FE515 2022A Assignment 2

Yufu Liao

03/11/2023

Question 1: (50 points)

1.1

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

```
jpm <- read.csv("JPM.csv")
jpm$X <- as.Date(jpm$X, origin = "2023/01/01")
jpm[1:10,]
```

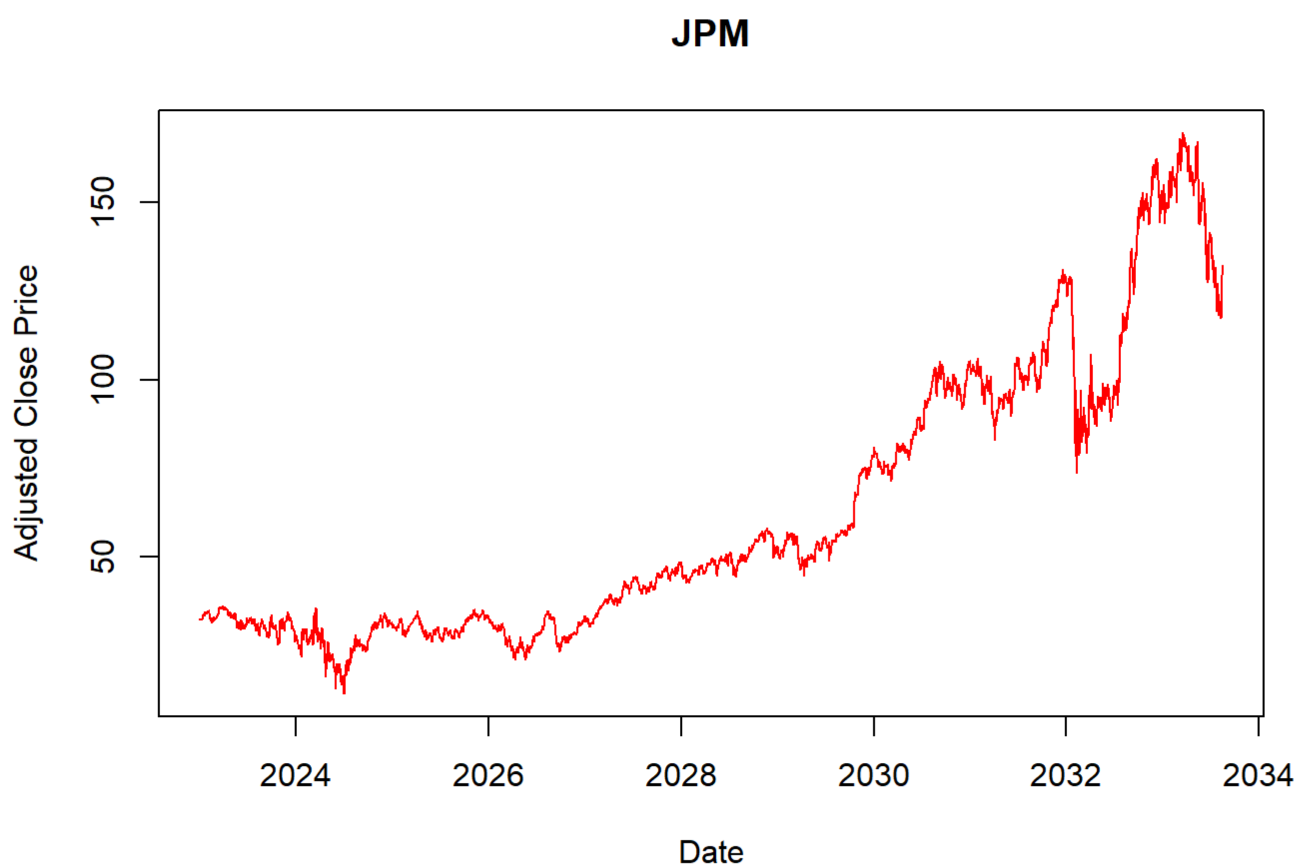
	X <date>	JPM.Open <dbl>	JPM.High <dbl>	JPM.Low <dbl>	JPM.Close <dbl>	JPM.Volume <int>	JPM.Adjusted <dbl>
1	2023-01-02	48.00	48.37	47.59	48.07	14244700	32.52235
2	2023-01-03	48.05	48.55	47.75	48.19	9471500	32.60353
3	2023-01-04	48.17	48.25	47.63	47.79	10760500	32.33291
4	2023-01-05	47.57	48.06	47.32	47.95	8239200	32.44115
5	2023-01-06	47.90	48.11	47.36	47.75	9276700	32.30586
6	2023-01-07	47.47	48.12	47.44	48.10	15597000	32.54265
7	2023-01-08	48.00	48.42	47.94	48.31	8049200	32.68473
8	2023-01-09	48.10	48.26	47.90	47.99	10646700	32.46823
9	2023-01-10	48.16	48.46	48.10	48.39	8696500	32.73885
10	2023-01-11	48.65	48.89	48.12	48.43	16291400	32.76591

1-10 of 10 rows

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.

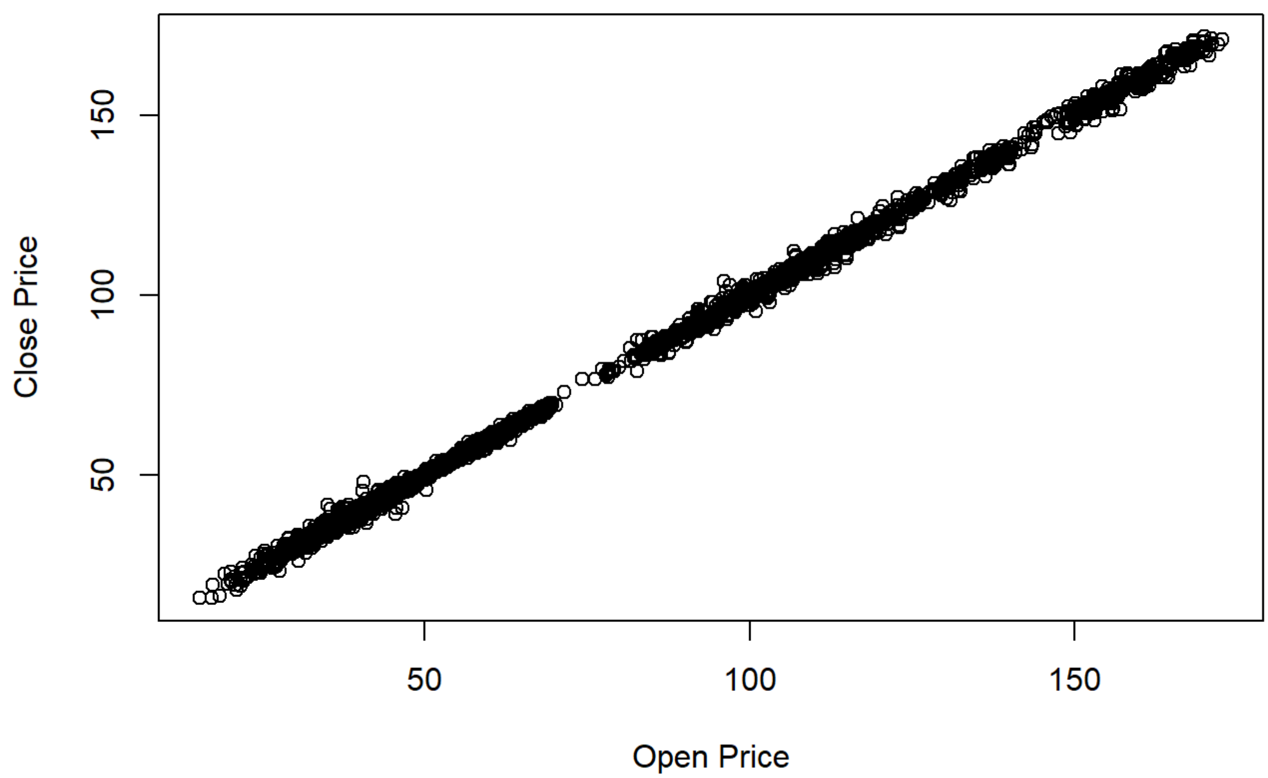
```
plot(jpm$X, jpm$JPM.Adjusted,
     main = "JPM",
     xlab = "Date",
     type = "l",
     ylab = "Adjusted Close Price",
     col = "red"
)
```



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".

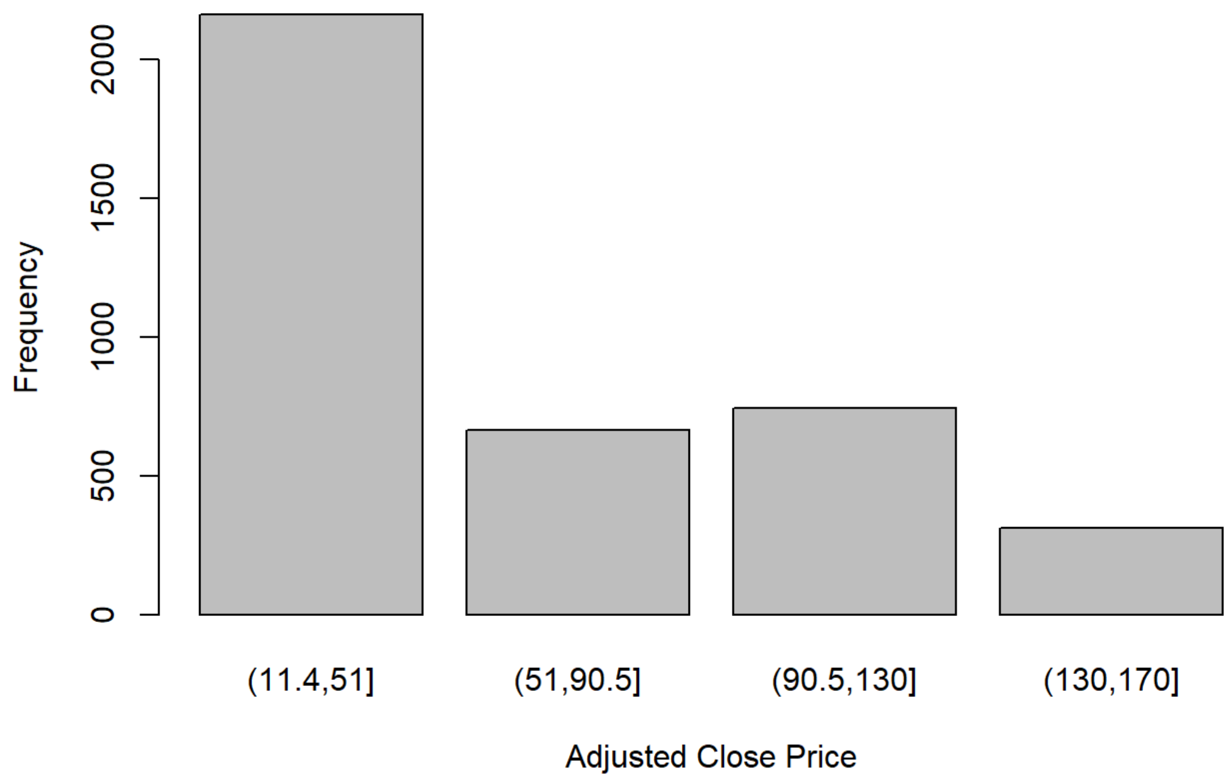
```
plot(jpm$JPM.Open, jpm$JPM.Close,
     xlab = "Open Price",
     ylab = "Close Price"
)
```



1.4

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

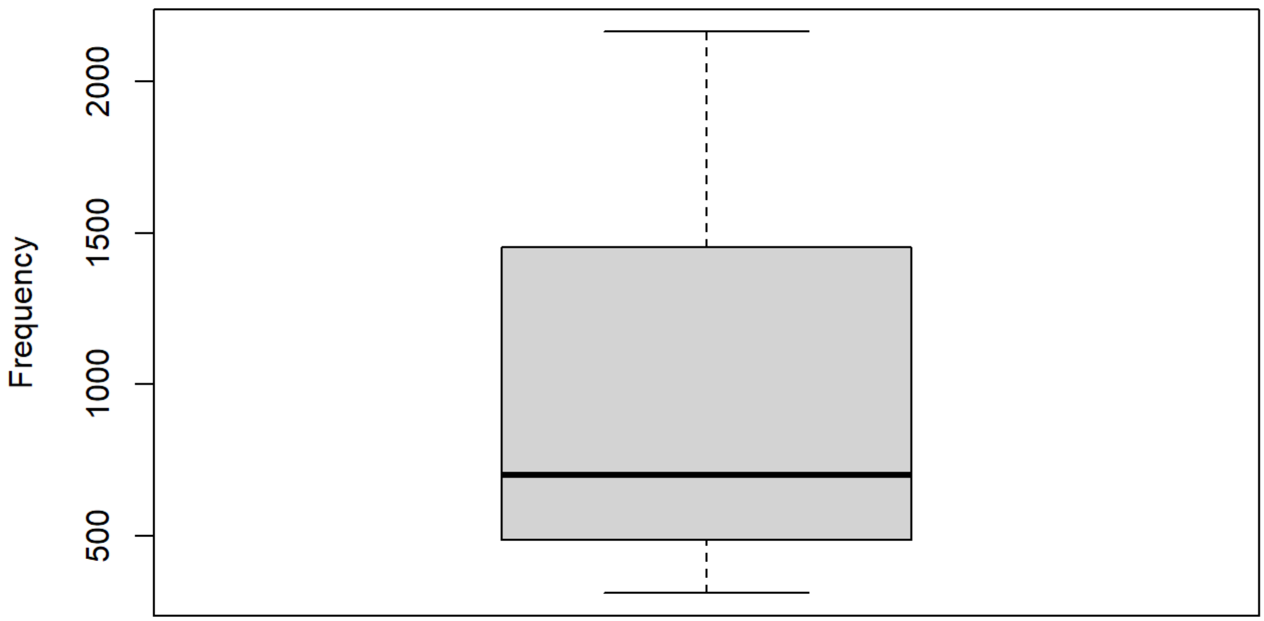
```
cut_close <- cut(jpm$JPM.Adjusted, 4)
barplot(table(cut_close), xlab = "Adjusted Close Price", ylab = "Frequency")
```



1.5

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

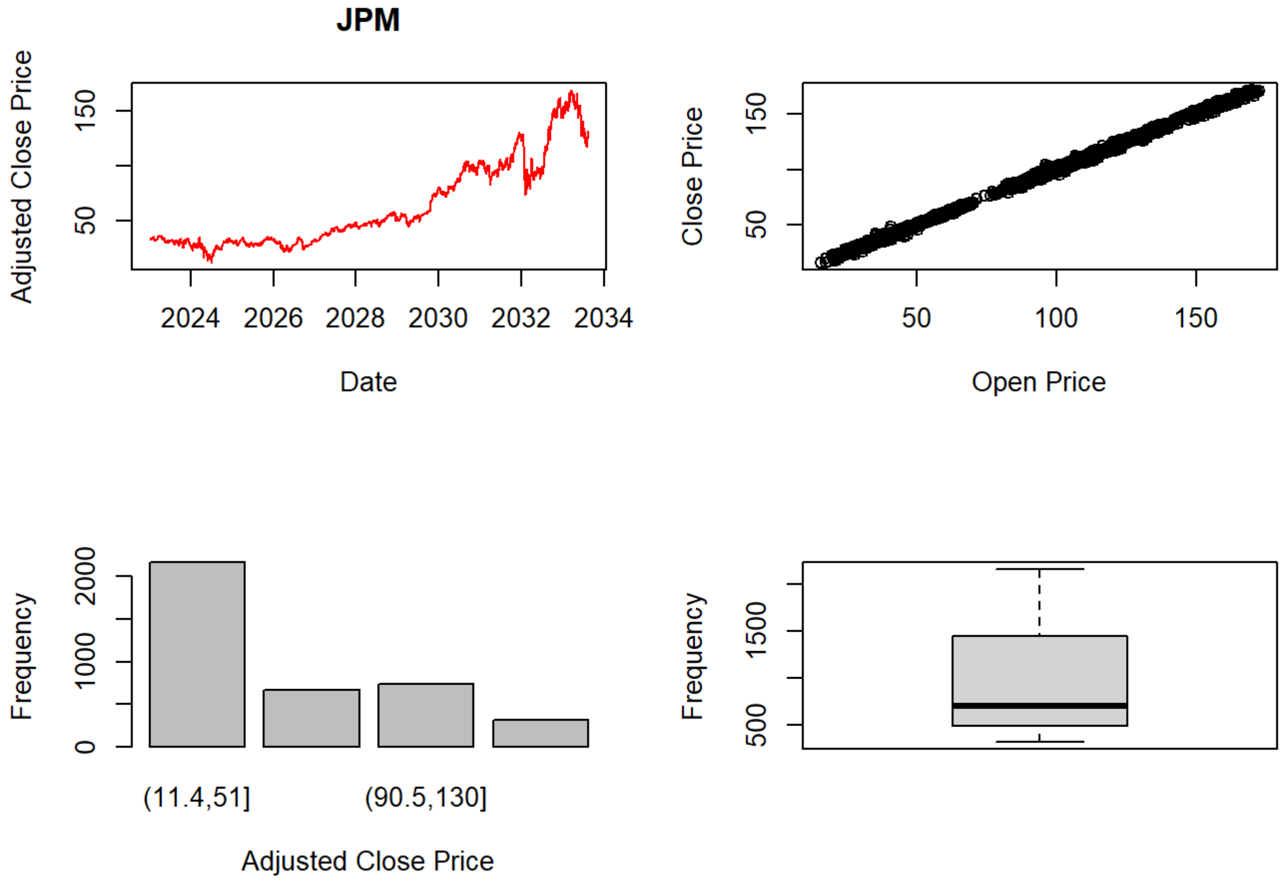
```
boxplot(table(cut_close), ylab = "Frequency")
```



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. (Hint. `par(mfrow = c(1,3))` will create a picture of three subplots. In the picture, the subplots are arranged into a 1 by 3 frame.)

```
par(mfrow = c(2, 2))
plot(jpm$X, jpm$JPM.Adjusted,
     main = "JPM",
     xlab = "Date",
     type = "l",
     ylab = "Adjusted Close Price",
     col = "red"
)
plot(jpm$JPM.Open, jpm$JPM.Close,
     xlab = "Open Price",
     ylab = "Close Price"
)
barplot(table(cut_close), xlab = "Adjusted Close Price", ylab = "Frequency")
boxplot(table(cut_close), ylab = "Frequency")
```



Question 2

Estimate the volume of the unit sphere (which is just 4π/3) by simulation.

```
seed <- 1
rnd <- function(n){
  m <- 2 ^ 31 - 1
  a <- 7 ^ 5
  b <- 0

  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)
}

num.total <- 100000
x <- rnd(num.total)
y <- rnd(num.total)
z <- rnd(num.total)

num.inner <- sum(x ^ 2 + y ^ 2 + z ^ 2 <= 1)
volume.eighth <- num.inner / num.total
(volume.sphere <- 8 * volume.eighth)
```

```
## [1] 4.16512
```

Question 3

3.1

Implement a Linear Congruential Generator (LCG)

```
LCG <- function(n) {
  m <- 244944
  a <- 1597
  b <- 51749

  x <- rep(NaN, n)
  x[1] <- (a * 1 + b) %% m

  for(i in 1:(n - 1)) {
    x[i + 1] <- (a * x[i] + b) %% m
  }

  return(x / m)
}
```

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. (Hint.: `X = qnorm(LCG(10000))` will generate a sample of 10000 numbers X which follows normal distribution. For chi-square case, please consider another function `qchisq()`.)

```
X <- qchisq(LCG(10000), df = 10)
head(X)
```

```
## [1]  6.382024  3.042432  9.342521 13.756028 11.820942  9.921533
```

3.3

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

```
hist(X, nclass = 40)
```