Transformations of Standard Uniform Distributions

We have seen that the R function `runif` uses a random number generator to simulate a sample from the standard uniform distribution $\mathsf{UNIF}(0,1)$. All of our simulations use standard uniform random variables or are based on transforming such random variables to obtain other distributions of interest. Included in the R language are some functions that implement suitable transformations. For example, `rnorm`, `rexp`, `rbeta`, and `rbinom` simulate samples from normal, exponential, beta, and binomial distributions, respectively. Also, the function `sample` is based on simulated realizations of $\mathsf{UNIF}(0,1)$.

A systematic study of the programming methods required to transform uniform distributions into other commonly used distributions involves technical details be beyond the scope of this book. (For a more extensive treatment, see Chapter 3 of Fishman (1996).) However, if you are going do simulations and trust the results, we feel you should have some idea how such transformations are accomplished—at least in a few familiar and elementary cases. The purpose of this section is to provide some of the basic theory and a few simple examples of transformations from uniform distributions to other familiar distributions. Also, this discussion provides the opportunity for a brief review of some distributions we will use later on.

EXAMPLE 1. A real function (transformation) of a random variable is again a random variable. For example, if $U \sim \mathsf{UNIF}(0,1)$, then the linear function $X = g(U) = 4U + 2$ is a random variable uniformly distributed on the interval $(2,6)$. That is, $X \sim \mathsf{UNIF}(2,6)$. The transformation g stretches the distribution of U by a factor of 4 and then shifts it two units to the right. Recalling that $F_U(u) = P\{U \le u\} = u$, for $0 < u < 1$, we have the following formal demonstration. For $2 < x < 6$,

$$\begin{aligned} F_X(x) &= P\{X \le x\} = P\{g(U) \le x\} = P\{4U + 2 \le x\} \\ &= P\{g^{-1}(X) \le g^{-1}(x)\} = P\{U \le (x-2)/4\} = (x-2)/4. \end{aligned}$$

Because the density function of a random variable is the derivative of its cumulative distribution function (CDF), we see that, for $2 < x < 6$, the density function of X is

$$f_X(x) = dF(x)/dx = x/4,$$

which is the density function of $\mathsf{UNIF}(2,6)$.

In R, the second and third parameters of the function `runif` specify the left and right endpoints, respectively, of $\mathsf{UNIF}(\theta_1, \theta_2)$, the uniform distribution on the interval (θ_1, θ_2). Thus each of the statements `4*runif(10) + 2`, `4*runif(10, 0, 1) + 2`, and `runif(10, 2, 6)` simulates 10 observations from $\mathsf{UNIF}(2,6)$. PROBLEM 1 asks you to consider a more general version of this example. ♢

In EXAMPLE 1, we have found the CDF of the transformed random variable, and then used the CDF to find its density function. This method works in a large variety of situations. Next, we see that a particular nonlinear transformation of a standard uniform random distribution is a member of the beta family of distributions. We leave the formal demonstration to PROBLEM 2 and use a simulation and graphics to illustrate the effect of the transformation.

EXAMPLE 2. Suppose $U \sim \mathsf{UNIF}(0, 1)$ and $X = \sqrt{U}$. Then $P\{0 < X < 1\} = 1$. Also, because the square root of a number in $(0, 1)$ is larger than the number itself, we know intuitively that the distribution of X must concentrate its probability toward the right end of $(0, 1)$. Specifically, the method of EXAMPLE 1 shows that X has the cumulative distribution function $F_X(x) = x^2$, and the density function $f_X(x) = 2x$, for $0 < x < 1$. Recall that if $Y \sim \mathsf{BETA}(\alpha, \beta)$ then its density function is

$$f_Y(y) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, y^{\alpha-1}(1-y)^{\beta-1},$$

for $0 < y < 1$, and positive parameters α and β. Here Γ denotes the gamma function, which has $\Gamma(n+1) = n!$ for positive integer n, and may be evaluated more generally in R using `gamma`. Thus $X = \sqrt{U} \sim \mathsf{BETA}(2, 1)$.

The following simulation shows what happens when one takes the square root of randomly chosen points u in the ten intervals $(0, 0.1], (0.1, 0.2]$, through $(0.9, 1)$. In R, the names of density functions of programmed distributions begin with the letter `d`: thus the functions `dunif` and `dbeta` in the code above.

```
set.seed(1212)
m = 10000;  u = runif(m);  x = sqrt(u)
par(mfrow=c(1,2))
  hh = seq(-.1, 1.1, length=1000); cutp = seq(0, 1, by = .1)
  hist(u, breaks=cutp, prob=T, ylim=c(0,2), xlim=c(-.1, 1.1))
    lines(hh, dunif(hh), lwd=2)
  hist(x, breaks=sqrt(cutp), prob=T, xlim=c(-.1, 1.1))
    lines(hh, dbeta(hh, 2, 1), lwd=2)
par(mfrow=c(1,1))
```

Graphical results are shown in FIGURE A. Each bar in each histogram represents about a thousand points, representing one tenth of the total probability. Density functions of $\mathsf{UNIF}(0, 1)$ and $\mathsf{BETA}(2, 1)$ are superimposed on their respective histograms.

By taking different powers of a standard uniform random variable one can obtain random variables with distributions $\mathsf{BETA}(\alpha, 1)$ (see PROBLEM 2). More intricate methods are required to sample from some other members of the distribution family $\mathsf{BETA}(\alpha, \beta)$ (see PROBLEM 3). Optimal methods for all cases are available in R as the function `rbeta`. Thus either of the statements `sqrt(runif(10))` or `rbeta(10, 2, 1)` could be used to simulate 10 observations from $\mathsf{BETA}(2, 1)$, but the latter code is more convenient because it can be used for any member of the beta family. $\diamondsuit$

Now we summarize what we have seen so far.

- In EXAMPLE 1, the CDF of X is $F_X(x) = (x - 2)/4$, for $2 < x < 6$. The inverse of the CDF is called the **quantile function**. Here it is

$F_X^{-1}(u) = 2 + 4u$, obtained by solving $F_X(x) = u$ for x in terms of u. This is the function g we used to transform $U \sim \mathsf{UNIF}(0,1)$ to get the random variable $X = g(U) \sim \mathsf{UNIF}(2,6)$.

- In EXAMPLE 2, the CDF $F_X(x) = x^2$, is used to obtain $f_X(x) = 2x$, for $0 < x < 1$. Thus X has quantile function $F_X^{-1}(u) = \sqrt{u}$, which is the function g used to transform $U \sim (0,1)$ to get the the random variable $X \sim \mathsf{BETA}(2,1)$.

Suppose we want to simulate values from a distribution whose quantile function is known. A general principle is that this quantile function is the function g such that $X = g(U)$ has the desired distribution, where $U \sim \mathsf{UNIF}(0,1)$. Specifically, in the next example, we want to simulate observations $X \sim \mathsf{EXP}(1)$, the exponential distribution with rate 1. Accordingly, we find the quantile function of $\mathsf{EXP}(1)$ and use it to transform observations from $\mathsf{UNIF}(0,1)$.

EXAMPLE 3. Throughout this example let $x > 0$ and $0 < u < 1$. We wish to simulate observations from the distribution $\mathsf{EXP}(1)$, which has density function $f(x) = e^{-x}$ and CDF $F(x) = 1 - e^{-x}$. Solving $u = 1 - e^{-x}$ for x in terms of u, we have the quantile function $F^{-1}(u) = -\ln(1-u)$. Thus $X = -\ln(1-U) \sim \mathsf{EXP}(1)$. Because $1 - U \sim \mathsf{UNIF}(0,1)$ it is simpler to simulate observations from this exponential distribution as $X = -\ln U$ (see PROBLEM 1).

The following R code demonstrates that a histogram of 100 000 observations generated in this way very nearly fits the density function of $\mathsf{EXP}(1)$, as seen in FIGURE B. Furthermore, the mean and standard deviation of the simulated values are both nearly 1, which is the mean and standard deviation of the distribution $\mathsf{EXP}(1)$.

```
set.seed(1234)
m = 100000;  u = runif(m);  x = -log(u)
hist(x, prob=T)
  xx = seq(0, max(x), length=100)
  lines(xx, dexp(xx, 1), lwd=2)
mean(x);  sd(x)

> mean(x);  sd(x)
[1] 0.9988505
[1] 0.9984966
```

For most purposes, any of the following statements could be used to sample 10 observations from $\mathsf{EXP}(1)$: `-log(runif(10))`, `qexp(runif(1), 1)`, or `rexp(10, 1)`. The second statement works because `qexp` (with second parameter 1) is the quantile function of $\mathsf{EXP}(1)$. (PROBLEM 4 uses the quantile transformation to sample from $\mathsf{EXP}(1/2)$.) However, the method using `rexp` is preferable because it uses an algorithm that is technically superior to our log-transform method, especially in its treatment of very large simulated values. $\diamondsuit$

So far, all of our examples have dealt with continuous distributions. Now we turn to an example where we sample from a binomial distribution.

EXAMPLE 4. According to genetic theory the probability that any one offspring of a particular pair of guinea pigs will have straight hair is 1/4. Suppose we want to simulate births of six offspring. That is, we want to simulate one realization of $X \sim \mathsf{BINOM}(6, 1/4)$. One way to do this is to simulate six observations from $\mathsf{UNIF}(0, 1)$. The probability that any one of these uniform observations is less than 1/4 is 1/4. So X can be simulated as the sum of six logical variables, where `FALSE` is interpreted as 0 and `TRUE` as 1: `sum(runif(6) < 1/4))`. The `sample` function is also programmed to use `runif`. So `sum(sample(c(0,1), 6, repl=T, prob=c(3/4, 1/4))` is an equivalent way to simulate X as a sum.

Because R defines the quantile function for a discrete random variable in just the right way, one can use the quantile function approach: `qbinom(runif(1), 6, 1/4)`. The the second method has the advantage of requiring only one random value from $\mathsf{UNIF}(0, 1)$, while the first—somewhat wastefully—requires six. In this case, it turns out that the quantile transform method is exactly equivalent to `rbinom(1, 6, 1/4)`.

For a discrete random variable X, R defines $F_X^{-1}(u)$ as the minimum of the values x such that $F_X(x) \geq u$. The left panel of FIGURE C shows the CDF of $\mathsf{BINOM}(6, 1/4)$, where the vertical reference segments (dotted) represent individual binomial probabilities $P\{X = i\}, i = 0, 1, \ldots, 6$. The right panel shows the corresponding quantile function, where the horizontal segments of the function (heavy) represent these same probabilities. PROBLEM 5 shows R code for a simplified version of this figure. ♢

In practice, when available, it is best to use random functions programmed into R (for example, `rbeta`, `rbeta`, `rbinom`) because they implement algorithms that are fast and accurate. However, some useful distributions are not programmed into the base package of R. It may be possible to use the quantile transformation of standard uniform to simulate observations from such a distribution.

EXAMPLE 5. The Pareto family of distributions is sometimes useful in economics, actuarial science, geology, and other sciences, but it is not included the base package of R. One member of this family has density function $f(x) = 3/x^4$ and CDF $F(x) = 1 - x^{-3}$, for $x > 1$; mean 1.5 and variance 0.75. The following R code simulates a sample of 5000 observations from this distribution.

```
set.seed(123)
m = 5000;  kap = 3
xx = seq(1, 10, length=1000)
pdf = kap/xx^(kap+1)
x = (1 - runif(m))^(-1/kap)
mean(x);  var(x)
cutp=seq(0, max(x)+.5, by=.5)
hist(x[x<10], prob=T);  lines(xx, pdf)

> mean(x);  var(x)
[1] 1.492558
[1] 0.7048778
```

FIGURE D shows a histogram of the results (except for the six observations that exceed 10) along with the density function. ♢

Transformations Involving Standard Normal Distributions

Normal distributions play an important role in probability and statistics, and so it is important to know how to simulate samples from normal distributions. The R function `rnorm` samples from the standard normal distribution. At the end of this section we indicate how to transform standard uniform observations into standard normal ones. In the first example below, we look at some relationships between standard normal and other distributions.

EXAMPLE 6. If $Z \sim \mathsf{NORM}(0,1)$, then $X = Z^2 \sim \mathsf{CHISQ}(1)$, that is, the chi-squared distribution with one degree of freedom. Also, if Z_1 and Z_2 are independently standard normal, then $Q = Z_1^2 + Z_2^2 \sim \mathsf{CHISQ}(2) = \mathsf{EXP}(1/2)$, where $\mathrm{E}(Q) = 2$ and $\mathrm{V}(Q) = 4$. These are standard results from probability theory used in mathematical statistics. Formal proofs, not shown here, use transformation theory or moment generating functions. We illustrate these results via simulations.

```
set.seed(12)
m = 10000;  z1 = rnorm(m);  z2 = rnorm(m)
x = z1^2;  q = z1^2 + z2^2
par(mfrow=c(2,1))
 mx=max(x, q);  xx = seq(0, mx, length=1000)
 hist(x, prob=T, ylim=c(0,.7), xlim=c(0, mx), main="CHISQ(1)")
  lines(xx, dchisq(xx, 1), lwd=2)
 hist(q, prob=T, ylim=c(0,.7), xlim=c(0, mx), main="CHISQ(2)")
  lines(xx, dexp(xx, 1/2))
  lines(xx, dchisq(xx, 2), lwd=2, lty="dashed")
par(mfrow=c(1,1))
mean(x);  var(x)
mean(q);  var(q)

> mean(x);  var(x)
[1] 1.018115
[1] 2.054014
> mean(q);  var(q)
[1] 2.023093
[1] 4.118364
```

Graphical results are shown in FIGURE E. In the lower panel, the double plotting with two line styles shows that the density functions of $\mathsf{EXP}(1/2)$ and $\mathsf{CHISQ}(2)$ are the same. ♢

The following example illustrates the idea behind the most common method of generating standard normal random variables from standard uniform random variables.

EXAMPLE 7. Suppose an archer shoots arrows at a distant target. She is aiming at the bull's eye, which we take to be the origin of a plot, but the hits are subject to random error. We model the vertical and horizontal displacements from the origin as independent standard normal random variables

Z_1 and Z_2. We know from EXAMPLE 6 that each arrow hits at a random distance $D = \sqrt{Z_1^2 + Z_2^2}$ from the origin, where $D^2 = Q \sim \mathsf{EXP}(1/2)$.

Now consider a line through the arrow's position to the origin, and the angle Θ it makes with the positive Z_1-axis measured in degrees counterclockwise. Intuitively, it seems that $\Theta \sim \mathsf{UNIF}(0, 360)$, which is illustrated by the following simulation. In the code below, the arctangent takes values between -90 and 90 degrees. Adding 180 degrees precisely when Z_1 is negative completes the circle from -90 to 270 degrees, and taking the resulting value modulo 360 (code `%%`) adjusts the values to lie in the interval $(0, 360)$. The resulting graph is shown in Figure F.

```
set.seed(1212)
m = 10000
z1 = rnorm(m);  z2 = rnorm(m)

par(mfrow=c(2,1))
# squared distance from origin
  d2 = z1^2 + z2^2
  hist(d2, prob=T)
    dd = seq(0, max(d2), length=1000)
    lines(dd, dchisq(dd, 2))

# angle in degrees (counterclockwise from right)
  th = ((180/pi)*atan(z1/z2) + 180*(z1<0)) %% 360
  hist(th, prob=T)
    tt = seq(0, 360, length = 1000)
    lines(tt, dunif(tt, 0, 360))
par(mfrow=c(1,1))
```

Thus the position of the hit can be modeled in polar coordinates by using two standard uniform random variables:

- The angle can be simulated as a linear transformation of a simulated observation from a standard uniform distribution (see EXAMPLE 1 and PROBLEM 1).
- The distance from the origin is the square root of an exponential random variable, and that exponential random variable can be obtained as a log transformation of a standard uniform (see EXAMPLE 3 and PROBLEM 4).

Conversion from polar to rectangular coordinates reverts to the two independent standard normal random variables with which we started. This procedure of simulating two independent standard normal observations from two simulated independent standard uniform ones is known as the Box-Muller transformation. It is explored further in PROBLEM 9. $\diamondsuit$

PROBLEMS

1. *General linear transformation.* Let $U \sim \mathsf{UNIF}(0,1)$ and $X = aU + b$, where $a \neq 0$. Use the method of EXAMPLE 1 to find the distribution of X. In particular, what is the distribution of $Y = 1 - U$?

2. Let $U \sim \mathsf{UNIF}(0,1)$. Use the method of EXAMPLE 1 to find the density function of X in parts (a) and (b). Be sure to specify the interval on which each density function takes positive values.

a) As in EXAMPLE 2, let $X = \sqrt{U}$. Show formally that $X \sim \mathsf{BETA}(2,1)$.

c) In general, if $X = U^{1/\alpha}$, where $\beta > 0$, show that $\mathsf{EXP} \sim \mathsf{BETA}(\alpha, 1)$.

c) Modify the R code in EXAMPLE 2 to illustrate part (b) with $\alpha = 2$. Write a suitable caption for the resulting figure.

3. (Intermediate) *Acceptance-rejection sampling.* Sometimes the quantile function is difficult or impossible to find or to express in closed form. Here we explore a possible alternative method of sampling from such a distribution. Suppose we wish to sample from the distribution $\mathsf{BETA}(2,2)$.

a) Sketch the density function of this distribution, and show that the rectangle with diagonal corners at $(0,0)$ and $(1,1.5)$ contains the non-negative part of the density curve. Also, find the mean and variance of this distribution.

b) We generate random points within the rectangle of part (a), accepting those that fall beneath the density function. The x-coordinates of the accepted points form the simulated sample. Run the R code below and explain the purpose of each statement. For your run, what is the size of the simulated sample?

```
al = 2;   be = 2;  m = 600000
h = runif(m);  v = runif(m, 0, 1.5)
x = h[v < dbeta(h, al, be)]
hist(x, prob=T)
xx = seq(0, 1, length=1000)
lines(xx, dbeta(xx, al, be))
length(x);  mean(x);  var(x)
```

4. Consider the distribution $\mathsf{EXP}(1/2)$. That is, the exponential distribution with rate $1/2$ and mean 2.

a) Write the density function of this distribution. Find its quantile function.

b) Simulate 10 000 observations from this distribution.

c) Modify the R code of EXAMPLE 3 to make a histogram of the observations in part (b). making a figure similar to FIGURE B.

5. Run the R code below to make a somewhat simplified version of FIGURE C. Explain the code. (In `plot`, paramter `type="s"` indicates a step graph.)

```
par(mfrow=c(1,2))
 xx = seq(-.5, 6.5, length=1000)
 plot(xx, pbinom(xx, 6, 1/4), type="s",
   xlab="Successes", ylab="CDF")
 qq = seq(0, 1, length=1000)
 plot(qq, qbinom(qq, 6, 1/4), type="s",
   xlab="Cum Prob", ylab="Quantile")
par(mfrow=c(1,1))
```

6. Consider the distribution with density function $f(x) = 0.8 + 1.2x$, for $0 < x < 1$. Use integration to find the CDF, and then find the quantile function. Use the quantile transformation to simulate 100 000 observations from this distribution. Compare the mean and standard deviation of your observations with the mean and standard deviation of this distribution.

7. If $Z_1, \ldots, Z_5$ are independently distributed standard normal random variables, then $Q = Z_1 + \cdots + Z_5 \sim \mathsf{CHISQ}(5)$. Simulate 10 000 observations from this distribution using the R function `rnorm`. Make a histogram of the resulting observations and superimpose the density function of $\mathsf{CHISQ}(5)$ on it.

8. In EXAMPLE 7, suppose one arrow hits 2 units to the right of the origin (bull's eye) and 1 unit above, so that $Z_1 = 2$ and $Z_2 = 1$. Show that $D = 1.73$ and $\Theta = 26.6$ degrees. What if $Z_1 = -2$ and $Z_2 = 1$?

9. *Box-Muller Transformation.* Let U_1 and U_2 be independent observations from $\mathsf{UNIF}(0, 1)$. Then transform the joint distribution of (U_1, U_2) to the disjoint distribution of (Z_1, Z_2) according to the Box-Muller transformation, expressed by the following equations.

$$Z_1 = \sqrt{-2 \ln U_1} \sin 2\pi U_2,$$
$$Z_2 = \sqrt{-2 \ln U_1} \cos 2\pi U_2.$$

The R code below uses this transformation to simulate 2500 pairs of standard normal values (5000 altogether). These 2500 points are plotted in the left panel of the resulting figure. Then 2500 pairs of standard normal values are generated with the R function `runif` and plotted in the right panel. Because `rnorm` also also implements the Box-Muller transformation it is not surprising that the two plots are very similar.

```
set.seed(1212)
m = 2500;  u1 = runif(m);  u2 = runif(m)
z1.BoxM = sqrt(-2*log(u1))*sin(2*pi*u2)
z2.BoxM = sqrt(-2*log(u1))*cos(2*pi*u2)
z1.rnorm = rnorm(m);  z2.rnorm = rnorm(m)
par(mfrow=c(1,2))
  plot(z1.BoxM, z2.BoxM, pch=20, xlim=c(-4,4), ylim=c(-5,5))
  plot(z1.rnorm, z2.rnorm, pch=20, xlim=c(-4,4), ylim=c(-5,5))
par(mfrow=c(1,1))
```

10. Let $Z = U_1 + U_2 + \cdots + U_{12} - 6$, where the U_i are independent and identically distributed (iid) as $\mathsf{UNIF}(0,1)$.

a) Show that $\mathrm{E}(U_i) = 1/2$ and $\mathrm{V}(U_i) = 1/12$. Thus argue that $\mathrm{E}(Z) = 0$ and $\mathrm{V}(Z) = 1$. Because Z is based on a sum of iid random variables the Central Limit Theorem, discussed in the next chapter, indicates that Z may be nearly normal—and, in view of its mean and variance, nearly standard normal. As we see below, the agreement with a standard normal distribution is reasonably good. Before computers could easily do transcendental functions this method was sometimes used to get one (roughly) standard normal observation from 12 standard uniform observations.

b) Run the R code below to simulate 1000 "standard normal" random variables and test agreement with a normal distribution. Reasonably good fit of the histogram to the density curve in the left panel of the resulting figure is a good indication of normality. A more precise assessment of the fit to normality appears in the right panel. If the points in the quantile-quantile (Q-Q) plot are essentially linear that indicates a good fit to normality. The P-value of a Kolmogorov-Smirnov test of goodness-of-fit is also shown in the right panel; a value above 5% indicates that no significant evidence against normality has been found. Make several different runs and report your findings.

```
m = 1000;  n = 12
u = runif(m*n);  DTA = matrix(u, nrow=m)
z = rowSums(DTA) - 6
par(mfrow=c(1,2))
mn = min(-3.8, min(z));  mx = max(3.8, max(z))
hist(z, prob=T, ylim=c(0,.42), xlim=c(mn,mx), col="wheat")
  zz = seq(mn, mx, length=100)
  lines(zz, dnorm(zz), col="blue", lwd = 2)
qqnorm(z, datax=T, ylim=c(mn, mx))
  pval = round(ks.test(z, pnorm)$p,2)
  text(1.5, -2.5, paste("KS.P-val =",pval))
par(mfrow=c(1,1))
```

c) The fit assessed in part (b) cannot be absolutely perfect. What is the probability that Z, as simulated by the program in part (b), will lie in the interval $[-6, 6]$? What is the probability a true standard normal observation will lie in this interval?