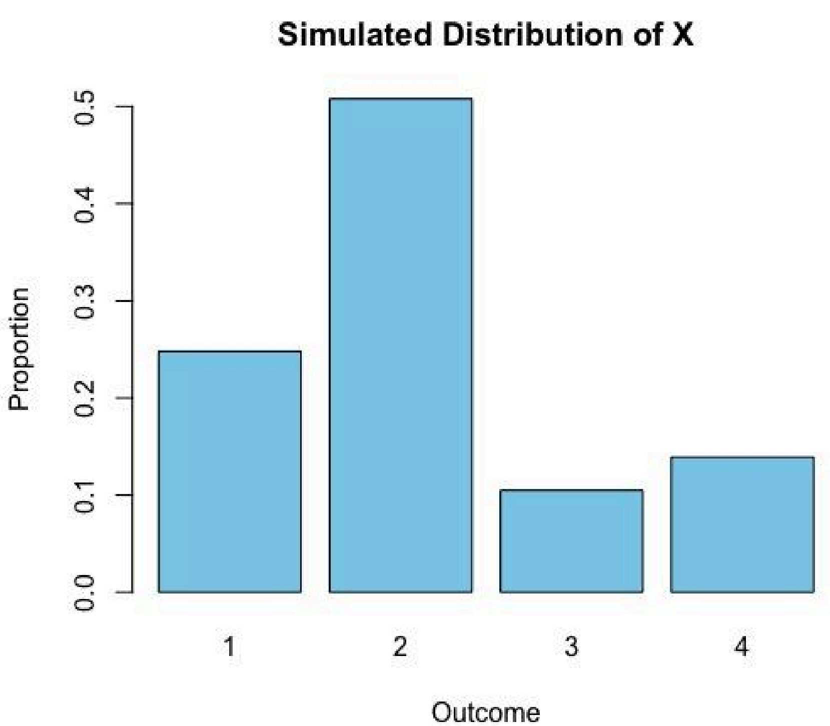
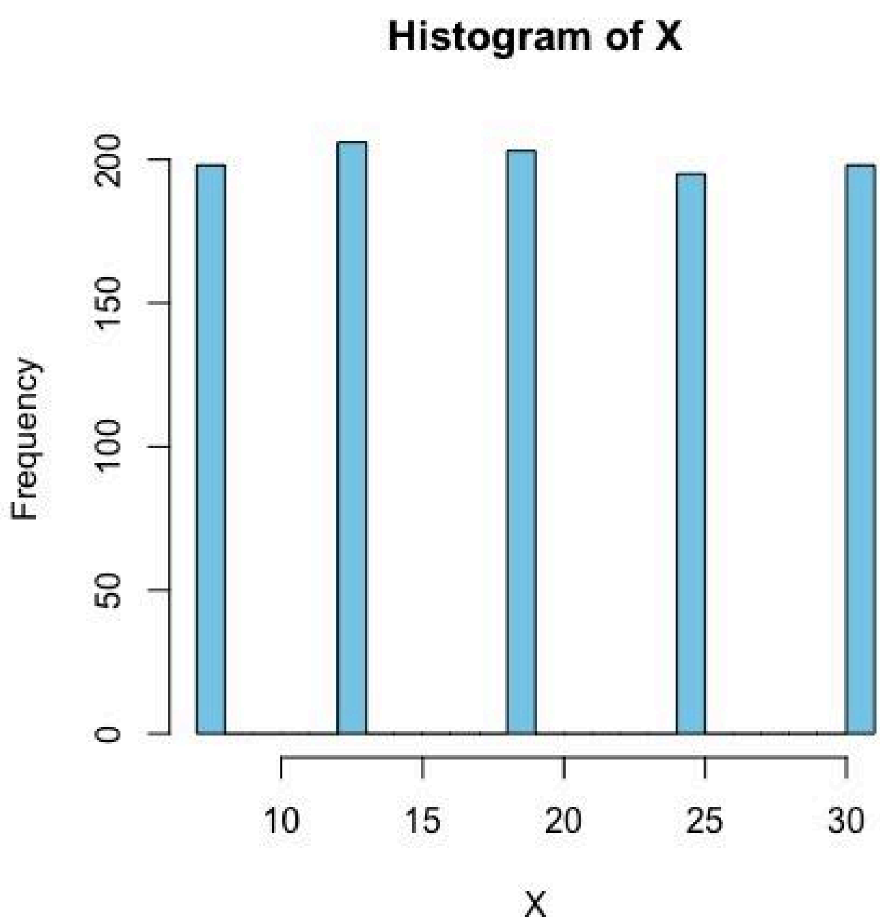
1.

```r
# Function to simulate rand X
simulate_X <- function(n) {
  # generate uniform random number
  rn <- runif(n)

  # Initialize counts
  counts <- c(0, 0, 0, 0)

  # Counting occurrences of each outcome
  for (i in 1:n) {
    if (rn[i] < 0.25) {
      counts[1] <- counts[1] + 1
    } else if (rn[i] < 0.75) {
      counts[2] <- counts[2] + 1
    } else if (rn[i] < 0.85) {
      counts[3] <- counts[3] + 1
    } else {
      counts[4] <- counts[4] + 1
    }
  }

  # Calculatingg proportions
  proportions <- counts / n

  # Returning proportions
  return(proportions)
}

# Simulate 1000 samples
simulations <- simulate_X(1000)

#plot
barplot(simulations, names.arg = c(1, 2, 3, 4), xlab = "Outcome", ylab = "Proportion", main = "Simulated Distribution of X")
```



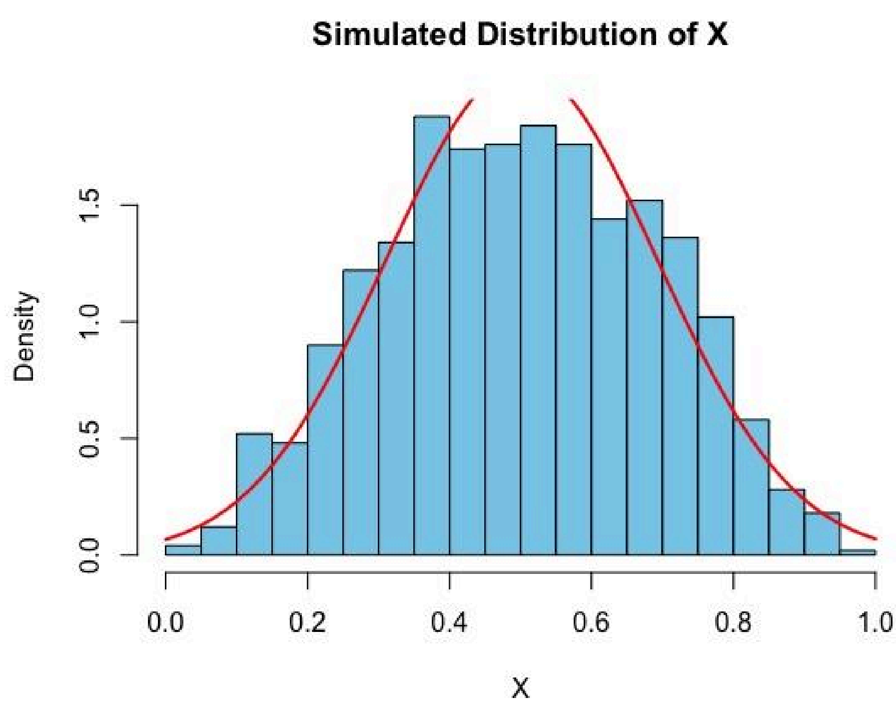
2. Work shown in writing

```r
# homework1 problem2
# Setting seed
set.seed(123)

# Number of simulations
n <- 1000

# Generating n uniform random number between 0 and 1
U <- runif(n)

# Calculate X using the given formula
X <- 6 * floor(5 * U) + 7

#histogram of X
hist(X, breaks = 20, col = "skyblue", main = "Histogram of X", xlab = "X")
```



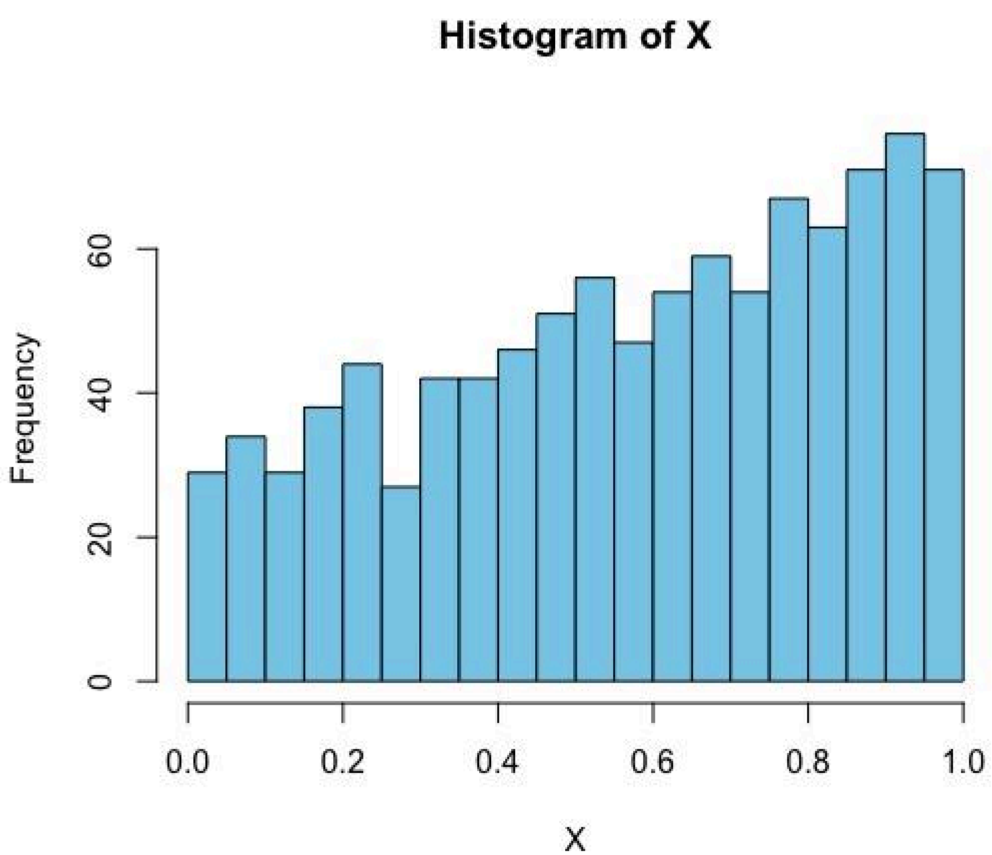
3. Work showed in writing

```
# Define the density function h(x)
h <- function(y) {
  return(30 * (y^2 - 2 * y^3 + y^4))
}

# Define the calculated constant c
c <- 1.875

# Generate random variable using rejection-acceptance method
generate_X <- function(n) {
  X <- numeric(n)
  count_accept <- 0

  while (count_accept < n) {
    y <- runif(1)  # Generate a uniform random variable y

    # Generate a uniform random variable u2 for acceptance-rejection
    u2 <- runif(1)

    # Check acceptance condition
    if (u2 <= h(y)/c) {
      X[count_accept + 1] <- y
      count_accept <- count_accept + 1
    }
  }

  return(X)
}

# Simulate 1000 samples
simulations <- generate_X(1000)

# Create density plot
hist(simulations, breaks = 30, col='skyblue', freq = FALSE, main = "Simulated Distribution of X", xlab = "X", ylab = "Density")
curve(dnorm(x, mean(simulations), sd(simulations)), col = "red", lwd = 2, add = TRUE, yaxt = "n")
```



4. Work shown in writing

```
# Set the seed for reproducibility
set.seed(123)

# Number of simulations
n <- 1000

# Generate n uniform random numbers between 0 and 1
U <- runif(n)

# Calculate X using the given formula
X <- log(U * (exp(1) - 1) + 1)

# Plot the histogram of X
hist(X, breaks = 20, col = "skyblue", main = "Histogram of X", xlab = "X")
```



5. Worked shown in writing

```r
# h(x)
h <- function(y) {
  return(3/sqrt(pi) * y^(1/2) * exp(-y/3))
}

# Generate random variable using rejection acceptance method
generate_X <- function(n) {
  X <- numeric(n)
  count_accept <- 0

  while (count_accept < n) {
    y <- -3/2 * log(1 - runif(1))
    u2 <- runif(1)
    if (u2 <= h(y)/1.275) {
      X[count_accept + 1] <- y
      count_accept <- count_accept + 1
    }
  }

  return(X)
}

# Simulate 1000 samples
simulations <- generate_X(1000)

# Create bar plot
hist(simulations, breaks = 30, freq = FALSE, main = "Simulated Distribution of X", xlab = "X", ylab = "Density")

# Overlay the density function
curve(f(x), add = TRUE, col = "red", lwd = 2, n = 1000, from = 0)
```