APMA 3100 Probability

Project 2: Monte Carlo Simulation

This project may be done in groups of two or three. Include group partners' names on your submission.

A representative of a high-speed Internet provider calls customers to assess their satisfaction with the service.

- It takes her 6 seconds to turn on a phone and dial a number.
  If the phone number is busy or it goes straight to voicemail, it takes 5 additional seconds to detect that. (She does not leave a voice message.) Otherwise, she will wait up to 20 additional seconds as it rings. It takes 1 second to end a call.
- If the call was unsuccessful, she will try again in the course of several days, until the customer answers or she has tried four times.
- The outcome of each dialing is determined in an identical way: the line is busy or goes straight to voicemail with probability 0.2; the phone rings but the customer is unavailable to answer the call with probability 0.3; otherwise, the customer is available and can answer the call within X seconds, which is an exponential random variable with a mean of 12 seconds. The calling process ends when the customer answers the call, or when four unsuccessful calls have been completed.

Let W denote the total amount of time spent by the representative in trying to contact one customer. (If the customer answers, we do not include the time spent actually talking to them, as this is considered more "productive" time.)
We do not know what type of random variable W is. (It is not any of the "named" types that we have studied; it is unique to this scenario.) We will discover the distribution of W using a Monte Carlo simulation. This will allow us to answer questions such as: What is the probability the representative spends more than 50 seconds on a customer? What are the possible amounts of time the representative could spend calling a customer?

What You Should Do:

Part 1
Due Friday March 19.

Submit a document that contains the following:

1. Draw the detailed diagram/flowchart of the calling process. Include times and probabilities in the diagram. Hints: Consider the calling process from two points of view: that of the representative and that of the customer. The tree diagram may include "if" statements and loops (like a flowchart does).

2. Write code that implements a random number generator as described in the Appendix of this document, using the parameters mentioned there. Your code should produce the sequence of random numbers as specified. (It should not simply use built-in random number generators that the programming language already has available for use.) Include a copy of your code in the document.

3. Test your code by viewing the sequence of random numbers $u_1, u_2, u_3, \ldots$ that is generated. In your document, include the sequence up to u_{55}.

Part 2
Due Friday April 2.

Submit a document that contains the following:

4. Write code that is able to simulate an exponential(λ) random variable X, as explained in the appendix. In your document, provide the mathematical formula you are using, and include a copy of the code.

5. Write a program that simulates the calling process.
   It should include your code from 2 and 4 above, and the simulation should use that code. Again, do not use any built-in random number generators that your programming language may already have.
   a. The program should be able to perform the simulation n times, and thus produce a sample of size n of the random variable W.
   b. The program should then compute estimates of the following statistics:
      i. the mean / expected value of W;
      ii. the first quartile, the median, the third quartile;
      iii. the probabilities of the events
      $$W \leq 15,\ \ W \leq 20,\ \ W \leq 30,\ \ W > 40,\ \ W > w_5,\ \ W > w_6,\ \ W > w_7,$$
      where w_5, w_6, w_7 are values that you choose in order to depict the right tail of the CDF of W as well as possible.
   c. Run your program with $n = 1000$.
      i. Adjust your choices for w_5, w_6, and w_7 based on your results, and re-run. Continue until you are satisfied with your choices.
   d. Save your final results.

Include a copy of your code in the document, as well as your results. Make sure your results are user-friendly, self-explanatory, and easy to read.

6. Analyze your results by answering the following questions:
    a. Compare the mean with the median. What does this comparison suggest about the shape of the probability density function of W?
    b. What seems to be the sample space of W? i.e. all its possible values
    c. Create a graph of the CDF of W using the probabilities that you have estimated, interpolating between them as appropriate.
    d. Could W be a true exponential random variable? Justify your answer.

7. Reflection on your work by answering the following questions. **This section should be answered by each member of the group individually.**
    a. Which step was the most/least challenging?
    b. Which step was the most/least time-consuming?
    c. How certain are you that your results are without error? Answer by subjectively assessing the probability that reflects your judgment.
    d. In what ways could this simulation be extended to be more realistic? What other questions about the calling process could be answered?

Appendix

Simulation System

A Monte Carlo simulation has three main components (Yakowitz, 1977):

1. A **random number generator** — a numerical algorithm which outputs a sequence of real numbers from the interval (0, 1), termed pseudo-random numbers, which the observer unfamiliar with the algorithm, as well as the standard statistical tests, cannot distinguish from a sample of independent realizations (observations) of a uniform random variable.
2. A **random variable generator** — an algorithm that maps any realization of the uniform random variable into a realization of the random variable having a specified cumulative distribution function. (The specification comes from the third component.)
3. A **mathematical model** of the process, system, or problem whose behavior or solution is to be simulated.

Sections A and B below detail algorithms for the first two components; your task will be to implement them on a computer, in whichever way you choose.

Reference: Yakowitz, S.J. (1977). *Computational Probability and Simulation*, Addison-Wesley, Reading, Massachusetts.

A. Random Number Generator

A random number generator is an algorithm that produces a sequence of numbers that mimic a continuous Uniform(0,1) random variable. So all the numbers are between 0 and 1 and appear

"random", even though a computer can't produce true randomness. They are called pseudo-random numbers. There are different ways to accomplish this.

A popular algorithm, known as the *linear congruential random number generator*, specifies the recursive rule:

$$x_i = (ax_{i-1} + c)(modulo\ K)$$

where a, c, and K are parameters, along with a designated starting number x_0, called the seed. [The operation "modulo K" means: divide by K but take the remainder. All programming languages can do this operation.]

Then, the actual pseudo-random numbers are given by:

$$u_i = x_i/K$$

Every sequence of pseudo-random numbers will begin to repeat itself in cycles. In order to maximize the length of the cycles, the parameters must satisfy certain properties. For this project, we are choosing:

starting value (seed)	$x_0 = 1000$
multiplier	$a = 24693$
increment	$c = 1753$
modulus	$K = 2^{15}$

With these values, the pseudo-random number sequence u_1, u_2, u_3 … will repeat after $2^{15} = 32768$ terms.

The first three pseudo-random numbers are $u_1 = 0.6241$, $u_2 = 0.8178$, and $u_3 = 0.2373$ (rounded).

B. Random Variable Generator

How do we use the above random number generator to mimic random events and random variables that are not Uniform(0,1)?

1. If an event is to happen with probability p, we can simulate this by generating a uniform random number u between 0 and 1. If $u < p$, the event happens. Otherwise, it does not.

2. Suppose we want to simulate a discrete random variable X. Then its CDF is a step function. For example, for Binomial(2,0.3), the CDF would be

$$F_X(x) = \begin{cases} 0 & x < 0 \\ 0.49 & 0 \le x < 1 \\ 0.91 & 1 \le x < 2 \\ 1 & 2 \le x \end{cases}$$

We now generate a random number u between 0 and 1. If u is between 0 and 0.49, then $X = 0$. If u is between 0.49 and 0.91, then $X = 1$. And if u is between 0.91 and 1, then $X = 2$. This method will produce the different values of X with exactly the correct probabilities.

The rule can be stated as: X is the smallest value x for which $F(x) \ge u$.

3. Suppose we want to simulate a <u>continuous</u> random variable X. Then its CDF is a continuous function. So we generate a random number u between 0 and 1. Then, as in the discrete case, the value for X is the smallest value x for which $F_X(x) = u$.

   If you can solve this equation algebraically for x, so that it looks like $x = F_X^{-1}(u)$, then you can simply plug u into this and get the value x for X.