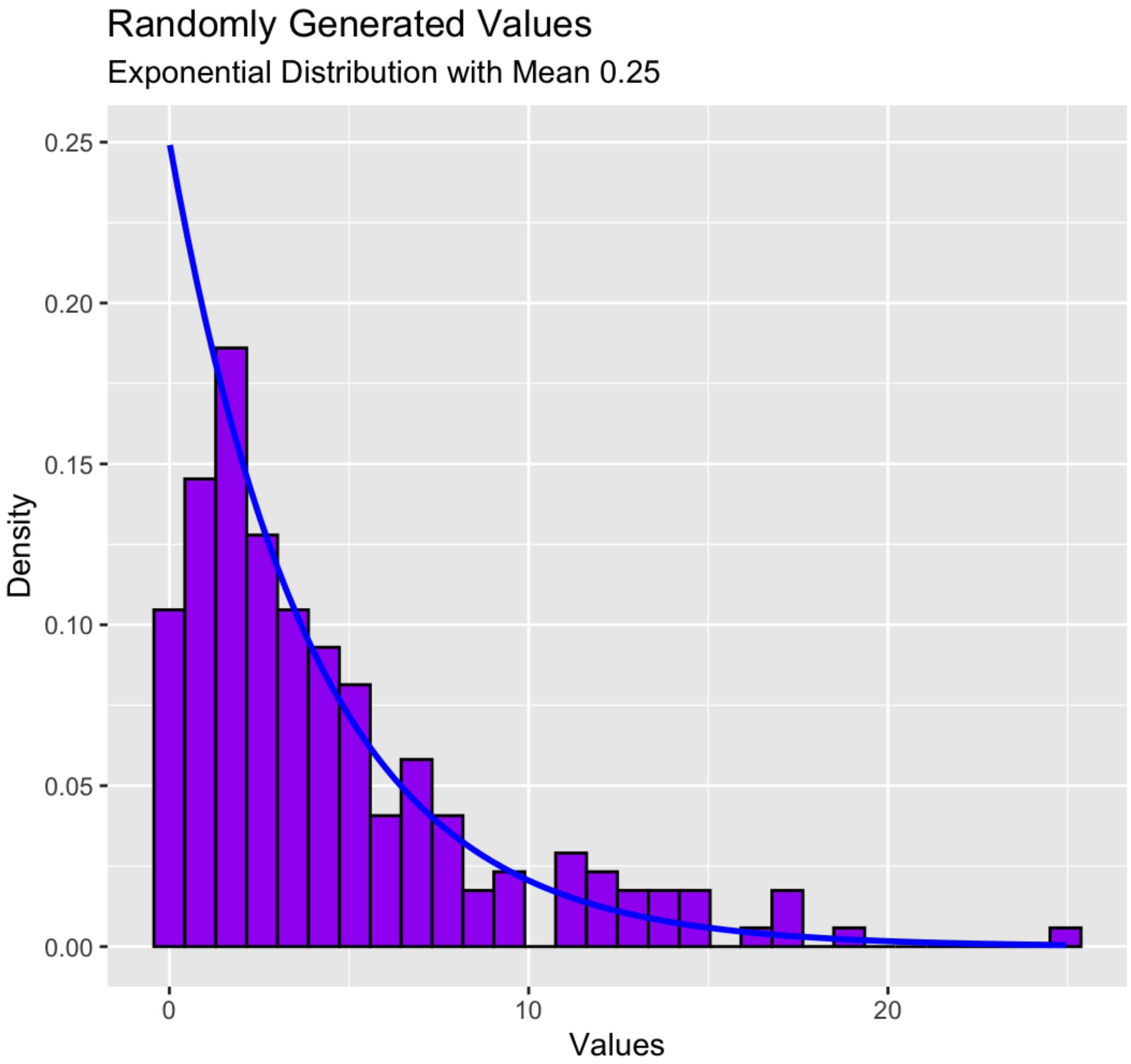
Lab 7
Sebastian Martinez

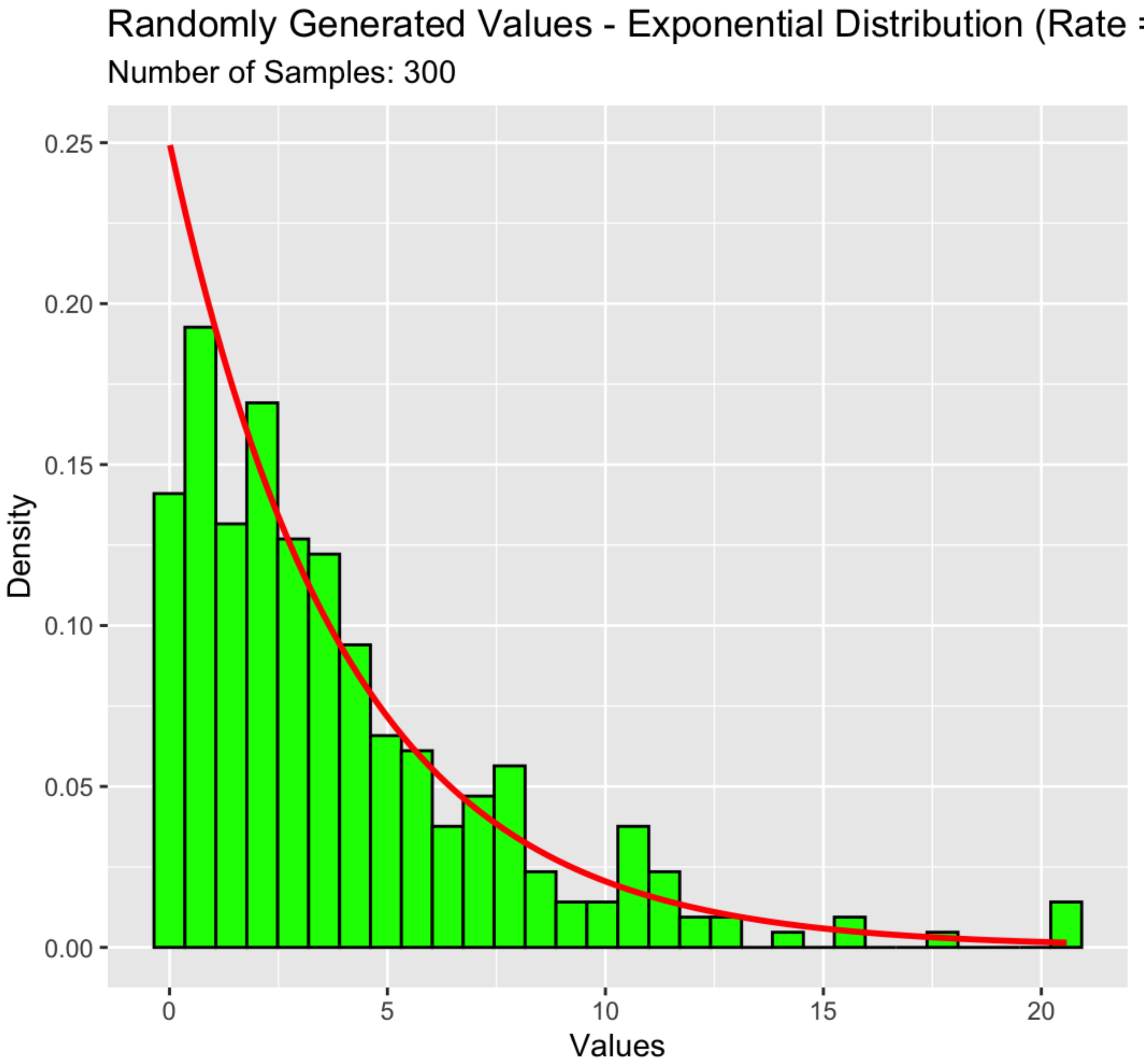
Problem 1

```
library("ggplot2")
x1 <- rexp(200,0.25)

ggplot(data.frame(x1), aes(x1)) +
  geom_histogram(aes(x=x1, y=after_stat(density)),
                 bins=30,
                 fill="purple",
                 colour="black") +
  labs(x="Values", y="Density") +
  ggtitle(label="Randomly Generated Values",
          subtitle="Exponential Distribution with Mean 0.25") +
  stat_function(fun = function(x) dexp(x, rate = 0.25),
                color="blue",
                linewidth=1)
```



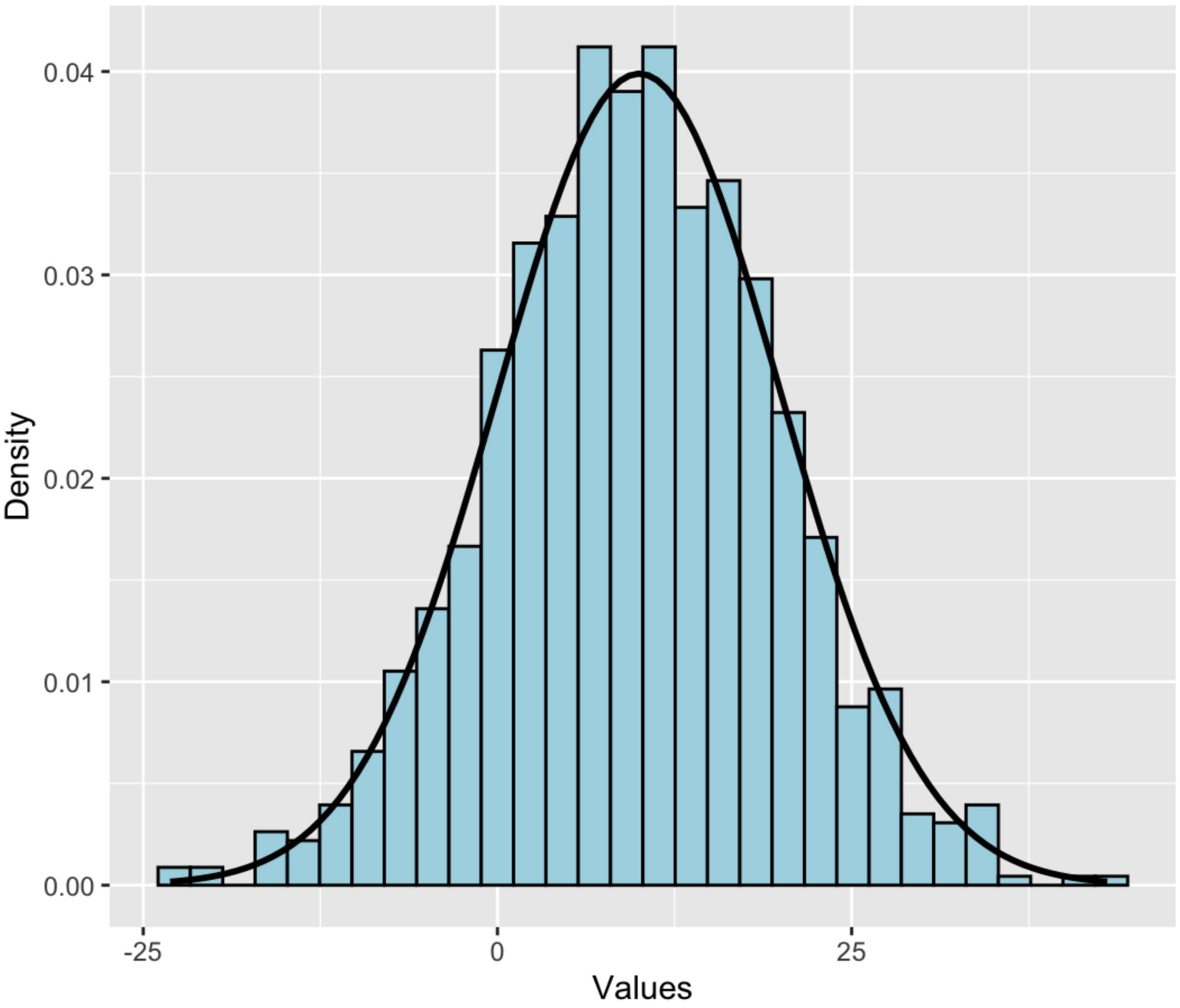
Problem 2

```
expplot <- function(rate, n, line_color, fill_color ) {
  x <- rexp(n, rate)
    ggplot(data.frame(x), aes(x)) +
    geom_histogram(aes(x = x, y = after_stat(density)),
                   bins = 30,
                   fill = fill_color,
                   colour = "black") +
    labs(x = "Values", y = "Density") +
    ggtitle(label = paste("Randomly Generated Values - Exponential Distribution (Rate =", rate,
")", sep = " "),
            subtitle = paste("Number of Samples:", n)) +
    stat_function(fun = function(x) dexp(x, rate = rate),
                  color = line_color,
                 linewidth = 1)
}
expplot(rate = 0.25, n = 300, line_color = "red", fill_color = "green")
```



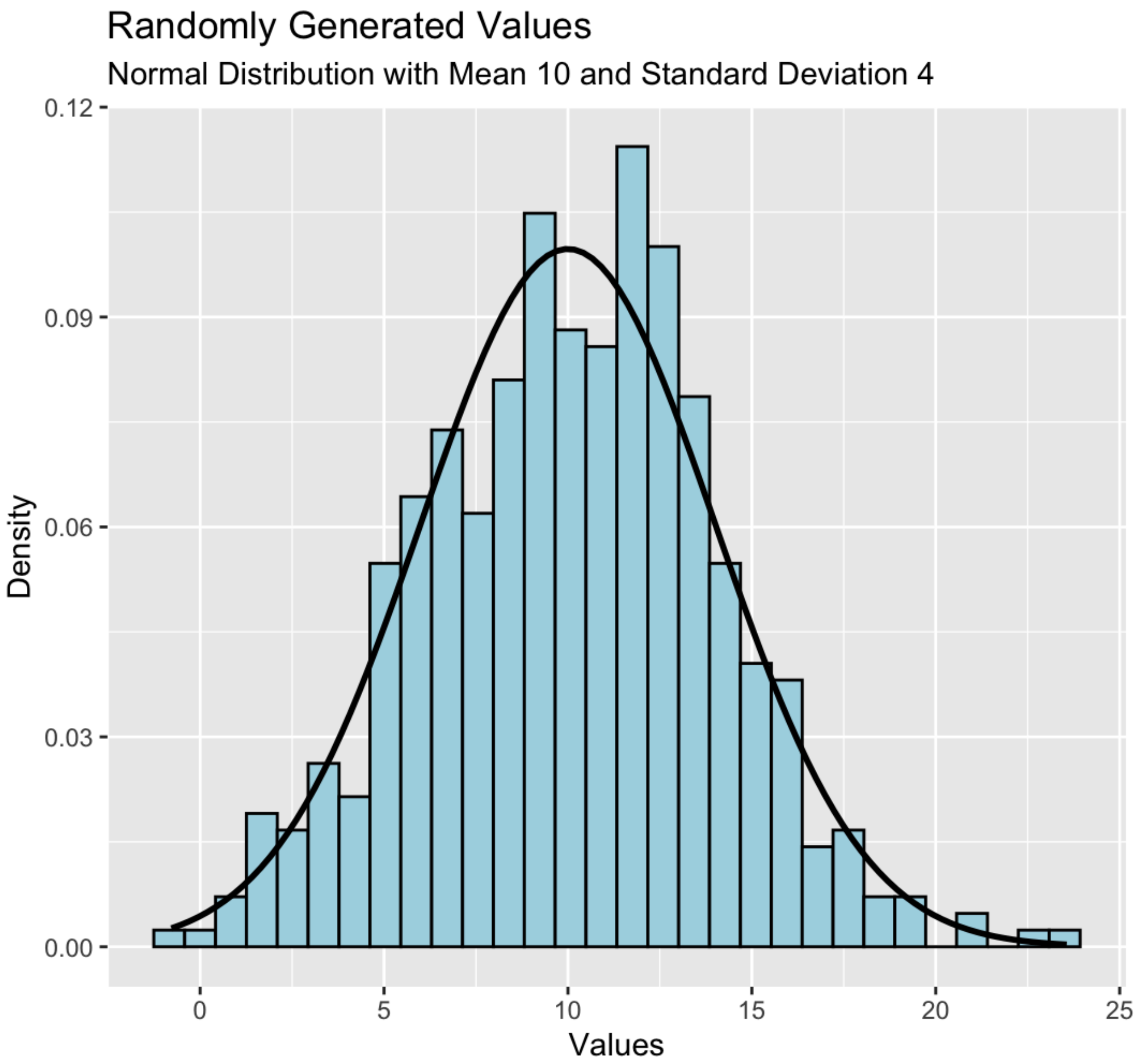
Problem 3

```
norm.or.exp.opt <- function(n, mean_val, sd_val = mean_val) {
  dist <- sample(c("Exponential", "Normal"), 1, prob = c(0.4, 0.6))

  if(dist == "Exponential") {
    x1 <- rexp(n, 1/mean_val)
    st <- paste("Exponential Distribution with Mean ", mean_val, sep="")
    fn <- function(x) dexp(x, rate = 1/mean_val)
  } else if(dist == "Normal") {
    x1 <- rnorm(n, mean_val, sd_val)
    st <- paste("Normal Distribution with Mean ", mean_val,
          " and Standard Deviation ", sd_val, sep="")
    fn <- function(x) dnorm(x, mean = mean_val, sd = sd_val)
  }

  ggplot(data.frame(x1), aes(x1)) +
    geom_histogram(aes(x=x1, y=after_stat(density)),
            bins=30,
            fill="lightblue",
            colour="black") +
    labs(x="Values", y="Density") +
    ggtitle(label="Randomly Generated Values",
        subtitle=st) +
    stat_function(fun = fn,
           color="black",
           linewidth=1)
}

set.seed(123)
norm.or.exp.opt(n = 1000, mean_val = 10)
norm.or.exp.opt(n = 500, mean_val = 10, sd_val = 4)
```



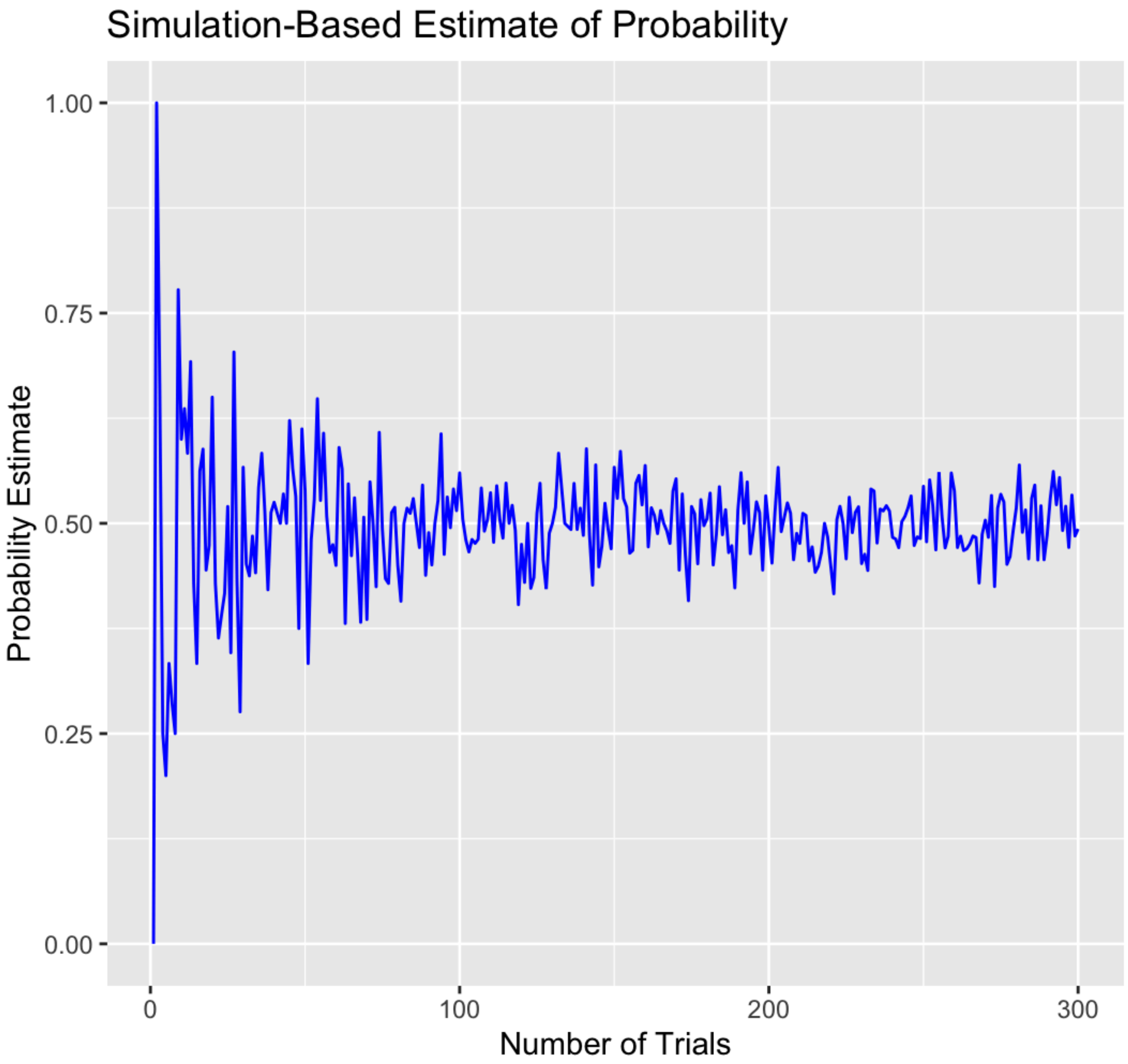
Randomly Generated Values

Normal Distribution with Mean 10 and Standard Deviation 10



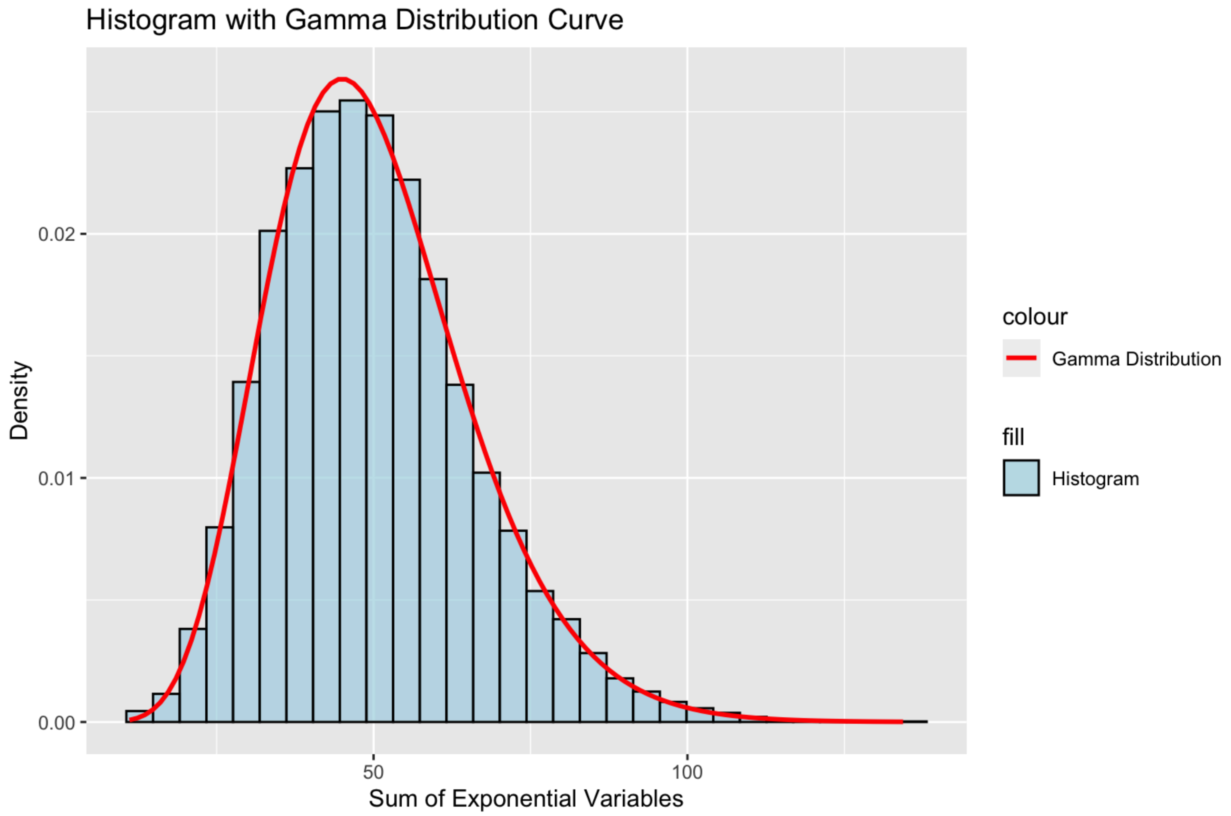
Problem 4

```
library(ggplot2)

calculate_probability <- function(numDice, numSides, targetValue, numTrials) {
  sum_high_enough <- HighRoll(numDice, numSides, targetValue, numTrials)
  mean(sum_high_enough)
}

numDice <- 3
numSides <- 6
targetValue <- 11
num_trials_range <- 1:300

probabilities <- sapply(num_trials_range, function(numTrials) {
  calculate_probability(numDice, numSides, targetValue, numTrials)
})

plot_data <- data.frame(numTrials = num_trials_range, Probability = probabilities)

ggplot(plot_data, aes(x = numTrials, y = Probability)) +
  geom_line(color = "blue") +
  labs(title = "Simulation-Based Estimate of Probability",
       x = "Number of Trials",
       y = "Probability Estimate")
```



Problem 5

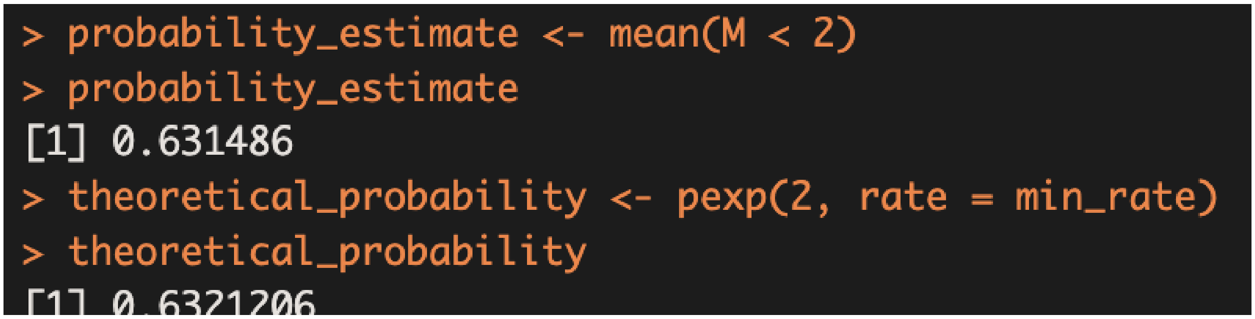
```
library(ggplot2)

plot_gamma_distribution <- function(reps, N, M) {
  random_values <- replicate(reps, sum(rexp(n = N, rate = M)))
    plot_data <- data.frame(random_values)
    histogram <- ggplot(plot_data, aes(x = random_values)) +
    geom_histogram(aes(y = after_stat(density), fill = "Histogram"), bins = 30, color = "black",
alpha = 0.7) +
        stat_function(fun = function(x) dgamma(x, shape = N, scale = 1/M),
                  aes(color = "Gamma Distribution"), linewidth = 1) +
        labs(x = "Sum of Exponential Variables", y = "Density",
         title = "Histogram with Gamma Distribution Curve") +
        scale_fill_manual(values = "lightblue") +
    scale_color_manual(values = "red")

  print(histogram)
}

plot_gamma_distribution(reps = 10000, N = 10, M = 0.2)
```



Problem 6

```
rate_X1 <- 0.3
```

```
rate_X2 <- 0.2
min_rate <- rate_X1 + rate_X2
replicates <- 500000
X1 <- rexp(replicates, rate = rate_X1)
X2 <- rexp(replicates, rate = rate_X2)
M <- pmin(X1, X2)

probability_estimate <- mean(M < 2)
probability_estimate
theoretical_probability <- pexp(2, rate = min_rate)
theoretical_probability
```



```
> probability_estimate <- mean(M < 2)
> probability_estimate
[1] 0.631486
> theoretical_probability <- pexp(2, rate = min_rate)
> theoretical_probability
[1] 0.6321206
```