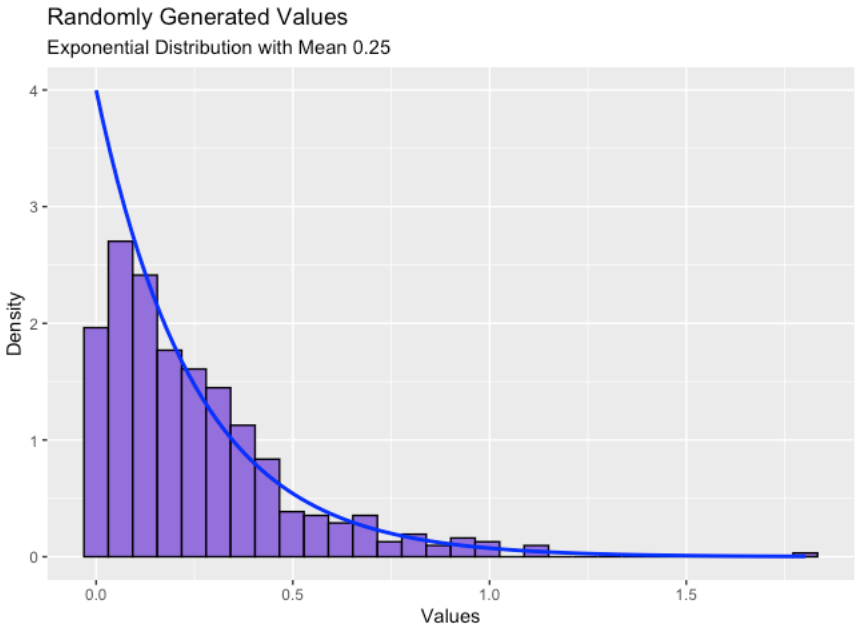
Name: Yuhui Wang
UID: 606332401

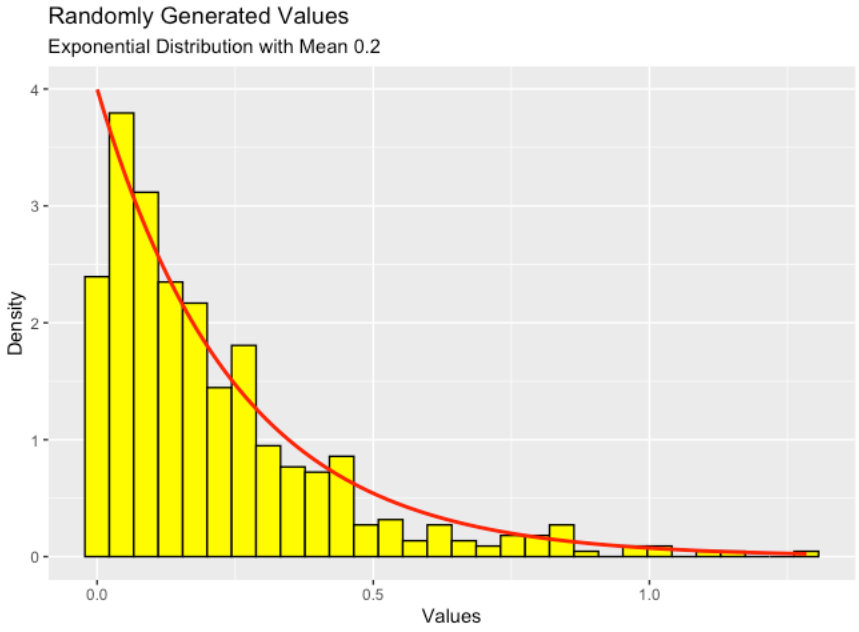
1.
#1
```
set.seed(123)
expo <- rexp(500, rate = 4)

ggplot(data.frame(expo), aes(expo)) +
  geom_histogram(aes(x=expo, y=after_stat(density)),
                 bins=30,
                 fill="mediumpurple",
                 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=4),
                color="blue",
                linewidth=1)
```



2.
#2

```
expplot  <- function(rate, size, line_color, fill_color){
  expo <- rexp(size, rate)
  ggplot(data.frame(expo), aes(expo)) +
```

```
  geom_histogram(aes(x=expo, y=after_stat(density)),
                 bins=30,
                 fill=fill_color,
                 colour="black") +
  labs(x="Values", y="Density") +
  ggtitle(label="Randomly Generated Values",
          subtitle=paste("Exponential Distribution with Mean ", 1 / rate, sep="")) +
  stat_function(fun = function(x) dexp(x, rate=4),
                color=line_color,
                linewidth=1)
}

expplot(rate=5, size=500, line_color="red", fill_color="yellow")
```

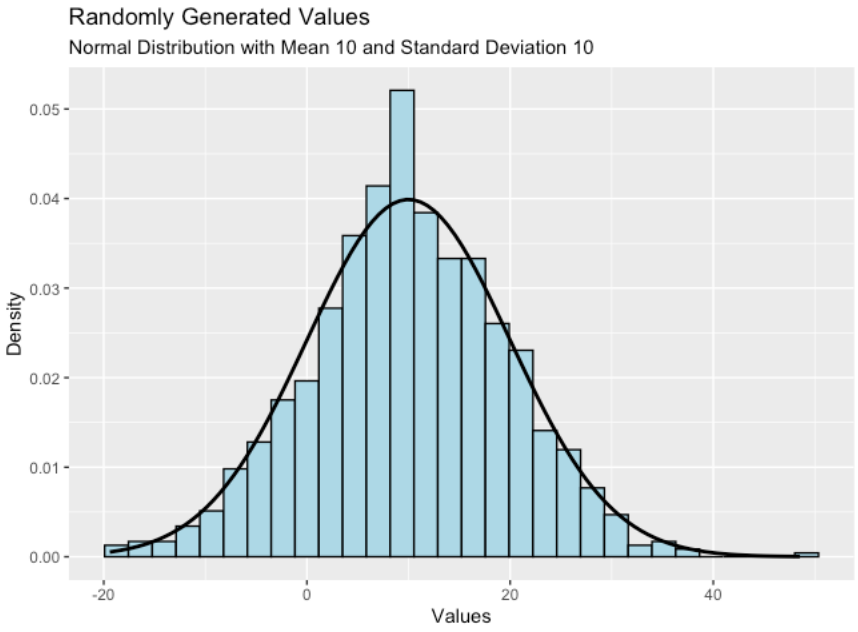


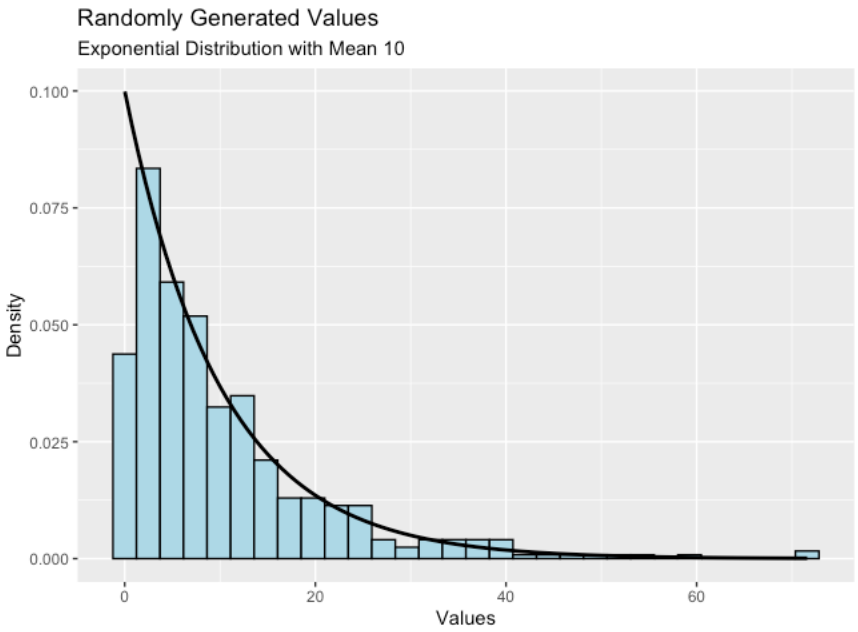
3.

#3

```
norm.or.exp.opt <- function(n,par1,par2=NULL) {
 dist <- sample(c("Exponential", "Normal"), 1, prob = c(0.4, 0.6))
 if(is.null(par2)) {
  par2 = par1
 }

 if(dist == "Exponential") {
  x1 <- rexp(n,1/par1)
```

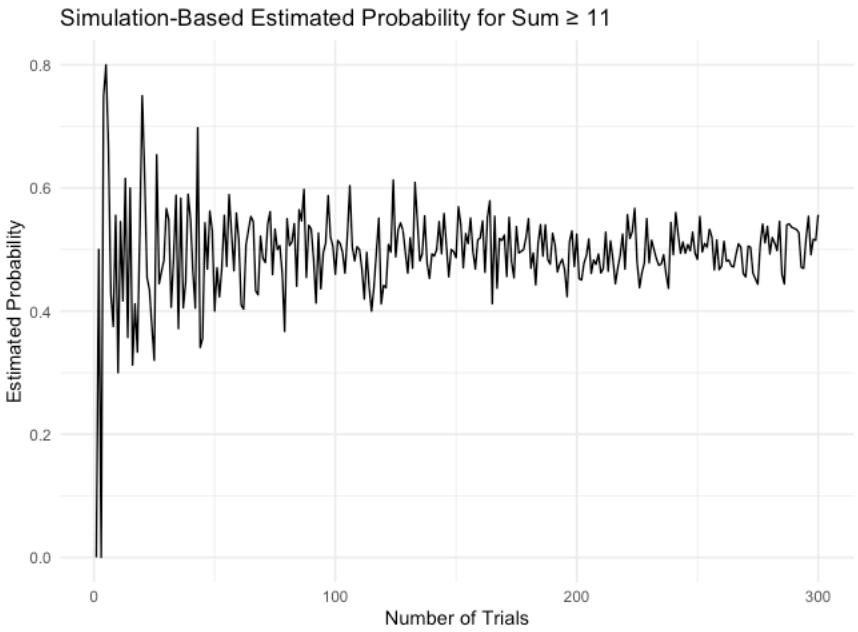
```
  st <- paste("Exponential Distribution with Mean ",par1, sep="")
  fn <- function(x) dexp(x, rate = 1/par1)
 } else if(dist == "Normal") {
  x1 <- rnorm(n,par1,par2)
  st <- paste("Normal Distribution with Mean ", par1,
           " and Standard Deviation ", par2,   sep="")
  fn <- function(x) dnorm(x, mean = par1, sd = par2)
 }
 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)
}

norm.or.exp.opt(1000, 10)

norm.or.exp.opt(500, 10, 4)
```





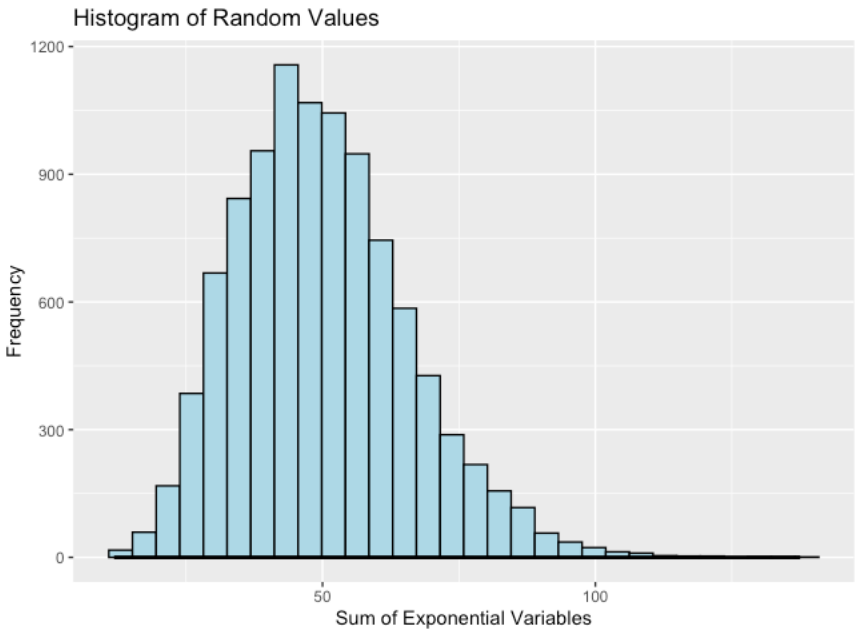
4.
#4

```
HighRoll <- function(numDice, numSides, targetValue, numTrials) {
  apply(matrix(sample(1:numSides, numDice * numTrials, replace = TRUE), nrow = numDice), 2,
sum) >= targetValue
}

plot_simulation_estimate <- function(targetValue, maxTrials) {

  num_trials <- 1:maxTrials
  prob_estimates <- sapply(num_trials, function(trials) mean(HighRoll(3, 6, targetValue, trials)))

  plot_data <- data.frame(Trials = num_trials, Probability = prob_estimates)

  ggplot(plot_data, aes(x = Trials, y = Probability)) +
    geom_line() +
    labs(title = paste("Simulation-Based Estimated Probability for Sum ⩾", targetValue),
         x = "Number of Trials",
         y = "Estimated Probability") +
    theme_minimal()
}

plot_simulation_estimate(targetValue = 11, maxTrials = 300)
```



5.
#5

```
plot_gamma_histogram <- function(reps, N, M) {

  random_values <- replicate(reps, sum(rexp(n = N, rate = M)))

  plot_data <- data.frame(Values = random_values)

  ggplot(plot_data, aes(x = Values)) +
    geom_histogram(bins = 30, fill = "lightblue", color = "black") +
    labs(title = "Histogram of Random Values",
         x = "Sum of Exponential Variables",
         y = "Frequency") +
    stat_function(fun = function(x) dgamma(x, shape = N, scale = 1/M),color="black",
                  linewidth=1)
}

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



6.
#6

```
rate1 <- 0.3
rate2 <- 0.2

num_replicates <- 500000

x1 <- rexp(num_replicates, rate = rate1)
x2 <- rexp(num_replicates, rate = rate2)

min_values <- pmin(x1, x2)

estimated_prob <- mean(min_values < 2)

true_prob <- pexp(2, rate = rate1 + rate2)

estimated_prob
true_prob
```