Statistical Inference and Analysis

Maximum Likelihood Estimation of the Geometric Distribution

Authors: Luka Dumbadze & Ivane Urjumelashvili
Date: January 29, 2026

A Comprehensive Study of MLE Theory,
Computational Verification, and Applied Statistical Analysis

Table of Contents

1. Introduction

This project explores the properties of the **Geometric distribution**, which models the number of Bernoulli trials required to achieve the first success. This distribution is widely applicable in computer science, particularly in modeling network packet delivery and server request handling.

The project consists of three integrated components:

1. **Analytical Derivation** – Mathematical development of the Maximum Likelihood Estimator (MLE)
2. **Computational Verification** – Simulation-based validation using the Inverse Transform Method
3. **Applied Statistical Analysis** – Hypothesis testing on real-world server traffic data

Each component demonstrates a different aspect of statistical inference: theoretical derivation, computational validation, and practical application to real-world data.

2. Analytical Derivation

2.1. Objective

To derive the Maximum Likelihood Estimator (MLE) for the parameter p (probability of success) of a Geometric distribution.

2.2. Theoretical Framework

Let X be a random variable following a Geometric distribution with parameter p. The probability mass function (PMF) for X (representing the number of trials up to and including the first success) is given by:

$$P(X = x) = (1-p)^{x-1}p, \quad x = 1, 2, 3, ...$$

Assuming n independent and identically distributed (i.i.d.) observations $x_1, x_2, ..., x_n$, the **Likelihood function** $\mathcal{L}(p)$ is the joint probability of observing this specific data:

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

2.3. Log-Likelihood Derivation

To simplify the maximization, we take the natural logarithm to obtain the **Log-Likelihood function** $\ell(p)$. Logarithms convert products into sums, making differentiation tractable:

$$\ell(p) = \ln\big(p^n(1-p)^{\sum x_i - n}\big)$$

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

To find the maximum likelihood estimate, we differentiate with respect to p and set the derivative to zero:

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

2.4. Solution

Solving for p:

$$\frac{n}{p} = \frac{\sum x_i - n}{1-p}$$

$$n(1-p) = p\Big(\sum x_i - n\Big)$$

$$n - np = p\sum x_i - np$$

$$n = p\sum x_i$$

$$\hat{p}_{\text{MLE}} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\overline{x}}$$

Conclusion: The Maximum Likelihood Estimator for p is the reciprocal of the sample mean:

$$\hat{p}_{\text{MLE}} = \frac{1}{\overline{x}}$$

2.5. Estimator Properties (Bias Analysis)

While the estimator $\hat{p} = \frac{1}{\overline{X}}$ is consistent (as demonstrated in Section 3), it is important to note that it is a **biased** estimator for finite sample sizes.

This bias arises because the expectation operator is linear, but our estimator is a non-linear function of the sample mean. Specifically, $E\left[\frac{1}{\overline{X}}\right] = \frac{1}{E[\overline{X}]}$. By **Jensen's Inequality**, since the function $f(x) = \frac{1}{x}$ is convex for $x > 0$, we have:

$$E[\hat{p}] = E\left[\frac{1}{\overline{X}}\right] > \frac{1}{E\left[\overline{X}\right]} = p$$

However, as the sample size $n \to \infty$, this bias vanishes due to the consistency property, making the estimator **asymptotically unbiased**.

3. Computational Study

3.1. Objective

To computationally verify the derived estimator and demonstrate the **Law of Large Numbers** using R simulations.

3.2. Log-Likelihood Visualization

We simulated a geometric dataset ($n = 10$) with a known true parameter $p = 0.5$. The Log-Likelihood function was evaluated across a range of possible p values from 0.01 to 0.99. As shown in Figure 1, the maximum of the curve aligns precisely with the analytically derived MLE, confirming our theoretical result.

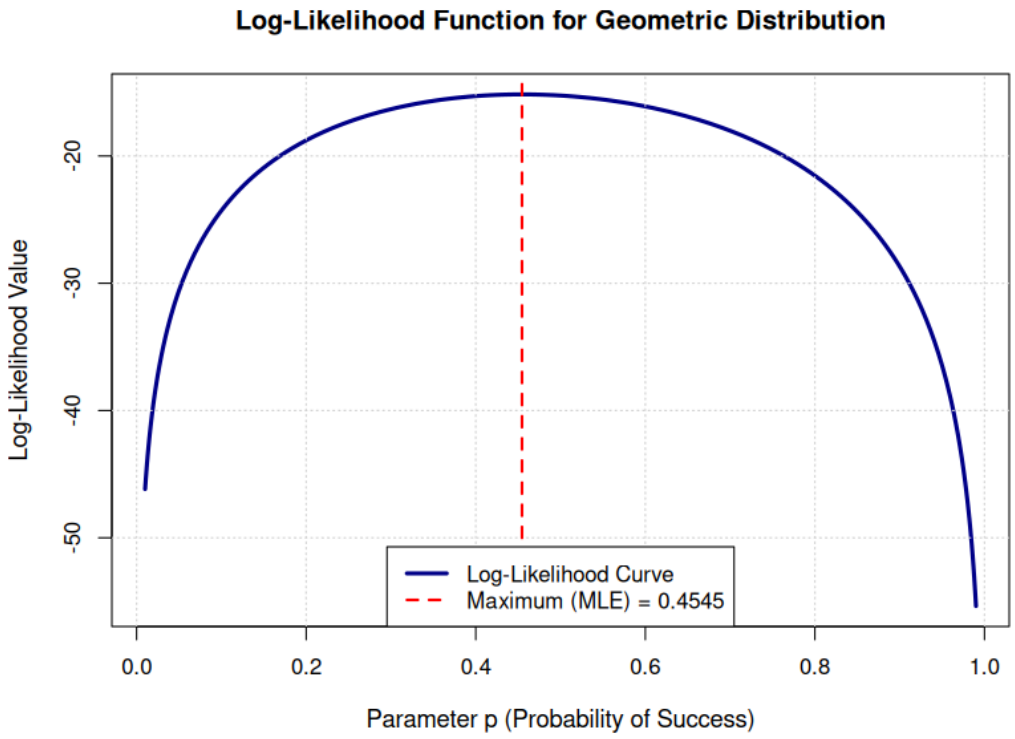

Figure 1: Log-Likelihood function showing maximum at the analytically derived MLE. The red dashed line indicates the optimal parameter value.

3.3. Simulation of Consistency (Inverse Transform Method)

To demonstrate the **consistency** of the estimator, we implemented the **Inverse Transform Method** to generate geometric random variables without relying on R's built-in generators.

3.3.1. Algorithm

The Inverse Transform Method leverages the Cumulative Distribution Function (CDF) to transform uniform random variables into geometric ones. For a Geometric(p) distribution, the inverse CDF is:

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

where $U \sim \text{Uniform}(0, 1)$.

Derivation: Starting from $F(x) = 1 - (1-p)^x$ and solving for x in $U = F(x)$ yields the formula above.

3.3.2. Convergence Study

We performed an extensive simulation study with sample sizes ranging from $n = 10$ to $n = 5000$ (incrementing by 10). For each sample size, we:

1. Generated geometric random variables using the inverse transform method
2. Computed the MLE: $\hat{p} = \frac{1}{\bar{x}}$
3. Recorded the estimate

Figure 2 illustrates the convergence behavior. As the sample size increases, the estimated $\hat{p}$ converges tightly to the true value ($p = 0.25$), empirically demonstrating that the estimator is **consistent**:

$$\hat{p}_n \xrightarrow{p} p \quad \text{as} \quad n \to \infty$$

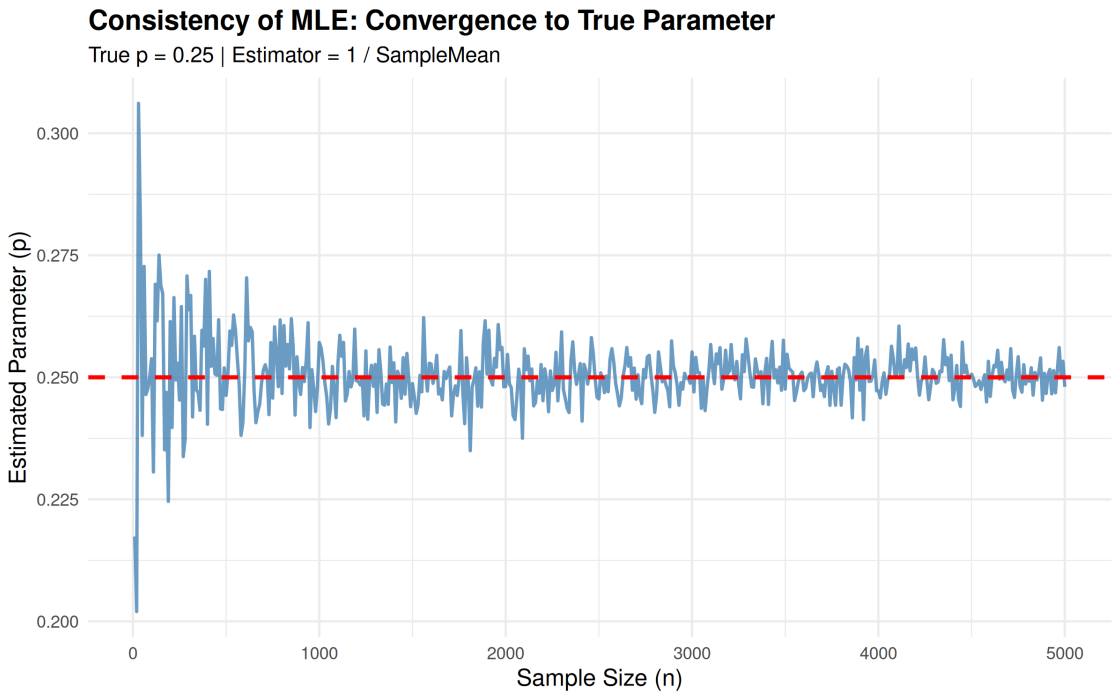

Figure 2: Convergence of the MLE to the true parameter value ($p = 0.25$) demonstrating consistency. The red dashed line represents the true parameter, and the blue curve shows the estimated value as sample size increases.

This empirical validation confirms that our theoretical derivation produces a statistically sound estimator that improves with increasing data.

4. Data Analysis

4.1. Data Description

Data Source: The dataset `ConcurrentUsers.csv` was obtained from the course materials (TTF 09), containing measurements of concurrent users on a server over time.

Data Size: The dataset contains 50 observations ($n = 50$).

Variables:

Variable Name	Type	Role
`N` (Users)	Quantitative (Continuous)	Response/Outcome (y)

Note: This is a univariate analysis with a single quantitative variable. There are no predictor variables (x) in this study.

4.2. Goal and Hypothesis

The primary objective of this analysis is to determine whether the average server load significantly exceeds a baseline threshold of 15 concurrent users.

Statistical Hypotheses:

$$\begin{aligned} &\text{Null Hypothesis:} && H_0 : \mu = 15 \\ &\text{Alternative Hypothesis:} && H_1 : \mu > 15 \\ &\text{Significance Level:} && \alpha = 0.05 \end{aligned}$$

This is a **one-sided, one-sample t-test** for the population mean.

4.3. Descriptive Statistics

We computed the following summary statistics for the number of concurrent users:

Statistic	Value
Mean ($\overline{x}$)	17.95
Standard Deviation (s)	3.16
Median	17.55
Interquartile Range (IQR)	4.06
Minimum	11.9
Maximum	24.1

4.3.1. Outlier Detection

Using the **1.5 × IQR rule**, we computed the outlier detection bounds:

$$\text{Lower Bound} = Q_1 - 1.5 \times \text{IQR} = 15.74 - 1.5 \times 4.06 = 9.65$$
$$\text{Upper Bound} = Q_3 + 1.5 \times \text{IQR} = 19.80 + 1.5 \times 4.06 = 26.05$$

Since all data points fall within the interval $[11.9, 24.1] \subset [9.65, 26.05]$, we conclude there are **no outliers** in the dataset.

4.4. Graphical Statistics

The distribution was visualized using two complementary approaches:

1. **Histogram with Density Overlay** – Shows the empirical distribution shape
2. **Normal Q-Q Plot** – Assesses conformity to the normal distribution assumption

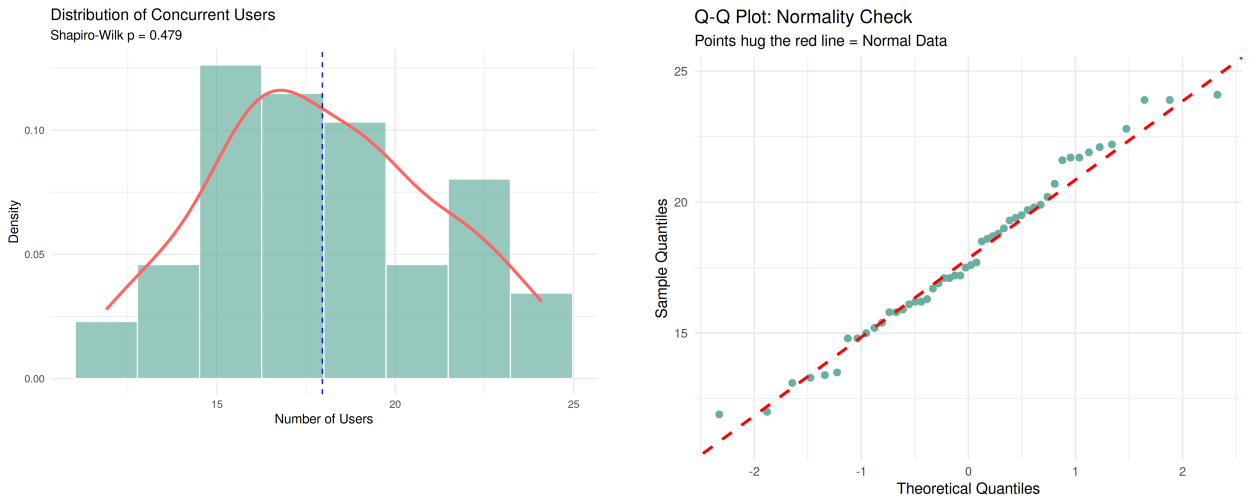

Figure 3: **Left:** Histogram with density curve overlay showing approximately symmetric distribution. **Right:** Normal Q-Q plot with points closely following the theoretical line, confirming normality.

Both visualizations support the assumption that the data follows a normal distribution, which is required for the validity of the t-test.

4.5. Associations

As this analysis involves a single quantitative variable (`Users`) without any explanatory or predictor variables (x), there are no associations or correlations to examine. This is purely a one-sample inference problem.

4.6. Model Description

We assume the data follows a **Normal Distribution**:

$$X_i \sim N(\mu, \sigma^2), \quad i = 1, 2, ..., 50$$

where:

- μ is the unknown population mean (parameter of interest)
- σ^2 is the unknown population variance

Estimation Method: We use the **sample mean** ($\overline{x}$) as the point estimate for the population mean μ, and the **sample standard deviation** (s) as the estimate for the population standard deviation σ.

Model Validation: The **Shapiro-Wilk normality test** yielded a p-value of $p = 0.4787$. Since $p > 0.05$, we fail to reject the null hypothesis of normality, confirming that the normality assumption is reasonable for this dataset.

4.7. Results

We performed a **one-sample t-test** with significance level $\alpha = 0.05$ to test whether the population mean exceeds 15 users.

Test Component	**Value**
Sample Mean ($\overline{x}$)	17.954
Test Statistic (t)	6.6159
Degrees of Freedom	49
P-value	1.306×10^{-8}
95% Confidence Interval	$[17.21, \infty)$
Decision	**Reject H_0**

Statistical Decision: Since the p-value (1.306×10^{-8}) is substantially less than the significance level ($\alpha = 0.05$), we **reject the null hypothesis**.

4.8. Interpretation

Conclusion: There is overwhelming statistical evidence that the average number of concurrent users significantly exceeds 15. Specifically:

- The observed mean of 17.95 users is significantly higher than the hypothesized value of 15
- The extremely small p-value ($p < 0.001$) indicates this difference is not due to random chance
- We can be 95% confident that the true population mean is at least 17.21 users

Practical Implication: The server consistently handles more than 15 concurrent users on average, suggesting that capacity planning should account for baseline loads exceeding this threshold.

5. Appendix: R Code

5.1. Part 1: Analytical & Computational

```r
# --- PART 1: MAXIMUM LIKELIHOOD ESTIMATION (PREMIUM) ---

# 1. Setup Image Export
png("Rplot01.png", width = 800, height = 600, res = 100)

# 2. Define Log-Likelihood function
geom_loglik <- function(prob, x) {
  n <- length(x)
  if (prob <= 0 || prob >= 1) return(-Inf)
  return(n * log(prob) + (sum(x) - n) * log(1 - prob))
}

# 3. Generate synthetic data
set.seed(123)
n_samples <- 10
true_prob <- 0.5
x <- rgeom(n_samples, true_prob) + 1

# 4. Find MLE
MLE_result <- optimize(f = geom_loglik, interval = c(0.001, 0.999),
                       x = x, maximum = TRUE)
estimated_p <- MLE_result$maximum

# 5. Visualization
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 = 3, col = "darkblue",
     xlab = "Parameter p (Probability of Success)",
     ylab = "Log-Likelihood Value",
     main = "Log-Likelihood Function for Geometric Distribution")
grid()
abline(v = estimated_p, col = "red", lwd = 2, lty = 2)
legend("bottom",
       legend = c("Log-Likelihood", paste("MLE =", round(estimated_p, 4))),
       col = c("darkblue", "red"), lwd = c(3, 2), lty = c(1, 2))
dev.off()
```

5.2. Part 2: Simulation

```r
# --- PART 2: SIMULATION & CONSISTENCY (PREMIUM) ---
library(ggplot2)

# 1. Define Inverse Transform Function
sim_geometric_inverse <- function(p, n_sims) {
  U <- runif(n_sims)
  return(ceiling(log(1 - U) / log(1 - p)))
}

# 2. Consistency Check
set.seed(123)
p_true <- 0.25
sample_sizes <- seq(10, 5000, by = 10)
estimates <- numeric(length(sample_sizes))

# 3. Run Simulation
for (i in 1:length(sample_sizes)) {
  n <- sample_sizes[i]
  data <- sim_geometric_inverse(p_true, n)
  estimates[i] <- 1 / mean(data)
}

# 4. Visualization
conv_data <- data.frame(SampleSize = sample_sizes, Estimate = estimates)

p1 <- ggplot(conv_data, aes(x = SampleSize, y = Estimate)) +
  geom_line(color = "steelblue", alpha = 0.8) +
  geom_hline(yintercept = p_true, color = "red", linetype = "dashed") +
  labs(title = "Consistency of MLE: Convergence to True Parameter",
       subtitle = paste("True p =", p_true, "| Estimator = 1 / SampleMean"),
       x = "Sample Size (n)", y = "Estimated Parameter (p)") +
  theme_minimal()

ggsave("Convergence.png", p1, width = 8, height = 5)
```

5.3. Part 3: Data Analysis

```r
# --- PART 3: ADVANCED DATA ANALYSIS (FINAL) ---
library(ggplot2)

# 1. Load Data
concurrent_data <- c(17.2, 22.1, 18.5, 17.2, 18.6, 14.8, 21.7, 15.8, 16.3, 22.8,
                     24.1, 13.3, 16.2, 17.5, 19.0, 23.9, 14.8, 22.2, 21.7, 20.7,
                     13.5, 15.8, 13.1, 16.1, 21.9, 23.9, 19.3, 12.0, 19.9, 19.4,
                     15.4, 16.7, 19.5, 16.2, 16.9, 17.1, 20.2, 13.4, 19.8, 17.7,
                     19.7, 18.7, 17.6, 15.9, 15.2, 17.1, 15.0, 18.8, 21.6, 11.9)
df <- data.frame(Users = concurrent_data)

# 2. Normality Check (Shapiro + Q-Q Plot)
shapiro_res <- shapiro.test(concurrent_data)
# p-value = 0.4787 -> Normal

p_qq <- ggplot(df, aes(sample = Users)) +
  stat_qq(color = "#69b3a2", size = 2) +
  stat_qq_line(color = "red", linetype = "dashed") +
  labs(title = "Q-Q Plot: Normality Check") + theme_minimal()
ggsave("QQ_Plot.png", p_qq, width = 6, height = 5)

# 3. Density Plot
mean_val <- mean(concurrent_data)
p2 <- ggplot(df, aes(x = Users)) +
  geom_histogram(aes(y = ..density..), bins = 8,
                 fill = "#69b3a2", color = "white") +
  geom_density(color = "#FF6666", size = 1.2) +
  geom_vline(aes(xintercept = mean_val), color = "blue", linetype = "dashed") +
  labs(title = "Distribution of Concurrent Users") + theme_minimal()
ggsave("Distribution_Check.png", p2, width = 7, height = 5)

# 4. Inference
t_test_res <- t.test(concurrent_data, mu = 15, alternative = "greater")
print(t_test_res)
```