Acceptance-Rejection Method

In many cases the direct method of evaluating the inverse cdf at a random uniform will not work because either it's difficult to find the inverse, or it does not uniquely exists, such as in the case of multivariate distributions. An alternative is the **acceptance-rejection method**.

Suppose we wish to draw realizations of a random variable X that has pdf f. The technique is to identify a random variable Y that can be sampled, and has a pdf g that is in some sense close to f. It is useful when sampling from $f(\cdot)$ distribution is hard, but sampling from distribution $g(\cdot)$ is easy.

Procedures

Specifically:

1. Find Y with pdf g satisfying $f(t) < t(x) \equiv c \cdot g(t)$ whenever $f(t) > 0$.

The value c can be any positive number satisfying the inequality, but the smaller the better.

2. Generate Y from the density g.
3. Generate U from a Unif(0,1) distribution.
4. If $U < \frac{f(Y)}{cg(Y)}$ accept Y and set $X = Y$. Otherwise repeat steps 2-4.

Proof

- First, let's consider the probability of acceptance.

$$P[\text{accept}] = \int_{-\infty}^{\infty} P[\text{accept} \mid y] g(y) dy = \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y) dy$$

$$= \int_{-\infty}^{\infty} \frac{f(y)}{c} dy = \frac{1}{c}$$

It obviously helps if c is very close to 1, because then drawing from f is very much like drawing from g.

Let's work through a little proof that when X is generated in this way,

it has the correct cdf $F(x) = \int_{-\infty}^{x} f(t) dt$.

Of course, we only need to consider the accepted values.

$$P[X \le x \mid \text{accept}] = P[Y \le x \mid \text{accept}]$$

because we set $Y = X$.

$$P[Y \le x \mid \text{accept}] = P[Y \le x, \text{accept}] / P[\text{accept}]$$

$$= P[U \le \frac{f(Y)}{cg(Y)}, Y \le x] / (1/c)$$

$$= c\int_{-\infty}^{x} P[U \le \frac{f(y)}{cg(y)}]g(y)dy = c\int_{-\infty}^{x} \frac{f(y)}{cg(y)}g(y)dy$$

$$= \int_{-\infty}^{x} f(y)dy = F(x)$$

Let's look at the examples.

Example 1.

Generate a r.v. with pdf $f(x) = 60x^3(1-x)^2,\ 0 \le x \le 1.$

Can't invert this analytically.

Note that the maximum occurs at $x = 0.6$ and $f(0.6) = 2.0736.$

Let $c = 2.0736$ (which isn't actually very efficient)

Let Y be Unif(0,1) so that $g(y) = 1$ for $y \in (0, 1)$.

Here we see both X and Y have the same support, the interval (0,1)

Then,

$$f(y)/cg(y) = 60y^3(1-y)^2/2.0736.$$

E.g. If we generate $u = 0.13$ and $Y = 0.25$, then it turns out that

$U \le \frac{f(Y)}{cg(Y)} = 60Y^3(1-Y)^2/2.0736,$ so we take $X \leftarrow 0.25$.

Example 2.

Generate a standard half-normal r.v., with pdf $f(x) = \sqrt{\frac{2}{\pi}}e^{-x^2/2}, \quad x \ge 0.$

Let Y be the exp(1) density so that $g(y) = e^{-x}$ for $y \ge 0$.

Here we see both X and Y have the same support, the interval (0,1).

Let $c = \sqrt{\frac{2e}{\pi}} = 1.3155$

Then, $f(y)/cg(y) = e^{-(y-1)^2/2}$.

We should accept 1/1.3155 of the time, so to get a sample of about 10000 from X ,let's draw about 1.3155×10000 from Y .

E.g. If we generate $u = 0.13$ and $Y = 0.25$, then it turns out that

$U \le \frac{f(Y)}{cg(Y)} = e^{-(Y-1)^2/2},$ so we take $X \leftarrow 0.25$.

Example 3.

Let X be $beta(2, 2)$ so that $f(x) = 6x(1 - x)$ for $0 < x < 1$.

Let Y be $Unif(0, 1)$ so that $g(y) = 1$ for $y \in (0, 1)$.

Here we see both X and Y have the same support, the interval (0,1).

Note that the maximum occurs at $x = 0.5$ and $f(0.6) = 3/2.$

Let $c = 3/2$

We should accept 2/3 of the time, so to get a sample of about 10000 from X , let's draw about 15000 from Y .

E.g. If we generate $u = 0.13$ and $Y = 0.25$, then it turns out that

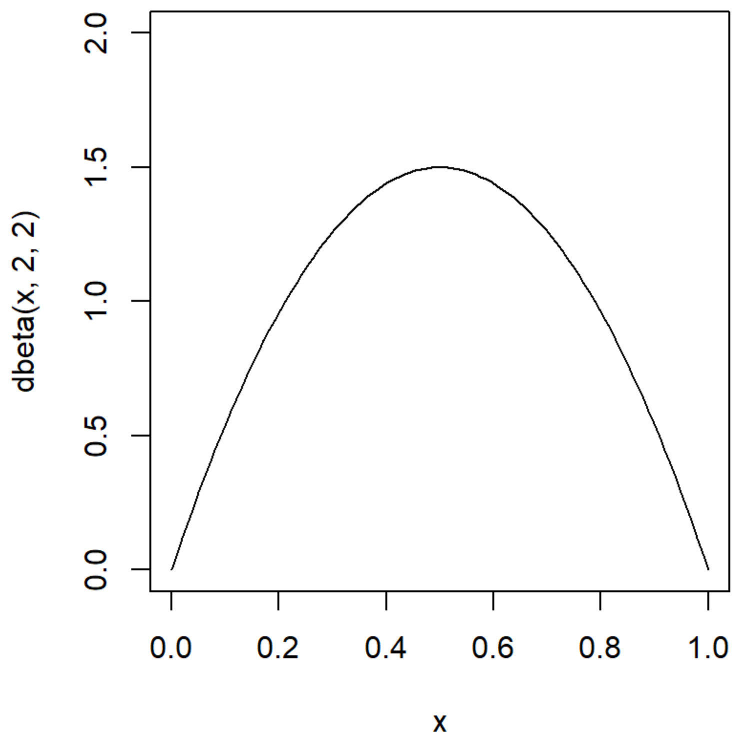
$U \leq \frac{f(Y)}{cg(Y)} = 6Y(1 - Y)/1.5,$ so we take $X \leftarrow 0.25$.

R: Generate about 100000 samples from Beta (2,2)

Let's simulate a Beta distribution with parameters 2 and 2. I.e., $f(x) = 6x(1 - x)$.

1) First draw the beta distribution

```
par(pty = "s")
x <- seq(0, 1, length=100)
plot(x, dbeta(x,2,2),type="l",ylim=c(0,2))
```



Whiat is the maximum of $f(x)$?

In a plot of the beta distribution with parameters 2 and 2 we can see that the $f(x)$ never goes above 1.5.

For this reason we choose to scale the uniform distribution $g(Y)$ by multiplying it by 1.5.

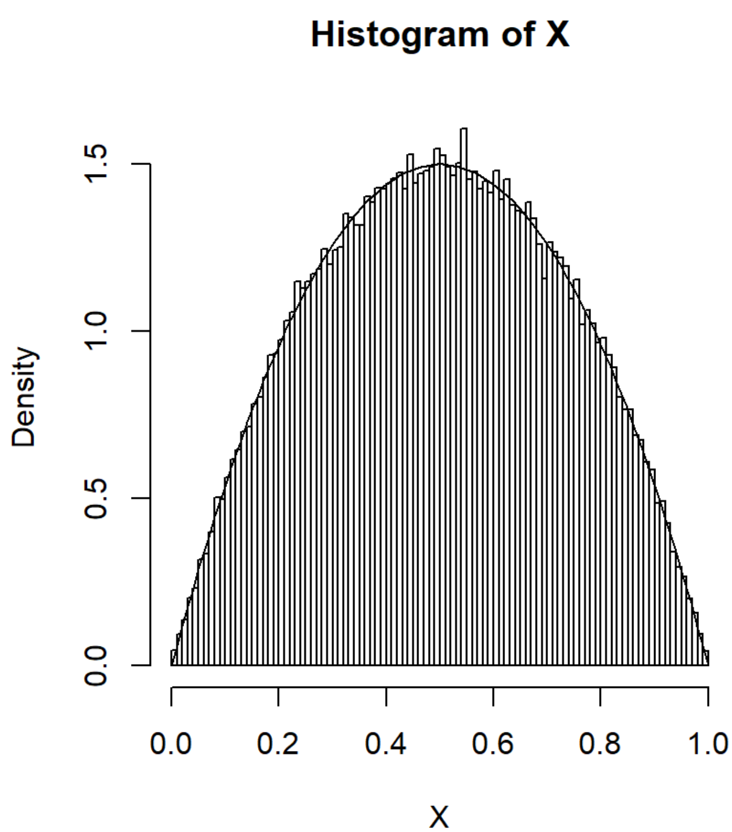
2) Implement rejection sampling to obtain 100000 samples from X .

We should accept 2/3 of the time, so to get a sample of about 100000 from X , let's draw about 150000 from Y .

```
accept = c()
Y=runif(150000)
U=runif(150000)
ratio=6*Y*(1-Y)/(3/2)
X=Y[U < ratio]
accept=ifelse(U < ratio,'Yes','No')
T = data.frame(Y, accept =
factor(accept, levels= c('Yes','No')))
```

3) Plot the results along with the true distribution.

```
par(pty = "s")
hist(T[,1][T$accept=='Yes'], breaks =
seq(0,1,0.01), freq = FALSE,
main = 'Histogram of X', xlab = 'X')
x <- seq(0, 1, length=100)
lines(x, dbeta(x,2,2))
```



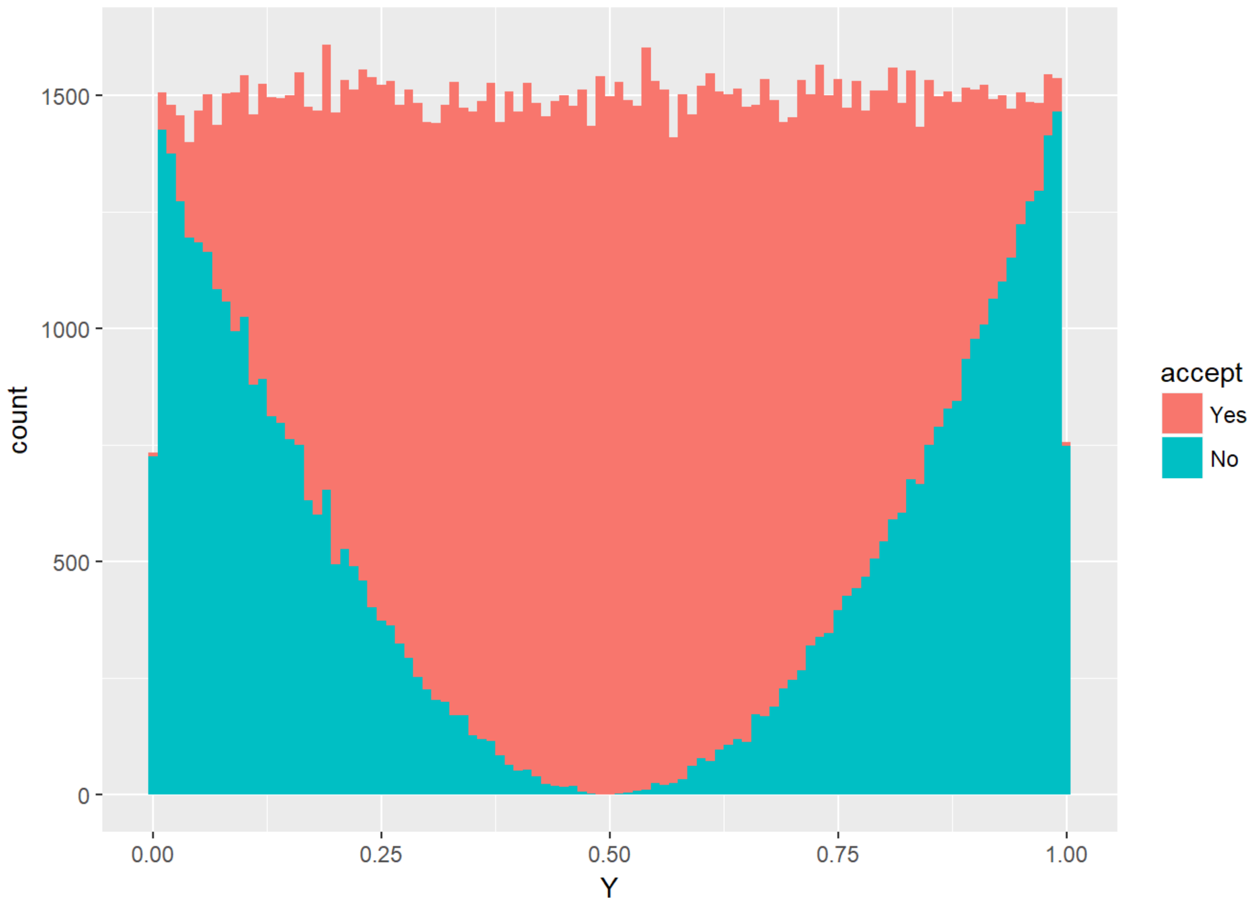
4) Let’s compare sample versus theoretical quantiles

```
p=seq(.1,.9,.1)
Qhat=quantile(X,p)
Q=qbeta(p,2,2)
round(rbind(Qhat,Q),3)
```

```
##        10%   20%   30%   40%   50%   60%   70%   80%   90%
## Qhat 0.196 0.287 0.364 0.434 0.501 0.567 0.637 0.714 0.805
## Q    0.196 0.287 0.363 0.433 0.500 0.567 0.637 0.713 0.804
```

5) Draw the stacked histogram

```
library(ggplot2)
print(qplot(Y, data = T, geom = 'histogram',fill = accept, binwidth=0.01))
```



Looking at a stacked histogram of all the sampled values together we can really see how much wasted data there are in this example.

It’s your turn

Repeat (1)–(5) to generate about 100000 samples from Beta distribution with parameter (2,3).