APMA 3100
Spring 2018

MINI-PROJECT 1

Due: Friday, 30 March

1. OBJECTIVE

This project will introduce you to the fundamentals of *Monte-Carlo simulation* — a technique of artificially creating samples of random variables which can be large, and which can be used to simulate the behavior of random processes, as well as to solve problems intractable analytically (e.g., those involving multivariate integration or differential equation systems).

In a nutshell, it is a technique for *numerical experimentation* on a computer. While many software packages offer various simulation programs, we shall not use them here. For our objective is to train not as users, but as engineers — the designers and developers of new or specialized systems. And to succeed in such creative tasks, we must understand the fundamentals of system simulation.

2. SIMULATION SYSTEM

A Monte-Carlo simulation system 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 3 and 4 detail algorithms for the first two components; your task is to implement them on a computer, whichever way you choose. Section 5 presents a problem; your task is to model it using constructs of probability theory, and then to solve it via Monte-Carlo simulation.

Reference

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

3. RANDOM NUMBER GENERATOR

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 $= 1, 2, 3, \dots$. 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	$a = 7893$,
increment	$c = 3517$,
modulus	$K = 2^{13}$.

They yield the cycle of length $K = 2^{13}$. The first three random numbers are $0.9303, 0.2703, 0.6205$. Show numbers u_{51}, u_{52}, u_{53} in your report.

4. RANDOM VARIABLE GENERATOR

4.1 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.

4.2 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, $F(x) = P(X \leq x)$, 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:

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

5. SIMULATION PROBLEM

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; then 3 additional seconds to detect a busy signal, or 25 additional seconds to wait for 5 rings and conclude that no one will answer; and 1 second to end a call. After an unsuccessful call, she re-dials (in the course of several days) until the customer answers or she has dialed four times. The outcome of each dialing is determined in an identical way: the customer being called is using the line with probability 0.2; or is unavailable to answer the call with probability 0.3; or is available and can answer the call within X seconds, which is a continuous random variable with the mean of 12 seconds and the exponential distribution.

Let W denote the total time spent by the representative on calling one customer. Your objective is to estimate several statistics of W. Toward this end, perform the following.

1. **Formulate a model** of the calling process.
    - 1.1 Define notation for all the elements of the model.
    - 1.2 Write the expression for the cumulative distribution function of X, and derive the expression for its inverse.
    - 1.3 Draw the tree diagram of the calling process.

2. **Collect data**. Perform an ad-hoc experiment using your phone and playing the roles of a representative and of a customer. Measure the times needed to perform the various tasks; repeat the measurements several times; report your average values that would replace the values of $6, 3, 25, 1, 12$ seconds specified in the statement of the problem. [However, use the specified values in the simulations.]

3. **Design a Monte-Carlo simulation algorithm**. A realization of the calling process should end when the customer answers the call, or when four unsuccessful calls have been completed; in either case, a realization of W can be calculated. The algorithm should be capable of generating n independent realizations (each starting with a different random number) and thereby outputting a sample of size n of random variable W. The way of implementing the algorithm on a computer is up to you.

4. **Simulate** the calling process $n = 500$ times.

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

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

where $w_5,\ w_6,\ w_7$ are 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:
   - 6.1 Compare the mean with the median. What does this comparison suggest about the shape of the probability density function of W?
   - 6.2 Determine the sample space of W.
   - 6.3 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? Justify your answer.

7. **Comment** on the approach. In particular, answer these questions: (i) Which of the steps was the most (the least) challenging? (ii) Which of the steps was the most (the least) time consuming? (iii) What is your probability that the results of the simulation you report are sans errors?

6. REPORT

Contents

1. Organize the report into 7 sections, parallel to the above steps.
2. Explain your work, summarize the results, answer questions.
3. Do not submit computer code or raw data, but archive them.

Format

4. Use standard letter-size, white paper; blank or lined.
5. Write legibly in longhand or type; write concisely.
6. Draw figures professionally: to scale, with labels on axes, and captions.
7. Paginate and staple.

Help and Honor Pledge

8. You may discuss the project with the instructors, teaching assistants, and classmates, but all work must be your own. The Honor Pledge must be signed by every member of the team.