Statistical Computing HW1

106033233 資工21 周聖諺

3/21/2021

Problem 1:

(a) Generate standard normal distribution by using Box-Muller approach with 10000 samples. Display the result by the histogram and the boxplot.

Pseudo Code:

Step 1. Generate U_1, U_2 from uniform $U(0, 1)$ independently

Step 2. Let variable

$$X = \sqrt{-2ln\,U_1}\cos(2\pi U_2)$$

$$Y = \sqrt{-2ln\,U_1}\sin(2\pi U_2)$$

Step 3. Return X or Y, since $X, Y \overset{i.i.d}{\sim} N(0, 1)$

```
library(compositions)
```

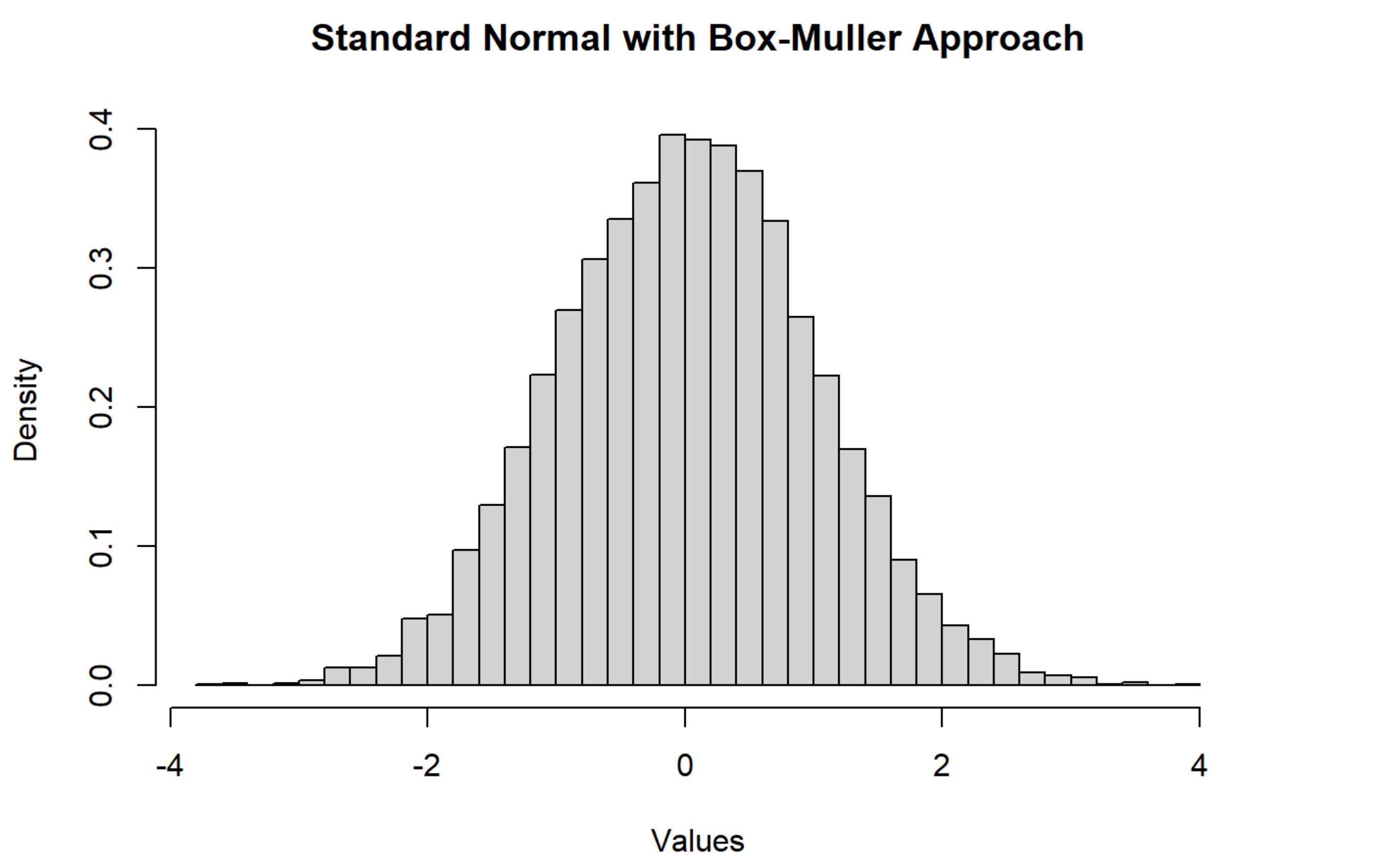
```
## Welcome to compositions, a package for compositional data analysis.
## Find an intro with "? compositions"
```

```
##
## Attaching package: 'compositions'
```

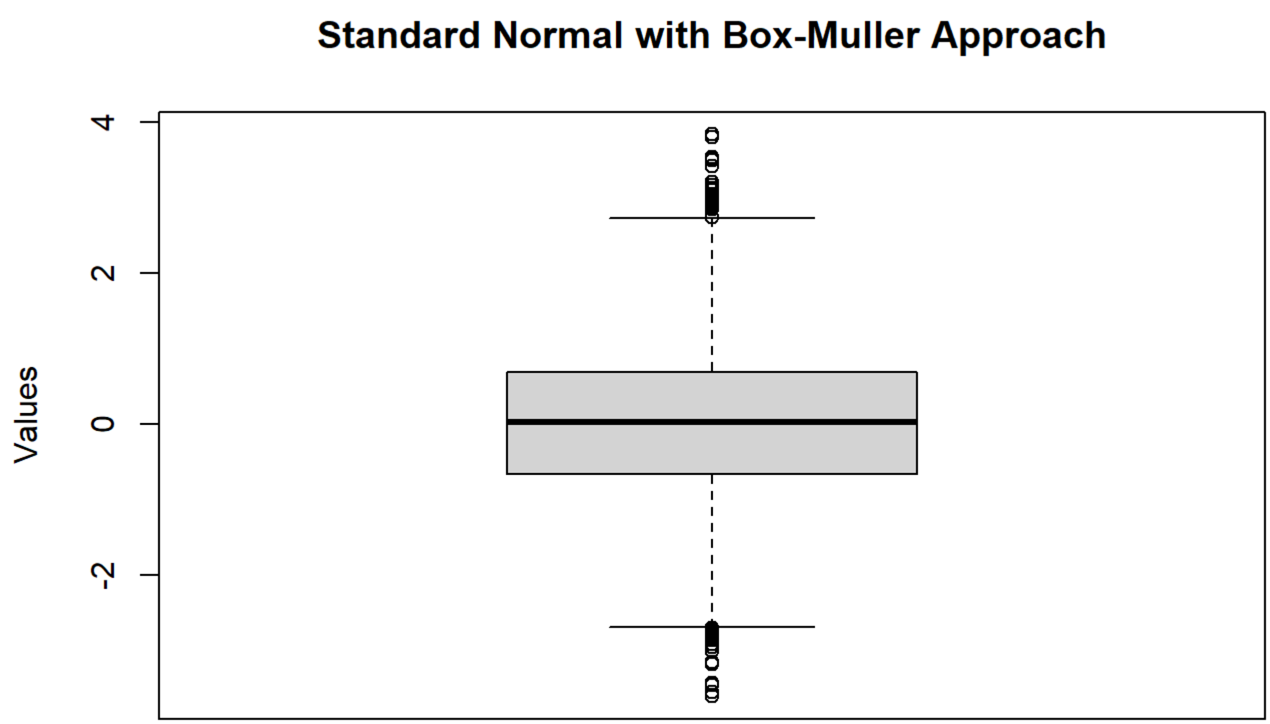
```
## The following objects are masked from 'package:stats':
##
##     cor, cov, dist, var
```

```
## The following objects are masked from 'package:base':
##
##     %*%, norm, scale, scale.default
```

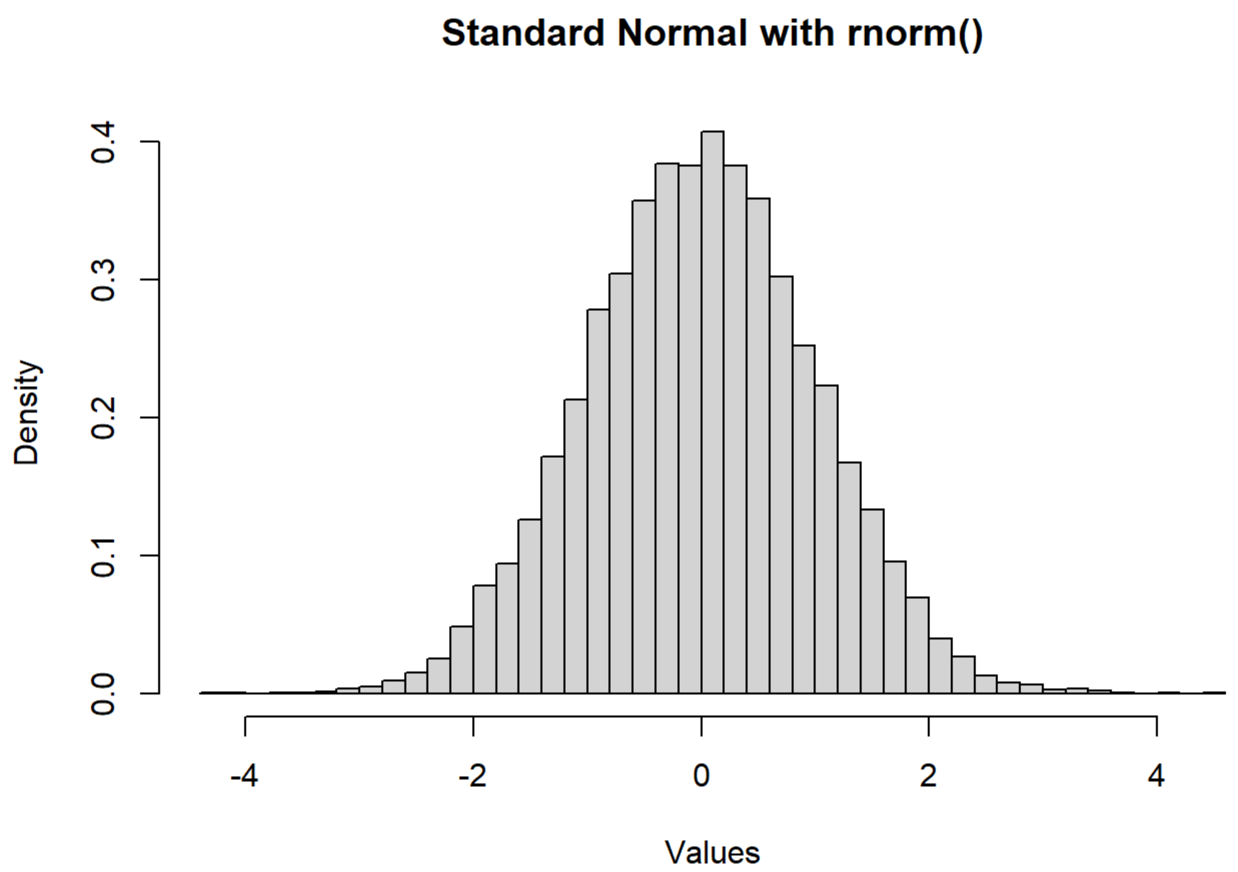
```
normal_box_muller <- function(n){
  res <- vector("numeric", length=n)

  for (i in 0:n) {
    u1 <- runif(1, 0, 1)
    u2 <- runif(1, 0, 1)

    radius <- sqrt(-2 * log(u1))
    angle <- 2 * pi * u2

    x <- radius * cos(angle)
    y <- radius * sin(angle)

    #print(x)
    #print(y)

    res[i] <- x
  }

  return(res)
}
n <- 10000
res <- normal_box_muller(n)

hist(res, main="Standard Normal with Box-Muller Approach", xlab="Values", breaks=50, freq = FALS
E)
```



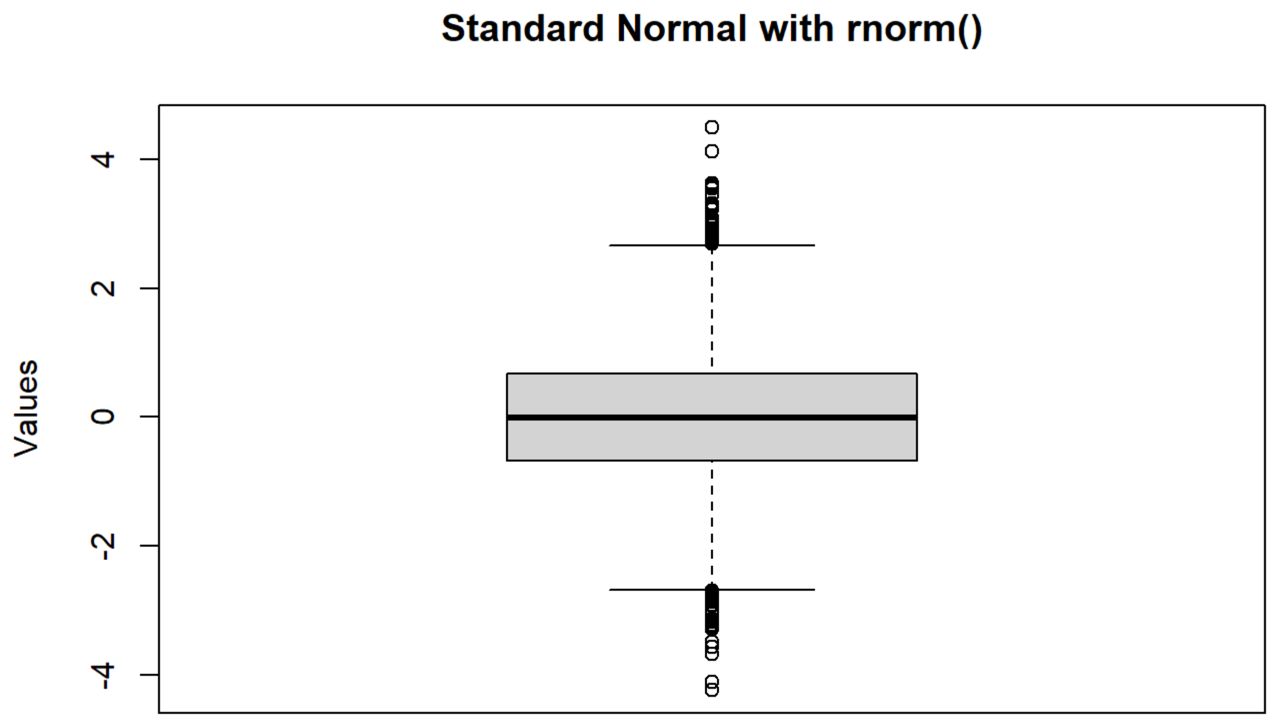
```
boxplot(res, main="Standard Normal with Box-Muller Approach", ylab="Values", freq = FALSE)
```




```
r_dist <- rnorm(n, 0, 1)
hist(r_dist, main="Standard Normal with rnorm()", xlab="Values", breaks=50, freq = FALSE)
```




```
boxplot(r_dist, main="Standard Normal with rnorm()", ylab="Values", freq = FALSE)
```



(b) Generate standard normal distribution by using Acceptance and Rejection approach with 10000 samples. Display the result by the histogram and the boxplot.

Pseudo Code:

Step 1. Generate $Y \sim Exp(1), U_1 \sim U(0,1)$

Step 2. If $U_1 \leq \frac{f_{|X|}(Y)}{cg(X)} = e^{-(Y-1)^2}$, set $X = Y$. Otherwise, go back to Step 1.

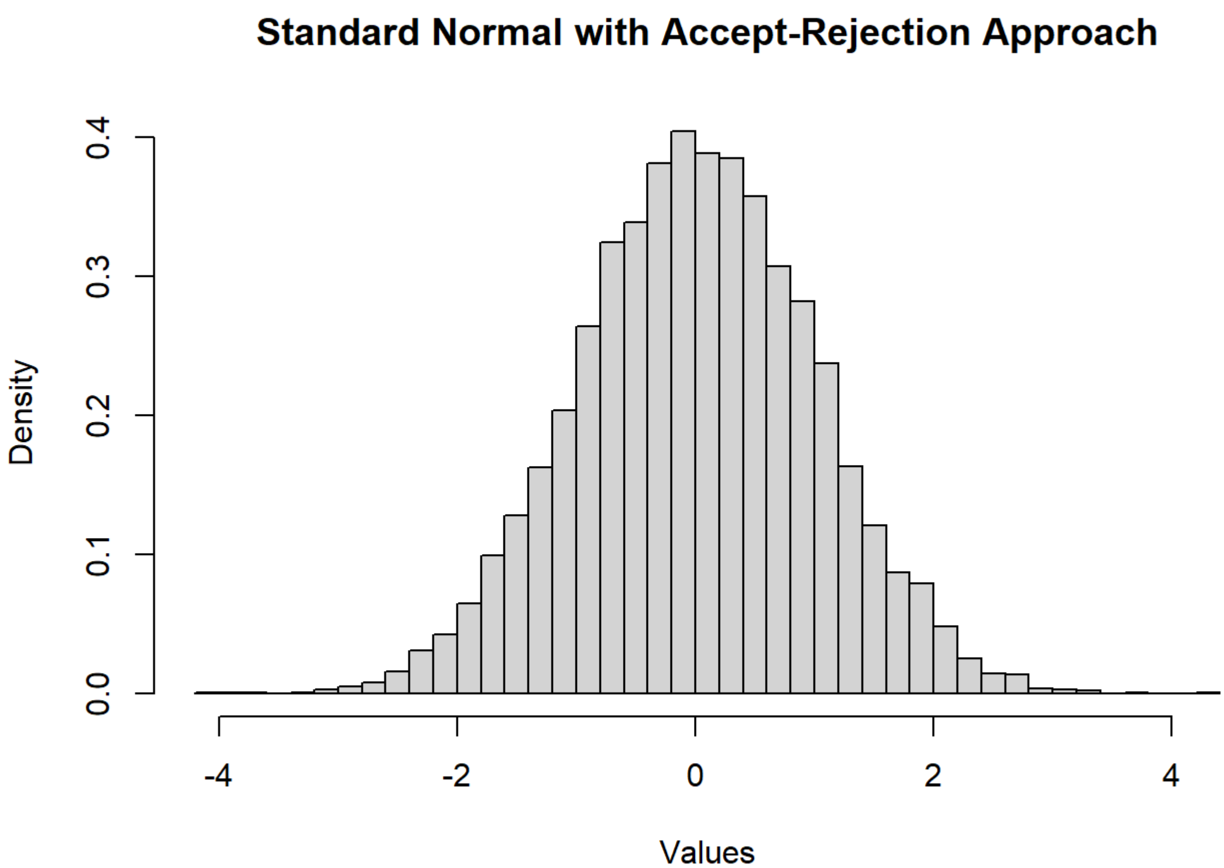
Step 3. Generate $U_2 \sim U(0,1)$. If $U_2 \leq 0.5$, set $X = |X|$. Otherwise, $X = -|X|$.

Step 4. Return X

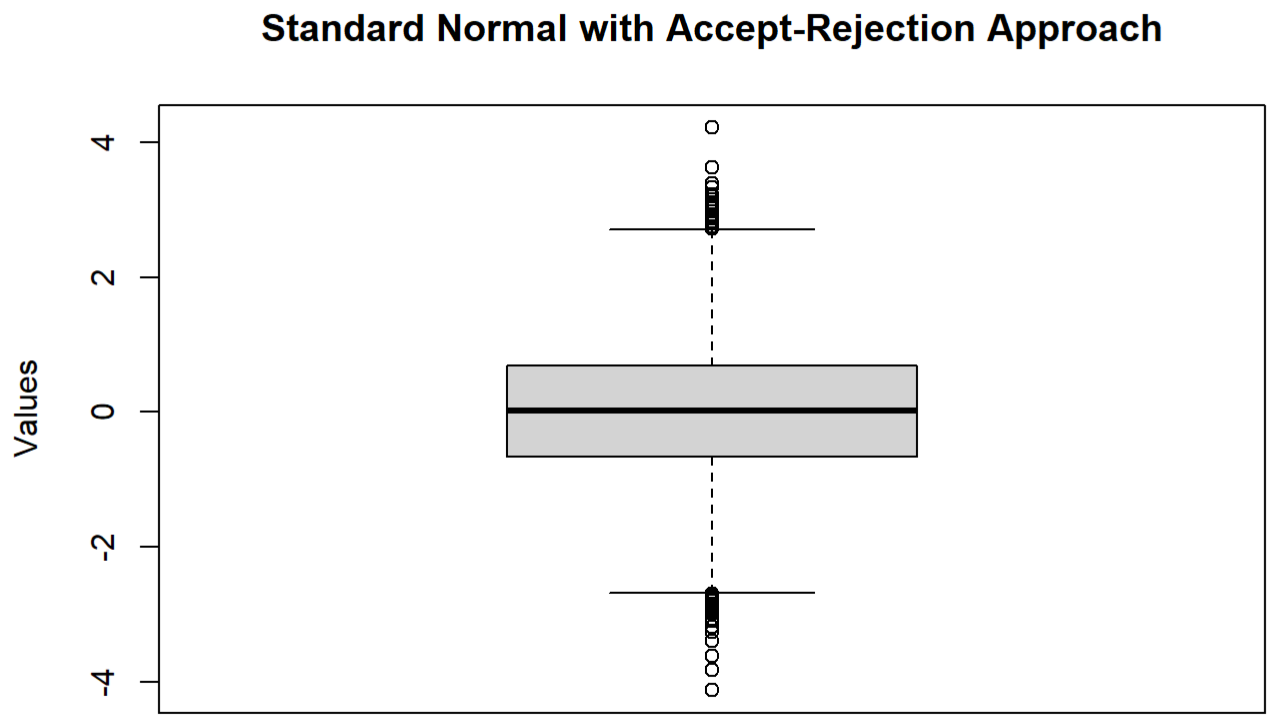
```
exponential <- function(n, lambda){
  res <- vector("numeric", length=n)

  for (i in 1:n) {
    u <- runif(1, 0, 1)

    res[i] <- -(1/lambda) * log(u)
  }

  return(res)
}

normal_acc_rej <- function(n){
  res <- vector("numeric", length=n)
  total_num <- 0
  acc_num <- 0

  for (i in 1:n) {
    y <- exponential(1, 1)
    u1 <- runif(1, 0, 1)
    u2 <- runif(1, 0, 1)
    x <- 0
    total_num <- total_num + 1

    while (!(u1 <= exp(-((y - 1)**2) / 2))) {
      y <- exponential(1, 1)
      u1 <- runif(1, 0, 1)
      u2 <- runif(1, 0, 1)
      total_num <- total_num + 1
    }
    # Accept
    x <- y
    acc_num <- acc_num + 1

    if(u2 <= 0.5){
      x = abs(x)
    }else{
      x = -abs(x)
    }

    res[i] <- x
  }

  print("Acceptance Rate(%)")
  print(100*acc_num/total_num)

  return(res)
}

#n <- 10000
res <- normal_acc_rej(n)
```

```
## [1] "Acceptance Rate(%)"
## [1] 76.41755
```

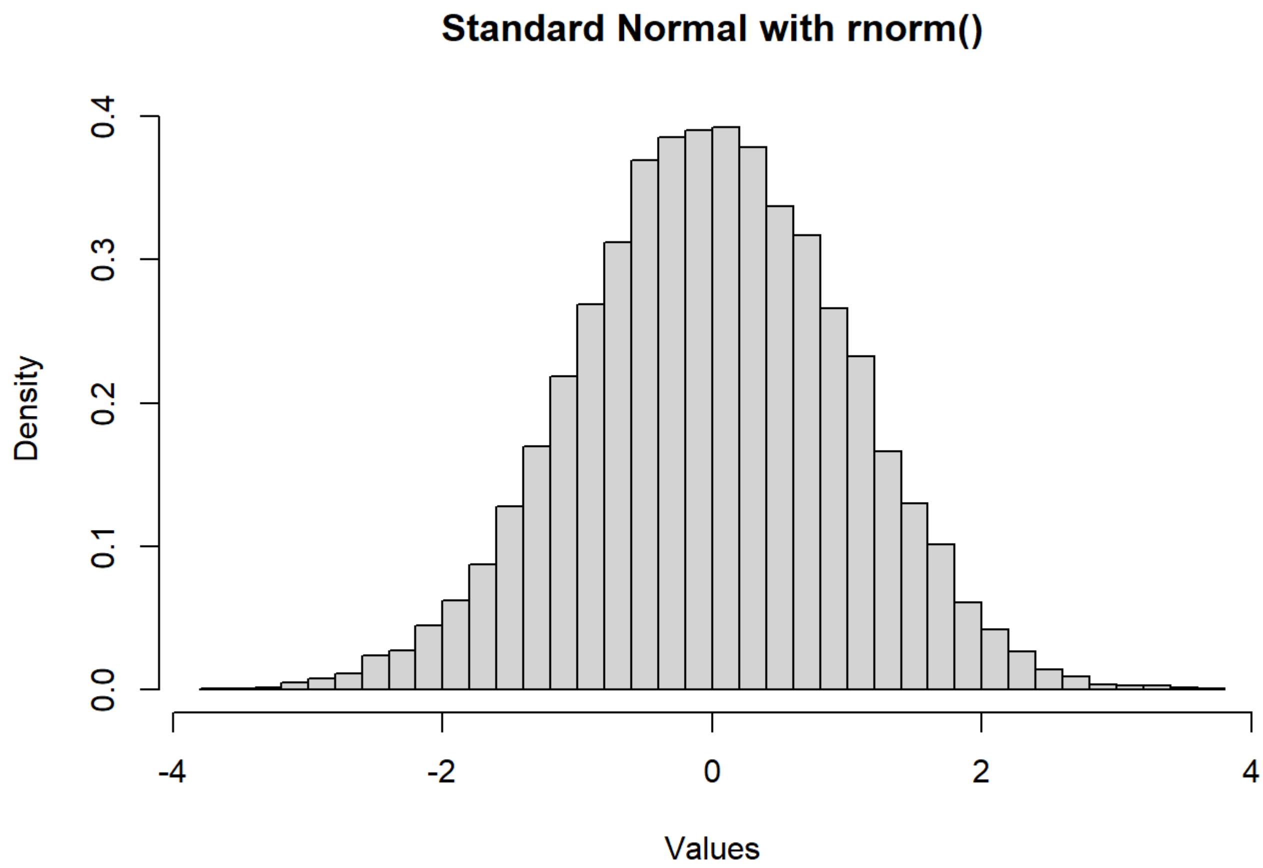
```
hist(res, main="Standard Normal with Accept-Rejection Approach", xlab="Values", breaks=50, freq
 = FALSE)
```



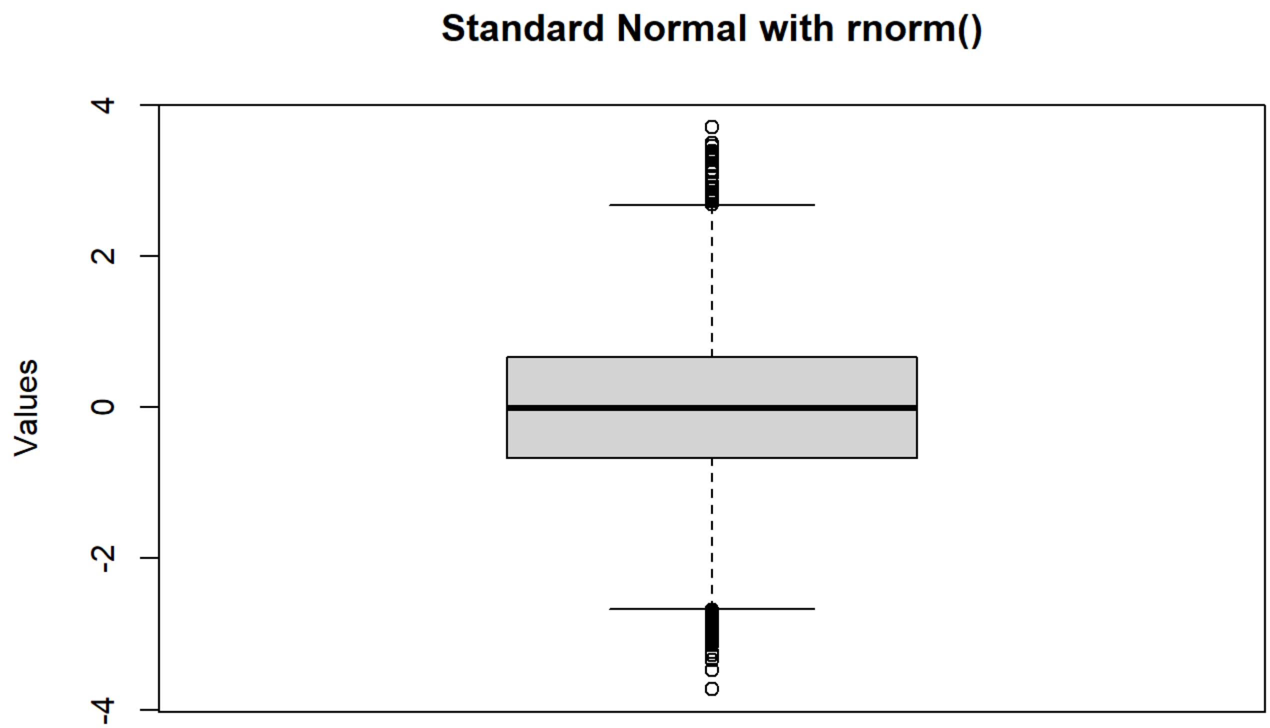

```
boxplot(res, main="Standard Normal with Accept-Rejection Approach", ylab="Values", freq = FALSE)
```




```
r_dist <- rnorm(n, 0, 1)
hist(r_dist, main="Standard Normal with rnorm()", xlab="Values", breaks=50, freq = FALSE)
```




```
boxplot(r_dist, main="Standard Normal with rnorm()", ylab="Values", freq = FALSE)
```



Problem 2:

(a) Generate Poisson distribution with 10000 samples. Display the result by the histogram and the boxplot.

$$X \sim Poisson(\mu = 10)$$

where λ the happening rate of the event during T time and the μ means the average occurrence of the event during T time.

$$\lambda \cdot T = \mu$$

Pseudo Code

For $Poisson(\mu)$

Step 1. Let $t = 0, X = 0$

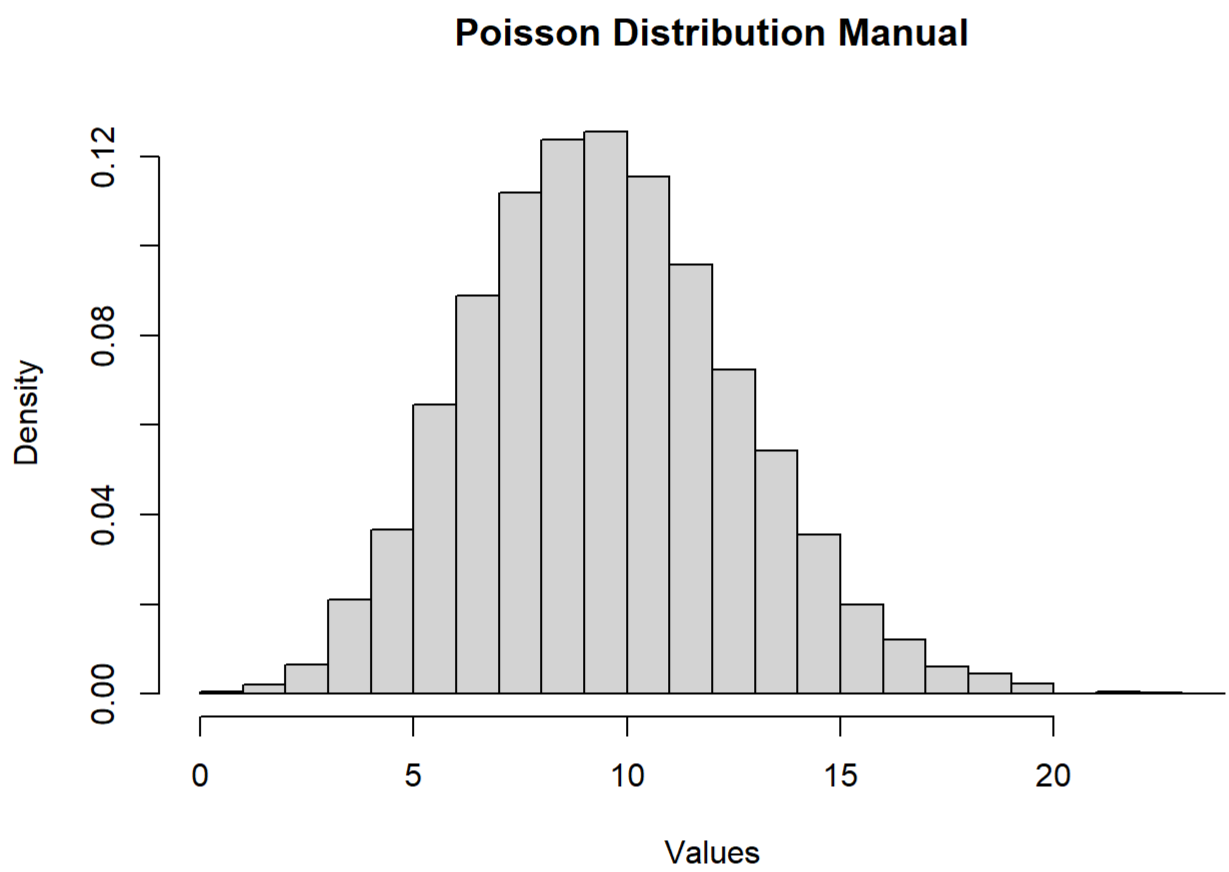
Step 2. If $t \leq \mu$, generate $U \sim U(0, 1)$. Otherwise, go to Step 5.

Step 3. $t = t - log(U)$

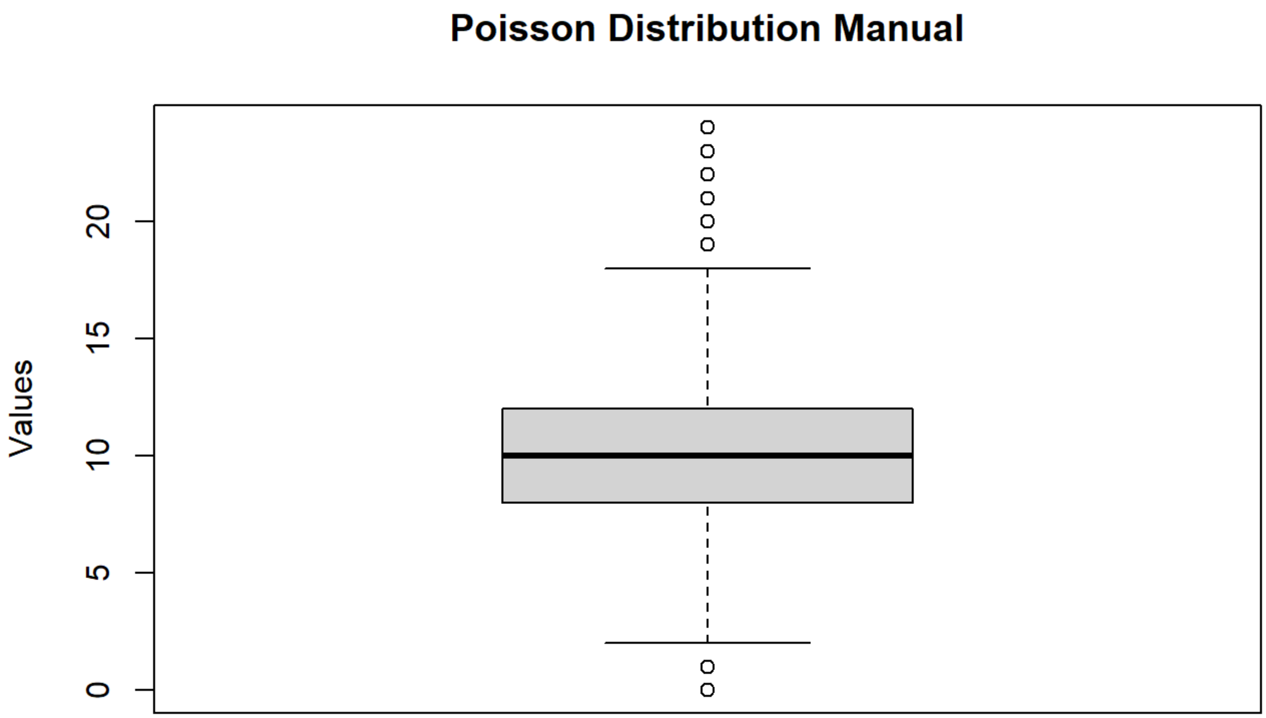
Step 4. if $t \leq \mu$, $X = X + 1$. Otherwise, go back to Step 2.

Step 5. Return X

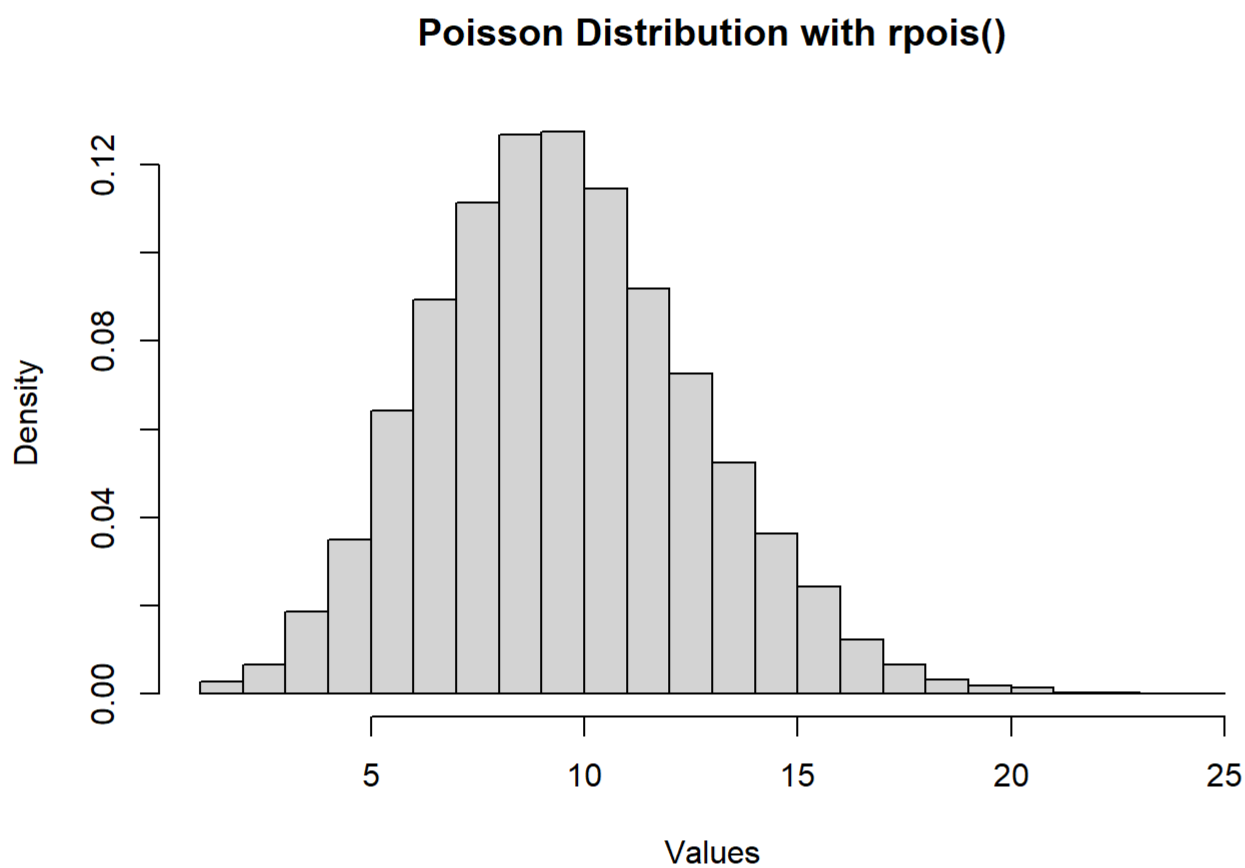
```
poisson <- function(n, mu){
  res <- vector("numeric", length=n)

  for (i in 1:n) {
    T <- mu

    t <- 0
    x <- 0

    while (t <= T) {
      u <- runif(1, 0, 1)
      # Lambda = 1
      t <- t - log(u)

      if(t <= as.numeric(T)){
        x <- x + 1
      }
    }

    res[i] <- x
  }

  return(res)
}

#n <- 10000
mu <- 10
res <- poisson(n, mu)

hist(res, main="Poisson Distribution Manual", xlab="Values", freq = FALSE, breaks=25)
```



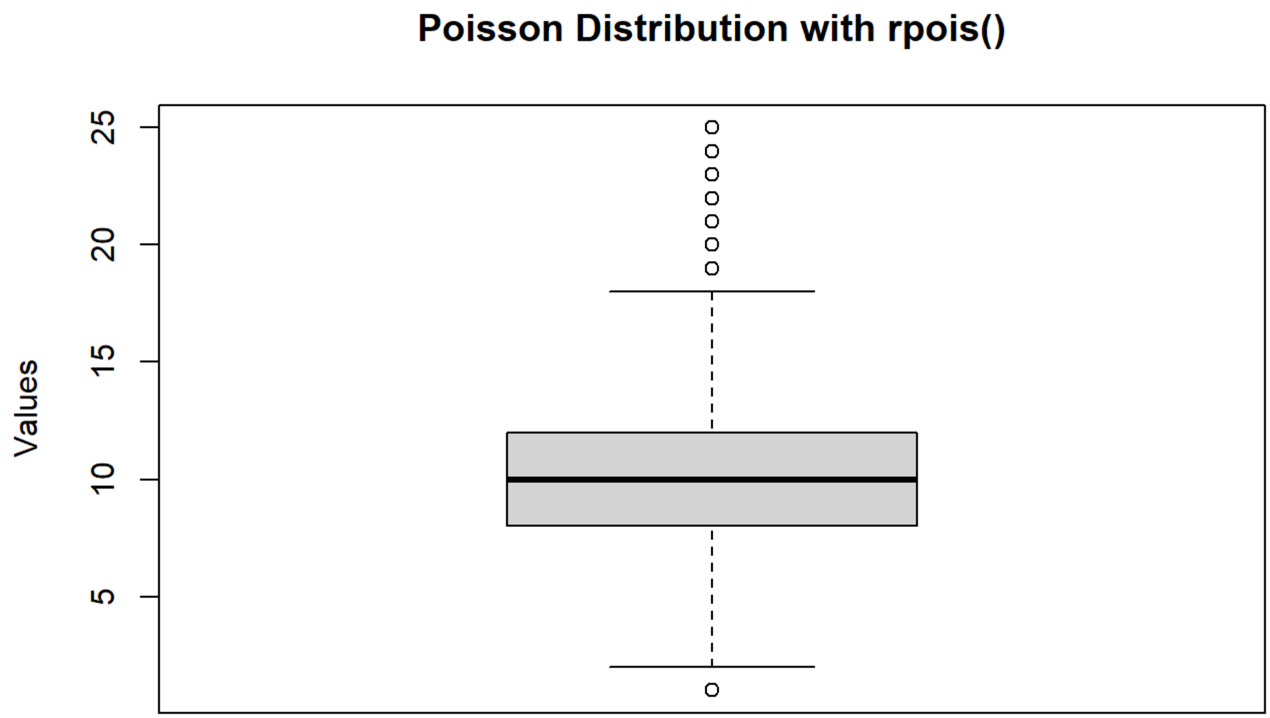

```
boxplot(res, main="Poisson Distribution Manual", ylab="Values", freq = FALSE)
```




```
r_dist <- rpois(n, mu)
hist(r_dist, main="Poisson Distribution with rpois()", xlab="Values", breaks=25, freq = FALSE)
```




```
boxplot(r_dist, main="Poisson Distribution with rpois()", ylab="Values", freq = FALSE)
```



(b) Generate Gamma distribution with 10000 samples. Display the result by the histogram and the boxplot.

$X \sim Gamma(\alpha = 5, \beta = 3)$

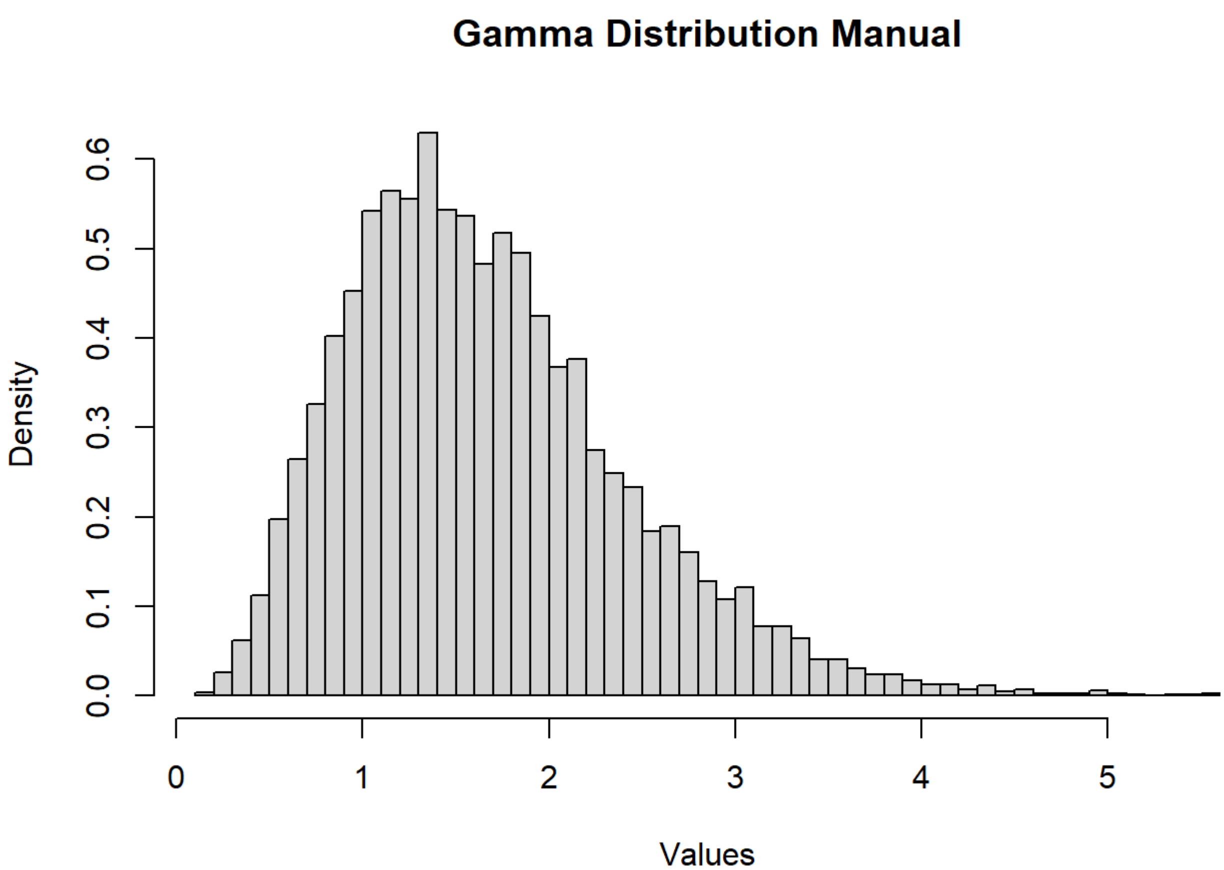
Pseudo Code

For $Gamma(\alpha, \beta)$

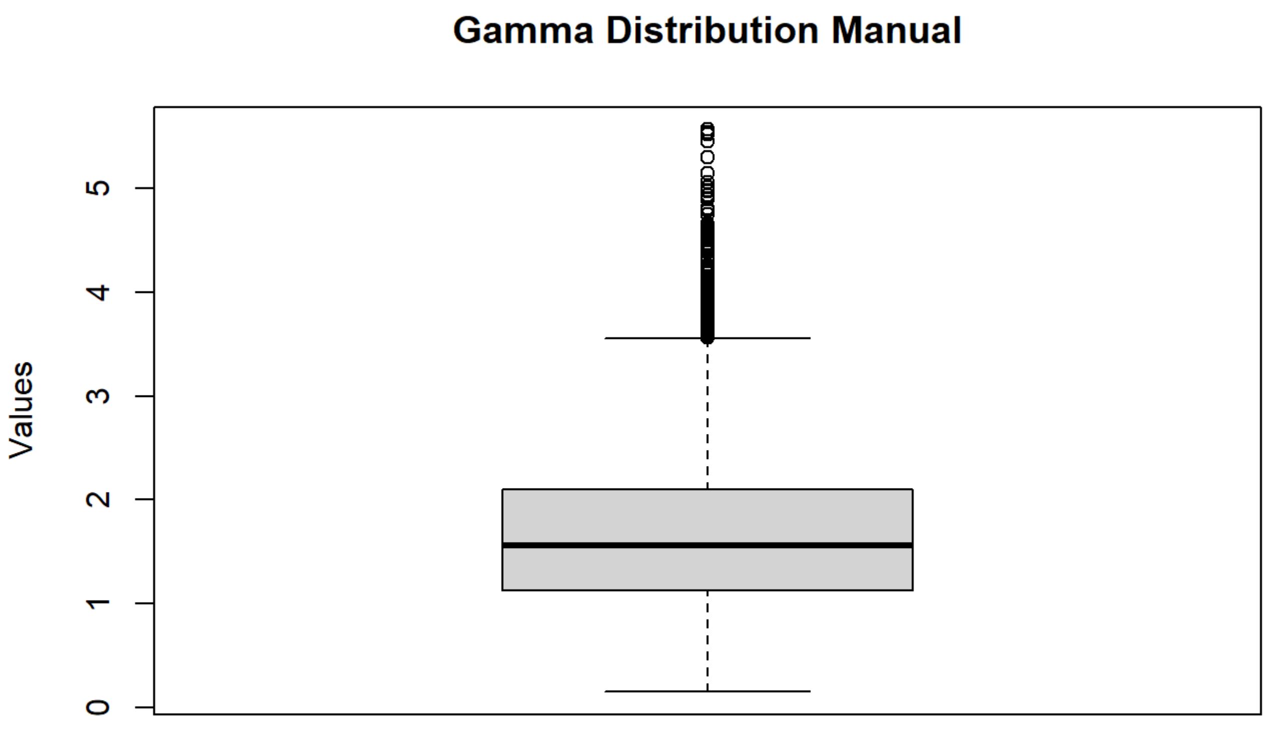
Step 1. Generate $X_1, X_2, \ldots, X_\alpha \overset{i.i.d}{\sim} Exp(\beta)$

Step 2. Return $\sum_{i=1}^{\alpha} X_i$

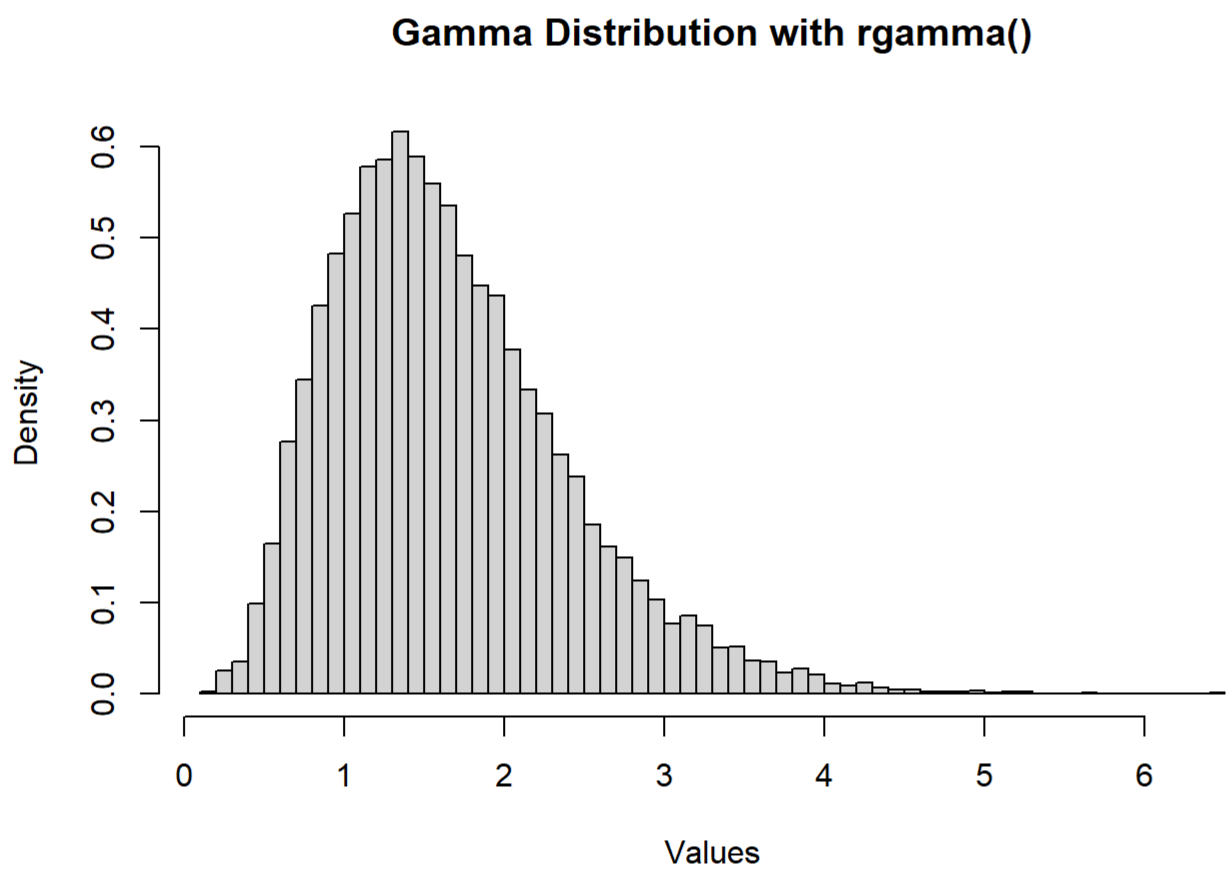
```
gamma <- function(n, alpha, beta){
  res <- vector("numeric", length=n)

  for (i in 0:n) {
    # u <- runif(alpha, 0, 1)
    # y <- vector("numeric", length=alpha)

    # for (i in 0:alpha) {
      #y[i] <- -1 / beta * log(u[i])
    #}
    #res[i] <- sum(y)

    res[i] = sum(exponential(alpha, beta))
  }

  return(res)
}

#n <- 10000
alpha <- 5
beta <- 3
res <- gamma(n, alpha, beta)

hist(res, main="Gamma Distribution Manual", xlab="Values", freq = FALSE, breaks=50)
```



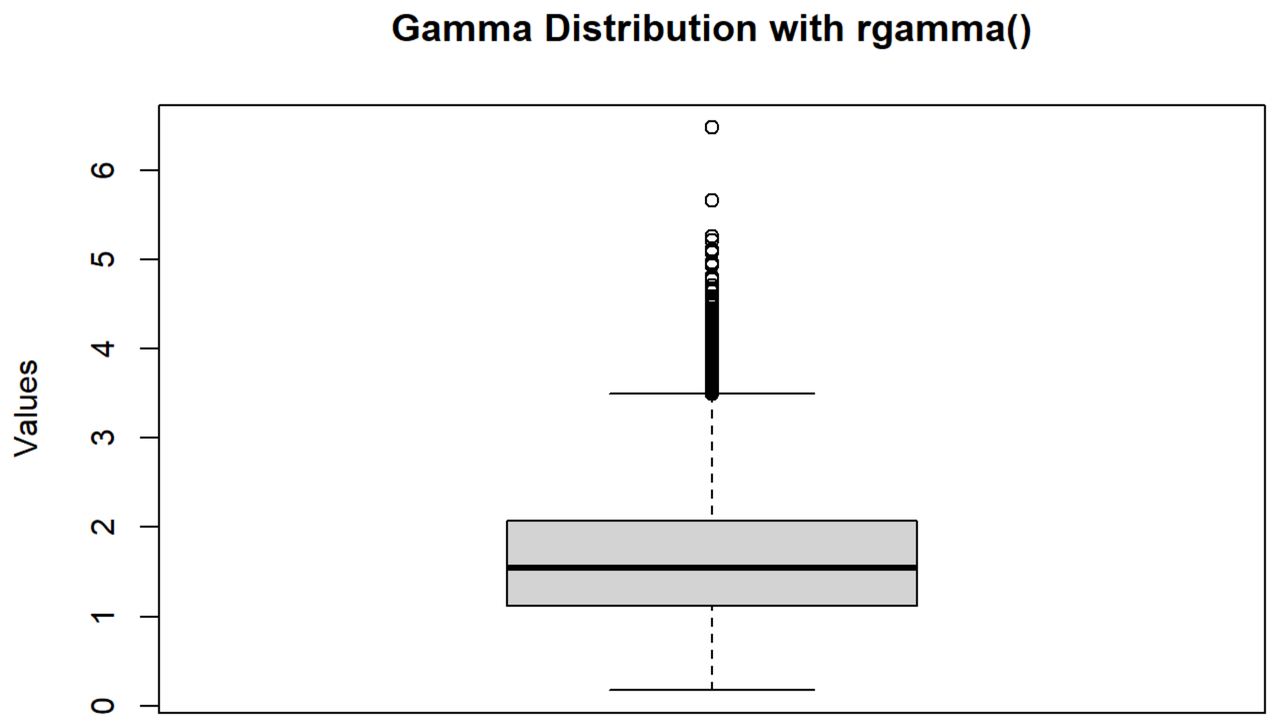
```
boxplot(res, main="Gamma Distribution Manual", ylab="Values", freq = FALSE)
```




```
r_dist <- rgamma(n, shape=alpha, rate=beta)
hist(r_dist, main="Gamma Distribution with rgamma()", xlab="Values", breaks=50, freq = FALSE)
```




```
boxplot(r_dist, main="Gamma Distribution with rgamma()", ylab="Values", freq = FALSE)
```



Problem 3

(a)

Suppose

$$X|\mu \sim Poisson(\mu) = \frac{\mu^x e^{-\mu}}{x!}$$

$$\mu \sim Gamma(\alpha, \beta) = \frac{\mu^{\alpha-1} e^{-\frac{\mu}{\beta}}}{\beta^\alpha \Gamma(\alpha)}$$

The marginal distribution $f_X(x)$ of X is

$$f_X(x) = \int_\mu p(X, \mu)\, d\mu = \int_\mu p(X|\mu)p(\mu)\, d\mu$$

$$= \int_\mu \frac{\mu^x e^{-\mu}}{x!} \cdot \frac{\mu^{\alpha-1} e^{-\frac{\mu}{\beta}}}{\beta^\alpha \Gamma(\alpha)}\, d\mu$$

$$= \frac{1}{x!\Gamma(\alpha)\beta^\alpha} \int_0^\infty \mu^x e^{-\mu} \mu^{\alpha-1} e^{-\frac{\mu}{\beta}}\, d\mu$$

$$= \frac{1}{x!\Gamma(\alpha)\beta^{\alpha}} \int_0^{\infty} \mu^{\alpha+x-1} e^{-\mu(1+\frac{1}{\beta})} \, d\mu$$

$$= \frac{1}{\Gamma(x+1)\Gamma(\alpha)\beta^{\alpha}} \Gamma(\alpha+x) \frac{\beta}{1+\beta}$$

$$= \binom{\alpha-1+x}{x} (\frac{1}{1+\beta})^{\alpha} \, (1-\frac{1}{1+\beta})^{x}$$

Let $n = \alpha, p = \frac{1}{1+\beta}$

$$= \binom{n-1+x}{x} p^n (1-p)^x$$

It is a Negative Binomial distribution $\mathcal{NB}(n,p)$

Pseudo Code Of Geometric

For $Geo(p)$

Step 1. Generate $U \sim U(0,1)$

Step Return $\lfloor \frac{log\, U}{log\, (1-p)} \rfloor$

Pseudo Code Of Negative Binomial

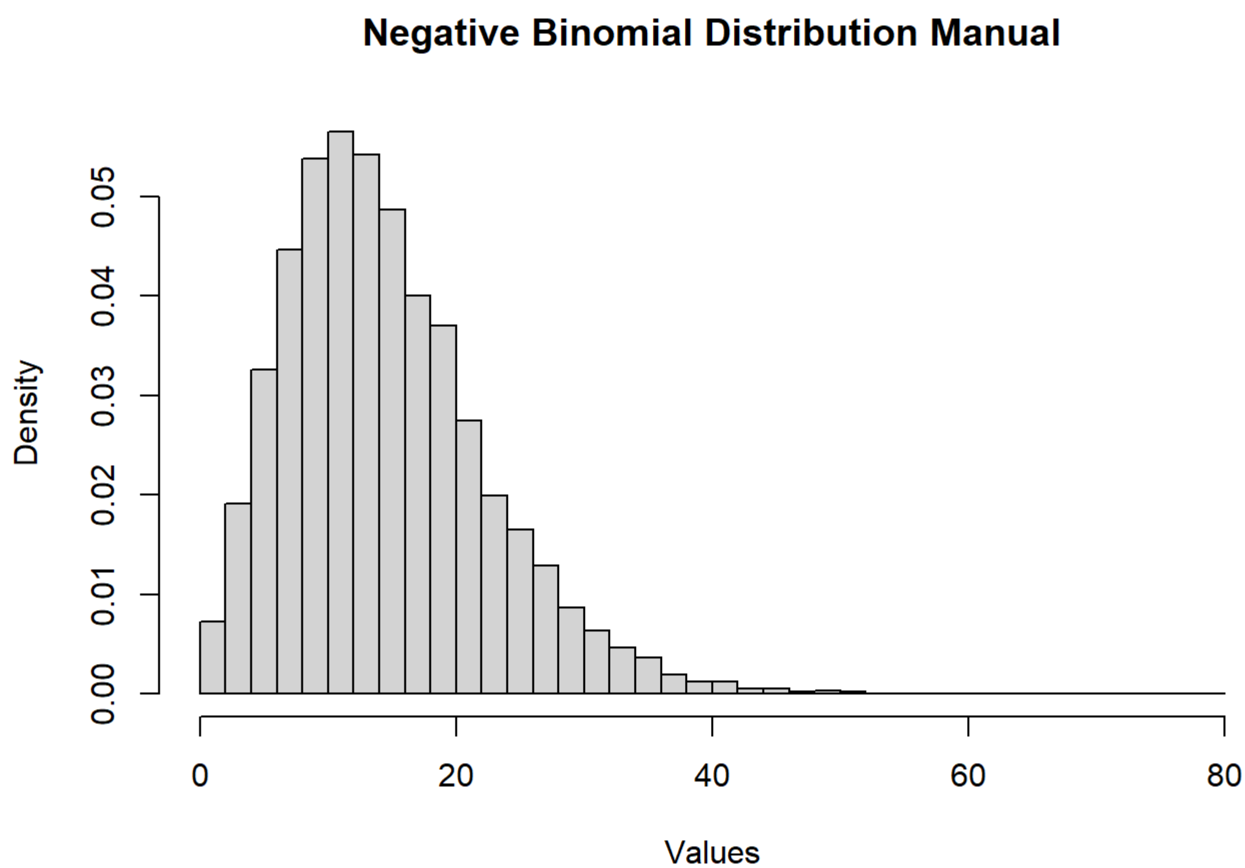
For $NB(n,p)$

Step 1. Generate $X_1, X_2, \ldots, X_n \overset{i.i.d}{\sim} Geo(p)$

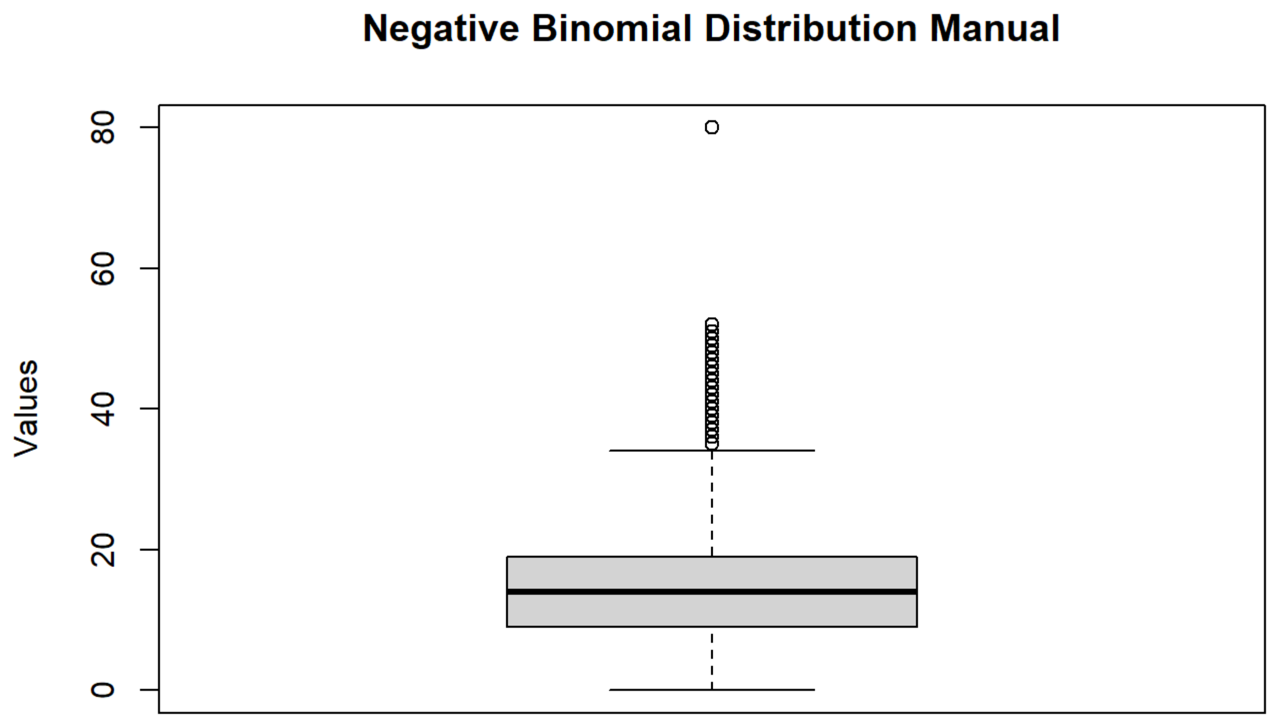
Step 2. Return $\sum_{i=1}^{n} X_i$

(b)

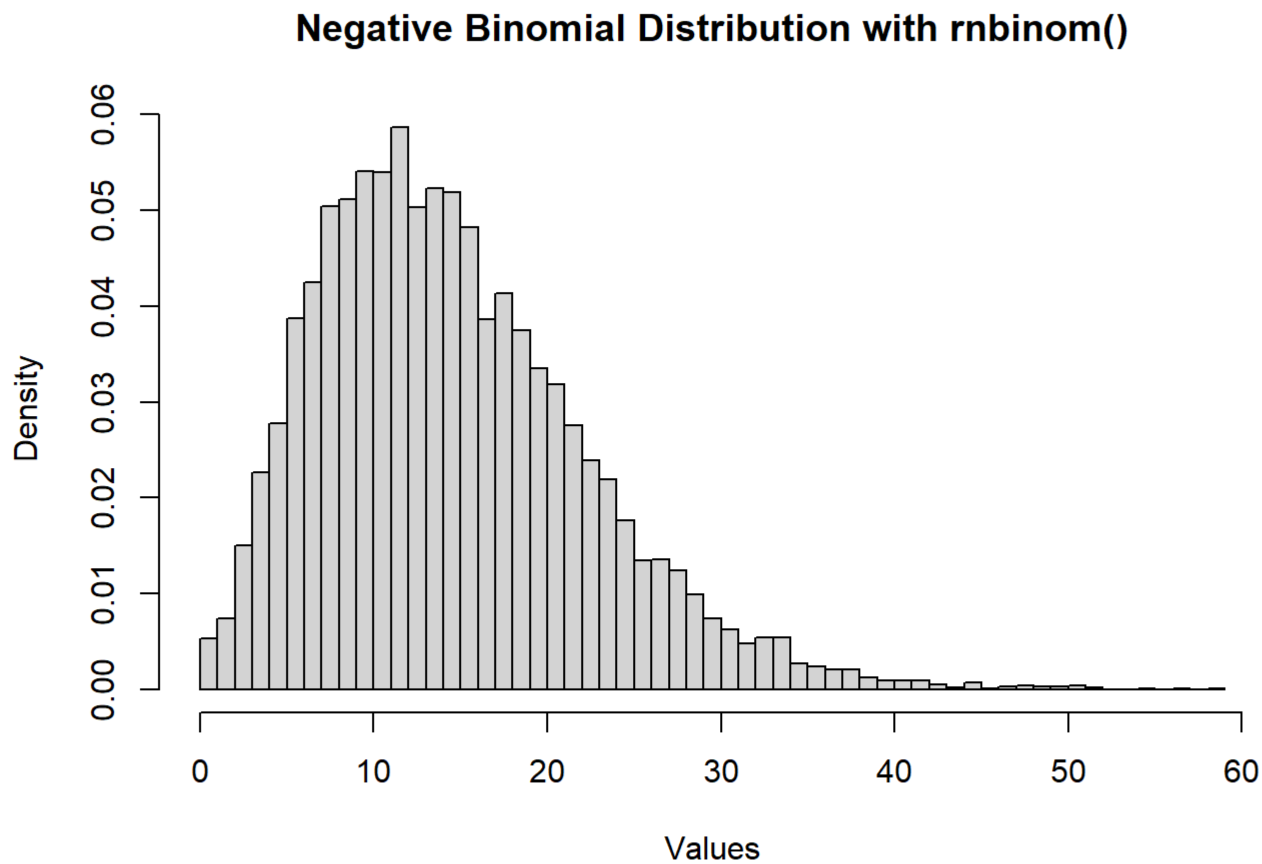
```
geo <- function(n, p){
  res <- vector("numeric", length=n)

  for (i in 1:n) {
    u <- runif(1, 0, 1)
    #print(u)
    #print(log(u))
    #print(log(1 - p))
    res[i] <- floor(log(u) / log(1 - p))
  }

  return(res)
}

nb <- function(n, m, p){
  res <- vector("numeric", length=n)

  for (i in 1:n) {
    geo_res <- vector("numeric", length=m)

    geo_res <- geo(m, p)

    #print(geo_res)

    res[i] <- sum(geo_res)
  }

  return(res)
}

#n <- 10000
alpha <- 5
beta <- 3
res <- nb(n, alpha, 1/(1 + beta))

#print(res)

hist(res, main="Negative Binomial Distribution Manual", xlab="Values", freq = FALSE, breaks=50)
```



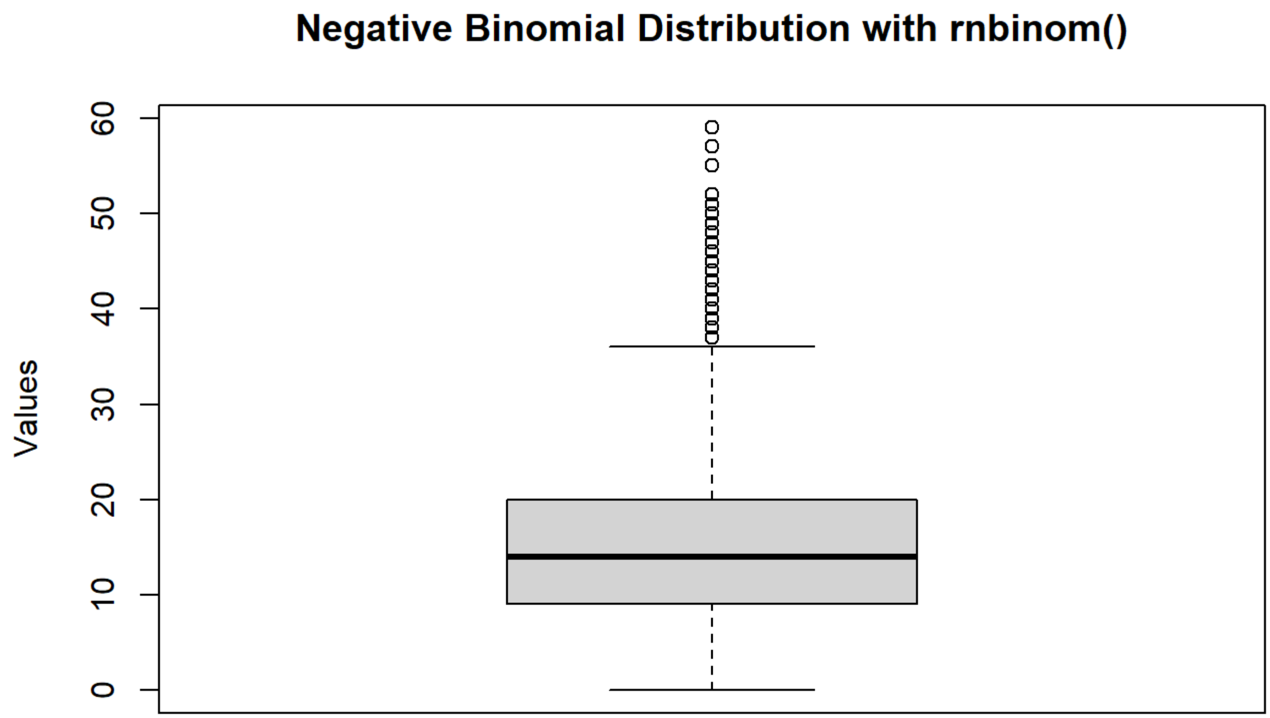

```
boxplot(res, main="Negative Binomial Distribution Manual", ylab="Values", freq = FALSE)
```




```
r_dist <- rnbinom(n, alpha, 1/(1 + beta))
hist(r_dist, main="Negative Binomial Distribution with rnbinom()", xlab="Values", breaks=50, fre
q = FALSE)
```




```
boxplot(r_dist, main="Negative Binomial Distribution with rnbinom()", ylab="Values", freq = FALS
E)
```



(c) What are the mean and variance of X?

Mean

$$\frac{pr}{1-p} = \frac{\frac{1}{1+\beta}\alpha}{1-\frac{1}{1+\beta}} = \frac{\alpha}{\beta} = \frac{5}{3}$$

Variance

$$\frac{pr}{(1-p)^2} = \frac{\frac{1}{1+\beta}\alpha}{(1-\frac{1}{1+\beta})^2} = \frac{\alpha(1+\beta)}{\beta^2} = \frac{20}{9}$$

Problem 4

(a)

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2),\ X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

A mixture model of X_1, X_2

$$f_{X_1,X_2}(x) = p_1 \cdot p_{X_1}(x) + p_2 \cdot p_{X_2}(x)$$

$$= p_1 \cdot \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_1}{\sigma_1})^2} + p_2 \cdot \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_2}{\sigma_2})^2}$$

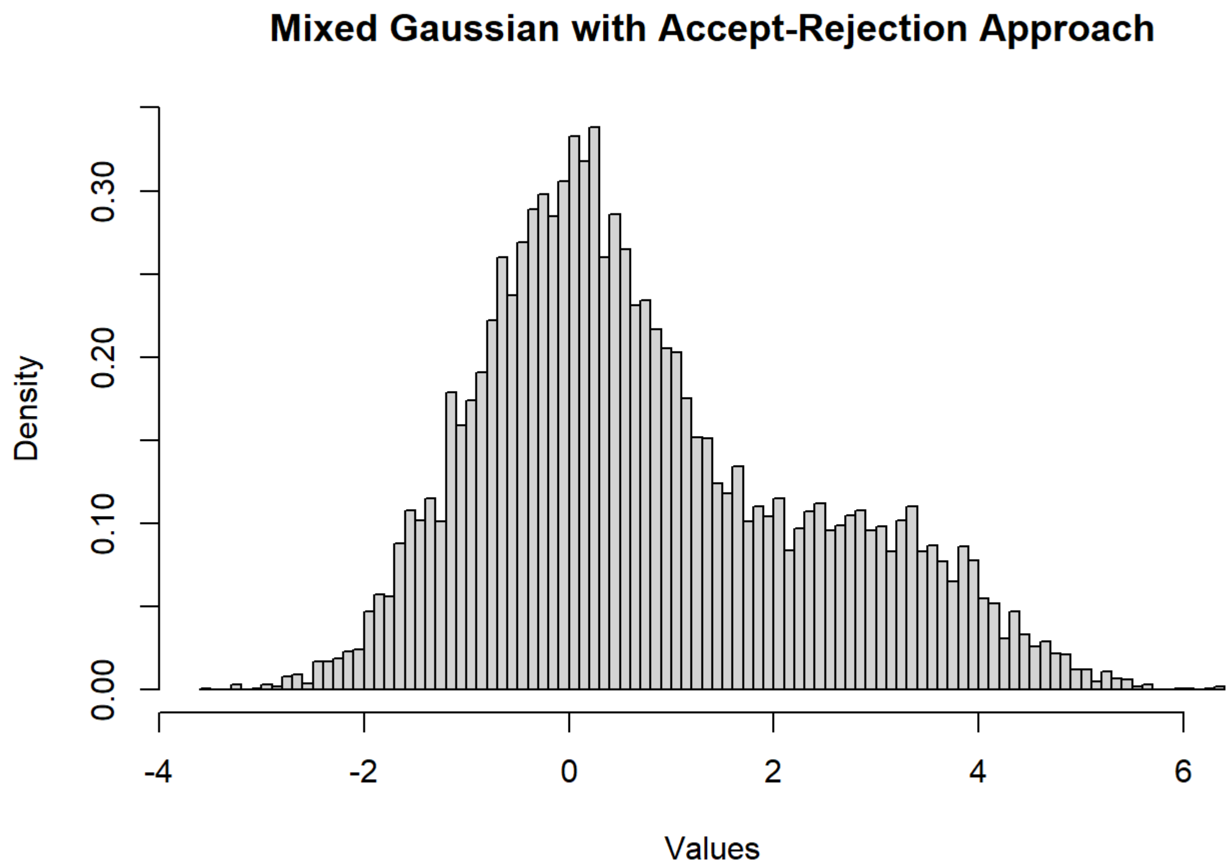
Let $\mu_1 = 0, \mu_2 = 3$ and $\sigma_1^2 = \sigma_2^2 = 1$

$$= p_1 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} + (1 - p_1) \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-3)^2}$$

(b)

Let $p_1 = 0.75$ and generate 10000 samples from the mixture model.

```
mix_acc_rej <- function(n, p_1, mu_1, mu_2, sigma_1, sigma_2){
  res <- vector("numeric", length=n)

  for (i in 0:n) {
    p <- runif(1, 0, 1)
    shift <- 0
    scale <- 0
    if(p <= p_1){
      shift <- mu_1
      scale <- sigma_1
    }else{
      shift <- mu_2
      scale <- sigma_2
    }

    y <- exponential(1, 1)
    u1 <- runif(1, 0, 1)
    u2 <- runif(1, 0, 1)
    x <- 0

    while (!(u1 <= exp(-((y - 1)**2) / 2))) {
      y <- rexp(1, 1)
      u1 <- runif(1, 0, 1)
      u2 <- runif(1, 0, 1)
    }
    # Accept
    x <- y

    if(u2 <= 0.5){
      x = abs(x)
    }else{
      x = -abs(x)
    }

    x <- x * scale + shift

    res[i] <- x
  }

  return(res)
}

#n <- 10000
res <- mix_acc_rej(n, 0.75, 0, 3, 1, 1)

hist(res, main="Mixed Gaussian with Accept-Rejection Approach", xlab="Values", breaks=100, freq
 = FALSE)
```



The distribution seems bimodal.