STATS 782, Semester One, 2022
Assignment 2

Instructions

1. Please read these instructions carefully.
2. Submit two files to Canvas. (1) Your source code — either a .R or .Rmd file that works when run in R or knitted respectively. (2) A final PDF containing your code and answers. Generally, the marker will only read (2) unless there is a problem.
3. Note the time difference between countries if you're not in New Zealand.
4. Coversheet: please make sure you do one of the following else your assignment will not be marked: (a) Sign the Cover Sheet and combine with your assignment document (pdf or Word) into a single file before submission, OR (b) Type or write for the following at the beginning of your assignment: Your name (as it appears in Canvas), your UPI, and the following statement: "I have read the declaration on the cover sheet and confirm my agreement with it."
5. Please comment on almost all of your output, especially parts that need human interpretation, or marks will be deducted. That is, you need to convince the marker that you understand what the solution is doing.
6. Comment your code if appropriate, e.g., for functions, blocks of code, and key variables.
7. You may occasionally need to look online or in R's help documentation for details (e.g., about functions) that are not found in the coursebook or lectures.
8. Your mark for this assignment will depend on getting the right answer, the elegance/efficiency of your approach, and the tidiness and documentation of your code/report. Marks will be deducted for messy code, etc.

Question 1 [10 marks]

Pascal's triangle looks like this:

```
      1
    1   1
  1   2   1
1   3   3   1
1   4   6   4   1
```

The values in each row, apart from the edges which are always 1, are found by adding the two adjacent numbers from the previous row.

(a) [10 marks] Write a function called `pascal()` that takes an argument n and then produces the first n rows of Pascal's triangle, where $n > 0$. Each row of the triangle should be a numeric vector, and the final result returned by the function should be a list of such vectors.

Question 2 [15 marks]

In this question you will implement a very simple version of 'Approximate Bayesian Computation' — a method for inferring parameters from data. It will seem like a 'guess and check' sort of algorithm that hopefully matches common sense. Suppose that there are some data values $x_1, x_2, ..., x_n$ and that a Cauchy distribution is appropriate:

$$x_1, x_2, ..., x_n \,|\, \mu \sim \text{Cauchy}(\mu, 1). \tag{1}$$

(a) [3 marks] Write a function to generate 7 values from a Cauchy distribution with location parameter ('centre') μ, given as an argument. You can use `rt()` for this, since a Cauchy distribution is the same thing as a t-distribution with `df=1`.

(b) [7 marks] Suppose $\boldsymbol{x} = \{7, 4, 10, 11, 6\}$ is observed. The median of these values is 7, and you are going to generate possible scenarios with median close to 7, to see what μ values are compatible with this observation. Write a function that generates a possible μ value between 0 and 20 uniformly, and simulates a dataset using it, until the median is between 6.99 and 7.01, returning the value of μ that was used when that happened.

(c) [5 marks] Plot a histogram of 100 or more such μ-values.

Question 3 [15 marks]

The 'Bradley-Terry model' is used to analyse and predict sporting events. Suppose each team in a competition has a certain ability, described by a positive real number a_i (for $i \in \{1, 2, ..., N\}$, where

N is the number of teams). If team x plays against team y, the probability that the former wins is given by

$$P(\text{team } x \text{ wins} \,|\, a_x, a_y) = \frac{a_x}{a_x + a_y}, \tag{2}$$

which is team x's fraction of the total ability involved in the match. For a tournament with many matches and many teams, let $\boldsymbol{a} = \{a_1, a_2, ...a_N\}$ be the vector of unknown abilities. The probability of the particular sequence of winners that occurs (the data) is given by a product of terms. Let match i be between team x_i and team y_i, such that team x_i is the winner.

$$P(\text{data} \,|\, \boldsymbol{a}) = \prod_{i=1}^{N_{\text{matches}}} \frac{a_{x_i}}{a_{x_i} + a_{y_i}}. \tag{3}$$

(a) [2 marks] Write R code to read in the data from `matches.csv` and make sure the result is a data frame called `data`.

(b) [7 marks] Write a function called `minus_log_likelihood()`, that takes a vector $\boldsymbol{a} = \{a_2, ..., a_N\}$ as input and returns the negative log likelihood. The ability of team 1 should be assumed to be 1, because only ability ratios matter. If any of the inputs are negative, the function should return `Inf`. Some example uses of the function are given below.

```
> minus_log_likelihood(rep(1, 3))
[1] 12.47665
> minus_log_likelihood(c(2, 3, 4))
[1] 16.60872
> minus_log_likelihood(c(-2, 3, 4))
[1] Inf
```

(c) [4 marks] Use R's `optim()` function to find the maximum likelihood estimate of the $\{a_i\}$ parameters.

(d) [2 marks] Assuming the point estimate from (c) is correct, find the probability that team 3 would beat team 4 in the next match[1].

[1] A more proper way to do this prediction would be to average over all plausible values for the parameters, rather than assuming a single point estimate is true.