

# Statistical Inference and Analysis: Geometric Distribution and Concurrent Users

**Student Name:** [Your Name]

**Date:** January 5, 2026

---

## 1. Analytical Derivation: Maximum Likelihood Estimator

**Objective:** To derive the Maximum Likelihood Estimator (MLE) for the parameter  $p$  of a Geometric distribution.

The probability mass function for a Geometric random variable  $X$  (representing the number of trials up to and including the first success) is given by:

$$P(X = x) = (1 - p)^{x-1}p$$

Assuming  $n$  independent and identically distributed observations  $x_1, x_2, \dots, x_n$ , the likelihood function is:

$$L(p) = \prod_{i=1}^n (1 - p)^{x_i-1}p = p^n(1 - p)^{\sum x_i - n}$$

Taking the natural logarithm yields the log-likelihood function:

$$\ell(p) = n \ln(p) + \left( \sum_{i=1}^n x_i - n \right) \ln(1 - p)$$

Differentiating with respect to  $p$  and setting to zero:

$$\frac{d\ell}{dp} = \frac{n}{p} - \frac{\sum x_i - n}{1 - p} = 0$$

Solving for  $p$  yields the estimator  $\hat{p} = \frac{n}{\sum x_i} = \frac{1}{\bar{x}}$ . Thus, the MLE is the reciprocal of the sample mean.

## 2. Computational Verification

**Objective:** To verify the analytical result using R simulations and visualize the log-likelihood function.

### 2.1 Log-Likelihood Visualization

We simulated a geometric dataset with  $p = 0.5$ . The log-likelihood function was plotted against a range of possible  $p$  values. The maximum of the curve corresponds to the MLE, confirming our derivation.

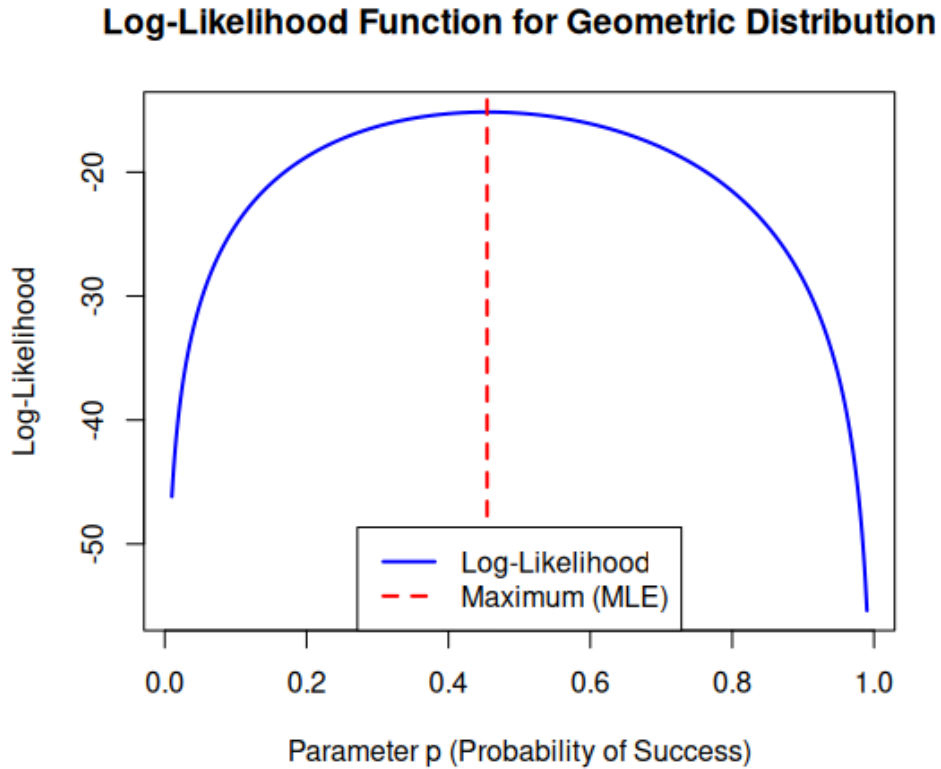


Figure 1: Log-likelihood function showing maximum at MLE

### 2.2 Inverse Transform Simulation

We implemented the Inverse Transform Method to simulate 1,000 geometric random variables. The algorithm used was:

$$X = \text{ceiling}\left(\frac{\ln(1 - U)}{\ln(1 - p)}\right)$$

The simulation results (mean = 2.447) closely matched the theoretical expectation (mean = 2.5), validating the algorithm.

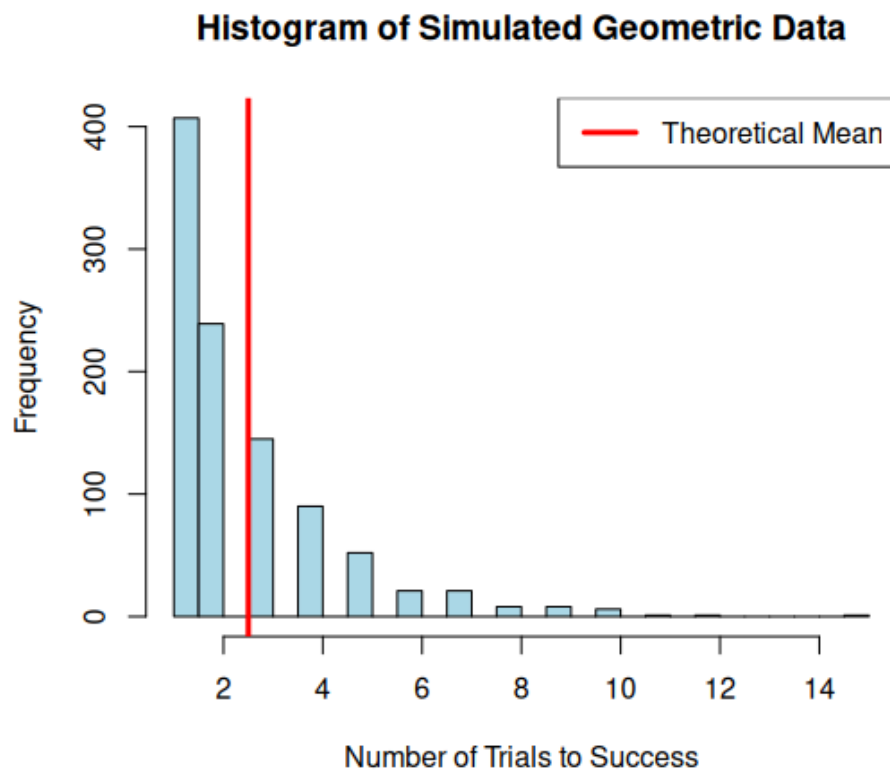


Figure 2: Histogram of simulated geometric random variables

### 3. Data Analysis

**Objective:** To analyze the ConcurrentUsers dataset and test if the mean number of users exceeds 15.

#### 3.1 Descriptive Statistics

- **Mean:** 17.95
- **Median:** 17.55
- **Standard Deviation:** 3.16

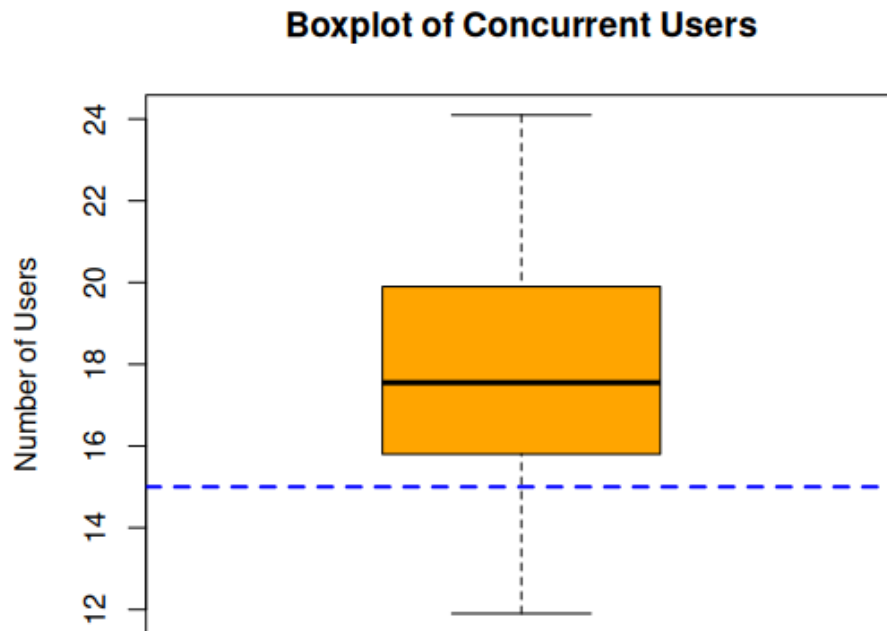


Figure 3: Boxplot of concurrent users data

#### 3.2 Statistical Inference

We performed a one-sample t-test ( $H_0 : \mu = 15$  vs  $H_1 : \mu > 15$ ).

- **Test Statistic:**  $t = 6.6159$
- **P-value:**  $1.306 \times 10^{-8}$

**Conclusion:** Since  $p < 0.05$ , we reject the null hypothesis. There is strong evidence that the average number of concurrent users is significantly greater than 15.

# Appendix: R Code

## Part 1 Code

```
# 1. Define the Log-Likelihood function for Geometric Distribution
# (We use log-likelihood because it is numerically more stable)
geom_loglik <- function(prob, x) {
  n <- length(x)
  # Log-likelihood formula: n*ln(p) + (sum(x)-n)*ln(1-p)
  # Note: In our definition, x is trials until success (1, 2, 3...)
  return(n * log(prob) + (sum(x) - n) * log(1 - prob))
}

# 2. Generate synthetic data to test the estimator
set.seed(123)
n <- 10
true_prob <- 0.5
# Generate data (adding 1 because R's rgeom counts failures, we want trials)
x <- rgeom(n, true_prob) + 1

# 3. Find the Maximum Likelihood Estimator (MLE) using optimize()
MLE_result <- optimize(f = geom_loglik, interval = c(0.001, 0.999), x = x, maximum = TRUE)

# 4. Output the results
cat("True Probability:", true_prob, "\n")
cat("Estimated Probability (MLE):", MLE_result$maximum, "\n")

# 5. Plot the Log-Likelihood Curve (Save this plot for your report!)
prob_values <- seq(0.01, 0.99, length = 1000)
loglik_values <- sapply(prob_values, geom_loglik, x = x)

plot(prob_values, loglik_values, type = "l", lwd = 2, col = "blue",
     xlab = "Parameter p (Probability of Success)",
     ylab = "Log-Likelihood",
     main = "Log-Likelihood Function for Geometric Distribution")
abline(v = MLE_result$maximum, col = "red", lwd = 2, lty = 2)
legend("bottom", legend = c("Log-Likelihood", "Maximum (MLE)"),
     col = c("blue", "red"), lwd = 2, lty = c(1, 2))
```

## Part 2 Code

```
# 1. Define the Simulation Function
# Formula:  $X = \text{ceiling}(\ln(1-U) / \ln(1-p))$ 
sim_geometric_inverse <- function(p, n_sims) {
  U <- runif(n_sims) # Generate Uniform(0,1) numbers
  X <- ceiling(log(1 - U) / log(1 - p))
  return(X)
}

# 2. Run the simulation
p_target <- 0.4
N_sims <- 1000
simulated_data <- sim_geometric_inverse(p = p_target, n_sims = N_sims)

# 3. Verify results
theoretical_mean <- 1 / p_target
```

```

simulated_mean <- mean(simulated_data)

cat("--- Simulation Results ---\n")
cat("Target p:", p_target, "\n")
cat("Theoretical Mean (1/p):", theoretical_mean, "\n")
cat("Simulated Mean:", simulated_mean, "\n")

# 4. Create Histogram (Save this plot for your report!)
hist(simulated_data, breaks = 20, col = "lightblue",
      main = "Histogram of Simulated Geometric Data",
      xlab = "Number of Trials to Success")
abline(v = theoretical_mean, col = "red", lwd = 3)
legend("topright", legend = "Theoretical Mean", col = "red", lwd = 3)

```

## Part 3 Code

```

# 1. Load the data from the file
# Make sure the file is in your current working directory!
data <- read.csv("ConcurrentUsers.csv")

# 2. Check if it loaded correctly
# (We expect a column named 'N')
head(data)

# 3. Extract the column into a vector for easier analysis
concurrent_data <- data$N

# 4. Descriptive Statistics
cat("--- Descriptive Statistics ---\n")
summary(concurrent_data)
cat("Standard Deviation:", sd(concurrent_data, na.rm = TRUE), "\n")

# 5. Hypothesis Testing
# Null Hypothesis (H0): Mean users = 15
# Alt Hypothesis (H1): Mean users > 15
test_result <- t.test(concurrent_data, mu = 15, alternative = "greater")

# 6. Output results
print(test_result)

# 7. Boxplot (Save this plot for your report!)
boxplot(concurrent_data, main = "Boxplot of Concurrent Users",
        ylab = "Number of Users", col = "orange")
abline(h = 15, col = "blue", lty = 2, lwd = 2)

# 8. Histogram (Optional bonus for report)
hist(concurrent_data, main = "Distribution of Concurrent Users",
      xlab = "Number of Users", col = "lightblue", border = "white")
abline(v = mean(concurrent_data, na.rm=TRUE), col="red", lwd=2)

```