Simulating Random Variables

Timothy Hanson

Department of Statistics, University of South Carolina

Stat 740: Statistical Computing

R has many built-in random number generators...

Beta, gamma (also χ^2 and exponential), normal (also Student's t & Cauchy, F, log-normal), Weibull, logistic, binomial, geometric, hypergeometric, Poisson, etc.

For each distribution, R has the pdf/pmf, quantile function, cdf, and an independent random number generator.

R also has the distribution of different test statistics, e.g. Tukey's studentized range, Wilcoxin rank sum statistic, etc.

There are packages to sample multivariate normal, Wishart and inverse Wishart, multivariate t, Pareto, etc. Google is your friend.

We will discuss methods for simulating random variables anyway for when you run into non-standard ones.

Everything starts with uniform...

Simulating $U_1, U_2, U_3, \cdots \stackrel{iid}{\sim} U(0,1)$ is the main building block for all that follows.

- Random uniform generators are not random. In R try `set.seed(1)` then `runif(10)` several times.
- They are said to be "pseudo random" – they satisfy certain statistical tests we'd expect independent uniforms to pass, e.g. Kolmogorov-Smirnov. Look up "Diehard tests" in Wikipedia.
- Try `?RNG` to see what R is capable of and the default.
- Historically common: conguential generators, see pp. 72–75.
- R sets the seed by the current time and process ID.

Inverse transformation

- Important result (p. 39): $U \sim U(0,1)$, and $X = F^{-}(U)$ implies $X \sim F(\cdot)$. The generalized inverse of a non-decreasing cdf $F(\cdot)$ is $F^{-}(u) = \inf\{x : F(x) \geq u\}$.
- If $F(\cdot)$ is monotone increasing and continuous over its support, representing a continuous random variable, $F^{-}(u) = F^{-1}(u)$. You just need to find the inverse function. Proof of result straightforward (board).
- (p. 44) If $F(u)$ is a "stair function" with jumps at $x_1, x_2, x_3, \ldots$, representing a discrete random variable, then $U \sim U(0,1)$ and $X = x_j \Leftrightarrow F(x_{j-1}) < U < F(x_j)$ implies $X \sim F(\cdot)$. Here, $F(x_0) = 0$.
- `sample` automates this last result; `ddiscrete`, `pdiscrete`, `qdiscrete`, and `rdiscrete` are in the `e1071` package.

Examples...

Inverses can be easily derived in closed-form:

- $\exp(\lambda)$ (Ex. 2.5, p. 39)
- Weibull(α, β)
- Pareto
- Cauchy

Inverses not available in closed-form:

- Normal (although R uses inversion as the default!)
- beta
- gamma
- F (is there *another way*?)

Section 2.2 Tricks and relationships...

- Lots of clever tricks, relationships among variables, etc. on pp. 42–46:
- Box-Muller for normal r.v. (can be implemented in `rnorm`),
- Poisson via waiting times,
- beta from order statistics,
- gamma from beta & exponential, etc.
- These can be used but are often not optimal.
- There are a few that are good for MCMC (coming up).

Sampling multivariate normals

Want to simulate $\mathbf{y} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Recall if $z_1, \ldots, z_p \stackrel{iid}{\sim} N(0,1)$, $\mathbf{z} = (z_1, \ldots, z_p)'$, $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times p}$ then

$$\mathbf{a} + \mathbf{A}\mathbf{z} \sim N_m(\mathbf{a}, \mathbf{A}\mathbf{A}').$$

A Cholesky decomposition produces a $\mathbf{C}$ such that $\boldsymbol{\Sigma} = \mathbf{C}'\mathbf{C}$ where $\mathbf{C}$ is upper triangular. Thus

$$\boldsymbol{\Sigma} = \mathbf{C}'\mathbf{C} \Rightarrow \mu + \mathbf{C}'\mathbf{z} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Some other multivariate distributions

- To sample $(q_1, \ldots, q_p) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_p)$ take $y_i \stackrel{ind.}{\sim} \Gamma(\alpha_i, 1)$ and $q_i = \frac{y_i}{\sum_{j=1}^k y_j}$ for $i = 1, \ldots, p$.
- To sample $\boldsymbol{\Sigma} \sim \text{Wishart}_p(k, \mathbf{S}_0)$ the def'n

$$\boldsymbol{\Sigma} = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{x}_i', \quad \mathbf{x}_1, \ldots, \mathbf{x}_k \stackrel{iid}{\sim} N_p(\mathbf{0}, \boldsymbol{\Sigma}),$$

  is impractical when k is large. Odell, and Feiveson (1966, JASA) give an efficient method based on χ^2 and normal r.v.
- To sample $\mathbf{n} \sim \text{mult}(n, \mathbf{q})$, independently sample discrete $Y_1, \ldots, Y_n$ where $P(Y_i = k) = q_k$ for $k = 1, \ldots, p$ and set

$$n_k = \sum_{i=1}^{n} I\{Y_i = k\}, \; k = 1, \ldots, p.$$

Examples: multivariate normal; Dirichlet.

Some other multivariate distributions

- There are algorithms to simulate many other multivariate distributions (e.g. multivariate t); Google is your friend.
- R has `rWishart` and `rmultinom`, `rdirichlet` is in `MCMCpack`, `rmvnorm` is in `mvtnorm`, etc. Many more versions of all of these floating around different packages as well as functions to evaluate the pdf/pmf/cdf, etc.
- IMSL is a package of numeric routines for FORTRAN 90/95 that includes various random number generators, pmf/pdf/cdf/quantile functions, etc.

Fundamental theorem of simulation

Back to univariate simulation...

Over pp. 47–50 is a general idea that can be paraphrased as follows.

To simulate from a (possibly unnormalized) density $Y \sim f(\cdot)$, we can find a density $g(x)$ such that $f(x) \leq Mg(x)$ for all x, then (a) simulate from $X \sim g(\cdot)$ and (b) accept $Y = X \Leftrightarrow$ with probability $\frac{f(X)}{Mg(X)}$. If Y not accepted repeat (a) and (b).

This is the same as $X \sim g(\cdot)$ indep. $U \sim U(0,1)$ and accepting $Y = X \Leftrightarrow U \leq \frac{f(X)}{Mg(X)}$.

Called the *accept-reject algorithm*. Read Section 2.3.2 for examples and implementation notes. In particular, the probability of accepting is $\frac{1}{M}$ when both densities are normalized.

Some simple ideas

Show (x_1, x_2) with joint density $h(x_1, x_2) = I\{0 < x_2 < g(x_1)\}$ implies $x_1 \sim g(\cdot)$. This proves the fundamental theorem of simulation.

Show if $x_1 \sim g(\cdot)$ and $x_2|x_1 \sim U(0, g(x_1))$ then the joint density is $h(x_1, x_2) = I\{0 < x_2 < g(x_2)\}$. This is how the pair (x_1, x_2) is sampled.

Finally, if $f(x) \leq Mg(x)$ for all x, then $x_1 \sim g(\cdot)$, $x_2|x_1 \sim U(0, Mg(x_1))$, and $x_2 < f(x_1) \Rightarrow x_1 \sim f(\cdot)$.

Accept-reject

Direct, unintuitive proof that it works...

$$
\begin{aligned}
P\left(Y \le x \middle| U \le \frac{f(Y)}{Mg(Y)}\right) &= \frac{P\left(Y \le x, U \le \frac{f(Y)}{Mg(Y)}\right)}{P\left(U \le \frac{f(Y)}{Mg(Y)}\right)} \\
&= \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/[Mg(y)]} \mathrm{d}u \;\, g(y) \;\, \mathrm{d}y}{\int_{-\infty}^{\infty} \int_{0}^{f(y)/[Mg(y)]} \mathrm{d}u \;\, g(y) \;\, \mathrm{d}y} \\
&= \frac{\int_{-\infty}^{x} f(y)/[Mg(y)]g(y) \;\, \mathrm{d}y}{\int_{-\infty}^{\infty} f(y)/[Mg(y)]g(y) \;\, \mathrm{d}y} \\
&= \frac{\int_{-\infty}^{x} f(y) \;\, \mathrm{d}y}{\int_{-\infty}^{\infty} f(y) \;\, \mathrm{d}y}
\end{aligned}
$$

Example in R: multimodal density on p. 50.

Envelope accept-reject

If $f(\cdot)$ is costly to evaluate we can add a lower "squeezing" function. Say

$$g_l(x) \leq f(x) \leq Mg_m(x), \text{ all } x.$$

1. $X \sim g_m(\cdot)$ indep. of $U \sim U(0,1)$;
2. accept $Y = X$ if $U \leq \frac{g_l(X)}{Mg_m(X)}$;
3. otherwise accept $Y = X$ if $U \leq \frac{f(X)}{Mg_m(X)}$.

Repeat if necessary until Y accepted. Then $Y \sim f(\cdot)$.

Only more efficient if evaluating $f(\cdot)$ is costly. See examples pp. 54–55.

Adaptive rejection sampling

A widely applicable, adaptive version of envelope accept-reject is available for (possibly unnormalized) densities $f(\cdot)$ that are *log-concave*, $\frac{d^2}{dx^2} \log f(x) < 0$ for all x; the algorithm is called *adaptive rejection sampling* (ARS).

This method iteratively builds piecewise-linear envelope functions $\log g_l(x)$ and $\log g_m(x)$ around $\log f(x)$ and performs envelope accept-reject until acceptance. The rejected values $x_1, x_2, \ldots$ are where $\log f(x)$ is evaluated. You book tersely describes the algorthm on pp. 56–57; I'll attempt to illustrate on the board. Wikipedia also has a nice explanation.

Adaptive rejection sampling

- Each rejected x_j is incorporated into the upper and lower envelopes, making them tighter *where they need to be*. Eventually g_l and g_m will be close enough to $f(x)$ to easily accept.
- Sampling from g_m is simply truncated exponential distributions; easy!
- There is a derivative-free version and a slightly more efficient version that requires $\frac{d}{dx} \log f(x)$.
- For non-log-concave densities, i.e. any $f(x)$, one can use the adaptive rejection Metropolis sampling (ARMS) algorithm; more later.
- Coding by hand is possible (see Wild & Gilks 1993, *Applied Statistics*) but a pain. Tim did it for his dissertation work.

R packages that perform ARS...

- `ars`. Requires $\frac{d}{dx} \log f(x)$.
- `MfUSampler`. Also does ARMS, slice sampling and Metropolis-Hastings w/ Gaussian proposal.
- There are others not on CRAN. Google "adaptive rejection R package".
- Also found C and FORTRAN subroutines posted.

Example: ARS for $N(0, 1)$.

Metropolis-Hastings

We will cover Metropolis-Hastings (MH) in more detail later when we discuss MCMC for obtaining Bayesian inference for $\pi(\boldsymbol{\theta}|\mathbf{x})$, but for now let's briefly introduce it as another method for simulating from (a possibly unnormalized) $f(\cdot)$.

The MH algorthim produces a *dependent sample* $Y_1, \ldots, Y_n$ from $f(\cdot)$ that, if we are careful, we can use like an *iid* sample. Or we can take simply take the last one $Y = Y_n \sim f(\cdot)$.

Metropolis-Hastings

Here's one version called an independence sampler.

(0) Initialize $Y_0 = y_0$ for some y_0. Then for $j = 1, \ldots, n$ repeat (1) through (3):

(1) Generate $X \sim g(\cdot)$ indep. of $U \sim U(0,1)$;

(2) compute $\rho = 1 \wedge \frac{f(X)g(Y_{j-1})}{f(Y_{j-1})g(X)}$;

(3) if $U \leq \rho$ accept $Y_j = X$ otherwise $Y_j = Y_{j-1}$.

With positive probability successive values can be tied! Algorithm efficiency has to do with this probability. What is the probability of acceptance of a new value if $g(x) \propto f(x)$?

Example in R: multimodal density on p. 50.

Method of composition

For joint $(X, Y) \sim f(x, y) = f_{X|Y}(x|y) f_Y(y)$ you can sample

(a) $Y \sim f_Y(\cdot)$, then

(b) $X|Y = y \sim f_{X|Y}(\cdot|y)$.

The pair $(X, Y) \sim f(x, y)$ as required. This works when it is easy to sample Y marginally, and easy to sample $X|Y$. Use this to get $(X_1, Y_1), \dots, (X_n, Y_n)$.

This is useful in many situations, but here's a common one. We are interested in $X_1, \dots, X_n \overset{iid}{\sim} f_X(\cdot)$ where

$$f_X(x) = \int_{-\infty}^{\infty} \underbrace{f_{X|Y}(x|y) f_Y(y)}_{f(x,y)} \, dy.$$

The method of composition will allow us to get an *iid* sample $X_1, \dots, X_n$; we just throw away $Y_1, \dots, Y_n$.

Two examples...

Works the same for discrete mixtures

$$f_X(x) = \sum_{j=1}^{\infty} f_{X|Y}(x|y_j) \underbrace{P(Y = y_j)}_{\pi_j}.$$

A finite mixture of univariate normals has density

$$f(x) = \sum_{j=1}^{J} \pi_j \phi(x|\mu_j, \sigma_j^2).$$

Sampling $X \sim f(\cdot)$ is easily carried out via the method of composition: first sample Y where $P(Y = j) = \pi_j$, then sample $X|Y \sim N(\mu_Y, \sigma_Y^2)$.

Another example: t distribution: $Y \sim \chi^2_\nu$, $X|Y \sim N(0, \nu/Y)$
$\Rightarrow X \sim t_\nu$.

Note that this is just the definition. Let $Y \sim \chi^2_\nu$ indep. of $Z \sim N(0,1)$ and let

$$X = \frac{Z}{\sqrt{Y/\nu}} = Z\sqrt{\nu/Y}.$$

Then $X|Y \sim N(0, \nu/Y)$.

Frequentist properties of estimators

Most papers in JASA, *Biometrics*, *Statistics in Medicine*, JRSSB, etc. have a simulations section.

Data are generated under known conditions and then an estimator/inferential procedure is applied. That is, $x_1, \ldots, x_n \sim f(x_1, \ldots, x_n|\boldsymbol{\theta})$ is generated M times with known $\boldsymbol{\theta}_0 = (\boldsymbol{\theta}_{01}, \ldots, \boldsymbol{\theta}_{0k})'$ producing M estimates $\hat{\boldsymbol{\theta}}^1, \ldots, \hat{\boldsymbol{\theta}}^M$, M sets of k SEs or posterior SDs, and M sets of k CIs.

Typically $M = 500$ or $M = 1000$ and n reflects common sample sizes found in clinical setting or in the data analysis section, e.g. $n = 100$, $n = 500$, $n = 1000$, $n = 5000$, etc. Often two or three sample sizes are chosen.

Frequentist properties of estimators

Common things to look at are:

- k Biases $\frac{1}{M}\sum_{m=1}^{M}\hat{\theta}_j^m - \theta_{0j}$.
- k average SE or SD $\frac{1}{M}\sum_{m=1}^{M} se(\hat{\theta}_j^m)$. Sometimes instead average lengths of k CIs are reported; gets at the same thing.
- k MSEs $\frac{1}{M}\sum_{m=1}^{M}(\hat{\theta}_j^m - \theta_{0j})^2$.
- k SD of point estimates $\frac{1}{M}\sum_{m=1}^{M}(\hat{\theta}_j^m - \frac{1}{M}\sum_{s=1}^{M}\hat{\theta}_j^s)^2$.
- k Coverage probability of CIs $\frac{1}{M}\sum_{m=1}^{M} I\{L_j^m < \theta_{0j} < U_j^m\}$.