Summative Assessment Section B

Keli Niu (ad21083)

2024-11-05

B.1 Probability Density Function Analysis

Analyse a given probability density function (PDF) of a random variable. Its variable represents the time it takes for a product to return to the shelves after it has been sold out at a supermarket.

The probability density function is given by:

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

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

(1) Determining the Value of a

Since $p_\lambda(x)$ is a probability density function, it must satisfy two key properties: $p_\lambda(x) \geqslant 0$ for all x, and the total area under the PDF must be equal to 1. Therefore, we begin by ensuring that:

$$\int_{-\infty}^{\infty} p_\lambda(x)\, dx = 1$$

Since $p_\lambda(x) = 0$ for $x < b$, we only need to integrate from b to ∞:

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

To solve for a, we evaluate the integral:

$$\begin{aligned} \int_b^{\infty} ae^{-\lambda(x-b)} dx &= a \int_b^{\infty} e^{-\lambda(x-b)} dx \\ &= a \cdot \left[-\frac{1}{\lambda} e^{-\lambda(x-b)} \right]_{x=b}^{x=\infty} \\ &= 0 - (-\frac{1}{\lambda} \cdot a) \\ &= \frac{a}{\lambda} = 1 \end{aligned}$$

Thus, we find that:

$$a = \lambda$$

Now, we verify that $p_\lambda(x) > 0$ for all $x \geqslant b$ when $a = \lambda$. Since $\lambda > 0$, it follows that

$$p_\lambda(x) = \lambda e^{-\lambda(x-b)} > 0, \quad \text{for } x \geqslant b.$$

Thus, the PDF is non-negative across its domain, confirming that $a = \lambda$ is a valid choice.

(2) Population Mean and Standard Deviation

From the problem statement, we know that X is a continuous random variable, and the population mean is equivalent to the expected value of X. The expected value $\mathbb{E}(X)$ is given by:

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} x\, p_\lambda(x)\, dx$$

Since $p_\lambda(x) = 0$ for $x < b$, we only need to integrate from b to ∞:

$$
\begin{aligned}
\mathbb{E}(X) &= \int_b^{\infty} x\, p_\lambda(x)\, dx \\
&= \int_b^{\infty} x\lambda e^{-\lambda(x-b)} dx \\
&= e^{\lambda b} \int_b^{\infty} x\lambda e^{-\lambda x}\, dx \\
&= e^{\lambda b}(\left[-xe^{-\lambda x}\right]_b^{\infty} + \int_b^{\infty} e^{-\lambda x}\, dx) \\
&= e^{\lambda b}(0-(-be^{-\lambda b}) + (\left[-\frac{1}{\lambda}e^{-\lambda x}\right]_b^{\infty})) \\
&= e^{\lambda b}(b+\frac{1}{\lambda})e^{-\lambda b} \\
&= b+\frac{1}{\lambda}
\end{aligned}
$$

where in the 4th equality we have used integration by parts: let $u = \lambda x$, $dv = e^{-\lambda x}dx$, which gives $du = \lambda dx$ and $v = -\frac{1}{\lambda}e^{-\lambda x}$. Then using the integration by parts formula: $\int udv = uv - \int vdu$

To compute the variance $\mathrm{Var}(X)$, with integration by parts again we have

$$
\begin{aligned}
\mathbb{E}(X^2) &= \int_{-\infty}^{\infty} x^2\, p_\lambda(x)\, dx \\
&= \int_b^{\infty} x^2\lambda e^{-\lambda(x-b)} dx \\
&= e^{\lambda b} \int_b^{\infty} x^2\lambda e^{-\lambda x}\, dx \\
&= e^{\lambda b}(\left[-x^2e^{-\lambda x}\right]_b^{\infty} + 2\int_b^{\infty} xe^{-\lambda x}\, dx) \\
&= e^{\lambda b}(0-(-b^2e^{-\lambda b}) + (\frac{2e^{-\lambda b}}{\lambda}\mathbb{E}(X))) \\
&= e^{\lambda b}((b^2e^{-\lambda b}) + \frac{2}{\lambda}(b+\frac{1}{\lambda})e^{-\lambda b}) \\
&= b^2+\frac{2b}{\lambda}+\frac{2}{\lambda^2}
\end{aligned}
$$

where in the 4th equality we have also used integration by parts.

Now, we can get the variance: $\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = b^2 + \frac{2b}{\lambda} + \frac{2}{\lambda^2} - (b+\frac{1}{\lambda})^2 = \frac{1}{\lambda^2}$

So, the standard deviation σ_X is the square root of the variance:

$$
\sigma_X = \sqrt{\mathrm{Var}(X)} = \sqrt{\frac{1}{\lambda^2}} = \frac{1}{\lambda}.
$$

In summary, the population mean and standard deviation are $\mathbb{E}(X) = b + \frac{1}{\lambda}$ and $\sigma_X = \frac{1}{\lambda}$.

(3) Cumulative Distribution Function and the Quantile Function

The Cumulative Distribution Function:

The CDF $F_X(x)$ is defined as the probability that $X \leqslant x$:

$$
F_X(x) = P(X \leqslant x) = \int_{-\infty}^{x} p_\lambda(t)\, dt
$$

We consider two cases: 1. Since $F_X(x) = 0$ for $x < b$:

$$
F_X(x) = 0
$$

2. Since $F_X(x) = p_\lambda(t)$ for $x \geqslant b$, we have:

$$
\begin{aligned}
F_X(x) &= \int_b^x p_\lambda(t)\,dt \\
&= \int_b^x \lambda e^{-\lambda(t-b)} dt \\
&= e^{\lambda b} \int_b^x \lambda e^{-\lambda t}\,dt \\
&= e^{\lambda b} \left[-e^{-\lambda t}\right]_b^x \\
&= 1 - e^{-\lambda(x-b)}
\end{aligned}
$$

The Quantile Function:

The quantile function $F_X^{-1} : [0,1] \to \mathbb{R}$ is defined by:

$F_X^{-1}(p) := \inf\{x \in \mathbb{R} : F_X(x) = \mathbb{P}(X \leqslant x) \geqslant p\}$ for $p \in [0,1]$.

The quantile function is the inverse of the CDF, so we starting from the equation $F_X(x) = 1 - e^{-\lambda(x-b)} = p$, solve for x:

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

Taking the natural logarithm on both sides:

$$
-\lambda(x-b) = \ln(1-p)
$$

Thus, the quantile function is:

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

for $0 < p < 1$, $F_X^{-1}(p) = b$ for $p = 0$ and $F_X^{-1}(p) = +\infty$ for $p = 1$.

(4) Maximum Likelihood Estimate

The likelihood function $L(\lambda)$ for n independent observations is given by the product of their PDFs:

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

The log-likelihood function is obtained by taking the natural logarithm of the likelihood function to find the maximum likelihood estimate:

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

$$
\frac{d}{d\lambda} logL(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^{n}(X_i - b)
$$

We find:

If $\lambda < \frac{n}{\sum_{i=1}^{n}(X_i-b)}$, then $\frac{d}{d\lambda} logL(\lambda) > 0$;

If $\lambda > \frac{n}{\sum_{i=1}^{n}(X_i-b)}$, then $\frac{d}{d\lambda} logL(\lambda) < 0$.

So the maximum likelihood estimate for λ is $\hat{\lambda}_{\text{MLE}} = \frac{n}{\sum_{i=1}^{n}(X_i-b)} = \frac{1}{\overline{X}-b}$, where $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$.

(5) Calculate the MLE of the parameter λ.

Step 1: Load the data and set the parameters

```r
#Load the library and read the file
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 become err
ors
```

```
supermarket_data <- read_csv("supermarket_data_2024.csv")
```

```
## Rows: 2500 Columns: 1
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (1): TimeLength
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Set the parameters
b<-300
n<-nrow(supermarket_data)
```

Step 2: Calculate the MLE and display it

```
#Calculate the MLE using the function in question (4)
lambda_MLE <- 1/(mean(supermarket_data$TimeLength, na.rm=TRUE)-b)
lambda_MLE
```

```
## [1] 0.01988426
```

(6) Calculate the MLE of the parameter λ.

Step 1: Set Up Parameters and Initialize Storage

```
num_resamples <- 10000
# Create a numerical vector to store the results of the MLE
bootstrap_estimates <- numeric(num_resamples)
```

Step 2: Perform Bootstrap Resampling and Compute MLE for Each Resample

```
# Set a random seed to ensure the results can be reproduced
set.seed(0)
for (i in 1:num_resamples) {
  #Samples are randomly selected from the original data
  sample_data <- sample(supermarket_data$TimeLength, size = length(supermarket_data$TimeLength), repl
ace = TRUE)

  # Calculate the lambda MLE of the current resampling sample
  lambda_hat <- 1 / (mean(sample_data, na.rm = TRUE) - b)

  # Store the results of the MLE
  bootstrap_estimates[i] <- lambda_hat
}
```

Step 3: Calculate the 95% Confidence Interval for Lambda

```
# Calculate 95% confidence intervals
Confidence_lower <- quantile(bootstrap_estimates, 0.025)
Confidence_upper <- quantile(bootstrap_estimates, 0.975)
Confidence <- c(Confidence_lower, Confidence_upper)

# Print the result
cat("Lambda MLE (original data):", 1 / (mean(supermarket_data$TimeLength, na.rm = TRUE) - b), "\n")
```

```
## Lambda MLE (original data): 0.01988426
```

```
cat("95% Bootstrap Confidence Interval for Lambda:", Confidence, "\n")
```

```
## 95% Bootstrap Confidence Interval for Lambda: 0.01911276 0.02069813
```

Explanation

- `quantile()` is used to construct confidence intervals based on the results of Bootstrap resampling. Specifically, it determines the upper and lower limits of the parameter estimates at the 95% confidence level by calculating the 2.5% and 97.5% quantiles.

(7) Display a plot of the mean square error of $\lambda_M LE$

Step 1: Set parameters

```
library(tidyverse)
set.seed(0)
# Set parameters
lambda_true <- 2
b <- 0.01
num_trials_per_sample_size <- 100
sample_sizes <- seq(100, 5000, by = 10)
```

Step 2: Create Data Frame

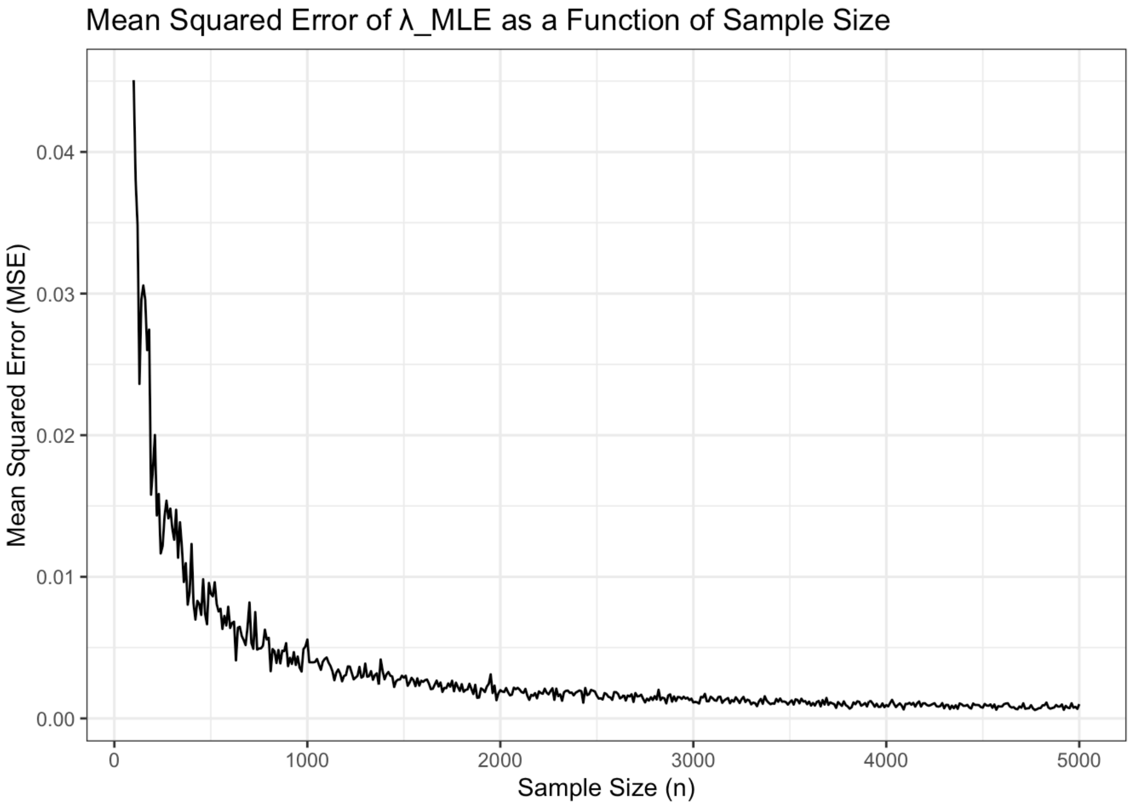
```
# Create data boxes with different sample sizes and trial times
df <- crossing(
  sample_size = sample_sizes,
  trial = 1:num_trials_per_sample_size) %>%
  # create samples
  mutate(samples = map(sample_size, ~ rexp(.x, rate = lambda_true) + b)) %>%
  # compute MLE
  mutate(lambda_mle = map_dbl(samples, ~ 1 / (mean(.x) - b)))
```

Step 3: Calculate Mean Squared Error (MSE) for Each Sample Size

```
# Calculate the MSE for each sample size
df_mse <- df %>%
  group_by(sample_size) %>%
  summarise(mse = mean((lambda_mle - lambda_true)^2))
```

Step 4: Display a plot

```
# Show the plot
ggplot(data = df_mse, aes(x = sample_size, y = mse)) +
  geom_line() +
  theme_bw() +
  xlab("Sample Size (n)") +
  ylab("Mean Squared Error (MSE)") +
  ggtitle("Mean Squared Error of λ_MLE as a Function of Sample Size")
```



Explanation

- `crossing()` function creates a data box df that contains all the sample sizes combined with the number of trials.
- `map()` generates a set of random sample samples for each sample size.
- `rexp(.x, rate = lambda_true)` generates the sample of exponential distribution, and + b offsets the sample.
- `group_by(sample size)` : Groups data by sample size.
- Note: Note: Although `crossing()` and `map()` make the code simpler, they may reduce randomness in small samples, leading to smoother MSE results and may not capture the variability as effectively as independently generated samples in a simple loop.

B.2 Ball Sampling Problem

(1) The Probability Mass Function

First, from the problem, we know that the random variable X represents the difference between the number of red balls drawn and the number of blue balls drawn. Therefore, X has three possible values: 1. If we draw two red balls, then $X = 2$. 2. If we draw one red ball and one blue ball, then $X = 0$. 3. If we draw two blue balls, then $X = -2$.

Thus, the goal is to find the probability mass function (PMF) of the random variable X, that is, $p_X(x) = P(X = x)$ for $x \in \{2, 0, -2\}$.

Calculating the probabilities for each case

1. Case 1: $X = 2$

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

2. Case 2: $X = 0$

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

3. Case 3: $X = -2$

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

Therefore, the PMF is:

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

(2) The Expression of the Expectation

The expectation $\mathbb{E}(X)$ is defined as the sum of each possible value of the random variable multiplied by its corresponding probability.

So, we have:

$$\begin{aligned} \mathbb{E}(X) &= \sum_x x \cdot p_X(x) \\ &= 2 \cdot \frac{a(a-1)}{(a+b)(a+b-1)} + 0 \cdot \frac{2ab}{(a+b)(a+b-1)} + (-2) \cdot \frac{b(b-1)}{(a+b)(a+b-1)} \\ &= \frac{2(a(a-1) - b(b-1))}{(a+b)(a+b-1)} \end{aligned}$$

(3) The Expression of the Variance

We can use the formula, and have:

$$\begin{aligned} \text{Var}(X) &= \mathbb{E}(X^2) - (\mathbb{E}(X))^2 \\ &= (2^2 \cdot \frac{a(a-1)}{(a+b)(a+b-1)} + 0^2 \cdot \frac{2ab}{(a+b)(a+b-1)} + (-2)^2 \cdot \frac{b(b-1)}{(a+b)(a+b-1)}) - \left(\frac{2(a(a-1) - b(b-1))}{(a+b)(a+b-1)}\right)^2 \\ &= \frac{4(a(a-1) + b(b-1))}{(a+b)(a+b-1)} - \frac{4(a(a-1) - b(b-1))^2}{(a+b)^2(a+b-1)^2} \end{aligned}$$

(4) Use R to Represent the Expectation and the Variance

```r
# Step 1: Function to compute the expectation E(X)
compute_expectation_X <- function(a, b) {
  # Calculating E(X) using the formula derived previously
  expectation_X <- (2 * (a * (a - 1) - b * (b - 1))) / ((a + b) * (a + b - 1))
  return(expectation_X)
}

# Step 2: Function to compute the variance Var(X)
compute_variance_X <- function(a, b) {
  # Calculating E(X^2) based on the possible values of X
  E_X_squared <- (4 * (a * (a - 1) + b * (b - 1))) / ((a + b) * (a + b - 1))

  E_X <- compute_expectation_X(a, b)

  # Calculating Var(X) = E(X^2) - (E(X))^2
  variance_X <- E_X_squared - E_X^2
  return(variance_X)
}
```

(5) The Expectation of the Random Variable $\overline{X}$

By the linearity of expectation, we can find the expectation of the sample mean $\mathbb{E}(\overline{X})$ as follows:

$$
\begin{aligned}
\mathbb{E}(\overline{X}) &= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(X_i) \\
&= \frac{1}{n}\cdot n \cdot \mathbb{E}(X) \\
&= \mathbb{E}(X) \\
&= \frac{2(a(a-1)-b(b-1))}{(a+b)(a+b-1)}
\end{aligned}
$$

where in the third equality, we have used the fact that $X_1, X_2, \ldots, X_n$ are $i.i.d.$ random variables, meaning $E(X_i) = E(X)$ for each i.

(6) The Variance of the Random Variable $\overline{X}$

For the sample mean $\overline{X}$ of $i.i.d.$ random variables, we can derive the following formula step by step according to the nature of variance:

$$
\begin{aligned}
\text{Var}(\overline{X}) &= \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \\
&= \frac{1}{n^2}\text{Var}\left(\sum_{i=1}^{n} X_i\right) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(X_i) \\
&= \frac{1}{n^2}\cdot n \cdot \text{Var}(X) \\
&= \frac{\text{Var}(X)}{n} \\
&= \frac{1}{n}\left(\frac{4(a(a-1)+b(b-1))}{(a+b)(a+b-1)} - \frac{4(a(a-1)-b(b-1))^2}{(a+b)^2(a+b-1)^2}\right)
\end{aligned}
$$

(7) Generate $i.i.d.$ Sample of X

```r
# Function to generate a sample of independent copies of X
sample_Xs <- function(a, b, n) {
  # Define the possible values of X
  values <- c(2, 0, -2)

  # Calculate the probabilities
  p_X_2 <- a * (a - 1) / ((a + b) * (a + b - 1))
  p_X_0 <- 2 * a * b / ((a + b) * (a + b - 1))
  p_X_minus2 <- b * (b - 1) / ((a + b) * (a + b - 1))

  # Generate n samples of X based on the probabilities
  samples <- sample(values, size = n, replace = TRUE, prob = c(p_X_2, p_X_0, p_X_minus2))

  return(samples)
}
```

(8) Validation derivation

Step 1: Compute theoretical expectation and variance

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

theoretical_expectation <- compute_expectation_X(a, b)
theoretical_variance <- compute_variance_X(a, b)
cat("Theoretical Expectation E(X):", theoretical_expectation, "\n")
```

```
## Theoretical Expectation E(X): -0.5
```

```
cat("Theoretical Variance Var(X):", theoretical_variance, "\n")
```

```
## Theoretical Variance Var(X): 1.607143
```

Step 2: Generate a sample and compute sample mean and variance

```
set.seed(0)
samples_X <- sample_Xs(a, b, n)
samples_mean <- mean(samples_X)
samples_variance <- var(samples_X)
cat("Sample Mean:", samples_mean, "\n")
```

```
## Sample Mean: -0.50252
```

```
cat("Sample Variance:", samples_variance, "\n")
```

```
## Sample Variance: 1.60765
```

Step 3: Compare results

```
different_mean<-samples_mean - theoretical_expectation
different_variance<-samples_variance - theoretical_variance
cat("Difference between Sample Mean and Theoretical E(X):",different_mean , "\n")
```

```
## Difference between Sample Mean and Theoretical E(X): -0.00252
```

```
cat("Difference between Sample Variance and Theoretical Var(X):",different_variance, "\n")
```

```
## Difference between Sample Variance and Theoretical Var(X): 0.000506869
```

Explanation (observation)

Expectation

Through computation, the difference between the sample mean and the theoretical expectation is found to be -0.00252. According to the , for independent and identically distributed (i.i.d.) random variables, as the sample size $n \to \infty$, the sample mean $X = \frac{1}{n} \sum_{i=1}^{n} X_i$ converges in probability to the population mean $E(X)$. This is formally expressed as:

$$\lim_{n\to\infty} P\left(\left|X - E(X)\right| \geqslant \epsilon\right) = 0, \quad \forall \epsilon > 0.$$

In the case of $n = 100,000$, the small difference -0.00252 confirms that the sample mean is a highly accurate estimate of the theoretical expectation, validating the law of large numbers in practice.

Variance

The difference between the sample variance and the theoretical variance is 0.000506869. By the properties of , the sample variance is an unbiased estimator of the population variance $\mathrm{Var}(X)$. Moreover, the variance of the sample mean can be derived as follows:

$$\mathrm{Var}(\overline{X}) = \frac{\sigma^2}{n} = \frac{\mathrm{Var}(X)}{n}.$$

As n increases, $\mathrm{Var}(\overline{X})$ decreases, and the sample variance converges to the theoretical variance. For large n, the effect of randomness diminishes, and the small difference observed here further verifies that the sample variance reliably approximates the population variance, consistent with statistical theory.

(9) Conduct a Simulation study with 50000 Trials

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

# Store sample means
sample_means <- numeric(trials)
set.seed(6)
for (i in 1:trials) {
  sample_data <- sample_Xs(a, b, n)
  # Calculate the mean of every trial
  sample_means[i] <- mean(sample_data)
}
```

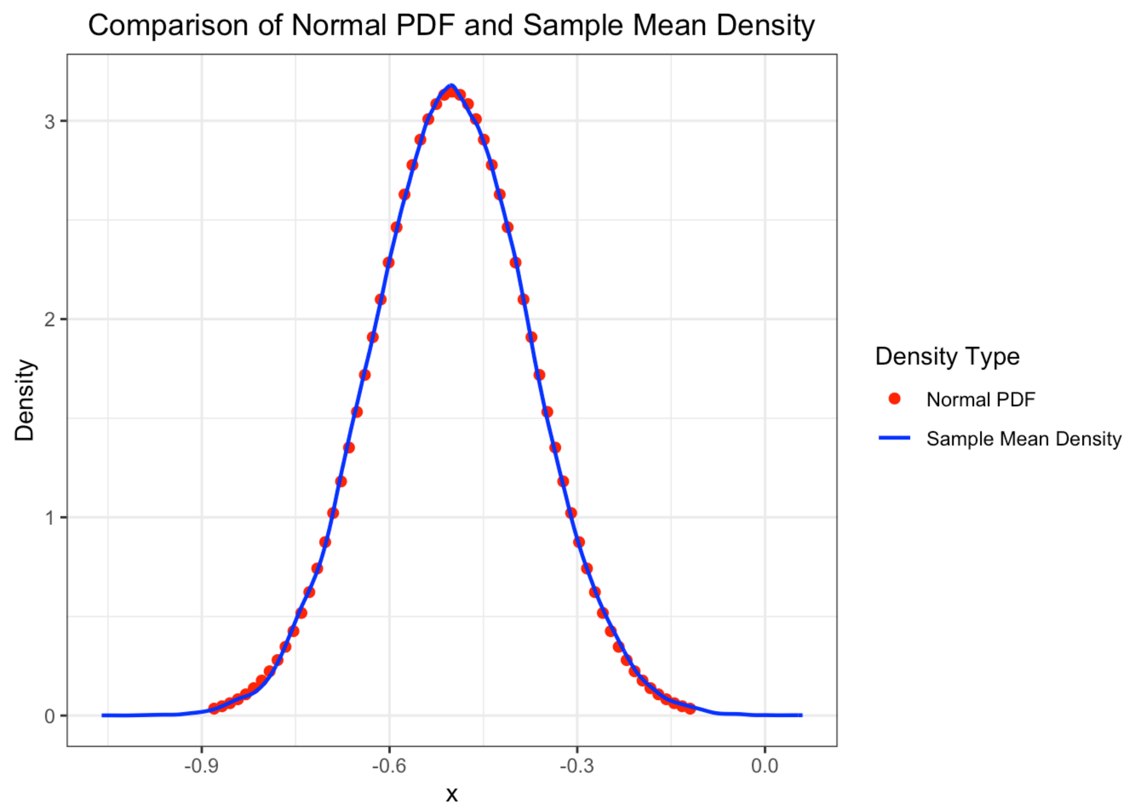
(10) Plot a Comparison

```
# Using the function to calculate the expectation and standard deviation
mu<-compute_expectation_X(a,b)
sigma <- sqrt(compute_variance_X(a, b) / n)

# Set the range of xi
xi <- seq(mu - 3 * sigma, mu + 3 * sigma, by = 0.1 * sigma)
# Calculate the probability density fμ,σ(xi) for each xi
f_mu_sigma <- dnorm(xi, mean = mu, sd = sigma)

ggplot() +
  # Scatter plot for the normal PDF points
  geom_point(data = data.frame(x = xi, y = f_mu_sigma), aes(x = x, y = y,color = "Normal PDF"), size=
1.8) +
  # Kernel density line for the sample means
  geom_line(stat = "density", data = data.frame(x = sample_means), aes(x = x,color = "Sample Mean Den
sity"),size=0.8) +

  labs(x = "x", y = "Density", title = "Comparison of Normal PDF and Sample Mean Density")  +
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_color_manual(name = "Density Type", values = c("Normal PDF" ="red" ,"Sample Mean Density" ="b
lue"))
```



(11) Describe and Explain the Relationship

Description the Relationship

In the plot, we observe that the kernel density estimation of the sample mean (blue curve) closely aligns with the theoretical normal distribution $f_{\mu,\sigma}$ (red points). This phenomenon can be explained by the ***Central Limit Theorem (CLT)***.

Explanation

Description and explanation of the central limit theorem

The Central Limit Theorem describes the behavior of the sample mean distribution when the sample size is sufficiently large. Specifically, consider a set of independent and identically distributed ($i.i.d.$) random variables $X_1, X_2, \ldots, X_n$, each with expected value μ and variance σ^2. When the sample size n is large enough, the distribution of the sample mean $\overline{X}$ approximates a normal distribution with mean μ and variance $\frac{\sigma^2}{n}$.

This means that, as long as n is sufficiently large, the sample mean distribution will approximate a normal distribution.

The Central Limit Theorem can be expressed as:

$$\lim_{n\to\infty} \mathbb{P}\left(\sqrt{\frac{n}{\sigma^2}}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right) \leqslant x\right) = \mathbb{P}(Z \leqslant x),$$

Combined with the theorem to explain this problem

In the context of this question, we selected a sample size of $n = 100$ and conducted 50,000 trials. In each trial, we generated a sample mean X, resulting in a total of 50,000 sample means. According to the Central Limit Theorem, because the sample size $n = 100$ is sufficiently large, the distribution of the sample means should approximate a normal distribution $N\left(\mu, \frac{\sigma^2}{n}\right)$.

Therefore, the close alignment between the kernel density of the sample means (blue curve) and the theoretical normal distribution $f_{\mu,\sigma}$ (red points) in the plot is a direct manifestation of the Central Limit Theorem.