DSCI 552 Worksheet 8

Maximum Likelihood Estimation

Contents


```
BEGIN ASSIGNMENT
requirements: requirements.R
generate:
    show_stdout: true
    show_hidden: true
environment: environment.yml
export_cell: false
```

Mechanics

- Please paste the URL to your GitHub repo here: **#INSERT YOUR GITHUB REPO URL HERE**
- Once you finish with the assignment, you must knit the notebook to create a `.pdf` file and pushed everything to your GitHub repo. It is your responsibility to make sure all the figures, texts, equations in the `.pdf` file are rendered properly.

- You must also submit this `.Rmd` AND the rendered `.pdf` files to Gradescope. It is your responsibility to double check your notebook can pass the autograder without errors.
- Follow the MDS general lab instructions.

Heads-up *Once you finish this assignment, make ONE commit to your GitHub repo.*

Setup

Run the cell below to load the libraries needed for this lab, as well as the test file so you can check your answers as you go!

```
library(tidyverse)
library(cowplot)
library(digest)
library(testthat)
```

Maximum Likelihood Estimation with Discrete Variables

During lecture time, we explore maximum likelihood estimation (MLE) for continuous variables. This approach implied the use of continuous probability density functions. But what about those discrete cases? How can we apply MLE?

Suppose we are interested in the **unknown** population mean $\lambda > 0$ related to **the number of car accidents in Vancouver's downtown from 9:00 a.m. to 12:00 p.m.** Therefore, you draw a random sample of size $n = 100$ days which contains the car accident records in Vancouver's downtown from 9:00 a.m. to 12:00 p.m. These records are stored in the tibble `accident_sample`.

```
accident_sample <- tibble(day_observed = as.integer(c(
  6, 5, 2, 1, 5, 6, 4, 3, 4, 0, 2, 4, 5, 5, 4, 2, 4, 3, 2, 3, 4, 4,
  1, 3, 0, 2, 4, 1, 2, 1, 3, 2, 4, 4, 2, 4, 0, 2, 3, 1, 3, 5, 4, 2,
  3, 6, 4, 2, 6, 2, 3, 2, 4, 6, 3, 5, 6, 4, 6, 2, 2, 3, 0, 5, 2, 2,
  2, 1, 5, 3, 0, 4, 3, 2, 3, 2, 4, 2, 6, 6, 2, 4, 6, 4, 1, 0, 3, 7,
  1, 2, 1, 1, 3, 2, 1, 2, 7, 2, 8, 4
)))
```

Exercise 1

rubric={autograde:1}

Note that the observations in `accident_sample` are non-negative integer values that count something. **These counts (i.e., car accidents) happen during a given time frame and geographical unit.** Let Y_i be the number of car accidents in Vancouver's downtown from 9:00 a.m. to 12:00 p.m. in the ith day ($i = 1, \ldots, n$) with an unknown population mean λ. What theoretical distribution can we assume for Y_i to perform MLE for λ?

A. $Y_i \sim \text{Geometric}(\lambda)$

B. $Y_i \sim \text{Bernoulli}(\lambda)$

C. $Y_i \sim \text{Poisson}(\lambda)$

D. $Y_i \sim \text{Binomial}(n, \lambda)$

Assign your answer to an object called `answer1`. Your answer should be a single character surrounded by quotes.

```
BEGIN QUESTION
name: Q1
points:
 - 1
```

```
answer1 <- NULL

# BEGIN SOLUTION
answer1 <- "C"
# END SOLUTION

answer1
```

```
## [1] "C"
```

```
## Test ##
testthat::expect_true(exists("answer1"))
testthat::expect_match(answer1, "a|b|c|d", ignore.case = TRUE)
testthat::expect_equal(digest(tolower(answer1)), "6e7a8c1c098e8817e3df3fd1b21149d1")
```

Exercise 2

rubric={autograde:1}

Since our variable of interest is discrete, visualize the sample distribution of `accident_sample` using a **proper plot**. Ensure that your x-axis is human-readable. Moreover, include a title. Assign your plot to an object called `sample_dist`.

Fill out those parts indicated with `...`, uncomment the corresponding code in the cell below, and run it.

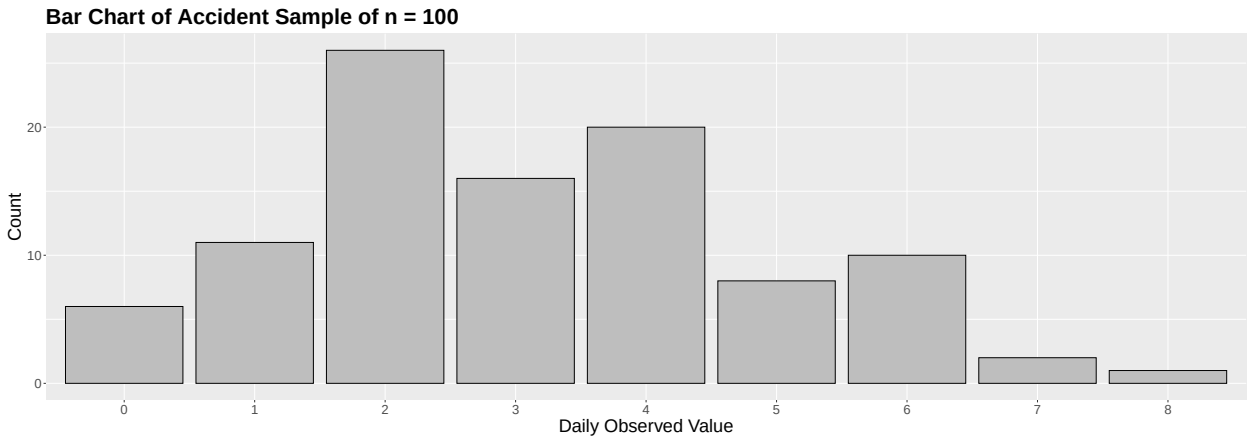
```
BEGIN QUESTION
name: Q2
points:
 - 0.25
 - 0.25
 - 0.25
 - 0.25
```

```
sample_dist <- NULL
# sample_dist <- ggplot(accident_sample) +
#   ...(aes(x = as.factor(...)), fill = "grey", color = "black") +
#   theme(
#     plot.title = element_text(size = 25, face = "bold"),
#     axis.text = element_text(size = 15),
#     axis.title = element_text(size = 21),
#     legend.text = element_text(size = 21),
#     legend.title = element_text(size = 18, face = "bold")
#   ) +
#   ...(...) +
```

```
#   labs(x = ..., y = "Count")

# BEGIN SOLUTION
sample_dist <- ggplot(accident_sample) +
  geom_bar(aes(x = as.factor(day_observed)), fill = "grey", color = "black") +
  theme(
    plot.title = element_text(size = 25, face = "bold"),
    axis.text = element_text(size = 15),
    axis.title = element_text(size = 21),
    legend.text = element_text(size = 21),
    legend.title = element_text(size = 18, face = "bold")
  ) +
  ggtitle("Bar Chart of Accident Sample of n = 100") +
  labs(x = "Daily Observed Value", y = "Count")
# END SOLUTION

sample_dist
```



Exercise 3

rubric={autograde:1}

Based on your answer in **Exercise 1**, what is the correct probability mass function for Y_i?

A. $P_{Y_i}(Y_i = y_i \mid \lambda) = \frac{\exp(-\lambda)\lambda^{y_i}}{y_i!}$ for $y_i = 0, 1, \ldots, \infty$

B. $P_{Y_i}(Y_i = y_i \mid \lambda) = \binom{n}{y_i}\lambda^{y_i}(1-\lambda)^{n-y_i}$ for $y_i = 0, 1, \ldots, n$

C. $P_{Y_i}(Y_i = y_i \mid \lambda) = (1-\lambda)^{y_i}\lambda$ for $y_i = 0, 1, \ldots, \infty$

D. $P_{Y_i}(Y_i = y_i \mid \lambda) = \lambda^{y_i}(1-\lambda)^{1-y_i}$ for $y_i = 0, 1$

Assign your answer to an object called `answer3`. Your answer should be a single character surrounded by quotes.

```
BEGIN QUESTION
name: Q3
points:
 - 1
```

```
answer3 <- NULL

# BEGIN SOLUTION
answer3 <- "A"
# END SOLUTION

answer3
```

```
## [1] "A"
```

```
## Test ##
testthat::expect_true(exists("answer3"))
testthat::expect_match(answer3, "a|b|c|d", ignore.case = TRUE)
testthat::expect_equal(digest(tolower(answer3)), "127a2ec00989b9f7faf671ed470be7f8")
```

Exercise 4

rubric={autograde:1}

Based on your answer in **Exercise 3** and assuming that the n Y_i are *independent and identically distributed*, what is the joint likelihood function of the random sample?

A.

$$l(\lambda \mid y_1, \ldots, y_n) = \lambda^{\sum_{i=1}^{n} y_1}(1-\lambda)^{n-\sum_{i=1}^{n} y_i}$$

B.

$$l(\lambda \mid y_1, \ldots, y_n) = (1-\lambda)^{\sum_{i=1}^{n} y_i}\lambda^{n}$$

C.

$$l(\lambda \mid y_1, \ldots, y_n) = \prod_{i=1}^{n} \binom{n}{y_i} \lambda^{y_i}(1-\lambda)^{n-y_i}$$

D.

$$l(\lambda \mid y_1, \ldots, y_n) = \frac{\exp(-n\lambda)\lambda^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!}$$

```
BEGIN QUESTION
name: Q4
points:
 - 1
```

```
answer4 <- NULL

# BEGIN SOLUTION
answer4 <- "D"
# END SOLUTION

answer4
```

```
## [1] "D"
```

```
## Test ##
testthat::expect_true(exists("answer4"))
testthat::expect_match(answer4, "a|b|c|d", ignore.case = TRUE)
testthat::expect_equal(digest(tolower(answer4)), "d110f00cfb1b248e835137025804a23b")
```

Exercise 5

rubric={autograde:1}

Let us try the empirical MLE approach by computing the likelihood and log-likelihood values on λ range **from `0.25` to `5` by increments of `0.1`**. Your **data frame** should have the following three columns:

- `possible_lambdas`: the λ range.
- `likelihood`: the likelihood values associated with the λ range (you can use a function analogous to `dexp()` as in the lectures notes, **but applicable to your answer in Exercise 1**).
- `log_likelihood`: the logarithmic transformation on the base e of column `likelihood`.

Bind your results to the object `lambda_sequence_values`.

```
BEGIN QUESTION
name: Q5
points:
 - 0
 - 0.33
 - 0.33
 - 0.34
```

```
lambda_sequence_values <- NULL

# BEGIN SOLUTION
lambda_sequence_values <- tibble(
  possible_lambdas = seq(0.25, 5, 0.1),
  likelihood = map_dbl(possible_lambdas, ~
  prod(dpois(accident_sample$day_observed, .))),
  log_likelihood = map_dbl(possible_lambdas, ~
  log(prod(dpois(accident_sample$day_observed, .))))
)
# END SOLUTION

lambda_sequence_values
```

```
## # A tibble: 48 x 3
##    possible_lambdas likelihood log_likelihood
##               <dbl>      <dbl>          <dbl>
##  1             0.25  3.97e-305          -701.
##  2             0.35  9.87e-264          -606.
##  3             0.45  6.51e-234          -537.
##  4             0.55  5.61e-211          -484.
##  5             0.65  1.30e-192          -442.
##  6             0.75  1.67e-177          -407.
##  7             0.85  7.84e-165          -378.
```

```
##  8             0.95  4.69e-154          -353.
##  9             1.05  8.57e-145          -332.
## 10             1.15  9.04e-137          -313.
## # ... with 38 more rows
```

Exercise 6

rubric={autograde:1}

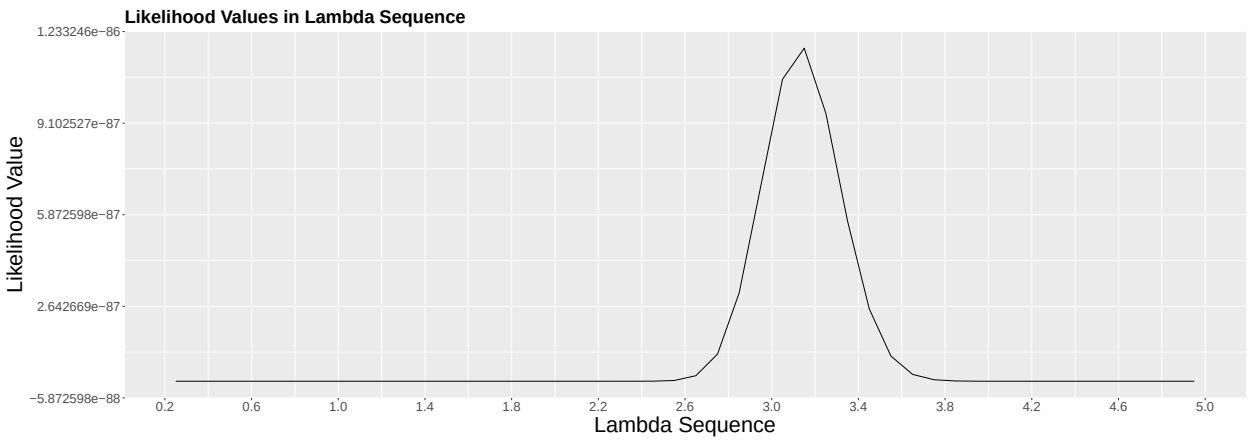
Using your results in `lambda_sequence_values`, create a proper plot for the column `likelihood` values to the column `possible_lambdas`. Ensure that your x and y-axes are human-readable. Moreover, include a title. Assign your plot to an object called `likelihood_plot`.

```
BEGIN QUESTION
name: Q6
points:
 - 0
 - 0.16
 - 0.16
 - 0.17
 - 0.17
 - 0.17
 - 0.17
```

```
likelihood_plot <- NULL

# BEGIN SOLUTION
likelihood_plot <- ggplot(lambda_sequence_values, aes(
  x = possible_lambdas,
  y = likelihood
)) +
  geom_line() +
  theme(
    plot.title = element_text(size = 21, face = "bold"),
    axis.text = element_text(size = 15),
    axis.title = element_text(size = 25)
  ) +
  ggtitle("Likelihood Values in Lambda Sequence") +
  labs(x = "Lambda Sequence", y = "Likelihood Value") +
  scale_x_continuous(breaks = seq(0.2, 5, 0.4))
# END SOLUTION

likelihood_plot
```



Exercise 7

rubric={autograde:1}

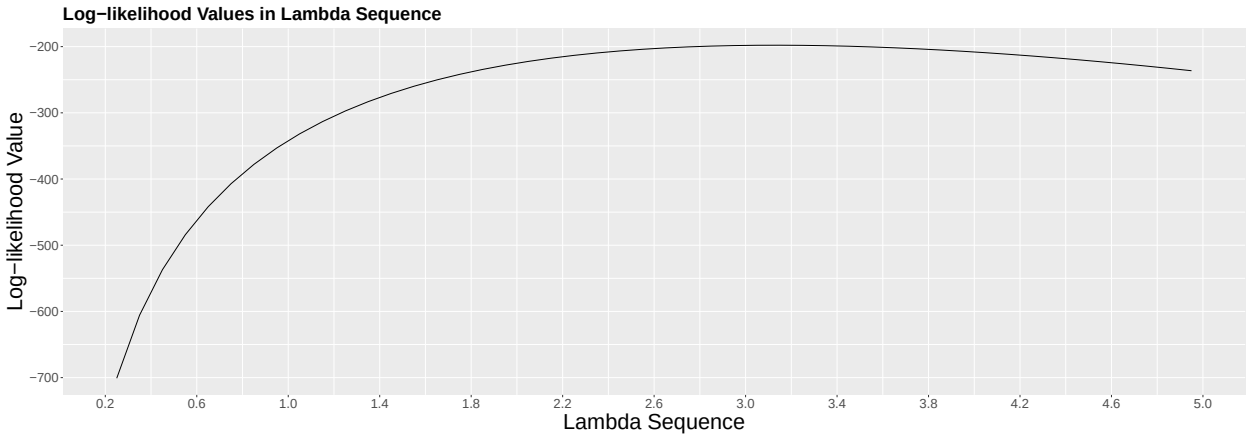
Now, using your results in `lambda_sequence_values`, create another proper plot for the column `log_likelihood` values to the column `possible_lambdas`. Ensure that your x and y-axes are human-readable. Moreover, include a title. Assign your plot to an object called `log_likelihood_plot`.

```
BEGIN QUESTION
name: Q7
points:
 - 0
 - 0.16
 - 0.16
 - 0.17
 - 0.17
 - 0.17
 - 0.17
```

```
log_likelihood_plot <- NULL

# BEGIN SOLUTION
log_likelihood_plot <- ggplot(lambda_sequence_values, aes(
  x = possible_lambdas,
  y = log_likelihood
)) +
  geom_line() +
  theme(
    plot.title = element_text(size = 21, face = "bold"),
    axis.text = element_text(size = 15),
    axis.title = element_text(size = 25)
  ) +
  ggtitle("Log-likelihood Values in Lambda Sequence") +
  labs(x = "Lambda Sequence", y = "Log-likelihood Value") +
  scale_x_continuous(breaks = seq(0.2, 5, 0.4))
# END SOLUTION

log_likelihood_plot
```



Exercise 8

rubric={autograde:1}

Using the correct joint-likelihood function from **Exercise 4**, via the analytical method discussed during lecture time, it can be shown that the maximum likelihood estimator for λ is:

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} Y_i}{n}$$

> **Heads-up:** Note we are using the uppercase notation here for Y_i since "*estimator*" refers to the general expression in function of any random sample. Recall the term "*estimate*" is used when we have an observed sample (i.e., realizations y_i of the n random variables).

8.1

Obtain this analytical maximum likelihood estimate with `accident_sample`. Bind the name `analytical_mle` to this estimate.

```
BEGIN QUESTION
name: Q8.1
points:
 - 0
 - 1
```

```
analytical_mle <- NULL

# BEGIN SOLUTION
analytical_mle <- mean(accident_sample$day_observed)
# END SOLUTION

analytical_mle
```

```
## [1] 3.13
```

8.2

Furthermore, using your results in `lambda_sequence_values`, identify the row in column `possible_lambdas` for which you obtain the maximum `likelihood`/`log_likelohood`. Bind the name `empirical_mle` to this **single** λ value.

```
BEGIN QUESTION
name: Q8.2
points:
 - 0
 - 1
```

```
empirical_mle <- NULL

# BEGIN SOLUTION
maximum_value <- lambda_sequence_values %>%
  arrange(desc(likelihood)) %>%
  slice(1)
maximum_value
```

```
## # A tibble: 1 x 3
##   possible_lambdas likelihood log_likelihood
##              <dbl>      <dbl>          <dbl>
## 1             3.15   1.17e-86          -198.
```

```
empirical_mle <- maximum_value$possible_lambdas
# END SOLUTION

empirical_mle
```

```
## [1] 3.15
```

Exercise 9

rubric={autograde:1}

Uncomment and run the code below:

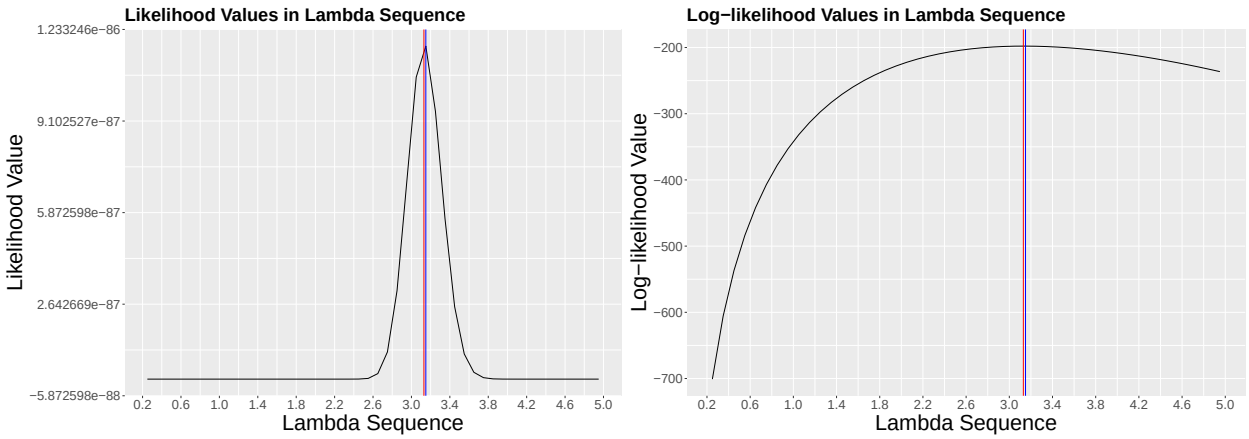
```
# likelihood_plot <- likelihood_plot +
#   geom_vline(xintercept = analytical_mle, colour = "red") +
#   geom_vline(xintercept = empirical_mle, colour = "blue")
#
# log_likelihood_plot <- log_likelihood_plot +
#   geom_vline(xintercept = analytical_mle, colour = "red", show.legend = TRUE) +
#   geom_vline(xintercept = empirical_mle, colour = "blue", show.legend = TRUE)

# BEGIN SOLUTION
likelihood_plot <- likelihood_plot +
  geom_vline(xintercept = analytical_mle, colour = "red") +
  geom_vline(xintercept = empirical_mle, colour = "blue")

log_likelihood_plot <- log_likelihood_plot +
```

```
  geom_vline(xintercept = analytical_mle, colour = "red", show.legend = TRUE) +
  geom_vline(xintercept = empirical_mle, colour = "blue", show.legend = TRUE)
# END SOLUTION

plot_grid(likelihood_plot, log_likelihood_plot)
```



Above, we can see our maximum likelihood estimates: `analytical_mle` in red and `empirical_mle` in blue. Both of them are pretty similar. Now, it is time to remove the curtain! The `accident_sample` was actually taken from a population with $\lambda = 3$. Hence, your estimates should be close to the value they are aiming to estimate.

Nonetheless, as we have discussed beforehand, a single point is not enough to communicate our findings. We also need to control and measure the uncertainty in our estimates since they are computed from random samples. Theoretically speaking, we can compute a confidence interval (CI) for the analytical maximum likelihood estimate in this case. This CI is an asymptotic result (i.e., when $n \to \infty$), and the standard error is obtained via a tool called the Fisher information. With a $(1 - \alpha) \times 100\%$ confidence and the standard normal quantile $z_{1-\alpha/2}$, the asymptotic interval for $\hat{\lambda}$ is given by:

$$\hat{\lambda} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\lambda}}{n}}$$

Compute the lower and upper bounds of a 95% CI using `analytical_mle`, and bind them to the variables `lower_bound_analytical_mle` and `upper_bound_analytical_mle` respectively.

```
BEGIN QUESTION
name: Q9
points:
 - 0
 - 0
 - 0.5
 - 0.5
```

```
lower_bound_analytical_mle <- NULL
upper_bound_analytical_mle <- NULL

# BEGIN SOLUTION
lower_bound_analytical_mle <- analytical_mle - qnorm(1 - 0.05 / 2) *
  sqrt(analytical_mle / 100)
upper_bound_analytical_mle <- analytical_mle + qnorm(1 - 0.05 / 2) *
```

```
  sqrt(analytical_mle / 100)
# END SOLUTION

lower_bound_analytical_mle
```

```
## [1] 2.783247
```

```
upper_bound_analytical_mle
```

```
## [1] 3.476753
```

Finally, we can state with a 95% confidence that λ is between our `lower_bound_analytical_mle` and `upper_bound_analytical_mle`.

Submission

Congratulations! You are done the lab!!! Don't forget to:

- Knit the assignment to generate `.pdf` file and push everything to your Github repo
- Double check all the figures, texts, equations are rendered properly in the `.pdf` file
- Submit the `.Rmd` AND the `.pdf` files to Gradescope, make sure it can pass the autograder without errors.