

8 Simulation and Resampling Methods

8.1 Pseudo-random number generation

A generator of pseudo-random numbers from (the distribution of) a random variable Z is an **algorithm** which, starting from an initial value z_0 (called the **seed**) and a transformation D , produces a sequence of values

$$z_j = D(z_{j-1}) = D(D(z_{j-2})) = \cdots = D^j(z_0), \quad j = 1, \dots, m,$$

that reproduce the behavior of an IID sample from the random variable Z .

For many applications, the key ingredient is being able to generate **uniform pseudo-random numbers**, u_1, u_2, \dots , i.e. pseudo-random numbers from a standard uniform distribution.

A uniform pseudo-random number generator is **acceptable** if uniformity of its output is **not** rejected by a set of tests for uniformity, such as the Kolmogorov-Smirnov test, or by tests based on the degree of correlation between u_j and u_{j-1}, \dots, u_{j-p} .

The standard uniform distribution

Recall that a continuous random variable U has a **standard uniform** distribution if its DF is

$$F(u) = \begin{cases} 0, & \text{if } u \leq 0, \\ u, & \text{if } 0 < u < 1, \\ 1, & \text{otherwise,} \end{cases}$$

and its density is

$$f(u) = \begin{cases} 1, & \text{if } 0 < u < 1, \\ 0, & \text{otherwise.} \end{cases}$$

The Uniform Song

There are continuous distributions,
discrete ones too.
Some are heavy tailed
and some are skew.
There are logistics and chi squares,
but these we will scorn,
'Cause the loveliest of them all
is the Uniform.

Shorack G.R., and Wellner J.A. (1986), *Empirical Processes with Applications to Statistics*, p. xxxiii.

8.1.1 Uniform pseudo-random number generators

The most common uniform pseudo-random number generators are based on the **recurrence relation**

$$r_{j+1} = \lambda r_j + \mu \pmod{M}, \quad j = 0, 1, 2, \dots,$$

where λ , μ , M and r_0 are **integers** chosen by the user. Uniform pseudo-random numbers are obtained by computing $u_{j+1} = r_{j+1}/M$, $u_{j+2} = r_{j+2}/M$, etc. Generators of this form are called **congruential generators**. They all have a **finite** period p , i.e. $r_{j+p} = r_j$, with the magnitude of p depending on λ , μ and M . It is clearly desirable to choose these parameters so that p is large.

The pseudo-uniform generators implemented in Stata are:

- the **KISS (Keep It Simple Stupid) generator** by Marsaglia and Zaman (1993);
- the **MT64 generator** by Matsumoto and Nishimura (1998).

8.1.2 The inverse transform method

Given a **uniform** pseudo-random number generator, you may use the following result to generate pseudo-random numbers from any other **continuous** distribution.

Theorem 8.1 (Probability integral transform) *Suppose the random variable U is distributed as standard uniform. If Z is a continuous random variable with DF $G(z)$ and quantile function (QF) $G^{-1}(u) = \inf\{z: G(z) \geq u\}$, $0 < u < 1$, then the random variable $G^{-1}(U)$ has the same distribution as Z .*

It follows from Theorem 8.1 that, if u_1, \dots, u_m are uniform pseudo-random numbers and Φ^{-1} is the QF of the $\mathcal{N}(0, 1)$ distribution, then z_1, \dots, z_m , with $z_j = \Phi^{-1}(u_j)$, is a pseudo-random sample from the $\mathcal{N}(0, 1)$ distribution.

Although Φ^{-1} is not available in closed form, a useful **approximation** is

$$\Phi^{-1}(u) = t - \frac{a_0 + a_1 t}{1 + b_1 t + b_2 t^2},$$

where $t = -2 \ln u$ and

$$a_0 = 2.30753, \quad a_1 = .27061, \quad b_1 = .99229, \quad b_2 = .04481$$

(Robert and Casella 2004, pp. 40–41).

For producing **pairs** of pseudo-random numbers from the $\mathcal{N}(0, 1)$ distribution, an alternative is the **Box-Müller algorithm**. This is based on the fact that the **polar representation** of a pair of numbers x_1 and x_2 is

$$x_1 = |r| \cos \alpha, \quad x_2 = |r| \sin \alpha,$$

where $r^2 = x_1^2 + x_2^2$ and α ranges between 0 and 2π . If x_1 and x_2 are independent draws from a $\mathcal{N}(0, 1)$ distribution, then r^2 is a draw from a χ_2^2 distribution, or equivalently an exponential distribution with parameter 1/2, and α is a draw from a uniform distribution on the interval $[0, 2\pi]$.

Thus, if (u_1, u_2) are uniform pseudo-random numbers, then

$$z_1 = \sqrt{-2 \ln u_1} \cos(2\pi u_2), \quad z_2 = \sqrt{-2 \ln u_1} \sin(2\pi u_2),$$

are pseudo-random numbers from the $\mathcal{N}(0, 1)$ distribution. This method tends to be **slow**, and may be **problematic** when used in conjunction with a congruential pseudo-uniform generator.

8.1.3 Location-scale transformations

A parametric model may often be interpreted as the result of certain transformations applied to some basic distribution. This relationship may in turn be exploited to produce pseudo-random samples from any member of a parametric model of interest.

An important class of transformations is the class of **affine transformations**. Given a random variable Z , consider all random variables that may be represented as $Y = \mu + \sigma Z$ for some $\mu \in \mathbb{R}$ and $\sigma > 0$. If G is the DF of Z , then the DF of any random variable Y of this type is

$$F(y; \mu, \sigma) = \mathbb{P}\{Y \leq y\} = \mathbb{P}\left\{Z \leq \frac{y - \mu}{\sigma}\right\} = G\left(\frac{y - \mu}{\sigma}\right).$$

The parametric family $\mathcal{F} = \{F(y; \mu, \sigma), \mu \in \mathbb{R}, \sigma > 0\}$ of distributions is called a **location and scale model**, μ is called the **location parameter**, σ is called the **scale parameter**, and we say that Z **generates** \mathcal{F} . If Z has a finite variance, then $\mathbb{E}Y = \mu + \sigma \mathbb{E}Z$ and $\text{Var}Y = \sigma^2 \text{Var}Z$. In general, however, μ and σ^2 need not correspond to the mean and variance of Y .

If G^{-1} is the QF of Z , the QF associated to any member Y of a location and scale model is

$$Q(u; \mu, \sigma) = \mu + \sigma G^{-1}(u), \quad 0 < u < 1.$$

Thus, if u_1, \dots, u_m are uniform pseudo-random numbers, then y_1, \dots, y_m , with $y_j = \mu + \sigma G^{-1}(u_j)$, is a pseudo-random sample from Y .

For example, $Y \sim \mathcal{N}(\mu, \sigma^2)$ **if and only** if $Y = \mu + \sigma Z$, where $\sigma > 0$ and $Z \sim \mathcal{N}(0, 1)$. The family of $\mathcal{N}(\mu, \sigma^2)$ distributions is therefore a location and scale model generated by the $\mathcal{N}(0, 1)$ distribution, so the DF and the QF of any member Y of this family are

$$F(y; \mu, \sigma) = \Phi\left(\frac{y - \mu}{\sigma}\right), \quad Q(u; \mu, \sigma) = \mu + \sigma \Phi^{-1}(u),$$

where Φ and Φ^{-1} respectively denote the DF and the QF of the $\mathcal{N}(0, 1)$ distribution. If u_1, \dots, u_m are uniform pseudo-random numbers, then y_1, \dots, y_m , with $y_j = \mu + \sigma \Phi^{-1}(u_j)$, is a pseudo-random sample from a $\mathcal{N}(\mu, \sigma^2)$ distribution.

8.1.4 Simulation experiments

The generation of pseudo-random numbers from a **known** distribution is a key component of simulation experiments (sometimes called **Monte Carlo experiments**) aimed at assessing the finite sample properties of estimators and test procedures under a variety of alternative DGPs.

For example, to investigate the sampling distribution of the sample mean for a random sample of 100 observations from a **known** distribution (e.g., a Student's t with 2 degrees of freedom) one could simply draw a large number of samples of size 100 from that distribution, compute the sample mean for each, and plot the result.

8.1.5 Quasi-Monte-Carlo simulations

There are two problems with traditional simulation experiments:

- To keep things simple, the DGPs considered are often not very realistic.
- With complex estimators or complex models, the number of permutations of the assumptions that one should consider becomes rapidly very large.

An alternative is the **quasi-Monte-Carlo approach**, where statistical models are estimated on a randomly selected subset of the sample and evaluated on the remaining subset. This approach is essentially similar to the **cross-validation approach** discussed in Section [8.5](#).

8.2 Monte Carlo methods

Monte Carlo methods (the name was introduced by Metropolis and Ulam 1949) represent a general approach whereby mathematical problems of an analytical nature which prove technically intractable (or their solution involves prohibitively expensive costs) can be “solved” by substituting an equivalent stochastic problem and solving the latter.

In this section I discuss accept-reject methods (Section 8.2.1), Monte Carlo integration (Section 8.2.2), importance sampling (Section 8.2.3), the Metropolis-Hastings algorithm (Section 8.2.4), and Gibbs sampling (8.2.5).

8.2.1 The accept-reject method

There are many distributions from which it is difficult, or even impossible, to simulate by the methods discussed so far. Even if it is possible, there may be more efficient methods for generating pseudo random numbers from those distributions.

The accept-reject method only requires you to know the functional form of the density of interest (or **target density**) f up to a multiplicative constant. More precisely, the basic idea is to find an alternative distribution, with density function g (called the **instrumental density**), which is “close” to the target density and for which you already have an efficient generator (e.g., by the inverse transform method).

An important requirement on g is that the ratio $f(z)/g(z)$ is bounded by some positive constant c , that is,

$$\sup_z \frac{f(z)}{c g(z)} \leq 1,$$

with c possibly close to 1.

This method is especially useful in Bayesian analysis, when you need to simulate from the posterior density $p(\boldsymbol{\theta} | \mathbf{z}) = f(\mathbf{z} | \boldsymbol{\theta}) p(\boldsymbol{\theta}) / f(\mathbf{z})$, which is typically known up to the multiplicative constant $1/f(\mathbf{z})$.

The accept-reject algorithm

For a **continuous** random variable Z with density f , the accept-reject method works as follows:

Algorithm 8.1

- (1) *Generate a pseudo-random number y from an instrumental density g .*
- (2) *Independently generate a **uniform** pseudo-random number u .*
- (3) *If*

$$u \leq \frac{f(y)}{c g(y)},$$

then set $z = y$ (“accept”), otherwise go back to Step 1 (“reject”).

Notice that:

- $f(Y)$ and $g(Y)$ are random variables, and so is their ratio $R(Y) = f(Y)/[c g(Y)]$, which takes values in the interval $(0, 1]$ and is distributed independently of U in Step 2.
- The number of iterations N needed to successfully generate z is a random variable with a **geometric distribution**, i.e., if $\pi = \mathbb{P}\{U \leq R(Y)\}$, then

$$\mathbb{P}\{N = n\} = (1 - \pi)^{n-1} \pi, \quad n = 1, 2, \dots,$$

and the average number of iteration required is $\mathbb{E} N = 1/\pi$.

- You can show that $\pi = 1/c$, so $\mathbb{E} N = c$.
- Given an instrumental density g , the bounding constant is therefore

$$c = \sup_z \frac{f(z)}{g(z)}.$$

A similar algorithm is available for **discrete** random variables.

Example: The normal distribution

Consider, without loss of generality, the problem of a pseudo-random sample from $Z \sim \mathcal{N}(0, 1)$.

If you can generate from the absolute value $X = |Z|$, then by symmetry you can generate from Z by (i) independently generating from a random variable S that is equal to 1 or -1 with probability 1/2, and (ii) setting $Z = S X$. The density of the nonnegative random variable X is

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \geq 0.$$

As an instrumental density, choose the **unit exponential** $g(y) = e^{-y}$, $y \geq 0$. Using standard calculus, you can show that the ratio $f(y)/g(y) = e^{x-x^2} \sqrt{2/\pi}$ is uniquely maximized when $x = 1$. Thus you can set $c = \sqrt{2e/\pi} \approx 1.32$, so $R(y) = f(y)/[c g(y)] = e^{-(y-1)^2/2}$. Therefore, an algorithm for generating Z is:

Algorithm 8.2

- (1) *Generate a pseudo-random number y from a unit exponential distribution.*
- (2) *Independently generate a **uniform** pseudo-random number u .*
- (3) *If $u \leq e^{-(y-1)^2/2}$, set $x = y$; otherwise go back to Step 1.*
- (4) *Independently generate another **uniform** pseudo-random number u' . Set $z = x$ if $u' \leq 0.5$ and $z = -x$ otherwise.*

Steps 3–4 can be simplified by noticing that $u \leq e^{-(y-1)^2/2}$ if and only if $-\ln u \geq (y-1)^2/2$, where $-\ln u$ is generated from the unit exponential. This gives:

Algorithm 8.3

- (1) *Independently generate uniform pseudo-random numbers u_1 and u_2 , and compute $y_j = -\ln u_j$, $j = 1, 2$.*
- (2) *If $y_1 \geq (y_2 - 1)^2/2$, set $x = y_1$; otherwise go back to Step 1.*
- (3) *Independently generate a **uniform** pseudo-random number u_3 . Set $z = x$ if $u_3 \leq 0.5$ and $z = -x$ otherwise.*

8.2.2 Classical Monte Carlo integration

Two major classes of numerical problems that arise in statistical inference are **optimization problems** and **integration problems**.

I already discussed optimization problems in Section 7.1.2. In this section I discuss integration problems.

Consider first the problem of numerically computing the integral

$$\mathbb{E} h(Z) = \int_{\mathcal{Z}} h(z) f(z) dz. \quad (50)$$

Given a pseudo-random sample z_1, \dots, z_m from f , **classical Monte Carlo integration** approximates this integral by the average

$$\bar{h}_m = \frac{1}{m} \sum_{j=1}^m h(z_j).$$

Clearly, \bar{h}_m is an **unbiased** estimator of $\mathbb{E} h(Z)$ and is **strongly consistent** as $m \rightarrow \infty$. When $\mathbb{E} h(Z)^2$ is finite, you may estimate the sampling variance of \bar{h}_m by

$$v_m^2 = \frac{1}{m^2} \sum_{j=1}^m [h(z_j) - \bar{h}_m]^2.$$

Since

$$\frac{\bar{h}_m - \mathbb{E} h(Z)}{v_m} \Rightarrow \mathcal{N}(0, 1) \quad \text{as } m \rightarrow \infty,$$

you may also construct a test of convergence and 95% confidence bounds on the approximation to (50) of the form $\bar{h}_m \pm 1.96 v_m$.

8.2.3 Importance sampling

This method is based on the following alternative representation of (50)

$$\mathbb{E} h(Z) = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x) dx,$$

where g is some instrumental density different from f but easy to simulate, and approximates $\mathbb{E} h(Z)$ by the average

$$\bar{h}_m^* = \frac{1}{m} \sum_{j=1}^m h(x_j) \frac{f(x_j)}{g(x_j)},$$

where x_1, \dots, x_m is a pseudo-random sample from the density g .

Clearly, \bar{h}_m^* is **unbiased** for $\mathbb{E} h(Z)$. It is also **strongly consistent** as $m \rightarrow \infty$ provided that \bar{h}_m^* has finite variance under g or, equivalently, $h(X)^2 f(X)^2 / g(X)^2$ has finite mean under g , that is,

$$\int_{\mathcal{X}} h(x)^2 \frac{f(x)^2}{g(x)^2} g(x) dx = \int_{\mathcal{Z}} h(z)^2 \frac{f(z)}{g(z)} f(z) dz = \mathbb{E} h(Z)^2 \frac{f(Z)}{g(Z)}$$

is finite.

This means that instrumental distributions for which the ratio f/g is unbounded, i.e. those that have **lighter tails** than f , are **inappropriate** because the estimate \bar{h}_m^* may change a lot from one sample to the next. If instead g has thicker tails than f , the ratio f/g will not cause divergence of $\mathbb{E} h(Z)^2 f(Z)/g(Z)$.

An alternative that addresses this issue and generally yields a **more stable** estimator is

$$\bar{h}_m^\dagger = \frac{\sum_{j=1}^m h(x_j) f(x_j) / g(x_j)}{\sum_{j=1}^m f(x_j) / g(x_j)}.$$

Although \bar{h}_m^\dagger is **biased** for $\mathbb{E} h(Z)$, its bias is **small** and vanishes with m , so the improvement in variance makes it an attractive alternative to \bar{h}_m and \bar{h}_m^* .

As argued by Robert and Casella (2004, p. 96):

“importance sampling methods can bring considerable improvement over naive Monte Carlo estimates when implemented with care. However, they can encounter disastrous performances and produce poor estimates when the variance conditions are not met.”