Purpose: Explore analytic and iterative optimization methods, with applications to maximum likelihood estimation.

Assignment: Each problem involves mathematical and computational parts. For the computational parts, please turn in the simplest/shortest algorithm you can write that accomplishes the task. You may discuss the problems with other students, but all work must be your own. If you consult a book or online resource, cite it.

Your code: For each computational problem, you need to show your code and demonstrate via sample output that it works.

Built-in functions: `R` provides several built-in functions for numerical solving and optimization such as `optim`, `uniroot`, and `nlm`. You may use any of these to check your results, but you must write your algorithms using only basic `R` functionality. If you are not sure whether you can use a certain function, please ask.

Problem 1: Truncated exponential

Consider finding the MLE of n observations from the truncated exponential distribution:

$$f(x) = \begin{cases} \dfrac{\lambda e^{-\lambda x}}{1 - e^{-\lambda a}} & x \in [0, a] \\ 0 & \text{otherwise} \end{cases}$$

where $a > 0$ is fixed and you want to estimate $\lambda > 0$.

a. Write the likelihood of n independent observations $X_1, \ldots, X_n \sim f(x)$, where all the observations are less than a. What is the log-likelihood?

b. Can you find the MLE of λ analytically? Explain why or why not.

c. Consider generating data like this:

```
n = 1000
lam = 2
a = 2
x = rep(NA,n)
for(i in 1:n) {
  repeat {
    xi = rexp(1,lam)
    if(xi<a) break
  }
```

```
    x[i] = xi
  }
```

What distribution do the elements of `x` come from?

d. Implement a Newton's method algorithm for finding the MLE of λ using the simulated data above. Report the estimate of λ that you obtain.

Problem 2: Non-differentiable likelihood

Consider drawing $X_1, \ldots, X_n$ iid from the distribution

$$f(x) = \frac{a}{2} e^{-a|x-b|}$$

where $a > 0$ and $b \in (-\infty, \infty)$.

a. Derive the likelihood and log-likelihood for n observations.

b. Find maximum likelihood estimates of a and b. Show your work and explain your reasoning.

c. Generate 100 random variables from this distribution using $a = 0.1$ and $b = 3$ using the `rexp` and `runif` functions. Do not use any special packages or non-standard functions. [Hint: You can do this in one line of `R` code.]

d. Make a histogram of the random values you generated. Plot the log-likelihood function for $a = 0.1$ as a function of b. What do you see?

e. Implement a routine to find the MLE of (a, b) for the data you generated. Report the estimates.

Extra Credit: Zero-inflated Poisson

The usual distribution used for count data is the Poisson distribution. But sometimes there is something special about the count outcome 0. For example, consider a study in which n people are asked how many times they have been arrested by the police. Many people have been never been arrested, so they report zero. Those who have been arrested report their count, which is greater than zero. This type of data can contain zero counts in excess of what you might expect based on the Poisson assumption. One model for dealing with count data containing excessive zeros is the Zero-inflated Poisson (ZIP) model. Subject i reports

$$Y_i = \begin{cases} 0 & \text{with probability } \alpha + (1-\alpha)e^{-\lambda} \\ y_i > 0 & \text{with probability } (1-\alpha)\frac{\lambda^{y_i}}{y_i!}e^{-\lambda} \end{cases}$$

The unknown parameters are the Poisson mean λ and zero proportion α.

a. Verify that the ZIP model results in a well-defined probability distribution on the non-negative integers.

b. Write the likelihood and log-likelihood of the data $Y_1, \ldots, Y_n$ generated in this way.

c. Write a function `rzip(n,lambda,alpha)` that returns a vector of `n` independent ZIP variates with $\lambda =$`lambda` and $\alpha =$`alpha`. Generate $n = 1000$ observations from the ZIP($\lambda = 5, \alpha = 0.7$) model and plot them in a histogram with bin size 1.

d. Implement an iterative method for finding the MLE of (λ, α) in a ZIP model. Test the algorithm using your simulated data and turn in the results.