Unit 1: Generating Random Variables

Chapter 3 in "Statistical Computing with R"

Anna Ly

Department of Mathematical and Computational Sciences
University of Toronto Mississauga

January 5, 2026

Overview

1. The Inverse Transform Method
2. The Acceptance-Rejection Method
3. Transformation and Convolution Methods
4. Mixtures Methods

The Inverse Transform Method

The Inverse Transform Method

- Strictly speaking, it is impossible to get random numbers from a computer; but programs can produce pseudo-random numbers.
- In this course we assume that a suitable uniform pseudo-random number generator is available. (The methods for creating pseudo-randomness is not the focus of this course, and shall be omitted.)
- Refer to ‘help(.Random.seed)‘ for details about the default random number generator in R.

The Inverse Transform Method

Theorem 3.1 (Probability Integral Transform)

If X is a continuous random variable with cdf $F_X(x)$ then $U = F_X(x) \sim \mathit{Uniform}(0, 1)$.

Proof.

The Inverse Transform Method

Corollary

Let $U \sim Uniform(0,1)$. Define $X = F^{-1}(U)$, where F is a cdf. Then, F is the cdf of X.

Proof.

The Inverse Transform Method

The inverse transform method (continuous case) can be summarized as follows:

1. Find the cdf, $F_X(x)$.
2. Define the inverse function $F_X^{-1}(u)$.
3. For each random variate required:
   a. Generate a random u from $Uniform(0, 1)$.
   b. Deliver $x = F_X^{-1}(u)$.

Warning: we should only use this method if a closed form of $F_X^{-1}(x)$ exists!

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the distribution with density:

$$f_X(x) = 3x^2, \quad 0 < x < 1.$$

Additionally, write the R code.

Solution.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the exponential distribution, which has the density:

$$f_X(x) = \frac{1}{\theta} e^{-\frac{1}{\theta}x}, \quad x > 0, \theta > 0.$$

Additionally, write the R code.

Solution.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the Weibull distribution, which has the density:

$$f_X(x) = \begin{cases} \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & x \geq 0, \alpha > 0, \beta > 0, \\ 0 & \text{otherwise.} \end{cases}$$

To make life easier, we'll consider the case where $\beta = 1$.

1. Additionally, write the R code.
2. Find $\mathbb{E}[X]$, the theoretical mean, and check that the simulated mean is close to the theoretical mean.

The Inverse Transform Method

If we just want to generate a random variable X with pmf:

$$\mathbb{P}(X = x_i) = p_i, \;\; i \in \mathbb{N}, \;\; \sum_i p_i = 1,$$

then inverse transform method (discrete case) can be summarized as follows:

1. Generate a random u from $Uniform(0, 1)$.
2. Transform u into X as follows:

$$X = x_j \;\text{ if }\; F_X(x_{j-1}) < u \leq F_X(x_j)$$

3. It follows that,

$$X = \begin{cases} x_1 & u \leq F_X(x_1), \\ x_2 & F_X(x_1) < u \leq F_X(x_2), \\ \dots & \\ x_j & F_X(x_{j-1}) < u \leq F_X(x_j), \\ \dots & \end{cases}$$

The Inverse Transform Method

In other words, discrete random variables can be generated by slicing up the interval $(0, 1)$ into subintervals which define a partition on $(0, 1)$:

$$(0, F_X(x_1)], (F_X(x_1), F_X(x_2)], (F_X(x_2), F_X(x_3)], \ldots, (F_X(x_{k-1}), 1].$$

We can also define:

$$p_1 = F_X(x_1), \quad p_2 = F_X(x_2) - p_1, \quad \ldots, \quad p_j = F_X(x_j) - \sum_{i=1}^{j-1} p_i$$

Proof of Previous Algorithm

Given X that was defined in the algorithm of the previous slide, Prove that $\mathbb{P}(X = x_j) = p_j$.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the Bernoulli distribution with $p = 0.4$. Additionally, write the R code.

Solution.

The Inverse Transform Method

Example

Let X be a discrete random variable with the following pmf:

x	1	2	3	4
$p_X(x)$	0.2	0.5	0.2	0.1

Find $F_X(x)$. Additionally, write the R code.

Solution.

The Inverse Transform Method

Generalized Inverse Function

For a discrete random variable X with cdf $F_X(x)$, the inverse CDF $F_X^{-1}(p)$, also called the quantile function, is defined as:

$$F_X^{-1}(x) := \inf\{x : F_X(x) \geq p\}, \quad 0 < p < 1.$$

Note that there isn't a unique x where $F(x) = p$ for every $p \in (0, 1)$.

Example

From the previous example,

- Find $F_X^{-1}(0.1)$.
- Find $F_X^{-1}(0.5)$.
- Find $F_X^{-1}(0.85)$.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the Geometric distribution with pmf:

$$\mathbb{P}(X = i) = pq^i, \quad i \in \mathbb{N}, \quad q = 1 - p.$$

Additionally, write the R code.

Solution.

The Inverse Transform Method

Remember that now we are slicing up the interval $(0, 1)$ into subintervals which define a partition on $(0, 1)$:

$$(0, F_X(x_1)], (F_X(x_1), F_X(x_2)], (F_X(x_2), F_X(x_3)], \ldots, (F_X(x_{k-1}), 1].$$

Which basically means we are trying to evaluate $F_X(x_i)$ for $i \in \mathbb{N}$.

- However, sometimes it's hard to obtain a closed form for $F_X(x_i)$.
- Thus, it is more useful to find a recursive formula.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the Binomial distribution with pmf:

$$p_i := \mathbb{P}(X = i) = \binom{n}{i} p^i (1-p)^{n-i}, \quad i = 0, 1, \ldots, n.$$

1. First, derive the following recursive formula:

$$p_{i+1} = \left(\frac{n-i}{i+1}\right)\left(\frac{p}{1-p}\right) p_i, \quad i = 0, 1, \ldots, n.$$

2. Write the R code to simulate the algorithm.

The Inverse Transform Method

Example

Use the inverse transform method to simulate a random sample from the logarithmic distribution with pmf:

$$p_i := \mathbb{P}(X = i) = \frac{a\theta^i}{i}, \quad i \in \mathbb{N}.$$

Where $0 < \theta < 1$ and $a = (-\ln(1-\theta))^{-1}$.

1. First, derive the following recursive formula:

$$p_{i+1} = \left(\frac{\theta i}{i+1}\right) p_i, \quad i \in \mathbb{N}.$$

2. Write the *R* code to simulate the algorithm.

The Inverse Transform Method

In general:

$$X \quad \Rightarrow \quad F_X^{-1}(x) \quad \Rightarrow \quad F_X^{-1}(u)$$

However, this only works if we can define the inverse. We can think of many different functions where the inverse would be hard to find: the Gaussian distribution, beta, etc...

The Acceptance-Rejection Method

Acceptance-Rejection

Suppose we want to generate the random variable X with target density f using the acceptance-rejection method.

1. Find another random variable, Y with trial/candidate/envelope density g where there exists $c \in \mathbb{R}$ such that:
$$\frac{f(t)}{g(t)} < c.$$
2. For each random variate required:
   2.1 Generate y from the distribution with density g.
   2.2 Generate u from the $Uniform(0, 1)$ distribution.
   2.3 If
$$u < \frac{f(y)}{cg(y)}$$
   accept y and deliver $x = y$. Otherwise, reject y and generate a random variate again.

Review

Consider the probability triple $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\{B_1, B_2, \dots\}$ be a partition of Ω. Then, for any $A \in \mathcal{F}$, we have that:

Total Law of Probability

$$\mathbb{P}(A) = \sum_i \mathbb{P}(A|B_i)\mathbb{P}(B_i), \quad i \in \mathbb{N}$$

Bayes Rule

$$\mathbb{P}(B_j|A) = \frac{\mathbb{P}(B_j \cap A)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A|B_j)\mathbb{P}(B_j)}{\sum_{i=1}^{n} \mathbb{P}(B_i)\mathbb{P}(A|B_i)}, \quad i, j \in \mathbb{N}$$

Acceptance-Rejection

Probability of Accepting

Given the acceptance-rejection algorithm, evaluate the probability of acceptance for any iteration (see part 2.3 in the previous slide).

Proof.

Acceptance-Rejection

Probability of Accepting

Given the acceptance-rejection algorithm, prove that the accepted sample has the same distribution as X.

Proof.

The Choice of c

Given the acceptance-rejection algorithm, show that that $c \geq 1$.

Solution.

Acceptance-Rejection

- Ideally, you want g to be easy to sample from, consistent with the support of f.
- Theoretically, as long as you know that f indeed has longer tails than g, you can choose c to be ridiculously large and this will still yield a valid algorithm.
- Since the acceptance rate is equal to $1/c$, you will have to, on average, generate $c \times n$ draws from the trial distribution and from the uniform distribution just to get n draws from the target distribution.
- So choosing c to be too large will yield an inefficient algorithm.
- Ideally: choose c close to 1; so f and g are similar.

Acceptance-Rejection

The Distribution of N

Let N represent the number of iterations that the acceptance-rejection algorithm needs to successfully generate one value of X. What is the distribution of N?

Solution.

Acceptance-Rejection

Example

Use the acceptance-rejection method to simulate a random sample from the distribution with density:

$$f_X(x) = 3x^2, \quad 0 < x < 1.$$

Let the trial density be $Uniform(0, 1)$. Write the R code to simulate the algorithm. Compare the number of iterations to the value of $\frac{1}{c}$.

Solution.

Acceptance-Rejection

Example

Use the acceptance-rejection method to simulate a random sample from the beta distribution with density:

$$f_X(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1},, \quad 0 < x < 1.$$

Assume $\alpha = 2, \beta = 4$. Let the trial density be *Uniform*$(0, 1)$. Write the *R* code to simulate the algorithm. Compare the number of iterations to the value of $\frac{1}{c}$.

Solution.

Acceptance-Rejection

Example

Use the acceptance-rejection method to simulate a random sample from the distribution with density:

$$f_X(x) = \begin{cases} \frac{2}{\pi}\sqrt{1-x^2} & |x| < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Let the trial density be $Uniform(-1, 1)$. Write the R code to simulate the algorithm. Compare the number of iterations to the value of $\frac{1}{c}$.

Solution.

Transformation and Convolution Methods

Transformation and Convolution Methods

You've done transformations or convolution methods before (or at least I hope!)

- If $X_i \sim Exponential(a)$ with $i \in \{1, 2, ..., n\}$ then $\sum_i X_i \sim Gamma(n, a)$.
- If $U \sim Gamma(r, \lambda)$ and $V \sim Gamma(s, \lambda)$ and $U \perp\!\!\!\perp V$ then

$$X := \frac{U}{U+V} \sim Beta(r, s).$$

- If $Z \sim N(0, 1)$ then $V := Z^2 \sim \chi^2(1)$.
- If $U \sim \chi^2(m)$ and $V \sim \chi^2(n)$ and $U \perp\!\!\!\perp V$ then

$$\tilde{F} := \frac{U/m}{V/n} \sim F(m, n).$$

- If $Z \sim N(0, 1)$ and $V \sim \chi^2(n)$ and $Z \perp\!\!\!\perp V$ then:

$$\tilde{T} := \frac{Z}{\sqrt{V/n}} \sim t(n).$$

Example

Use rexp() to generate the *Gamma*$(\alpha = 10, \beta = 1/2)$ distribution.

Solution.

Example

Use rexp() to generate the *Beta*($\alpha = 2, \beta = 3$) distribution.

Solution.

Mixture Methods

Finite Mixture Model

A finite mixture model is a statistical model that represents a probability distribution as a mixture of several component distributions. Mathematically, given k component distributions $f_1(x), ..., f_k(x)$, each with associated mixing probabilities (also known mixing weights) $\pi_1, \ldots, \pi_k$, a finite mixture model $f(x)$ is defined as:

$$f(x) := \sum_{i=1}^{k} \pi_i f_i(x),$$

where $0 \leq \pi_i \leq 1$ and $\sum_{i=1}^{k} \pi_i = 1$.

Mixture Model Properties

We want to show that $f(x)$ satisfies the properties of a density function. That is, show that $f(x)$ is non-negative and $\int_{-\infty}^{\infty} f(x)dx = 1$.

Solution.

Mixture Model CDF

Compute $F(x)$.

Solution.

Mixture Model Expected Value

Let X_i have density f_i. Denote $\mu_i := \mathbb{E}[X_i] = \int_{-\infty}^{\infty} x f_i(x) dx$; this represents the mean for the i-th component distribution. Compute $\mu := \mathbb{E}[X]$. What does it represent?

Solution.

Mixture Model Variance

Let $\sigma_i^2 := \mathbb{V}[X_i]$ represent the standard deviation for the i-th component distribution. Show that:

$$\sigma^2 := \mathbb{V}[X] = \sum_{i=1}^{k} \pi_i \sigma_i^2 + \sum_{i=1}^{k} \pi_i (\mu_i - \mu)^2$$

What does it represent?

Solution.

Mixture Models

Convolutions and mixtures look similar but the represented distributions differ!

Example

Suppose $X_1 \sim N(0,1)$, $X_2 \sim N(3,1)$, and $X_1 \perp\!\!\!\perp X_2$.

- Convolution representation:

$$S := 0.4X_1 + 0.6X_2$$

- Mixture representation:

$$F_X(x) := 0.4F_{X_1}(x_1) + 0.6F_{X_2}(x_2)$$

Note: for the convolution, the coefficients in front of the random variables do not necessarily have to add to one; we could have done $S := aX_1 + bX_2$ for any $a, b \in \mathbb{R}$. However, for finite mixture models, these coefficients must add to 1.

Mixture Models

Simulating a variable from a finite k-mixture distribution is typically carried out by the composition method: Consider $F(x) = \sum_{i=1}^{k} \pi_i F_{X_i}(x)$. Then,

Composition Method

- Generate an integer $I \in \{1, \ldots, k\}$ such that:

$$\mathbb{P}(I = i) = \pi_i, \quad i \in \{1, 2, \ldots, k\}.$$

- Deliver X with cdf $F_{X_I}(x)$.

Example

Suppose $X_1 \sim N(0,1)$, $X_2 \sim N(3,1)$, and $X_1 \perp\!\!\!\perp X_2$. Simulating the following using R:

$$F_X(x) := 0.4F_{X_1}(x_1) + 0.6F_{X_2}(x_2)$$

Then, compare the above to the following convolution counterpart:

$$S := 0.4X_1 + 0.6X_2$$

Solution.

Mixture Models

The first two methods we discussed, inverse-transform and acceptance-rejection method, depend on the uniform distribution. Similarly, we can do the same for generating finite mixture models. Consider $F(x) = \sum_{i=1}^{k} \pi_i F_{X_i}(x)$. Then:

Modified Composition Method

- Generate u from the $Uniform(0, 1)$ distribution.
- If:

$$\sum_{i=1}^{l-1} \pi_i \leq u < \sum_{i=1}^{l} \pi_i$$

then generate a random x from $F_{X_l}(x)$ where $l = 1, \ldots, k$ with the convention $\sum_{i=1}^{0} \pi_i = 0$.

Example

Using the alternative method... Suppose $X_1 \sim N(0,1)$, $X_2 \sim N(3,1), X_3 \sim N(5,1)$, and assume they're all independent of each other. Code the following algorithm using R:

$$F_X(x) := 0.4F_{X_1}(x_1) + 0.3F_{X_2}(x_2) + 0.3F_{X_3}(x_3)$$

Solution.