# HW7

Siwei Tang

3/23/2022

## Q1

### (a)

```
amazon = read.csv("AMZN_March16_2020_March16_2022.csv")
google = read.csv("GOOGL_March16_2020_March16_2022.csv")
apple = read.csv("AAPL_March16_2020_March16_2022.csv")
nasdaq = read.csv("Nasdaq_March16_2000_March 16_2022.csv")
```

```
# calculate the closing price
n = dim(nasdaq)[1]
fix_return = 0.025/253
EX_R_nasdaq = nasdaq$Adj.Close[2:n]/ nasdaq$Adj.Close[1:n - 1] - 1 - fix_return
EX_R_apple = apple$Adj.Close[2:n]/apple$Adj.Close[1:n-1] - 1- fix_return
EX_R_amazon = amazon$Adj.Close[2:n]/amazon$Adj.Close[1:n-1] - 1 - fix_return
EX_R_google = google$Adj.Close[2:n]/google$Adj.Close[1:n-1] - 1 - fix_return
```

```
fit_apple = lm(EX_R_apple ~ EX_R_nasdaq)
fit_amazon = lm(EX_R_amazon ~ EX_R_nasdaq)
fit_google = lm(EX_R_google ~ EX_R_nasdaq)
model_names = c("fit_apple","fit_amazon","fit_google")
list_models = lapply(model_names, get)
alpha_list = rep(0,3)
beta_list = rep(0,3)
R2_list = rep(0,3)

# iterate over the fit results
for (i in 1:3){
  fit = list_models[[i]]
  alpha = fit$coefficients[1]
  beta = fit$coefficients[2]
  r = summary(fit)$r.squared
  # store results
  alpha_list[i] = alpha
  beta_list[i] = beta
  R2_list[i] = r
}
res_df = data.frame(alpha = alpha_list,
                    beta = beta_list,
                    precent_mkt_risk = R2_list,
                    row.names = c("apple","amazon","google")
                    )
```

```
print(res_df)
```

```
##                alpha      beta precent_mkt_risk
## apple  6.296803e-04 1.0851737        0.6689344
## amazon 1.334689e-06 0.9620414        0.5133706
## google 5.987528e-04 0.9623872        0.6379769
```

##(2)

```
var_F = var(EX_R_nasdaq)
epsilon_mat = diag(c(var(summary(fit_apple)$residuals),
                     var(summary(fit_amazon)$residuals),
                     var(summary(fit_google)$residuals)))
est_cov = as.matrix(beta_list,3,1)%*%var_F%*%as.vector(beta_list) + epsilon_mat
cat("Estimated cov among three stocks, \n")
```

```
## Estimated cov among three stocks,
```

```
print(est_cov)
```

```
##              [,1]         [,2]         [,3]
## [1,] 0.0004286985 0.0002542319 0.0002543233
## [2,] 0.0002542319 0.0004390293 0.0002254658
## [3,] 0.0002543233 0.0002254658 0.0003535344
```

```
emp_cov = matrix(rep(0,9),3,3)
a = EX_R_apple
b = EX_R_amazon
c = EX_R_google
emp_cov[1,] = c(var(a), cov(a,b),cov(a,c))
emp_cov[2,] = c(cov(a,b), cov(b,b),cov(b,c))
emp_cov[3,] = c(cov(a,c), cov(c,b),cov(c,c))
cat("Empirical covariance of return \n")
```

```
## Empirical covariance of return
```

```
print(emp_cov)
```

```
##              [,1]         [,2]         [,3]
## [1,] 0.0004286985 0.0002659781 0.0002432843
## [2,] 0.0002659781 0.0004390293 0.0002323363
## [3,] 0.0002432843 0.0002323363 0.0003535344
```

The estimated result is very close to the empirical,as the estimates 2-3 sigificant figures. The diagnal is exactly the same, which means the estimated variance is the the true variance. # Q2 ## (a)

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
```

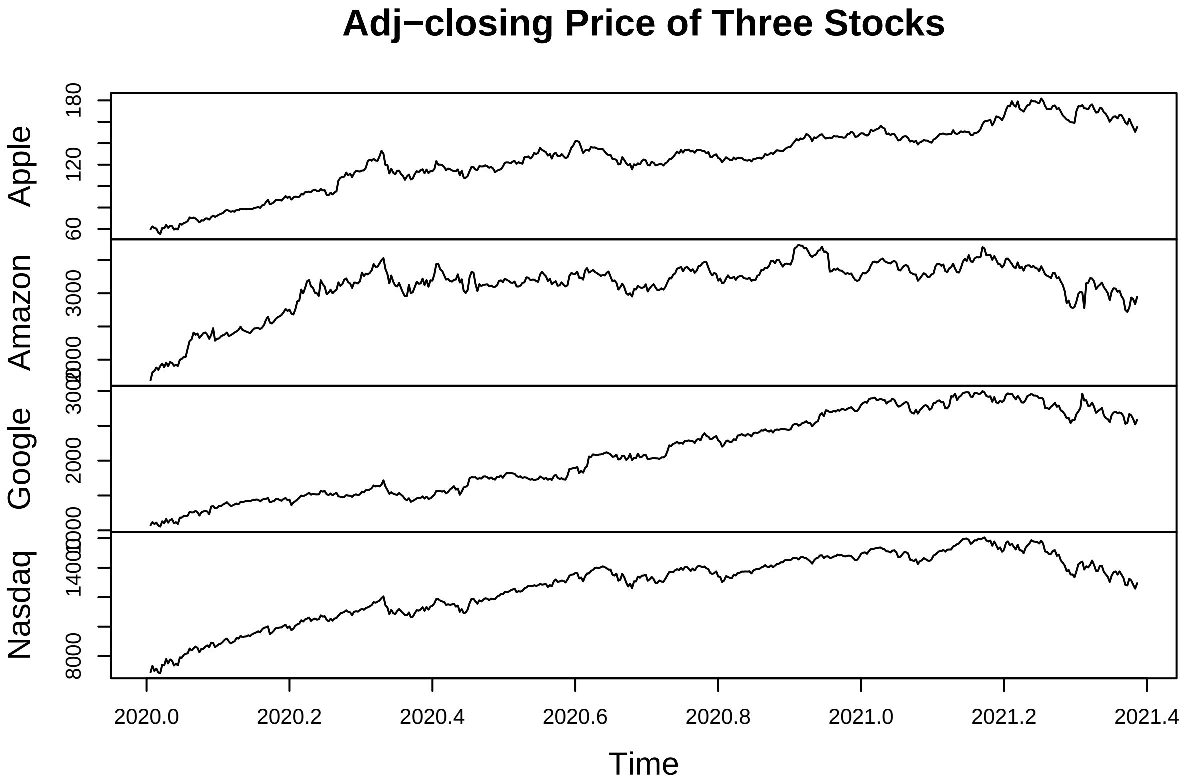
```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

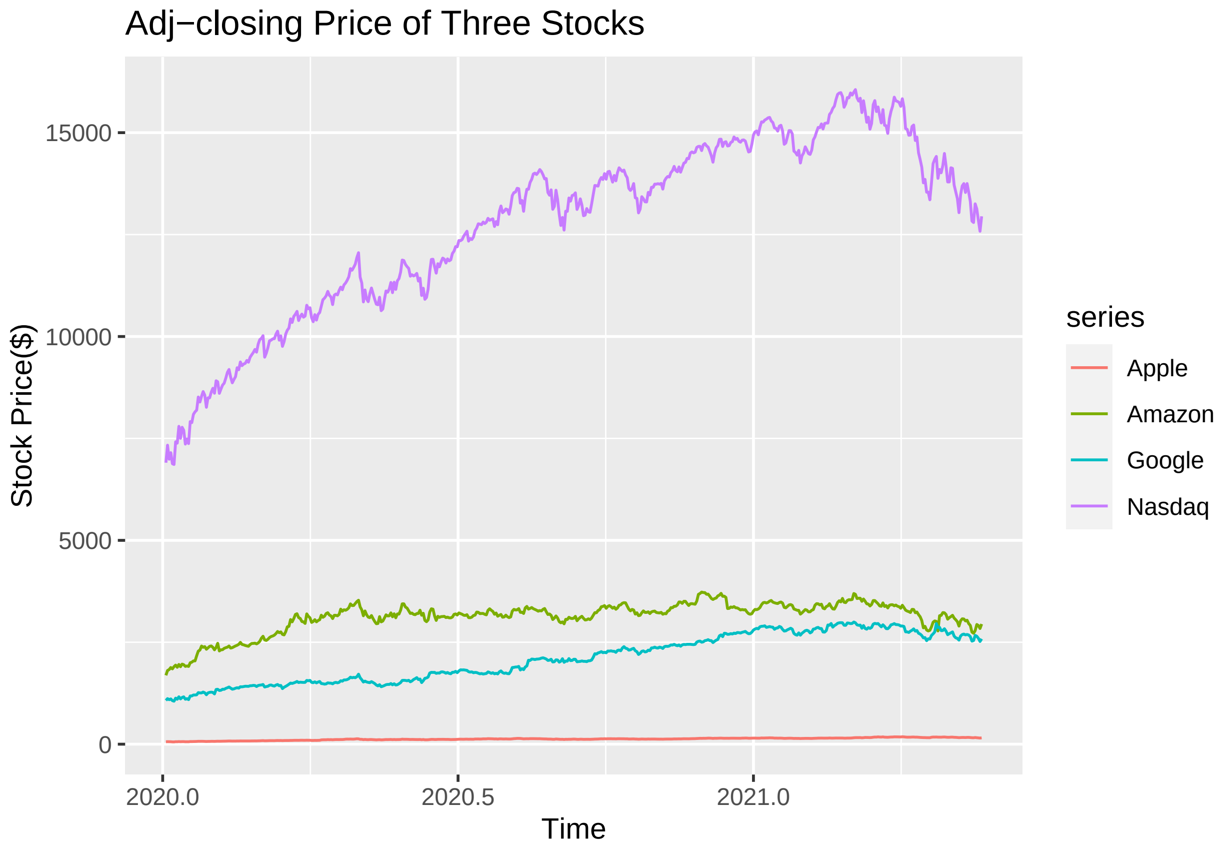
```
library(forecast)
```

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

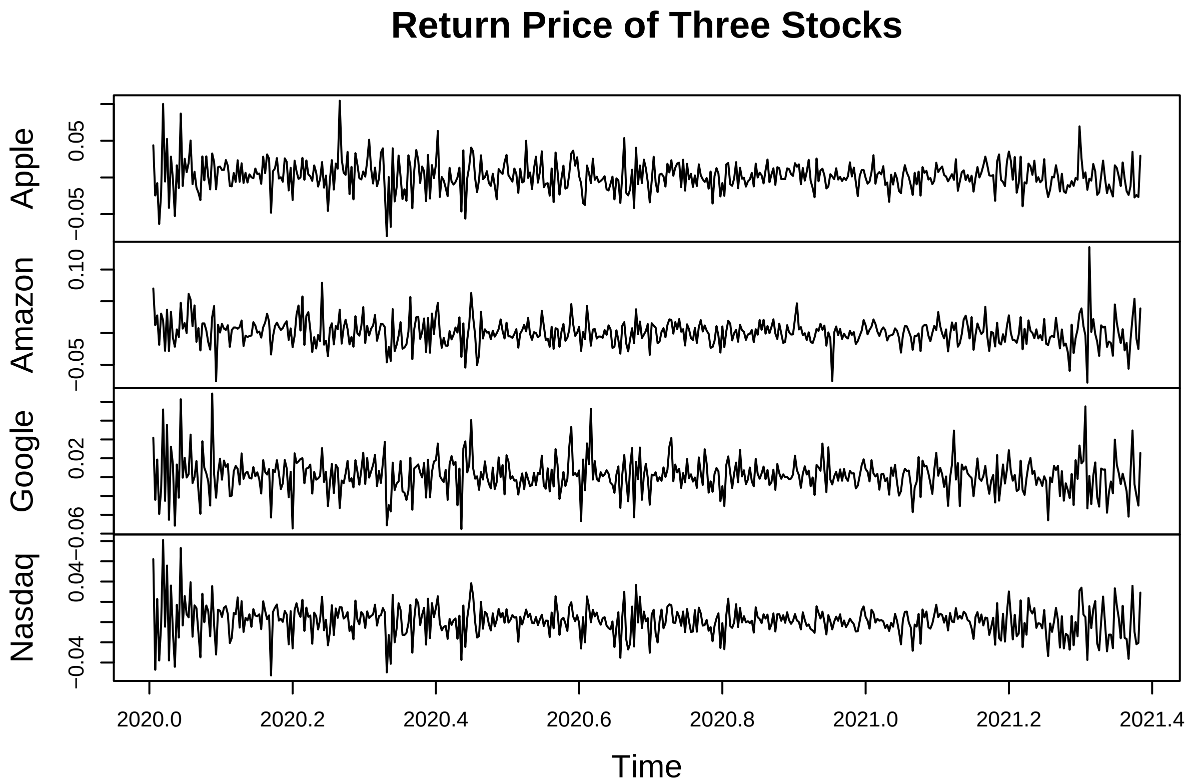
```
data_mat_close = data.frame(Apple = apple$Adj.Close,
                            Amazon = amazon$Adj.Close,
                            Google = google$Adj.Close,
                            Nasdaq = nasdaq$Adj.Close)
close_ts = ts(data = data_mat_close,
              start = c(2020,3,16),
              frequency = 365)
plot(close_ts,
     main = "Adj-closing Price of Three Stocks",  ylab ="Stock Price($)")
```



```
autoplot(close_ts,
     main = "Adj-closing Price of Three Stocks") +  ylab("Stock Price($)")
```

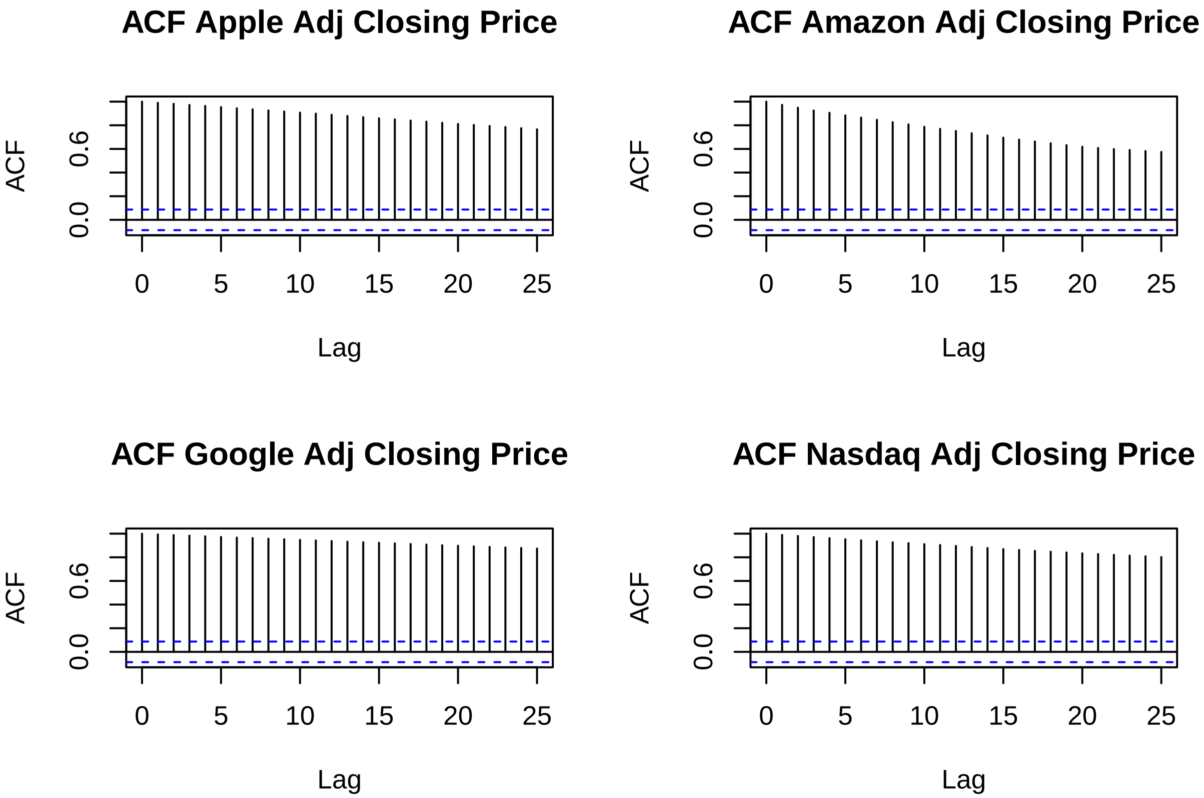


```
data_mat_return = data.frame(Apple = EX_R_apple,Amazon = EX_R_amazon,Google = EX_R_google,Nasdaq = EX_R_
close_ts = ts(data = data_mat_return,
              start = c(2020,3,16),
              frequency = 365)
plot(close_ts,
     main = "Return Price of Three Stocks"
     )
```



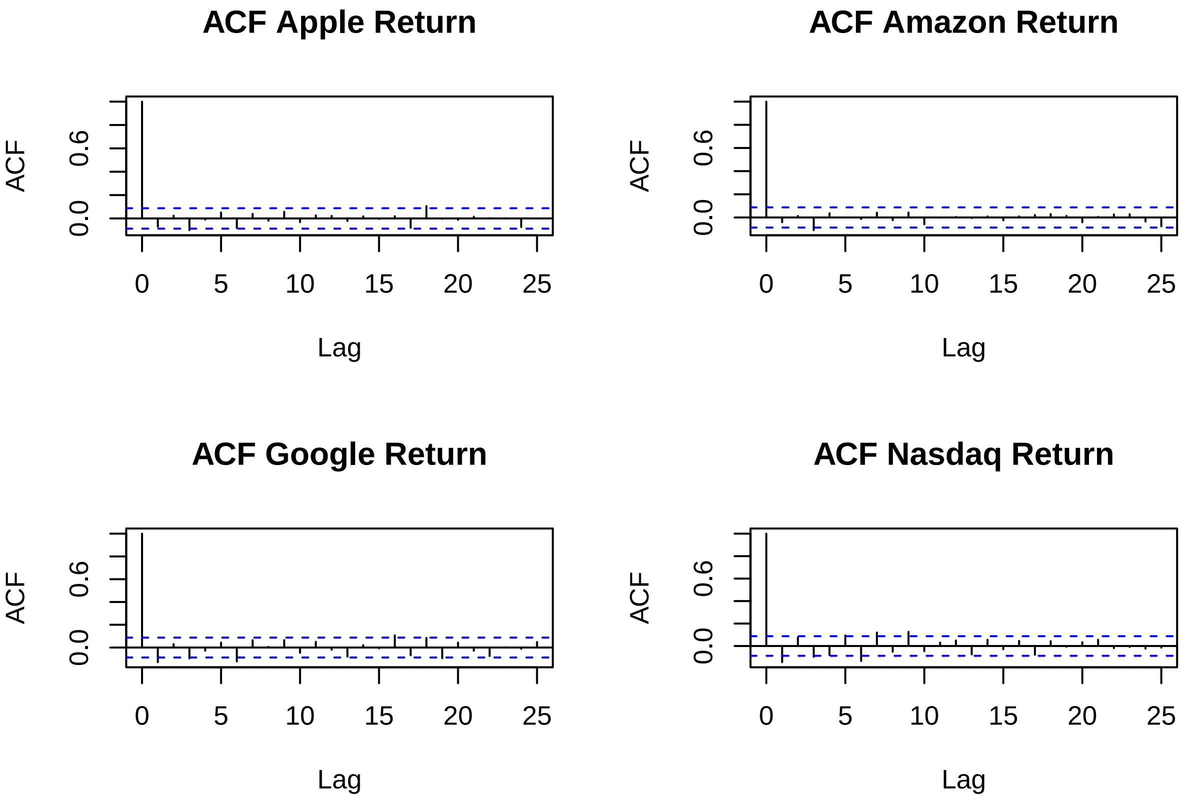
As for returns, they seems to have similar volatility, as their variance are similiar in the same time periods. As for the stock prices, they are quite similiar in the early periods, but differ when t approximates to infinity. ## (b)

```
par(mfrow = c(2,2))
acf(as.vector(data_mat_close$Apple), lag =25, main  = "ACF Apple Adj Closing Price")
acf(as.vector(data_mat_close$Amazon), lag = 25, main = "ACF Amazon Adj Closing Price")
acf(as.vector(data_mat_close$Google), lag = 25, main = "ACF Google Adj Closing Price")
acf(as.vector(data_mat_close$Nasdaq), lag = 25, main = "ACF Nasdaq Adj Closing Price")
```



There is no stationarity for any of the four adj-closing prices, because the auto-correlation doesn't decay to zero when lag<=25.

```
par(mfrow=c(2,2))
acf(as.vector(data_mat_return$Apple), lag =25, main  = "ACF Apple Return")
acf(as.vector(data_mat_return$Amazon), lag = 25, main = "ACF Amazon Return")
acf(as.vector(data_mat_return$Google), lag = 25, main = "ACF Google Return")
acf(as.vector(data_mat_return$Nasdaq), lag = 25, main = "ACF Nasdaq Return")
```



The Returns for individual stocks are generally stationary, though Google's return have some violations when lag = 1,6 and 16. For Nasdaq, the stationarity doesn't maifest itself until lag = 10.