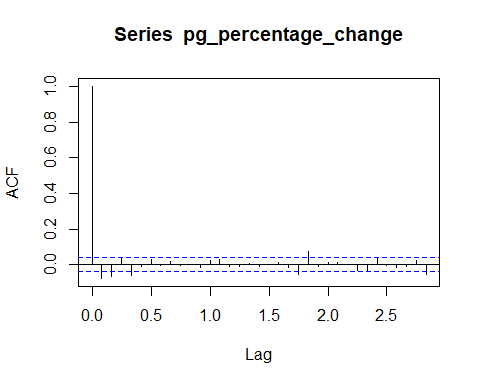
# C  
c\_percentage\_change <- ts(transformed\_data.returns$returns.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
acf(c\_percentage\_change)



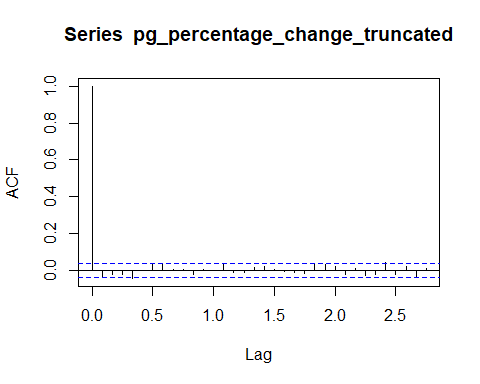
acf(c\_percentage\_change\_truncated)



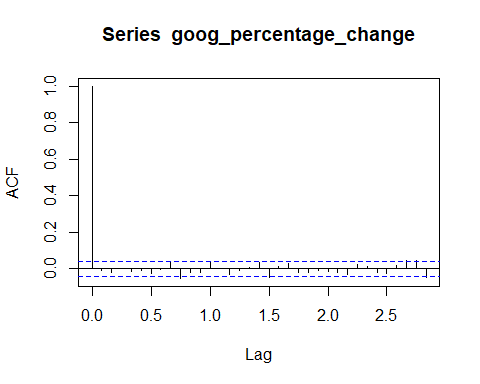
# PG  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
acf(pg\_percentage\_change)



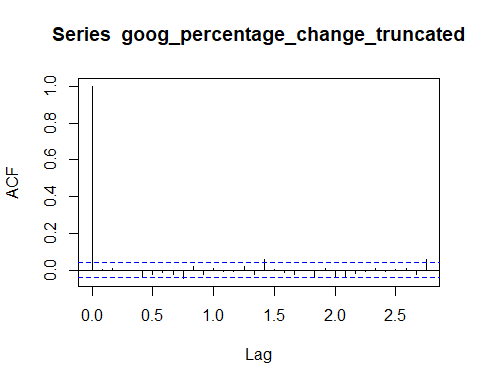
acf(pg\_percentage\_change\_truncated)



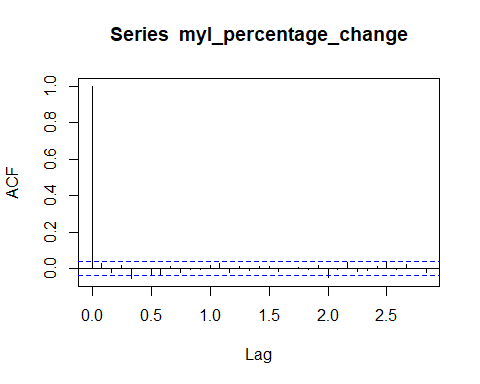
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
acf(goog\_percentage\_change)



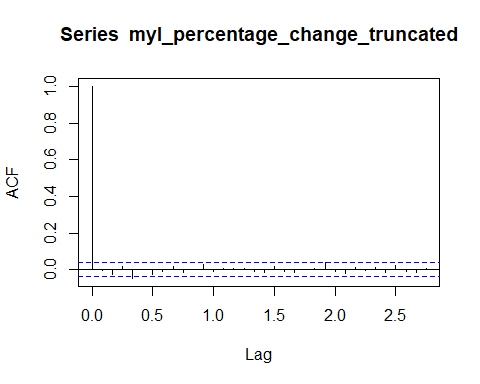
acf(goog\_percentage\_change\_truncated)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
acf(myl\_percentage\_change)



acf(myl\_percentage\_change\_truncated)



Discussion: Similarities

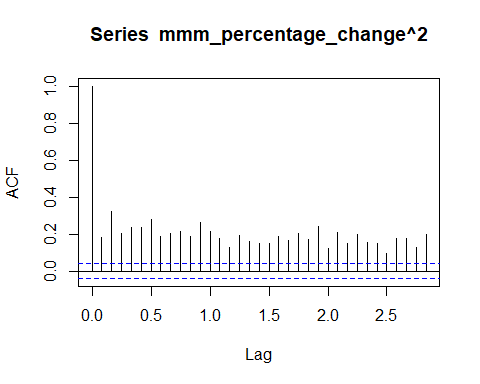
Difference - MMM (Surya)

Both full and truncated autocorrelation of 3M asset return price shows a similar result that there is a significant lag that we need to focus, especially on 0.1~0.2, 1.5, 1.9, and 2.9 because all of it was below the threshold. However, every lag is almost zero which we could interpret that there is no lag that influence the other lags that could give a significant spike.

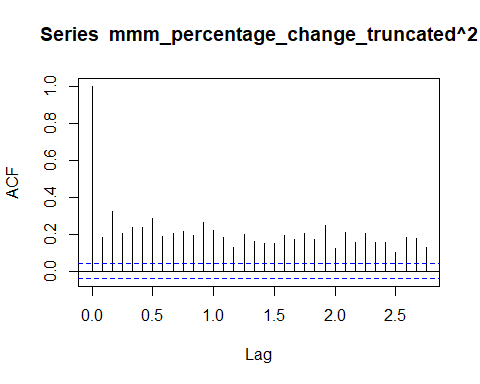
* C
* GOOG
* MYL
* PG

Stationary analysis of the square of the returns series

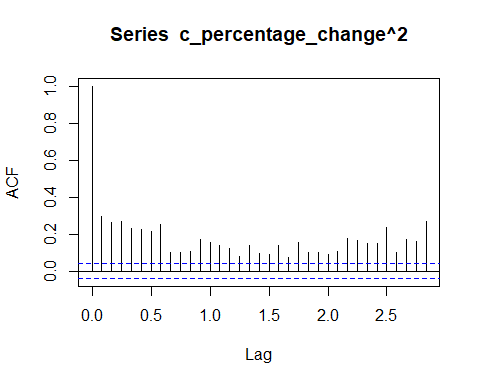
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
acf(mmm\_percentage\_change^2)



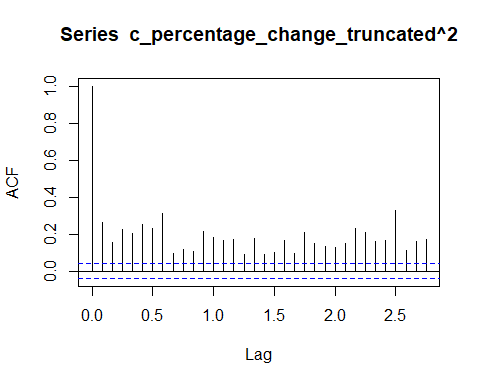
acf(mmm\_percentage\_change\_truncated^2)



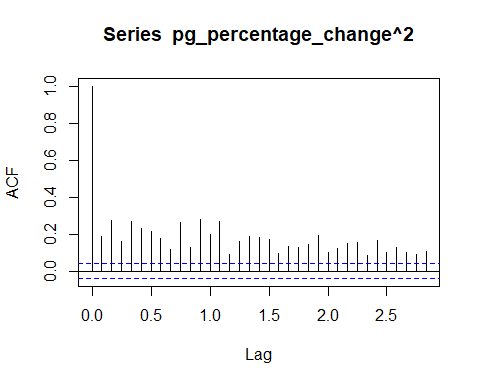
# C  
c\_percentage\_change <- ts(transformed\_data.returns$returns.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
acf(c\_percentage\_change^2)



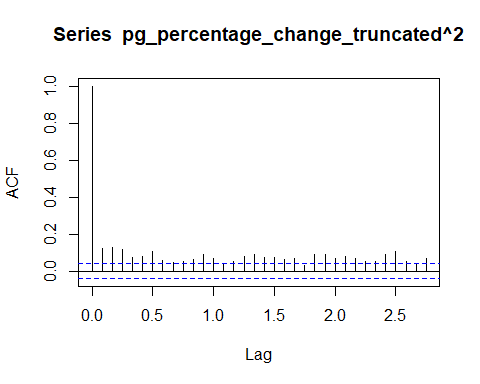
acf(c\_percentage\_change\_truncated^2)



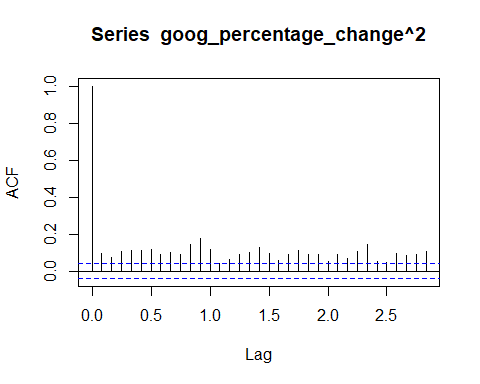
# PG  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
acf(pg\_percentage\_change^2)



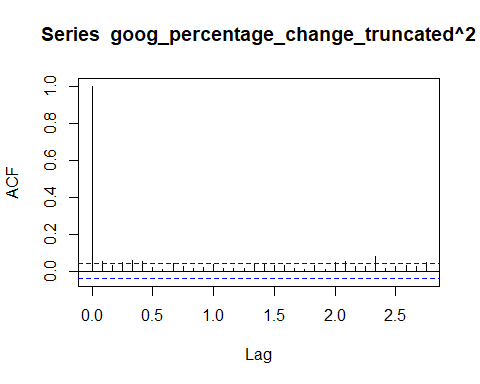
acf(pg\_percentage\_change\_truncated^2)



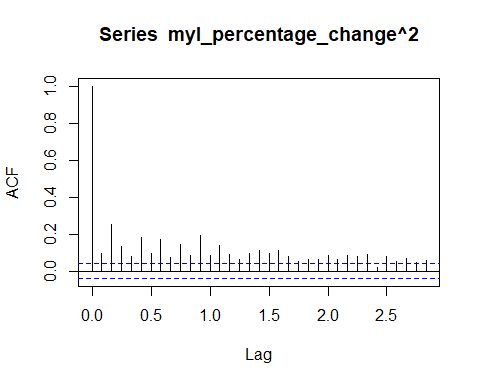
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
acf(goog\_percentage\_change^2)



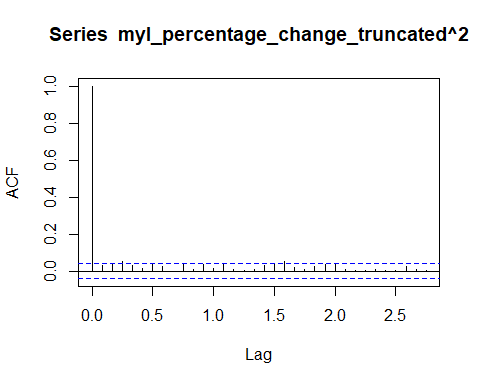
acf(goog\_percentage\_change\_truncated^2)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
acf(myl\_percentage\_change^2)



acf(myl\_percentage\_change\_truncated^2)



Discussion: Similarities

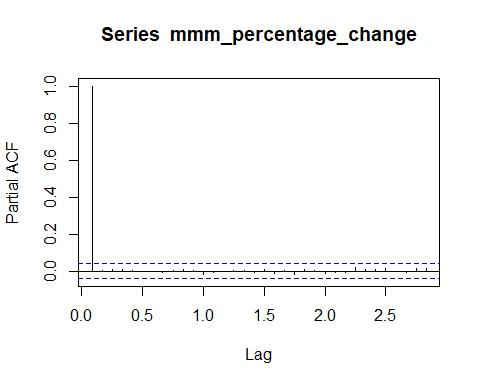
Difference - MMM (Surya)

On the squared return of 3M asset squared return price, we could see that the progress of each lag was quite stable. There is no extreme increase nor decrease. I also could concluded that MMM stock didn’t experience a significant increase or decrease of their price for quite a long time. It’s positively stable.

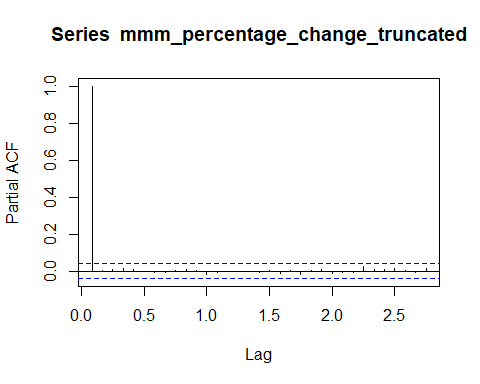
* C
* GOOG
* MYL
* PG

SECTION 2.2 - PARTIAL AUTOCORRELATION Stationary analysis of the price series

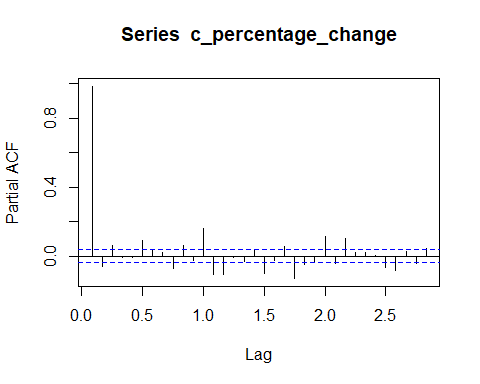
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.close$close\_price.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
pacf(mmm\_percentage\_change)



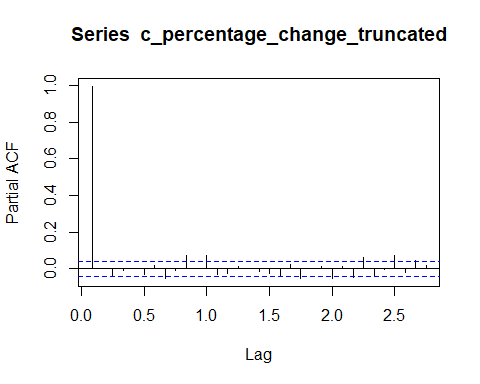
pacf(mmm\_percentage\_change\_truncated)



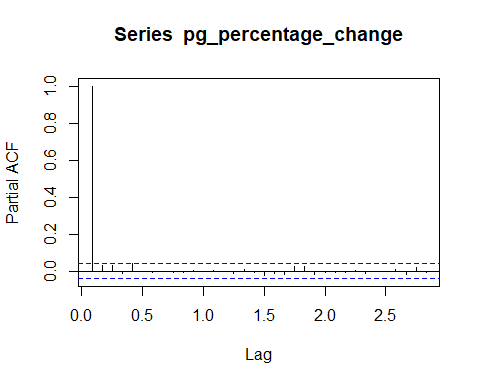
# C  
c\_percentage\_change <- ts(transformed\_data.close$close\_price.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
pacf(c\_percentage\_change)



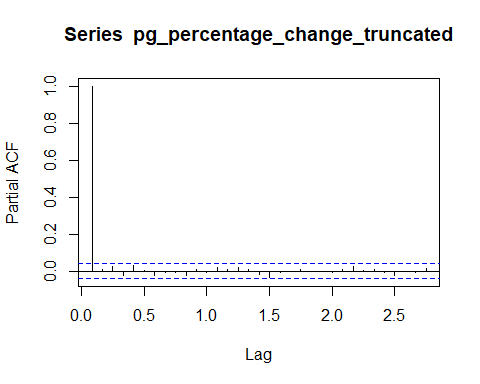
pacf(c\_percentage\_change\_truncated)



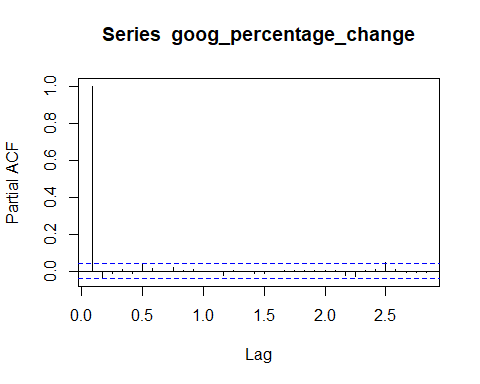
# PG  
pg\_percentage\_change <- ts(transformed\_data.close$close\_price.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
pacf(pg\_percentage\_change)



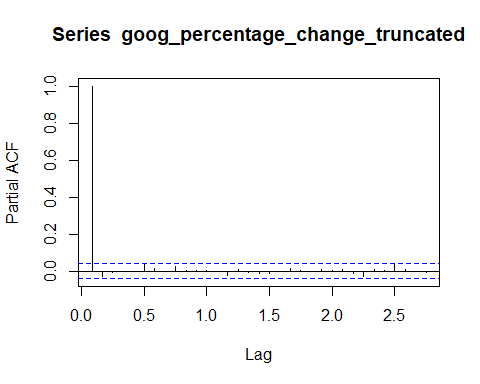
pacf(pg\_percentage\_change\_truncated)



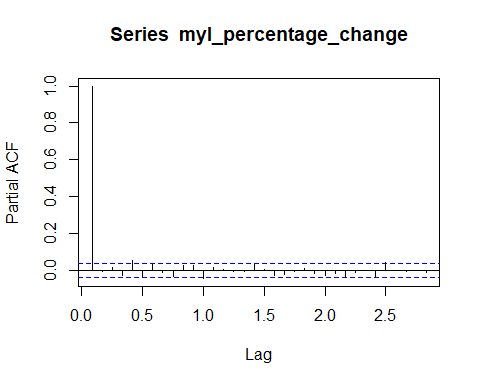
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.close$close\_price.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
pacf(goog\_percentage\_change)



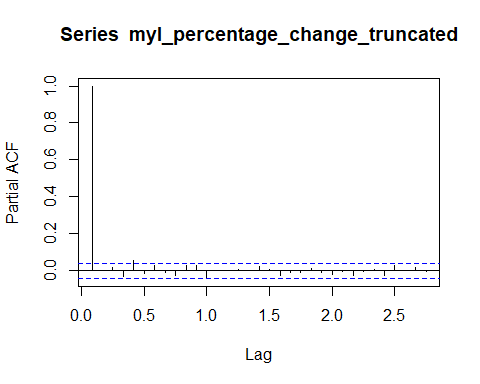
pacf(goog\_percentage\_change\_truncated)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.close$close\_price.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
pacf(myl\_percentage\_change)



pacf(myl\_percentage\_change\_truncated)



Discussion: Similarities

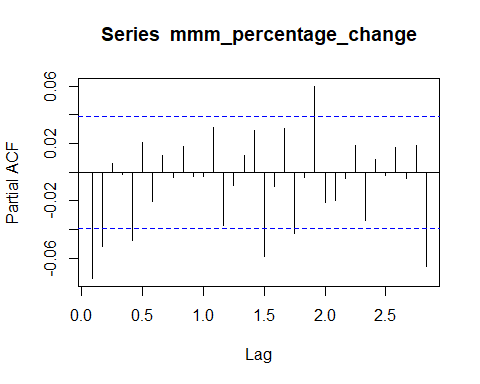
Difference - MMM (Surya)

Both full and truncated partial correlation of 3M asset close price shows that there is no lag that exceed the threshold and there is no strong impact from any lag. From here, we could say there this asset is quite stable– there is no particular lag that could influence the other lags to get a significant spike.

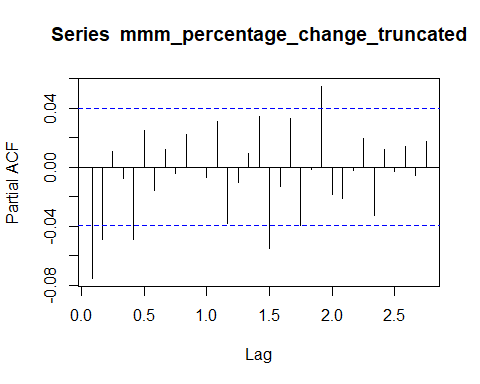
* C
* GOOG
* MYL
* PG

Stationary analysis of the returns series

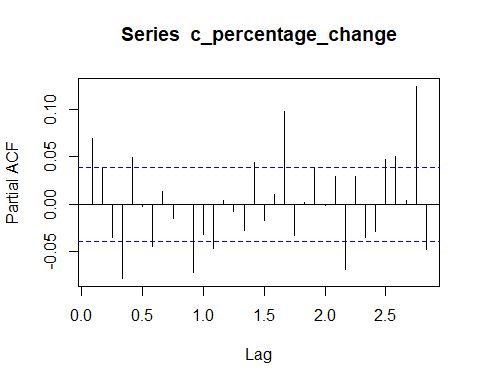
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
pacf(mmm\_percentage\_change)



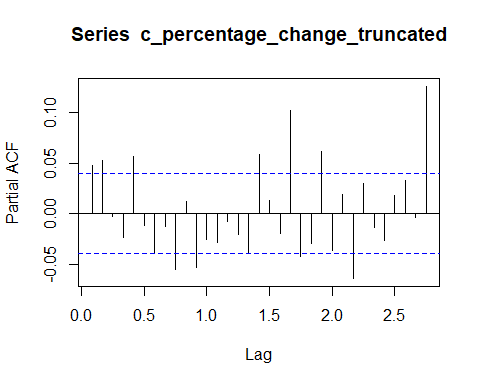
pacf(mmm\_percentage\_change\_truncated)



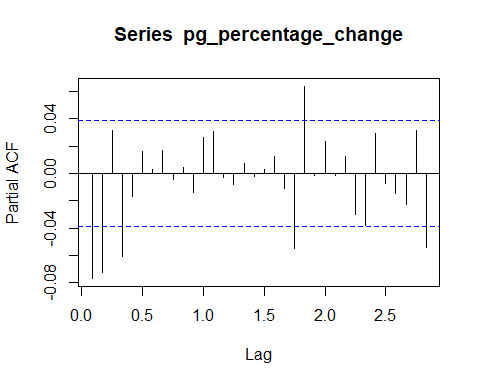
# C  
c\_percentage\_change <- ts(transformed\_data.returns$returns.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
pacf(c\_percentage\_change)



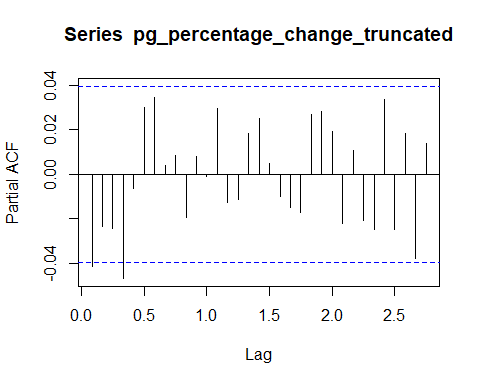
pacf(c\_percentage\_change\_truncated)



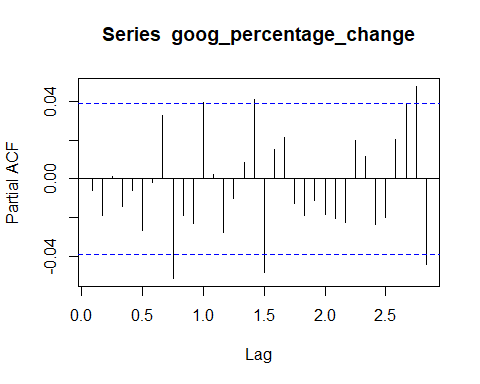
# PG  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
pacf(pg\_percentage\_change)



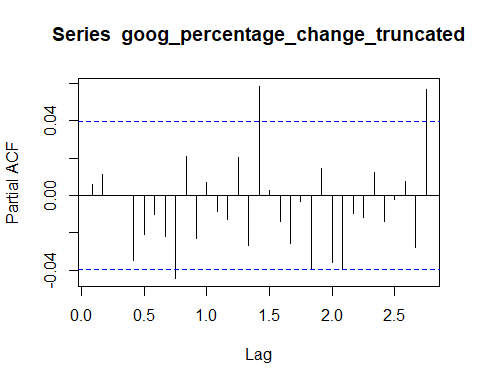
pacf(pg\_percentage\_change\_truncated)



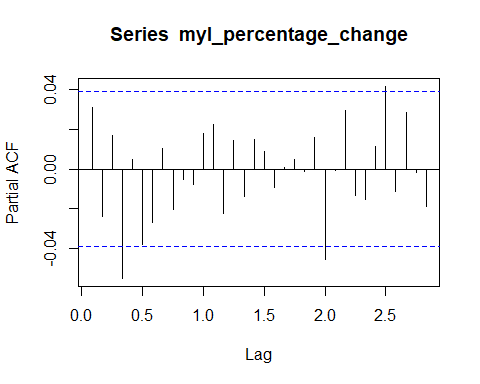
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
pacf(goog\_percentage\_change)



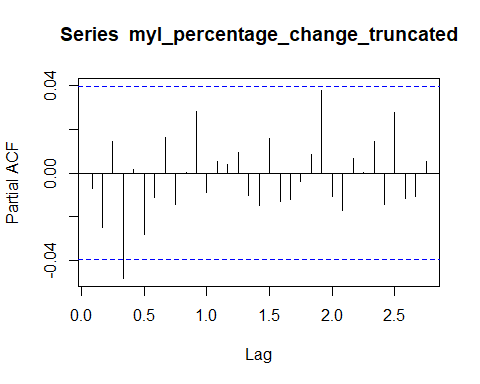
pacf(goog\_percentage\_change\_truncated)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
pacf(myl\_percentage\_change)



pacf(myl\_percentage\_change\_truncated)



Discussion: Similarities

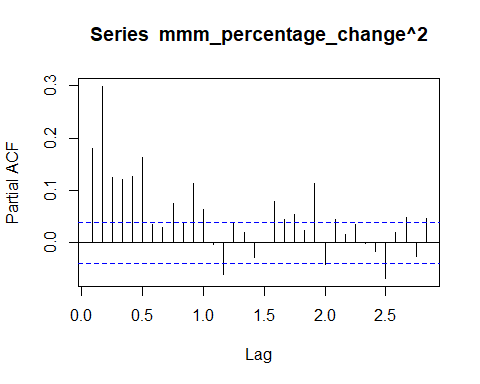
Difference - MMM (Surya)

Both full and truncated version show a similar result. Even though the graph shows a spiky lags, we need to focus on the y-range which is between -0.06~0.06. From here, we could say that all lags was almot zero barely significant.

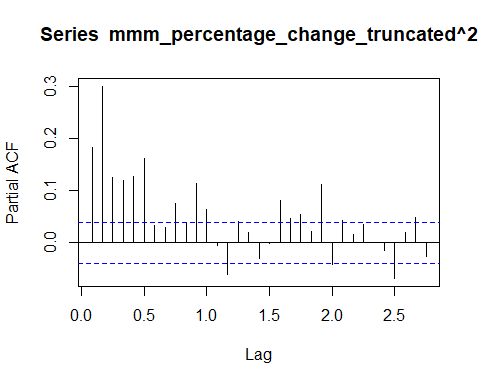
* C
* GOOG
* MYL
* PG

Stationary analysis of the square of the returns series

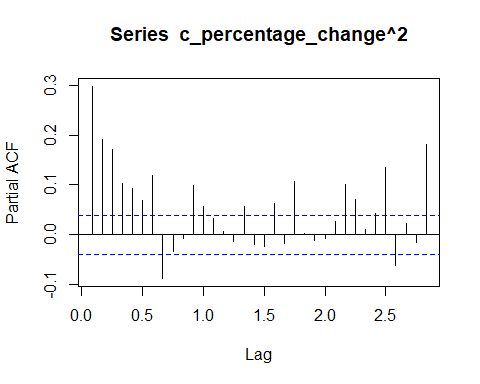
# MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM, frequency = 12, start = 2008)  
mmm\_percentage\_change\_truncated <- window(mmm\_percentage\_change, start = 2013)  
pacf(mmm\_percentage\_change^2)



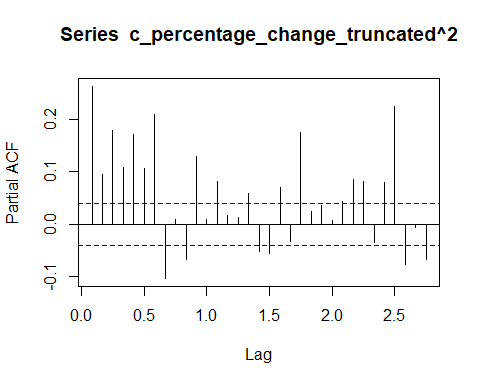
pacf(mmm\_percentage\_change\_truncated^2)



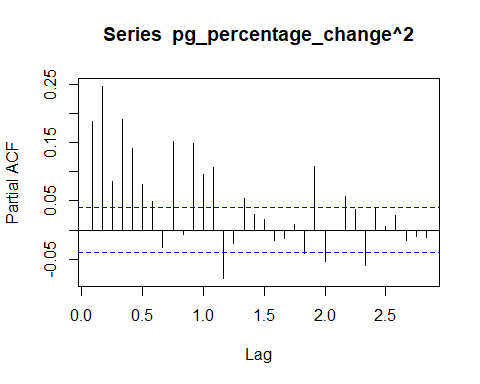
# C  
c\_percentage\_change <- ts(transformed\_data.returns$returns.C, frequency = 12, start = 2008)  
c\_percentage\_change\_truncated <- window(c\_percentage\_change, start = 2013)  
pacf(c\_percentage\_change^2)



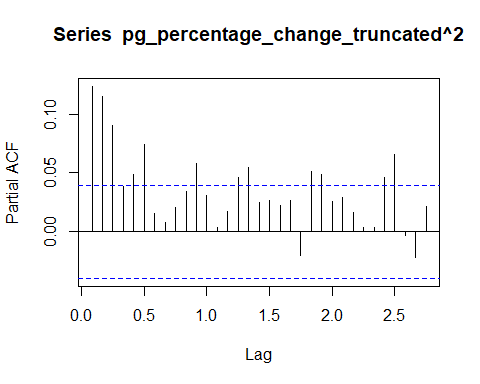
pacf(c\_percentage\_change\_truncated^2)



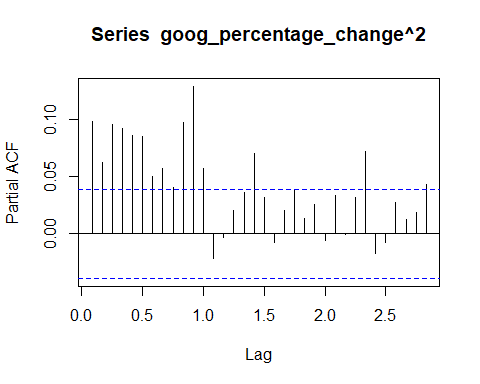
# PG  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG, frequency = 12, start = 2008)  
pg\_percentage\_change\_truncated <- window(pg\_percentage\_change, start = 2013)  
pacf(pg\_percentage\_change^2)



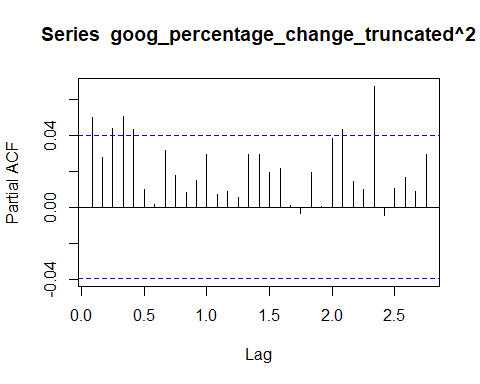
pacf(pg\_percentage\_change\_truncated^2)



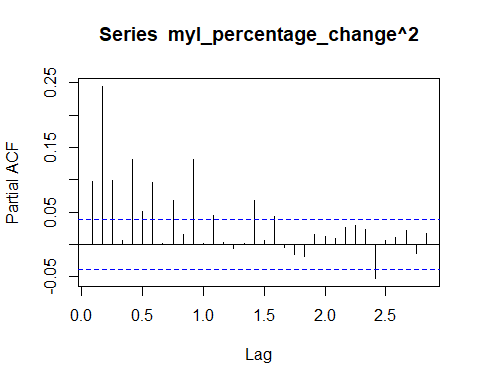
# GOOG  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG, frequency = 12, start = 2008)  
goog\_percentage\_change\_truncated <- window(goog\_percentage\_change, start = 2013)  
pacf(goog\_percentage\_change^2)



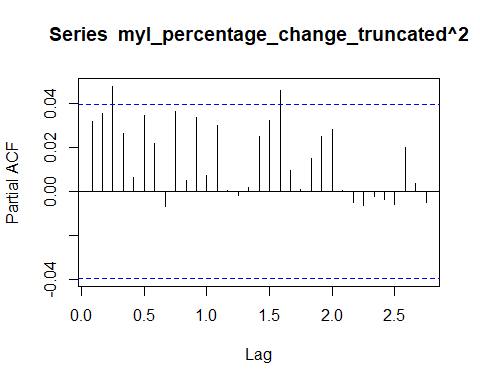
pacf(goog\_percentage\_change\_truncated^2)



# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL, frequency = 12, start = 2008)  
myl\_percentage\_change\_truncated <- window(myl\_percentage\_change, start = 2013)  
pacf(myl\_percentage\_change^2)



pacf(myl\_percentage\_change\_truncated^2)



Discussion: Similarities

Difference - MMM (Surya)

Both full and truncated version shows a similar result. There are strong lags on the quarter half of the half which suddenly decrease and become stable after that.

* C
* GOOG
* MYL
* PG

SECTION 2.3 - ARIMA MODEL Stationary analysis of the returns series

# MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM, frequency = 12, start = 2008)  
mmm\_arma\_model <- arma(window(mmm\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(mmm\_arma\_model)

##   
## Call:  
## arma(x = window(mmm\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0324426 -0.0075725 -0.0009894 0.0069365 0.0357028   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.0145113 0.0898988 0.161 0.8718   
## ar2 -0.0647369 0.0899230 -0.720 0.4716   
## ar3 -0.0979299 0.0907766 -1.079 0.2807   
## ar4 0.1297410 0.0949019 1.367 0.1716   
## ar5 -0.1549885 0.0950373 -1.631 0.1029   
## ar6 -0.0614024 0.0944583 -0.650 0.5157   
## ar7 -0.0450565 0.0941122 -0.479 0.6321   
## ar8 -0.0189136 0.0919360 -0.206 0.8370   
## ar9 0.0655022 0.0919659 0.712 0.4763   
## ar10 -0.0285652 0.0889823 -0.321 0.7482   
## ar11 0.0119657 0.0889045 0.135 0.8929   
## ar12 0.1893739 0.0891274 2.125 0.0336 \*  
## intercept -0.0005619 0.0011256 -0.499 0.6176   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.0001506, Conditional Sum-of-Squares = 0.02, AIC = -695.53

Box.test(mmm\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: mmm\_arma\_model$residuals  
## X-squared = 2.1304, df = 12, p-value = 0.9992

# Citi  
citi\_percentage\_change <- ts(transformed\_data.returns$returns.C, frequency = 12, start = 2008)  
citi\_arma\_model <- arma(window(citi\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(citi\_arma\_model)

##   
## Call:  
## arma(x = window(citi\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.302880 -0.051461 0.001419 0.059792 0.454817   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.0156695 0.0898282 0.174 0.8615   
## ar2 -0.0077968 0.0901904 -0.086 0.9311   
## ar3 -0.0394521 0.0902975 -0.437 0.6622   
## ar4 -0.2119590 0.0915374 -2.316 0.0206 \*  
## ar5 0.0006166 0.0943380 0.007 0.9948   
## ar6 -0.1011711 0.0941963 -1.074 0.2828   
## ar7 -0.0198987 0.0949200 -0.210 0.8340   
## ar8 -0.0292726 0.0947978 -0.309 0.7575   
## ar9 -0.0961146 0.0920927 -1.044 0.2966   
## ar10 -0.0013918 0.0924316 -0.015 0.9880   
## ar11 -0.1133014 0.0933195 -1.214 0.2247   
## ar12 -0.0355866 0.0938663 -0.379 0.7046   
## intercept -0.0279216 0.0120132 -2.324 0.0201 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.01272, Conditional Sum-of-Squares = 1.37, AIC = -158.68

Box.test(citi\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: citi\_arma\_model$residuals  
## X-squared = 0.8854, df = 12, p-value = 1

# P & G  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG, frequency = 12, start = 2008)  
pg\_arma\_model <- arma(window(pg\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(pg\_arma\_model)

##   
## Call:  
## arma(x = window(pg\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.166e-02 -1.437e-02 -5.889e-05 1.330e-02 6.798e-02   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 -0.127964 0.089143 -1.435 0.1511   
## ar2 -0.351302 0.089388 -3.930 8.49e-05 \*\*\*  
## ar3 0.129405 0.094875 1.364 0.1726   
## ar4 -0.206723 0.093947 -2.200 0.0278 \*   
## ar5 -0.062551 0.095569 -0.655 0.5128   
## ar6 -0.052074 0.095180 -0.547 0.5843   
## ar7 -0.110901 0.095170 -1.165 0.2439   
## ar8 0.075547 0.095569 0.791 0.4292   
## ar9 -0.178967 0.093333 -1.918 0.0552 .   
## ar10 0.124352 0.093957 1.324 0.1857   
## ar11 -0.110504 0.089892 -1.229 0.2190   
## ar12 0.197165 0.090001 2.191 0.0285 \*   
## intercept -0.005439 0.002664 -2.041 0.0412 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.0005901, Conditional Sum-of-Squares = 0.06, AIC = -530.28

Box.test(pg\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: pg\_arma\_model$residuals  
## X-squared = 1.0749, df = 12, p-value = 1

# Google  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG, frequency = 12, start = 2008)  
goog\_arma\_model <- arma(window(goog\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(goog\_arma\_model)

##   
## Call:  
## arma(x = window(goog\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.1113916 -0.0244762 -0.0002017 0.0257513 0.1201712   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 -0.074930 0.088791 -0.844 0.3987   
## ar2 -0.065617 0.088548 -0.741 0.4587   
## ar3 0.019455 0.088022 0.221 0.8251   
## ar4 -0.060231 0.087361 -0.689 0.4905   
## ar5 0.002706 0.086499 0.031 0.9750   
## ar6 -0.031473 0.086231 -0.365 0.7151   
## ar7 -0.003360 0.086741 -0.039 0.9691   
## ar8 0.218926 0.086677 2.526 0.0115 \*  
## ar9 -0.148542 0.087044 -1.707 0.0879 .  
## ar10 -0.111019 0.088068 -1.261 0.2074   
## ar11 -0.056751 0.088459 -0.642 0.5212   
## ar12 0.170000 0.088538 1.920 0.0548 .  
## intercept -0.002884 0.003564 -0.809 0.4184   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.001489, Conditional Sum-of-Squares = 0.16, AIC = -418.27

Box.test(goog\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: goog\_arma\_model$residuals  
## X-squared = 2.1882, df = 12, p-value = 0.9991

# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL, frequency = 12, start = 2008)  
myl\_arma\_model <- arma(window(myl\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(myl\_arma\_model)

##   
## Call:  
## arma(x = window(myl\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.159218 -0.028373 -0.000815 0.025062 0.192793   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.1925776 0.0889799 2.164 0.0304 \*  
## ar2 -0.1059335 0.0888914 -1.192 0.2334   
## ar3 0.1173492 0.0892931 1.314 0.1888   
## ar4 -0.1351307 0.0901116 -1.500 0.1337   
## ar5 0.0218342 0.0908577 0.240 0.8101   
## ar6 -0.0443907 0.0906198 -0.490 0.6242   
## ar7 -0.0597240 0.0904172 -0.661 0.5089   
## ar8 0.0335063 0.0883905 0.379 0.7046   
## ar9 -0.0564381 0.0877390 -0.643 0.5201   
## ar10 0.0578230 0.0876944 0.659 0.5097   
## ar11 -0.1872436 0.0876691 -2.136 0.0327 \*  
## ar12 0.1818918 0.0879739 2.068 0.0387 \*  
## intercept 0.0007502 0.0045092 0.166 0.8679   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.002476, Conditional Sum-of-Squares = 0.27, AIC = -356.74

Box.test(myl\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: myl\_arma\_model$residuals  
## X-squared = 1.369, df = 12, p-value = 0.9999

Discussion: - MMM (Surya)

Stationary analysis of the square of the returns series

#MMM  
mmm\_percentage\_change <- ts(transformed\_data.returns$returns.MMM^2, frequency = 12, start = 2008)  
mmm\_arma\_model <- arma(window(mmm\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(mmm\_arma\_model)

##   
## Call:  
## arma(x = window(mmm\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.593e-04 -1.375e-04 -8.267e-05 4.313e-05 1.031e-03   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 -6.480e-02 9.049e-02 -0.716 0.473921   
## ar2 -3.830e-02 9.086e-02 -0.422 0.673344   
## ar3 -9.044e-02 9.082e-02 -0.996 0.319337   
## ar4 -9.063e-02 9.658e-02 -0.938 0.348027   
## ar5 5.976e-03 9.675e-02 0.062 0.950752   
## ar6 7.714e-02 9.170e-02 0.841 0.400204   
## ar7 -1.111e-02 9.199e-02 -0.121 0.903903   
## ar8 -4.366e-02 9.200e-02 -0.475 0.635102   
## ar9 3.953e-02 9.232e-02 0.428 0.668553   
## ar10 -4.680e-02 8.613e-02 -0.543 0.586921   
## ar11 -4.395e-02 8.635e-02 -0.509 0.610801   
## ar12 -6.493e-02 8.566e-02 -0.758 0.448412   
## intercept 2.317e-04 6.476e-05 3.577 0.000347 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 5.772e-08, Conditional Sum-of-Squares = 0, AIC = -1647.4

Box.test(mmm\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: mmm\_arma\_model$residuals  
## X-squared = 1.5962, df = 12, p-value = 0.9998

# Citi  
citi\_percentage\_change <- ts(transformed\_data.returns$returns.C^2, frequency = 12, start = 2008)  
citi\_arma\_model <- arma(window(citi\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(citi\_arma\_model)

##   
## Call:  
## arma(x = window(citi\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.061610 -0.009673 -0.005568 0.003683 0.161272   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.141974 0.091097 1.558 0.11911   
## ar2 0.260726 0.092265 2.826 0.00472 \*\*  
## ar3 0.099952 0.094951 1.053 0.29249   
## ar4 0.020302 0.096839 0.210 0.83394   
## ar5 -0.052762 0.096715 -0.546 0.58538   
## ar6 0.017602 0.096990 0.181 0.85598   
## ar7 -0.013319 0.096959 -0.137 0.89074   
## ar8 -0.061361 0.097014 -0.632 0.52706   
## ar9 -0.043283 0.097219 -0.445 0.65617   
## ar10 -0.035301 0.096787 -0.365 0.71532   
## ar11 -0.007012 0.094471 -0.074 0.94083   
## ar12 -0.014828 0.093232 -0.159 0.87364   
## intercept 0.009528 0.003488 2.731 0.00631 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 0.0005946, Conditional Sum-of-Squares = 0.06, AIC = -529.36

Box.test(citi\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: citi\_arma\_model$residuals  
## X-squared = 0.1968, df = 12, p-value = 1

# P & G  
pg\_percentage\_change <- ts(transformed\_data.returns$returns.PG^2, frequency = 12, start = 2008)  
pg\_arma\_model <- arma(window(pg\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(pg\_arma\_model)

##   
## Call:  
## arma(x = window(pg\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0020200 -0.0005957 -0.0002668 0.0002317 0.0070782   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.0217533 0.0902252 0.241 0.8095   
## ar2 0.1118982 0.0888761 1.259 0.2080   
## ar3 -0.0304556 0.0890249 -0.342 0.7323   
## ar4 0.1849274 0.0882302 2.096 0.0361 \*  
## ar5 0.0990457 0.0891763 1.111 0.2667   
## ar6 0.0520752 0.0895548 0.581 0.5609   
## ar7 -0.0401423 0.0895233 -0.448 0.6539   
## ar8 -0.1175043 0.0891432 -1.318 0.1875   
## ar9 0.1309813 0.0882602 1.484 0.1378   
## ar10 -0.0988277 0.0890254 -1.110 0.2670   
## ar11 0.1730075 0.0888389 1.947 0.0515 .  
## ar12 0.1251068 0.0901806 1.387 0.1654   
## intercept 0.0003062 0.0001920 1.595 0.1108   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 1.673e-06, Conditional Sum-of-Squares = 0, AIC = -1240

Box.test(pg\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: pg\_arma\_model$residuals  
## X-squared = 0.6873, df = 12, p-value = 1

# Google  
goog\_percentage\_change <- ts(transformed\_data.returns$returns.GOOG^2, frequency = 12, start = 2008)  
goog\_arma\_model <- arma(window(goog\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(goog\_arma\_model)

##   
## Call:  
## arma(x = window(goog\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0038033 -0.0014094 -0.0006714 0.0004642 0.0151333   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 -0.1357382 0.0897207 -1.513 0.130306   
## ar2 -0.0879018 0.0851736 -1.032 0.302057   
## ar3 0.0031792 0.0838486 0.038 0.969755   
## ar4 -0.0047535 0.0838111 -0.057 0.954770   
## ar5 0.0075570 0.0837100 0.090 0.928068   
## ar6 0.0836445 0.0836991 0.999 0.317627   
## ar7 0.0251652 0.0836267 0.301 0.763473   
## ar8 0.0112041 0.0836075 0.134 0.893396   
## ar9 0.0356526 0.0836383 0.426 0.669911   
## ar10 0.1919197 0.0837104 2.293 0.021868 \*   
## ar11 0.3304719 0.0851134 3.883 0.000103 \*\*\*  
## ar12 0.1374415 0.0890351 1.544 0.122667   
## intercept 0.0006787 0.0005394 1.258 0.208302   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 7.273e-06, Conditional Sum-of-Squares = 0, AIC = -1062.21

Box.test(goog\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: goog\_arma\_model$residuals  
## X-squared = 1.5261, df = 12, p-value = 0.9999

# MYL  
myl\_percentage\_change <- ts(transformed\_data.returns$returns.MYL^2, frequency = 12, start = 2008)  
myl\_arma\_model <- arma(window(myl\_percentage\_change, start = 2008, end = 2018), order = c(12,0,0) )  
summary(myl\_arma\_model)

##   
## Call:  
## arma(x = window(myl\_percentage\_change, start = 2008, end = 2018), order = c(12, 0, 0))  
##   
## Model:  
## ARMA(12,0)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0099871 -0.0018227 -0.0007111 0.0008296 0.0300078   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## ar1 0.0681011 0.0904834 0.753 0.451668   
## ar2 0.4851143 0.0878311 5.523 3.33e-08 \*\*\*  
## ar3 -0.0065975 0.0979222 -0.067 0.946283   
## ar4 -0.3243027 0.0983586 -3.297 0.000977 \*\*\*  
## ar5 0.2407870 0.1024121 2.351 0.018715 \*   
## ar6 0.1178138 0.1031777 1.142 0.253515   
## ar7 0.0473189 0.1035042 0.457 0.647550   
## ar8 -0.1050498 0.0981495 -1.070 0.284483   
## ar9 -0.0068142 0.0953029 -0.072 0.943000   
## ar10 0.0199553 0.0943782 0.211 0.832544   
## ar11 0.2446159 0.0855662 2.859 0.004253 \*\*   
## ar12 -0.0803309 0.0882804 -0.910 0.362848   
## intercept 0.0007981 0.0006148 1.298 0.194239   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Fit:  
## sigma^2 estimated as 2.254e-05, Conditional Sum-of-Squares = 0, AIC = -925.36

Box.test(myl\_arma\_model$residuals, lag = 12)

##   
## Box-Pierce test  
##   
## data: myl\_arma\_model$residuals  
## X-squared = 2.7062, df = 12, p-value = 0.9973

Discussion: - MMM (Surya)