Assignment 1

Siddharth Nilakhe

2023-10-02

PART 1 1.1

```r
#PART 1
set.seed(123)
#Creating two vectors with 10 rn

vector_1 <- runif(10, min = 0, max = 1)
vector_2 <- runif(10, min = 0, max = 1)

#Appending the second vector to the first one

new_vector <- c(vector_1, vector_2)

print(new_vector)
```

```
##  [1] 0.28757752 0.78830514 0.40897692 0.88301740 0.94046728 0.04555650
##  [7] 0.52810549 0.89241904 0.55143501 0.45661474 0.95683335 0.45333416
## [13] 0.67757064 0.57263340 0.10292468 0.89982497 0.24608773 0.04205953
## [19] 0.32792072 0.95450365
```

```r
#Calculating the mean of the new vector

mean_ <- mean(new_vector)

#We will print 'True' for values greater than the mean, 'False' if it is lesser than the mean

output <- ifelse(new_vector > mean_, 'True', 'False')

#Output
print(output)
```

```
##  [1] "False" "True"  "False" "True"  "True"  "False" "False" "True"  "True"
## [10] "False" "True"  "False" "True"  "True"  "False" "True"  "False" "False"
## [19] "False" "True"
```

```r
######################################################
```

1.2

```r
#1.2

set.seed(456)

#Creating a vector with 100 random numbers

vector_ <- runif(100, min = 0, max = 1)

#Converting the vector into a 10 by 10 matrix

M <- matrix(vector_, nrow = 10, ncol = 10)

#Finding the transposed matrix

M_t <- t(M)

#Printing elements as asked in the question

element_2r_1c <- M_t[2, 1]
print("Element in the second row and first column of Mt:")
```

```
## [1] "Element in the second row and first column of Mt:"
```

```r
print(element_2r_1c)
```

```
## [1] 0.3729459
```

```r
#Calculating the inner product to create matrix N

#Intitializing N first as an empty matrix
N <- matrix(0, nrow = 10, ncol = 10)

#Calculating
for (i in 1:10) {
  for (j in 1:10) {
    N[i, j] <- sum(M_t[i,] * M[,j])
  }
}

#Calculating the inner product using the %*% operator

N_2 <- M_t %*% M

# Comparing both the results by if-else

if (identical(N, N_2)) {
  cat("Same results for N.\n")
} else {
  cat("Different results for N.\n")
}
```

```
## Different results for N.
```

```r
##################################################################
```

1.3

```r
library(ggplot2)
#Directory
setwd("C:/Users/siddh/Downloads/try")

#CSV file
data <- read.csv("stock_data-1.csv")

#Deleting the columns containing NA
data <- data[ , -c(match("CRM", names(data)), match("DOW", names(data)), match("GS", names(data)), match("V", nam
es(data)))]

head(data)
```

```
##            X     AAPL     AMGN      AXP       BA      CAT     CSCO      CVX
## 1 1996-01-02 0.286830 14.56250 12.10832 39.93750 14.87500 4.243055 26.43750
## 2 1996-01-03 0.286830 14.40625 12.10832 39.56250 15.12500 4.076389 26.50000
## 3 1996-01-04 0.281808 13.78125 11.99890 38.56250 15.00000 3.923611 27.25000
## 4 1996-01-05 0.305804 14.09375 11.96243 39.25000 15.25000 3.972222 27.68750
## 5 1996-01-08 0.309152 13.85938 11.96243 40.12500 15.18750 3.934028 27.81250
## 6 1996-01-09 0.292411 13.53125 11.78008 39.67969 14.78125 3.631944 27.92188
##        DIS        HD      IBM     INTC      JNJ      JPM       KO     MCD
## 1 20.01773 10.527778 22.71875 7.328125 21.06250 19.58333 18.75000 22.7500
## 2 20.14104 10.333333 22.31250 7.218750 21.90625 19.58333 18.90625 22.7500
## 3 19.89442 10.305555 21.71875 7.187500 21.68750 18.75000 18.75000 22.8750
## 4 20.26435 10.055555 22.15625 7.187500 21.68750 18.66667 18.65625 22.5000
## 5 20.38767  9.777778 22.28125 7.203125 21.96875 18.66667 18.78125 22.5625
## 6 20.41850  9.666667 21.68750 6.875000 22.09375 18.20833 18.53125 22.1875
##        MMM     MRK     MSFT      NKE       PG     TRV      UNH       VZ     WBA
## 1 33.87500 32.1250 5.609375 4.445313 20.78125 28.2500 8.078125 30.46456 7.53125
## 2 33.81250 31.6875 5.429688 4.312500 21.40625 28.6250 8.109375 31.42009 7.50000
## 3 33.68750 31.8750 5.460938 4.265625 21.75000 29.0000 8.187500 30.85801 7.40625
## 4 33.75000 31.5000 5.398438 4.132813 21.84375 29.0625 7.859375 31.19526 7.68750
## 5 33.50000 31.9375 5.390625 4.203125 21.93750 29.1875 7.703125 30.97043 7.62500
## 6 33.01562 31.7500 5.011719 4.117188 21.90625 28.9375 7.265625 30.93530 7.56250
##       WMT
## 1 11.6250
## 2 11.7500
## 3 11.8750
## 4 11.6875
## 5 11.6875
## 6 11.5000
```

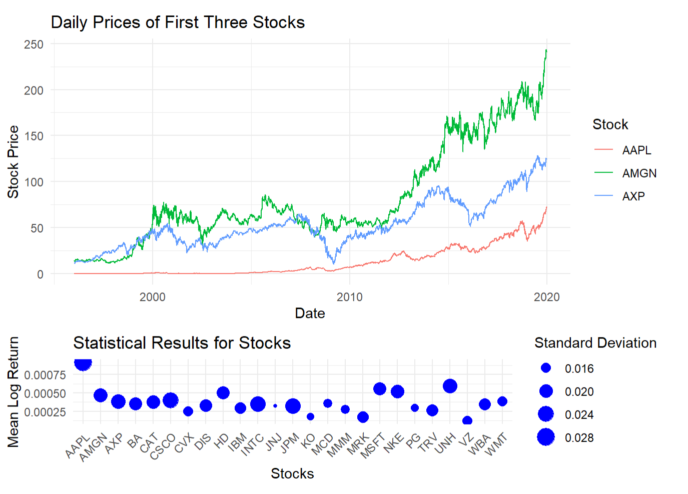
```r
cols = ncol(data)
#Daily log return for each stock
log_data = data

log_data[1,2:cols] = 0
for(i in 2:nrow(data)){
  log_data[i, 2:cols] = log(data[i,2:cols])-log(data[i-1,2:cols])
}

head(log_data)
```

```
##            X        AAPL        AMGN          AXP           BA          CAT
## 1 1996-01-02  0.00000000  0.00000000  0.000000000  0.000000000  0.000000000
## 2 1996-01-03  0.00000000 -0.01078759  0.000000000 -0.009434032  0.016667052
## 3 1996-01-04 -0.01766372 -0.04435317 -0.009077177 -0.025601398 -0.008298803
## 4 1996-01-05  0.08171839  0.02242246 -0.003044156  0.017671143  0.016529302
## 5 1996-01-08  0.01088869 -0.01676954  0.000000000  0.022048137 -0.004106782
## 6 1996-01-09 -0.05567272 -0.02396007 -0.015361271 -0.011160162 -0.027113235
##           CSCO         CVX          DIS           HD          IBM         INTC
## 1  0.000000000 0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
## 2 -0.040071981 0.002361276  0.006141243 -0.018642404 -0.018043515 -0.015037877
## 3 -0.038199144 0.027908788 -0.012320435 -0.002691813 -0.026971117 -0.004338402
## 4  0.012313233 0.015927527  0.018424292 -0.024557852  0.019943681  0.000000000
## 5 -0.009661798 0.004504512  0.006066728 -0.028012958  0.005625894  0.002171554
## 6 -0.079895795 0.003924872  0.001510949 -0.011428684 -0.027009460 -0.046623316
##            JNJ          JPM           KO          MCD          MMM          MRK
## 1  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
## 2  0.039277776  0.000000000  0.008298803  0.000000000 -0.001846723 -0.013712262
## 3 -0.010035927 -0.043485146 -0.008298803  0.005479466 -0.003703708  0.005899722
## 4  0.000000000 -0.004454386 -0.005012542 -0.016529302  0.001853569 -0.011834458
## 5  0.012884931  0.000000000  0.006677821  0.002773927 -0.007434978  0.013793322
## 6  0.005673774 -0.024859965 -0.013400536 -0.016760169 -0.014564505 -0.005888143
##           MSFT         NKE           PG          TRV          UNH           VZ
## 1  0.000000000  0.00000000  0.000000000  0.000000000  0.000000000  0.000000000
## 2 -0.032557631 -0.03033250  0.029631798  0.013187004  0.003861009  0.030883522
## 3  0.005738896 -0.01092907  0.015930822  0.013015368  0.009587801 -0.018051105
## 4 -0.011510917 -0.03163042  0.004301082  0.002152853 -0.040901514  0.010869672
## 5 -0.001448319  0.01687001  0.004282662  0.004291852 -0.020080996 -0.007233283
## 6 -0.072882364 -0.02065789 -0.001425517 -0.008602204 -0.058471768 -0.001134952
##            WBA         WMT
## 1  0.000000000  0.00000000
## 2 -0.004158010  0.01069529
## 3 -0.012578782  0.01058211
## 4  0.037271395 -0.01591546
## 5 -0.008163311  0.00000000
## 6 -0.008230499 -0.01617286
```

```r
#Mean
mean_log_return <- apply(log_data[2:nrow(log_data), 2:cols], 2, mean)

#Standard deviation
sd_log_return <- apply(log_data[2:nrow(log_data), 2:cols], 2, sd)

#Dataframe for both
mean_sd_df <- data.frame(mean_log_return, sd_log_return)

head(mean_sd_df)
```

```
##      mean_log_return sd_log_return
## AAPL    0.0009168344    0.02840478
## AMGN    0.0004641248    0.02085579
## AXP     0.0003855638    0.02197895
## BA      0.0003478159    0.01937864
## CAT     0.0003798480    0.02049348
## CSCO    0.0004002217    0.02476240
```

```r
# Creating a subset of the first three stocks daily prices
three_stock_data <- data[, c("X", "AAPL", "AMGN", "AXP")]

library(reshape2)

melted_stock_data <- melt(three_stock_data, id.vars = "X", variable.name = "Stock", value.name = "Price")
melted_stock_data$X <- as.Date(melted_stock_data$X)

# Creating the first sub-plot
plot1 <- ggplot(data = melted_stock_data, aes(x = X, y = Price, color = Stock)) +
  geom_line() +
  labs(title = "Daily Prices of First Three Stocks",
       x = "Date",
       y = "Stock Price") +
  theme_minimal()

# Creating the second sub-plot
plot2 <- ggplot(data = mean_sd_df, aes(x = rownames(mean_sd_df), y = mean_log_return)) +
  geom_point(aes(size = sd_log_return), color = "blue") +
  labs(title = "Statistical Results for Stocks",
       x = "Stocks",
       y = "Mean Log Return",
       size = "Standard Deviation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Combining the two sub-plots
library(gridExtra)
output_plot <- grid.arrange(plot1, plot2, ncol = 1, heights = c(2, 1))
```




```r
print(output_plot)
```

```
## TableGrob (2 x 1) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
```

NEXT QUESTION

PART 2

```r
#install.packages("quantmod")
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
```

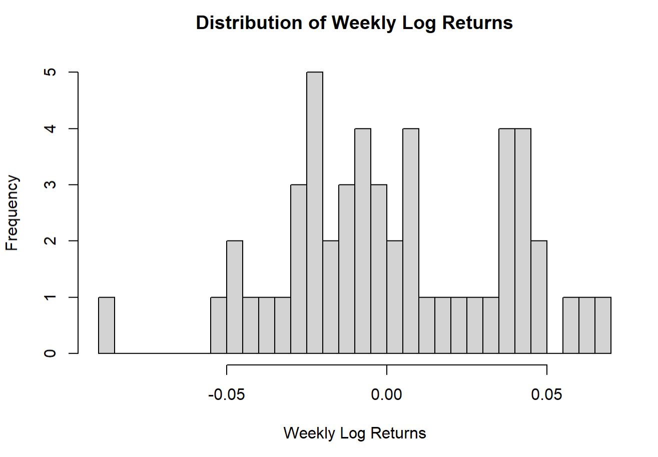
```r
library(ggplot2)

#Getting the data
start <- "2021-01-01"
end <- "2021-12-31"
getSymbols("AMZN", from = start, to = end)
```

```
## [1] "AMZN"
```

```r
#Calculating weekly log returns
weekly_log_returns <- periodReturn(AMZN, period = "weekly", col = "Adjusted", leading = FALSE)

head(weekly_log_returns)
```

```
##            weekly.returns
## 2021-01-08             NA
## 2021-01-15    -0.02464889
## 2021-01-22     0.06055571
## 2021-01-29    -0.02613122
## 2021-02-05     0.04552118
## 2021-02-12    -0.02220664
```

```r
adjusted_col_name <- colnames(weekly_log_returns)[1]

#Mean
mean_return <- mean(weekly_log_returns[, adjusted_col_name])

#Median
median_return <- median(weekly_log_returns[, adjusted_col_name])

#Standard deviation
std_return <- sd(weekly_log_returns[, adjusted_col_name])

# Step 4: Plot the distribution of weekly log returns (histogram)
hist(weekly_log_returns[, adjusted_col_name], breaks = 30, main = "Distribution of Weekly Log Returns",
     xlab = "Weekly Log Returns", ylab = "Frequency", col = "lightgray")
```




```r
# Step 5: Count observations with log returns between 0.01 and 0.015
count_ <- sum(weekly_log_returns[, adjusted_col_name] >= 0.01 & weekly_log_returns[, adjusted_col_name] <= 0.015)

# Save the data to a CSV file
write.csv(weekly_log_returns, file = "out.csv")

# Print outputs
cat("Mean Log Return:", mean_return, "\n")
```

```
## Mean Log Return: NA
```

```r
cat("Median Log Return:", median_return, "\n")
```

```
## Median Log Return: NA
```

```r
cat("Standard Deviation of Log Returns:", std_return, "\n")
```

```
## Standard Deviation of Log Returns: NA
```

```r
cat("Number of Observations between 0.01 and 0.015:", count_, "\n")
```

```
## Number of Observations between 0.01 and 0.015: NA
```

END OF THE ASSIGNMENT

NAME - SIDDHARTH PARAAG NILAKHE