MTH208a: Worksheet 3

Simulating Experiments

In this worksheet, we will learn about simulating random experiments in R. Let's start simple.

1. One of the simplest random experiment is tossing a fair coin. `R` can do this for us using command `rbinom()`.

   ```
   # n =  number of coin tosses
   # size = 1
   # prob = probability of success
   rbinom(n = 1, size = 1, prob = 0.5)
   ```

   ```
   [1] 0
   ```

   ```
   # 1 = success (heads), 0 = failure (tails)
   ```

   We can also do multiple coin tosses

   ```
   rbinom(n = 10, size = 1, prob = 0.5)
   ```

   ```
    [1] 0 1 1 0 1 1 1 1 0 0
   ```

   Now that you've understood the commands, write `R` code for the following:

   a. Simulate 1000 fair coin tosses and calculate the proportion of heads.

   b. Simulate 1000 tosses of a coin that has probability of heads equal to 0.30. Calculate the proportion of heads.

2. We can simulate other experiments, like rolling a die:

   ```
   # Rolling a die
   sample(x = 1:6, size = 1)
   ```

   ```
   [1] 3
   ```

   ```
   # Rolling an unfair die
   sample(x = 1:6, size = 1, prob = c(.1, .2, .1, .1, .3, .2))
   ```

   ```
   [1] 2
   ```

```r
# drawing a random number between [a,b]
# n =  number of random numbers
# min = a
# max = b
runif(n = 1, min = 0, max = 1)
```

```
[1] 0.3841037
```

Now that you understand simulating the experiments above, try the following exercises:

a. In a bag, there are 7 balls of 3 different colors: 3 are red, 2 are green, 2 are blue. Write a code to randomly draw a ball from the bag.

b. Consider the following matrix A and let A_1, A_2, A_3 denote the columns of A

$$A = \begin{pmatrix} 3 & 4 & -1 \\ 1 & 5 & 2 \\ -2 & 3 & -2 \end{pmatrix}$$

```r
A <- matrix(c(3, 1, -2, 4, 5, 3, -1, 2, -2), nrow = 3, ncol = 3)
A
```

```
     [,1] [,2] [,3]
[1,]    3    4   -1
[2,]    1    5    2
[3,]   -2    3   -2
```

Write an `R` code to choose column i with probability p_i:

$$p_i = \frac{\|A_i\|}{\sum_{j=1}^{3} \|A_j\|}\,.$$

Here, $\|\cdot\|$ denotes Euclidean norm and can be calculated using the function `norm()`.

c. Suppose I throw a dart anywhere at random on a thread of length 5 cm (assume I will always throw the dart on the thread and never miss the thread – I am very good at throwing darts). Write an `R` code to simulate where the dart lands on the thread.

3. We will try to run a simulation whose answer should be close to `exp(1)`. You will need to use a few new commands in this. Note that, to define a vector of length 1000, you can use command

```r
new <- numeric(length = 1000)
```

Also, we have learned `for()` loops, which can be used to implement a loop when the number of loopings are known. However, when the number loops are unknown and based on some condition, we can use the `while(condition){---}` loop. This runs the loop as long as the `condition` within the `while` command is satisfied.

a. Write an `R` function to count the number of random $[0, 1]$ draws it takes for their sum to exceed 1. The function should have no inputs and should return just one numeric output.

b. Write an `R` program to call the above function 1000 times and store all 1000 outputs in a numeric vector.

c. Return the average of the 1000 outputs. This should be close to `exp(1)`.

4. It's your 25th birthday, and your friends bought you a cake with 25 candles on it. You make a wish and try to blow them out. Every time, you blow out a random number of candles between one and the number that remain.

a. Write an `R` function that `age` as an input and returns the number of attempts it takes to blow out all the candles. You may need the `break` command to write this function or use the `while` loop.

b. Write an `R` program to call the above function 1000 times and store all 1000 outputs in a numeric vector. You now have 1000 simulated candle blowing experiments.

c. How many times, **on average**, do you need to blow at the cake until all the candles are extinguished?

d. Repeat the above for you 30th birthday.