Time Series Analysis

Moyi Li

Setup

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(PerformanceAnalytics)
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

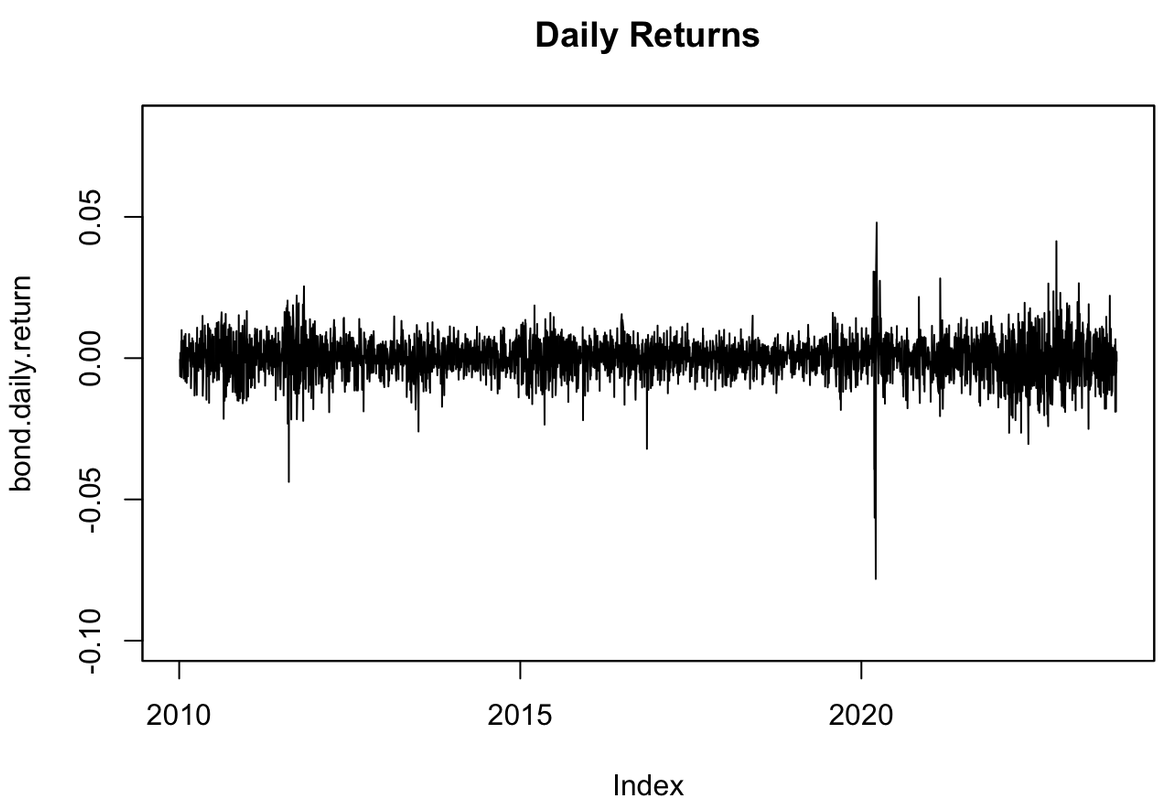
```
bond.data <- getSymbols("BLV", auto.assign=FALSE, from="2010-01-01", to="2023-10-01")$BLV.Adjuste
d
bond.daily.return <- na.omit(Return.calculate(bond.data, method="log"))
bond.monthly.return <- na.omit(Return.calculate(to.monthly(bond.data, OHLC=FALSE), method="log"))
```

Problem 1

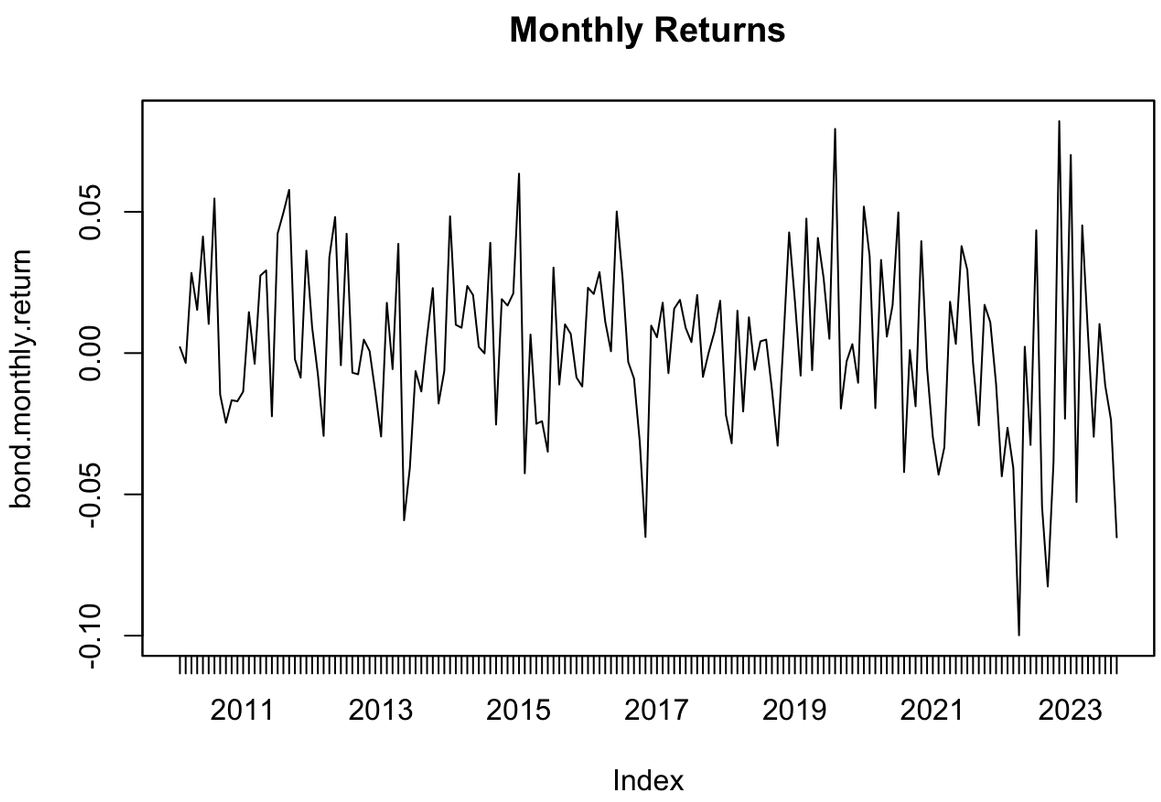
```
##### Problem 1 Question A
result_matrix <- matrix(NA, nrow = 4, ncol = 2)
rownames(result_matrix) <- c("Sample_Mean", "Sample_Variance", "Sample_Skewness", "Sample_Kurtosi
s")
colnames(result_matrix) <- c("daily_log_return", "monthly_log_return")

# Daily
T <- length(bond.daily.return)
sample_mean <- sum(bond.daily.return) / T
sample_var <- sum((bond.daily.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)
sample_skew <- sum((bond.daily.return - sample_mean)^3) / (T * sample_sd^3)
sample_kurtosis <- sum((bond.daily.return - sample_mean)^4) / (T * sample_sd^4)

result_matrix[, 1] <- c(sample_mean, sample_var, sample_skew, sample_kurtosis)

# Monthly
T <- length(bond.monthly.return)
sample_mean <- sum(bond.monthly.return) / T
sample_var <- sum((bond.monthly.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)
sample_skew <- sum((bond.monthly.return - sample_mean)^3) / (T * sample_sd^3)
sample_kurtosis <- sum((bond.monthly.return - sample_mean)^4) / (T * sample_sd^4)


result_matrix[, 2] <- c(sample_mean, sample_var, sample_skew, sample_kurtosis)
result_matrix
```

```
##                 daily_log_return monthly_log_return
## Sample_Mean         1.345753e-04       0.0027474199
## Sample_Variance     5.171445e-05       0.0009231832
## Sample_Skewness    -7.116212e-01      -0.1675594966
## Sample_Kurtosis     1.127342e+01       3.4456792287
```

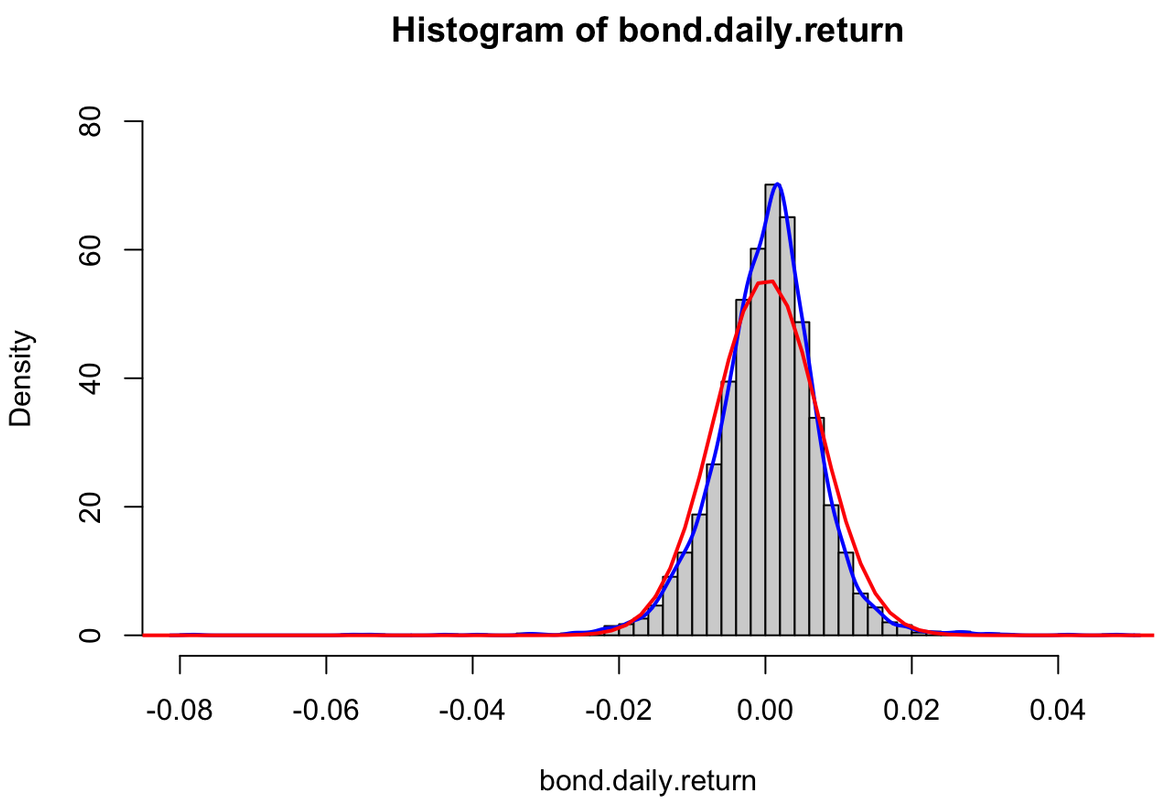
```
##### Problem 1 Question B
ylim_bond <- c(min(bond.monthly.return, bond.daily.return),
               max(bond.monthly.return, bond.daily.return))

plot.zoo(bond.daily.return, main="Daily Returns", ylim=ylim_bond)
```



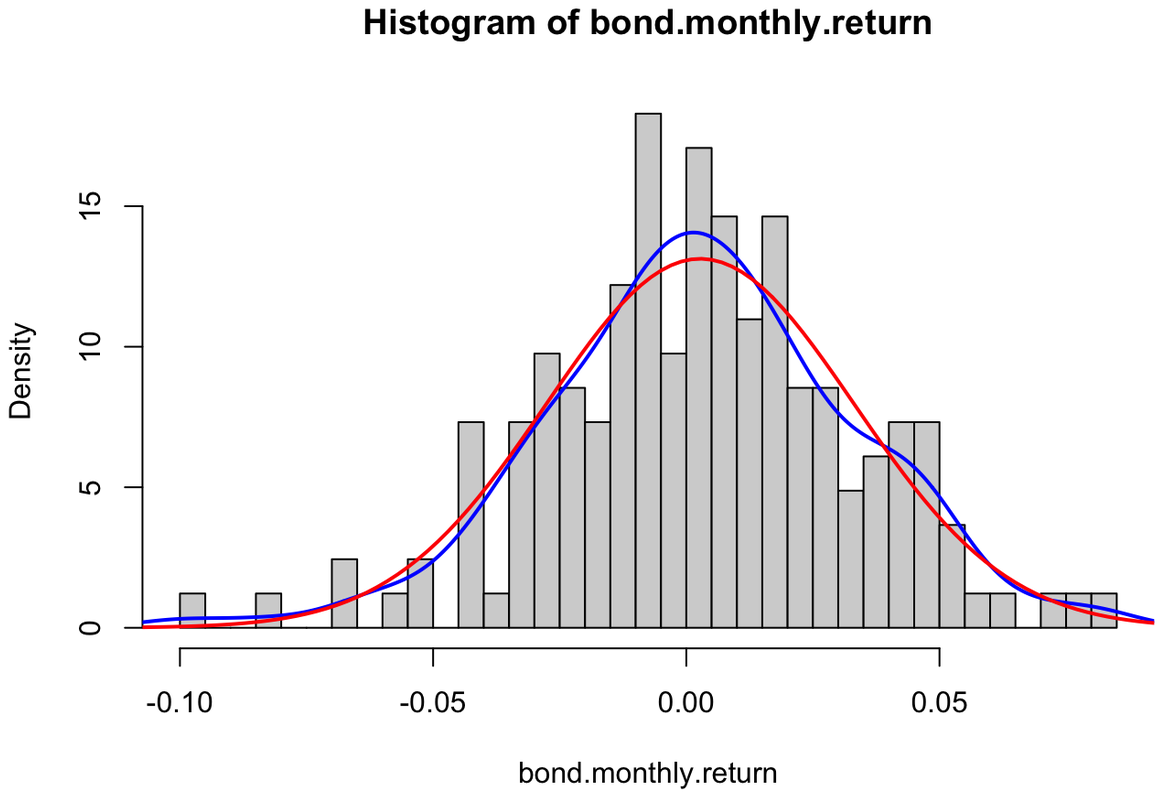
```
plot.zoo(bond.monthly.return, main="Monthly Returns", ylim=ylim_bond)
```




```
##### Problem 1 Question C
x <- seq(-1, 1, len=1000)
hist(bond.daily.return, breaks=50, probability=TRUE, ylim=c(0, 80))
lines(density(bond.daily.return), col="blue", lwd=2)
lines(x, dnorm(x, mean=mean(bond.daily.return), sd=sd(bond.daily.return)), col="red", lwd=2)
```




```
##### Problem 1 Question D
x <- seq(-1, 1, len=1000)
hist(bond.monthly.return, breaks=30, probability=TRUE)
lines(density(bond.monthly.return), col="blue", lwd=2)
lines(x, dnorm(x, mean=mean(bond.monthly.return), sd=sd(bond.monthly.return)), col="red", lwd=2)
```



Problem 1 Question E

From observing the images we obtained, I conclude that the daily returns are not normally distributed, while the monthly returns follow a normal distribution. This conclusion can be drawn from the match-comparison of the red line (representing the probability density function of a normal distribution) and the blue line (representing the kernel density estimation of the distribution) in the previous graphs. For daily returns, the daily variations cause a lack of alignment between the blue and red lines, indicating that daily returns do not follow a normal distribution. For monthly returns, the balance of a full month tends to make the blue and red lines relatively aligned, suggesting that monthly returns do follow a normal distribution.

Besides, we can tell from the skewness and kurtosis that we observed form question A. For the Daily log return distribution, it has skewness of -0.7116300 and kurtosis of 11.27353. The skewness is a bit away from normal distribution, and the kurtosis is immensely away from normal distribution. For the Monthly log return distribution, it has skewness of -0.1675629989 and kurtosis of 3.4456900378, and both are very close to the normal distribution. Therefore, it reinforces our conclusions.

Problem 2

```
##### Problem 2 Question A
vanguard.data <- getSymbols("VWNDX", auto.assign=FALSE, from="2010-01-01", to="2023-10-01")$VWND
X.Adjusted
vanguard.daily.return <- na.omit(Return.calculate(vanguard.data, method="log"))

result_matrix <- matrix(NA, nrow = 4, ncol = 1)
rownames(result_matrix) <- c("Sample_Mean", "Sample_Variance", "Sample_Skewness", "Sample_Kurtosi
s")
colnames(result_matrix) <- "daily_log_return"

T <- length(vanguard.daily.return)
sample_mean <- sum(vanguard.daily.return) / T
sample_var <- sum((vanguard.daily.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)
sample_skew <- sum((vanguard.daily.return - sample_mean)^3) / (T * sample_sd^3)
sample_kurtosis <- sum((vanguard.daily.return - sample_mean)^4) / (T * sample_sd^4)

result_matrix[, 1] <- c(sample_mean, sample_var, sample_skew, sample_kurtosis)
result_matrix
```

```
##                 daily_log_return
## Sample_Mean           0.0004043817
## Sample_Variance       0.0001473453
## Sample_Skewness      -0.8286100710
## Sample_Kurtosis      16.7757434904
```

Problem 2 Question B

Since $X = aY + b$, thus we can plug in formulas:

$Kurt[X] = 3 + \frac{6}{v-4} = 16.7757276452$; Thus: $v = (6 + (16.7757276452 - 3) * 4)/(16.7757276452 - 3) = 4.435549$

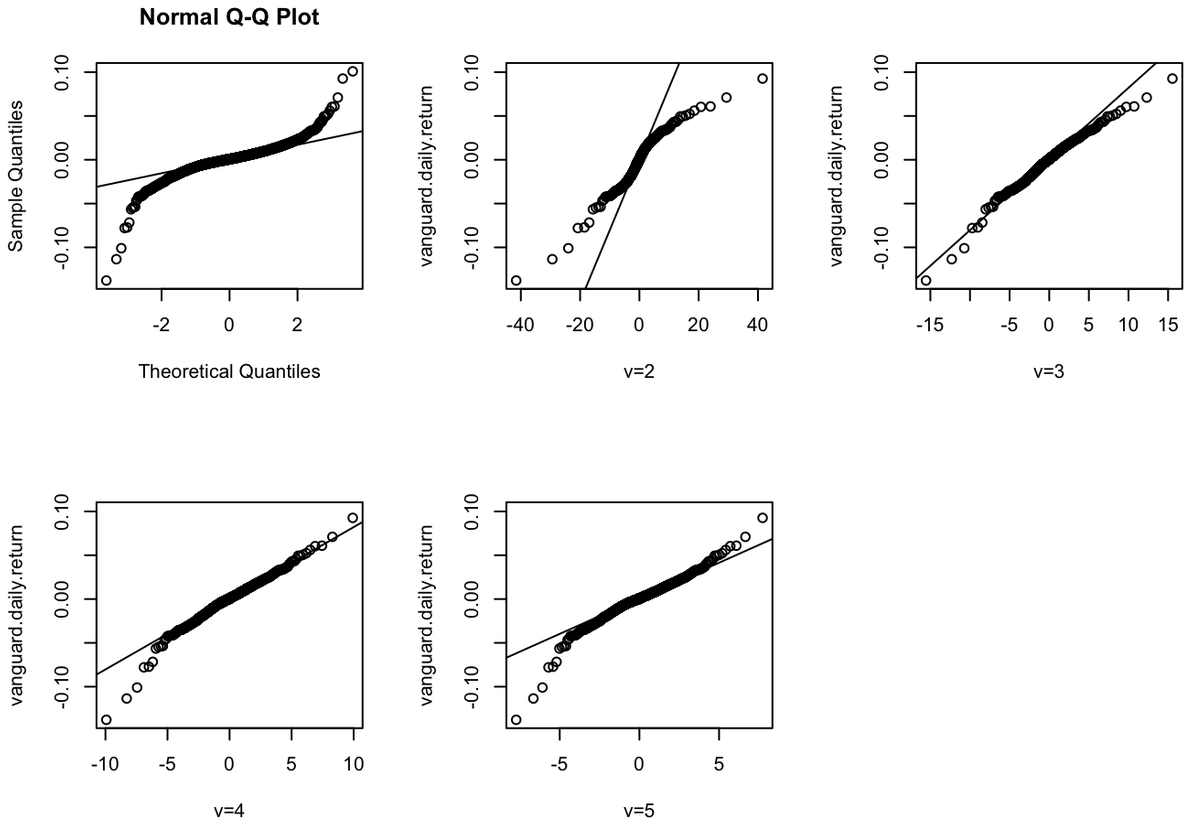
$Var[X] = \frac{a^2 v}{v-2} = 0.0001473454$; Thus, plug in v = 4.435549, we can get:
$a = \sqrt{(0.0001473454 * (4.435549 - 2))/(4.435549)} = 0.008994832$

$E[X] = b = 0.0004043817$; Thus: b = 0.0004043817

$Skew[X] = 0 \neq -0.8286060915$; Since the Students t distribution is always symmetrical, it always have Skew[X] = 0. Therefore, it is impossible to match the skewness of -0.8286082423 that we get in the previous part.

As a conclusion: v = 4.435549; a = 0.008994832; b = 0.0004043817;

```
##### Problem 2 Question C
par(mfrow=c(2,3))
qqnorm(vanguard.daily.return)
qqline(vanguard.daily.return)
vanguard.daily.return.ecdf <- ecdf(coredata(vanguard.daily.return))

plot(qt(vanguard.daily.return.ecdf(vanguard.daily.return), df=2), vanguard.daily.return, xlab="v=
2")
qqline(vanguard.daily.return)
plot(qt(vanguard.daily.return.ecdf(vanguard.daily.return), df=3), vanguard.daily.return, xlab="v=
3")
qqline(vanguard.daily.return)
plot(qt(vanguard.daily.return.ecdf(vanguard.daily.return), df=4), vanguard.daily.return, xlab="v=
4")
qqline(vanguard.daily.return)
plot(qt(vanguard.daily.return.ecdf(vanguard.daily.return), df=5), vanguard.daily.return, xlab="v=
5")
qqline(vanguard.daily.return)
```



Problem 2 Question D

Based on all the QQ plots, the plot with v=3 appears to be the best model for the daily log returns. In this plot, almost all data points closely follow the QQ line, indicating that the distribution has the same shape as the theoretical normal distribution. In this way, v=4 and v=5 also show relatively good performances, but they also exhibit relatively heavy tails compared to v=3, making them not as good. The case for v=2 is the worst because the data points do not match the QQ line at all.

This result doesn't quite align with what we obtained in question B (where v = 4.435549), which could be due to a mismatch in skewness as observed in the second question, resulting in discrepancies in the results presented in question D.

Problem 3

```
##### Problem 3
# Monthly log-returns of Vanguard long-term bond ETF (BLV)
bond.monthly.return <- na.omit(Return.calculate(to.monthly(bond.data, OHLC=FALSE), method="log"))

# Monthly log-returns of Vanguard Windsor Investor Shares Mutual Fund (VWNDX)
vanguard.monthly.return <- na.omit(Return.calculate(to.monthly(vanguard.data, OHLC=FALSE), method
="log"))

# Monthly log-returns ofNASDAQ Composite (^IXIC)
composite.data <- getSymbols("^IXIC", auto.assign=FALSE, from="2010-01-01", to="2023-10-01")$IXI
C.Adjusted
composite.daily.return <- na.omit(Return.calculate(composite.data, method="log"))
composite.monthly.return <- na.omit(Return.calculate(to.monthly(composite.data, OHLC=FALSE), meth
od="log"))

# Create Matrix
result_matrix <- matrix(NA, nrow = 2, ncol = 3)
rownames(result_matrix) <- c("Sample_Mean", "Sample_standard_deviation")
colnames(result_matrix) <- c("BLV", "VWNDX", "^IXIC")

# BLV
T <- length(bond.monthly.return)
sample_mean <- sum(bond.monthly.return) / T
sample_var <- sum((bond.monthly.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)

result_matrix[, 1] <- c(sample_mean, sample_sd)

# VWNDX
T <- length(vanguard.monthly.return)
sample_mean <- sum(vanguard.monthly.return) / T
sample_var <- sum((vanguard.monthly.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)

result_matrix[, 2] <- c(sample_mean, sample_sd)

# ^IXIC
T <- length(composite.monthly.return)
sample_mean <- sum(composite.monthly.return) / T
sample_var <- sum((composite.monthly.return - sample_mean)^2) / (T - 1)
sample_sd <- sqrt(sample_var)

result_matrix[, 3] <- c(sample_mean, sample_sd)

result_matrix
```

```
##                                BLV       VWNDX      ^IXIC
## Sample_Mean               0.00274742 0.008819521 0.01108198
## Sample_standard_deviation 0.03038393 0.048098086 0.05006360
```

Based on the result matrix we got, as the expected returns (sample mean) increases, the volatility (sample standard deviation) also increases. It is reasonable since assets with the potential for higher returns typically come with increased risks.

```
library(quantmod)
library(PerformanceAnalytics)
nvda.data <- getSymbols("NVDA", auto.assign=FALSE, from="2010-01-01", to="2023-01-01")
nvda.prices <- nvda.data$NVDA.Adjusted
nvda.returns <- coredata(na.omit(Return.calculate(nvda.prices, method="log")))
nvda.prices <- coredata(nvda.prices)
T <- length(nvda.returns)
```

Problem 4

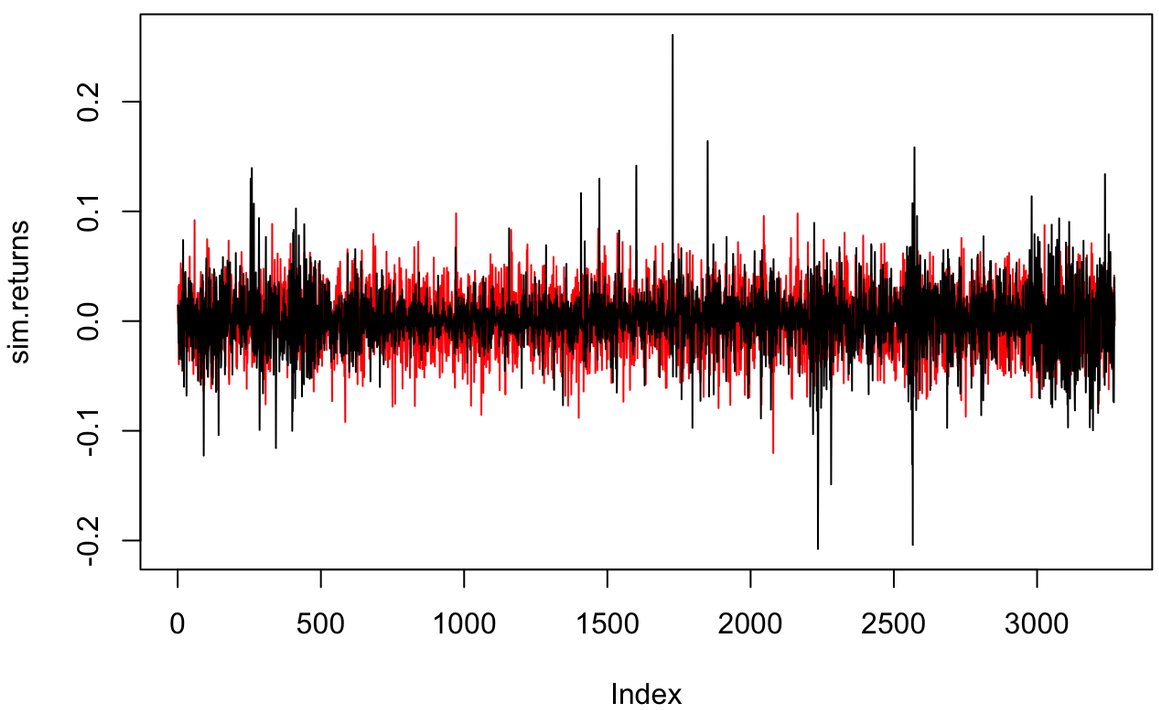
```
##### Problem 1 Question A
sample_mean <- mean(nvda.returns)
sample_sd <- sd(nvda.returns)
sample_mean
```

```
## [1] 0.001082072
```

```
sample_sd
```

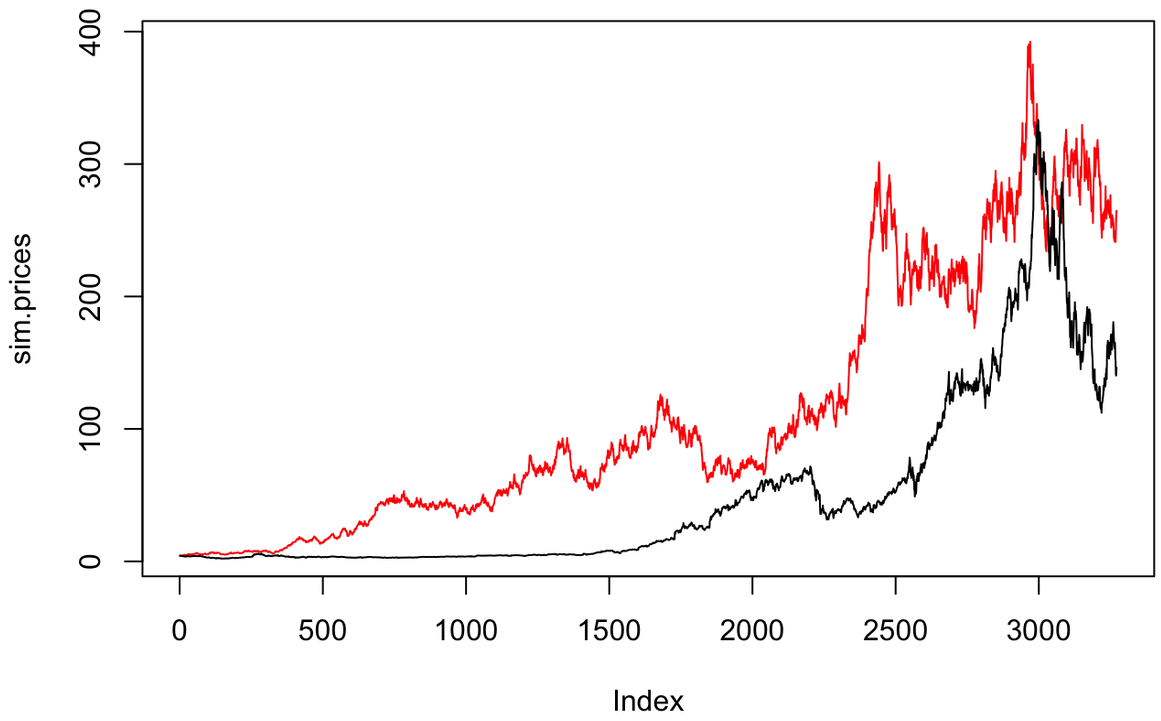
```
## [1] 0.02795697
```

```
##### Problem 1 Question B
P0 <- nvda.prices[1]
set.seed(16)
sim.returns <- rnorm(T, mean=sample_mean, sd=sample_sd)
sim.prices <- c(P0, P0 * exp(cumsum(sim.returns)))
y_limit <- c(min(sim.returns, nvda.returns), max(sim.returns, nvda.returns))

plot(sim.returns, type="l", col="red", ylim=y_limit)
lines(nvda.returns, type="l", col="black")
```



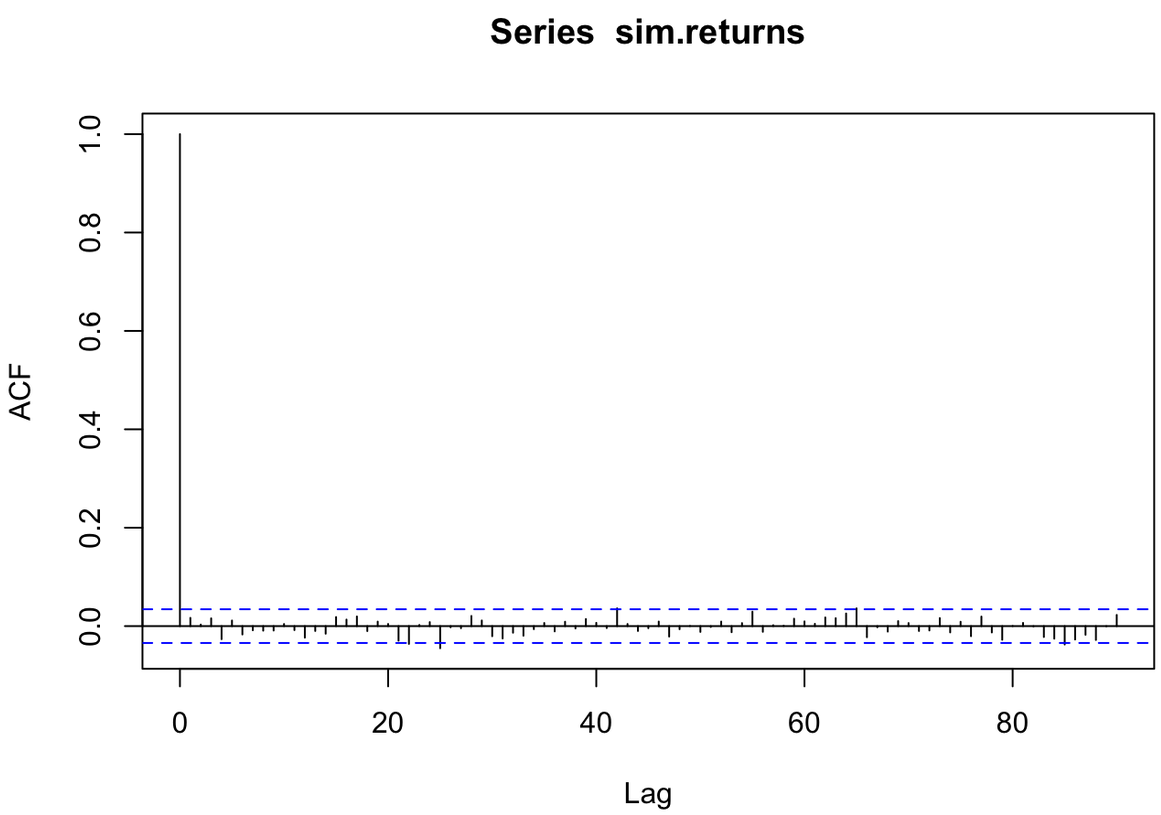
These two plots are not similar with each other. The red line which represents the simulated returns have less extreme values, while the black line which represent actual returns have larger fluctuations.

```
##### Problem 1 Question C
plot(sim.prices, type="l", col="red")
lines(nvda.prices, type="l", col="black")
```

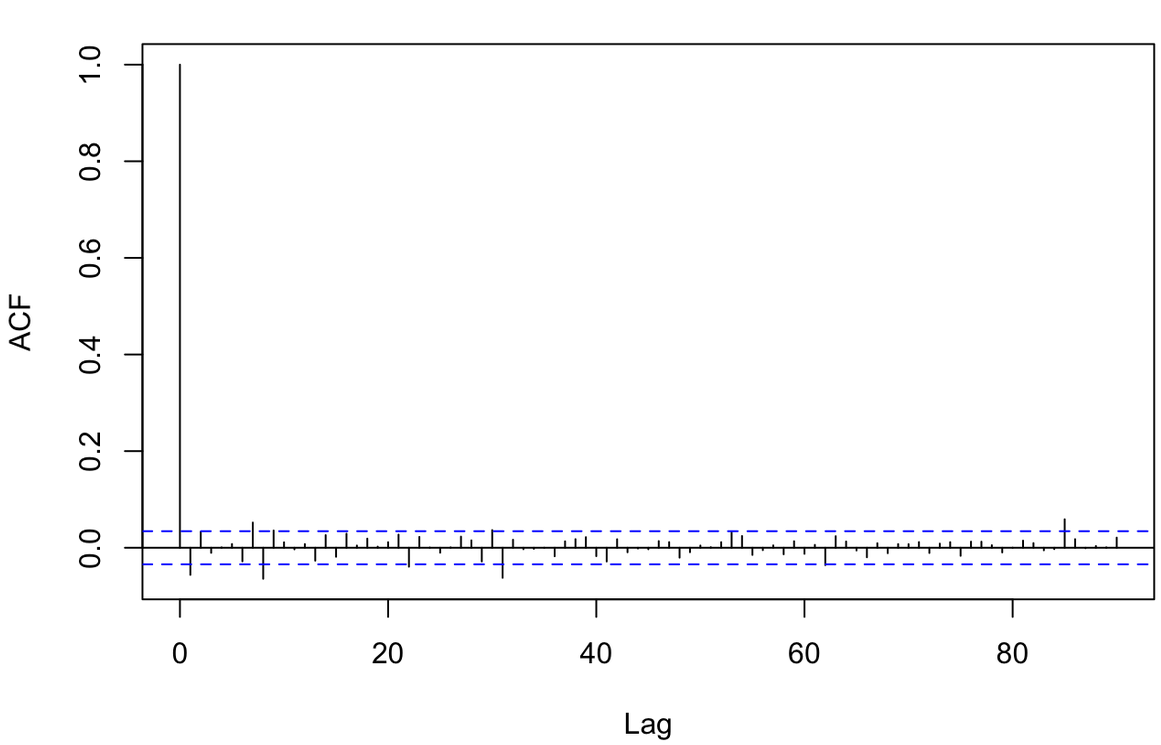


These two plots are not similar with each other. Although they are somewhat similar with an increasing trend, but there are also gaps and various fluctuations between the exact values that made the lines not that similar as well. Moreover, if we set a different seed, the randomly generated simulations may look very different.

```
##### Problem 1 Question D
acf(sim.returns,lag.max=90,pl=TRUE)
```

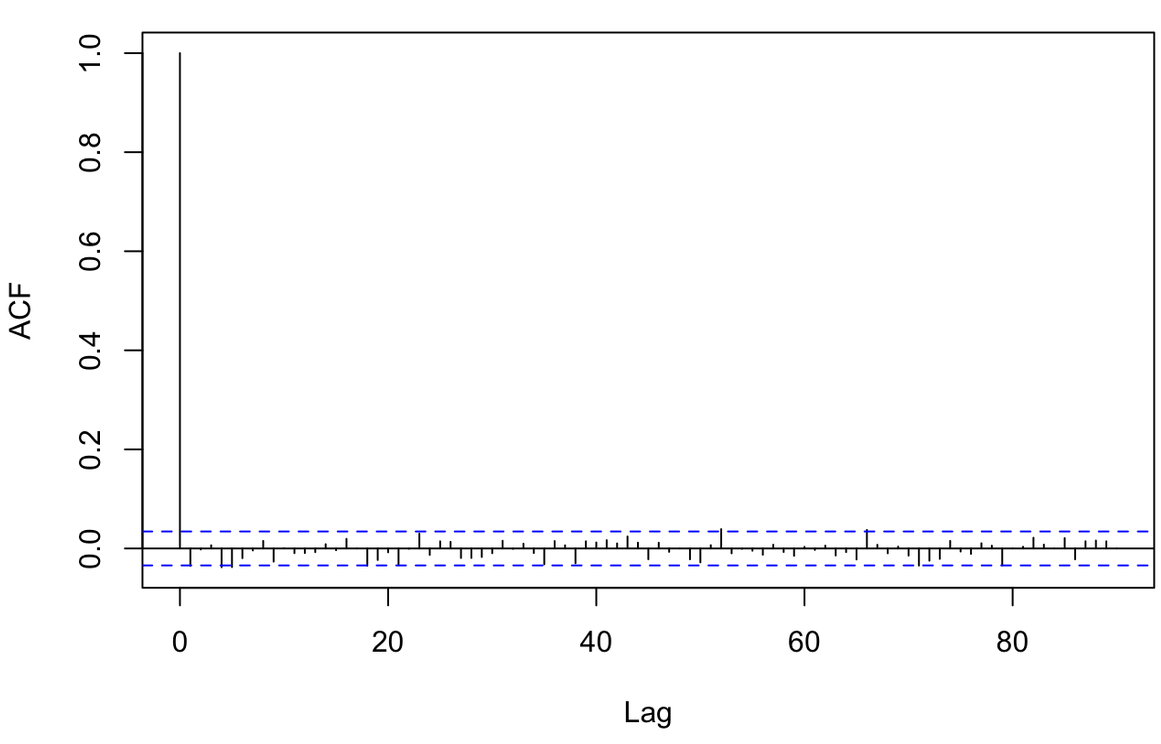



```
acf(nvda.returns,lag.max=90,pl=TRUE)
```

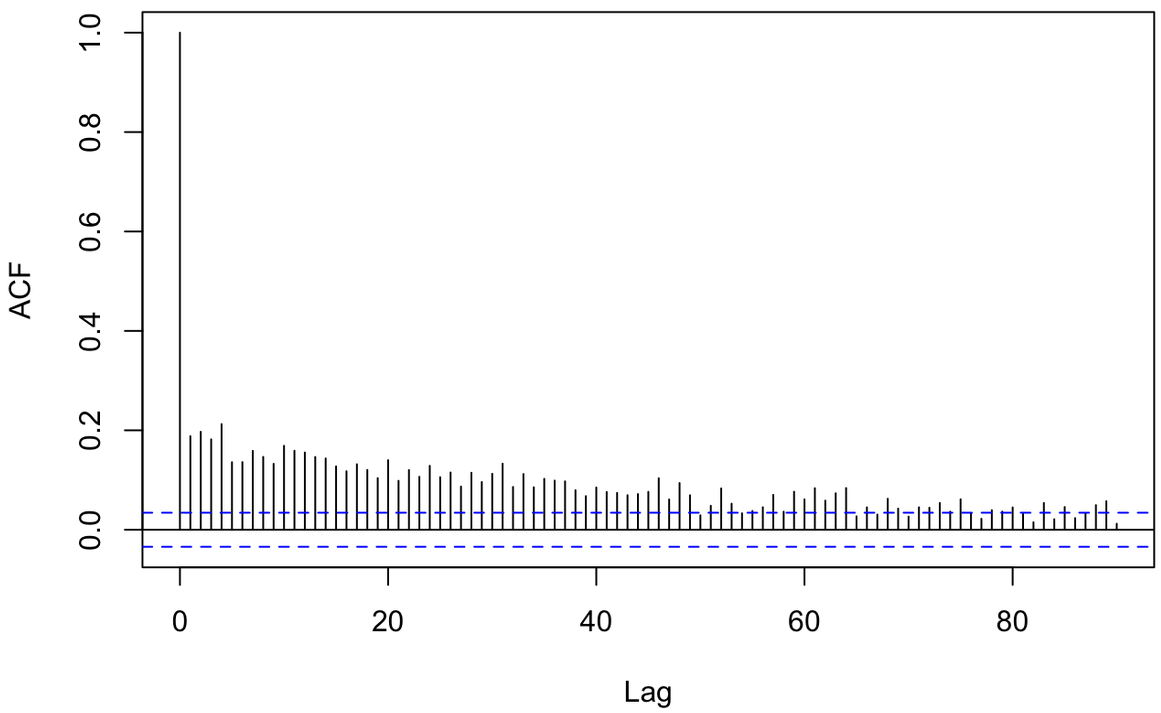


These two plots look similar to each other, as we can observed similar fluctuations and trends in both plots. The autocorrelation for both plots are very centered to zero, indicating that there is no time dependence (uncorrelated) for both plots (the values between consecutive observations are not correlated with one another).

```
##### Problem 1 Question E
acf(abs(sim.returns),lag.max=90,pl=TRUE)
```



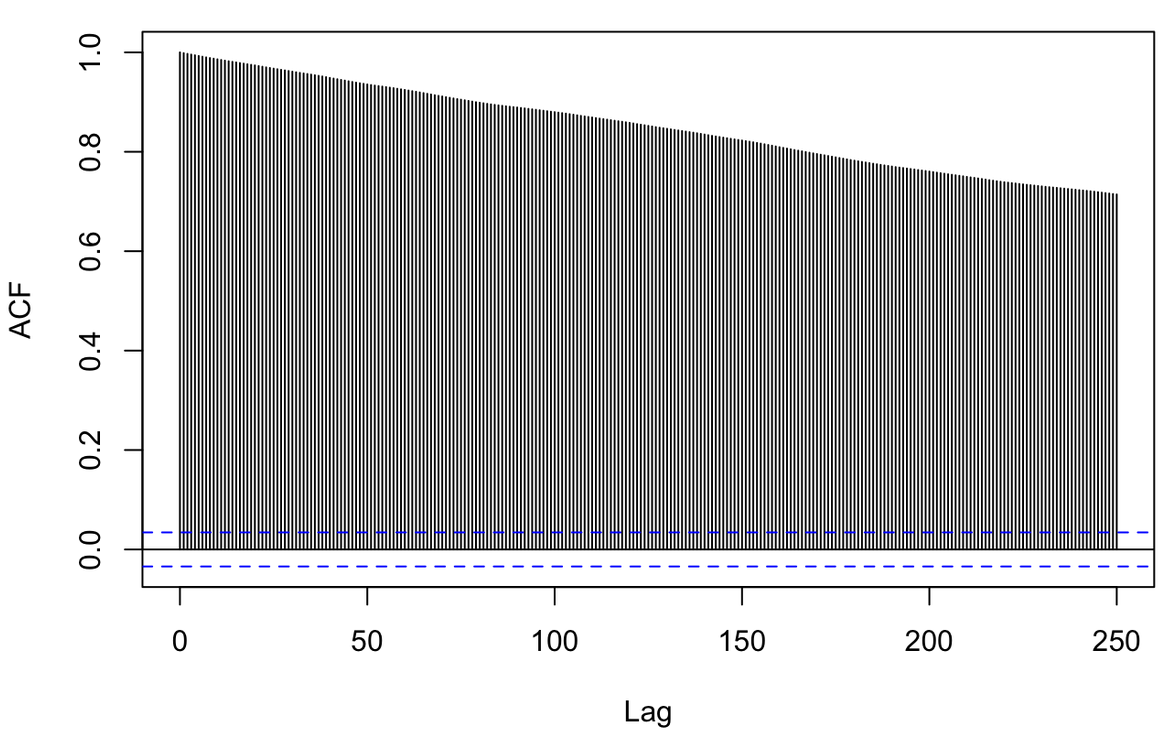

```
acf(abs(nvda.returns),lag.max=90,pl=TRUE)
```



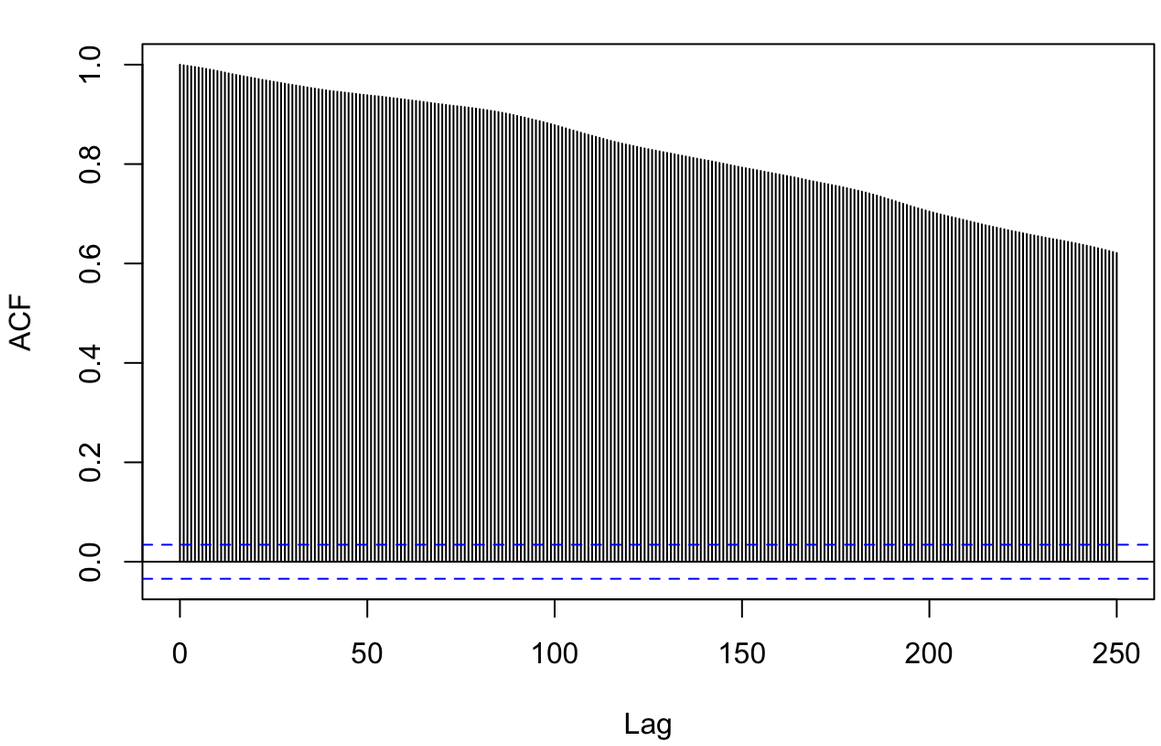
The two plots are not similar to each other. From observing the plots, we can find that for the absolute actual nvda.returns plot, as the lag increases, the autocorrelation are gradually decreasing from 0.2, which indicated that there is time dependency (correlated) for the absolute actual nvda.returns.

However, for the absolute simulated returns plot, we can also observed that the autocorrelation is very centered to zero, indicating that there is no time dependency (uncorrelated) for the absolute simulated returns.

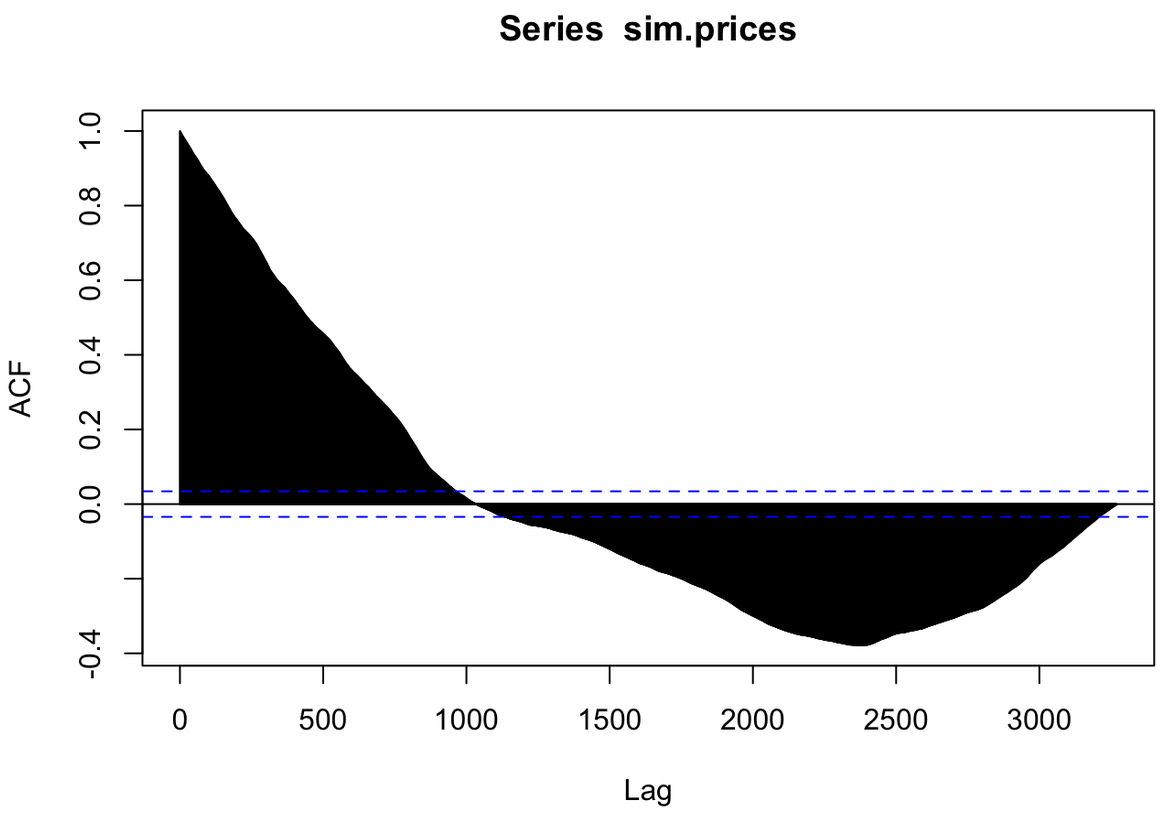
```
##### Problem 1 Question F

# For normal business analysis:
acf(sim.prices,lag.max=250,pl=TRUE)
```



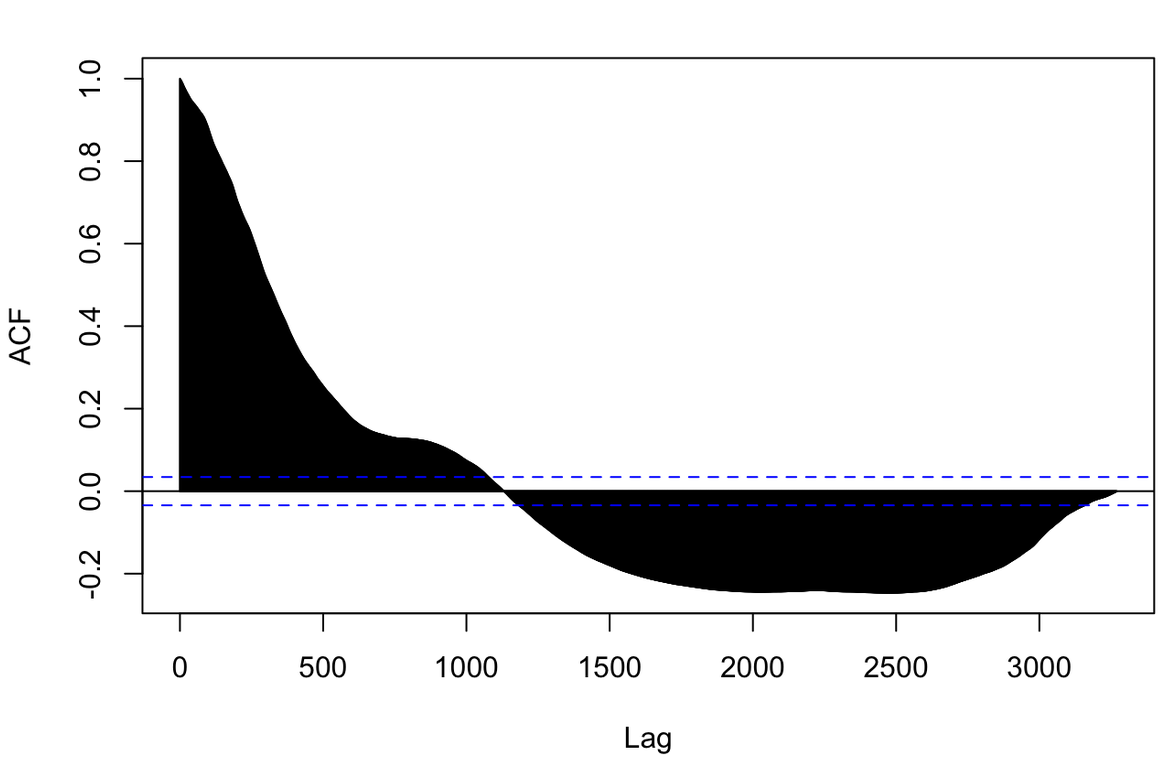

```
acf(nvda.prices,lag.max=250,pl=TRUE)
```




```
# For a comprehensive analysis:
acf(sim.prices,lag.max=T-1,pl=TRUE)
```




```
acf(nvda.prices,lag.max=T-1,pl=TRUE)
```



Both plots are very similar to each other.

For this problem, there is no need to analysis for overly large lax, so I picked lag.max = 250. It is a large enough lag for general analysis. From this plot, we can find that the autocorrelation for both simulated price and actual historical nvda price have time dependence (positive correlated).

Although there is no need to analysis for an overly large lax for it, I also created the plots as a references for it. From observing the plots, we can find that the autocorrelation was initially positive correlated. As lag increases, the autocorrelation gradually decrease to zero at lag equals 1000-1100 or so. Then, the autocorrelation become more and more negatively correlated. When the lag equals 2400-2500 or so, the autocorrelation become less negatively correlated, and gradually turn to zero (uncorrelated).

Problem 5

Followed our the covariance matrix formula:

As an example, when there is 2 different assets in our portfolio, the covariance matrix for daily log returns is

$$\Sigma_d = \begin{bmatrix} \sigma_{d1}^2 & \rho\sigma_{d1}\sigma_{d2} \\ \rho\sigma_{d1}\sigma_{d2} & \sigma_{d2}^2 \end{bmatrix}$$

The correlation matrix is

$$Corr_d = \frac{\Sigma_d}{\sigma_{d1}\sigma_{d2}} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

As our conversion formula mentioned in our lecture is:

$\sigma_m^2 = 21\sigma_d^2$; $\sigma_m = \sqrt{21}\sigma_d$; $\sigma_y^2 = 252\sigma_d^2$; $\sigma_y = \sqrt{252}\sigma_d$;

Therefore, the covariance matrix for monthly log returns is

$$\begin{aligned} \Sigma_m &= \begin{bmatrix} (\sqrt{21}\sigma_{d1})^2 & \rho(\sqrt{21}\sigma_{d1})(\sqrt{21}\sigma_{d2}) \\ \rho(\sqrt{21}\sigma_{d1})(\sqrt{21}\sigma_{d2}) & (\sqrt{21}\sigma_{d2})^2 \end{bmatrix} \\ &= \begin{bmatrix} 21\sigma_{d1}^2 & 21\rho\sigma_{d1}\sigma_{d2} \\ 21\rho\sigma_{d1}\sigma_{d2} & 21\sigma_{d2}^2 \end{bmatrix} \\ &= 21\begin{bmatrix} \sigma_{d1}^2 & \rho\sigma_{d1}\sigma_{d2} \\ \rho\sigma_{d1}\sigma_{d2} & \sigma_{d2}^2 \end{bmatrix} \\ &= 21\Sigma_d \end{aligned}$$

and the covariance matrix for yearly log returns is

$$
\begin{aligned}
\Sigma_y &= \begin{bmatrix} (\sqrt{252}\sigma_{d1})^2 & \rho(\sqrt{252}\sigma_{d1})(\sqrt{252}\sigma_{d2}) \\ \rho(\sqrt{252}\sigma_{d1})(\sqrt{252}\sigma_{d2}) & (\sqrt{252}\sigma_{d2})^2 \end{bmatrix} \\
&= \begin{bmatrix} 252\sigma_{d1}^2 & 252\rho\sigma_{d1}\sigma_{d2} \\ 252\rho\sigma_{d1}\sigma_{d2} & 252\sigma_{d2}^2 \end{bmatrix} \\
&= 252 \begin{bmatrix} \sigma_{d1}^2 & \rho\sigma_{d1}\sigma_{d2} \\ \rho\sigma_{d1}\sigma_{d2} & \sigma_{d2}^2 \end{bmatrix} \\
&= 252\Sigma_d
\end{aligned}
$$

As a result, the correlation matrix for monthly log returns is:

$$
\begin{aligned}
\text{Corr}_m &= \frac{\Sigma_m}{\sigma_{m1}\sigma_{m2}} \\
&= \frac{21\Sigma_d}{(\sqrt{21}\sigma_{d1})(\sqrt{21}\sigma_{d2})} \\
&= \frac{21\Sigma_d}{21\sigma_{d1}\sigma_{d2}} \\
&= \text{Corr}_d = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}
\end{aligned}
$$

Similarly, the correlation matrix for yearly log returns is:

$$
\begin{aligned}
\text{Corr}_y &= \frac{\Sigma_y}{\sigma_{y1}\sigma_{y2}} \\
&= \frac{252\Sigma_d}{(\sqrt{252}\sigma_{d1})(\sqrt{252}\sigma_{d2})} \\
&= \frac{252\Sigma_d}{252\sigma_{d1}\sigma_{d2}} \\
&= \text{Corr}_d = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}
\end{aligned}
$$

Therefore, daily log returns, monthly log returns and yearly log returns have the same correlation matrices. Expanding to assume there are N different assets in our portfolio, we can find the patterns of $\text{Corr}[\mathbf{X}]_{ij} = \frac{Cov[\mathbf{X}]_{ij}}{\sqrt{Cov[\mathbf{X}]_{ii}Cov[\mathbf{X}]_{jj}}}$ and $\text{Corr}[\mathbf{R}^{(daily)}]_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}$. Besides, the $\Sigma_m = 21\Sigma_d$ and $\Sigma_y = 252\Sigma_d$;

Therefore, we can get $\text{Corr}[\mathbf{R}^{(monthly)}]_{ij} = \frac{21\Sigma_{d_{ij}}}{\sqrt{21\Sigma_{d_{ii}}21\Sigma_{d_{jj}}}}$ = $\text{Corr}[\mathbf{R}^{(yearly)}]_{ij} = \frac{252\Sigma_{d_{ij}}}{\sqrt{252\Sigma_{d_{ii}}252\Sigma_{d_{jj}}}}$

After simplifying these formulas, we can find there are all exactly the same: $\text{Corr}[\mathbf{R}^{(daily)}]_{ij}$ = $\text{Corr}[\mathbf{R}^{(monthly)}]_{ij}$ = $\text{Corr}[\mathbf{R}^{(yearly)}]_{ij}$

As a conclusion, even for N assets, the correlation matrices for the daily log returns, monthly log returns and yearly log returns are all the same.

Problem 6

```
##### Problem 3 Question A
n.sims <- 1000000
given_mean <- 1
given_sd <- 0.5

ln_distribution <- rlnorm(n.sims, meanlog=given_mean, sdlog=given_sd)

sample_mean <- mean(ln_distribution)
exact_mean <- exp(given_mean + (given_sd)^2 / 2)
p3qa_differences <- abs(sample_mean - exact_mean)
sample_mean
```

```
## [1] 3.077143
```

```
exact_mean
```

```
## [1] 3.080217
```

```
p3qa_differences
```

```
## [1] 0.003073936
```

```
##### Problem 3 Question B
sample_var <- var(ln_distribution)
exact_var <- (exp(given_sd^2) - 1) * (exp(2*given_mean + given_sd^2))
p3qb_differences <- abs(sample_var - exact_var)
sample_var
```

```
## [1] 2.686881
```

```
exact_var
```

```
## [1] 2.694758
```

```
p3qb_differences
```

```
## [1] 0.007877401
```

```
##### Problem 3 Question C
sample_skew <- skewness(ln_distribution)
exact_skew <- (exp(given_sd^2) + 2) * sqrt(exp(given_sd^2) - 1)
p3qc_differences <- abs(sample_skew - exact_skew)
sample_skew
```

```
## [1] 1.737529
```

```
exact_skew
```

```
## [1] 1.75019
```

```
p3qc_differences
```

```
## [1] 0.0126602
```

```
##### Problem 3 Question D
sample_kurt <- kurtosis(ln_distribution) + 3
exact_kurt <- exp(4 * given_sd^2) + 2 * exp(3 * given_sd^2) + 3 * exp(2 * given_sd^2) -3
p3qd_differences <- abs(sample_kurt - exact_kurt)
sample_kurt
```

```
## [1] 8.761879
```

```
exact_kurt
```

```
## [1] 8.898446
```

```
p3qd_differences
```

```
## [1] 0.1365664
```