STA 313: Topics in Statistics

Al Nosedal.
University of Toronto.

Fall 2015

"essentially, all models are wrong, but some are useful"

George E. P. Box

(one of the great statistical minds of the 20th century).

Pseudo random number generators

There are two fundamentally different classes of methods to generate random numbers:

a) True random numbers are generated using some physical phenomenon which is random. Generating such numbers requires specialized hardware and can be expensive and slow.

b) Pseudo random numbers are generated by computer programs. While these methods are normally fast and resource effective, a challenge with this approach is that computer programs are inherently deterministic and therefore cannot produce "truly random" output.

In this course we will only consider pseudo random number generators.

A pseudo random number generator (PRNG) is an algorithm which outputs a sequence of numbers that can be used as a replacement for an independent and identically distributed (i. i. d.) sequence of "true random numbers".

In order to use a computer to initiate a simulation study, we must be able to generate the value of a uniform (0, 1) random variable; such variates are called random numbers. One way to do this would be to take 10 identical slips of paper, numbered 0, 1, 2, . . . , 9, place them in a hat and then successively select n slips, with replacement, from the hat. The sequence of digits obtained (with a decimal point in front) can be regarded as the value of a uniform (0, 1) random variable rounded off to the nearest $(\frac{1}{10})^n$. For instance, if the sequence of digits selected is 3, 8, 7, 2, 1, then the value of the uniform (0, 1) random variable is 0.38721 (to the nearest 0.00001).

However, this is not the way in which digital computers simulate uniform $(0, 1)$ random variables. In practice, they use pseudo random numbers instead of truly random ones. Most random number generators start with an initial value X_0, called the seed, and then recursively compute values by specifying positive integers a, c, and m, and then letting

$$X_{n+1} = (aX_n + c) \text{ modulo } m, \quad n \geq 0$$

where the preceding means that $aX_n + c$ is divided by m and the remainder is taken as the value of X_{n+1}. It can be shown that subject to suitable choices for a, c, m, the preceding gives rise to a sequence of numbers that looks as if it were generated from independent uniform $(0, 1)$ random variables.

Example

For parameters $m = 8$, $a = 5$, $c = 1$ and seed $X_0 = 0$, the
algorithm described above gives the following output:

n	$5X_{n-1} + 1$	X_n
1	1	**1**
2	6	**6**
3	31	**7**
4	36	**4**
5	21	**5**
6	26	**2**
7	11	**3**
8	16	**0**
9	1	**1**
10	6	**6**

- It is normally a bad idea to implement your own PRNG. Therefore, it is advisable to use a well-established method for random number generation, typically the random number generator built into a well-known software package or provided by a well-established library.

- While different PRNGs differ greatly in implementation details, they all use a seed to initialise the state of the random number generator. Often, when non-predictability is required, it is useful to set the seed to some volatile quantity to get a different sequence of random numbers for different runs of the program. At other times it can be more useful to get reproducible results, for example to aid debugging. In these cases, the seed should be set to a known, fixed value.

Let U be a uniform random variable on the interval $(0, 1)$.
a) Find the cumulative distribution function of U.
b) Find the density function of $Y = -\beta ln(1 - U)$.

Let U be a uniform random variable on the interval (0, 1). Find a transformation $G(U)$ such that $G(U)$ possesses an exponential distribution with mean β.

If U possesses a uniform distribution on the interval $(0, 1)$, then the distribution function of U is given by

$$F_U(u) = \begin{cases} 0, & u < 0, \\ u, & 0 \le u \le 1, \\ 1 & u > 1. \end{cases}$$

Let Y denote a random variable that has an exponential distribution with mean β. Then Y has distribution function

$$F_Y(y) = \begin{cases} 0, & y < 0, \\ 1 - e^{-y/\beta} & y \geq 0. \end{cases}$$

Notice that $F_Y(y)$ is strictly increasing on the interval $[0, \infty)$. Let $0 < u < 1$ and observe that there is a unique value y such that $F_Y(y) = u$. Thus, $F_Y^{-1}(u)$, $0 < u < 1$, is well defined. In this case, $F_Y(y) = 1 - e^{-y/\beta} = u$ if and only if $y = -\beta \ln(1 - u) = F_Y^{-1}(u)$.

Consider the random variable $F_Y^{-1}(U) = -\beta ln(1 - U)$ and observe that, if $y > 0$,

$$P(F_Y^{-1}(U) \leq y) = P[-\beta ln(1 - U) \leq y]$$
$$= P[ln(1 - U) \geq -\frac{y}{\beta}]$$
$$= P[U \leq 1 - e^{-y/\beta}]$$
$$= 1 - e^{-y/\beta}.$$

Also, $P[F^{-1}(U) \leq y] = 0$ if $y \leq 0$. Thus, $F^{-1}(U) = -\beta ln(1 - U)$ possesses an exponential distribution with mean β, as desired.

Definition 1.12

Let F be a distribution function. Then the inverse of F is defined by

$$F^{-1}(u) = \inf\{x \in \Re | F(x) \geq u\}$$

for all $u \in (0, 1)$.

Let $F : \Re \to [0, 1]$ be a distribution function and $U \sim U[0, 1]$. Define $X = F^{-1}(U)$. Then X has distribution function F.

Using the definitions of X and F^{-1} we find
$P(X \leq a) = P(F^{-1}(U) \leq a) = P(inf\{x|F(x) \geq U\} \leq a)$.
Since $inf\{x|F(x) \geq U\} \leq a$ holds if and only if $F(a) \geq U$, we can conclude
$P(X \leq a) = P(F(a) \geq U) = F(a)$.

Let X have density

$$f(x) = \begin{cases} 3x^2 & x \in [0,\ 1]\ and \\ 0 & otherwise \end{cases}$$

Example (cont.)

Then

$$F(a) = \int_{-\infty}^{a} f(x)dx = \begin{cases} 0 & \text{if } a < 0 \\ a^3 & \text{if } 0 \le a < 1 \\ 1 & \text{for } 1 \le a \end{cases}$$

F^{-1} is given by the usual inverse function and consequently $F^{-1}(u) = u^{1/3}$ for all $u \in (0, 1)$. Thus, by proposition 1.14, if $U \sim U[0, 1]$, the cubic root $U^{1/3}$ has the same distribution as X.

R code

```
U=runif(10000);
X=U^(1/3);
# runif generates random numbers from U[0,1];
# in this case, 10000 of them;
# X contains 10000 random numbers from r.v.  X;
```

```
hist(X, freq=FALSE);
x=seq(0,1,by=0.01);
f=3*x^(2);
lines(x,f,lty=2);
# hist makes a histogram of X;
# freq = FALSE is telling R to use relative;
# frequencies;
# lines adds the pdf we want;
# lty = 2 tells R that we want a dotted line;
```

Write a program which uses the inverse transform method to generate random numbers with the following density:

$$f(x) = \begin{cases} 1/x^2 & \text{if } x \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

To test your program, plot a histogram of 10 000 random numbers together with the density f.

Lemma 1.8

Let $U \sim U[0,1]$ and $n \in \mathbb{N}$. Define a random variable X by $X = \lfloor nU \rfloor$, where $\lfloor \cdot \rfloor$ denotes rounding down. Then $X \sim U\{0, 1, ..., n-1\}$.

By the definition of X we have
$P(X = k) = P(\lfloor nU \rfloor = k) = P(nU \in [k, \ k+1)) =$
$P\left(U \in \left[\frac{k}{n}, \frac{k+1}{n}\right)\right) = \frac{k+1}{n} - \frac{k}{n} = \frac{1}{n}$
for all $k = 0, 1, ..., n-1$. This completes the proof.

Suppose that Y is a random variable that takes on only integer values 1, 2, ... and has distribution function $F(y)$. Show that the probability function $p(y) = P(Y = y)$ is given by

$$p(y) = \left\{ \begin{array}{ll} F(1), & y = 1, \\ F(y) - F(y-1), & y = 2, \ 3, ... \end{array} \right.$$

Solution

For $y = 2, 3, ...,$
$F(y) - F(y-1) = P(Y \leq y) - P(Y \leq y-1) = P(Y = y) = p(y).$

Also,
$F(1) = P(Y \leq 1) = P(Y = 1) = p(1).$

Suppose that Y is a random variable that takes on only integer values $1,\ 2,\ ...$. Let $F(y)$ denote the cumulative distribution function of this random variable. Let U be a continuous random variable that is uniformly distributed on the interval $(0,\ 1)$. Define a variable X such that $X = k$ if and only if $F(k-1) < U \leq F(k)$, $k = 1,\ 2,\ ...$. Recall that $F(0) = 0$ because Y takes on only positive integer values. Show that
$P(X = i) = F(i) - F(i-1) = P(Y = i)$, $i = 1,\ 2,\ ...$. That is, X has the same distribution as Y.

By definition,
$P(X = i) = P[F(i-1) < U \leq F(i)] = F(i) - F(i-1)$, for
$i = 1, 2, ...$
From a previous problem, we know that
$P(Y = i) = F(i) - F(i-1)$, for $i = 1, 2, ...$
Thus, X and Y have the same distribution.

Example. The Geometric Distribution

Suppose that independent trials each of which results in a "success" with probability p, $0 < p < 1$, are continually performed until a success occurs. Letting X denote the number of necessary trials, then

$P(X = k) = (1-p)^{k-1}p, \quad k \geq 1$.

The random variable X is said to be a Geometric random variable with parameter p.

Example. The Geometric Distribution

Using the formula for geometric sums we find
$F(k) = \sum_{j=1}^{k} P(X = j) = p \sum_{j=1}^{k} (1-p)^{j-1} = p \frac{1-(1-p)^k}{p} = 1 - (1-p)^k$.

We can simulate such a random variable by generating a random number U and then setting X equal to that value k for which $1 - (1-p)^{k-1} < U \leq 1 - (1-p)^k$ or, equivalently, for which $(1-p)^k \leq 1 - U < (1-p)^{k-1}$.

Example. Geometric Distribution

Since $1 - U$ has the same distribution as U, we can thus define X by

$X = min\{k : U \geq (1 - p)^k\}$
$\quad = min\{k : logU \geq klog(1 - p)\}$
$\quad = min\{k : \frac{logU}{log(1-p)} \leq k\}$ (recall that $log(1 - p)$ is negative)

Thus, the value

$$X = a_k = K = \left\lceil \frac{logU}{log(1 - p)} \right\rceil,$$

where $\lceil \cdot \rceil$ denotes the operation of rounding up a number to the nearest integer, is geometrically distributed with parameter p.

```
U=runif(20000);
p=1/5;
X= ceiling(log(U)/log(1-p));
# p =1/5 probability of success;
# ceiling tells R to round up;
# X contains 20000 random numbers from Geometric(p);
```

R code

```
sample<-table(X)/20000;
plot(sample);
# table creates a frequency table for X;
```

Lemma 1.9

Let $p \in [0, 1]$ and $U \sim U[0,1]$ and define the event E as $E = \{U \leq p\}$. Then $P(E) = p$.

We have
$P(E) = P(U \leq p) = P(U \in [0, p]) = p - 0 = p.$
This completes the proof.

Lemma 1.10

Assume $A = \{a_i | i \in I\}$ where either $I = \{1, 2, ..., n\}$ for some $n \in \mathbb{N}$ or $I = \mathbb{N}$, and where $a_i \neq a_j$ whenever $i \neq j$. Let $p_i \geq 0$ be given for $i \in I$ with $\sum_{i \in I} p_i$. Finally let $U \sim U[0, 1]$ and define

$$K = min\left\{ k \in I | \sum_{i=1}^{k} p_i \geq U \right\}$$

Then $X = a_k \in A$ satisfies $P(X = a_k) = p_k$ for all $k \in I$.

Proof

$$P(X = a_k) = P(K = k) = P\left(\sum_{i=1}^{k-1} p_i < U, \sum_{i=1}^{k} p_i \geq U\right)$$
$$= P\left(U \in \left(\sum_{i=1}^{k-1} p_i, \sum_{i=1}^{k} p_i\right]\right)$$
$$= \sum_{i=1}^{k} p_i - \sum_{i=1}^{k-1} p_i = p_k.$$

for all $k \in I$, where we interpret the sum $\sum_{i=1}^{0} p_i$ for $k = 1$ as 0.
This completes the proof.

Beta Probability Distribution

A random variable Y is said to have a **beta probability distribution** with parameters $\alpha > 0$ and $\beta > 0$ if and only if the density function of Y is

$$f(y) = \begin{cases} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1}, & 0 \le y \le 1, \\ 0, & \text{elsewhere.} \end{cases}$$
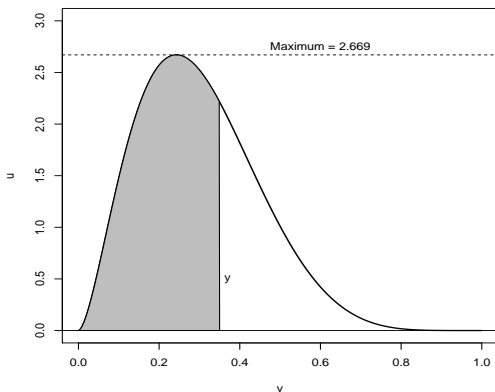
where $\Gamma(\alpha) = \int_0^\infty y^{\alpha-1} e^{-y} dy$.

Gamma Function

The quantity $\Gamma(\alpha)$ is known as the **gamma function**. Direct integration will verify that $\Gamma(1) = 1$. Integration by parts will verify that $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$ for any $\alpha > 1$ and that $\Gamma(n) = (n - 1)!$, provided that n is an integer.

Beta random variable generation

Suppose the goal is to generate $Y \sim Beta(a, b)$. For definitness, set $a = 2.7$ and $b = 6.3$. Below we have put the beta density $f_Y(y)$ inside a box with sides 1 and $c \geq max\ f_Y(y)$.

Now consider the following method of calculating $P(Y \leq y)$. If (U, V) are independent uniform $(0, 1)$ random variables, then the probability of the shaded area is

$$P(V \leq y, U \leq \tfrac{1}{c} f_Y(V)) = \int_o^y \int_0^{f_Y(v)/c} du dv$$
$$= \tfrac{1}{c} \int_0^y f_Y(v) dv$$
$$= \tfrac{1}{c} P(Y \leq y).$$

So we can calculate the beta probabilities from the uniform probabilities, which suggests that we can generate the beta random variable from the uniform random variables.

If we set $y = 1$, then we have $\frac{1}{c} = P(U < \frac{1}{c}f_Y(V))$, so

$$P(Y \le y) = \frac{P(V \le y, \ U \le \frac{1}{c}f_Y(V))}{P(U \le \frac{1}{c}f_Y(V))}$$
$$= P(V \le y | U \le \frac{1}{c}f_Y(V)),$$

which suggests the following algorithm.
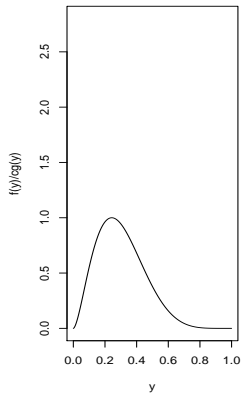
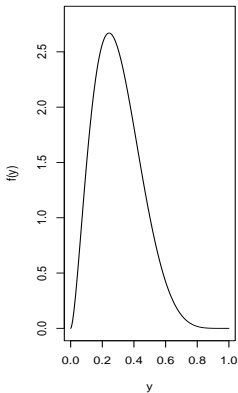To generate $Y \sim beta(a, b)$:

Step 1. Generate (U, V) independent uniform $(0, 1)$.

Step 2. If $U < \frac{1}{c} f_Y(V)$, set $Y = V$; otherwise, return to step 1.

Step 0. Find c such that $\frac{f_Y(y)}{cg_V(y)} \leq 1$. Note that, in this case, it is equivalent to finding $f_Y(y) \leq c$.

```
x=seq(0,1,0.001);
a=2.7;
b=6.3;
f=dbeta(x,a,b);
# dbeta gives you the pdf of a beta;
plot(x,f,type="l")
c=max(f);
c;
```

R code

Step 1. Generate U and V, independent uniform random variables.

```
set.seed(0);
U=runif(10000);
V=runif(10000);
# V = candidates;
# U is used to "decide" which ones we keep;
# by using set.seed, we will have the same samples;
```

Step 2. Determine which "candidates" you keep.

```
weights<-dbeta(V,a,b)/c;
# weights=f(x)/c*g(x);
```
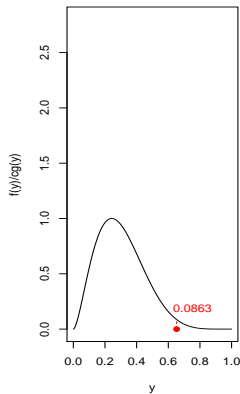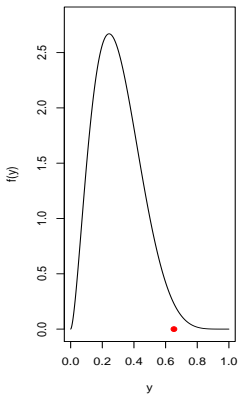(This is not the "final version" of Step 2. First we need to understand how it works).

Explanation of how Step 2 works

Now that we have U, V, and *weights*, let us see how this works, step-by-step. The first value drawn from the candidate density $g_V(y)$ is $V[1] = 0.652677$ and the "weight" function at $V[1]$ will be *weights*$[1] = 0.08632485$. The next step is comparing *weights*$[1]$ to $U[1] = 0.8966972$. Since $U[1] > $ *weights*$[1]$, we reject $V[1] = 0.652677$ as a number coming from a Beta(a=2.7,b=6.3).
Let us see a graph of this situation.

Rejected candidate

From the graph, it makes sense that $V[1]$ was rejected. It is not very likely to draw a 0.65 from a Beta(2.7,6.3) distribution, Right?

Let us see what happens with another candidate. The 919th value drawn from the candidate density $g_V(y)$ is $V[919] = 0.2426574$ and the "weight" function at $V[919]$ will be $weights[919] = 0.9999996$. The next step is comparing $weights[919]$ to $U[919] = 0.7188601$. Since $U[919] < weights[919]$, we **accept** $V[919] = 0.2426574$ as a number coming from a Beta(a=2.3,b=6.7).

Accepted candidate

From the graph, it makes sense that $V[919]$ was accepted. It is very likely to draw a 0.24 from a Beta(2.7,6.3) distribution, Right?
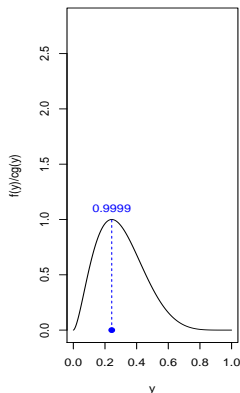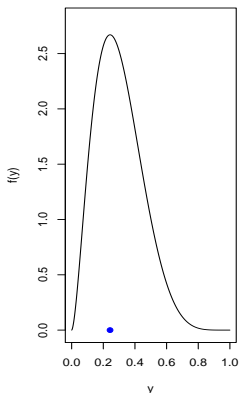
Step 2. Determine which "candidates" you keep.

```
weights<-dbeta(V,a,b)/c;
random.sample=V[U<weights];
# weights=f(x)/c*g(x);
```

A few comments about our last example

Note that we generated $10,000$ candidates but we only ended up with a random sample of size 3707 (you can verify this by typing length(random.sample).

Another thing worth mentioning is that the proportion of accepted candidates is $p = \frac{1}{c} \approx \frac{1}{2.67} = 0.3745$. Thus, the expected number of candidates you have to draw, until your first accepted candidate is $\frac{1}{p} \approx 2.67$.

Algorithm 1.22

Envelope Rejection Sampling.

input: a function f with values in $[0, \infty)$ (the non-normalized target density),

a probability density g (the proposal density),

a constant $c > 0$ such that $f(x) \leq cg(x)$ for all x

randomness used:

X_n i. i. d. with density g (the proposals),

$U_n \sim U[0, 1]$ i. i. d.

output: a sequence of i. i. d. random variables with density $\hat{f}(x) = \frac{1}{Z_f} f(x)$ where $Z_f = \int f(x) dx$.

Suppose that we have a method for simulating a random variable having density function $g(x)$. We can use this as the basis for simulating from the continuous distribution having density $f(x)$ by simulating Y from g and then accepting this simulated value with a probability proportional to $f(Y)/g(Y)$. Specifically, let c be a constant such that

$$\frac{f(y)}{g(y)} \leq c \quad \text{for all y}$$

We then have the following technique for simulating a random variable having density f.

Step 1. Simulate Y having density g and simulate a random number U, where U is a uniform $(0, 1)$ random variable.

Step 2. If $U \leq f(Y)/cg(Y)$, set $X = Y$. Otherwise return to step 1.

We now prove that it works.

Let X be the value obtained and let N denote the number of necessary iterations. Then

$$P[X \leq x] = P[Y_N \leq x]$$
$$= P\left[Y \leq x | U \leq \frac{f(Y)}{cg(Y)}\right]$$
$$= \frac{P\left[Y \leq x, U \leq \frac{f(Y)}{cg(Y)}\right]}{K}$$

where $K = P[U \leq f(Y)/cg(Y)]$. Now the joint density function of Y and U is, by independence,

$$f_{Y,U}(y, u) = g(y) \quad 0 < u < 1$$

so, we have
$P[X \leq x] = \frac{1}{K} \int_{-\infty}^{x} \int_{0}^{f(y)/cg(y)} du\, g(y) dy$
$\qquad = \frac{1}{cK} \int_{-\infty}^{x} f(y) dy$
Letting X approach ∞ and using the fact that f is a density gives
$1 = \frac{1}{cK} \int_{-\infty}^{\infty} f(y) dy = \frac{1}{cK}$.
Hence we obtain that

$$P[X \leq x] = \int_{-\infty}^{x} f(y) dy$$

which completes the proof.

We can use rejection sampling to generate samples from the half-normal distribution with density

$$f(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right), & x \geq 0, \\ 0, & otherwise. \end{cases}$$

Example

If we assume that the proposals are $Exp(\lambda)$-distributed, then the density of the proposals is

$$g(x) = \begin{cases} \lambda exp\left(-\lambda x\right), & x \geq 0, \\ 0, & otherwise. \end{cases}$$

Example

In order to apply algorithm 1.22 we need to determine a constant $c > 0$ such that $f(x) \leq cg(x)$ for all $x \in \Re$. For $x < 0$ we have $f(x) = g(x) = 0$. For $x \geq 0$ we have

$$\frac{f(x)}{g(x)} = \frac{2}{\sqrt{2\pi}\lambda} exp\left(-\frac{x^2}{2} + \lambda x\right).$$

It is easy to check that the quadratic function $-x^2/2 + \lambda x$ attains its maximum at $x = \lambda$.

Example

Thus we have $\frac{f(x)}{g(x)} \leq c^*$ for all $x \geq 0$, where

$$c^* = \frac{2}{\sqrt{2\pi}\lambda} exp\left(-\frac{\lambda^2}{2} + \lambda \cdot \lambda\right) = \sqrt{\frac{2}{\pi\lambda^2}} exp(\lambda^2/2).$$

(we should choose c as small as possible and $c = c^*$ is the optimal choice).

Example

Given our choice of g and c, the acceptance criterion from algorithm 1.22 can be simplified as follows:

$cg(x) \ U \leq f(x)$

$\sqrt{\frac{2}{\pi\lambda^2}} exp\left(\frac{\lambda^2}{2}\right) \lambda exp(-\lambda x)U \leq \frac{2}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right)$

$U \leq exp\left(-\frac{x^2}{2} + \lambda x - \frac{\lambda^2}{2}\right)$

$U \leq exp\left(-\frac{1}{2}(x-\lambda)^2\right).$

Example

This leads to the following algorithm for generating samples from the half-normal distribution:

1. **for** $n = 1, \ 2, 3, ...$ **do**
2. generate $X_n \sim Exp(\lambda)$
3. generate $U_n \sim U[0, \ 1]$
4. **if** $U_n \leq exp(-\frac{1}{2}(X_n - \lambda)^2)$ **then**
5. output X_n
6. **end if**
7. **end for**

Implement the rejection method from example 1.24 to generate samples from a half-Normal distribution from Exp(1)-distributed proposals. Test your program by generating a histogram of the output and by comparing the histogram with the theoretical density of the half-Normal distribution.

Let Y have probability density function $f_Y(y)$. If $h(y)$ is either increasing or decreasing for all y such $f_Y(y) > 0$, then $U = h(Y)$ has density function

$$f_U(u) = f_Y[h^{-1}(u)] \left| \frac{dh^{-1}}{du} \right|,$$

where $\frac{dh^{-1}}{du} = \frac{d[h^{-1}(u)]}{du}$.

Bivariate Transformation Method

Let X_1 and X_2 be jointly continuous random variables with joint probability density function f_{X_1, X_2}. It is sometimes necessary to obtain the joint distribution of the random variables Y_1 and Y_2, which arise as functions of X_1 and X_2. Specifically, suppose that $Y_1 = g_1(X_1, X_2)$ and $Y_2 = g_2(X_1, X_2)$ for some functions g_1 and g_2. Assume that the functions g_1 and g_2 satisfy the following conditions:

1. The equations $y_1 = g_1(x_1, x_2)$ and $y_2 = g_2(x_1, x_2)$ can be uniquely solved for x_1 and x_2 in terms of y_1 and y_2 with solutions given by, say, $x_1 = h_1(y_1, y_2)$, $x_2 = h_2(y_1, y_2)$.

Bivariate Transformation Method

2. The functions g_1 and g_2 have continuous partial derivatives at all points (x_1, x_2) and are such that the following 2×2 determinant

$$J(x_1, x_2) = det \left[\begin{array}{cc} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{array} \right] \neq 0$$

at all points (x_1, x_2).

Under these two conditions it can be shown that the random variables Y_1 and Y_2 are jointly continuous with joint density function given by

$f_{Y_1, Y_2}(y_1, y_2) = f_{X_1, X_2}(x_1, x_2)|J(x_1, x_2)|^{-1}$,

where $x_1 = h_1(y_1, y_2)$, $x_2 = h_2(y_1, y_2)$ and $|J(x_1, x_2)|$ is the absolute value of the Jacobian.

(We will not prove this result, but it follows from Calculus results used for change of variables in multiple integration.)

Let X_1 and X_2 be jointly continuous random variables with probability density function f_{X_1,X_2}. Let $Y_1 = X_1 + X_2$, $Y_2 = X_1 - X_2$. Find the joint density function of Y_1 and Y_2 in terms of f_{X_1,X_2}.

Example

Let (X, Y) denote a random point in the plane and assume that the rectangular coordinates X and Y are independent standard random Normal random variables. We are interested in the joint distribution of R and Θ, the polar coordinate representation of this point (see Figure below).
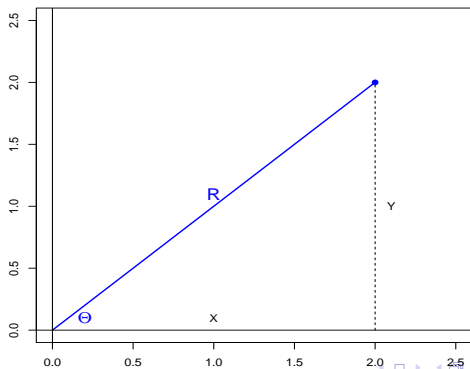
Example

Letting $r = g_1(x, \ y) = \sqrt{x^2 + y^2}$ and $\theta = g_2(x, \ y) = tan^{-1}\left(\frac{y}{x}\right)$, we see that

$\frac{\partial g_1}{\partial x} = \frac{x}{\sqrt{x^2+y^2}}$

$\frac{\partial g_1}{\partial y} = \frac{y}{\sqrt{x^2+y^2}}$

$\frac{\partial g_2}{\partial x} = \frac{1}{1+(y/x)^2}\left(-\frac{y}{x^2}\right) = \frac{-y}{x^2+y^2}$

$\frac{\partial g_2}{\partial y} = \frac{1}{x[1+(y/x)^2]} = \frac{x}{x^2+y^2}$

Example

Hence
$J = \frac{x^2}{(x^2+y^2)^{3/2}} + \frac{y^2}{(x^2+y^2)^{3/2}} = \frac{1}{\sqrt{x^2+y^2}} = \frac{1}{r}$.
As the joint density function of X and Y is

$$f_{X,\,Y}(x,y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}$$

we see that the joint density function of $R = \sqrt{x^2 + y^2}$,
$\Theta = tan^{-1}(y/x)$, is given by

$$f_{R,\,\Theta}(r,\,\theta) = \frac{1}{2\pi} r e^{-r^2/2} \quad 0 < \theta < 2\pi, \ \ 0 < r < \infty.$$

As this joint density factors into the marginal densities for R and Θ, we obtain that R and Θ are independent random variables, with Θ being uniformly distributed over $(0, 2\pi)$ and R having the Rayleigh distribution with density

$$f_R(r) = re^{-r^2/2} \quad 0 < r < \infty.$$

If we wanted the joint distribution of R^2 and Θ, then, as the transformation $d = h_1(x, y) = x^2 + y^2$ and $\theta = h_2(x, y) = \tan^{-1}(y/x)$ has a Jacobian

$$J = \det \begin{bmatrix} 2x & 2y \\ \frac{-y}{x^2+y^2} & \frac{x}{x^2+y^2} \end{bmatrix} = 2$$

we see that

$$f_{D, \Theta}(d, \theta) = \frac{1}{2} e^{-d/2} \frac{1}{2\pi} \quad 0 < d < \infty, \ \ 0 < \theta < 2\pi.$$

Therefore, R^2 and Θ are independent, with R^2 having an exponential distribution with parameter $\beta = 2$.

Box-Muller Approach

The preceding result can be used to simulate Normal random variables by making a suitable transformation on uniform random variables. Let U_1 and U_2 be independent random variables each uniformly distributed over $(0, 1)$. We will transform U_1, and U_2 into two independent unit Normal random variables X_1 and X_2 by first considering the polar coordinate representation (R, Θ) of the random vector (X_1, X_2).

From the above, R^2 and Θ will be independent, and, in addition, $R^2 = X_1^2 + X_2^2$ will have an exponential distribution with parameter $\beta = 2$. But $-2log(U_1)$ has such a distribution since, for $x > 0$,

$$
\begin{aligned}
P[-2log(U_1) < x] &= P\left[logU_1 > -\tfrac{x}{2}\right] \\
&= P\left[U_1 > e^{-x/2}\right] \\
&= 1 - e^{-x/2}
\end{aligned}
$$

Also, as $2\pi U_2$ is a uniform $(0, 2\pi)$ random variable, we can use it to generate Θ. That is, if we let

$$R^2 = -2logU_1$$

$$\Theta = 2\pi U_2.$$

Then R^2 can be taken to be the square of the distance from the origin and Θ as the angle of orientation of (X_1, X_2). As $X_1 = R\cos\Theta$, $X_2 = R\sin\Theta$, we obtain that

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$

$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

are independent standard Normal random variables.