ARVIE SA1

SINOCRUZ, ARVIE

2025-03-16

Question 1

Note that the "x" is the factory's production and "y" is the defect rates.

```r
# Function to validate inputs
validate_inputs <- function(x, y) {
  # Check constraints for x
  if (sum(x) != 1) {
    return("Error: The sum of x1, x2, and x3 must be 1.")
  }
  if (any(x < 0.10 | x > 0.40)) {
    return("Error: Each x value must be between 0.10 and 0.40.")
  }

  # Checking the constraints for y
  if (sum(y) != 0.12) {
    return("Error: The sum of y1, y2, and y3 must be 0.12.")
  }
  if (any(y < 0.01 | y > 0.05)) {
    return("Error: Each y value must be between 0.01 and 0.05.")
  }

  return(TRUE)

}
```

```r
# Function to calculate probability of selecting a defective product
compute_defective_probability <- function(x, y) {
  probability_defective <- sum(x * y)
  return(probability_defective)
}
```

```r
# Taking user input
cat("Enter values for x1, x2, and x3 (separated by space): ")
```

```
## Enter values for x1, x2, and x3 (separated by space):
```

```r
x <- as.numeric(strsplit(readline(), " ")[[1]])
```

```r
cat("Enter values for y1, y2, and y3 (separated by space): ")
```

```
## Enter values for y1, y2, and y3 (separated by space):
```

```r
y <- as.numeric(strsplit(readline(), " ")[[1]])
```

```r
# Validate input
validation_result <- validate_inputs(x, y)
```

```r
if (validation_result == TRUE) {
  # Compute probability
  defective_prob <- compute_defective_probability(x, y)
  cat(sprintf("The probability of selecting a defective product is: %.4f\n", defective_prob))
} else {
  # Print validation error
  cat(validation_result, "\n")
}
```

```
## Error: The sum of x1, x2, and x3 must be 1.
```

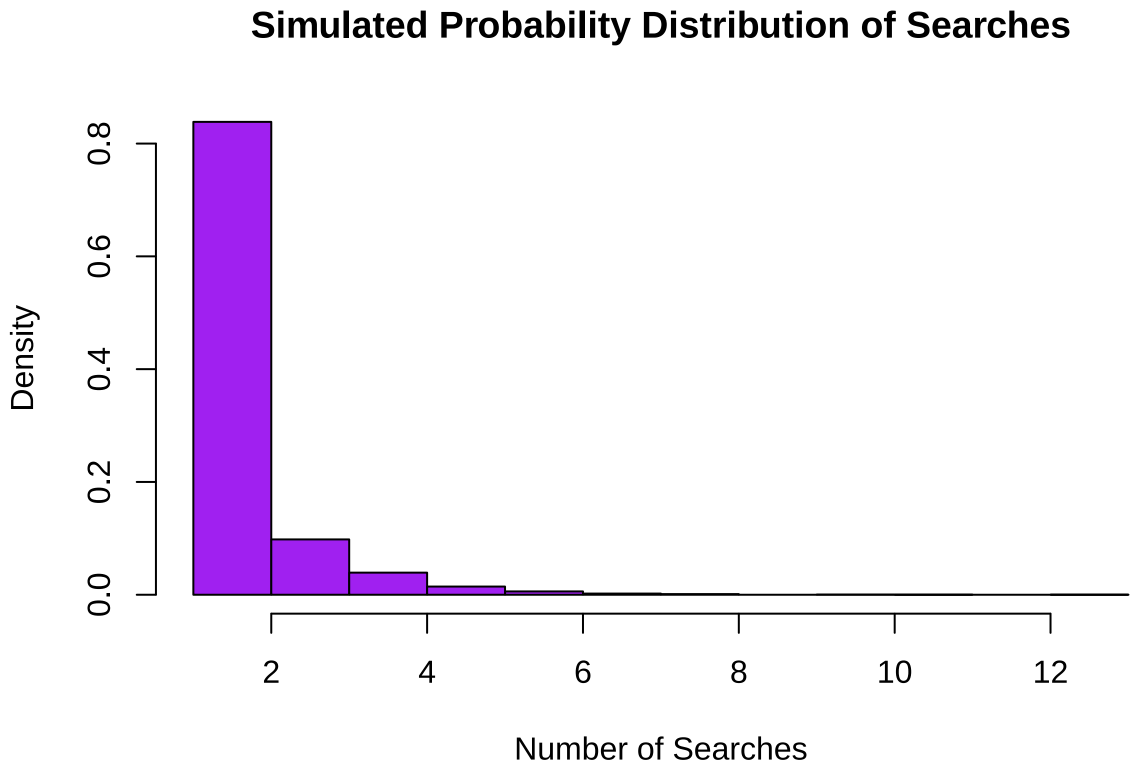
Question 3

```r
set.seed(123)
```

```r
# Defining the probability of success
prob_success <- 0.6
num_simulations <- 10000
```

```r
# Simulating search process
attempts_searches <- rgeom(num_simulations, prob_success) + 1  # Geometric distribution with X = search
```

```r
# Ploting simulated pdf
hist(attempts_searches, probability = TRUE, breaks = max(attempts_searches),
     main = "Simulated Probability Distribution of Searches",
     xlab = "Number of Searches", col = "purple")
```



```r
# Mean and variance
mean_searches <- mean(attempts_searches)
variance_searches <- var(attempts_searches)
cat("Mean:", mean_searches, "\nVariance:", variance_searches, "\n")
```

```
## Mean: 1.6612
## Variance: 1.104725
```

```r
# Conditional distribution: given that 3 searches failed
conditional_searches <- attempts_searches[attempts_searches > 3]
```

```r
# Computing the Mean and variance for the conditional distribution
mean_conditional <- mean(conditional_searches)
variance_conditional <- var(conditional_searches)
cat("Conditional Mean:", mean_conditional, "\nConditional Variance:", variance_conditional, "\n")
```

```
## Conditional Mean: 4.643533
## Conditional Variance: 1.16499
```

Part A

```r
prob_X4_given_X3 <- mean(attempts_searches == 4) / mean(attempts_searches > 3)
prob_X1 <- mean(attempts_searches == 1)
```

```r
cat("P(X=4 | X>3):", prob_X4_given_X3, "\nP(X=1):", prob_X1, "\n")
```

```
## P(X=4 | X>3): 0.6182965
## P(X=1): 0.6045
```