University of Sydney
School of Information Technologies

Generating Random Variables

Pseudo-Random Numbers

Definition: A sequence of *pseudo-random* numbers (U_i) is a deterministic sequence of numbers in $[0,1]$ having the same relevant statistical properties as a sequence of random numbers.

The most widely used method of generating pseudo-random numbers are the congruential generators:

$$\begin{aligned} X_i &= (aX_{i-1} + c) \bmod M \\ U_i &= X_i / M \end{aligned}$$

for a multiplier *a*, shift *c*, and modulus *M*, all integers.

The sequence is clearly periodic, with maximum period *M*.
The values of *a* and *c* must be carefully chosen to maximise the period of the generator, and to ensure that the generator has good statistical properties.

Some examples:

M	*a*	*c*
2^{59}	13^{13}	0
2^{32}	69069	1
2^{31}-1	630360016	0
2^{32}	2147001325	715136305

Reference: Ripley, *Stochastic Simulation*, Chapter 2

Generating Arbitrarily Distributed Random Variables from Uniform Random Variables

We need to be able to convert uniformly distributed [0,1) random variables into rv's drawn from the distributions we are interested in. This can be done using the transform method as follows:

Transform Method

Let X be the random variable we are interested in generating, and let $F(x)$ be its distribution function, ie

$$F(x) = \Pr\{X \le x\}$$

There is a theorem which says that, if $F(x)$ is continuous, then $F(X)$ is uniformly distributed on the interval $(0,1)$. This can be used to generate random variables with specified distributions:

Let $F(X) = Y$. Then Y is defined on the range between 0 and 1. We can show that if Y is a random variable uniformly distributed between 0 and 1, the variable X defined by

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

has the cumulative distribution $F(X)$, ie

$$\Pr\{X \le x\} = \Pr\{Y \le F(x)\} = F(x)$$

For our simulation, we generate Y with the built-in pseudo-random number generator, and apply the inverse function F^{-1} to it to give us our desired random variable.

Example

Let's assume that the arrival process to the single server queue mentioned above is a Poisson Process with rate λ, so that the times between arrivals have a negative exponential distribution, with mean $1/\lambda$, ie $F(x) = 1 - e^{-\lambda x}$. The procedure outlined above says that we generate a uniform random variable Y, set $Y = 1 - e^{-\lambda X}$, so that
$X = F^{-1}(Y) = -\frac{\ln(1-Y)}{\lambda}$ is our desired variable.

This procedure works well if we can find an explicit formula for the inverse transform $F^{-1}(.)$, if we can’t find an explicit formula for this, then we have to resort to other methods.

Generating Gaussian Random Variables

The method described above requires an analytic expression for the distribution function of the random variable. For a Gaussian rv, we do not have such an expression, so we need to find another way of generating such a random variable.

Method 1

One option is to use a rational approximation to the inverse of the distribution function (Abramowitz & Stegun, "Handbook of Mathematical Functions") as follows:

To find x_p such that $Q(x_p) = p$, where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$:

$$x_p = t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}, \qquad t = \sqrt{\ln \frac{1}{p^2}}$$

$c_0 = 2.515517 \quad d_1 = 1.432788$
$c_1 = 0.802853 \quad d_2 = 0.189269$
$c_2 = 0.010328 \quad d_3 = 0.001308$

The approximation has maximum error of 4.5×10^{-4} in p.

Method 2

Let $\{U_i\}$ be a sequence of iid uniform random variables on [0,1). Then $X = \sum_{i=1}^{12} U_i - 6$ is approximately Gaussian with zero mean and unit variance (by the central limit theorem).

Method 3 (Box-Muller)

Let U_1 and U_2 be independent and uniformly distributed on (0, 1).
Set $\Theta = 2\pi U_1$ and $R = \sqrt{12 \ln U_2}$.
Then $X = R\cos\Theta$ and $Y = R\sin\Theta$ are independent standard normal variates.

Method 4

Ripley lists several other methods for doing this. One of the simplest is as follows (Ripley, Algorithm 3.6):

The algorithm generates two independent Gaussian random variables, with zero mean and unit variance.

```
1.    Repeat
          Generate $V_1, V_2$ independent and uniformly distributed on (–1, 1)
          until $W = V_1^2 + V_2^2 < 1$

2.    Let $C = \sqrt{-2W^{-1} \ln W}$
```

3. Return $X = C\mathrm{V}_1$ and $Y = CV_2$

This method is usually faster than method 3 because it avoids having to evaluate the trigonemetric functions sin and cos.

How does it work?
Step 1 gives a pair of rv's (V_1, V_2) uniformly distributed on the unit disk.
Let (R, Θ) be the polar coordinates of (V_1, V_2), so $W = R^2$.
Then (W, Θ) has joint pdf $1/2\pi$ on $(0,1)\times(0,2\pi)$.
Let $E = -\ln W$. Then $X = \sqrt{2E}\cos\Theta = \sqrt{-2\ln W}\left(V_1/\sqrt{W}\right) = CV_1$,
and similarly $Y = \sqrt{2E}\sin\Theta = CV_2$.

In reverse, consider a pair of independent standard normal random variables *X* and *Y*.

Their joint pdf is $p_{X,Y}(x,y) = \frac{1}{2\pi}e^{-\left(x^2+y^2\right)/2}$ on $(-\infty,\infty)\times(-\infty,\infty)$.
Let (R, Θ) be the polar coordinates of (X, Y).
Then (R, Θ) has joint pdf $\frac{1}{2\pi}\left(re^{-r^2/2}\right)$ on $(0,\infty)\times(0,2\pi)$ (by transformation of random variables),
i.e. R and Θ are independent (R is Rayleigh distributed and Θ is uniformly distributed)

Now, $S = R^2 = X^2 + Y^2$ has a χ_2^2 distribution, which is the same as a negative exponential distribution of mean 2.
Now let $U = e^{-S/2}$, so $\Pr\{U \le u\} = \Pr\{-S/2 \le \ln u\} = \Pr\{S \ge -2\ln u\} = e^{-(-2\ln u)/2} = u$
For $0 \le u < 1$, i.e. (U, Θ) is uniformly distributed on the unit disk.

Reference

B.D. Ripley, *Stochastic Simulation*, Wiley, 1987

Lognormal Distribution - Base *e*

Let $X = e^{Y}$ where *Y* is a Normal random variable, $Y \sim N(\mu, \sigma^2)$
Then *X* is said to have a lognormal distribution, with parameters μ andσ.

By a simple transformation of random variables: $p_X(x) = \dfrac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2/(2\sigma^2)}$

$$E\{X\} = e^{(\mu + \sigma^2/2)}$$
$$\mathrm{var}\{X\} = e^{(2\mu+\sigma^2)}(e^{\sigma^2} - 1) = (E\{X\})^2 (e^{\sigma^2} - 1)$$
$$c_X{}^2 = \frac{\mathrm{var}\{X\}}{E\{X\}^2} = (e^{\sigma^2} - 1)$$
$$E\{X^n\} = e^{(n\mu + n^2\sigma^2/2)}$$

To generate a lognormal rv, first generate the normal rv *Y*, then transform by $X = e^{Y}$.

Lognormal Distribution - Base 10 (decibel form)

Let $X = 10^{Y/10}$ where *Y* is a Normal random variable, $Y \sim N(\mu, \sigma^2)$
Then *X* is said to have a lognormal distribution, with “mean” μ dB, and “variance” σ dB.

Note that these are not really the true mean and true variance of the lognormal distribution - but it is easier to specify the parameters as parameters of the normal distribution instead of the lognormal.

To find the real mean and variance:

Let $X = 10^{Y/10} = e^{Y \ln 10/10} = e^{Z}$ with $Z \sim N\left(\mu \ln 10/10, (\sigma \ln 10/10)^2\right)$

Therefore

$$E\{X\} = e^{(\mu \ln 10/10 + (\sigma \ln 10/10)^2/2)} = 10^{(\mu/10 + \sigma^2 \ln 10/200)}$$
$$\mathrm{var}\{X\} = (E\{X\})^2 (e^{(\sigma \ln 10/10)^2} - 1) = (E\{X\})^2 (10^{\sigma^2 \ln 10/100} - 1)$$

To generate a lognormal rv, first generate the normal rv *Y*, then transform by $X = 10^{Y/10}$.

General Discrete Distribution

Assume that we want to generate discrete random variables from a specified distribution, e.g. $\{p_k; k = 0,1,\cdots,N\}$.

Inversion Method

The inversion method reduces to searching for an appropriate index in a table of probabilities:

If $\sum_{k=0}^{j-1} p_k \le U < \sum_{k=0}^{j} p_k$ then return $X = j$.

Example

Assume that the required distribution is $p_0 = 0.5; p_1 = 0.3; p_2 = 0.2$.

Then,	if $0.0 \le U < 0.5$	return $X = 0$
	if $0.5 \le U < 0.8$	return $X = 1$
	if $0.8 \le U < 1.0$	return $X = 2$

Table Method

Generate a large array of size *M*, with

elements 1	to	Mp_0	having the value 0,
elements $Mp_0 + 1$	to	$Mp_0 + Mp_1$	having the value 1,
elements $Mp_0 + Mp_1 + 1$	to	$Mp_0 + Mp_1 + Mp_2$	having the value 2,

etc

Then generate a uniform integer rv *X* from 1 to *M*. The *X*th element of the array is a rv with the required distribution.

This algorithm is very fast to run, but this is achieved at the expense of often requiring the allocation of large amounts of storage for the arrays.

Example 1

Assume that the required distribution is $p_0 = 0.5; p_1 = 0.3; p_2 = 0.2$.

Then construct an array of size 10 where 5 elements are zero, 3 are 1, and 2 are 2, i.e.
$[0, 0, 0, 0, 0, 1, 1, 1, 2, 2]$.

Now sample uniformly from this array. To do this, generate a uniform integer rv distributed from 1 to 10. If the generated rv is *X*, choose the *X*th element of the array.

Example 2: Poisson distribution

Assume that the required distribution is $p_k = \frac{\frac{A^k}{k!}}{\sum_{i=0}^{N}\frac{A^i}{i!}}$, i.e. a truncated Poisson distribution.

Assume that A = 5 and N = 8, so $\{p_k\}$ =
{0.0072, 0.0362, 0.0904, 0.1506, 0.1883, 0.1883, 0.1569, 0.1121, 0.0700}

Construct an array of size (say) 10000 where

elements 1	to	72	have the value 0,
elements 73	to	434	have the value 1,
elements 435	to	1338	have the value 2,
elements 1339	to	2844	have the value 3,
elements 2845	to	4727	have the value 4,
elements 4728	to	6610	have the value 5,
elements 6611	to	8179	have the value 6,
elements 8180	to	9300	have the value 7,
elements 9301	to	10000	have the value 8.

Now sample randomly from this array. The resulting rv will be approximately the required truncated Poisson.