The downloaded data from yahoo is of range from **2017-09-21** to **2018-09-21.** The stocks used are **'C'**,**'JPM'**,**'GOOGL'**,**'MCO'**,**'NFLX'.**

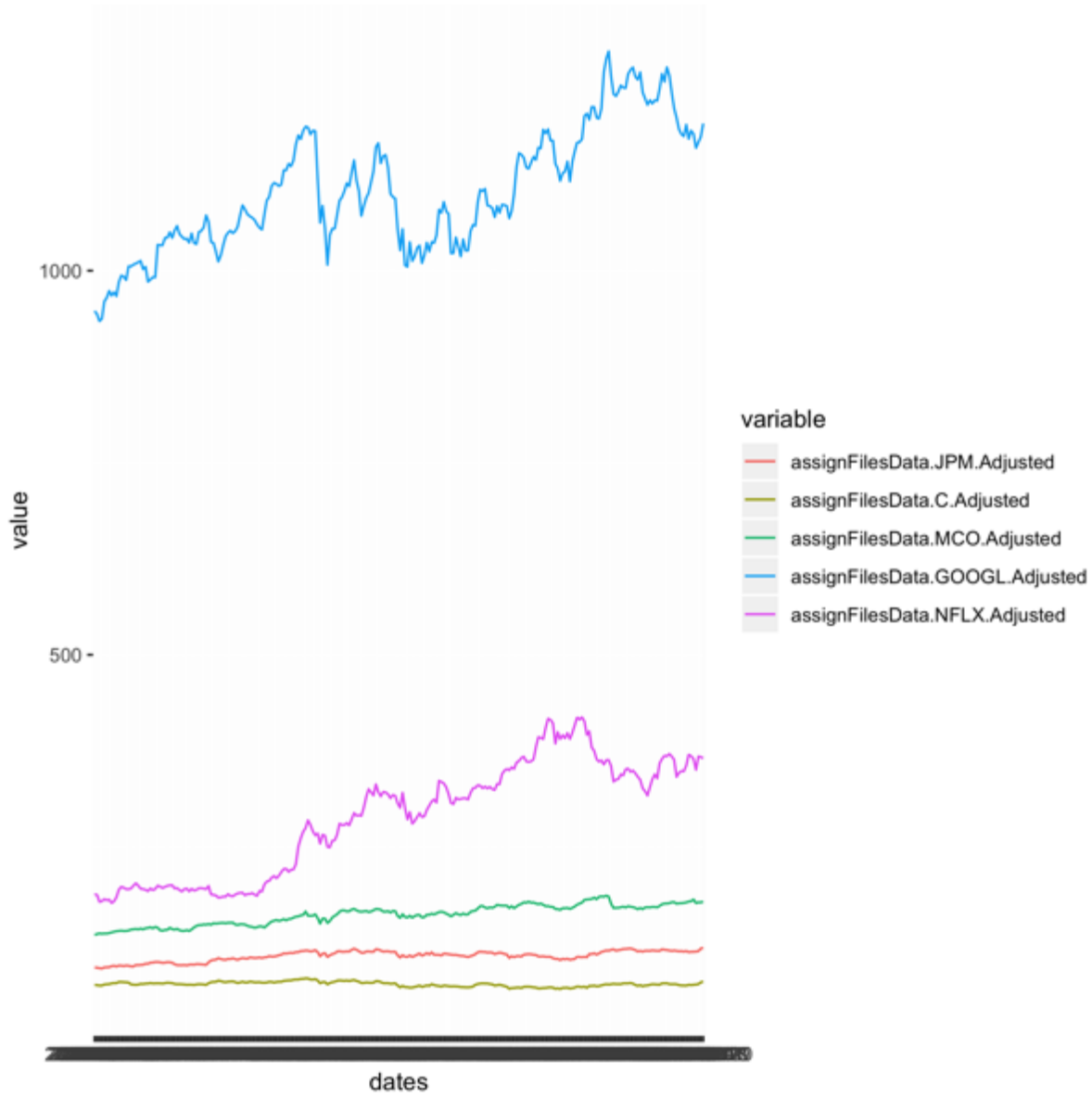
The snapshot of data is –

Date,JPM.Adjusted,C.Adjusted,MCO.Adjusted,GOOGL.Adjusted,NFLX.Adjusted
21/9/17,93.008469,70.385216,135.240707,947.549988,188.779999
22/9/17,92.812729,70.032112,136.606491,943.26001,187.350006
25/9/17,92.117836,69.482834,137.34874,934.280029,178.550003
26/9/17,91.706757,69.580917,136.923187,937.429993,179.380005

The return calculated as per formula and snapshot is as follows

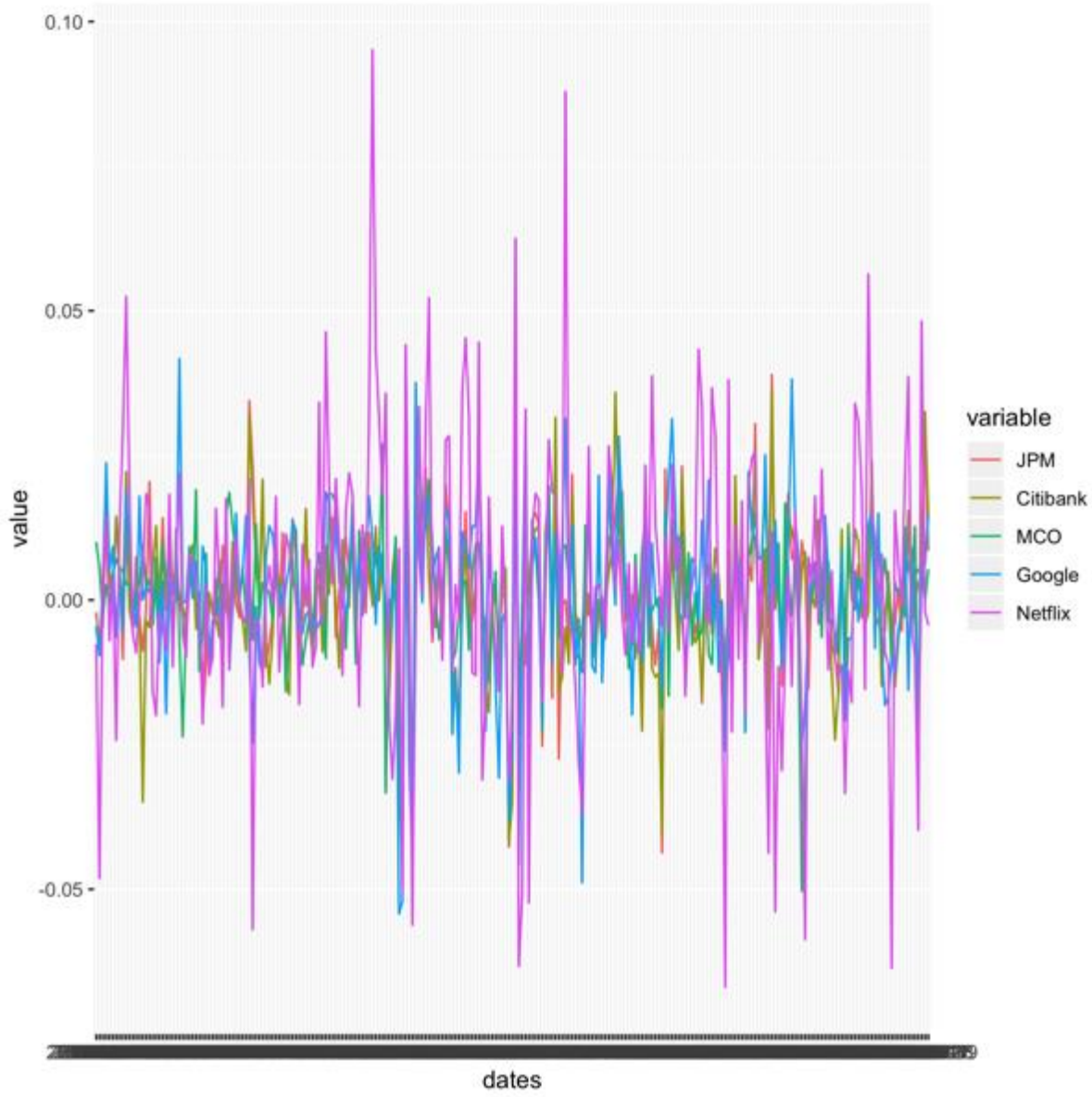
	dates	JPM	Citibank	MCO	Google
1	2017-09-21	-2.106757e-03	-0.0050293614	0.0100482587	-4.537722e-03
2	2017-09-22	-7.515213e-03	-0.0078741505	0.0054187745	-9.565761e-03
3	2017-09-25	-4.472521e-03	0.0014106195	-0.0031031491	3.365871e-03
4	2017-09-26	1.567173e-02	0.0187130586	0.0025987581	2.368706e-02
5	2017-09-27	2.099018e-03	0.0051059737	-0.0007213001	5.102051e-03
6	2017-09-28	1.362131e-03	0.0012380107	0.0043194582	9.192570e-03
7	2017-09-29	1.382914e-02	0.0144673285	0.0071577001	-6.439372e-03
8	2017-10-02	5.252463e-03	0.0044614695	0.0056186440	4.753736e-03
9	2017-10-03	-1.022139e-02	-0.0009447067	0.0050226417	-5.467131e-03
10	2017-10-04	1.337559e-02	0.0221668398	0.0027484838	1.886353e-02

Time Plot for Stock prices

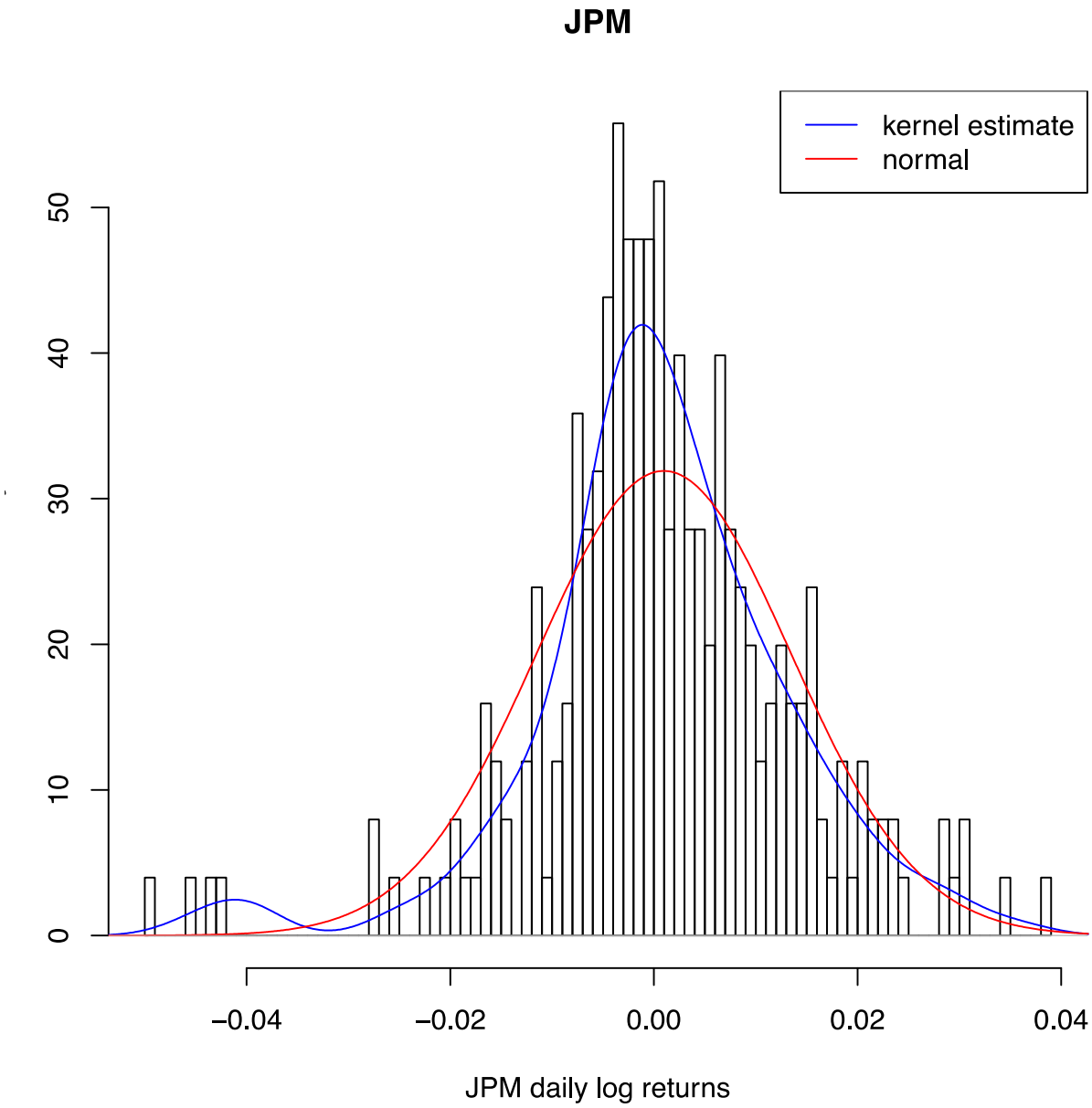


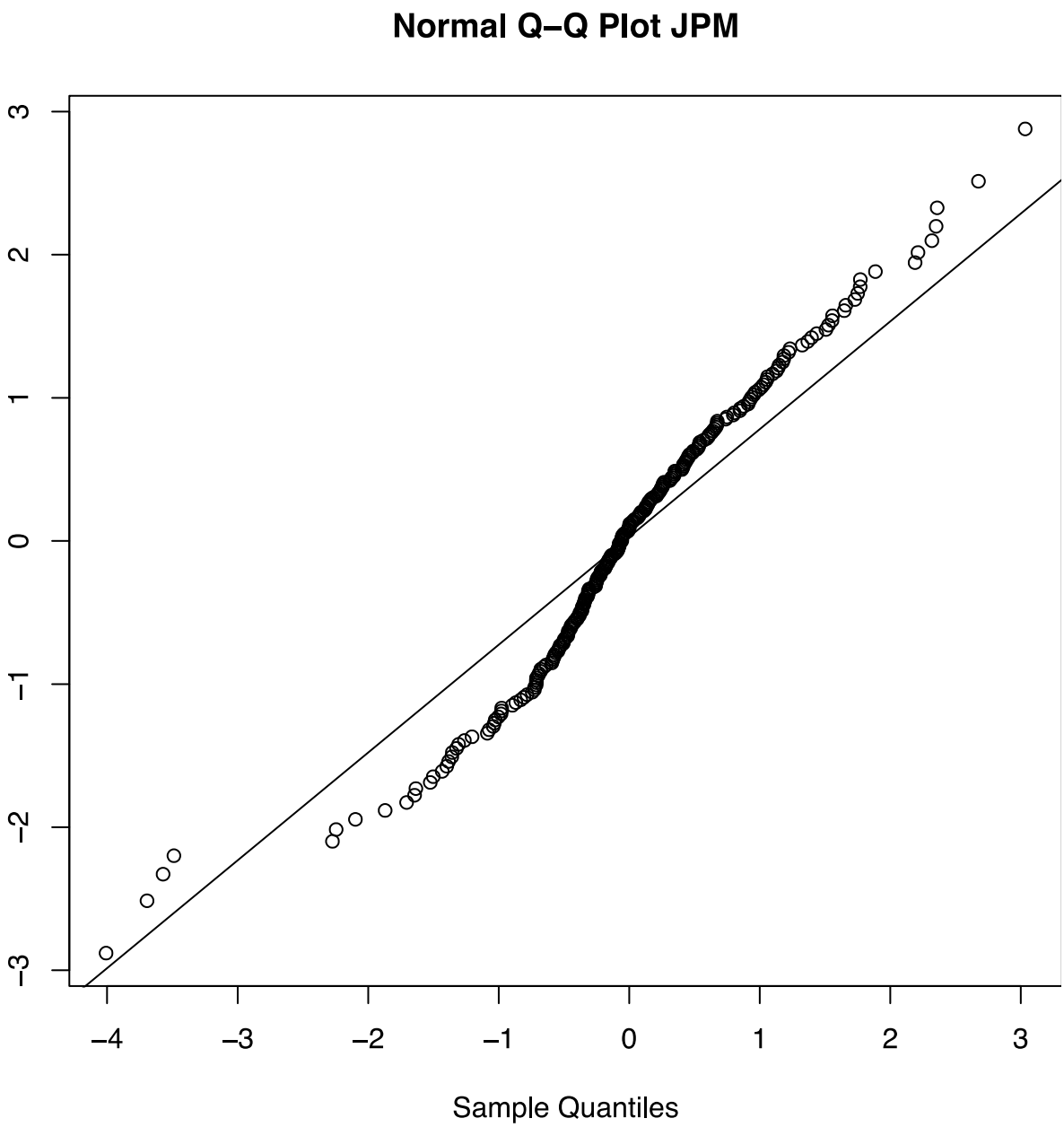
It is evident from the graph google and Netflix are more volatile and have more spikes. On the other hand J P Morgan, MCO and Citibank are stable and almost constant in the time with very less turbulence.

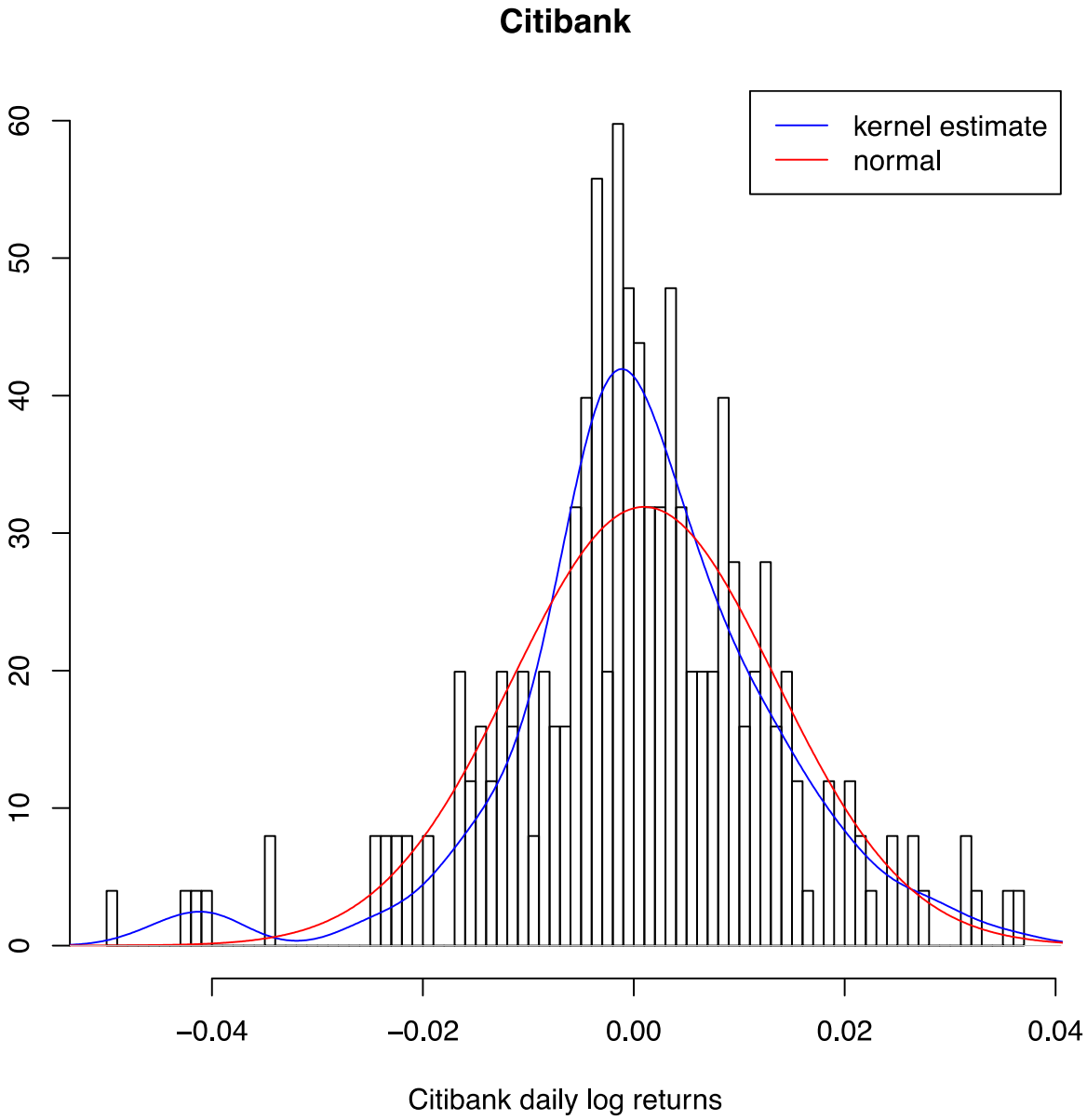
Time Plot for Stock returns

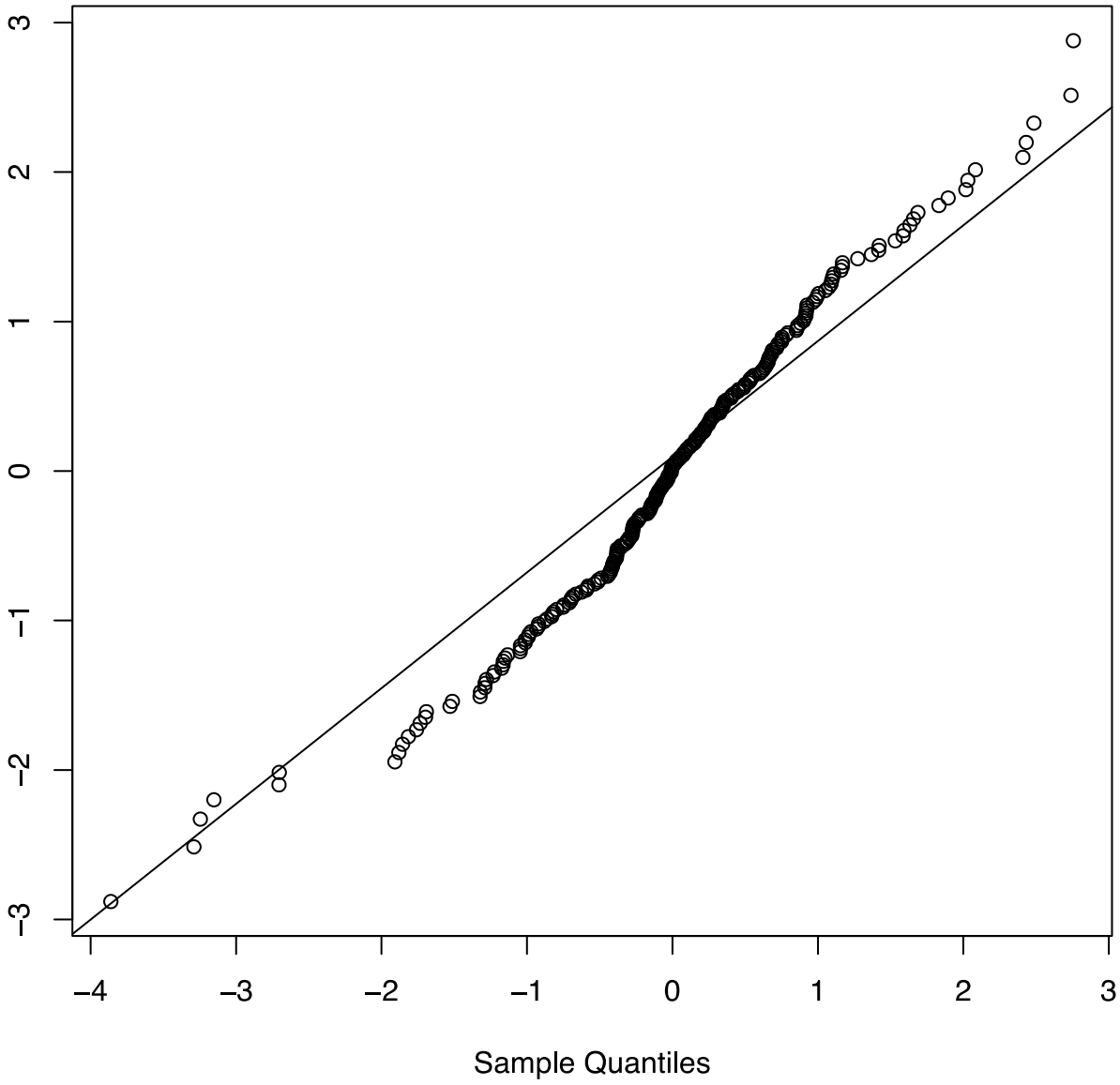


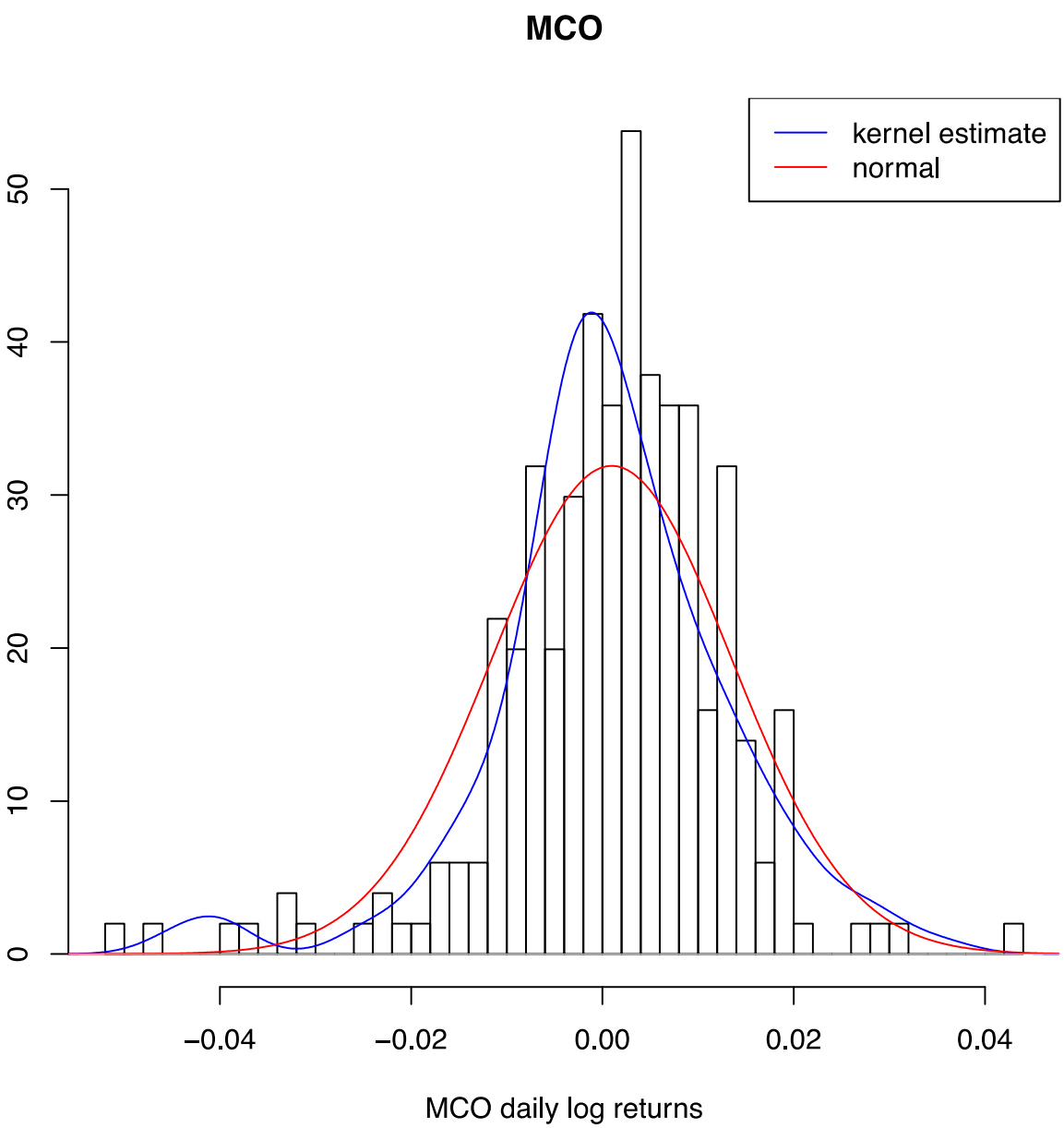
KDensity and Q-Q plot for all stocks

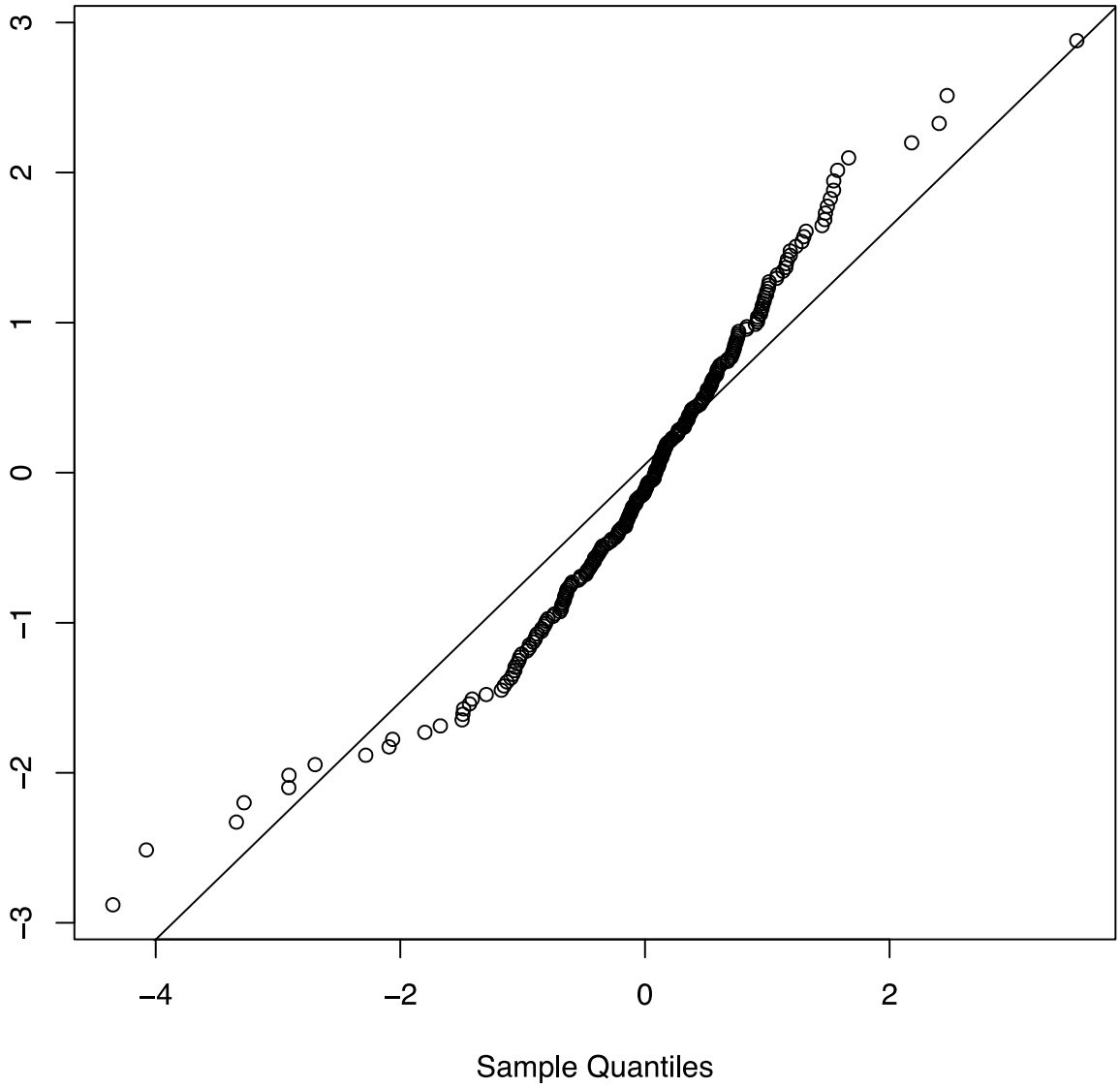

JPM
kernel estimate
normal
0
10
20
30
40
50
−0.04
−0.02
0.00
0.02
0.04
JPM daily log returns

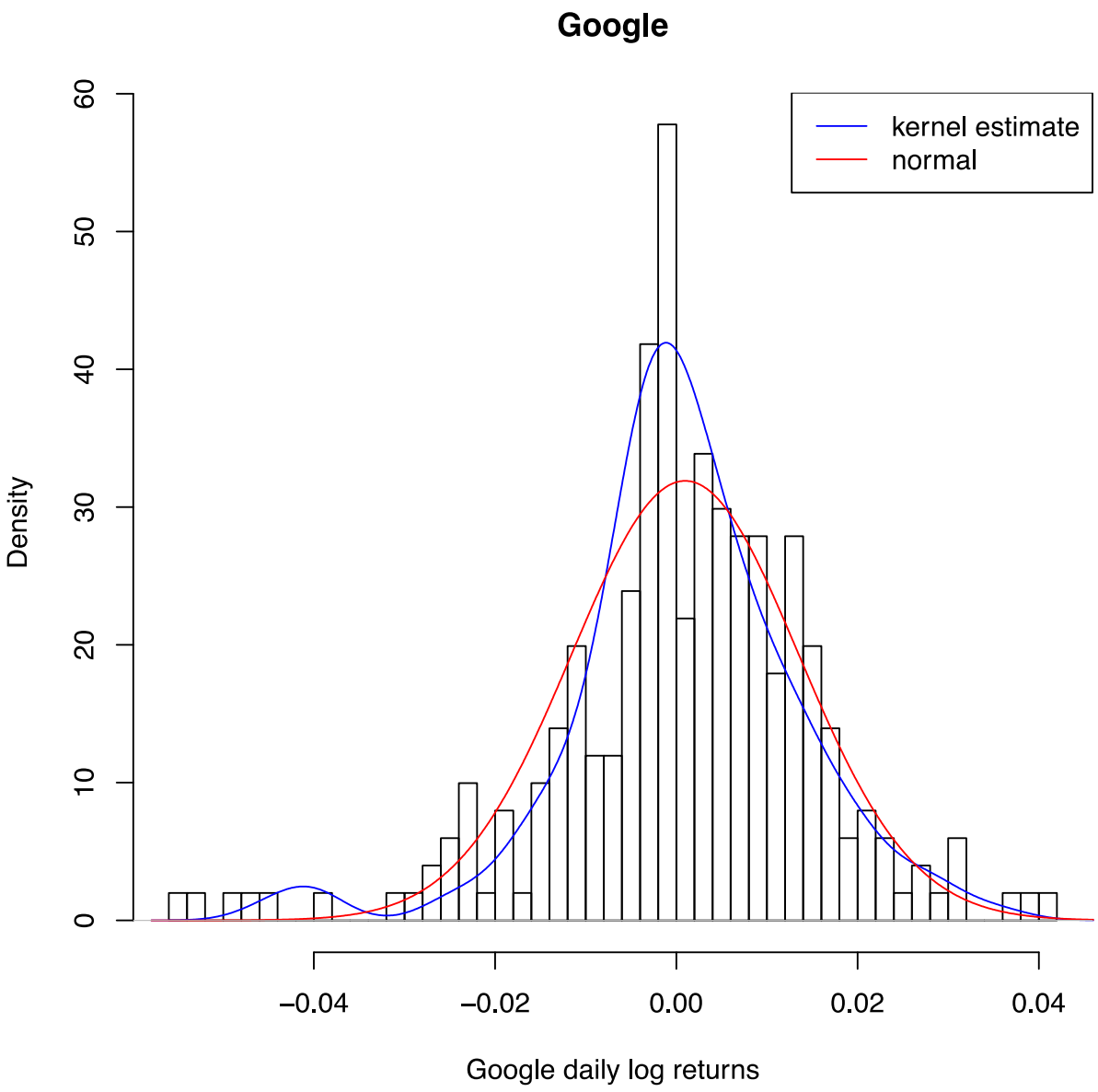

Normal Q–Q Plot JPM
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
Sample Quantiles

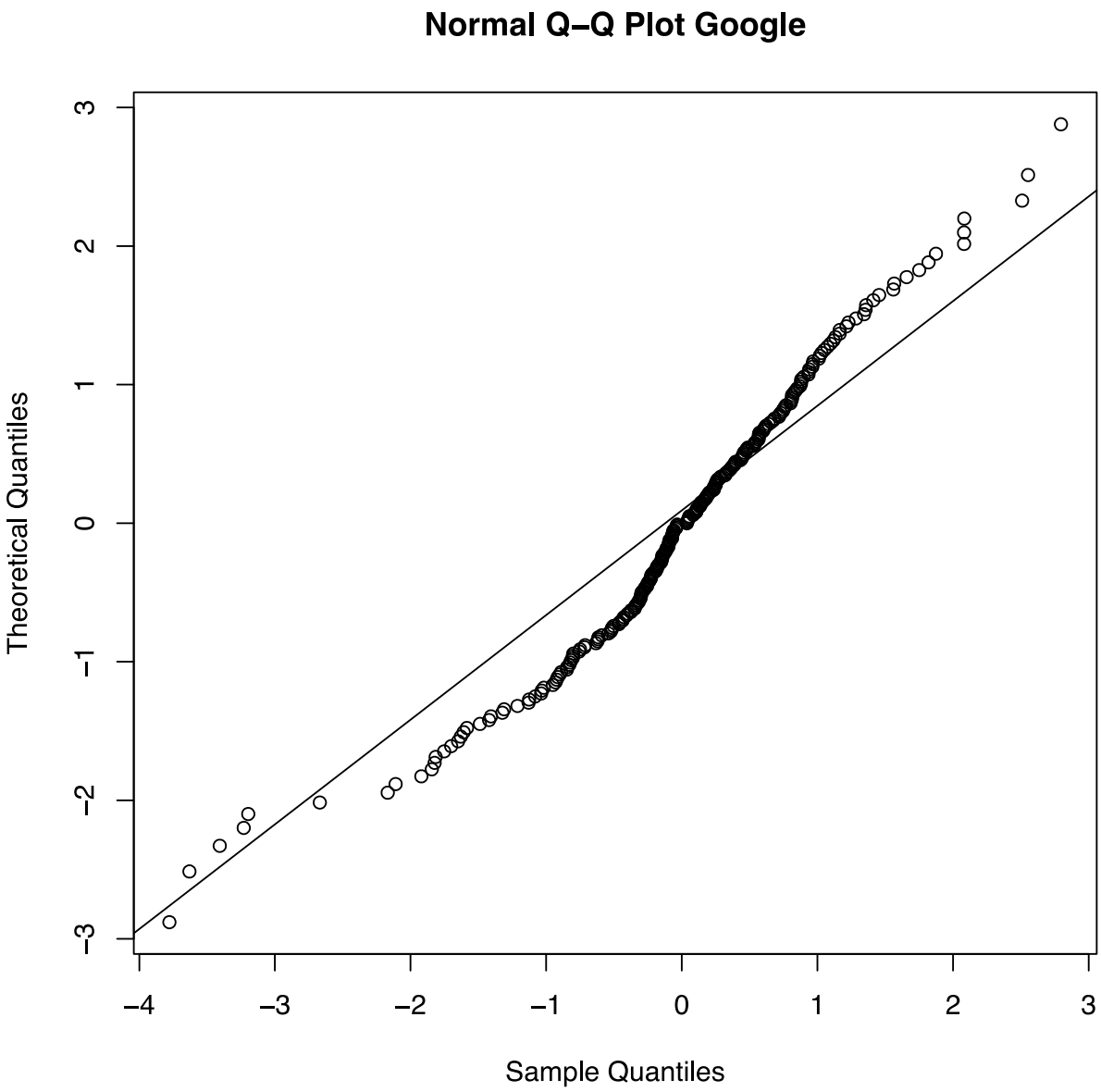

Citibank
kernel estimate
normal
0
10
20
30
40
50
60
−0.04
−0.02
0.00
0.02
0.04
Citibank daily log returns

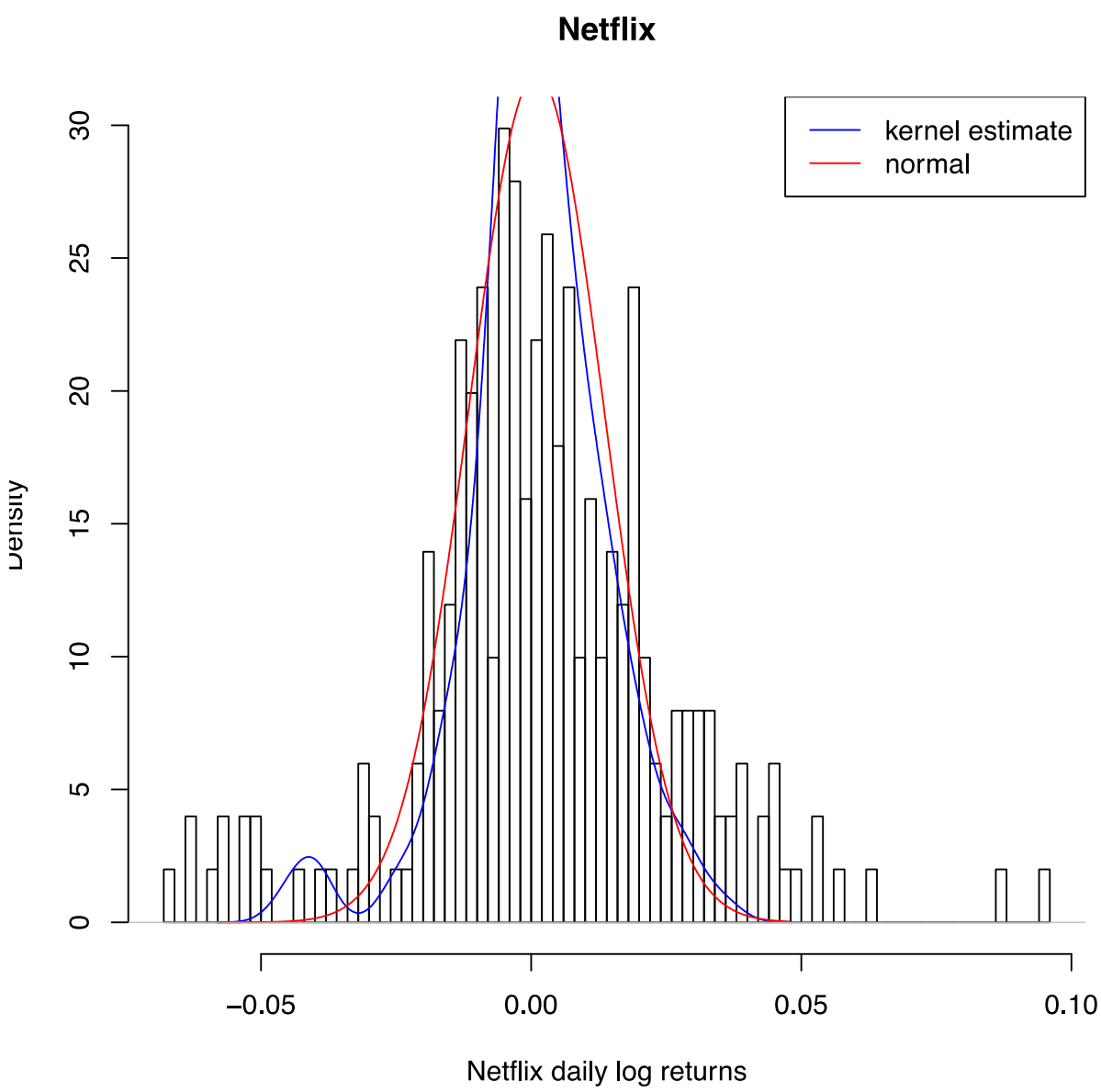

3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
Sample Quantiles

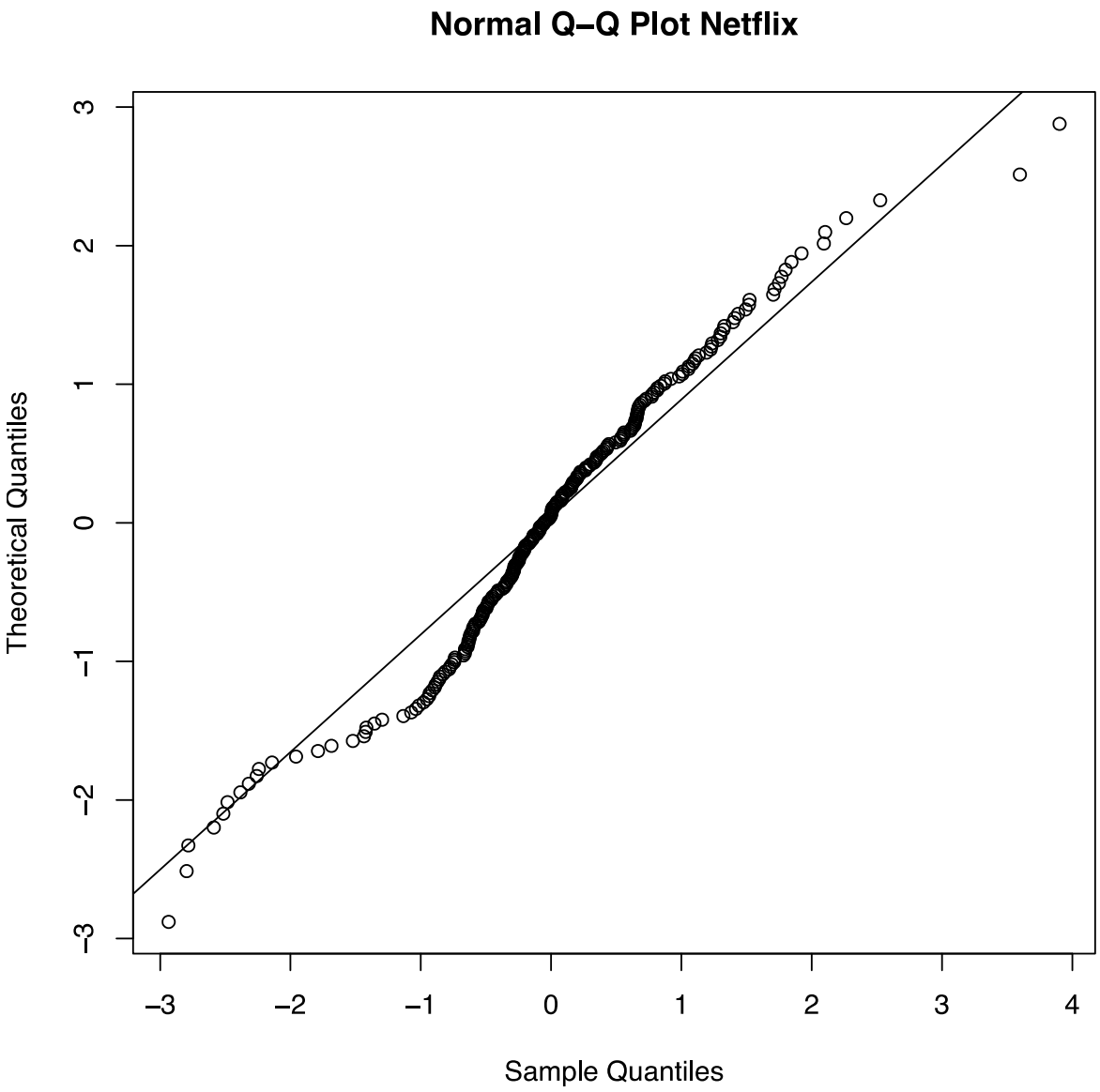

MCO
kernel estimate
normal
0
10
20
30
40
50
−0.04
−0.02
0.00
0.02
0.04
MCO daily log returns


3
2
1
0
−1
−2
−3
−4
−2
0
2
Sample Quantiles

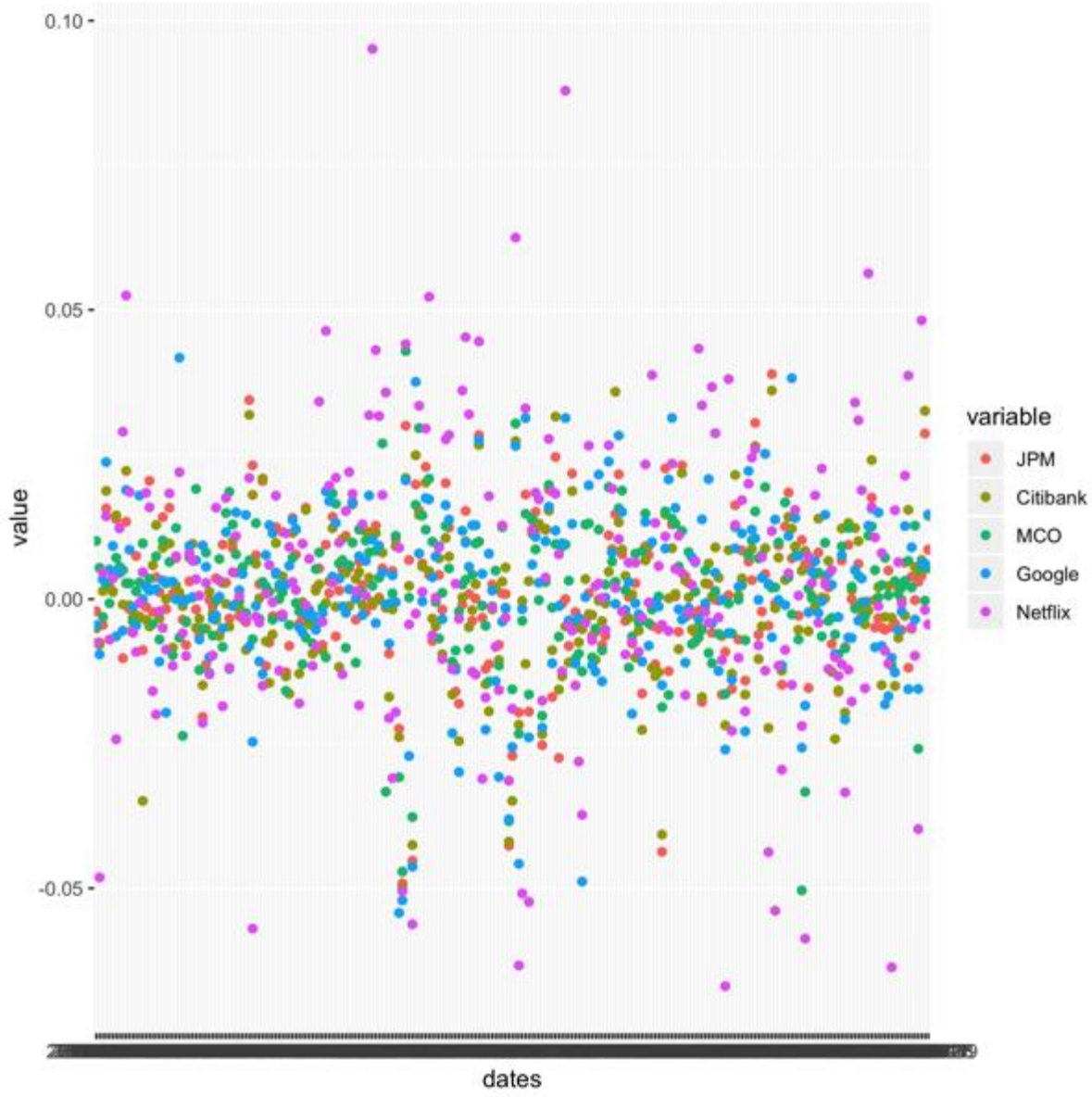

Google
kernel estimate
normal
Density
0
10
20
30
40
50
60
−0.04
−0.02
0.00
0.02
0.04
Google daily log returns


Normal Q–Q Plot Google
Theoretical Quantiles
Sample Quantiles
−3
−2
−1
0
1
2
3
−4
−3
−2
−1
0
1
2
3


Netflix
kernel estimate
normal
Density
0
5
10
15
20
25
30
−0.05
0.00
0.05
0.10
Netflix daily log returns


Normal Q–Q Plot Netflix
Theoretical Quantiles
Sample Quantiles
−3
−2
−1
0
1
2
3
4

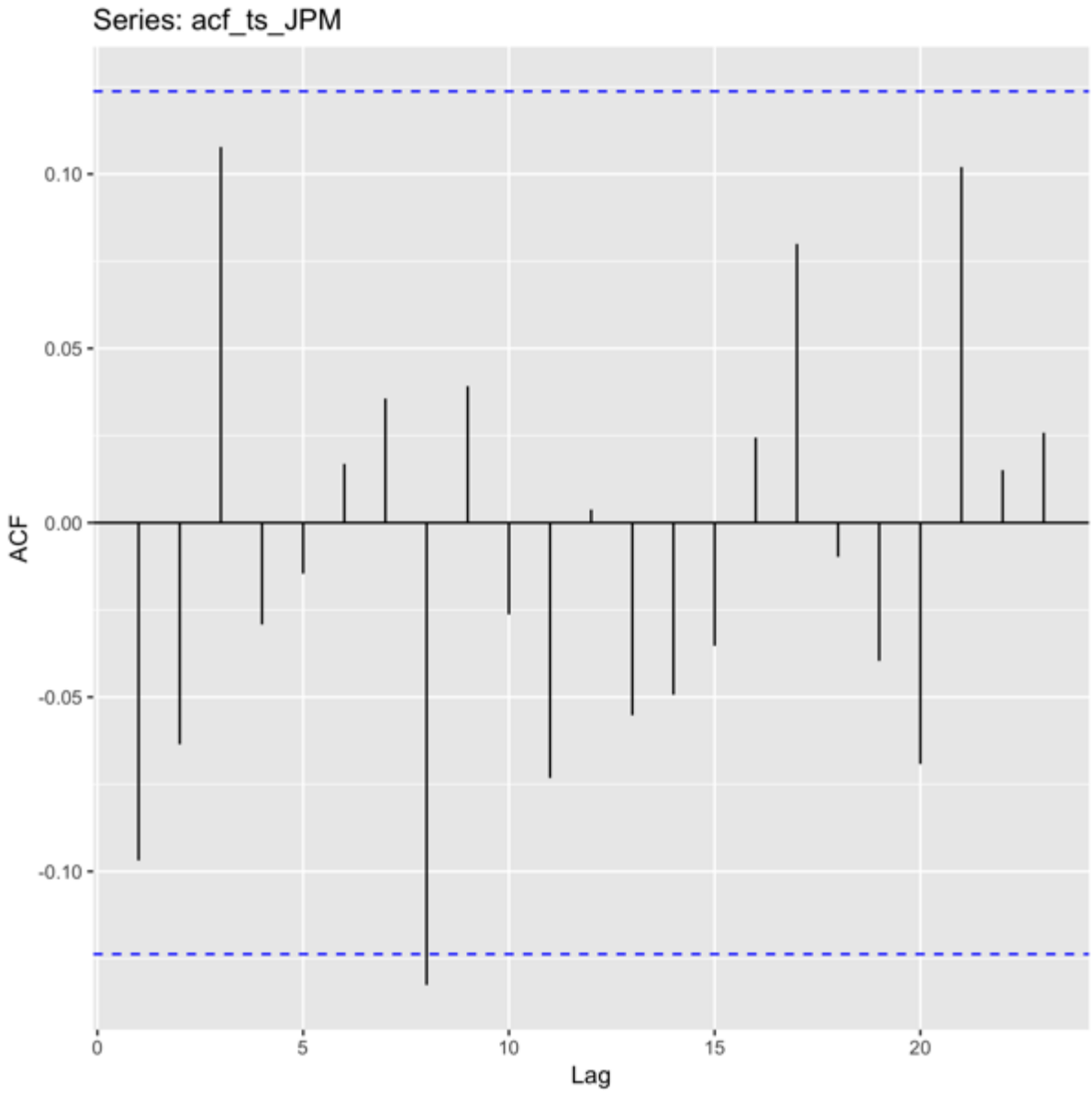
Scatter Plot

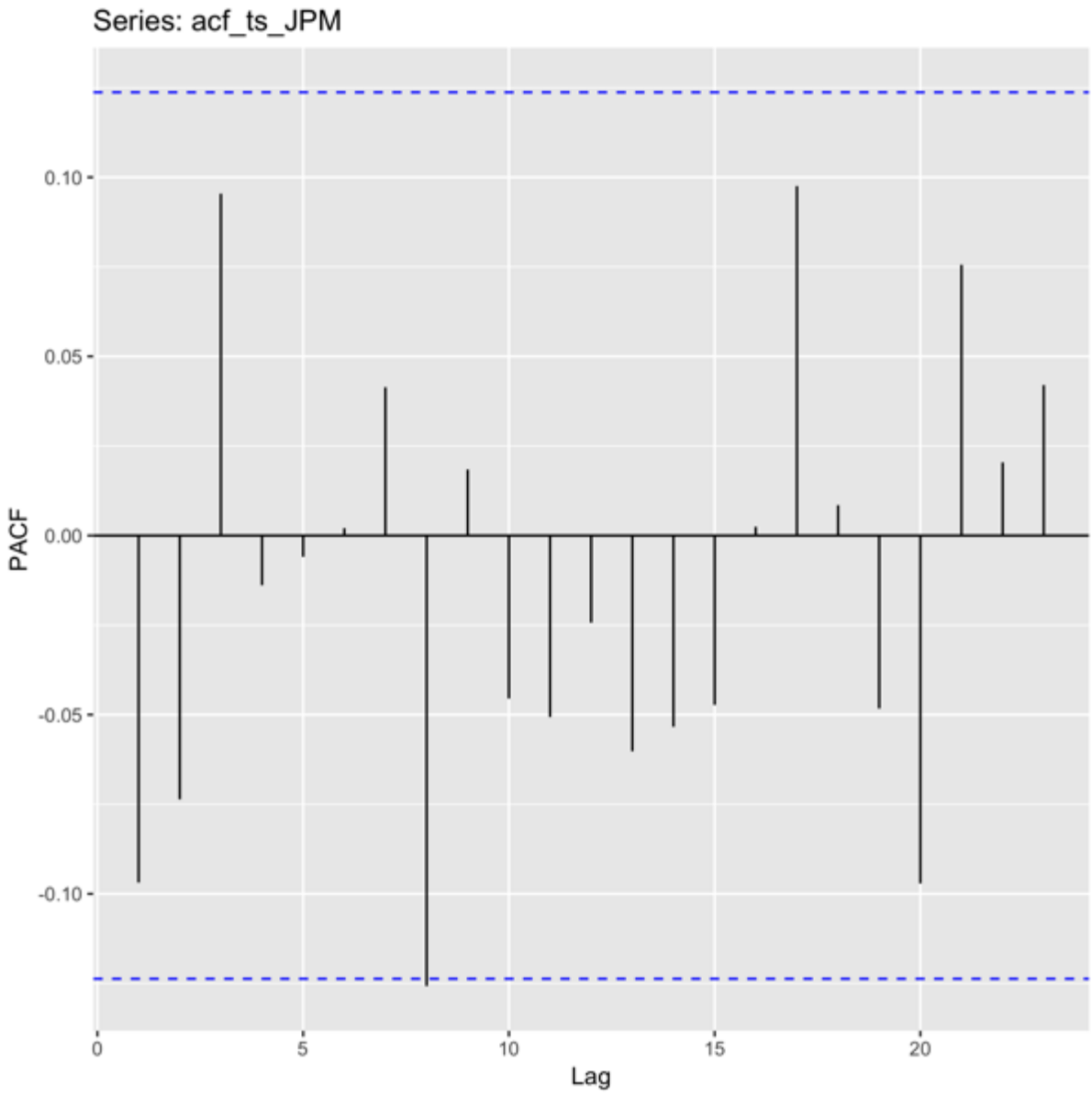


Correlation matrix of stocks

	JPM	Citibank	MCO	Google	Netflix
JPM	1.0000000	0.8656407	0.5567489	0.5056059	0.3481915
Citibank	0.8656407	1.0000000	0.4979284	0.4668286	0.3164161
MCO	0.5567489	0.4979284	1.0000000	0.5849182	0.4283851
Google	0.5056059	0.4668286	0.5849182	1.0000000	0.5259787
Netflix	0.3481915	0.3164161	0.4283851	0.5259787	1.0000000

ACF and PACF for JPM


Series: acf_ts_JPM
0.10
0.05
0.00
-0.05
-0.10
ACF
0
5
10
15
20
Lag



ARIMA model for JPM

Augmented Dickey-Fuller Test

Dickey-Fuller = -5.4444, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

KPSS Test for Level Stationarity

KPSS Level = 0.093725, Truncation lag parameter = 3, p-value = 0.1

Box-Pierce test

X-squared = 2.3546, df = 1, p-value = 0.1249

Coefficients:

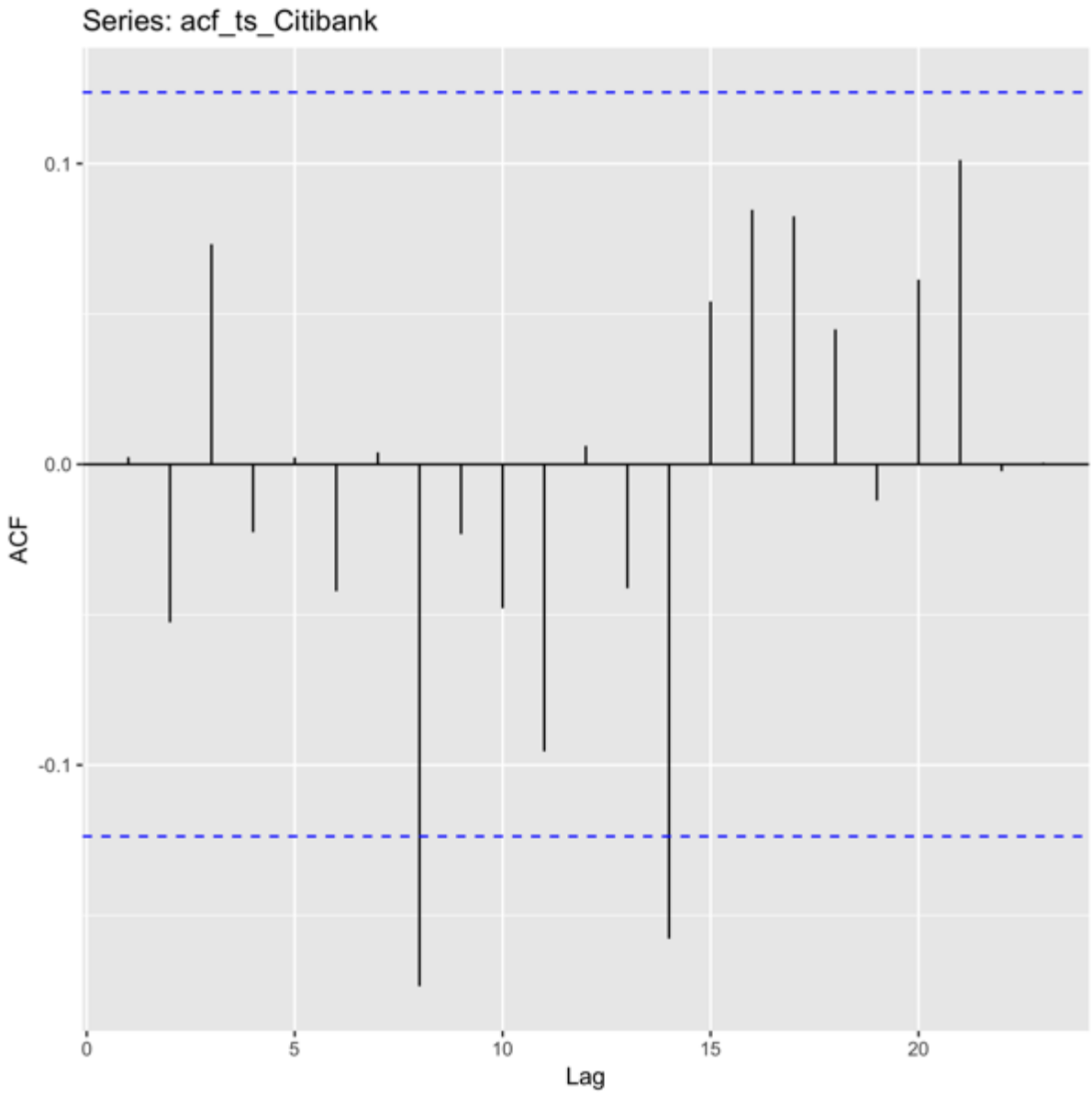
	intercept
	1e-03
s.e.	8e-04

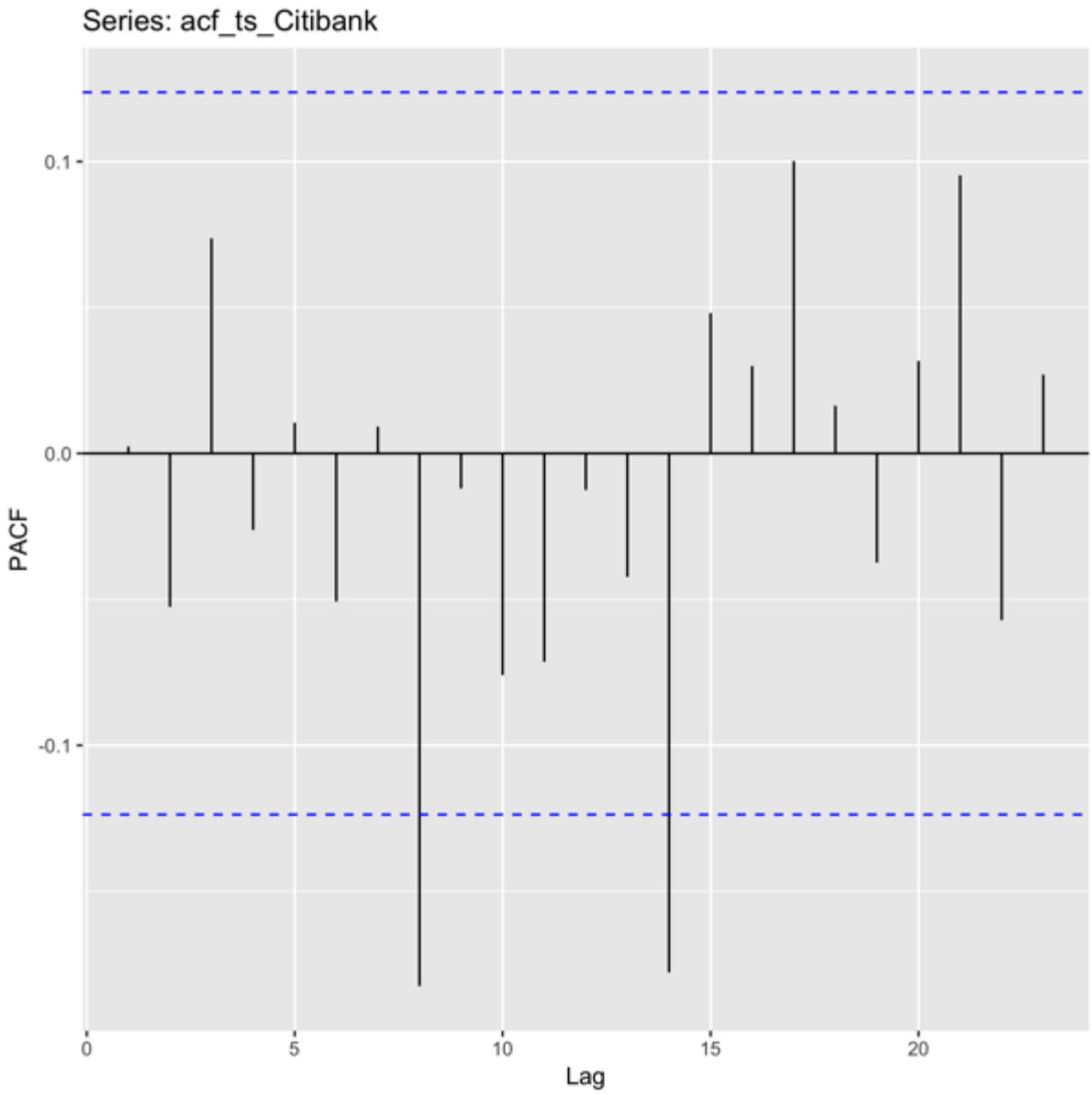
sigma^2 estimated as 0.0001557: log likelihood = 744.15, aic = -1484.29

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	4.025094e-19	0.01247956	0.009031312	-Inf	Inf	0.669607	-0.09685461

ACF and PACF for Citibank





ARIMA model for Citibank

Augmented Dickey-Fuller Test

Dickey-Fuller = -5.7589, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

KPSS Test for Level Stationarity

KPSS Level = 0.075473, Truncation lag parameter = 3, p-value = 0.1

Box-Pierce test

X-squared = 0.0014283, df = 1, p-value = 0.9699

Coefficients:

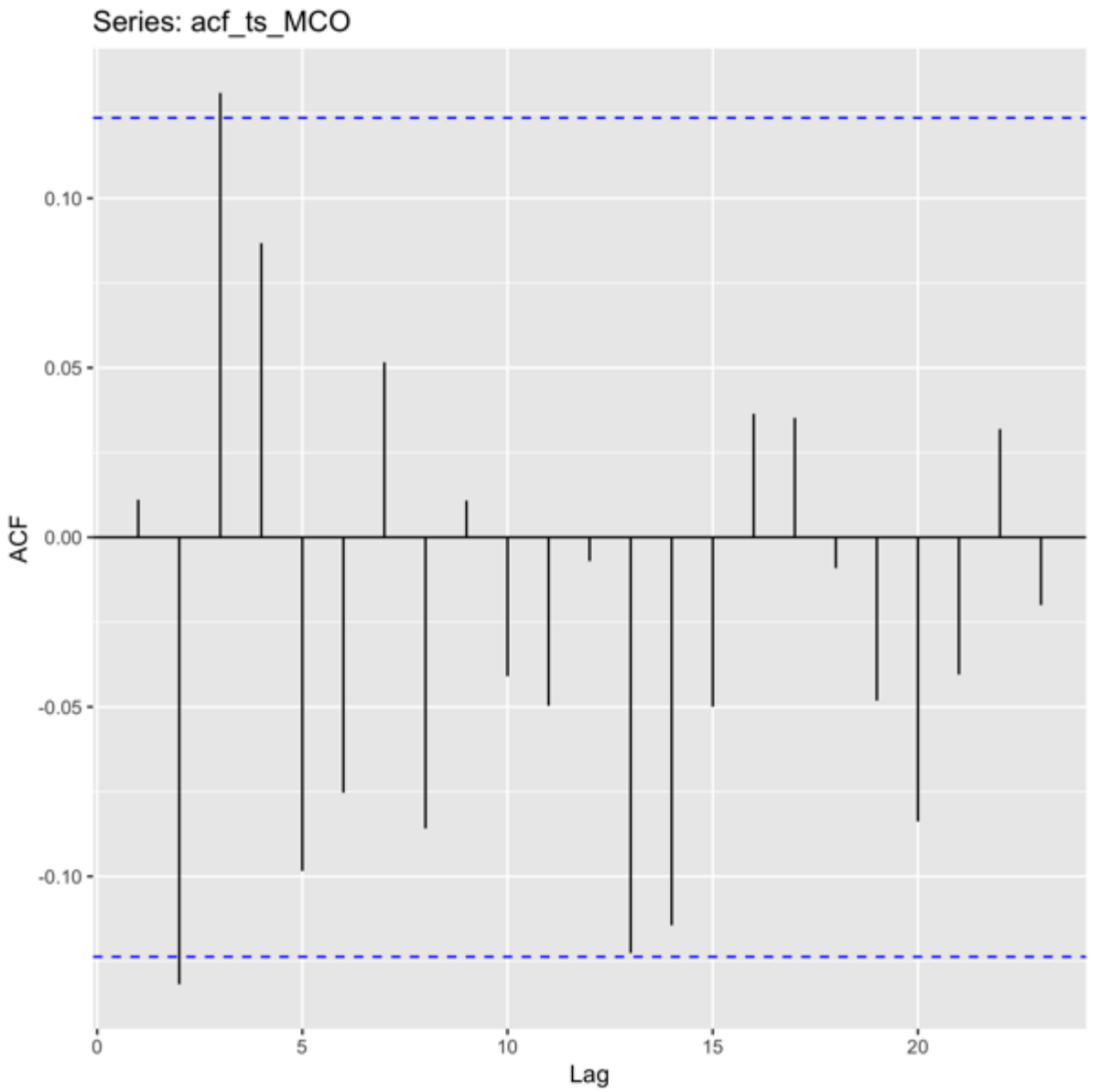
	intercept
	2e-04
s.e.	8e-04

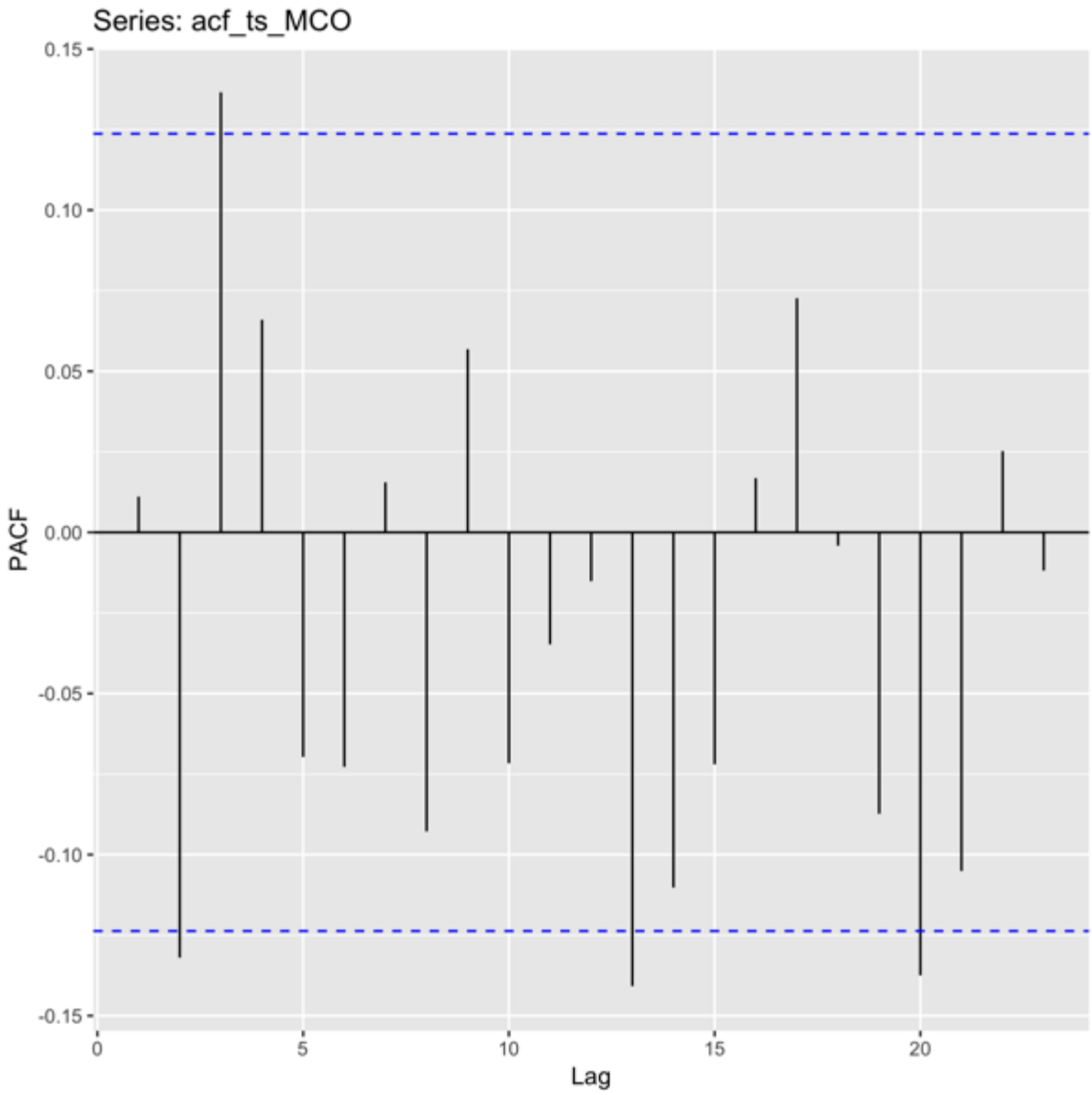
sigma^2 estimated as 0.0001681: log likelihood = 734.57, aic = -1465.13

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	3.641329e-15	0.01296511	0.009448879	-Inf	Inf	0.6991171	0.002385469

ACF and PACF for MCO





ARIMA model for MCO

Augmented Dickey-Fuller Test

Dickey-Fuller = -6.0317, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

KPSS Test for Level Stationarity

KPSS Level = 0.066547, Truncation lag parameter = 3, p-value = 0.1

Box-Pierce test

X-squared = 0.030886, df = 1, p-value = 0.8605

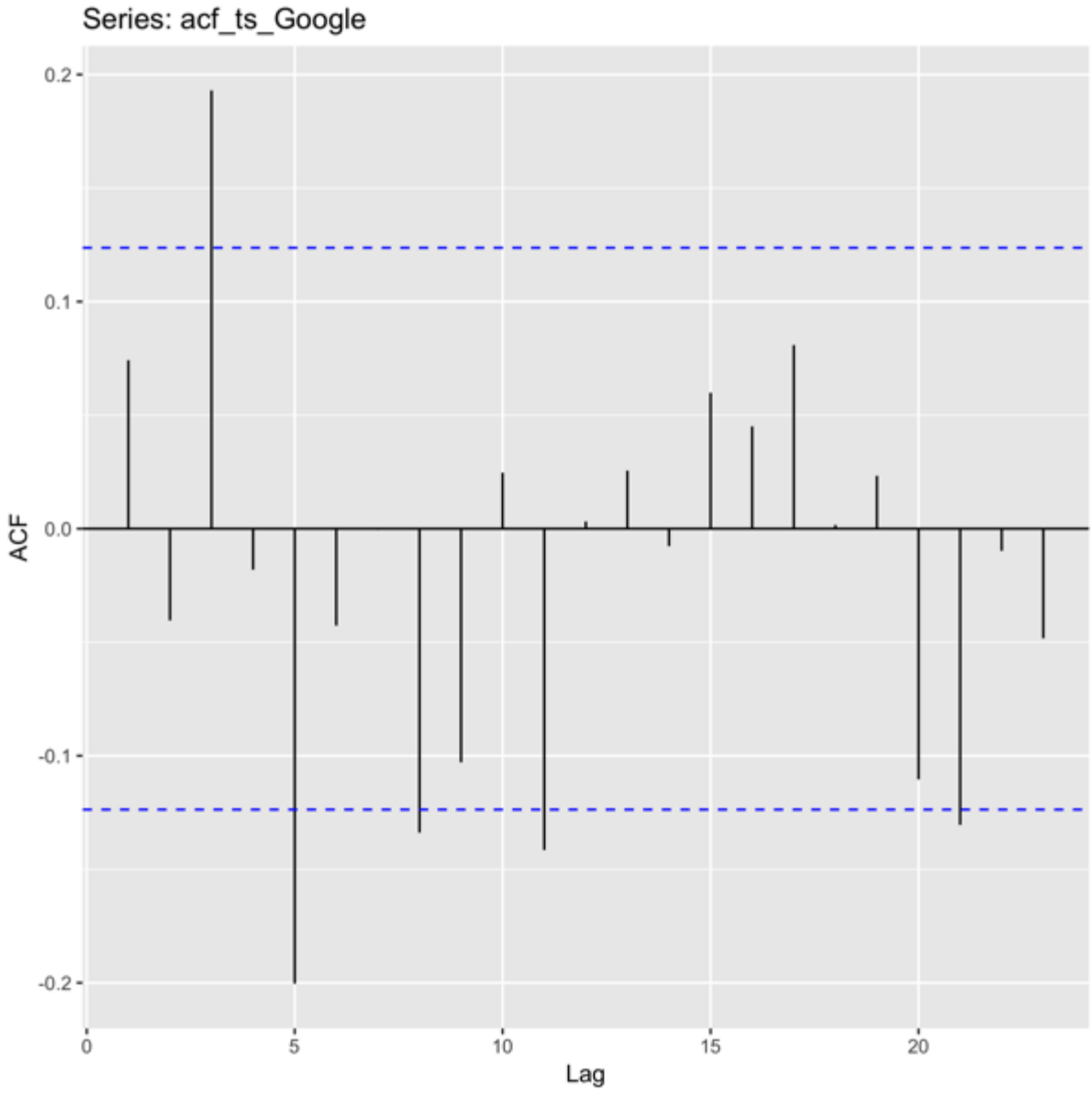
Coefficients:
intercept
0.0011
s.e. 0.0007

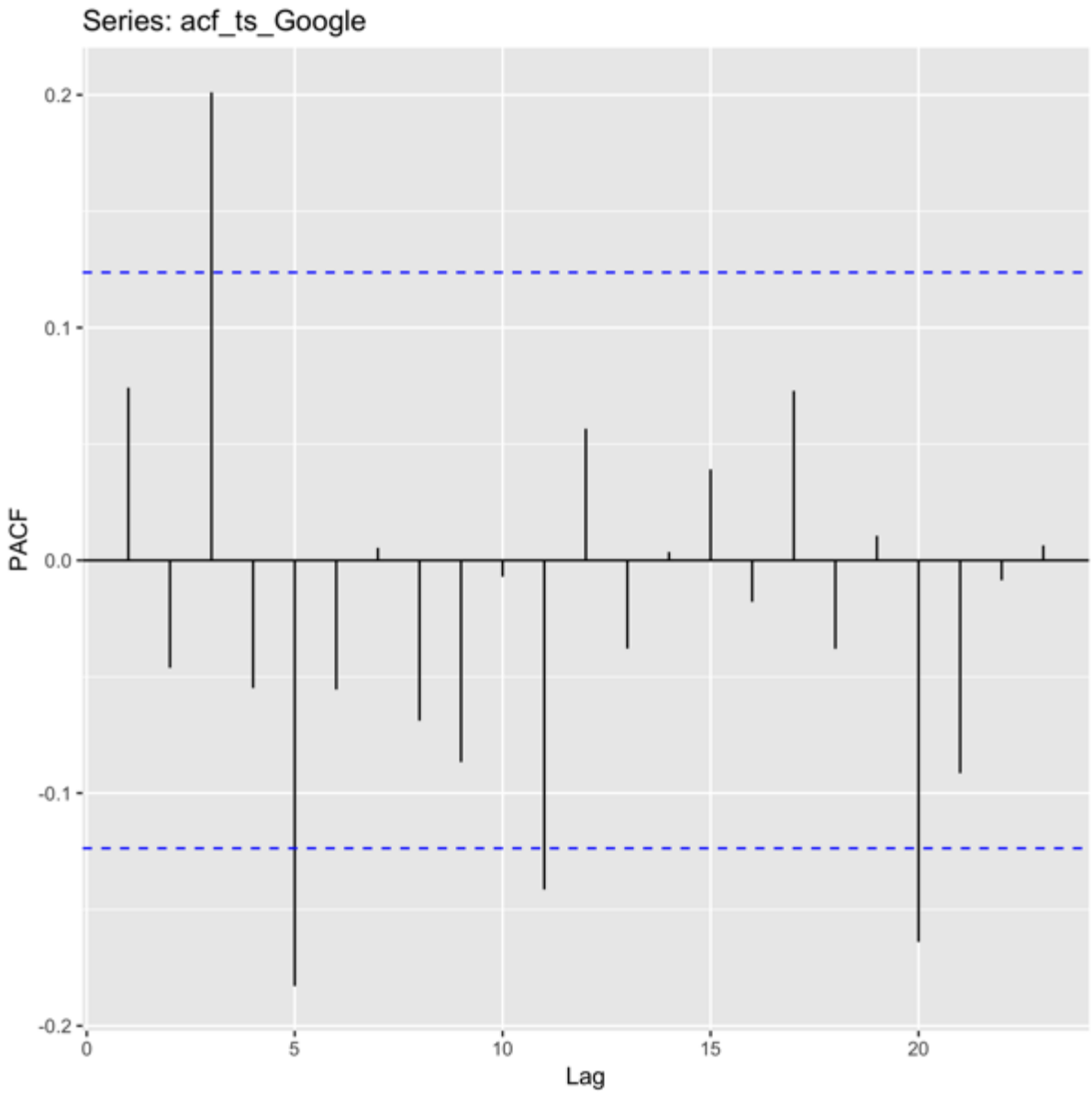
sigma^2 estimated as 0.0001395: log likelihood = 758, aic = -1512.01

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	3.317012e-15	0.01180924	0.008492142	-Inf	Inf	0.7076676	0.01109291

ACF and PACF for Google





ARIMA model for Google

Augmented Dickey-Fuller Test

Dickey-Fuller = -6.5003, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

KPSS Test for Level Stationarity

KPSS Level = 0.053713, Truncation lag parameter = 3, p-value = 0.1

Box-Pierce test

X-squared = 1.3815, df = 1, p-value = 0.2398

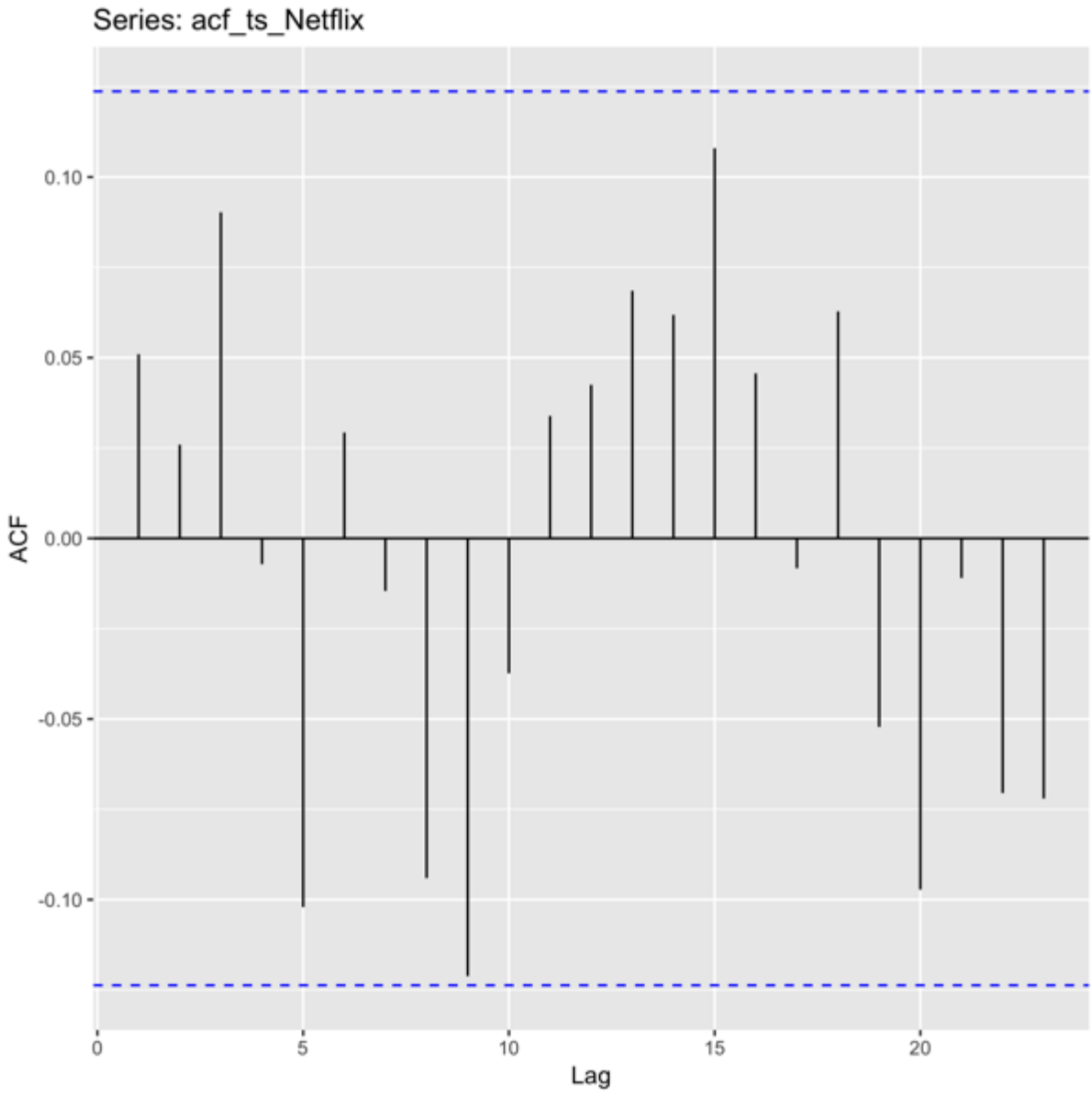
Coefficients:
intercept
9e-04
s.e. 9e-04

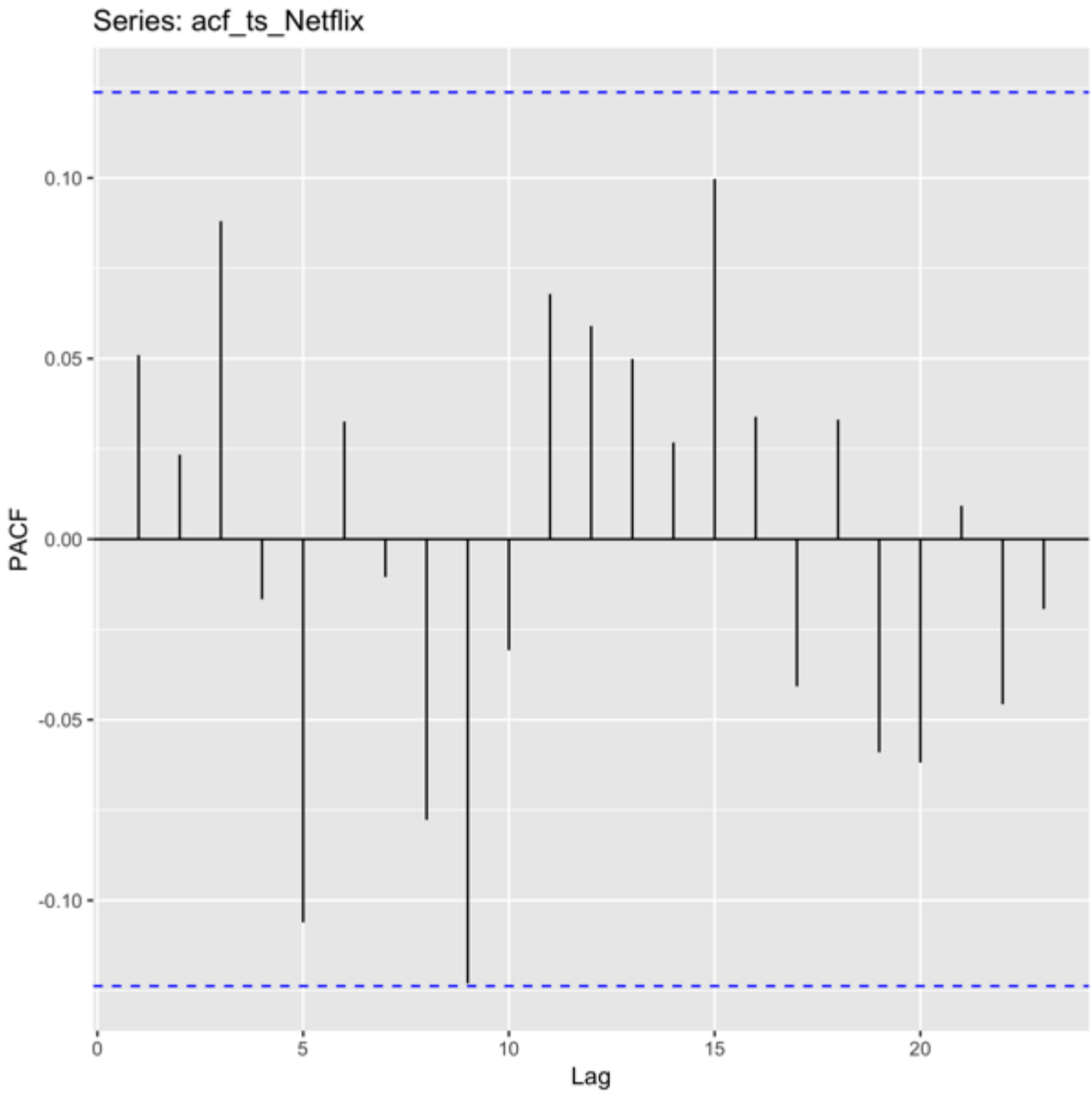
sigma^2 estimated as 0.0002124: log likelihood = 705.22, aic = -1406.44

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.517139e-19	0.01457316	0.01058716	-Inf	Inf	0.7150419	0.07418842

ACF and PACF for Netflix





ARIMA model for Netflix

Augmented Dickey-Fuller Test

Dickey-Fuller = -6.0382, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

KPSS Test for Level Stationarity

KPSS Level = 0.10999, Truncation lag parameter = 3, p-value = 0.1

Box-Pierce test

X-squared = 0.65193, df = 1, p-value = 0.4194

```
Coefficients:
      intercept
         0.0026
s.e.     0.0015

sigma^2 estimated as 0.0005595:  log likelihood = 583.66,  aic = -1163.32

Training set error measures:
                      ME      RMSE       MAE      MPE     MAPE      MASE
Training set -3.320567e-15 0.02365293 0.01730075 84.23851 129.0978 0.7135184
                    ACF1
Training set 0.05096403
```

Code

```
# Title    : TODO
# Objective : TODO
# Created by: adityaprasann
# Created on: 22/9/18
library(tidyverse)
library(ggplot2)
library(reshape2)
library(kdensity)
library(stats)
library(tseries)
library(forecast)
# Read data from CSV file
assignFilesData <- read.csv("./EconometricsGroupAssign.csv")
# Load data to a data frame
assignData =
data.frame(assignFilesData$JPM.Adjusted,assignFilesData$C.Adjusted,assignFilesData$MCO.Adjusted,ass
ignFilesData$GOOGL.Adjusted,assignFilesData$NFLX.Adjusted)
symbols <- list('JPM','Citibank','MCO','Google','Netflix')
count <- 1
logReturnsDf <- data.frame(diff(log(assignData[,count]), lag=1),
                           diff(log(assignData[,count+1]), lag=1),
                           diff(log(assignData[,count+2]), lag=1),
                           diff(log(assignData[,count+3]), lag=1),
                           diff(log(assignData[,count+4]), lag=1))

names(logReturnsDf) <- c(symbols[[count]][1], symbols[[count+1]][1], symbols[[count+2]][1],
symbols[[count+3]][1], symbols[[count+4]][1])

dates <- c('2017-09-21','2017-09-22','2017-09-25','2017-09-26','2017-09-27','2017-09-28','2017-09-
29','2017-10-02','2017-10-03','2017-10-04','2017-10-05','2017-10-06','2017-10-09','2017-10-10','2017-
10-11','2017-10-12','2017-10-13','2017-10-16','2017-10-17','2017-10-18','2017-10-19','2017-10-
20','2017-10-23','2017-10-24','2017-10-25','2017-10-26','2017-10-27','2017-10-30','2017-10-31','2017-
```

11-01','2017-11-02','2017-11-03','2017-11-06','2017-11-07','2017-11-08','2017-11-09','2017-11-10','2017-11-13','2017-11-14','2017-11-15','2017-11-16','2017-11-17','2017-11-20','2017-11-21','2017-11-22','2017-11-24','2017-11-27','2017-11-28','2017-11-29','2017-11-30','2017-12-01','2017-12-04','2017-12-05','2017-12-06','2017-12-07','2017-12-08','2017-12-11','2017-12-12','2017-12-13','2017-12-14','2017-12-15','2017-12-18','2017-12-19','2017-12-20','2017-12-21','2017-12-22','2017-12-26','2017-12-27','2017-12-28','2017-12-29','2018-01-02','2018-01-03','2018-01-04','2018-01-05','2018-01-08','2018-01-09','2018-01-10','2018-01-11','2018-01-12','2018-01-16','2018-01-17','2018-01-18','2018-01-19','2018-01-22','2018-01-23','2018-01-24','2018-01-25','2018-01-26','2018-01-29','2018-01-30','2018-01-31','2018-02-01','2018-02-02','2018-02-05','2018-02-06','2018-02-07','2018-02-08','2018-02-09','2018-02-12','2018-02-13','2018-02-14','2018-02-15','2018-02-16','2018-02-20','2018-02-21','2018-02-22','2018-02-23','2018-02-26','2018-02-27','2018-02-28','2018-03-01','2018-03-02','2018-03-05','2018-03-06','2018-03-07','2018-03-08','2018-03-09','2018-03-12','2018-03-13','2018-03-14','2018-03-15','2018-03-16','2018-03-19','2018-03-20','2018-03-21','2018-03-22','2018-03-23','2018-03-26','2018-03-27','2018-03-28','2018-03-29','2018-04-02','2018-04-03','2018-04-04','2018-04-05','2018-04-06','2018-04-09','2018-04-10','2018-04-11','2018-04-12','2018-04-13','2018-04-16','2018-04-17','2018-04-18','2018-04-19','2018-04-20','2018-04-23','2018-04-24','2018-04-25','2018-04-26','2018-04-27','2018-04-30','2018-05-01','2018-05-02','2018-05-03','2018-05-04','2018-05-07','2018-05-08','2018-05-09','2018-05-10','2018-05-11','2018-05-14','2018-05-15','2018-05-16','2018-05-17','2018-05-18','2018-05-21','2018-05-22','2018-05-23','2018-05-24','2018-05-25','2018-05-29','2018-05-30','2018-05-31','2018-06-01','2018-06-04','2018-06-05','2018-06-06','2018-06-07','2018-06-08','2018-06-11','2018-06-12','2018-06-13','2018-06-14','2018-06-15','2018-06-18','2018-06-19','2018-06-20','2018-06-21','2018-06-22','2018-06-25','2018-06-26','2018-06-27','2018-06-28','2018-06-29','2018-07-02','2018-07-03','2018-07-05','2018-07-06','2018-07-09','2018-07-10','2018-07-11','2018-07-12','2018-07-13','2018-07-16','2018-07-17','2018-07-18','2018-07-19','2018-07-20','2018-07-23','2018-07-24','2018-07-25','2018-07-26','2018-07-27','2018-07-30','2018-07-31','2018-08-01','2018-08-02','2018-08-03','2018-08-06','2018-08-07','2018-08-08','2018-08-09','2018-08-10','2018-08-13','2018-08-14','2018-08-15','2018-08-16','2018-08-17','2018-08-20','2018-08-21','2018-08-22','2018-08-23','2018-08-24','2018-08-27','2018-08-28','2018-08-29','2018-08-30','2018-08-31','2018-09-04','2018-09-05','2018-09-06','2018-09-07','2018-09-10','2018-09-11','2018-09-12','2018-09-13','2018-09-14','2018-09-17','2018-09-18','2018-09-19')
logReturnsDateDf <- cbind(dates,logReturnsDf)
print(logReturnsDateDf)
dates <- c('2017-09-21','2017-09-22','2017-09-25','2017-09-26','2017-09-27','2017-09-28','2017-09-29','2017-10-02','2017-10-03','2017-10-04','2017-10-05','2017-10-06','2017-10-09','2017-10-10','2017-10-11','2017-10-12','2017-10-13','2017-10-16','2017-10-17','2017-10-18','2017-10-19','2017-10-20','2017-10-23','2017-10-24','2017-10-25','2017-10-26','2017-10-27','2017-10-30','2017-10-31','2017-11-01','2017-11-02','2017-11-03','2017-11-06','2017-11-07','2017-11-08','2017-11-09','2017-11-10','2017-11-13','2017-11-14','2017-11-15','2017-11-16','2017-11-17','2017-11-20','2017-11-21','2017-11-22','2017-11-24','2017-11-27','2017-11-28','2017-11-29','2017-11-30','2017-12-01','2017-12-04','2017-12-05','2017-12-06','2017-12-07','2017-12-08','2017-12-11','2017-12-12','2017-12-13','2017-12-14','2017-12-15','2017-12-18','2017-12-19','2017-12-20','2017-12-21','2017-12-22','2017-12-26','2017-12-27','2017-12-28','2017-12-29','2018-01-02','2018-01-03','2018-01-04','2018-01-05','2018-01-08','2018-01-09','2018-01-10','2018-01-11','2018-01-12','2018-01-16','2018-01-17','2018-01-18','2018-01-19','2018-01-22','2018-01-23','2018-01-24','2018-01-25','2018-01-26','2018-01-29','2018-01-30','2018-01-31','2018-02-01','2018-02-02','2018-02-05','2018-02-06','2018-02-07','2018-02-08','2018-02-09','2018-02-12','2018-02-13','2018-02-14','2018-02-15','2018-02-16','2018-02-20','2018-02-21','2018-02-22','2018-02-23','2018-02-26','2018-02-27','2018-02-28','2018-03-01','2018-03-02','2018-03-05','2018-03-06','2018-03-07','2018-03-08','2018-03-09','2018-03-12','2018-03-13','2018-03-14','2018-03-15','2018-03-16','2018-03-19','2018-03-20','2018-03-21','2018-03-22','2018-03-23','2018-03-26','2018-03-27','2018-03-28','2018-03-29','2018-04-02','2018-04-03','2018-04-04','2018-04-05','2018-04-06','2018-04-09','2018-04-10','2018-04-11','2018-04-12','2018-04-13','2018-04-16','2018-04-17','2018-04-18','2018-04-19','2018-04-20','2018-04-23','2018-04-24','2018-04-25','2018-04-26','2018-04-27','2018-04-30','2018-05-01','2018-05-02','2018-05-03','2018-05-04','2018-05-07','2018-05-08','2018-05-09','2018-05-10','2018-05-11','2018-05-14','2018-05-15','2018-05-16','2018-

```
05-17','2018-05-18','2018-05-21','2018-05-22','2018-05-23','2018-05-24','2018-05-25','2018-05-
29','2018-05-30','2018-05-31','2018-06-01','2018-06-04','2018-06-05','2018-06-06','2018-06-07','2018-
06-08','2018-06-11','2018-06-12','2018-06-13','2018-06-14','2018-06-15','2018-06-18','2018-06-
19','2018-06-20','2018-06-21','2018-06-22','2018-06-25','2018-06-26','2018-06-27','2018-06-28','2018-
06-29','2018-07-02','2018-07-03','2018-07-05','2018-07-06','2018-07-09','2018-07-10','2018-07-
11','2018-07-12','2018-07-13','2018-07-16','2018-07-17','2018-07-18','2018-07-19','2018-07-20','2018-
07-23','2018-07-24','2018-07-25','2018-07-26','2018-07-27','2018-07-30','2018-07-31','2018-08-
01','2018-08-02','2018-08-03','2018-08-06','2018-08-07','2018-08-08','2018-08-09','2018-08-10','2018-
08-13','2018-08-14','2018-08-15','2018-08-16','2018-08-17','2018-08-20','2018-08-21','2018-08-
22','2018-08-23','2018-08-24','2018-08-27','2018-08-28','2018-08-29','2018-08-30','2018-08-31','2018-
09-04','2018-09-05','2018-09-06','2018-09-07','2018-09-10','2018-09-11','2018-09-12','2018-09-
13','2018-09-14','2018-09-17','2018-09-18','2018-09-19','2018-09-20')
assignDataDateDf <-cbind(dates,assignData)
print(assignDataDateDf)

assignDataDateDf1 <- melt(assignDataDateDf,id="dates")
ggplot(assignDataDateDf1,aes(x=dates,y=value,colour=variable,group=variable)) + geom_line()
ggsave("EconometricsGroupAssign1StockReturns.png")

logReturnsDateDf1 <- melt(logReturnsDateDf,id="dates")
ggplot(logReturnsDateDf1,aes(x=dates,y=value,colour=variable,group=variable)) + geom_line()
ggsave("EconometricsGroupAssign1LogReturns.png")

ggplot(data = logReturnsDateDf1, aes(x = dates, y = value, color = variable)) + geom_point()
ggsave("EconometricsGroupAssign1ScatterPlots.png")

print("The corelaatin matrix is")
print(cor(logReturnsDf))

hist(logReturnsDateDf$JPM,breaks=64,freq=FALSE,main="JPM",xlab='JPM daily log returns')
kde_JPM <- kdensity(logReturnsDateDf$JPM,start="normal")
lines(kde_JPM, col="blue")
lines(kde_JPM, plot_start=TRUE, col="red")
legend("topright",c("kernel estimate","normal"), lty=c(1,1), lwd=c(1,1), col=c("blue","red"))
mean_JPM = mean(logReturnsDateDf$JPM)
sd_JPM = sd(logReturnsDateDf$JPM)
JPM_std = (logReturnsDateDf$JPM - mean_JPM)/sd_JPM
qqnorm(JPM_std, main="Normal Q-Q Plot JPM", plot.it=TRUE, datax=TRUE)
qqline(JPM_std, datax=FALSE, distribution=qnorm, probs=c(0.25,0.75), qtype=7)

hist(logReturnsDateDf$Citibank,breaks=64,freq=FALSE,main="Citibank",xlab='Citibank daily log returns')
kde_Citibank <- kdensity(logReturnsDateDf$JPM,start="normal")
lines(kde_Citibank, col="blue")
lines(kde_Citibank, plot_start=TRUE, col="red")
legend("topright",c("kernel estimate","normal"), lty=c(1,1), lwd=c(1,1), col=c("blue","red"))
mean_Citibank = mean(logReturnsDateDf$Citibank)
sd_Citibank = sd(logReturnsDateDf$Citibank)
Citibank_std = (logReturnsDateDf$Citibank - mean_Citibank)/sd_Citibank
qqnorm(Citibank_std, main="Normal Q-Q Plot JPM", plot.it=TRUE, datax=TRUE)
qqline(Citibank_std, datax=FALSE, distribution=qnorm, probs=c(0.25,0.75), qtype=7)


hist(logReturnsDateDf$MCO,breaks=64,freq=FALSE,main="MCO",xlab='MCO daily log returns')
kde_MCO <- kdensity(logReturnsDateDf$JPM,start="normal")
```

```
lines(kde_MCO, col="blue")
lines(kde_MCO, plot_start=TRUE, col="red")
legend("topright",c("kernel estimate","normal"), lty=c(1,1), lwd=c(1,1), col=c("blue","red"))
mean_MCO = mean(logReturnsDateDf$MCO)
sd_MCO = sd(logReturnsDateDf$MCO)
MCO_std = (logReturnsDateDf$MCO - mean_MCO)/sd_MCO
qqnorm(MCO_std, main="Normal Q-Q Plot JPM", plot.it=TRUE, datax=TRUE)
qqline(MCO_std, datax=FALSE, distribution=qnorm, probs=c(0.25,0.75), qtype=7)

hist(logReturnsDateDf$Google,breaks=64,freq=FALSE,main="Google",xlab='Google daily log returns')
kde_Google <- kdensity(logReturnsDateDf$JPM,start="normal")
lines(kde_Google, col="blue")
lines(kde_Google, plot_start=TRUE, col="red")
legend("topright",c("kernel estimate","normal"), lty=c(1,1), lwd=c(1,1), col=c("blue","red"))
mean_Google = mean(logReturnsDateDf$Google)
sd_Google = sd(logReturnsDateDf$Google)
Google_std = (logReturnsDateDf$Google - mean_Google)/sd_Google
qqnorm(Google_std, main="Normal Q-Q Plot Google", plot.it=TRUE, datax=TRUE)
qqline(Google_std, datax=FALSE, distribution=qnorm, probs=c(0.25,0.75), qtype=7)

hist(logReturnsDateDf$Netflix,breaks=64,freq=FALSE,main="Netflix",xlab='Netflix daily log returns')
kde_Netflix <- kdensity(logReturnsDateDf$JPM,start="normal")
lines(kde_Netflix, col="blue")
lines(kde_Netflix, plot_start=TRUE, col="red")
legend("topright",c("kernel estimate","normal"), lty=c(1,1), lwd=c(1,1), col=c("blue","red"))
mean_Netflix = mean(logReturnsDateDf$Netflix)
sd_Netflix = sd(logReturnsDateDf$Netflix)
Netflix_std = (logReturnsDateDf$Netflix - mean_Netflix)/sd_Netflix
qqnorm(Netflix_std, main="Normal Q-Q Plot Netflix", plot.it=TRUE, datax=TRUE)
qqline(Netflix_std, datax=FALSE, distribution=qnorm, probs=c(0.25,0.75), qtype=7)

print("ARIMA model for JPM")

acf_ts_JPM <- ts(logReturnsDf$JPM)
ggAcf(acf_ts_JPM, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignACFJPM.png")

ggPacf(acf_ts_JPM, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignPACFJPM.png")

print(adf.test(acf_ts_JPM))

print(kpss.test(acf_ts_JPM))

jpmModel <- arima(acf_ts_JPM, method="ML")
print(Box.test(jpmModel$residuals))
summary(jpmModel)

print("ARIMA model for Citibank")

acf_ts_Citibank <- ts(logReturnsDf$Citibank)
ggAcf(acf_ts_Citibank, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignACFCitibank.png")
```

```
ggPacf(acf_ts_Citibank, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignPACFCitibank.png")

print(adf.test(acf_ts_Citibank))

print(kpss.test(acf_ts_Citibank))

citibankModel <- arima(acf_ts_Citibank, method="ML")
print(Box.test(citibankModel$residuals))
summary(citibankModel)

print("ARIMA model for MCO")

acf_ts_MCO <- ts(logReturnsDf$MCO)
ggAcf(acf_ts_MCO, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignACFMCO.png")

ggPacf(acf_ts_MCO, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignPACFMCO.png")

print(adf.test(acf_ts_MCO))

print(kpss.test(acf_ts_MCO))

mcoModel <- arima(acf_ts_MCO, method="ML")
print(Box.test(mcoModel$residuals))
summary(mcoModel)

print("ARIMA model for Google")

acf_ts_Google <- ts(logReturnsDf$Google)
ggAcf(acf_ts_Google, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignACFGoogle.png")

ggPacf(acf_ts_Google, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignPACFGoogle.png")

print(adf.test(acf_ts_Google))

print(kpss.test(acf_ts_Google))

googleModel <- arima(acf_ts_Google, method="ML")
print(Box.test(googleModel$residuals))
summary(googleModel)

print("ARIMA model for Netflix")

acf_ts_Netflix <- ts(logReturnsDf$Netflix)
ggAcf(acf_ts_Netflix, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignACFNetflix.png")

ggPacf(acf_ts_Netflix, plot = TRUE, lag.MAX = 25)
ggsave("EconometricsGroupAssignPACFNetflix.png")
```

```
print(adf.test(acf_ts_Netflix))

print(kpss.test(acf_ts_Netflix))

netflixModel <- arima(acf_ts_Netflix, method="ML")
print(Box.test(netflixModel$residuals))
summary(netflixModel)
```