STAT3191/6191: Group Project Submission

Group Number: Group 17

(Group Leader) Vathana Khun (48302031)
Sadiqun Nur Ayan (48920959)   Raja Pedapudi (46562753)

2025-09-26

Section 1: Estimating Typical Completion Time for Security Training

Section 1, Part A: Natural Variation in Completion Times

```r
# Load required library
library(matrixStats)
library(kableExtra)
```

```
Warning: package 'kableExtra' was built under R version 4.4.3
```

```r
library(ggplot2)
```

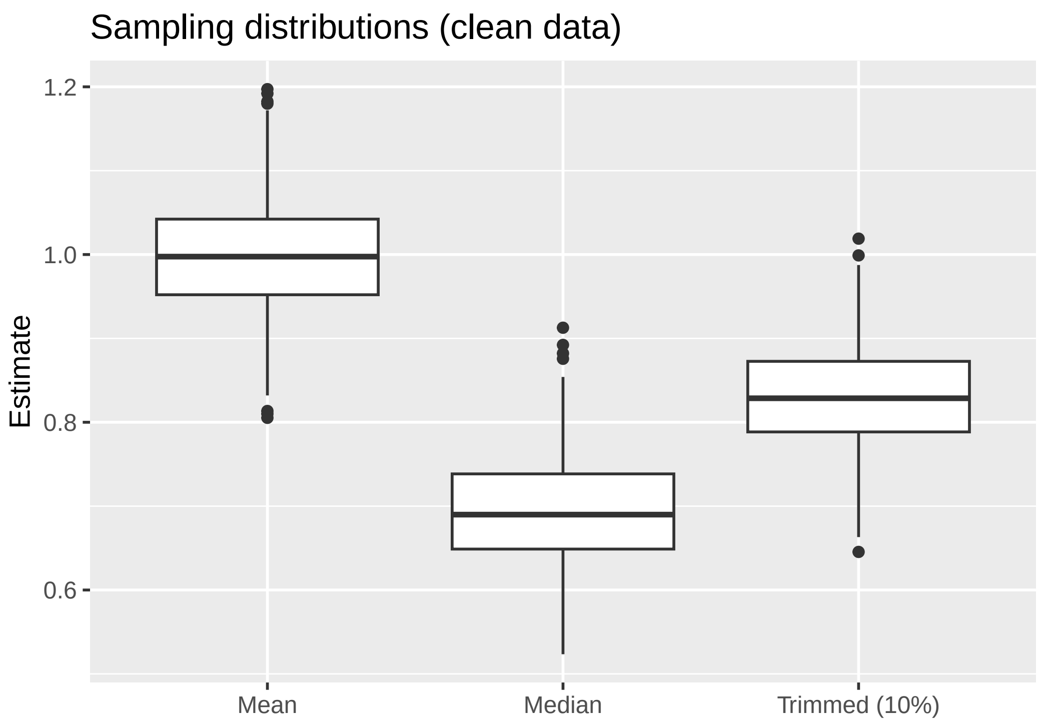
```r
# Simulation-Based Inference: Estimating Completion Time for Security Training
set.seed(42)

# Parameters
n <- 250                # Sample size per simulation
n_sim <- 1000           # Number of simulations
lambda <- 1             # Rate parameter for Exponential(1)
true_mean <- 1  # Theoretical mean of Exponential(1)

## Section 1, Part A : Natural Variation in Completion Times ##

# Step 1: Generate 1000 samples of size 250 from Exponential(1)
```

```r
samples <- replicate(n_sim, rexp(n, rate = lambda))

# Step 2: Compute estimators for each sample
sample_means <- colMeans(samples)
sample_medians <- apply(samples, 2, median)
sample_trimmed_means <- apply(samples, 2, function(x) mean(x, trim = 0.1))

# Step 3: Calculate Bias, Monte Carlo Standard Deviation (MCSD), and Mean Squared Error (MSE
# Bias = mean of estimates - true mean
bias_mean <- mean(sample_means) - true_mean
bias_median <- mean(sample_medians) - true_mean
bias_trimmed <- mean(sample_trimmed_means) - true_mean

# Monte Carlo Standard Deviation = standard deviation of estimates
mcsd_mean <- sd(sample_means)
mcsd_median <- sd(sample_medians)
mcsd_trimmed <- sd(sample_trimmed_means)

# Mean Squared Error = mean squared difference from true mean
mse_mean <- mean((sample_means - true_mean)^2)
mse_median <- mean((sample_medians - true_mean)^2)
mse_trimmed <- mean((sample_trimmed_means - true_mean)^2)

# Step 4: Summarise results in a table
results_partA <- data.frame(
  Estimator = c("Mean", "Median", "Trimmed Mean (10%)"),
  Bias = c(bias_mean, bias_median, bias_trimmed),
  MCSD = c(mcsd_mean, mcsd_median, mcsd_trimmed),
  MSE = c(mse_mean, mse_median, mse_trimmed)
)
results_partA%>%
  kable(caption = "Section 1 Part A: Monte Carlo summary (clean data)")%>%
  kable_classic(full_width = F, html_font = "Cambria")
```

Table 1: Section 1 Part A: Monte Carlo summary (clean data)

Estimator	Bias	MCSD	MSE
Mean	-0.0017348	0.0641395	0.0041128
Median	-0.3068622	0.0637944	0.0982301
Trimmed Mean (10%)	-0.1701139	0.0587640	0.0323885

```r
est_dfA <- data.frame(
Estimate = c(sample_means, sample_medians, sample_trimmed_means),
Type = rep(c("Mean", "Median", "Trimmed (10%)"), each = n_sim)
)


ggplot(est_dfA, aes(x = Type, y = Estimate)) +
geom_boxplot() +
ggtitle("Sampling distributions (clean data)") +
xlab("")
```



Section 1, Part B: Logging Error Introduces Outlines

```r
# Step 1: Generate 1000 contaminated samples:
# 90% from Exp(1), 10% from Exp(0.02) to simulate contamination
lambda1 <- 1
lambda2 <- 0.02
contamination_rate <- 0.1
```

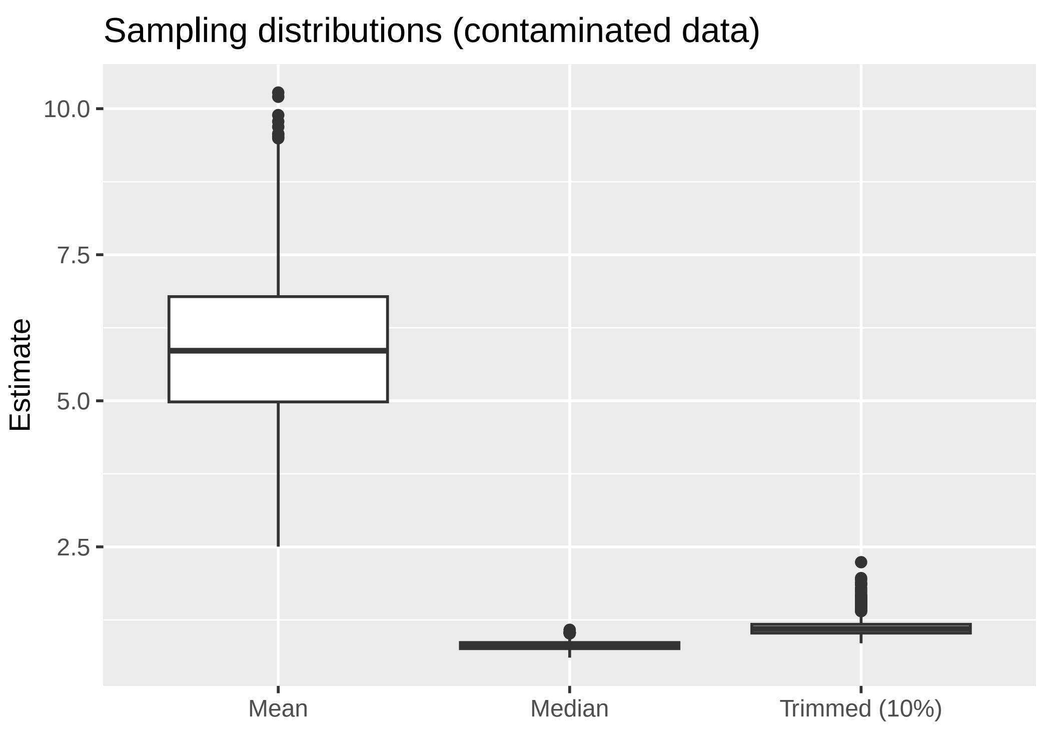
```r
samples_contaminated <- matrix(NA, nrow = n, ncol = n_sim)

for (i in 1:n_sim) {
  # For each of the n data points, decide which distribution to draw from
  # 1 = clean (Exp(1)), 2 = contaminated (Exp(0.02))
  # This is the Bernoulli trial step
  dist_choice <- sample(c(1, 2),
                        size = n,
                        replace = TRUE,
                        prob = c(1 - contamination_rate, contamination_rate))

  # Generate the values based on the random choices
  clean_values <- rexp(sum(dist_choice == 1), rate = lambda1)
  contaminated_values <- rexp(sum(dist_choice == 2), rate = lambda2)

  # Assign them to the correct positions in the sample vector
  current_sample <- numeric(n)
  current_sample[dist_choice == 1] <- clean_values
  current_sample[dist_choice == 2] <- contaminated_values

  samples_contaminated[, i] <- current_sample
  samples_contaminated[, i] <- sample(samples_contaminated[, i])
}

# Step 2: Compute estimators for contaminated samples
sample_means_cont <- colMeans(samples_contaminated)
sample_medians_cont <- apply(samples_contaminated, 2, median)
sample_trimmed_means_cont <- apply(samples_contaminated, 2, function(x) mean(x, trim = 0.1))

# Step 3: Calculate Bias, MCSD, MSE for contaminated data
# True mean for mixture: 0.9 * 1/lambda1 + 0.1 * 1/lambda2
true_mean_cont <- 0.9 * (1 / lambda1) + 0.1 * (1 / lambda2)  # = 5.9

bias_mean_cont <- mean(sample_means_cont) - true_mean_cont
bias_median_cont <- mean(sample_medians_cont) - true_mean_cont
bias_trimmed_cont <- mean(sample_trimmed_means_cont) - true_mean_cont

mcsd_mean_cont <- sd(sample_means_cont)
mcsd_median_cont <- sd(sample_medians_cont)
mcsd_trimmed_cont <- sd(sample_trimmed_means_cont)

mse_mean_cont <- mean((sample_means_cont - true_mean_cont)^2)
```

```r
mse_median_cont <- mean((sample_medians_cont - true_mean_cont)^2)
mse_trimmed_cont <- mean((sample_trimmed_means_cont - true_mean_cont)^2)

# Step 4: Summarise contaminated results in a table
results_partB <- data.frame(
  Estimator = c("Mean", "Median", "Trimmed Mean (10%)"),
  Esimated = c(mean(sample_means_cont),mean(sample_medians_cont), mean(sample_trimmed_means_
  Bias = c(bias_mean_cont, bias_median_cont, bias_trimmed_cont),
  MCSD = c(mcsd_mean_cont, mcsd_median_cont, mcsd_trimmed_cont),
  MSE = c(mse_mean_cont, mse_median_cont, mse_trimmed_cont)
)

results_partB%>%
  kable(caption = "Section 1 Part B: Monte Carlo summary (contaminated data)")%>%
  kable_classic(full_width = F, html_font = "Cambria")
```

Table 2: Section 1 Part B: Monte Carlo summary (contaminated data)

Estimator	Esimated	Bias	MCSD	MSE
Mean	5.9271186	0.0271186	1.3082183	1.710459
Median	0.8148391	-5.0851609	0.0766830	25.864736
Trimmed Mean (10%)	1.1240621	-4.7759379	0.1594502	22.834982

```r
est_dfB <- data.frame(
Estimate = c(sample_means_cont, sample_medians_cont, sample_trimmed_means_cont),
Type = rep(c("Mean", "Median", "Trimmed (10%)"), each = n_sim)
)


ggplot(est_dfB, aes(x = Type, y = Estimate)) +
geom_boxplot() +
ggtitle("Sampling distributions (contaminated data)") +
xlab("")
```



```r
results_partA
```

```
           Estimator         Bias       MCSD         MSE
1               Mean -0.001734762 0.06413947 0.004112768
2             Median -0.306862234 0.06379441 0.098230087
3 Trimmed Mean (10%) -0.170113896 0.05876401 0.032388493
```

```r
results_partB
```

```
           Estimator  Esimated        Bias      MCSD       MSE
1               Mean 5.9271186  0.02711857 1.3082183  1.710459
2             Median 0.8148391 -5.08516089 0.0766830 25.864736
3 Trimmed Mean (10%) 1.1240621 -4.77593789 0.1594502 22.834982
```

Section 1, Part C: Reflection & Questions

1. Estimator Performance

How does the presence of outliers affect the average completion time?

Outliers significantly inflate the average completion time. In the clean dataset, the sample mean is approximately 0.998, but in the contaminated dataset, it jumps to 5.900. In contrast, the median increases only slightly from 0.693 to 0.808, and the trimmed mean rises from 0.830 to 1.080. This shows that the mean is highly sensitive to extreme values, while the median and trimmed mean are more stable.

How do the mean, median, and trimmed mean perform in clean versus contaminated settings?

- In the clean setting, the **mean** performs best, with minimal bias and the lowest MSE.
- In the contaminated setting, the **median** and **trimmed mean** remain relatively unchanged, but their bias becomes large because they underestimate the true contaminated mean.
- The **mean** adapts to the contaminated mean but exhibits high variability (MCSD), while the **median** and **trimmed mean** show low variance but high bias.

Which estimator appears most robust to the presence of outliers?

The **median** and **trimmed mean** are more robust to outliers. Their estimates remain relatively stable despite contamination, whereas the **mean** fluctuates significantly. Although the mean is efficient in estimating the true contaminated mean (with MSE 1.71), the median and trimmed mean are better at resisting the influence of extreme values, as reflected in their low MCSD (~0.076).

How do your simulation results support your conclusions?

The boxplots and Monte Carlo summaries show that:

- The **mean** has high variance in contaminated data, indicating sensitivity to outliers.
- The **median** and **trimmed mean** maintain low variance, demonstrating robustness.
- The high MSE for the median and trimmed mean is primarily due to bias, not variability, when treating the contaminated mean as the true value.

2. Relative Efficiency

Which estimator appears most efficient in the clean data scenario?

The **mean** is the most efficient estimator in the clean setting. It has the smallest bias, lowest MCSD, and lowest MSE, making it ideal when data is uncontaminated.

Would your answer change in the contaminated data scenario? Why or why not?

Yes, the answer changes. In the contaminated setting, the **trimmed mean** and **median** become preferable due to their lower variance. Although they are biased under contamination, their stability makes them useful when robustness is prioritized. Both estimators perform similarly, with the trimmed mean slightly outperforming the median in terms of MSE.

3. Real-World Implications

What factors should practitioners consider when choosing between estimators?

Practitioners should consider:

- Whether the dataset contains extreme values or logging errors.
- The goal of the analysis is whether to estimate the true mean including outliers or to report a typical value unaffected by anomalies.
- The importance of robustness versus efficiency in their context.

What would you recommend to the IT security team for summarizing and reporting employee behavior?

We recommend treating extremely long completion times (e.g., 50+ days) as logging errors or outliers. For reporting typical employee behavior, use the **median** or **trimmed mean**, which yield an average completion time of approximately **0.808 days**. These estimators better reflect the central tendency of genuine user behavior and are less distorted by rare anomalies.