SUMMATIVE ASSESSMENT Section B

Sonal Agarwal

2024-11-22

Section B.1

Let X be a continuous random variable representing the time between the product being sold out and restocked. The probability density function is given by:

$$p_\lambda(x) = \begin{cases} ae^{-\lambda(x-b)} & \text{if } x \geq b \\ 0 & \text{if } x < b \end{cases}$$

where $b > 0$ is a known constant, $\lambda > 0$ is a parameter of the distribution, and a is to be determined by λ and/or b.

(Q1)

To ensure that $p_\lambda(x)$ is a valid probability density function, the integral over the entire range of x must equal 1.

$$\int_b^\infty ae^{-\lambda(x-b)}\,dx = 1$$

Derivation

Solving the integral:

$$\int_b^\infty ae^{-\lambda(x-b)}\,dx = a\int_b^\infty e^{-\lambda(x-b)}\,dx$$

Make a substitution $u = x - b$:

$$= a\int_0^\infty e^{-\lambda u}\,du = a\left[-\frac{1}{\lambda}e^{-\lambda u}\right]_0^\infty = a\cdot\frac{1}{\lambda}$$

To ensure the total probability is 1:

$$a\cdot\frac{1}{\lambda} = 1 \implies a = \lambda$$

Thus, the value of a is λ.

(Q2)

Population Mean

The mean $E(X)$ is calculated as:

$$E(X) = \int_b^\infty x \cdot \lambda e^{-\lambda(x-b)}\, dx$$

Derivation of the Mean

Making a substitution $u = x - b$, hence $x = u + b$:

$$E(X) = \int_0^\infty (u+b)\lambda e^{-\lambda u}\, du$$

$$= \lambda \int_0^\infty u e^{-\lambda u}\, du + \lambda b \int_0^\infty e^{-\lambda u}\, du$$

$$= \frac{1}{\lambda} + b$$

So, the mean is:

$$E(X) = \frac{1}{\lambda} + b$$

Standard Deviation

The variance $\operatorname{Var}(X)$ is:

$$\operatorname{Var}(X) = E(X^2) - (E(X))^2$$

Compute $E(X^2)$:

$$E(X^2) = \int_b^\infty x^2 \cdot \lambda e^{-\lambda(x-b)}\, dx$$

Substitute $u = x - b$:

$$E(X^2) = \int_0^\infty (u+b)^2 \lambda e^{-\lambda u}\, du$$

$$= \lambda \int_0^\infty (u^2 + 2ub + b^2) e^{-\lambda u}\, du$$

$$= \frac{2}{\lambda^2} + \frac{2b}{\lambda} + b^2$$

So:

$$\operatorname{Var}(X) = \frac{1}{\lambda^2}$$

The S.D. σ is:

$$\sigma = \frac{1}{\lambda}$$

(Q3) Cumulative Distribution Function

The cumulative distribution function $F(x)$ is:

$$F(x) = P(X \leq x) = \int_b^x \lambda e^{-\lambda(t-b)}\, dt$$

Compute the integral:

$$F(x) = 1 - e^{-\lambda(x-b)}$$

So, CDF is:

$$F(x) = \begin{cases} 0 & \text{if } x < b \\ 1 - e^{-\lambda(x-b)} & \text{if } x \geq b \end{cases}$$

Quantile Function

To find the quantile function $Q(p)$, solving $F(x) = p$:

$$1 - e^{-\lambda(x-b)} = p \implies e^{-\lambda(x-b)} = 1 - p$$

Taking natural logarithm of both sides:

$$-\lambda(x-b) = \ln(1-p) \implies x = b - \frac{\ln(1-p)}{\lambda}$$

So the quantile function is:

$$Q(p) = b - \frac{\ln(1-p)}{\lambda}$$

(Q4) Likelihood Function

Given $X_1, \cdots, X_n$ are independent observations, the likelihood function $L(\lambda)$ is:

$$L(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda(X_i - b)}$$

The log-likelihood function is:

$$\log L(\lambda) = n \log \lambda - \lambda \sum_{i=1}^{n} (X_i - b)$$

Maximum Likelihood Estimate

To find the maximum likelihood estimate $\hat{\lambda}_{\text{MLE}}$, setting the derivative of the log-likelihood to 0:

$$\frac{\partial}{\partial \lambda} \log L(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^{n} (X_i - b) = 0$$

Solve for λ:

$$\hat{\lambda}\text{MLE} = \frac{n}{\sum i = 1^n (X_i - b)}$$

Thus, the MLE for λ is:

$$\hat{\lambda}\text{MLE} = \frac{n}{\sum i = 1^n (X_i - b)}$$

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
library(boot)
```

(Q5)

```
df <- read.csv("supermarket_data_2024.csv",header = TRUE,sep=",")

head(df) #checking the data
```

```
##   TimeLength
## 1   304.1401
## 2   504.4332
## 3   388.3975
## 4   326.7841
## 5   310.8762
## 6   340.0900
```

```
# Function to calculate MLE Lambda

MLE_lambda <- function(data, b) {
  n <- length(data)
  return(n / sum(data - b))
}

b <- 300

lambda_mle <- MLE_lambda(df$TimeLength, b)

cat("Max Likelihood Estimate for lambda: ", lambda_mle)
```

```
## Max Likelihood Estimate for lambda:  0.01988426
```

(Q6) Bootstrap Confidence Levels

```
set.seed(42)

#Applying bootstrap confidence

bootstrap_res <- boot(data=df$TimeLength,statistic = MLE_lambda, R= 10000)

boot_cnf <- boot.ci(bootstrap_res, type = "perc")
print(boot_cnf)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstrap_res, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   (-0.0011, -0.0011 )
## Calculations and Intervals on Original Scale
```

```
cat("95% Confidence Interval for lambda: (", boot_cnf$percent[4], ", ", boot_cnf$percent[5], ")
\n")
```

```
## 95% Confidence Interval for lambda: ( -0.00114641 ,  -0.001076515 )
```

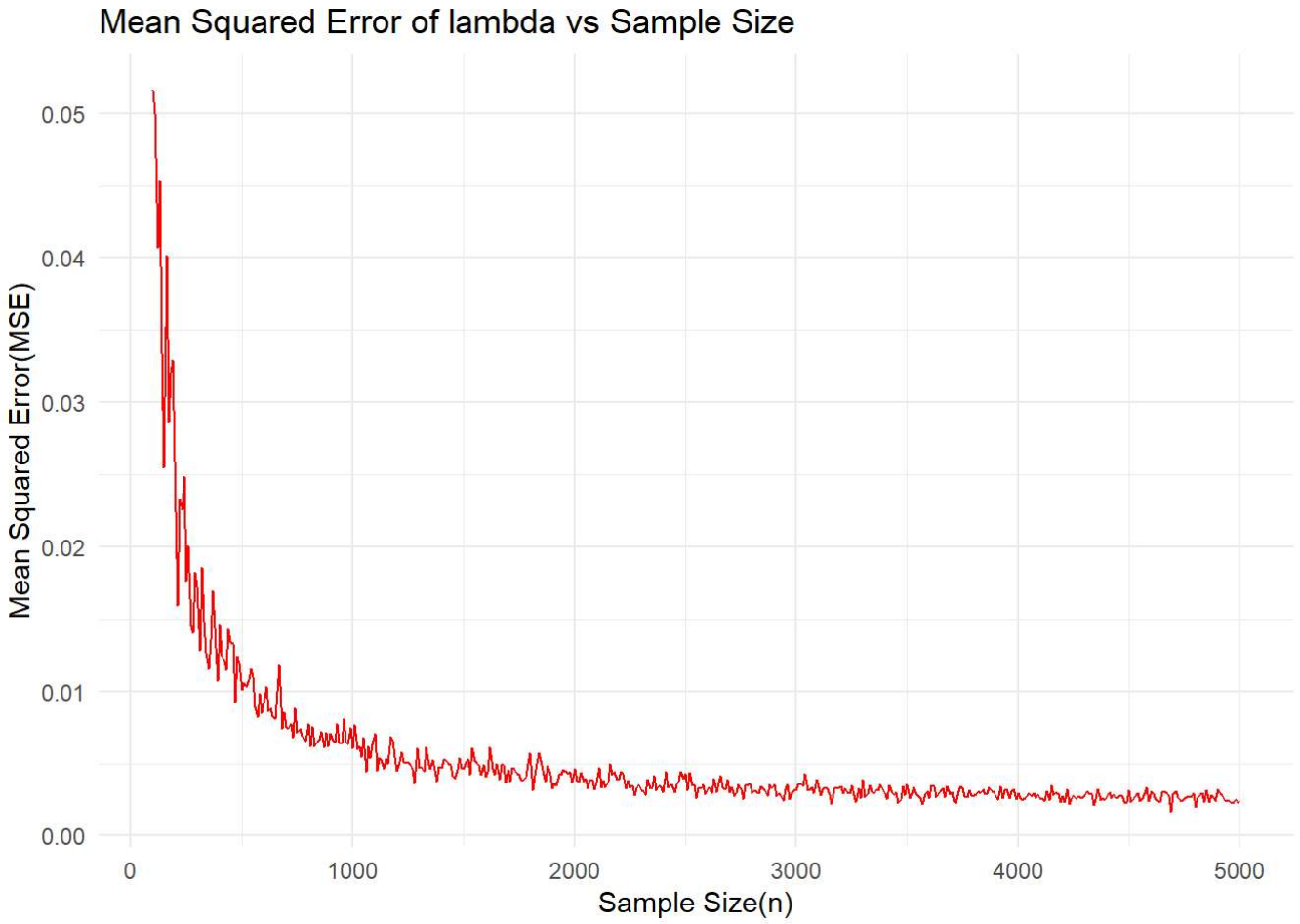
Q7 Simulation Study

```r
library(ggplot2)

true_lambda <- 2
b <- 0.01
sample_size <- seq(100,5000,by=10)
trials <- 100

gen_samples <-function(n,lambda){
  rexp(n,rate=lambda)
}

mse_vals <- numeric(length(sample_size))
```

```r
for( i in seq_along(sample_size)){

  n <- sample_size[i]
  mse_each_trial <- numeric(trials)

  for(trial in 1:trials){

    sample_df<- gen_samples(n, true_lambda) # generates random samples

    lambda_mle <- MLE_lambda(sample_df,b) #calculates MLE for lambda

    mse_each_trial[trial] <- (lambda_mle - true_lambda)^2 #Calculates squared error for this tri
al
  }

  mse_vals[i] <- mean(mse_each_trial) # Calculates MSE for each sample size

}

# Data frame for plotting

mse_data <- data.frame(
  SampleSize = sample_size,
  MSE = mse_vals
)

head(mse_data)
```

```
##   SampleSize        MSE
## 1        100 0.05163135
## 2        110 0.04997287
## 3        120 0.04072675
## 4        130 0.04538363
## 5        140 0.03338518
## 6        150 0.02545949
```

```
ggplot(mse_data, aes(x = SampleSize, y = MSE)) +
  geom_line(color = "red") +
  labs(title = "Mean Squared Error of lambda vs Sample Size",
       x = "Sample Size(n)",
       y = "Mean Squared Error(MSE)") +
  theme_minimal()
```



Section B.2

Simulation Study: Probability and Statistics of Drawing Balls from a Bag

We have a bag with a red balls and b blue balls, where $a \geq 1$ and $b \geq 1$. We randomly draw two balls from the bag without replacement. We define a discrete random variable X as the difference between the number of red balls and blue balls drawn.

Let X be the number of red balls minus the number of blue balls. Our goal is to calculate the probability mass function, expectation, variance, and perform simulations based on the given setup.

(Q1) Probability Mass Function $p_X(x)$

The possible values of X are: - $X = 2$ (if both drawn balls are red). - $X = -2$ (if both drawn balls are blue). - $X = 0$ (if one ball is red and the other is blue).

The possible values of X are:

$$X \in \{-2, 0, 2\}.$$

Derivation:

- $X = 2$: This occurs if both balls drawn are red. The probability is:

$$P(X = 2) = \frac{\binom{a}{2}}{\binom{a+b}{2}} = \frac{\frac{a(a-1)}{2}}{\frac{(a+b)(a+b-1)}{2}} = \frac{a(a-1)}{(a+b)(a+b-1)}.$$

- $X = -2$: This occurs if both balls drawn are blue. The probability is:

$$P(X = -2) = \frac{\binom{b}{2}}{\binom{a+b}{2}} = \frac{\frac{b(b-1)}{2}}{\frac{(a+b)(a+b-1)}{2}} = \frac{b(b-1)}{(a+b)(a+b-1)}.$$

- $X = 0$: This occurs if one ball is red and the other is blue. The probability is:

$$P(X = 0) = \frac{\binom{a}{1}\binom{b}{1}}{\binom{a+b}{2}} = \frac{ab}{\frac{(a+b)(a+b-1)}{2}} = \frac{2ab}{(a+b)(a+b-1)}.$$

Thus, the probability mass function $p_X(x)$ is:

$$p_X(x) = \begin{cases} \frac{a(a-1)}{(a+b)(a+b-1)}, & \text{if } x = 2, \\ \frac{b(b-1)}{(a+b)(a+b-1)}, & \text{if } x = -2, \\ \frac{2ab}{(a+b)(a+b-1)}, & \text{if } x = 0. \end{cases}$$

Calculating the probabilities for each of these outcomes using combinations.

```
pX <- function(x, a, b) {
  total <- a + b
  if (x == 2) {
    return((a * (a - 1)) / (total * (total - 1)))
  } else if (x == -2) {
    return((b * (b - 1)) / (total * (total - 1)))
  } else if (x == 0) {
    return((2 * a * b) / (total * (total - 1)))
  } else {
    return(0) # For values of x not in {-2, 0, 2}
  }
}
```

(Q2) Expectation of X

The expectation of X is given by:

$$E[X] = \sum_{x \in \{-2,0,2\}} x \cdot p_X(x).$$

Substituting the values:

$$E[X] = -2 \cdot \frac{b(b-1)}{(a+b)(a+b-1)} + 0 \cdot \frac{2ab}{(a+b)(a+b-1)} + 2 \cdot \frac{a(a-1)}{(a+b)(a+b-1)}.$$

Simplify:

$$E[X] = \frac{2a(a-1) - 2b(b-1)}{(a+b)(a+b-1)}.$$

(Q3) Variance of X

The variance is given by:

$$\mathrm{Var}(X) = E(X^2) - [E(X)]^2.$$

Step 1: Compute $E(X^2)$

$$E(X^2) = \sum_x x^2 \cdot p_X(x).$$

Substituting the values:

$$E(X^2) = 2^2 \cdot P(X=2) + 0^2 \cdot P(X=0) + (-2)^2 \cdot P(X=-2).$$

$$E(X^2) = 4 \cdot \frac{a(a-1)}{(a+b)(a+b-1)} + 4 \cdot \frac{b(b-1)}{(a+b)(a+b-1)}.$$

Simplify:

$$E(X^2) = \frac{4a(a-1) + 4b(b-1)}{(a+b)(a+b-1)}.$$

Step 2: Compute $\mathrm{Var}(X)$

Substitute $E(X^2)$ and $[E(X)]^2$:

$$\mathrm{Var}(X) = \frac{4a(a-1) + 4b(b-1)}{(a+b)(a+b-1)} - \left[\frac{2a^2 - 2a - 2b^2 + 2b}{(a+b)(a+b-1)}\right]^2.$$

(Q4) Functions to compute Expectation and Variance

```
#Expectation of X

compute_expectation_X <- function(a, b) {
  total <- a + b
  numerator <- 2 * a * (a - 1) - 2 * b * (b - 1)
  denominator <- total * (total - 1)
  return(numerator / denominator)
}



# Variance of X
compute_variance_X <- function(a, b) {
  total <- a + b

  # Compute E(X^2)
  EX2 <- (4 * a * (a - 1) + 4 * b * (b - 1)) / (total * (total - 1))

  # Compute E(X)
  EX <- compute_expectation_X(a, b)

  # Variance formula
  return(EX2 - EX^2)
}


# Display Functions
compute_expectation_X(5, 8)  # Example for a = 5, b = 8
```

```
## [1] -0.4615385
```

```
compute_variance_X(5,8)     # Example for a = 5, b = 8
```

```
## [1] 1.7357
```

(Q5) Expectation of Sample Mean X:

$$E(\bar{X}) = E(X) = b + \frac{1}{a}$$

```
# Expectation of the Sample Mean
expectation_sample_mean <- compute_expectation_X(5, 8)
expectation_sample_mean
```

```
## [1] -0.4615385
```

(Q6) Variance of Sample Mean X:

$$\text{Var}(\bar{X}) = \frac{\text{Var}(X)}{n} = \frac{1}{a^2 n}$$

```
# Variance of the Sample Mean
variance_sample_mean <- compute_variance_X(3, 5) / 100  # Example for n = 100
variance_sample_mean
```

```
## [1] 0.01607143
```

(Q7) Function to generate sample of Xs

```
# Generate Sample of X's
sample_Xs <- function(a, b, n)
{
  # Calculate probabilities of each outcome
  p2 <- (a/(a+b)) * ((a-1)/(a+b-1))
  p0 <- 2 * (a/(a+b)) * (b/(a+b-1))
  p_2 <- (b/(a+b)) * ((b-1)/(a+b-1))

  # Create a vector of possible outcomes and their probabilities
  outcomes <- c(2, 0, -2)
  probs <- c(p2, p0, p_2)

  # Sample n times from the discrete distribution
  return(sample(outcomes, size = n, replace = TRUE, prob = probs))
  }
```

(Q8)

```r
# Set parameters
a <- 3
b <- 5
n <- 100000

compute_expectation_X <- function(a, b)
{
   return( 2 * (a * (a - 1) - b * (b - 1)) /  ((a + b) * (a + b - 1)) )
}

compute_variance_X <- function(a, b)
{
  E_X2 <- 4 * (b * (b - 1) + a * (a - 1)) /  ((a + b) * (a + b - 1))
  E_X <- compute_expectation_X(a, b)
  return(E_X2 - E_X^2)
}

sample_Xs <- function(a, b, n)
{
  # Calculate probabilities of each outcome
  p2 <- (a/(a+b)) * ((a-1)/(a+b-1))
  p0 <- 2 * (a/(a+b)) * (b/(a+b-1))
  p_2 <- (b/(a+b)) * ((b-1)/(a+b-1))

  # Create a vector of possible outcomes and their probabilities
  outcomes <- c(2, 0, -2)
  probs <- c(p2, p0, p_2)

  # Sample n times from the discrete distribution
  sample(outcomes, size = n, replace = TRUE, prob = probs)
}

# Compute the theoretical expectation and variance
EX_theoretical <- compute_expectation_X(a, b)
Var_X_theoretical <- compute_variance_X(a, b)

#storing the sample of X
X_sample <- sample_Xs(a, b, n)


# Compute the sample mean and sample variance
sample_mean_X <- mean(X_sample)
sample_variance_X <- var(X_sample)

# Compare sample results with theoretical results
cat("Difference b/w sample mean and theoretical E(X):", sample_mean_X - EX_theoretical, "\n")
```

```
## Difference b/w sample mean and theoretical E(X): -0.0038
```

```
cat("Difference b/w sample variance and theoretical Var(X):", sample_variance_X - Var_X_theoreti
cal, "\n")
```

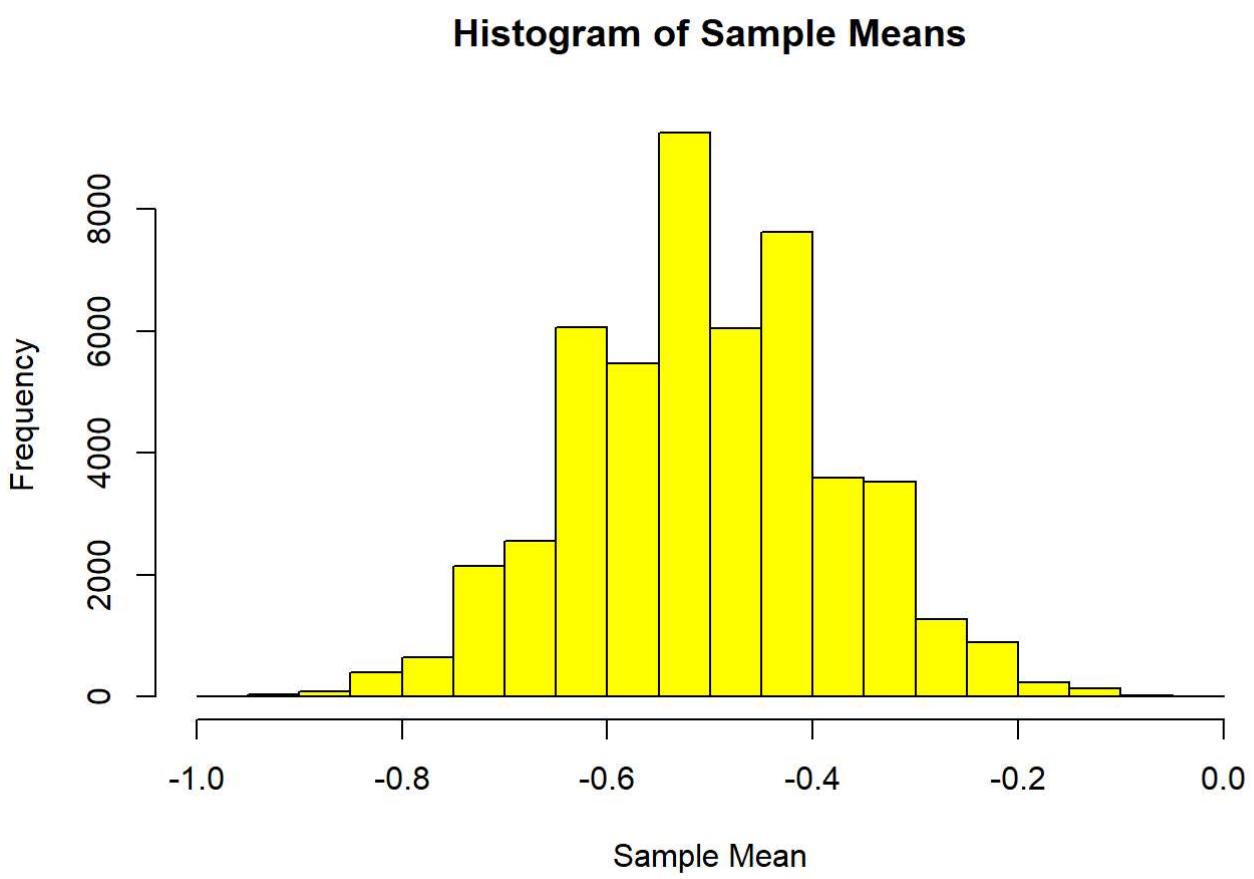
```
## Difference b/w sample variance and theoretical Var(X): -0.001581242
```

```
cat('Difference between sample and theoretical values is too close.')
```

```
## Difference between sample and theoretical values is too close.
```

(Q9) Simulation study with 50000 trials:

```
a <- 3
b <- 5
n <- 100
num_trials <- 50000

# Function to simulate a single trial
simulate_trial <- function(a, b, n) {
  sample_x <- sample_Xs(a, b, n)
  mean(sample_x)
}

# Simulate multiple trials
sample_means <- replicate(num_trials, simulate_trial(a, b, n))

# Visualize the distribution of sample means
hist(sample_means, breaks = 30, col="yellow",main = "Histogram of Sample Means", xlab = "Sample
Mean", )
```



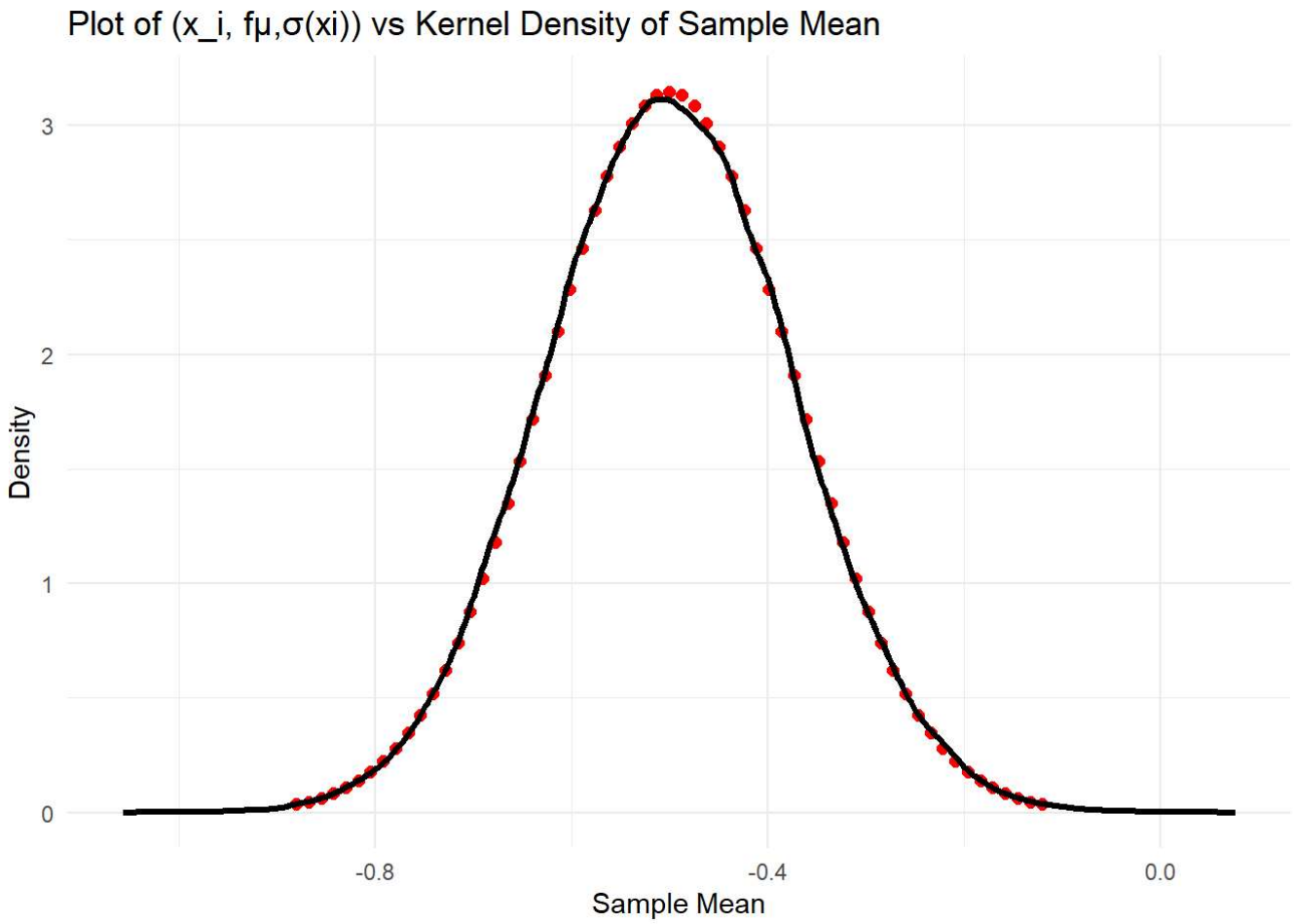

```
# Simulation of 50000 Trials
simulation_results <- replicate(50000, mean(sample_Xs(3, 5, 100)))

# Summary of Results
summary(simulation_results)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0400 -0.5800 -0.5000 -0.5016 -0.4200  0.0400
```

(Q10) Sample Mean's Kernel Density Plot

```
#params for normal distribution
mu <- compute_expectation_X(a, b)
sigma <- sqrt(compute_variance_X(a, b) / n)
# Function to calculate the Gaussian Dist
gaussian_df <- function(x, mu, sigma)
{
  return(1 / (sqrt(2 * pi * sigma^2)) * exp(- (x - mu)^2 / (2 * sigma^2)))
}

# Generate sequence of x values between μ - 3σ and μ + 3σ in increments of 0.1σ
x_vals <- seq(mu - 3 * sigma, mu + 3 * sigma, by = 0.1 * sigma)

# Calculate the corresponding Gaussian PDF values for each x_i
y_vals <- sapply(x_vals, function(x) gaussian_df(x, mu, sigma))

# Conduct simulation and generate the sample means (as before)
num_trials <- 50000
sample_means <- numeric(num_trials)

for (trial in 1:num_trials)
{
  X_sample <- sample_Xs(a, b, n)
  sample_means[trial] <- mean(X_sample)
}

# Create the scatter plot for the points (x_i, fμ,σ(xi))
scatter_data <- data.frame(x = x_vals, y = y_vals)
scatter_plot <- ggplot(scatter_data, aes(x = x, y = y)) +
  geom_point(color = "red", size = 2) +
  labs(title = "Plot of (x_i, fμ,σ(xi)) vs Kernel Density of Sample Mean",
       x = "Sample Mean", y = "Density") +
  theme_minimal()

# Kernel density estimate of the sample means
density_estimate <- density(sample_means)

# Add the kernel density estimate to the plot
scatter_plot +
  geom_line(data = data.frame(x = density_estimate$x, y = density_estimate$y),
            aes(x = x, y = y), color = "black", size = 1.2) +

  theme(legend.position = "none")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



(Q11) Relationship between Density of X and Gaussian Distribution

The relationship between the kernel density of the sample mean X and the Gaussian distribution can be understood as follows:

As the sample size n increases, the distribution of the sample mean $\bar{X}$ approaches a normal distribution. This phenomenon is explained by the Central Limit Theorem, which indicates, as the number of trials and the sample size increases, the KDE should approximate the Gaussian distribution.

In the simulation, the kernel density of the sample mean X is plotted along with the probability density function of a Gaussian distribution with mean $E(X)$ and variance $\frac{\mathrm{Var}(X)}{n}$. As expected:

- The kernel density closely resembles the Gaussian curve centered around the mean $E(X)$, which indicates that the distribution of the sample mean X is approximately normal.
- The spread of the kernel density shows that the variance decreases as n increases, leading to a narrower & tall distribution.

This confirms the behavior of sample mean X as predicted by CLT.