Lab 2 Assignment

Evan O'Toole

Feb 14th, 2025

General Instructions

- Put your name in the author section and fill in the date section above.
- Write R code in the R chunks provided to answer the questions posed.Always print your results to the question.
- You may use any code/materials, including past lab materials, on Brightspace.
- Execute each chunk of code to ensure that your code works properly.
- If you cannot compile all of your code without errors before the end of the lab, comment out the chunks that are not working.
- Save the Rmd file in your personal directory (refer to Lab 0 instructions if needed).
- Knit to PDF (or Word/HTML, but then print those to PDF).
- Upload the PDF file to Brightspace.

The Data

Suppose a city council was interested in improving accessibility to public amenities. They want to count the number of public restrooms available in their downtown area per city block. They sample 12 blocks and count the number of restrooms per city block. If the locations of restrooms are random, then a Poisson(lambda) model should be appropriate for the data.

1. Compute Sample Mean and Create Lambda Sequence

The sample mean of `pdat` is the MLE of lambda given this data. Then create a sequence of lambda values from 1 to 6 in steps of 0.05.

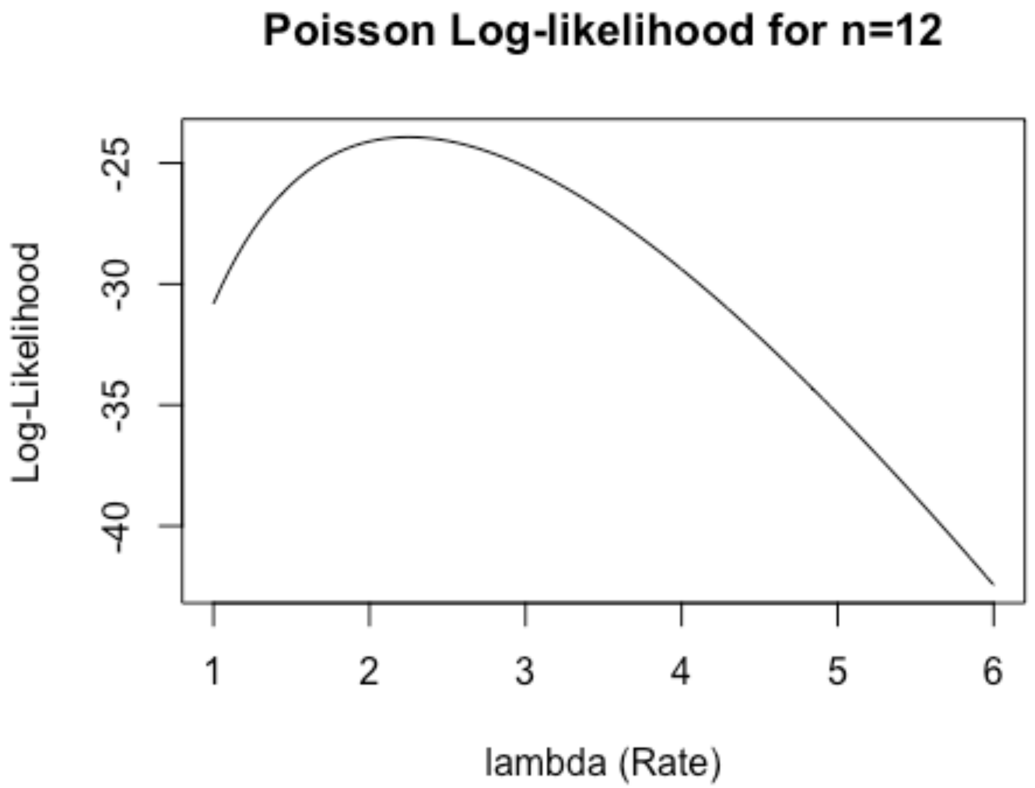
```r
pdat <- c(1, 6, 1, 2, 0, 1, 5, 1, 4, 3, 3, 0)
lambda_mle <- mean(pdat)

# Create a sequence of lambda values from 1 to 6 in steps of 0.05
lambda_seq <- seq(1, 6, 0.05)
```

2. Log-Likelihood Function and Plot

```
# Define the log-likelihood function for the Poisson distribution
log_likelihood <- function(lambda, x) {
  loglike <- 0
  for (i in 1:length(x)) {
    loglike <- loglike + dpois(x[i], lambda, log = TRUE)  # Sum of log of
Poisson PMF
  }
  return(loglike)
}

# Compute the log-likelihood for each value of lambda, given data `pdat`
loglike_vals <- sapply(lambda_seq, function(l) log_likelihood(l, pdat))

# Show the first few values of lambda and corresponding log-likelihoods
head(cbind(lambda_seq, loglike_vals))
```

```
##      lambda_seq loglike_vals
## [1,]       1.00    -30.82146
## [2,]       1.05    -30.10413
## [3,]       1.10    -29.44809
## [4,]       1.15    -28.84789
## [5,]       1.20    -28.29878
## [6,]       1.25    -27.79659
```

```
# Plot the log-likelihood values against lambda
plot(loglike_vals ~ lambda_seq, ylab = 'Log-Likelihood', xlab = 'lambda
(Rate)', type = 'l')

# Add a title with the correct value of n
n <- length(pdat)  # Number of observations
title(paste('Poisson Log-likelihood for n=', n, sep=''))
```


Poisson Log-likelihood for n=12
Log-Likelihood
-25
-30
-35
-40
1
2
3
4
5
6
lambda (Rate)

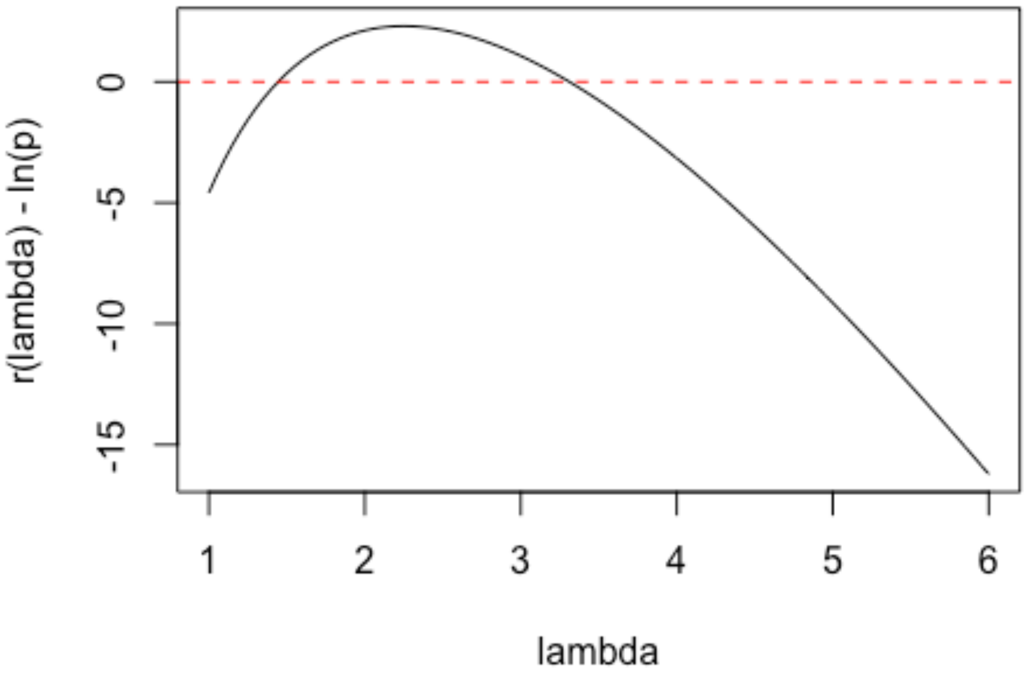
3. Compute MLE of the Log-Likelihood Function using optimize()

```r
# Find the MLE of lambda using optimize()
result <- optimize(log_likelihood, interval = c(0.0001, 15), maximum = TRUE,
x = pdat)
lambda_hat <- result$maximum  # MLE of lambda
lambda_hat  # Display the MLE of lambda

## [1] 2.249999
```

4. Log Relative Likelihood Minus ln(p) Function and Plot

Write a function to compute the log relative likelihood minus ln(p). Plot the function against the lambda sequence you created in part 3, using p=0.1. Your plot must have an appropriate title and labelled axes. Include a horizontal line at y=0, set the color of the line to be red.

```r
# Log relative likelihood function
log_relative_likelihood <- function(lambda, lambda_hat, x) {
  log_likelihood(lambda, x) - log_likelihood(lambda_hat, x)
}

# Log relative likelihood minus ln(p) function
log_relative_likelihood_minus_ln_p <- function(lambda, lambda_hat, x, p) {
  log_relative_likelihood(lambda, lambda_hat, x) - log(p)
}

# Set the threshold for 10% likelihood (p = 0.1)
p <- 0.1

# Plot the log relative likelihood minus ln(p) function against lambda
plot(log_relative_likelihood_minus_ln_p(lambda_seq, lambda_hat, pdat, p) ~
lambda_seq,
     ylab = 'r(lambda) - ln(p)', xlab = 'lambda', type = 'l')

# Add a horizontal line at y = 0 to indicate the likelihood threshold
abline(h = 0, col = "red", lty = 2)

# Add a title
title('Log Relative Likelihood for 10% Likelihood Interval')
```



Log Relative Likelihood for 10% Likelihood Interva

5. Compute 10% Likelihood Interval using uniroot()

```
p <- 0.1

# Use uniroot to find the lower bound of the 10% likelihood interval
lower_interval <- uniroot(log_relative_likelihood_minus_ln_p,
                          c(1, 2), lambda_hat = lambda_hat, x = pdat, p = p)
lower_root <- lower_interval$root  # Lower bound of the 10% likelihood
interval

# Use uniroot to find the upper bound of the 10% likelihood interval
upper_interval <- uniroot(log_relative_likelihood_minus_ln_p,
                          c(3, 15), lambda_hat = lambda_hat, x = pdat, p = p)
upper_root <- upper_interval$root  # Upper bound of the 10% likelihood
interval

# Display the roots
lower_root

## [1] 1.444032

upper_root

## [1] 3.311317
```