Lesson 10. Generating Randomness

1 Random numbers

- A **random number** is a random observation from a Uniform$[0,1]$ distribution
- How do we tell a computer to generate random numbers: i.e., sample independent values from the Uniform$[0,1]$ distribution?
- It is very difficult to get a computer to do something randomly
  - A computer, by design, follows its instructions blindly, and is therefore completely predictable
  - A computer that doesn't do this is broken!
- One approach: **pseudo-random number generators**

1.1 Pseudo-random number generators (PRNGs)

- "Psuedo" means having a deceptive resemblance
- PRNGs are (deterministic) <u>algorithms</u> that use mathematical formulas or precalculated tables to produce sequences of numbers that <u>appear</u> random

1.2 Desirable properties of a PRNG

Efficient. Can produce many numbers in a short amount of time

Deterministic. A given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known

- Useful for comparing different systems

Long cycle. If the PRNG is periodic (generates a sequence that eventually repeats itself), the cycle length should be sufficiently long

- Modern PRNGs have a period so long that it can be ignored for most practical purposes

Pass statistical tests for uniformity and independence. Most importantly: these numbers should <u>not</u> be statistically differentiable from a sequence of truly independently sampled values from the Uniform$[0,1]$ distribution

- We can test for uniformity using goodness-of-fit tests (e.g. Kolmogorov-Smirnov)

1.3 The linear congruential generator

- Produces sequence of integers $X_1, X_2, \dots$ using the following recursion:

  - The initial value X_0 is called the
  - The minimum possible value of $X_1, X_2, \dots$ is
  - The maximum possible value of $X_1, X_2, \dots$ is

- The **stream**, or the sequence of generated pseudo-random numbers is

- The modulus is often chosen to be a power of 2: binary computations are fast on a computer
- If $c = 0$, this is a **multiplicative congruential generator**
- If $c \neq 0$, this is a **mixed congruential generator**
- The **period** of a linear congruential generator (LCG) is the smallest integer n such that $X_0 = X_{n-1}$ (how many iterations of the LCG take place before the sequence starts to repeat itself)
- An LCG has **full period** if its period is m (Why?)
- **Theorem.** An LCG has full period if and only if:
  - (i) c and m are relatively prime: the only positive integer that divides both c and m is 1
  - (ii) If m is a multiple of 4, then $a - 1$ is a multiple of 4
  - (iii) If p is a prime number dividing m, then $a - 1$ is a multiple of p

Example 1. Consider the LCG with modulus 16, increment 11, and multiplier 9. Confirm that this LCG has full period.

2 Random variates

- A **random variate** is a particular outcome of a random variable
- In other words, a random variate is a sample from a probability distribution
- How can we generate random variates?
- One method: **the inverse transform method**
- Big picture:
  - We want to generate random variates of X with cdf F_X
  - We have a pseudo-random number generator
    - i.e. pseudo-random numbers, or samples from $U \sim \text{Uniform}[0,1]$
  - We will transform these pseudo-random numbers into random variates of X
- How do we do this transformation?

3 The discrete case

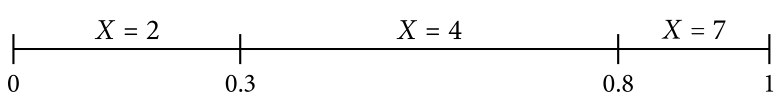
Example 2. Consider the random variable X with cdf

$$F_X(a) = \begin{cases} 0 & \text{if } a < 2, \\ 0.3 & \text{if } 2 \le a < 4, \\ 0.8 & \text{if } 4 \le a < 7, \\ 1 & \text{if } a \ge 7. \end{cases}$$

a. What values can X take?

b. What is $p_X(4) = \Pr\{X = 4\}$? *Hint.* $\Pr\{a < X \le b\} = \Pr\{X \le b\} - \Pr\{X \le a\}$.

c. Let's split the interval $[0,1]$ into subintervals according to the cdf F_X, and assign values of X to each subinterval like so:



Suppose you generate a lot of random numbers (samples from $\text{Uniform}[0,1]$). What percentage of them will belong to the interval corresponding to $X = 4$?

More generally...

- Let X be a discrete random variable taking values $a_1 < a_2 < a_3 < \dots$
- Define $a_0 = -\infty$ so that $F_X(a_0) = 0$
- A **random variate generator** for X is

$$X = a_i \quad \text{if } F_X(a_{i-1}) < U \leq F_X(a_i) \qquad \text{for } i = 1, 2, \dots \quad \text{where } U \sim \text{Uniform}[0,1]$$

- So, to generate a random variate of X with cdf F_X:

1: Generate pseudo-random number u (i.e. sample from $U \sim \text{Uniform}[0,1]$)
2: Find a_i such that $F_X(a_{i-1}) < u \leq F_X(a_i)$
3: Set $x \leftarrow a_i$
4: Output x, random variate of X

4 The continuous case

- Now suppose X is a continuous random variable
- We can't assign values of X to intervals of $[0,1]$ because X takes on a continuum of values!
- New, related idea: set $X = F_X^{-1}(U)$ where $U \sim \text{Uniform}[0,1]$
- Why does this transformation work?

$$\Pr\{X \leq a\} = \Pr\left\{F_X^{-1}(U) \leq a\right\} = \Pr\left\{F_X\big(F_X^{-1}(U)\big) \leq F_X(a)\right\} = \Pr\left\{U \leq F_X(a)\right\} = F_X(a)$$

- Therefore, $X = F_X^{-1}(U)$ is a **random variate generator** for X
- To generate a random variate of X with cdf F_X:

1: Generate pseudo-random number u
2: Set $x \leftarrow F_X^{-1}(u)$
3: Output x, random variate of X

Example 3.

Let X be a continuous random variable with cdf

$$F_X(a) = \begin{cases} 0 & \text{if } a < 0 \\ a^2 & \text{if } 0 \leq a < 1 \\ 1 & \text{if } a \geq 1 \end{cases}$$

Find a random variate generator for X.