FE515 2022A Assignment 2

Lorenzo Ausiello

10/17/2023

```r
library(knitr)
knitr::opts_chunk$set(echo = T, out.width="80%",out.height="80%", warning = F)
```

Question 1:

1.1

Find the attached JPM.csv file. Use as.Date() function to change the first column to Date object.

```r
setwd("C:/Users/loaus/OneDrive - stevens.edu/STEVENS/Intro to R/Assignment/Assignment_2")
JPM <- read.csv("FE515_hw2_JPM.csv")

# Changing first column to Date object
JPM$X <- as.Date(JPM$X)
head(JPM)
```

X	JPM.Open	JPM.High	JPM.Low	JPM.Close	JPM.Volume	JPM.Adjusted
1970-01-02	48.00	48.37	47.59	48.07	14244700	32.52235
1970-01-03	48.05	48.55	47.75	48.19	9471500	32.60353
1970-01-04	48.17	48.25	47.63	47.79	10760500	32.33291
1970-01-05	47.57	48.06	47.32	47.95	8239200	32.44115
1970-01-06	47.90	48.11	47.36	47.75	9276700	32.30586
1970-01-07	47.47	48.12	47.44	48.10	15597000	32.54265

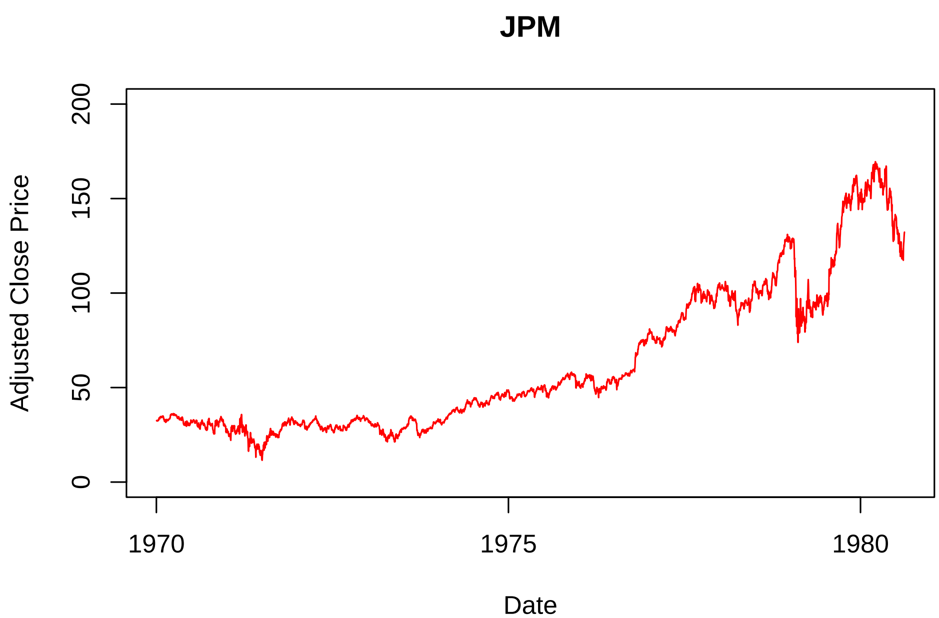
```r
tail(JPM)
```

	X	JPM.Open	JPM.High	JPM.Low	JPM.Close	JPM.Volume	JPM.Adjusted
3875	1980-08-11	120.46	126.57	120.45	124.60	24376700	124.60
3876	1980-08-12	124.36	127.49	123.11	126.36	19324900	126.36
3877	1980-08-13	126.28	127.86	125.17	127.24	12780400	127.24
3878	1980-08-14	128.37	130.19	128.21	129.44	15990900	129.44
3879	1980-08-15	130.16	131.27	129.14	131.27	11173400	131.27
3880	1980-08-16	130.18	133.15	129.71	132.23	17012800	132.23

1.2

Plot the adjusted close price against the date object (i.e. date object as x-axis and close price as y-axis) in red line (require no points). Set the title as JPM, the label for x-axis as Date and the label for y-axis as Adjusted Close Price.

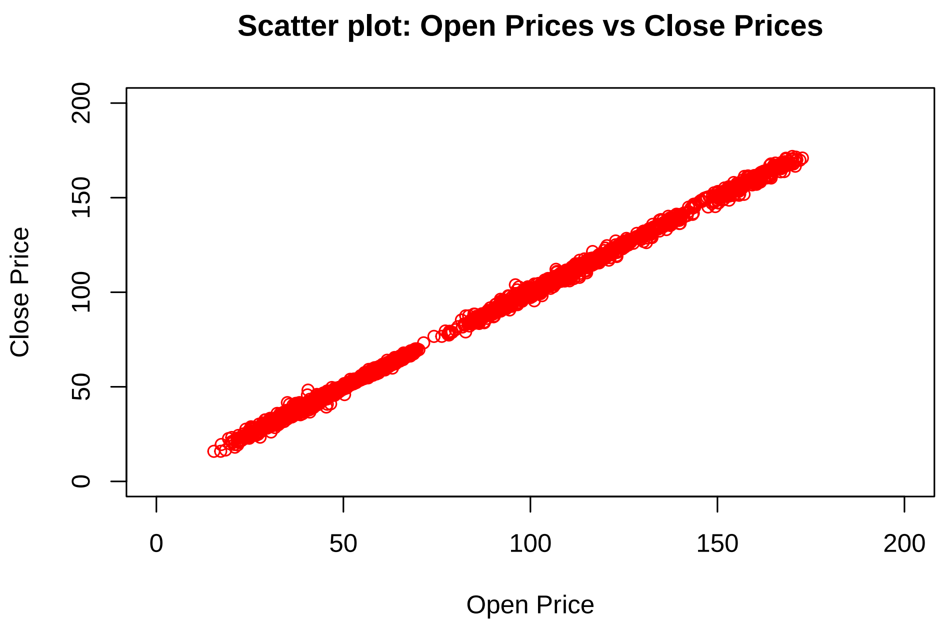
```
plot(JPM$X, JPM$JPM.Adjusted, type = "l", main = "JPM",
xlim = c(as.Date("1970-01-01"), as.Date("1980-08-16")),ylim = c(0,200),
xlab = "Date", ylab = "Adjusted Close Price", col = "red")
```



1.3

Create a scatter plot of close price against open price (i.e. open prices as x-axis, and close prices as y-axis). Set the x label as ”Open Price” and y label as ”Close Price”.

```
plot(JPM$JPM.Open, JPM$JPM.Close, type = "p", main = "Scatter plot: Open Prices vs Close Prices",
xlim = c(0,200),ylim = c(0,200),
xlab = "Open Price", ylab = "Close Price", col = "red")
```



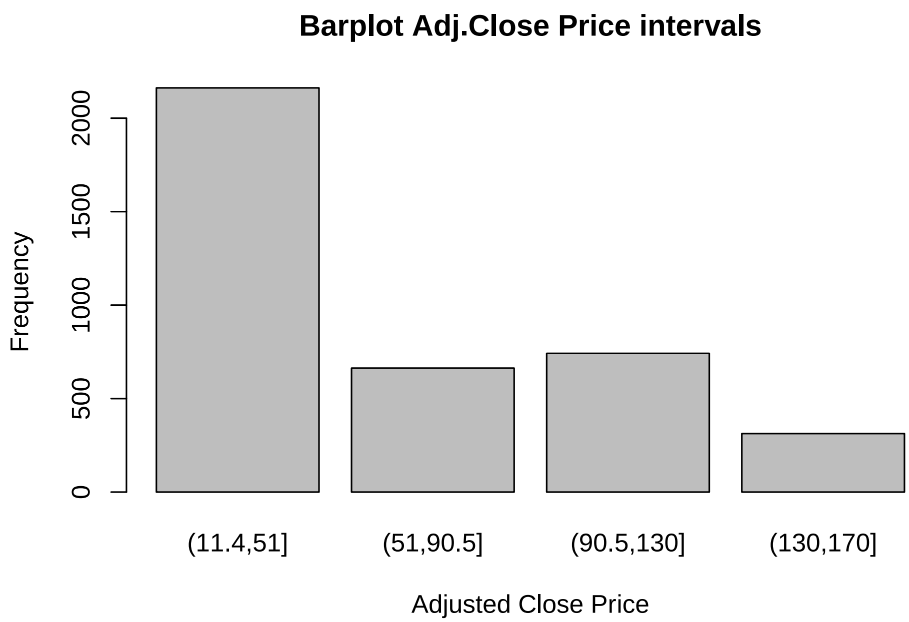
1.4

Use cut() function to divide adjusted close price into 4 intervals. Generate a barplot for the frequencies of these intervals. .

```
cut <- table(cut(JPM$JPM.Adjusted,4))
cut
```

```
##
##  (11.4,51]  (51,90.5] (90.5,130]  (130,170]
##       2162        663        742        313
```

```
barplot(cut, xlab = "Adjusted Close Price", ylab = "Frequency", main = "Barplot Adj.Close Price interval
```



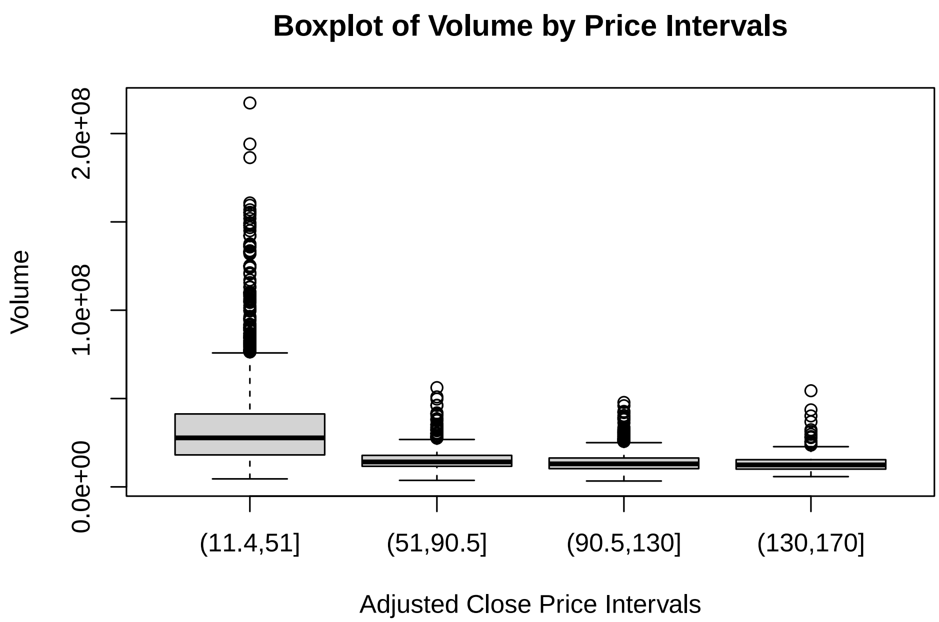
1.5

Generate a boxplot of volume against the 4 intervals of adjusted close prices.

```
Intervals <- cut(JPM$JPM.Adjusted,breaks = 4)
tail(Intervals)
```

```
## [1] (90.5,130] (90.5,130] (90.5,130] (90.5,130] (130,170]  (130,170]
## Levels: (11.4,51] (51,90.5] (90.5,130] (130,170]
```

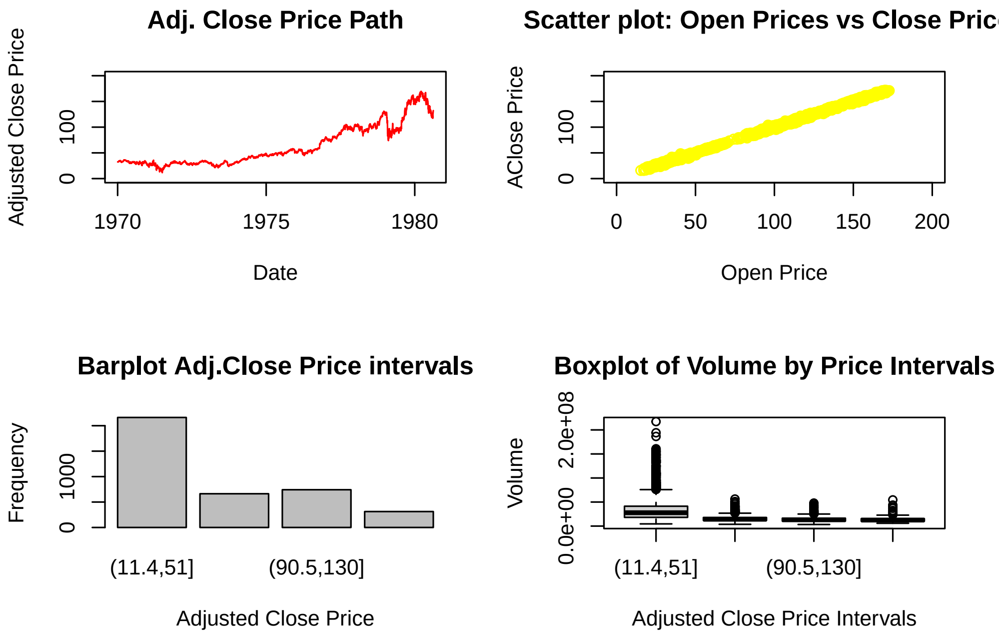
```
boxplot(JPM$JPM.Volume ~ Intervals, xlab = "Adjusted Close Price Intervals", ylab = "Volume", main = "B
```



1.6

Use par() function to create a picture of 4 subplots. Gather the 4 figures from 1.2 - 1.5 into ONE single picture. Please arrange the 4 subplots into a 2 by 2 frame, i.e. a frame consists of 2 columns and 2 rows.

```
{par(mfrow = c(2, 2))
  plot(JPM$X, JPM$JPM.Adjusted, type = "l", main = "Adj. Close Price Path",
  xlim = c(as.Date("1970-01-01"), as.Date("1980-08-16")),ylim = c(0,200),
  xlab = "Date", ylab = "Adjusted Close Price", col = "red")
  plot(JPM$JPM.Open, JPM$JPM.Close, type = "p", main = "Scatter plot: Open Prices vs Close Prices",
  xlim = c(0,200),ylim = c(0,200),
  xlab = "Open Price", ylab = "AClose Price", col = "yellow")
  barplot(cut, xlab = "Adjusted Close Price", ylab = "Frequency", main = "Barplot Adj.Close Price inter
  boxplot(JPM$JPM.Volume ~ Intervals, xlab = "Adjusted Close Price Intervals", ylab = "Volume",
          main = "Boxplot of Volume by Price Intervals")}
```



Question 2

Estimate the volume of the unit sphere (which is just 4 /3) by simulation. Suppose we draw N points from three dimensional uniform distribution.

```
N <- 10000
x <- runif(N)
y <- runif(N)
z <- runif(N)

num_inner_points <- 0
for (i in 1:N) {
  if(x[i]^2 + y[i]^2 + z[i]^2 <=1 & x[i]>= 0 & y[i]>= 0 & z[i]>= 0){
    num_inner_points <- num_inner_points + 1
  }
}
#or
num_inner_points <- sum(x^2 + y^2 + z^2 <= 1)

volume_sphere <- num_inner_points/N
volum_unit_spere <- 8 * volume_sphere
volum_unit_spere
```

```
## [1] 4.1528
```

Question 3

3.1

Implement a Linear Congruential Generator (LCG) which generates pseudo-random number from uniform distribution using m = 244944, a = 1597, b = 51749.

```
seed <- 1
LCG <- function(n){
m <- 244944
a <- 1597
b <- 51749
x <- rep(NA, n)
x[1] <- (a * seed + b) %% m
for(i in 1:(n-1)){
x[i + 1] <- (a * x[i] + b) %% m
}
seed <<- x[n]
return(x/m)
}

seed<-1
LCG(5)
```

```
## [1] 0.21778856 0.01959223 0.50006532 0.81558642 0.70278104
```

```
seed<-1
LCG(5)
```

```
## [1] 0.21778856 0.01959223 0.50006532 0.81558642 0.70278104
```

```
seed<-100
LCG(5)
```

```
## [1] 0.863254458 0.828638383 0.546765791 0.396237507 0.002567934
```

3.2

Use the LCG in the previous problem, generate 10000 random numbers from chi-square distribution with 10 degrees of freedom (i.e. df = 10), and assign to a variable.

```
seed <- as.numeric(Sys.time())
rnds <- qchisq(LCG(10000),df = 10)
head(rnds)
```

```
## [1] 11.851851 18.073544  8.248604 12.022642  8.208043  8.284887
```

3.3

Visualize the resulting sample from 3.2 using a histogram with 40 bins.

```r
hist(rnds, nclass = 40)
```