R code

```
# Defining a vector A with 2 elements, my first name and last name.

A <- c("Gang Ping", "Zhu")

# Defning a vector B with 1 string which is "is working on FE513 homework".

B <- "is working on FE513 homework"
length(B)

# Split B by space and make it to vector C. Show the length of vector C (should be 5).

X <- strsplit(B, " ")
C <- unlist(strsplit(B, " "))
length(C)

# Add a period to as the 6th element of vector C.

C <- c(C, ".")
C

# Transfer vector C into a matrix D with 2 columns and 3 rows.

D <- matrix(C, ncol = 2, nrow = 3, byrow = TRUE)

# rbind A and D into a matrix E. If you read it row by row, it should be a regular sentence.
```

```
E <- rbind(A, D)
E

# Define a 5 by 5 square matrix F from 25 random number (use rnorm()).
F <- matrix(rnorm(25), ncol = 5, nrow = 5)


# Get mean and sd of the values in matrix F.
mean(F)
sd(F)


# Transposing matrix G from F.

G <- t(F)

# dot product on matrices G and F.

G * F

# multiplication on matrices G and F.

G %% F

# Convert F into a data frame H.

H <- data.frame(F)
```

```
# Displaying rows which satisfy the conditions: 1) the first column is larger than 0; AND 2) the second column is less than 0.

View(H[H$X1 > 0 & H$X2 < 0])

View(H[which(H$X1 > 0 & H$X2 < 0)])

#Setting the working directory to access the stock that was downloaded

getwd()

setwd("C:/Users/gang.ping.m.zhu/Documents/Stevens/FE513/HW")

# Read the csv file into R, and show the number of rows, number of columns and column names.

library('readxl')

data <- read.csv('atvi.csv', header = TRUE)

summary(data)

# show the number of columns separately

length(data)

row(data)



# creating a column to return the log return of the stock (log(current price/original price).

data$n <- data$Close[255]

data$logreturn <- log(data$Close/data$n)

# data$return <- log(data$n/data$Close)

#check number of NA values

sum(is.na(data$logreturn))
```

```
#check number of infinite values

sum(is.infinite(data$logreturn))


# defining a function for the SMA for the past 10 days


SMA <- function(n) {

 sum <- 0

 for (i in ((n-9):n)) {

  sum <- sum + data$logreturn[i]

 }

 return(sum/10)

}


# testing the function

SMA(255)
```

Console Output

```
> # Defining a vector A with 2 elements, my first name and last name.
>
> A <- c("Gang Ping", "Zhu")
>
> # Defning a vector B with 1 string which is "is working on FE513 homework".
>
> B <- "is working on FE513 homework"
> length(B)
[1] 1
>
> # Split B by space and make it to vector C. Show the length of vector C (sh
ould be 5).
>
> X <- strsplit(B, " ")
> C <- unlist(strsplit(B, " "))
> length(C)
[1] 5
>
>
> # Add a period to as the 6th element of vector C.
>
```

```
> C <- c(C, ".")
> C
[1] "is"       "working"  "on"       "FE513"    "homework" "."
>
> # Transfer vector C into a matrix D with 2 columns and 3 rows.
>
> D <- matrix(C, ncol = 2, nrow = 3, byrow = TRUE)
>
> # rbind A and D into a matrix E. If you read it row by row, it should be a
regular sentence.
>
> E <- rbind(A, D)
> E
  [,1]        [,2]
A "Gang Ping" "Zhu"
  "is"        "working"
  "on"        "FE513"
  "homework"  "."
>
> # Define a 5 by 5 square matrix F from 25 random number (use rnorm()).
> F <- matrix(rnorm(25), ncol = 5, nrow = 5)
>
>
> # Get mean and sd of the values in matrix F.
> mean(F)
[1] -0.2759572
> sd(F)
[1] 0.8556594
>
>
> # Transposing matrix G from F.
>
> G <- t(F)
>
> # dot product on matrices G and F.
>
> G * F
            [,1]        [,2]         [,3]        [,4]        [,5]
[1,]  0.38523633  0.08012665  0.966657427  1.07164973 -0.14702673
[2,]  0.08012665  0.24910527  0.372611232 -0.19878571 -1.49572997
[3,]  0.96665743  0.37261123  0.006917814  0.02444417 -0.02133755
[4,]  1.07164973 -0.19878571  0.024444167  1.37632135 -1.18132194
[5,] -0.14702673 -1.49572997 -0.021337551 -1.18132194  0.25295327
>
> # multiplication on matrices G and F.
>
> G %% F
            [,1]        [,2]        [,3]        [,4]        [,5]
[1,]  0.00000000 -0.08869641 -0.61157663 -0.35404824 -0.39300362
[2,] -0.01641680  0.00000000 -0.36695718  0.38586271  0.53394670
[3,] -0.35744578 -0.28149356  0.00000000  0.01553087 -0.04025894
[4,] -0.54157378 -0.04029750  0.02153251  0.00000000  0.61123698
[5,]  0.07571125 -0.47457350  0.08706581 -0.21218156  0.00000000
>
> # Convert F into a data frame H.
>
> H <- data.frame(F)
```

```
>
> # Displaying rows which satisfy the conditions: 1) the first column is larg
er than 0; AND 2) the second column is less than 0.
>
> View(H[H$X1 > 0 & H$X2 < 0])
> View(H[which(H$X1 > 0 & H$X2 < 0)])
>
> #Setting the working directory to access the stock that was downloaded
>
> getwd()
[1] "C:/Users/gang.ping.m.zhu/Documents"
> setwd("C:/Users/gang.ping.m.zhu/Documents/Stevens/FE513/HW")
>
> # Read the csv file into R, and show the number of rows, number of columns
and column names.
> library('readxl')
>
> data <- read.csv('atvi.csv', header = TRUE)
> summary(data)
     ï..Date         Open             High             Low              Close
Volume
 1-Apr-16:  1   Min.   :28.56   Min.   :29.65   Min.   :28.55   Min.   :28.88
Min.   : 2552688
 1-Aug-16:  1   1st Qu.:36.54   1st Qu.:36.78   1st Qu.:36.09   1st Qu.:36.45
1st Qu.: 5641814
 1-Dec-16:  1   Median :38.74   Median :39.23   Median :38.31   Median :38.92
Median : 7601191
 1-Feb-17:  1   Mean   :38.61   Mean   :39.04   Mean   :38.18   Mean   :38.63
Mean   : 8625980
 1-Jul-16:  1   3rd Qu.:41.42   3rd Qu.:41.77   3rd Qu.:41.09   3rd Qu.:41.38
3rd Qu.: 9698930
 1-Jun-16:  1   Max.   :46.04   Max.   :47.64   Max.   :45.39   Max.   :47.23
Max.   :51703513
 (Other) :249
>
> # show the number of columns separately
> length(data)
[1] 6
> #row(data)


> # creating a column to return the log return of the stock (log(current price/original p
> data$n <- data$Close[255]
> data$logreturn <- log(data$Close/data$n)
> # data$return <- log(data$n/data$Close)
>
> #check number of NA values
> sum(is.na(data$logreturn))
[1] 0
>
> #check number of infinite values
> sum(is.infinite(data$logreturn))
[1] 0
>
> # defining a function for the SMA for the past 10 days
>
> SMA <- function(n) {
+   sum <- 0
```

```
+   for (i in ((n-9):n)) {
+     sum <- sum + data$logreturn[i]
+   }
+   return(sum/10)
+ }
>
> # testing the function
> SMA(255)
[1] 0.04441238



>
```