

Assignment 3

Student Name : 정석규

Student ID : 201724570-

Q1.

1) Source code :

```
x <- seq(-1, 1, by = 0.01)

Q_x <- -1 + ((exp(x)+exp(-x))/2^abs(x)) - (sqrt(abs(x)*(1-abs(x)))/(x+sqrt(2)))

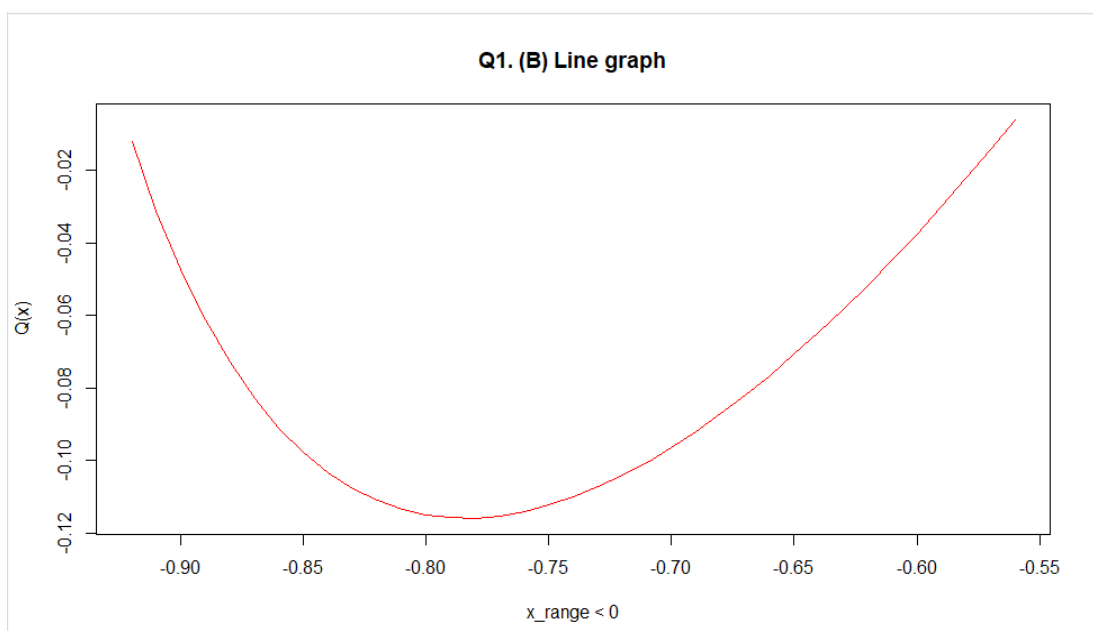
plot(x[which(Q_x<0)], Q_x[Q_x<0], type = 'n', main = 'Q1. (B) Line graph',
     xlab = 'x_range < 0', ylab = 'Q(x)')

lines(x[which(Q_x<0)], Q_x[Q_x<0], type = 'l', lty = 1, col = "red")
```

2) R Screenshot :

```
1 x <- seq(-1, 1, by = 0.01)
2 Q_x <- -1 + ((exp(x)+exp(-x))/2^abs(x)) - (sqrt(abs(x)*(1-abs(x)))/(x+sqrt(2)))
3
4 plot(x[which(Q_x<0)], Q_x[Q_x<0], type = 'n', main = 'Q1. (B) Line graph',
5      xlab = 'x_range < 0', ylab = 'Q(x)')
6 lines(x[which(Q_x<0)], Q_x[Q_x<0], type = 'l', lty = 1, col = "red")
7 |
```

3) Answer :



Q2.

1) Source code :

```
Q2_data <- data.frame(ToothGrowth)

boxplot(len~supp+dose, data = Q2_data,

        main = "Side by Side boxplot of Q2",

        xlab = "Supplement type + Amount of dose",

        ylab = "tooth length",

        col = 'pink')
```

2) R Screenshot :

```
8 Q2_data <- data.frame(ToothGrowth)
9 boxplot(len~supp+dose, data = Q2_data,
10         main = "Side by side boxplot of Q2",
11         xlab = "Supplement type + Amount of dose",
12         ylab = "tooth length",
13         col = 'pink')
```

3) Answer :



Q3.

4) Source code :

```
Q3_data <- as.data.frame(EuStockMarkets)

x <- c(time(EuStockMarkets))

plot(x, Q3_data[, 'DAX'], type = 'n',

      main = "Multiple line graph of Q3", xlab = 'Time', ylab = "Price")

lines(x, Q3_data[, 'DAX'], type = 'l', col = 'red')

lines(x, Q3_data[, 'SMI'], type = 'l', col = 'blue')

lines(x, Q3_data[, 'CAC'], type = 'l', col = 'green')

lines(x, Q3_data[, 'FTSE'], type = 'l', col = 'pink')

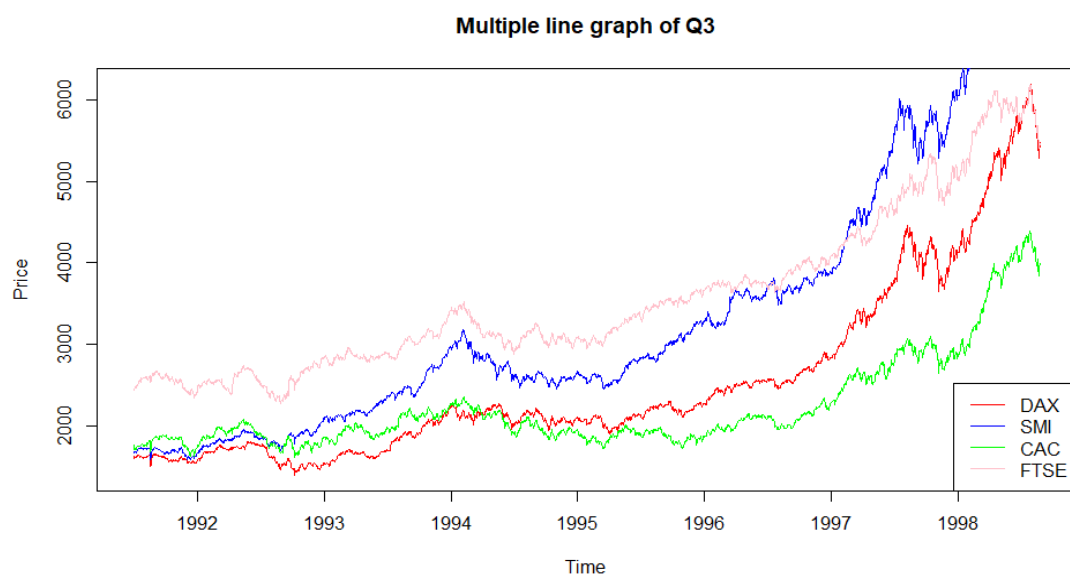
legend("bottomright", legend = c('DAX', 'SMI', 'CAC', 'FTSE'), lty = 1,

      col = c('red', 'blue', 'green', 'pink'), cex = 1)
```

5) R Screenshot

```
16 Q3_data <- as.data.frame(EuStockMarkets)
17 x <- c(time(EuStockMarkets))
18 plot(x, Q3_data[, 'DAX'], type = 'n',
19      main = "Multiple line graph of Q3", xlab = 'Time', ylab = "Price")
20 lines(x, Q3_data[, 'DAX'], type = 'l', col = 'red')
21 lines(x, Q3_data[, 'SMI'], type = 'l', col = 'blue')
22 lines(x, Q3_data[, 'CAC'], type = 'l', col = 'green')
23 lines(x, Q3_data[, 'FTSE'], type = 'l', col = 'pink')
24 legend("bottomright", legend = c('DAX', 'SMI', 'CAC', 'FTSE'), lty = 1,
25      col = c('red', 'blue', 'green', 'pink'), cex = 1)
```

6) Answer :



Q4.

1) Source code :

```
my_data <- as.data.frame(penguins)

Q4_data <- my_data[c('species', 'bill_length_mm', 'bill_depth_mm')]

x <- Q4_data$bill_length_mm; y <- Q4_data$bill_depth_mm

plot(x, y, type = 'n',

     main = 'Sactter plot of Q4', xlab = 'bill_length_mm', ylab = 'bill_depth_mm')

points(x[Q4_data['species'] == 'Adelie'], y[Q4_data['species'] == 'Adelie'], pch = 17, col = 'green')

points(x[Q4_data['species'] == 'Gentoo'], y[Q4_data['species'] == 'Gentoo'], pch = 18, col = 'blue')

points(x[Q4_data['species'] == 'Chinstrap'], y[Q4_data['species'] == 'Chinstrap'], pch = 19, col = 'red')

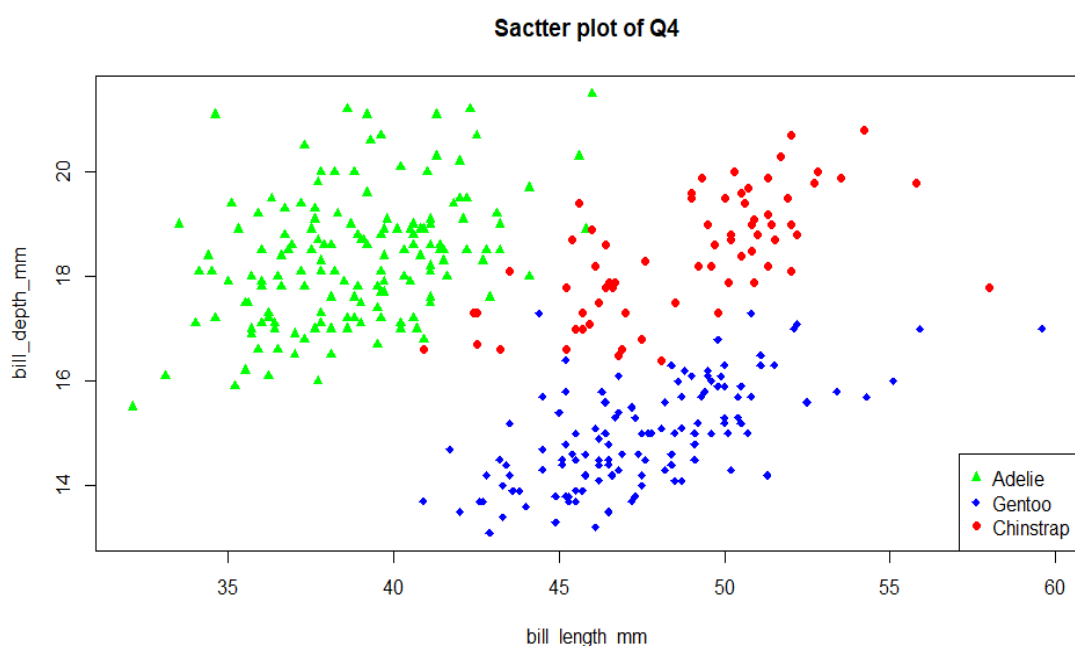
legend("bottomright", legend = c('Adelie', 'Gentoo', 'Chinstrap'),

     pch = c(17,18,19), col = c('green', 'blue', 'red'))
```

2) R Screenshot :

```
28 library('palmerpenguins')
29 my_data <- as.data.frame(penguins)
30 Q4_data <- my_data[c('species', 'bill_length_mm', 'bill_depth_mm')]
31 x <- Q4_data$bill_length_mm; y <- Q4_data$bill_depth_mm
32 plot(x, y, type = 'n',
33      main = 'Sactter plot of Q4', xlab = 'bill_length_mm', ylab = 'bill_depth_mm')
34 points(x[Q4_data['species'] == 'Adelie'], y[Q4_data['species'] == 'Adelie'], pch = 17, col = 'green')
35 points(x[Q4_data['species'] == 'Gentoo'], y[Q4_data['species'] == 'Gentoo'], pch = 18, col = 'blue')
36 points(x[Q4_data['species'] == 'Chinstrap'], y[Q4_data['species'] == 'Chinstrap'], pch = 19, col = 'red')
37 legend("bottomright", legend = c('Adelie', 'Gentoo', 'Chinstrap'),
38      pch = c(17,18,19), col = c('green', 'blue', 'red'))
```

3) Answer :



Q5.

1) Source code :

```
set.seed(1234)

U1 <- runif(1000); U2 <- runif(1000)

Z1 <- sqrt(-2*log(U1, base = exp(1)))*cos(2*pi*U1); Z2 <- sqrt(-2*log(U2, base = exp(1)))*cos(2*pi*U2)

sigma <- 1; mu <- 0

X1 <- sigma*Z1 + mu; X2 <- sigma*Z2 + mu

summary(X1); summary(X2)

xpts <- seq(min(X1), max(X1), length.out=50)

ypts <- dnorm(xpts, mean=mean(X1), sd=sd(X1))

par(mfrow = c(2,1))

plot(xpts, ypts, type = 'n', main = "Line plot of Q5", xlab = 'X1', ylab = 'dnorm(X1)')

lines(xpts, ypts, lwd=2, col="blue")

xpts1 <- seq(min(X2), max(X2), length.out=50)

ypts1 <- dnorm(xpts, mean=mean(X2), sd=sd(X2))

plot(xpts1, ypts1, type = 'n', main = "Line plot of Q5", xlab = 'X2', ylab = 'dnorm(X2)')

lines(xpts1, ypts1, lwd=2, col="blue")
```

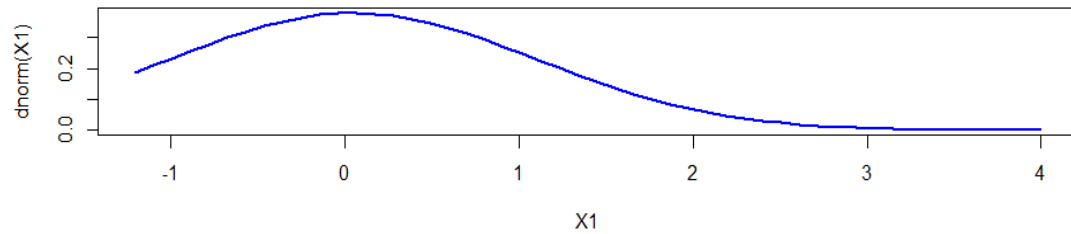
2) R Screenshot :

```
40 set.seed(1234)
41 U1 <- runif(1000); U2 <- runif(1000)
42 Z1 <- sqrt(-2*log(U1, base = exp(1)))*cos(2*pi*U1); Z2 <- sqrt(-2*log(U2, base = exp(1)))*cos(2*pi*U2)
43 sigma <- 1; mu <- 0
44 X1 <- sigma*Z1 + mu; X2 <- sigma*Z2 + mu
45 summary(X1); summary(X2)
46 xpts <- seq(min(X1), max(X1), length.out=50)
47 ypts <- dnorm(xpts, mean=mean(X1), sd=sd(X1))
48 par(mfrow = c(2,1))
49 plot(xpts, ypts, type = 'n', main = "Line plot of Q5", xlab = 'X1', ylab = 'dnorm(X1)')
50 lines(xpts, ypts, lwd=2, col="blue")
51
52 xpts1 <- seq(min(X2), max(X2), length.out=50)
53 ypts1 <- dnorm(xpts, mean=mean(X2), sd=sd(X2))
54 plot(xpts1, ypts1, type = 'n', main = "Line plot of Q5", xlab = 'X2', ylab = 'dnorm(X2)')
55 lines(xpts1, ypts1, lwd=2, col="blue")
--
```

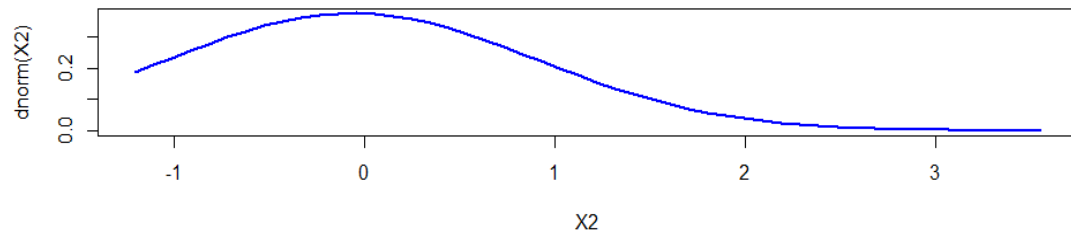
3) **Answer :** They are follow the standard normal distribution

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.207709	-0.809031	-0.003992	0.047391	0.358332	3.995300
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.20770	-0.86485	-0.06343	0.03964	0.36029	3.55093

Line plot of Q5



Line plot of Q5



Q6.

1) Source code :

```
f <- function(x) x^4-9*x^3-334*x^2+4416*x-10080

fprime <- function(x) 4*x^3-27*x^2-668*x + 4416

par(mfrow = c(2,2))

Newton <- function(x, tol = 10^(-10)) {

  curve(f, from = -35, to = 35, col = 'blue', main = 'Newton Method', xlab = 'X', ylab = 'f(x)')

  text(x=x, y=f(x), labels="1st label : ")

  while(abs(f(x)) > tol) {

    x <- x - (f(x)/fprime(x))

    points(x, f(x), pch = 18, col = 'red')

  }

}

Descent <- function(x, tol = 10^(-10)) {

  curve(fprime, from = -35, to = 35, col = 'blue',

    main = 'Steepest Descent Algorithm', xlab = 'X', ylab = 'fprime(x)')

  text(x=x, y=fprime(x), labels="1st label")

  alpha = 10^(-5)

  while(abs(alpha*fprime(x)) > tol) {

    x <- x -(alpha*fprime(x))

    points(x, fprime(x), pch = 18, col = 'red')

  }

}

Newton(-30)

Newton(30)

Descent(-30)

Descent(30)
```

2) R Screenshot :

```

58 f <- function(x) x^4-9*x^3-334*x^2+4416*x-10080
59 fprime <- function(x) 4*x^3-27*x^2-668*x + 4416
60 par(mfrow = c(2,2))
61 Newton <- function(x, tol = 10^(-10)) {
62   curve(f, from = -35, to = 35, col = 'blue', main = 'Newton Method', xlab = 'x', ylab = 'f(x)')
63   text(x=x, y=f(x), labels="1st label : ")
64   while(abs(f(x)) > tol) {
65     x <- x - (f(x)/fprime(x))
66     points(x, f(x), pch = 18, col = 'red')
67   }
68 }
69 }
70 Descent <- function(x, tol = 10^(-10)) {
71   curve(fprime, from = -35, to = 35, col = 'blue',
72         main = 'Steepest Descent Algorithm', xlab = 'x', ylab = 'fprime(x)')
73   text(x=x, y=fprime(x), labels="1st label")
74   alpha = 10^(-5)
75   while(abs(alpha*fprime(x)) > tol) {
76     x <- x - (alpha*fprime(x))
77     points(x, fprime(x), pch = 18, col = 'red')
78   }
79 }
80 Newton(-30)
81 Newton(30)
82 Descent(-30)
83 Descent(30)

```

3) Answer :

