3100 Project Fall 2025

Part 1 Due: Wednesday, Oct 15
Part 2 Due: Monday, Nov 3

1. Project Overview

This project will give you hands-on practice building a Monte Carlo simulation from scratch. You will connect probability models to a real-world call-center example, learn how to generate both discrete and continuous random variables, and use your simulated data to estimate key statistics and study the shape of the distribution.

2. Simulation Problem

A representative of a high-speed internet provider calls customers to check their satisfaction. She will call up to four times (in the course of several days) until the customer answers. Each call attempt follows this timeline:

- **Dialing:** 6 seconds to start the call.
- **Outcomes:**
  - With probability **0.2**, the customer is busy and the call goes straight to voicemail (3 seconds).
  - With probability **0.3**, the customer is unavailable and the phone rings for 25 seconds with no answer.
  - With probability **0.5**, the customer is available and answers within X seconds, where X follows an exponential distribution with mean 12 seconds. If $X > 25$, the representative hangs up after 25 seconds and counts it as no answer.
- **Ending the call:** add 1 second to hang up.

The *calling process* stops as soon as the customer answers or after four unsuccessful attempts. Let W denote the total time spent by the representative on calling one customer. Your objective is to create a computer simulation of the scenario and estimate several statistics of W.

3. Simulation System

This section introduces the **tools** you will use to build the simulation. First, you will learn how to create uniform random numbers using a random number generator. Then, you will see how to transform these numbers into realizations of other random variables, both discrete (call outcome) and continuous (answer time).

Random Number Generator: A random number generator is a numerical algorithm that produces numbers between 0 and 1 that behave like independent random samples from a uniform distribution.

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

$$x_i = (a\, x_{i-1} + c)(modulo\ K), \tag{1a}$$

$$u_i = decimal\ representation\ of\ x_i/K, \tag{1b}$$

for $i = 1, 2, 3, \dots$. The meaning of rule (1a) is: multiply x_{i-1} by a and add c; then divide the result by K, and set x_i to the remainder. Every term is an integer: a and K are positive; c and x_{i-1} are non-negative. Rule (1b) states that the i^{th} random number u_i is the quotient x_i/K in the decimal representation, which is a real number between 0 and 1.

Every sequence of pseudo-random numbers eventually cycles. To achieve maximum cycle length, the parameters must satisfy certain proven rules. For this project, we chose

starting value (seed)	$x_0 = 1000,$
multiplier	$a = 24\,693,$
increment	$c = 3517,$
modulus	$K = 2^{17}.$

They yield the cycle of length $K = 2^{17}$. The first three random numbers are $0.4195, 0.0425,$ 0.1274. Show numbers u_{51}, u_{52}, u_{53} in your report.

Random Variable Generator: A random variable generator is an algorithm that maps any realization of the uniform random variable into a realization of the random variable having a specified cumulative distribution function.

Discrete Random Variable

Let X be a discrete random variable with the sample space $\{x : x = 1, \dots, k\}$ and a probability mass function $p(x) = P(X = x)$ for $x = 1, \dots, k$. The corresponding cumulative distribution function F is specified by

$$F(x) = P(X \le x) = \sum_{y \le x} p(y), \qquad x = 1, \dots, k. \tag{2}$$

A given random number u_i generates realization x_i of the random variable X via the rule:

$$x_i = \min\{x : F(x) \ge u_i\}. \tag{3}$$

The rule may be executed by searching sequentially over $x = 1, \dots, k$ until $F(x) \ge u_i$ for the first time.

Continuous Random Variable

Let X be a continuous random variable having a continuous cumulative distribution function F whose inverse F^{-1} exists in a closed-form. That is, $F(x) = P(X \le x)$, and $x = F^{-1}(u)$ for any

$u \in (0, 1)$. A given random number u_i generates realization x_i of the random variable X through evaluation:

$$x_i = F^{-1}(u_i). \tag{4}$$

Additional reading:

- *Inverse Transform Method*
- *Generate Random Variable Using Inverse Transform Method in Python*

4. Tasks:

The project has two parts. Part 1 focuses on building the tools you needed. Part 2 use those tools to run the full simulation and analyze the results. You may use any coding language or environment (e.g. Python, R, MATLAB, etc)

Part 1: Building the Tools

1. Use the linear congruential generator (LCG) with parameters provided to compute at least 60 random numbers. The first three random numbers should match $0.4195, 0.0425, 0.1274$. Show numbers u_{51}, u_{52}, u_{53} in your report.

2. The simulation involves two types of random variables:
   - Discrete (call outcome): voice mail, unavailable, available.
   - Continuous (answer time when available): exponential with mean 12 second.

   (a) In your own words, explain how a random variable generator works for each case.
   (b) Generate 3 sample values for the call outcome and 3 sample values for the exponential answer time using the random numbers u_{51}, u_{52}, u_{53} from step 1.

3. **Formulate** a model of the calling process and represent it with a tree diagram or flowchart. Your diagram should show the branching of outcomes at each call attempt, with associated probabilities and times. Clearly list and define every symbol or variable you introduce. Make sure your diagram is neat and readable.

Part 2: Running the Simulation

4. **Generate** 1000 independent realizations (each starting with a different random number) of the calling process and thereby outputting a sample of size 1000 of random variable W. In your report, briefly describe your algorithm: explain how the random numbers were used at each step of the process (to choose call outcomes and to generate answer times) and how these choices combine to produce a value of W.

5. **Estimate** from the generated sample of W: (i) the mean; (ii) the first quartile, the median, the third quartile; (iii) the probabilities of events

   $$W \le 15,\ W \le 20,\ W \le 30,\ W > 40,\ W > w_5,\ W > w_6,\ W > w_7,$$

   where $w_5,\ w_6,\ w_7$ are the values you choose in order to depict well the right tail of the cumulative distribution function of W.

6. **Analyze** the results and draw conclusions. In particular:
    (a) Compare the mean with the median. What does this comparison suggest about the shape of the probability density function of W?
    (b) Determine the sample space of W.
    (c) Graph the cumulative distribution function of W using the probabilities estimated in Step 5, and interpolating between them whenever appropriate. Could W be an exponential random variable? Explain.

5. Report Submission Guidelines

- You may work with one or two partners. List all team member's names on the cover page.
- Begin with a short introduction that explains the project in your own words.
- Organize the report into 6 sections, parallel to the above steps.
- Part 1 (include introduction) is due *Wed, Oct 15*. Part 2 is due *Mon, Nov 3*.
- Draw figures professionally: to scale, with labels on axes, and captions.
- Do not submit your computer code but be prepared to provide them if asked.
- You may discuss the project with the instructor, TAs, and classmates, but all work must be your own. The Honor Pledge must be printed on a cover page and signed by every member of the team.