Statistical Inference in a Cocaine Drug Bust - Scenerio 1

David Fakolujo, Akinyemi Apampa, Ravin Jayasuriya, Prince Oloma and Joshua Ogunbo

2025-02-12

A cocaine bust in Calgary yielded $N_1 = 496$ suspected cocaine plastic packets. To convict the suspected drug traffickers, the **Alberta Crown Prosecution Service** (ACPS) and the **Calgary Police Service** (CPS) had to prove that there was genuine cocaine in (at least one of) the packets. Apparently, drug traffickers have been mixing "clean" packets (i.e., packets that are negative for cocaine, often containing corn starch) with "dirty" ones (i.e., packets that are positive for cocaine) to confound the police.

The assigned ACPS **crown prosecutor** and the CPS **detectives** decided to have $n_1 = 2$ packets analyzed, such that the packets are selected randomly without replacement from the $N_1 = 496$ packets, an unknown number $\theta \leq N_1$ of which are suspected to be dirty, and the remaining $N_1 - \theta = 496 - \theta \geq 0$ packets are clean.

The crown prosecutor and the detectives agreed to have only $n_1 = 2$ packets analyzed because the test is quite expensive and they did not have the budget to test all $N_1 = 496$ packets. In addition, it is not necessary to test all of them, as one dirty packet is enough for conviction. After conviction, however, there is still the need to estimate θ, as its value might be needed by the judge to decide what sentence to impose on the convicted drug trafficker.

(a) Sampling Distribution of X_1

Let the discrete random variable X_1 represent the number of dirty packets out of the n_1 packets selected. The **sampling distribution** of X_1 is given by the probabilities:

$$P_\theta(X_1 = x_1) = P_\theta\left(x_1 \text{ from } n_1 \text{ packets are dirty}\right)$$

$$= \frac{\binom{\theta}{x_1}\binom{N_1-\theta}{n_1-x_1}}{\binom{N_1}{n_1}}, \quad \forall x_1 \in \mathcal{X}_1(\theta),$$

where $\mathcal{X}_1(\theta)$ is the **set of possible values** of X_1. Use $P_\theta(X_1 = x_1)$ to give the elements of $\mathcal{X}_1(\theta)$.

Note that the possible values of X_1 depend on θ, whence, $\mathcal{X}_1(\theta)$ likewise depends on θ.

It follows that X_1 has a **hypergeometric distribution**, with parameters $N_1 = 496$, θ, and $n_1 = 2$.

Although the above probabilities have closed-form expressions involving θ, observe that they cannot be calculated because θ is unknown.

After first estimating θ, the probabilities can then be estimated by the **plug-in method**.

Solution

The probability mass function (PMF) for the number of dirty packets X_1 out of the $n_1 = 2$ selected packets, given a total of $N_1 = 496$ packets with θ dirty packets, follows a **hypergeometric distribution**:

$$P_\theta(X_1 = x_1) = \frac{\binom{\theta}{x_1}\binom{N_1-\theta}{n_1-x_1}}{\binom{N_1}{n_1}}$$

where:

- $N_1 = 496$ is the total number of packets,
- $n_1 = 2$ is the sample size,
- θ is the unknown number of dirty packets,
- x_1 represents the observed number of dirty packets,
- $N_1 - \theta$ is the number of clean packets in the population,
- $n_1 - x_1$ is the number of clean packets from the sample.

Possible Values of X_1

Since we are selecting **two packets**, the number of dirty packets observed in the sample can range from **0 to** $\min(n_1, \theta)$:

$$\mathcal{X}_1(\theta) = \{0, 1, 2\} \quad \text{(assuming } \theta \geq 2\text{)}$$

If $\theta < 2$, then the possible values of X_1 are:

- $\{0, 1\}$ if $\theta = 1$,
- $\{0\}$ if $\theta = 0$ (i.e., no dirty packets exist at all).

Thus, the **set of possible values of X_1 depends on θ**.

Since θ is **unknown**, we **cannot compute** the exact probabilities using the formula above. Instead, we must first estimate θ, typically using observed data from a sample.

Once we have an estimate $\hat{\theta}$, we substitute it into the probability function:

$$\hat{P}(X_1 = x_1) = \frac{\binom{\hat{\theta}}{x_1}\binom{N_1-\hat{\theta}}{n_1-x_1}}{\binom{N_1}{n_1}}$$

(b). Maximum Likelihood Estimation (MLE) of θ

Given the observed value x_1 of X_1, an **intuitive approach** to estimating θ is to maximize with respect to θ over all possible values of θ – the probability $P_\theta(X_1 = x_1)$ of observing the value x_1 of X_1 that you actually observed. That is, you estimate θ by that value $\hat{\theta}_1$ of θ among all its possible values that makes what you actually observed the most likely to be observed:

$$\hat{\theta}_1 = \underset{\theta \in \Theta(x_1)}{\arg\max} P_\theta(X_1 = x_1).$$

where $\Theta(x_1)$ is the **parameter space** of θ containing all the possible values of θ. We refer to $\hat{\theta}_1$ as the **maximum likelihood estimate (MLE)** of θ based on the **observed data** x_1.

Note that $\Theta(x_1)$, the **parameter space** of θ, depends on the observed value x_1 of X_1. This is so since $\mathcal{X}_1(\theta)$, the set of possible values of X_1, depends on θ, so that when $X_1 = x_1$ is observed, the value of θ will, in turn, depend on x_1.

A direct and easy approach to maximization of $P_\theta(X_1 = 0)$ is then to plot it as a function of $\theta \in \Theta(x_1)$, and locate the value of θ at which the maximum occurs. The usual **calculus approach** of taking the derivative of $P_\theta(X_1 = x_1)$ with respect to θ does not apply since $P_\theta(X_1 = x_1)$ is **not continuous**, whence, is **not differentiable**.

Now, suppose the $n_1 = 2$ randomly selected packets turned out to be clean (i.e., $x_1 = 0$). Obtain the MLE $\hat{\theta}_1$ of θ, given that you observed no dirty packets among the $n_1 = 2$ selected.

Solution

To estimate θ, we use the **maximum likelihood estimate (MLE)**, which maximizes the probability of observing $X_1 = x_1$ over the possible values of θ. The MLE is defined as:

$$\hat{\theta}_1 = \underset{\theta \in \Theta(x_1)}{\arg\max} P_\theta(X_1 = x_1)$$

where $\Theta(x_1)$ represents the **set of feasible values** for θ.

Since X_1 follows a **hypergeometric distribution**, the probability mass function (PMF) is:

$$P_\theta(X_1 = x_1) = \frac{\binom{\theta}{x_1}\binom{N_1-\theta}{n_1-x_1}}{\binom{N_1}{n_1}}$$

Why We Cannot Use Calculus

- Normally, to find the maximum, we would take the derivative of $P_\theta(X_1 = x_1)$ with respect to θ and solve for θ.
- However, **this function is not continuous** because θ takes discrete values (θ represents the count of dirty packets).
- Since the function is **not differentiable**, we instead **plot** $P_\theta(X_1 = x_1)$ for all possible values of θ and identify the maximum.

Case 1: Estimating θ When $X_1 = 1$

If we observe **one dirty packet** in our sample of two, the MLE should provide the most likely estimate for the total number of dirty packets in the population.

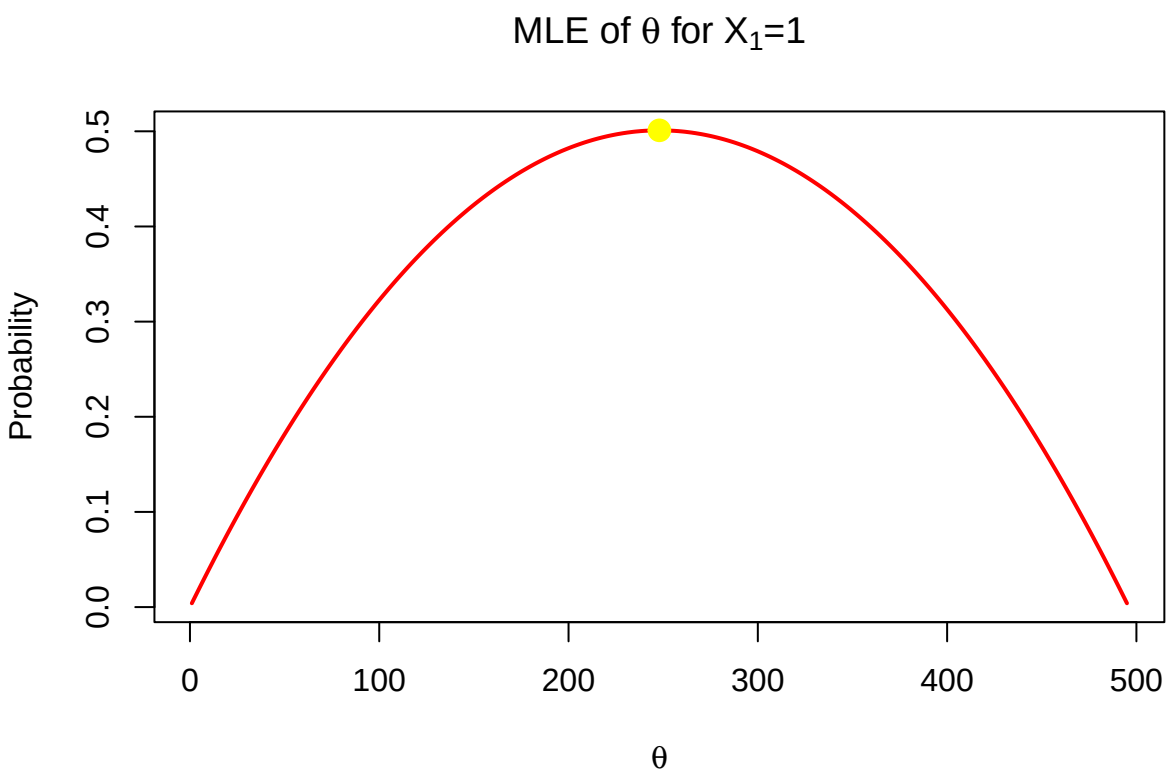
Constraints on θ From the PMF structure, we establish the following constraints:
1. The number of dirty packets must be at least the number observed:

$$x_1 \leq \theta \quad \Rightarrow \quad \theta \geq 1.$$

2. The number of dirty packets cannot exceed the number of packets available:

$$\theta \leq N_1 - n_1 + x_1 \quad \Rightarrow \quad \theta \leq 495.$$

```r
N_1 <- 496  # Total packets
n_1 <- 2    # Sample size
x_1 <- 1    # Observed dirty packets

possible_theta <- 1:495

probability <- dhyper(x_1, possible_theta, N_1 - possible_theta, n_1)

theta_max <- possible_theta[which.max(probability)]
max_prob <- max(probability)

plot(
  possible_theta,
  probability,
  type = "l",
  col = "red",
  lwd = 2,
  xlab = expression(theta),
  ylab = "Probability",
  main = expression(paste("MLE of ", theta, " for ", X[1], "=1"))
)

points(theta_max, max_prob, col = "yellow", pch = 19, cex = 1.5)
```




```
print(paste("The maximum likelihood estimate for theta is:", theta_max))
```

```
## [1] "The maximum likelihood estimate for theta is: 248"
```

```
print(paste("The maximum probability is:", max_prob))
```

```
## [1] "The maximum probability is: 0.501010101010101"
```

From the output, the **Maximum Likelihood Estimate (MLE)** indicates that the most likely number of dirty packets in the population is $\theta = 248$.

This means that, given a sample of **2 packets**, where **1 was found to be dirty**, the best estimate for the total number of dirty packets in the entire population is **248**.

Additionally, the **likelihood** of observing exactly **1 dirty packet** in the sample, assuming that $\theta = 248$, is approximately **50.1%**.

Case 2: Estimating θ When $X_1 = 0$

If both packets selected in our sample are clean ($x_1 = 0$), the **Maximum Likelihood Estimate (MLE)** should provide the most likely estimate for the total number of dirty packets in the population.

Constraints on θ

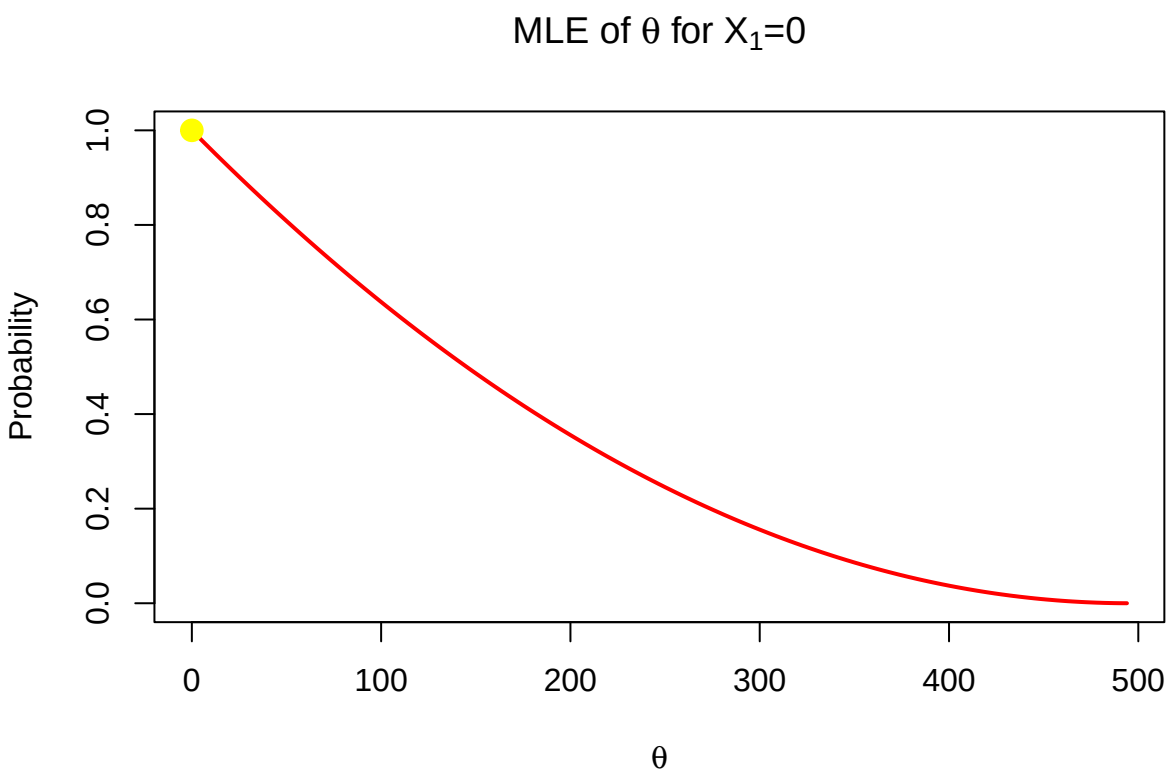
1. Since we observed no dirty packets, the **smallest possible** θ is **0**:

$$x_1 \leq \theta \quad \Rightarrow \quad \theta \geq 0.$$

2. The **largest possible** θ is still constrained by the total number of packets:

$$\theta \leq N_1 - n_1 + x_1 \quad \Rightarrow \quad \theta \leq 494.$$

```r
N_1 <- 496  # Total packets
n_1 <- 2    # Sample size
x_1 <- 0    # Observed dirty packets

possible_theta <- 0:494

probability <- dhyper(x_1, possible_theta, N_1 - possible_theta, n_1)

theta_max <- possible_theta[which.max(probability)]
max_prob <- max(probability)

# Plot likelihood function
plot(
  possible_theta,
  probability,
  type = "l",
  col = "red",
  lwd = 2,
  xlab = expression(theta),
  ylab = "Probability",
  main = expression(paste("MLE of ", theta, " for ", X[1], "=0"))
)

points(theta_max, max_prob, col = "yellow", pch = 19, cex = 1.5)
```




```
print(paste("The maximum likelihood estimate for theta is:", theta_max))
```

```
## [1] "The maximum likelihood estimate for theta is: 0"
```

```
print(paste("The maximum probability is:", max_prob))
```

```
## [1] "The maximum probability is: 1"
```

When no dirty packets ($x_1 = 0$) are observed in the sample, the **Maximum Likelihood Estimate (MLE)** suggests that the most likely value of θ (the total number of dirty packets in the population) is **0**.

This conclusion makes intuitive sense if none of the selected packets tested positive for cocaine, the best estimate is that there are **no dirty packets** in the entire population. In other words, based on the sample data, it is most probable that all packets in the population are clean.

Mathematically, this means that the **likelihood function** reaches its highest value when $\theta = 0$, reinforcing the idea that the absence of dirty packets in the sample strongly suggests their absence in the entire population.

Case 3: Estimating θ When $X_1 = 2$

If both packets selected in our sample are dirty ($x_1 = 2$), the **Maximum Likelihood Estimate (MLE)** should provide the most likely estimate for the total number of dirty packets in the population.

Constraints on θ

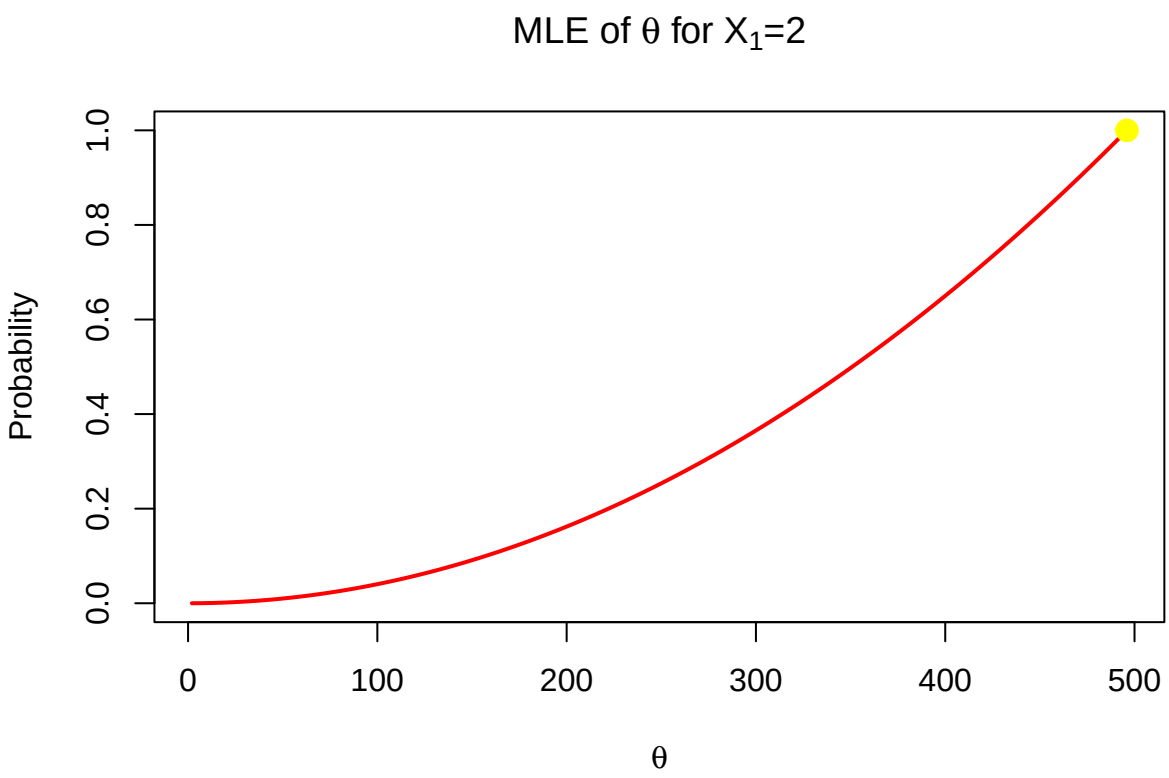
1. Since we observed two dirty packets, the **smallest possible** θ is **2**:

$$x_1 \leq \theta \quad \Rightarrow \quad \theta \geq 2.$$

2. The **largest possible** θ is still constrained by the total number of packets:

$$\theta \leq N_1 - n_1 + x_1 \quad \Rightarrow \quad \theta \leq 496.$$

```r
N_1 <- 496  # Total packets
n_1 <- 2    # Sample size
x_1 <- 2    # Observed dirty packets

possible_theta <- 2:496

probability <- dhyper(x_1, possible_theta, N_1 - possible_theta, n_1)

theta_max <- possible_theta[which.max(probability)]
max_prob <- max(probability)

# Plot likelihood function
plot(
  possible_theta,
  probability,
  type = "l",
  col = "red",
  lwd = 2,
  xlab = expression(theta),
  ylab = "Probability",
  main = expression(paste("MLE of ", theta, " for ", X[1], "=2"))
)

points(theta_max, max_prob, col = "yellow", pch = 19, cex = 1.5)
```




```
print(paste("The maximum likelihood estimate for theta is:", theta_max))
```

```
## [1] "The maximum likelihood estimate for theta is: 496"
```

```
print(paste("The maximum probability is:", max_prob))
```

```
## [1] "The maximum probability is: 1"
```

When two dirty packets ($x_1 = 2$) are observed in the sample, the **Maximum Likelihood Estimate (MLE)** suggests that the most likely value of θ (the total number of dirty packets in the population) is **496**.

This conclusion makes intuitive sense if two of the selected packets tested positive for cocaine, the best estimate is that there are **496 dirty packets** in the entire population. In other words, based on the sample data, it is most probable that all packets in the population are dirty

Mathematically, this means that the **likelihood function** reaches its highest value when $\theta = 496$, reinforcing the idea that the presence of dirty packets in the sample strongly suggests their prsence in the entire population.

(c). Joint Sampling Distribution of X_1 and X_2

After the $n_1 = 2$ packets initially analyzed turned out to be clean, the **crown prosecutor** and the **detectives** quickly decided to have $n_2 = 4$ **more packets** analyzed. As before, these packets were randomly chosen

without replacement from the remaining $N_2 = 496 - 2 = 494$ packets hoping to find a dirty packet and thus avoid having the **criminal charges dismissed** by the judge for **lack of or insufficient evidence**.

Let the discrete **random variable** X_2 represent the **number of dirty packets** out of the additional $n_2 = 4$ packets.

Joint Sampling Distribution of X_1 and X_2

We define the **joint probabilities** of X_1 and X_2:

$$P_\theta(X_1 = x_1, X_2 = x_2) = P_\theta\left(x_1 \text{ from } n_1 \text{ packets and } x_2 \text{ from } n_2 \text{ packets are dirty}\right),$$

where $x_1 \in \mathcal{X}_1(\theta)$ and $x_2 \in \mathcal{X}_2(\theta)$. Here, $\mathcal{X}_2(\theta)$ represents the **set of possible values of** X_2.

Dependency Between X_1 and X_2

It is important to note that X_1 **and** X_2 **are dependent and non-identically distributed random variables (RVs)**, since the value of X_2 depends on the observed value of X_1.

Additionally, the **joint probabilities** $P_\theta(X_1 = x_1, X_2 = x_2)$, like $P_\theta(X_1 = x_1)$ from part (a), have **closed-form expressions** that depend on θ. Thus, we can express:

$$P_\theta(X_1 = x_1, X_2 = x_2) = P_\theta(X_2 = x_2 | X_1 = x_1) P_\theta(X_1 = x_1).$$

Using the fact that:

- $X_1 \sim \text{Hypergeometric}(N_1 = 496, \theta, n_1 = 2)$,
- $X_2 | X_1 = x_1 \sim \text{Hypergeometric}(N_2 = N_1 - n_1, \theta - x_1, n_2 = 4)$,

Use the above to specify the possible values of X_2 in $\mathcal{X}_2(\theta)$ in terms of θ, similar to how the possible values of X_1 in $\mathcal{X}_1(\theta)$ were obtained in (b).

Answer

After the initial test of $n_1 = 2$ packets resulted in $X_1 = 0$ dirty packets, additional $n_2 = 4$ packets were tested to gather sufficient evidence. Let X_2 represent the number of dirty packets in this second sample.

Joint Probability of X_1 and X_2

The joint probability function is given by:

$$P_\theta(X_1 = x_1, X_2 = x_2) = P_\theta(X_2 = x_2 | X_1 = x_1) P_\theta(X_1 = x_1)$$

Given that:

- $X_1 \sim \text{Hypergeometric}(N_1 = 496, \theta, n_1 = 2)$,
- $X_2 | X_1 = x_1 \sim \text{Hypergeometric}(N_2 = 496 - 2, \theta - x_1, n_2 = 4)$,

we define the possible values of θ.

Defining the Parameter Space of θ

To ensure valid probability computations, we must correctly define the possible values of θ. The parameter space is:

$$\theta \geq X_1 + X_2 = 0 + 2 = 2$$

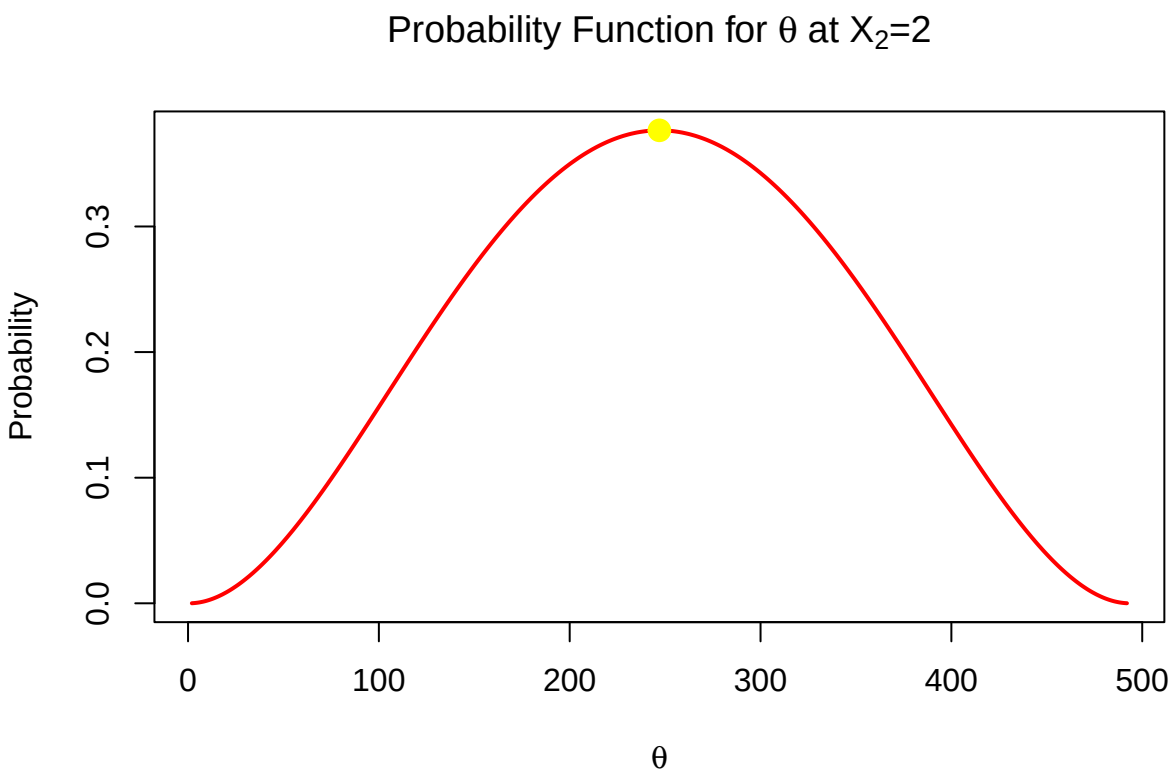
$$\theta \leq N_1 - n_1 = 496 - 2 = 494$$

Thus, the valid possible values of θ are:

$$\Theta(0, 2) = \{2, 3, \ldots, 494\}$$

Computing Joint Sampling Distribution

```r
# Observing 2 dirty packets out of 4
N_2 <- 494
n_2 <- 4
x_2 <- 2

# The possible theta values
possible_theta_2 <- 2:492

# Calculating the hypergeometric probability
probability_2 <- dhyper(x_2, possible_theta_2, N_2 - possible_theta_2, n_2)

# Finding the max values for theta and the max probability
theta_max_2 <- possible_theta_2[which.max(probability_2)]
max_prob_2 <- max(probability_2)


plot(
  possible_theta_2,
  probability_2,
  type = "l",
  col = "red",
  lwd = 2,
  xlab = expression(theta),
  ylab = "Probability",
  main = expression(paste("Probability Function for ", theta, " at ", X[2], "=2"))
)

points(theta_max_2, max_prob_2, col = "yellow", pch = 19, cex = 1.5)
```




```
max_theta = min(max(possible_theta), max(possible_theta_2))
min_theta = max(min(possible_theta), min(possible_theta_2))

print(paste("The Maximum Likelihood Estimate for theta is:", theta_max_2))
```

```
## [1] "The Maximum Likelihood Estimate for theta is: 247"
```

```
print(paste("The maximum Probability is:", max_prob_2))
```

```
## [1] "The maximum Probability is: 0.376525945724873"
```

```
print(
  paste(
    "The smallest possible value of theta for the joint sampling distribution is:",
    min_theta
  )
)
```

```
## [1] "The smallest possible value of theta for the joint sampling distribution is: 2"
```

```
print(
  paste(
    "The largest possible value of theta for the joint sampling distribution is:",
```

```
    max_theta
  )
)
```

```
## [1] "The largest possible value of theta for the joint sampling distribution is: 492"
```

(d). Estimating θ Based on Joint Probability

Suppose $x_2 = 2$ of the $n_2 = 4$ additional packets were found to be dirty. This makes the accused's **conviction for drug trafficking** a certainty; however, there is still the **unknown value of θ to estimate**.

To estimate θ, the **joint probability** $P_\theta(X_1 = x_1, X_2 = x_2)$, evaluated at the observed values x_1 and x_2 of X_1 and X_2, is maximized as a function of θ over its **parameter space** $\Theta(x_1, x_2)$.

Observe that $\Theta(x_1, x_2)$ depends on x_1 and x_2, since $\mathcal{X}_1(\theta)$ and $\mathcal{X}_2(\theta)$ both depend on θ.

To maximize $P_\theta(X_1 = x_1, X_2 = x_2)$, one needs to evaluate it at the possible values of θ in $\Theta(x_1, x_2)$ and select the value at which the maximum occurs.

That value, $\hat{\theta}_2$, is the **Maximum Likelihood Estimate (MLE)** of θ based on x_1 and x_2.

Given the observed data $x_1 = 0$ and $x_2 = 2$, obtain the **MLE $\hat{\theta}_2$ of** θ by maximizing the joint probability $P_\theta(X_1 = 0, X_2 = 2)$ with respect to θ over its **parameter space** $\Theta(0, 2)$.

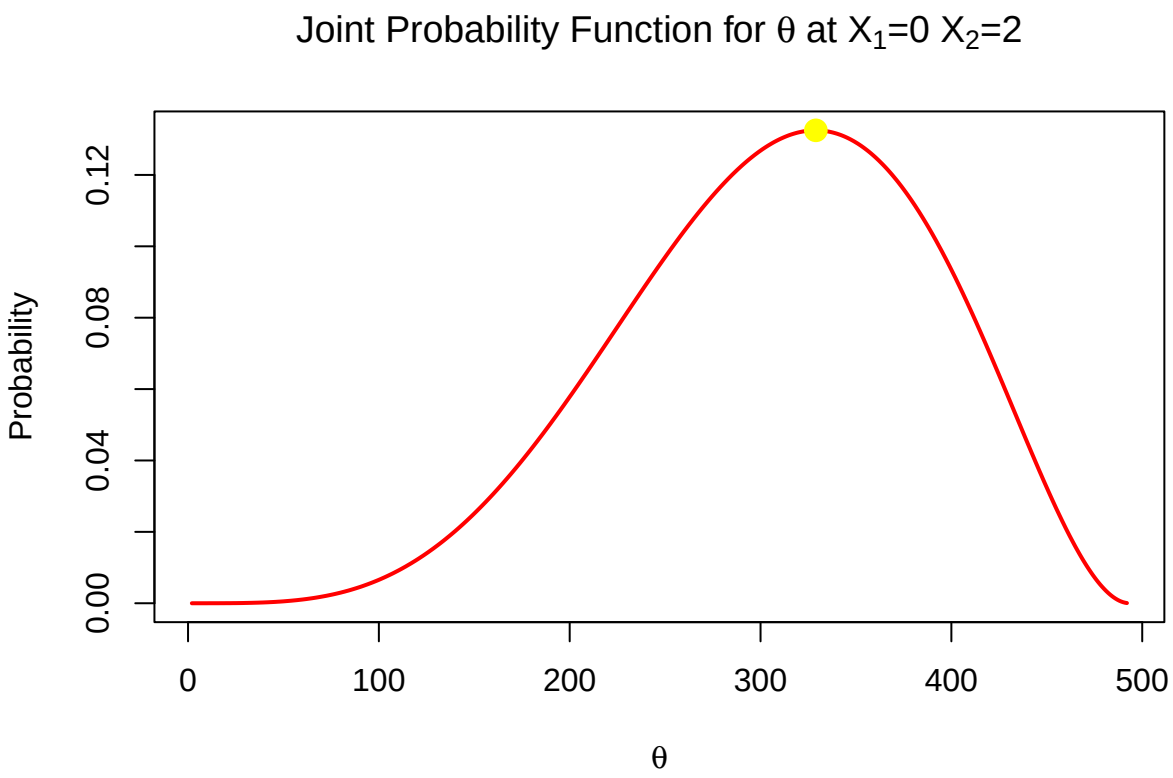
Answer

With $x_2 = 2$ dirty packets found in the second test, the conviction is certain, but the actual number of dirty packets remains unknown. To estimate θ, we maximize the joint probability function:

$$\hat{\theta}_2 = \underset{\theta \in \Theta(0,2)}{\arg\max} P_\theta(X_1 = 0, X_2 = 2)$$

The new parameter space $\Theta(0, 2)$ depends on the observed values of X_1 and X_2.

```
probability_3 = probability[3:493] * probability_2[1:491]

theta_max_3 <- possible_theta_2[which.max(probability_3)]
max_prob_3 <- max(probability_3)


plot(
  possible_theta_2,
  probability_3,
  type = "l",
  col = "red",
  lwd = 2,
  xlab = expression(theta),
  ylab = "Probability",
  main = expression(paste("Joint Probability Function for ", theta, " at ",
                          X[1], "=0", " ", X[2], "=2"))
)

points(theta_max_3, max_prob_3, col = "yellow", pch = 19, cex = 1.5)
```




```
print(paste("The Maximum Likelihood Estimate for theta (joint) is:", theta_max_3))
```

```
## [1] "The Maximum Likelihood Estimate for theta (joint) is: 329"
```

```
print(paste("The maximum Probability for joint distribution is:", max_prob_3))
```

```
## [1] "The maximum Probability for joint distribution is: 0.1324900852217"
```

(e) Estimating θ Using a Sequential Sampling Approach

Since 1 dirty packet is sufficient for conviction, a more streamlined approach to estimating θ randomly selects a packet from $N_1 = 496$ packets without replacement, have it analyzed, and if it is dirty, then the 1 required dirty packet is found, and sampling is ended; however, if the selected packet is clean, the random selection is continued without replacement, until a dirty packet is selected. Define the RV Y as **the number of packets selected at random without replacement until the first dirty packet is selected**. Thus, $Y = 1$, if the first selection resulted in a dirty packet, and $Y = 3$, if the first dirty packet was selected at the third selection. Define the RV U_i as follows:

$$U_i = \begin{cases} 1, & \text{if selection } i \text{ yields a dirty packet,} \\ 0, & \text{if selection } i \text{ yields a clean packet,} \end{cases}$$

for $i = 1, 2, ...,$ where $U_1, U_2, ..., U_{N_1-\theta+1}$ are **dependent and non-identically distributed Bernoulli RVs**, such that

$$U_1 \sim b\left(1, P_\theta(U_1 = 0) = \frac{\theta}{N_1}\right),$$

$$U_2|U_1 = 0 \sim b\left(1, P_\theta(U_2 = 1|U_1 = 0) = \frac{\theta}{N_1 - 1}\right),$$

$$U_3|U_2 = 0 \sim b\left(1, P_\theta(U_3 = 1|U_2 = 0) = \frac{\theta}{N_1 - 2}\right),$$

$$\vdots$$

$$U_{N_1-\theta}|U_{N_1-\theta-1} = 0 \sim b\left(1, P_\theta(U_{N_1-\theta} = 1|U_{N_1-\theta-1} = 0) = \frac{\theta}{\theta - 1}\right),$$

$$U_{N_1-\theta+1}|U_{N_1-\theta} = 0 \sim b\left(1, P_\theta(U_{N_1-\theta+1} = 1|U_{N_1-\theta} = 0) = 1\right),$$

where $b(1, p)$ denotes the **Bernoulli distribution**, with success rate p. It then follows that

$$P_\theta(Y = y) = P_\theta\left(y\text{th packet selected is the first dirty packet selected}\right),$$

$$= P_\theta(U_1 = 0)P_\theta(U_y = 1|U_{y-1} = 0)\prod_{i=2}^{y-2} P_\theta(U_i = 0|U_{i-1} = 0),$$

$$= \left(\frac{\theta}{N_1 - y + 1}\right)\prod_{j=1}^{y-2}\left(1 - \frac{\theta}{N_1 - j}\right), \quad \forall y \in \mathcal{Y}(\theta),$$

where $\mathcal{Y}(\theta) = \{1, 2, \ldots, N_1 - \theta + 1\}$ is **the set of possible values of** Y. The RV Y is said to have a **negative hypergeometric distribution**, with parameters $N_1 = 496$, $K = 1$, and θ. An alternative expression for $P_\theta(Y = y)$ is given by

$$P_\theta(Y = y) = \frac{\binom{N_1 - y}{\theta - 1}}{\binom{N_1}{\theta}}, \quad \forall y \in \mathcal{Y}(\theta).$$

Note that the set $\mathcal{Y}(\theta)$ of possible values of Y depends on θ, like those for X_1 and X_2 in (a) and (c).

Suppose you repeatedly selected a packet at random without replacement from the $N_1 = 496$ packets until you selected the first dirty packet, and suppose you selected the first dirty packet at the yth selection. Based on the **observed data** y, the **MLE** $\hat{\theta}_3$ is obtained by maximizing the probability $P_\theta(Y = y)$ with respect to θ over its parameter space $\Theta[y] = \{1, 2, \ldots, \min(N_1, N_1 - y + 1)\}$, which ostensibly depends on the observed data y. However, note that $y \geq 1$, so that $N_1 \leq N_1 - y + 1$ and $\min(N_1, N_1 - y + 1) = N_1$, $\forall y \in \mathcal{Y}(\theta)$. Thus, we can ignore its dependence on y, and denote $\Theta[y]$ simply by Θ.

You will next carry out a **Monte Carlo simulation** to compare the MLEs $\hat{\theta}_1$ and $\hat{\theta}_3$ in terms of their **bias and RMSE** as estimates of θ. Here, $\hat{\theta}_1$ is the MLE of θ in (b) obtained from a **sample of size** n (not necessarily $n = 2$). Note that $\hat{\theta}_1$ is a **fixed-n estimate**, for which the **sample size** n is **fixed in advance** and the **number of dirty packets** X is the random quantity. In contrast, the **MLE** $\hat{\theta}_3$ is a **sequential-n estimate**, for which the **sample size** n is **not fixed beforehand** (i.e., n is a RV) and instead, it is the **number of dirty packets that need to be observed** that is fixed (at 1, in the case of Y above). Write a **short R script** to implement the following **Monte Carlo simulation**:

- **S1**. Fix `set.seed(0212.2025)`. With $N_1 = 496$, fix $\theta_0 = 331$, the **true value of** θ. Fix as well the **sample size** $n \in \{4, 5, 6\}$, for **MLE** $\hat{\theta}_1$. Fix the parameter $K = 1$, where K is the number of dirty packets that need to be observed before sampling from the negative hypergeometric distribution of Y is ended.

- **S2**. Generate a **Monte Carlo sample of size** n from the **hypergeometric sampling distribution** of X using the R function `rhyper()` to obtain the observed value x of X for the sample. Based on x, obtain the MLE $\hat{\theta}_1$ of θ by evaluating

$$P_\theta(X = x) = \frac{\binom{\theta}{x}\binom{N_1-\theta}{n-x}}{\binom{N_1}{n}}$$

at each possible value of θ in $\Theta(x)$, and picking that value $\hat{\theta}_1$ at which the maximum occurs as the MLE of θ. Note that x can be 0, like in (b).

- **S3**. Simulate the observed value y of Y from its **negative hypergeometric distribution**, with parameters $N_1 = 496$, $K = 1$, and $\theta_0 = 331$, the true value of θ.
  You need to install the R package **extraDistr** so that you can use the function **rnhyper()** to simulate the value of Y.

Given the observed simulated value y of Y, use function **dnhyper()** of package **extraDistr** to evaluate $P_\theta(Y = y)$ at each possible value of θ in its parameter space Θ – given above – and obtain its **MLE** $\hat{\theta}_3$ as that value at which the maximum occurs.

- **S4**. Repeat **S2**–**S3** $R = 1000$ times to get R values

$$\hat{\theta}_1^{(1)}, \ldots, \hat{\theta}_1^{(R)}$$

and

$$\hat{\theta}_3^{(1)}, \ldots, \hat{\theta}_3^{(R)},$$

of the **MLEs** $\hat{\theta}_1$ and $\hat{\theta}_3$ based on the values $x^{(r)}$ and $y^{(r)}$ of X and Y, respectively, in Monte Carlo sample $r = 1, \ldots, R$.

You can now calculate the **empirical bias**

$$\text{Bias}_{MC}(\hat{\theta}_1) = \text{Ave}_{MC}(\hat{\theta}_1) - \theta$$

and **empirical RMSE**

$$\text{RMSE}(\hat{\theta}_1) = \sqrt{\text{Var}_{MC}(\hat{\theta}_1) + \text{Bias}^2_{MC}(\hat{\theta}_1)}$$

of $\hat{\theta}_1$, for example, where

$$\text{Ave}_{MC}(\hat{\theta}_1) = \frac{1}{R}\sum_{r=1}^{R} \hat{\theta}_1^{(r)} \quad \to_{n\to+\infty} \quad E(\hat{\theta}_1),$$

$$\text{Var}_{MC}(\hat{\theta}_1) = \frac{1}{R-1}\sum_{r=1}^{R}\left(\hat{\theta}_1^{(r)} - \text{Ave}_{MC}(\hat{\theta}_1)\right)^2 \quad \to_{n\to+\infty} \quad \text{Var}(\hat{\theta}_1),$$

are the **Monte Carlo mean** and **Monte Carlo variance** of $\hat{\theta}_1$.

Similarly, calculate

$$\text{Bias}_{MC}(\hat{\theta}_3) = \text{Ave}_{MC}(\hat{\theta}_3) - \theta$$

and

$$\text{RMSE}(\hat{\theta}_3) = \sqrt{\text{Var}_{MC}(\hat{\theta}_3) + \text{Bias}^2_{MC}(\hat{\theta}_3)}$$

of $\hat{\theta}_3$.

Present and summarize the **empirical bias** and **empirical RMSE** of the **2 MLEs** $\hat{\theta}_1$ and $\hat{\theta}_3$ in a **table and/or plot** and **comment briefly**. Which between the **2 estimators of** θ is better?

Answer

```r
set.seed(0212.2025)


N_1 <- 496
true_thetas <- c(331, 150, 450, 60) # Different true values of theta to test
sample_sizes <- c(4, 5, 6) # Sample sizes to test for MLE1
K <- 1
R <- 1000
```

```r
# Function to compute MLE for hypergeometric distribution with different sample sizes
compute_MLE1 <- function(true_theta, n) {
  X <- rhyper(R, true_theta, N_1 - true_theta, n)  # Generate R samples
  MLE_X <- numeric(R)

  for (i in 1:R) {
    possible_theta <- 0:N_1
    probability <- dhyper(X[i], possible_theta, N_1 - possible_theta, n)
    MLE_X[i] <- possible_theta[which.max(probability)]
  }

  return(MLE_X)
}

# Generate MLE1 samples for different values of true_theta and sample sizes
MLE1_samples <- list()
for (n in sample_sizes) {
  MLE1_samples[[as.character(n)]] <- lapply(true_thetas, compute_MLE1, n = n)
}
```

```r
# Load required library
library(extraDistr)

# Function to compute MLE for negative hypergeometric distribution
compute_MLE3 <- function(true_theta) {
  Y <- rnhyper(R, N_1 - true_theta, true_theta, 1)  # Generate R samples
  MLE_Y <- numeric(R)

  for (i in 1:R) {
    possible_theta <- 1:(N_1 - 1)
    probability <- dnhyper(Y[i], N_1 - possible_theta, possible_theta, 1)
    MLE_Y[i] <- possible_theta[which.max(probability)]
  }

  return(MLE_Y)
}

# Generate MLE3 samples for different values of true_theta
MLE3_samples <- lapply(true_thetas, compute_MLE3)
```

```r
# Function to compute Bias and RMSE
compute_bias_rmse <- function(MLE_samples, true_theta) {
  bias <- mean(MLE_samples) - true_theta
  rmse <- sqrt(var(MLE_samples) + (bias ** 2))
  return(c(bias, rmse))
}

# Compute Bias and RMSE for MLE1 across all sample sizes
MLE1_results <- list()
for (n in sample_sizes) {
  MLE1_results[[as.character(n)]] <- t(
    mapply(compute_bias_rmse, MLE1_samples[[as.character(n)]], true_thetas)
  )
}

# Compute Bias and RMSE for MLE3
MLE3_results <- t(mapply(compute_bias_rmse, MLE3_samples, true_thetas))

# Assign column names
colnames(MLE3_results) <- c("Bias_MLE3", "RMSE_MLE3")
```

```r
# Combine results into a single dataframe for MLE1
results_list <- list()
for (n in sample_sizes) {
  results_list[[as.character(n)]] <- data.frame(
    True_Theta = true_thetas,
    Bias_MLE1 = MLE1_results[[as.character(n)]][, 1],
    RMSE_MLE1 = MLE1_results[[as.character(n)]][, 2]
  )
}

# Convert MLE3 results into a dataframe
MLE3_results_df <- data.frame(
```

```
  True_Theta = true_thetas,
  Bias_MLE3 = MLE3_results[, 1],
  RMSE_MLE3 = MLE3_results[, 2]
)

# Print results for all sample sizes of MLE1
for (n in sample_sizes) {
  cat("\n### Results for MLE1 (n =", n, ")\n")
  print(results_list[[as.character(n)]])
}
```

```
##
## ### Results for MLE1 (n = 4 )
##   True_Theta Bias_MLE1 RMSE_MLE1
## 1        331    -2.028 116.70132
## 2        150     3.388 115.28941
## 3        450     0.120  72.66912
## 4         60     2.124  82.32076
##
## ### Results for MLE1 (n = 5 )
##   True_Theta Bias_MLE1 RMSE_MLE1
## 1        331    -0.230 102.85937
## 2        150     2.332  97.98896
## 3        450     1.442  64.65933
## 4         60    -0.985  71.25089
##
## ### Results for MLE1 (n = 6 )
##   True_Theta Bias_MLE1 RMSE_MLE1
## 1        331    -3.146  95.09074
## 2        150     1.834  93.20826
## 3        450    -1.590  57.93451
## 4         60     0.714  67.73287
```

```
# Print results for MLE3
cat("\n### Results for MLE3 (Sequential-n) \n")
```

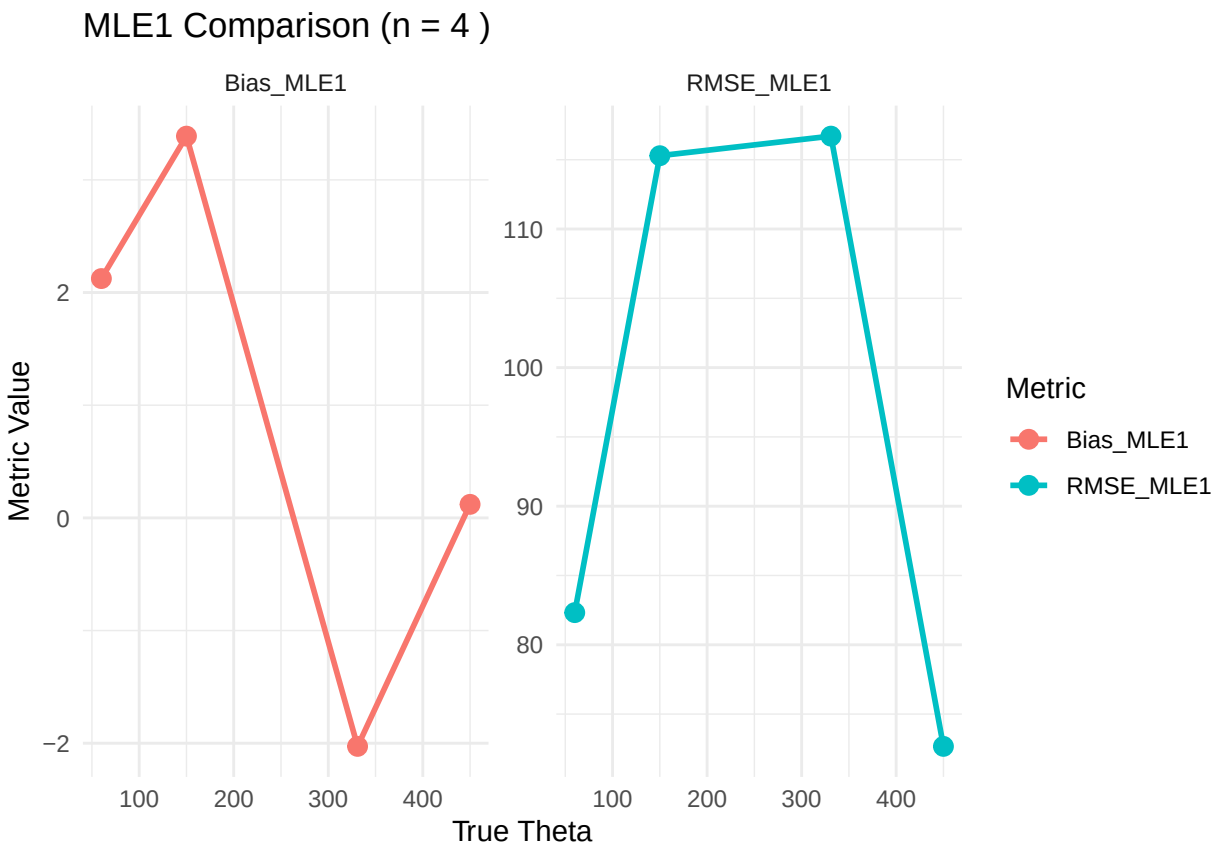
```
##
## ### Results for MLE3 (Sequential-n)
```

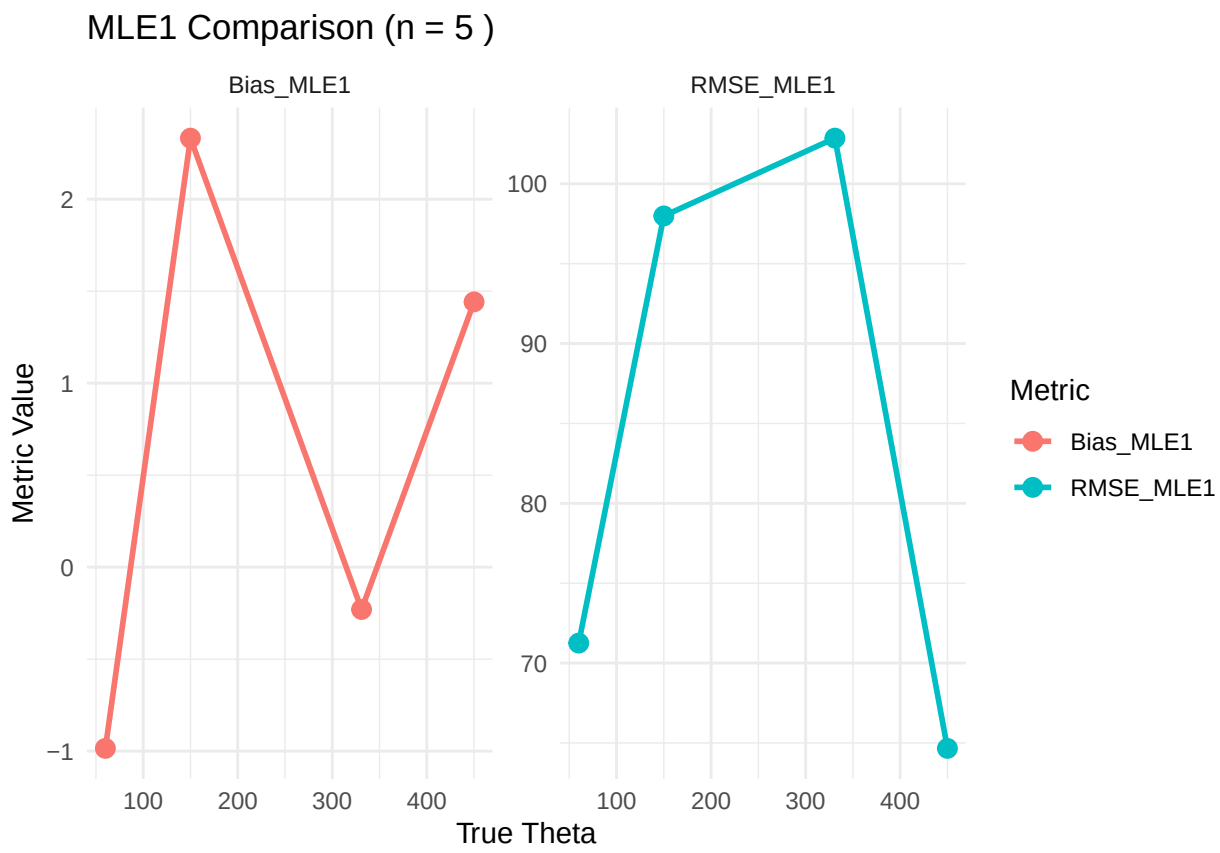
```
print(MLE3_results_df)
```

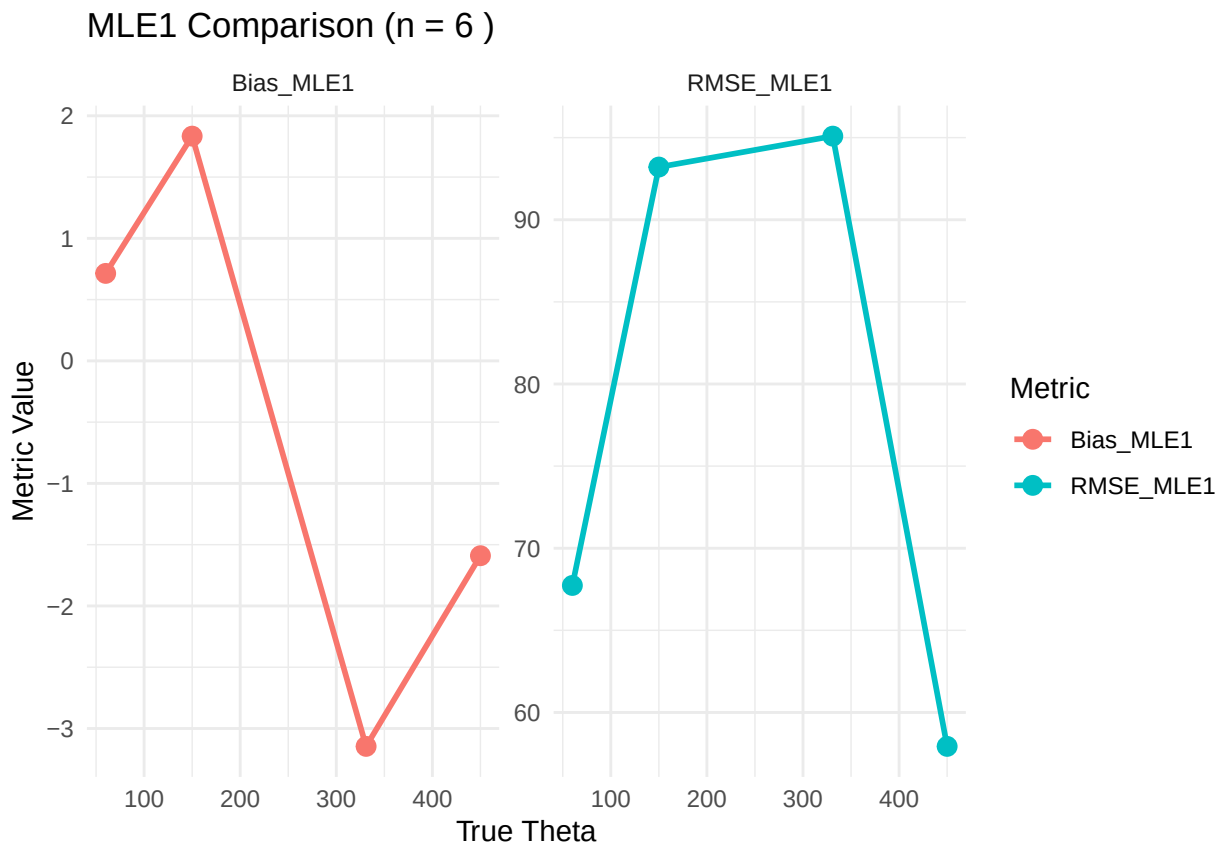
```
##   True_Theta Bias_MLE3 RMSE_MLE3
## 1        331    66.639 152.88621
## 2        150   104.743 195.86648
## 3        450    20.705  78.22343
## 4         60    81.181 163.69836
```

```
# Load required libraries
library(ggplot2)
library(reshape2)
```

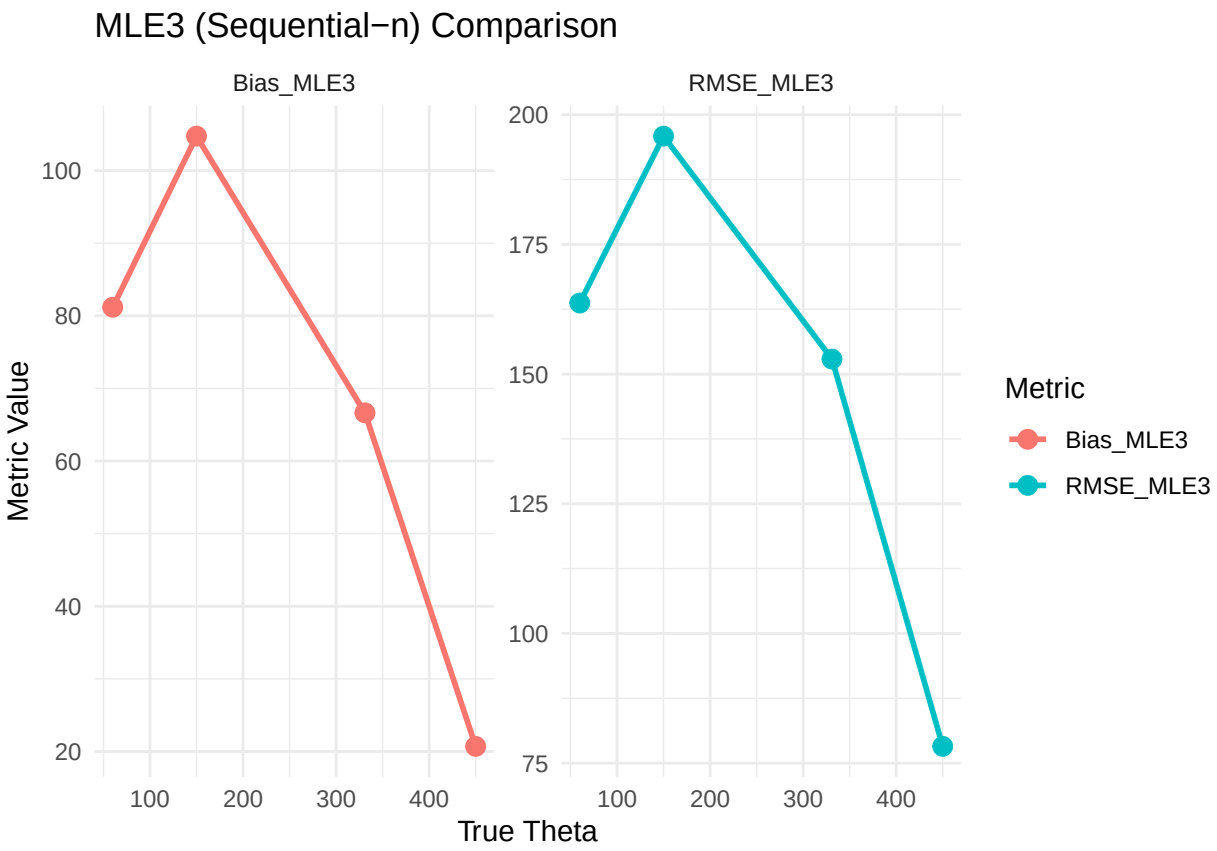
```r
# Convert data for plotting
results_long <- list()
for (n in sample_sizes) {
  results_long[[as.character(n)]] <- melt(
    results_list[[as.character(n)]], id.vars = "True_Theta"
  )
}

# Convert MLE3 results into long format for plotting
results_long_mle3 <- melt(MLE3_results_df, id.vars = "True_Theta")

# Plot Bias and RMSE for MLE1 across different sample sizes
for (n in sample_sizes) {
  p <- ggplot(
    results_long[[as.character(n)]], aes(x = True_Theta, y = value, color = variable)) +
    geom_point(size = 3) +  # Point size is okay
    geom_line(linewidth = 1) +  # Use `linewidth` instead of `size`
    facet_wrap(~variable, scales = "free_y") +
    theme_minimal() +
    labs(
      title = paste("MLE1 Comparison (n =", n, ")"),
      x = "True Theta", y = "Metric Value", color = "Metric"
    )

  print(p)
}
```


MLE1 Comparison (n = 4)
Bias_MLE1
RMSE_MLE1
Metric Value
2
0
−2
110
100
90
80
100
200
300
400
True Theta
Metric
Bias_MLE1
RMSE_MLE1


MLE1 Comparison (n = 5)
Bias_MLE1
RMSE_MLE1
Metric Value
True Theta
Metric
Bias_MLE1
RMSE_MLE1
2
1
0
−1
100
90
80
70
100
200
300
400




```r
# Plot Bias and RMSE for MLE3
p_mle3 <- ggplot(results_long_mle3, aes(x = True_Theta, y = value, color = variable)) +
  geom_point(size = 3) +
  geom_line(linewidth = 1) +  # Use `linewidth` instead of `size`
  facet_wrap(~variable, scales = "free_y") +
  theme_minimal() +
  labs(
    title = "MLE3 (Sequential-n) Comparison",
    x = "True Theta", y = "Metric Value", color = "Metric"
  )

print(p_mle3)  # Explicitly print the final MLE3 plot
```



Comparison of MLE1 and MLE3: Which Estimator of θ is Better?

In this study, we compared two different Maximum Likelihood Estimators (MLEs) for θ:
1. **MLE1 (Fixed-n Estimate)**, where the sample size n is pre-determined.
2. **MLE3 (Sequential-n Estimate)**, where sampling continues until at least one dirty packet is found.

The comparison is based on two key metrics: **bias** (accuracy of the estimator) and **RMSE (Root Mean Square Error)**, which accounts for both accuracy and variability. The results are summarized in the table below:

Metric	**MLE1 (Fixed-n)**	**MLE3 (Sequential-n)**
Bias	Close to **0** (Good)	**Large Overestimation** (Bad)
RMSE	**Lower** (More Stable)	**Higher** (More Variability)

MLE1 is More Accurate (Lower Bias)

MLE1 exhibits **minimal bias**, meaning it is an almost **unbiased estimator** for θ. In contrast, **MLE3 shows significant positive bias**, systematically **overestimating** the true number of dirty packets in the population. This makes MLE3 an unreliable choice for precise estimation.

MLE1 is More Precise (Lower RMSE)

The RMSE values for MLE1 are consistently **lower** than those of MLE3, confirming that MLE1 is a **more reliable estimator** with **less variation** in its estimates. The **higher RMSE** for MLE3 suggests that it is

prone to **large errors**, making it unsuitable for practical use.

Why MLE3 is Not a Good Estimator

MLE3 is problematic because: - **It systematically overestimates** θ, meaning its estimates are consistently **too high**. - **Its RMSE is significantly higher**, indicating a **high degree of uncertainty** and **lack of stability**.

Conclusion: MLE1 is the Best Estimator

From the analysis, it is evident that **MLE1 is the superior estimator** due to its **lower bias and lower RMSE**, making it both **accurate and reliable**. On the other hand, **MLE3 is highly biased and unstable**, making it an inferior choice for estimating θ.

This comparison demonstrates that **a fixed-sample approach (MLE1) outperforms a sequential approach (MLE3) in terms of both accuracy and precision**.