University of Illinois | Department of Economics
Spring 2017 | Econ 574 | Roger Koenker

Lecture 5
"Monte Carlo I: Generating random variables"

A necessary prerequisite for any monte carlo experiment in econometrics is the ability to generate random samples from specified probability models. We will begin by addressing this problem and make some more general comments on the "etiquette of monte-carlo."

We will begin by assuming that we have available a means of generating a sequence of iid $U[0,1]$ r.v.'s. There are reasonable requirements which constrain this "availability"

(i) reproducibility
(ii) validity

(i) is relatively easy to deal with. It should be possible to reproduce any given monte carlo experiment. This entails two requirements:

a: a means of setting a seed for the generator so that the sequence of r.v.'s generated can be reproduced
b: portability of the generator across different cpu's.

(ii) validity is more difficult to check. There are a million stories about simple (usually congruential) generators in the bad old days which failed simple tests. For example, a standard IBM system generator RANDU had triples (U_n, U_{n+1}, U_{n+2}) such that

$$U_{n+2} - 6U_{n+1} + 9U_n \in \{-5, -4, \ldots, 9\}$$

so all such triples lay on 15 planes in 3-space. Testing the validity of such generators is a difficult task, particularly the dependence aspects since dependence is inherently rather complicated. So it is best left to the experts. A rather complete discussion is Knuth "The Art of Computer Programming." There are also deep philosophical questions about such generators: are they random or only quasi-random? This is best left to others, who have nothing more serious to worry about. I like very much the comment quoted by Knuth at the beginning of Volume 2 from John von Neumann(1951), "Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin."

Non-Uniform Random Number Generation

An excellent general reference is Devroye (1986). *Non-Uniform Random Variate Generation* (Springer-Verlag)

I will consider two basic methods:

(1) Quantile Method

Thm 1: Let F be a continuous df with inverse $F^{-1}(u) = \inf\{x | F(x) \geq u\}$. If U is a $U[0,1]$ r.v., then $F^{-1}(U)$ has df F.

Proof: For all $y \in \mathbb{R}$

$$P(F^{-1}(U) \leq y) = P(U \leq F(y)) = F(y).$$

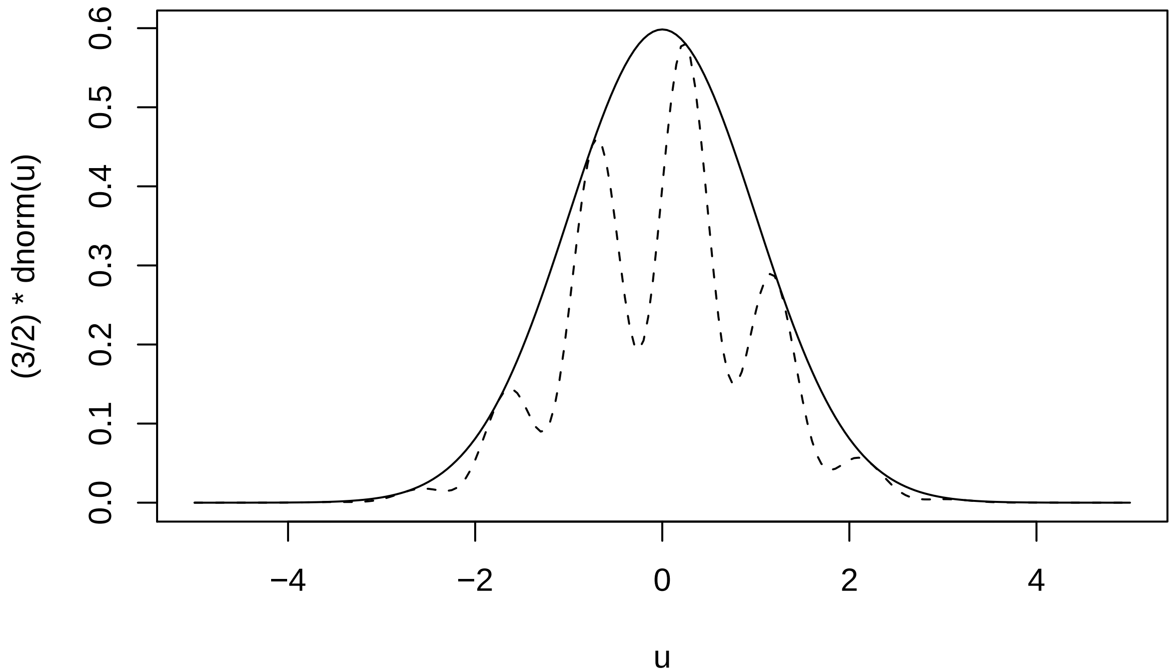

FIGURE 1. Illustration of the rejection method for random number generation using the McCullagh density.

This leads to a very good algorithm for any df F whose inverse is easily computed.

Examples:

(i)	Exponential	$F(x) = 1 - e^{-\lambda x}$	$F^{-1}(u) = -\lambda^{-1}\log(1-u)$
(ii)	Cauchy	$F(x) = \frac{1}{2} + \frac{1}{\pi} atan(x)$	$F^{-1}(u) = \tan(\pi(u - .5))$.
(iii)	Logistic	$F(x) = e^x/(1+e^x)$	$F^{-1}(u) = \log(u/(1-u))$

Related approximate methods can be based on discrete approximations of the quantile function, e.g. by doing piecewise linear approximation of the underlying density. This corresponds essentially to using the trapazoidal rule for integration.

(2) *Rejection Method*

When the quantile function isn't readily available, but the density function is, the following result often yields a good strategy.

Thm: Let f and g be densities on $\mathbb{R}$ such that for some $c > 1$

$$f(x) \leq cg(x) \qquad \text{for all } x.$$

If X is generated from g and u is independent $U[0,1]$, then Y may be generated as f by computing

$$Y = cg(X)/f(X)$$

and "rejecting" X until $YU \leq 1$.

Pf: (By picture)

If we draw from $X \sim g(x)$ and then compute $cg(X)$, the pair $(X, c \cdot U \cdot g(X))$ is uniformly distributed on the region illustrated below the curve $cg(x)$. The rejection strategy makes the selected pairs uniform on the lower region below $f(x)$, and the result follows from the following lemmas.

Lemma 1: (Devroye) Let X be a random variable with density f on $\Re$ and U, an independent $U[0,1]$ r.v. Then $(X, cUf(X))$ is uniformly distributed on the set

$$A = \{(x,u) | x \in \Re,\ \ 0 \le u \le cf(x)\}$$

where $c > 0$ is arbitrary. On the contrary, if (X, U) are uniform on A, then X has density f.

Pf.: For the first part, take any Borel set $B \subseteq A$ and let $B_x = \{u | (x,u) \in B\}$. By Tonelli's theorem, Davidson, p. 68, we have,

$$P((X, cUf(X)) \in B) = \int \int_{B_x} \frac{1}{cf(x)} du f(x) dx = \frac{1}{c} \int_B dudx$$

Since the area of the set A is c, this shows that $(X, cUf(X))$ is uniformly distributed.

For the second part we must show that for any Borel set $B \subset \Re^p$

$$P(X \in B) = \int_B f(x)dx,$$

but

$$P(X \in B) = P((X,U) \in B_1)$$

where

$$B_1 = \{(x,u) : x \in B,\ \ 0 \le u \le cf(x)\}$$

so

$$P(X \in B) = \frac{\int \int_{B_1} du\ \ dx}{\int \int_A du\ \ dx} = \frac{1}{c} \int cf(x)dx = \int f(x)dx$$

Lemma 2: (Devroye) Let $\{X_i\}$ be a sequence of iid r.v.s taking values in $\Re^p$ and let $A \subseteq \Re^p$ be a Borel set such that

$$P(X_1 \in A) = p > 0.$$

Let Y be the first X_i taking a value in A. Then Y has a distribution determined by

$$P(Y \in B) = \frac{P(X_1 \in A \cap B)}{p}$$

Pf.: For any Borel set B,

$$\begin{aligned} P(Y \in B) &= \sum_{i=1}^{\infty} P(X_1 \in A, \ldots, X_{i-1} \in A, X_i \in B \cap A) \\ &= \sum_{i=1}^{\infty} (1-p)^{i-1} P(X_1 \in A \cap B) \\ &= \frac{1}{1-(1-p)} P(X_1 \in A \cap B). \end{aligned}$$

Example: Consider $f(x) = \phi(x)(1 + \frac{1}{2}\sin(2\pi x))$ as in Problem Set 1. This isn't easily integrated let alone inverted for F^{-1}. But rejection is easy since $f(x) \le \frac{3}{2}\phi(x)$.

Obviously, the nearer the bound c is to 1, the better off you are. Devroye notes that N, the number of draws until a success, is a geometric r.v., i.e.,

$$P(N = n) = (1 - p)^{n-1} p$$

where

$$\begin{aligned} p &= P(f(x) \geq cUg(x)) \\ &= \int p(U \leq \frac{f(x)}{cg(x)}) dx \\ &= \int \frac{f(x)}{cg(x)} g(x) dx = \frac{1}{c} \end{aligned}$$

so

$$\begin{aligned} EN &= \frac{1}{p} = c \\ VN &= \frac{1-p}{p^2} = c^2 - c \end{aligned}$$

Thus, it is clear that a smaller value c is better, and these calculations allow one to estimate how many X's are needed to generate n observations from the density f. There are various special tricks and transformations for generating specific random variables, but the two simple strategies we have discussed are the primary ones. For a vastly more through discussion see Devroye. This concludes our discussion of generating r.v.'s except to suggest that in *R* and *Mathematica* most familiar distributions are already built in. Check the documentation closely before embarking on your own – in this domain, as in others, it is always easier, and generally more reliable to stand on the shoulders of your predecessors.