Tutorial 5 - Inference with Likelihood (Solutions).

Christopher A Oldnall

Welcome to the fifth tutorial of the Regression and Simulation methods module. This is the next script in developing your skills in R, whilst learning about the use of inference for likelihood. Throughout this notebook we will start to pickup concepts of simulating data ahead of part 2 of the course.

Exercise 0

Throughout remember we will need tidyverse, go ahead and do this as your first task.

Your Answer:

```
library(tidyverse)
```

Before we explore some exercises this week, we are going to see an example of generating normal data. To generate a dataset with a normal distribution in R, you can use the rnorm() function. The rnorm() function generates random numbers from a normal distribution with a specified mean and standard deviation.

```
mu <- 5
sigma <- 2
sample_size <- 100
data_example <- rnorm(sample_size, mean = mu, sd = sigma)
```

Exercise 1

Suppose you have a dataset of measurements from a scientific experiment, and you believe that the data follows a normal distribution with an unknown mean (μ) and a known standard deviation ($\sigma = 2$). Using the log-likelihood, derive the log-likelihood function for μ. Then, in R, simulate a dataset of 100 samples from a normal distribution with a specific μ (e.g., $\mu = 5$), and estimate the value of μ using the log-likelihood function. [Hint: There is a package called 'optimize' which can be very useful here.]

Include the R code for simulation, log-likelihood calculation, and estimation. Additionally write up any derivation that you do into this R Markdown file. [Extra hint: It isn't necessary to do the full derivation if you use the sum and 'dnorm' functions.]

```
likelihood <- function(mu) {
  log_lik <- sum(dnorm(data, mean = mu, sd = true_sigma, log = TRUE))
  return(-log_lik)
}

true_mu <- 5
true_sigma <- 2
data <- rnorm(100, mean = true_mu, sd = true_sigma)

mle_result <- optimize(f = likelihood, interval = c(0, 10))
```

```
estimated_mu <- mle_result$minimum
estimated_mu
```

```
## [1] 4.756464
```

Exercise 2

Compare the estimated value with the true μ, and discuss the results in your R Markdown document.

Your Answer:

```
difference <- abs(estimated_mu - mu)
difference
```

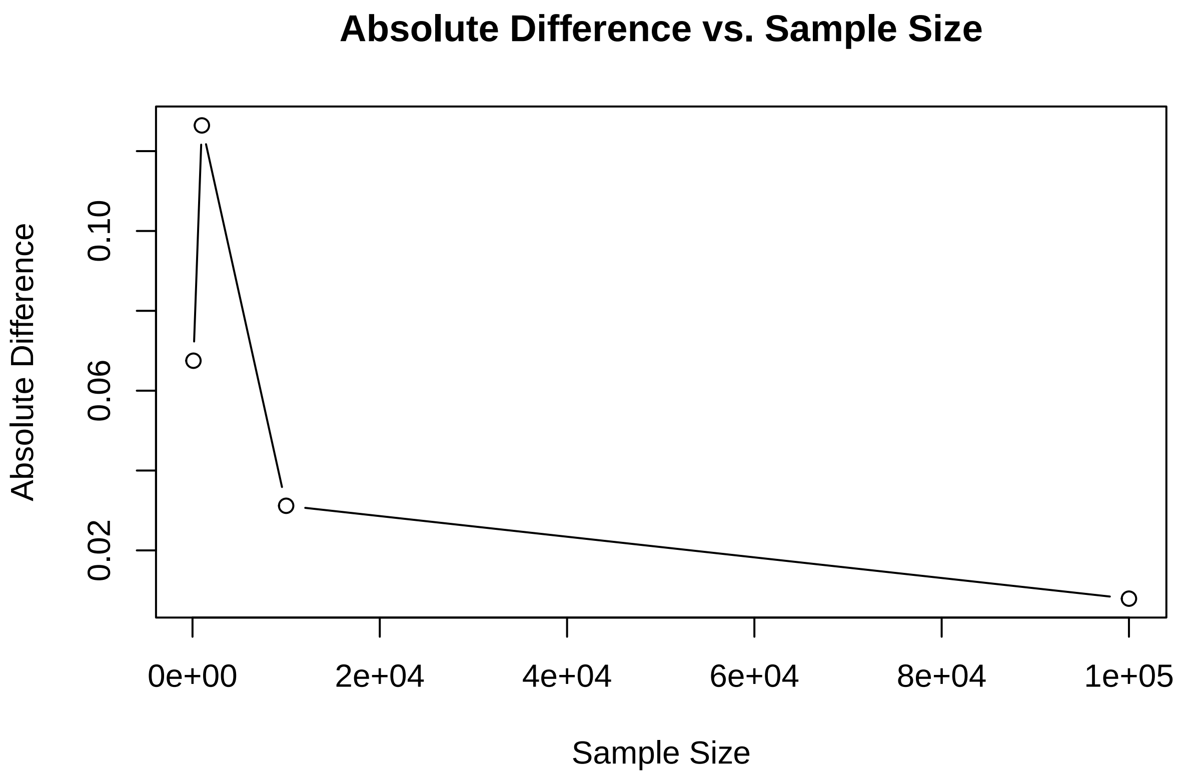
```
## [1] 0.2435364
```

Exercise 3

Consider what happens as you change the sample size and generate a plot which shows how the absolute difference in estimate varies as sample size grows.

```
true_mu <- 5
true_sigma <- 2
sample_sizes <- c(100, 1000, 10000, 100000)
absolute_differences <- numeric(length(sample_sizes))
for (i in 1:length(sample_sizes)) {
  sample_size <- sample_sizes[i]
  data <- rnorm(sample_size, mean = true_mu, sd = true_sigma)
  mle_result <- optimize(f = likelihood, interval = c(0, 10))
  estimated_mu <- mle_result$minimum
  absolute_diff <- abs(true_mu - estimated_mu)
  absolute_differences[i] <- absolute_diff
}

plot(sample_sizes, absolute_differences, type = "b",
     xlab = "Sample Size", ylab = "Absolute Difference",
     main = "Absolute Difference vs. Sample Size")
```



Exercise 4

You are conducting an analysis of a manufacturing process to determine whether a new procedure has an impact on product quality. You have collected data on product defect rates before and after implementing the new process. Using the log-likelihood, construct a confidence interval for the difference in defect rates before and after the process change. Calculate the log-likelihood ratio-based confidence interval and include the R code for its construction. Discuss the interpretation of the confidence interval and what it tells you about the effect of the new process on product quality. Explain the significance level you used in your analysis in your R Markdown document. [Hint: consider that the 'true' value can be the rates before the new process in this situation.]

```r
sample_size_before <- 100
sample_size_after <- 120
true_defect_rate_before <- 1.5
true_defect_rate_after <- 1.0

defects_before <- rpois(sample_size_before, lambda = true_defect_rate_before)
defects_after <- rpois(sample_size_after, lambda = true_defect_rate_after)

log_likelihood_before <- sum(dpois(defects_before, lambda = mean(defects_before), log = TRUE))
log_likelihood_after <- sum(dpois(defects_after, lambda = mean(defects_after), log = TRUE))
log_likelihood_ratio <- 2 * (log_likelihood_after - log_likelihood_before)
critical_value <- qchisq(0.975, df = 1)
confidence_interval <- c(log_likelihood_ratio - critical_value, log_likelihood_ratio + critical_value)
```

This is the end of this tutorial. You can now think about completing 5-1, from the SMSTC exercises this week.