Random Numbers & Simulation Basics

[Simulation Book Ch. 3-4 (see also Models § 11.1 - 11.4 approx)]

ch.3
- Building block of a simulation study is the ability to generate random numbers.
  - → default definition (Ross's books) : value of a uniform RV on (0,1)
  - → more generally, generate a random observation (value, variate) from a given prob. dist'n
- Use a computer to generate pseudorandom numbers.
  - → appear random, but are determined by an initial seed value so NOT truly random.
  - → important because these sequences are fast to generate, reproducible

[Read more about RNG & PRNG in Ross's books, if interested].

Multiplicative Congruential Method

x_0 = initial value (seed)

$x_n = a x_{n-1}$ modulo m , $n \geq 1$ recursively compute

remainder when $a x_{n-1}$ is divided by m

a & m are given positive integers

(Simple e.g. 10 mod 8 = 2)

In R: 10 %% 8

Thus, each x_n is either $0, 1, 2, \ldots, m-1$ &

$$U_n = x_n/m \in (0,1)$$

↗ pseudo random number

<u>Note</u>: Choose a & m s.t.

- For any seed, the sequence $U_1, U_2, \ldots, U_n$ appears to be i.i.d. $U(0,1)$ RVs
- Need a large # of values before x_n's will repeat themselves
- Efficient!

<u>Examples</u>: on 32 bit machine:

$a = 7^5 = 16{,}807$ & $m = 2^{31} - 1$ ↙ Source: Ross Sim.

≈ 2 billion

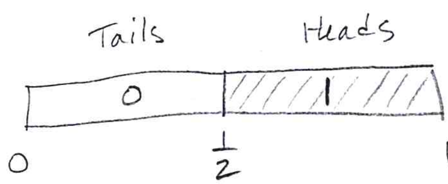
(also, $a = 23$ & $m = 10^8 + 1$
source: Nielsen's notes)

Now, we'll assume R is using good PRNGs & move on!

* Go to Inverse Transform Method

Ch. 4 Generating Discrete Random Variables

<u>Example 1</u>: Simulate a coin flip



$X = 0, 1$

$p = P(X=k) = \frac{1}{2}$ ← success prob. fair coin

$$\boxed{X := U > p}$$

(equiv. to $X := U \leq p$)

$$X = \begin{cases} 0 & \text{if } U \leq p \\ 1 & \text{if } U > p \end{cases}$$

where $U \sim U(0,1)$

<u>Problem</u>: Generate random observation from a given prob. distribution.

<u>Classical Solution</u>:

1. Generate uniform random number U
2. <u>Transform</u> U to get X

   Q. How?

Use the <u>Inverse Transform Method</u>:

Let $U \sim U(0,1)$. For any <u>continuous</u> distribution function F, the random variable X defined by

CDF

$$X = F^{-1}(U)$$

has distribution F.

i.e. $F^{-1}(u)$ is defined to be the value x s.t. $F(x) = u$.

<u>Quick Proof</u>: Let F_X denote the CDF of $X = F^{-1}(U)$.

Then

$$F_X(x) = P(X \le x) = P(F^{-1}(U) \le x).$$

Since F is a CDF, $a \le b \Rightarrow F(a) \le F(b)$.

it is monotone increasing

Thus,

$$F_X(x) = P(F(F^{-1}(U)) \le F(x))$$

$$= P(U \le F(x)) \quad \text{since} \quad F(F^{-1}(U)) = U$$

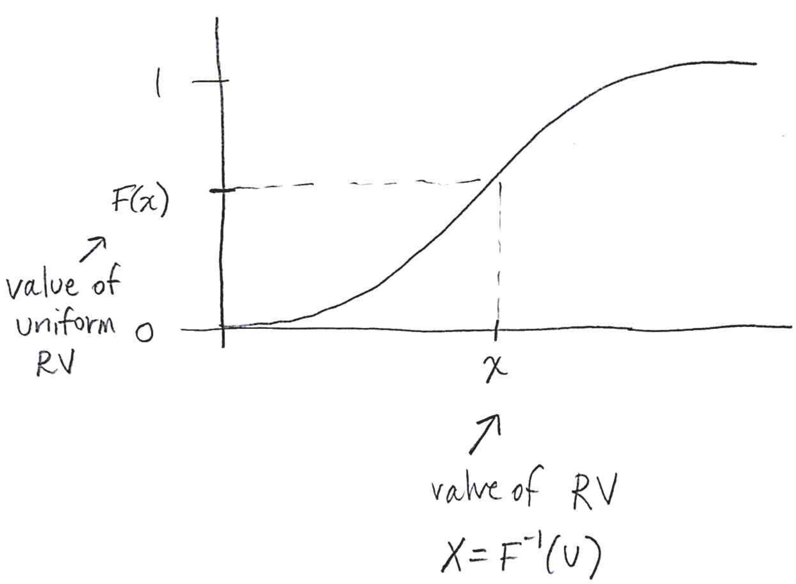
$$= F(x) \quad \text{since } U \text{ is a uniform } (0,1) \text{ RV.}$$

Main Idea

* Therefore,

we can generate a RV X from the continuous distribution function F by generating a random number U & "transforming" it into X by setting $X = F^{-1}(U)$.

CDF: $F(x) = P(X \le x)$



Y-axis of CDF is the interval $[0,1]$

draw uniform RV & transform to get X

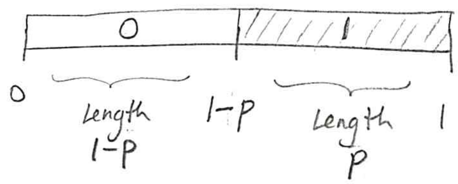
* Same Idea works for discrete distributions

Lecture 5 | Stat 753 | 9/12/17 | 2

More generally : <u>Bernoulli Trial</u>

$$X = \begin{cases} 0 & \text{if } 0 \le U \le 1-p \\ 1 & \text{if } 1-p < U \le 1 \end{cases}$$

separate interval into 2 compartment



Can rewrite this as

$$X = \begin{cases} 0 & \text{if } U \le 1-p \\ 1 & \text{if } U > 1-p \end{cases} \iff \boxed{X = \begin{cases} 1 & \text{if } U \le p \\ 0 & \text{if } U > p \end{cases}}$$

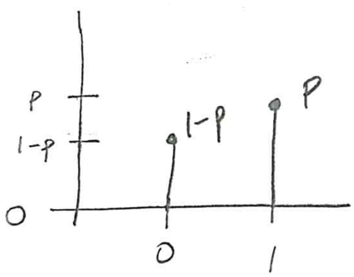
standard notation

length p

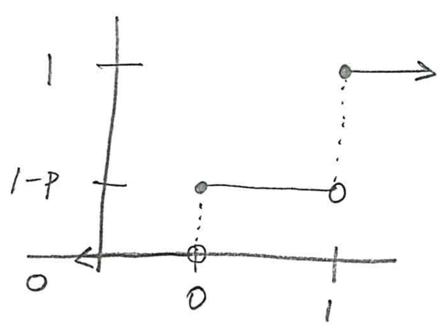
length 1-p

Q. Why are these equivalent? A: compartments have right length

PDF



CDF



R code to simulate:
- 1 toss of coin
- series of n tosses & verify LLN

We'll do this later today!

Example 2 : Simulate a binomial RV

Recall : $X \sim \text{binomial}(n, p)$

(toss a coin n times & count # of H's)

Generate n independent Bernoulli(p) trials & add them up.

$$X = X_1 + X_2 + \cdots + X_n \quad \text{where } X_i \sim \text{Bernoulli}(p)$$

In R: $X = \text{sum}(U < p)$ where U is an n-dim vector of iid $U(0,1)$ random numbers

Generate Arbitrary Discrete Distributions

Suppose we want to generate the value of a discrete RV X having PMF (mass function)

$$P(X = x_j) = p_j \quad \text{for } j = 0, 1, \ldots, \quad \sum_j p_j = 1.$$

First generate a uniform random number $U \sim \text{Uniform}(0,1)$ and then set

$$X = \begin{cases} x_0 & \text{if } U < p_0 \\ x_1 & \text{if } p_0 \le U < p_0 + p_1 \\ x_2 & \text{if } p_0 + p_1 \le U < p_0 + p_1 + p_2 \\ \vdots & \\ x_j & \text{if } \sum_{i=0}^{j-1} p_i \le U < \sum_{i=0}^{j} p_i \\ \vdots & \end{cases}$$

In other words, $X = x_j$ if $F(x_{j-1}) \le U < F(x_j)$

assuming the x_i's are ordered $x_0 < x_1 < x_2 < \cdots$ s.t. $F(x_k) = \sum_{i=0}^{k} p_i$

where $F(x)$ is the desired CDF.

details

Since for $0 < a < b < 1$, $P(a \le U < b) = b - a$,

it follows that

$$P(X = x_j) = P\left(\sum_{i=0}^{j-1} p_i \le U < \sum_{i=0}^{j} p_i\right) = \overbrace{p_0 + p_1 + \cdots + p_{j-1} + p_j}^{b} \underbrace{- p_0 - p_1 - \cdots - p_{j-1}}_{a}$$

$$= p_j$$

$\Rightarrow$ X has the desired distribution.

Write this as an algorithm:

Generate random number $U \sim U(0,1)$.

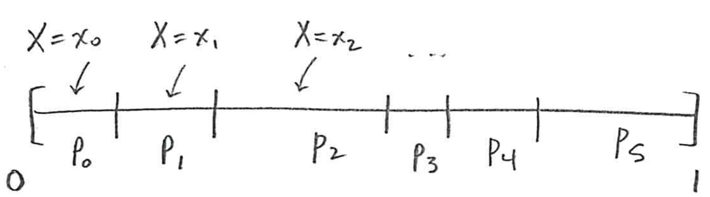
If $U < p_0$, set $X = x_0$ and STOP.

If $U < p_0 + p_1$, set $X = x_1$ and STOP.

If $U < p_0 + p_1 + p_2$, set $X = x_2$ and STOP.

$\vdots$

Divide the interval $[0,1]$ into subintervals s.t. the j^{th} subinterval has length p_j



example with 6 subintervals.

Example: Give an algorithm to simulate the value of a RV X such that

$$P(X=1) = 0.35 = p_0$$

$$P(X=2) = 0.15 = p_1$$

$$P(X=3) = 0.4 = p_2$$

$$P(X=4) = 0.1 = p_3$$

Soln: Divide $[0,1]$ into the following subintervals

$$A_0 = [0, 0.35)$$

$$A_1 = [0.35, \underbrace{0.5}_{0.35+0.15})$$

$$A_2 = [0.5, 0.9)$$

$$A_3 = [0.9, 1)$$

Note that subinterval A_i has length p_i.

Generate $U \sim U(0,1)$. If $U \in A_i$, then $X = x_i$.

Try All This in R!

Q. Compare to drawing a random multinomial!