Computational Statistics Assignment 1

Wouter van Elteren | SNR:2004379 | ANR:429898

Group 24

Exercise 1: The Lady Tasting Tea

Consider the famous 'Lady tasting tea' experiment. A group of friends, including the statistician Ronald Fisher, discusses the claim of one of the present ladies who declares that by tasting a cup of tea made with milk she can discriminate whether the milk or the tea infusion was first added to the cup. To test this claim, Ronald Fisher designed an experiment in which he mixed eight cups of tea, four in one way and four in the other, and presented them to the subject for judgment in a random order. The lady has been told in advance of what the test will consist, namely, that she will be asked to taste eight cups and that these shall be four of each kind. History accounts that the lady had 6 cups correct

Question 1

Write pseudo code that shows how to simulate the 'Lady tasting tea' experiment. Assume that the lady guesses (meaning that her choice of what was poured first can be considered to be completely random) and return the number of cups that are correctly identified. Take the following three steps:

1. Define the input your procedure needs.
2. Define the output that your procedure needs to return.
3. Describe, using at most 15 lines, how to derive the output from the given input. Structure your pseudo code using enumeration and/or indentation.

Algorithm 1: Lady Tasting Tea

Input : t: number of cups with tea first
m: number of cups with milk first
Output: c: amount of correct guesses

```
1  cups: array of t number of cups with tea first and m number of cups with milk first
2  guesses ← cups
3  shuffle(cups)
4  shuffle(guesses)
5  c ← 0
6  i ← 0
7  while length(cups) > i do
8  |  if cups[i] equals guesses[i] then
9  |  |  c ← c + 1
10 |  |  i ← i + 1
11 |  else
12 |  |  i ← i + 1
13 |  end
14 end
```

Question 2

Implement the 'Lady tasting tea' simulation experiment in the R language for statistical computing. Do not forget to include a seed to initialize the pseudo random numbergenerator (this can be done with set.seed(x) with x an integer). Report the R code in Appendix.

See Figure A1

Question 3

Report the results of your simulation experiment. What do you conclude about the claim that the lady has no such ability and that one could perform equally well by guessing? Make use of a table that summarizes the estimated probabilities for having 0, ..., 8 cups correctly guessed to present your results. You may consult the APA grammar and style guidelines.

Replicating the simulation 50.000 times shows that there is roughly a 23% chance to guess six cups correctly. The claim that the lady guessed six correctly is therefore more plausible than to believe her that she has a special ability.

Table 1:

Simulation of Lady Tea Tasting

Correct Guesses	Relative Frequency
0	0.01
2	0.23
4	0.52
6	0.23
8	0.01

Note. Simulation executed with the historical number of tea first (4) and milk first (4) cups

For the code, see Figure A2

Exercise 2: A comparison of some variance estimator

In this exercise, we study different estimators of the population variance. Popular estimators are the maximum likelihood estimator and the unbiased estimator. Here are the formulas of these estimators, including a third option:

- Maximum likelihood estimator
- Unbiased estimator
- MinMSE estimator

Assume that the data come from a normal distribution with mean zero and variance equal to five.

Question 1

What is the value of the population parameter estimated by the three variance estimators?

5

Question 2

The function for calculating the variance in R is var(x) with x a numeric vector. Which of the three variance estimators is calculated by this function?

The unbiased estimator

How can you obtain the two other estimates from the resulting estimate?

See Figure A3

Question 3

Compare the performance of the three estimators.

In R, set the seed as follows: set.seed(12052022). Then, generate a single sample of size n = 3 from the given population N(0,5). For this sample, report the estimate of the variance and the difference with respect to the population value in a table. Use the var function to obtain the results for the three types of variance estimators.

Table 2:

Population Variance Estimation via three different Estimators

	MLE	Unb.	MinMSE
Variance Estimate	5.86	8.79	4.40
Difference with Population Variance	0.86	3.79	0.60

For the code, see Figure A4

Repeat the experiment S = 10 000 times; use a for or repeat loop in R. This is, generate ten thousand samples of size n = 3 from the given population and calculate the three variance estimators for each of the samples. For each of the estimators, report the following statistics in a table:

- Average value of the estimate
- Estimated bias, this is the difference between the average value of the estimate and the population value
- Variance of the estimates
- Average of the squared differences between the estimate and the population value.

Repeat the experiment for sample sizes n = 20 and n = 200 and add the results to the table

Table 3:

Performance Comparison Table 1: Maximum Likelihood Estimator

	MLE n=3	MLE n=20	MLE n=200
Avg. Value of Estimate	3.31	4.76	4.97
Estimated bias	1.69	0.24	0.03
Variance of Estimates	11.10	2.37	0.25
Avg. Squared Difference	2.86	0.06	0.00

Note. Where MLE = Maximum Likelihood Estimator and n = the sample size

Performance Comparison Table 2: Unbiased Estimator

	Unb. n=3	Unb. n=20	Unb. n=200
Avg. Value of Estimate	4.96	5.01	5.00
Estimated bias	0.04	0.01	0.00
Variance of Estimates	24.98	2.63	0.25
Avg. Squared Difference	0.00	0.00	0.00

Note. Where Unb. = Unbiased Estimator and n = the sample size

Performance Comparison Table 3: MinMSE Estimator

	MinMSE n=3	MinMSE n=20	MinMSE n=200
Avg. Value of Estimate	2.48	4.53	4.95
Estimated bias	2.52	0.47	0.05
Variance of Estimates	6.25	2.15	0.25
Avg. Squared Difference	6.34	0.22	0.00

Note. Where MinMSE = MinMSE Estimator and n = the sample size

For the code, see Figure A5

Question 4

Discuss the results by briefly answering the following questions:

Which estimator is biased/unbiased, and what is the influence of the sample size on the bias?

> Both the Maximum Likelihood Estimator and the MinMSE Estimator are biased for all samples. The unbiased estimator is biased for sample sizes n=3 and n=20, but unbiased for n=200. An increase in sample size decreases the bias for all estimators.

How do the estimators compare in terms of variance and what is the influence of the sample size on the variance?

> The unbiased estimator's variance is the highest for sample size n=3. MinMSE has the lowest variance at sample size n=3. Most notable for the n=3 sample size is that there is a big difference in variance for the estimators (25.03 vs 11.13 vs 6.26). That difference becomes smaller rather quickly with sample size increase. At sample size n=200, all estimators have practically the same variance.

Which of the three estimators would you recommend to use as an estimator of the variance and why?

> The unbiased estimator, since it has the lowest bias for all sample sizes covered in the simulation.

Appendix A

R-code Chunks

Figure A1: Lady Tasting Tea

```
tasting_tea <- function(n_tcups, n_mcups) {
  #set.seed(32432)

  # Create cups vector
  t_cups <- rep(c("tea"), each = n_tcups)
  m_cups <- rep(c("milk"), each = n_mcups)
  cups <- c(t_cups, m_cups)

  # Create guesses vector
  guesses <- cups

  # Shuffle vectors
  cups_rand <- sample(cups)
  guesses_rand <- sample(guesses)

  # Compare
  comp <- cups_rand == guesses_rand
  comp
  correct_guesses <- sum(comp)
  return(correct_guesses)
}

tasting_tea(4, 4)
```

Figure A2: Lady Tasting Tea Simulation

```
library(rempsyc)
sim = replicate(50000, tasting_tea(4, 4))
result <- as.data.frame(table(sim)/length(sim))
colnames(result) <- c("Correct Guesses", "Relative Frequency")

nice_table(result,
           title = c("Simulation of Lady Tea Tasting"),
           footnote = c("Simulation executed with the historical number of tea first (4) and milk first
```

Figure A3: Variance Estimator Functions

```
# Maximum Likelihood Estimator
mle <- function(x) {
  1/length(x) * sum((x-mean(x))^2)
}

# MinMSE estmator
min_mse <- function(x) {
  1/(length(x)+1) * sum((x-mean(x))^2)
}
```

Figure A4: Population Variance Estimator table

```
set.seed(12052022)

# population variance estimators
mle <- function(x) {
  var(x) * (length(x)-1)/length(x)
}
unb <- function(x) {
  var(x)
}

min_mse <- function(x) {
  1/(length(x)+1) * sum((x-mean(x))^2)
}

# Generate sample
sample <- rnorm(3, mean = 0, sd = sqrt(5))

# calculate
mle.s <- mle(sample)
unb.s <- unb(sample)
min_mse.s <- min_mse(sample)

# Generate df
df <- data.frame(first_column =
                   c("Variance Estimate", "Difference with Population Variance"),
                 second_column =
                   c(mle.s, abs(mle.s-5)),
                  third_column =
                   c(unb.s, abs(unb.s-5)),
                  fourth_column =
                   c(min_mse.s, abs(min_mse.s-5))
                 )
colnames(df) <- c(" ", "MLE", "Unb.", "MinMSE")

# Generate table
nice_table(df,
           title = c("Population Variance Estimation via three different Estimators"))
```

Figure A5: Population Variance Estimator Simulation

```
# Create empty vectors
mle_3.results <- c()
unb_3.results <- c()
min_mse_3.results <- c()

mle_20.results <- c()
unb_20.results <- c()
min_mse_20.results <- c()

mle_200.results <- c()
unb_200.results <- c()
min_mse_200.results <- c()
```

```r
# Sample of 3
i <- 0
while (i < 9999) {
  sample <- rnorm(3, mean = 0, sd = sqrt(5))
  mle_3.results <- append(mle_3.results, mle(sample))
  unb_3.results <- append(unb_3.results, unb(sample))
  min_mse_3.results <- append(min_mse_3.results, min_mse(sample))
  i <- i+1
}

# Sample of 20
i <- 0
while (i < 9999) {
  sample <- rnorm(20, mean = 0, sd = sqrt(5))
  mle_20.results <- append(mle_20.results, mle(sample))
  unb_20.results <- append(unb_20.results, unb(sample))
  min_mse_20.results <- append(min_mse_20.results, min_mse(sample))
  i <- i+1
}

# Sample of 200
i <- 0
while (i < 9999) {
  sample <- rnorm(200, mean = 0, sd = sqrt(5))
  mle_200.results <- append(mle_200.results, mle(sample))
  unb_200.results <- append(unb_200.results, unb(sample))
  min_mse_200.results <- append(min_mse_200.results, min_mse(sample))
  i <- i+1
}

rownames <- c("Avg. Value of Estimate", "Estimated bias", "Variance of Estimates", "Avg. Squared Differ

mle_3.results_v <- (c(mean(mle_3.results),
  abs(mean(mle_3.results)-5),
  var(mle_3.results),
  abs(mean(mle_3.results)-5)^2)
)
unb_3.results_v <- (c(mean(unb_3.results),
  abs(mean(unb_3.results)-5),
  var(unb_3.results),
  abs(mean(unb_3.results)-5)^2)
)
min_mse_3.results_v <- (c(mean(min_mse_3.results),
  abs(mean(min_mse_3.results)-5),
  var(min_mse_3.results),
  abs(mean(min_mse_3.results)-5)^2)
)

mle_20.results_v <- (c(mean(mle_20.results),
  abs(mean(mle_20.results)-5),
  var(mle_20.results),
  abs(mean(mle_20.results)-5)^2)
)
```

```r
unb_20.results_v <- (c(mean(unb_20.results),
  abs(mean(unb_20.results)-5),
  var(unb_20.results),
  abs(mean(unb_20.results)-5)^2)
)
min_mse_20.results_v <- (c(mean(min_mse_20.results),
  abs(mean(min_mse_20.results)-5),
  var(min_mse_20.results),
  abs(mean(min_mse_20.results)-5)^2)
)

mle_200.results_v <- (c(mean(mle_200.results),
  abs(mean(mle_200.results)-5),
  var(mle_200.results),
  abs(mean(mle_200.results)-5)^2)
)
unb_200.results_v <- (c(mean(unb_200.results),
  abs(mean(unb_200.results)-5),
  var(unb_200.results),
  abs(mean(unb_200.results)-5)^2)
)
min_mse_200.results_v <- (c(mean(min_mse_200.results),
  abs(mean(min_mse_200.results)-5),
  var(min_mse_200.results),
  abs(mean(min_mse_200.results)-5)^2)
)


table_mle <- data.frame(rownames, mle_3.results_v, mle_20.results_v, mle_200.results_v)

table_unb <- data.frame(rownames, unb_3.results_v, unb_20.results_v, unb_200.results_v)

table_min_mse <- data.frame(rownames, min_mse_3.results_v, min_mse_20.results_v, min_mse_200.results_v)


colnames(table_mle) <- c(" ", "MLE n=3", "MLE n=20", "MLE n=200")

colnames(table_unb) <- c(" ", "Unb. n=3", "Unb. n=20", "Unb. n=200")

colnames(table_min_mse) <- c(" ", "MinMSE n=3", "MinMSE n=20", "MinMSE n=200")

nice_table(table_mle,
           title = c("Performance Comparison Table 1: Maximum Likelihood Estimator"),
           footnote = c("Where MLE = Maximum Likelihood Estimator and n = the sample size"))

nice_table(table_unb,
           title = c("Performance Comparison Table 2: Unbiased Estimator"),
           footnote = c("Where Unb. = Unbiased Estimator and n = the sample size"))

nice_table(table_min_mse,
           title = c("Performance Comparison Table 3: MinMSE Estimator"),
           footnote = c("Where MinMSE = MinMSE Estimator and n = the sample size"))
```