Box-Muller simulation of standard normal

1. Generate $U_1, U_2 \sim U(0,1)$.
2. Set $\Theta = 2\pi U_1$ and $R = \sqrt{-2\log(U_2)}$.
3. Return $X = R\cos(\Theta)$ and $Y = R\sin(\Theta)$.

Calculating sines and cosines can be expensive (in terms of the time required), but there is a version of the Box-Muller algorithm that avoids this. Suppose that the point $Q = (A, B)$ is uniformly distributed over the unit circle. Let (S, Ψ) be the polar coordinates of Q, then one can show that $S^2 \sim U(0,1)$ and $\Psi \sim U(0, 2\pi)$, independently of S. Thus, $(\sqrt{-2\log(S^2)}, \Psi)$ has the same distribution as the polar coordinates of P, namely bivariate standard normal. The advantage of this representation is that it we can easily calculate X and Y from A and B. A little trigonometry gives us that

$$\begin{aligned} S^2 &= A^2 + B^2 \\ \cos(\Psi) &= A/S \\ \sin(\Psi) &= B/S \end{aligned}$$

so, for $R = \sqrt{-2\log(S^2)}$,

$$\begin{aligned} X &= R\cos(\Psi) = A\sqrt{\frac{-2\log(S^2)}{S^2}} \\ Y &= R\sin(\Psi) = B\sqrt{\frac{-2\log(S^2)}{S^2}}. \end{aligned}$$

We still need to generate Q, but this can be easily achieved using a rejection algorithm. Generate $U, V \sim U(-1,1)$ independently, then accept the point (U, V) if it is inside the unit circle, that is, if $U^2 + V^2 < 1$.

Putting these together we get the following:

Improved Box-Muller simulation of standard normal, with rejection step

1. Generate $U, V \sim U(-1,1)$.
2. Accept $S^2 = U^2 + V^2$ provided $S^2 < 1$ else return to step 1.
3. Set $W = \sqrt{-2\log(S^2)/S^2}$.
4. Return $X = UW$ and $Y = VW$.

18.6 Exercises

1. Express 45 in binary.
   Now express 45 mod 16 and 45 mod 17 in binary.
   What can you say about these three binary representations?

2. Find all of the cycles of the following congruential generators. For each cycle identify which seeds X_0 lead to that cycle.

   (a). $X_{n+1} = 9X_n + 3 \bmod 11$.
   (b). $X_{n+1} = 8X_n + 3 \bmod 11$.
   (c). $X_{n+1} = 8X_n + 2 \bmod 12$.

3. Here is some pseudo-code of an algorithm for generating a sample $y_1, \ldots, y_k$ from the population $x_1, \ldots, x_n$ *without* replacement ($k \leq n$):

```
for (i in 1:k) {
  { Select j at random from 1:(n+1-i) }
  y[i] <- x[j]
  { Swap x[j] and x[n+1-i] }
}
```

   Implement this algorithm in R. (The built-in implementation is `sample`.)

4. Consider the discrete random variable with pmf given by:

$$\mathbb{P}(X = 1) = 0.1, \quad \mathbb{P}(X = 2) = 0.3, \quad \mathbb{P}(X = 5) = 0.6.$$

   Plot the cdf for this random variable.

   Write a program to simulate a random variable with this distribution, using the built-in function `runif(1)`.

5. How would you simulate a negative binomial random variable from a sequence of Bernoulli trials? Write a function to do this in R. (The built-in implementation is `rnbinom(n, size, prob)`.)

6. For $X \sim \text{Poisson}(\lambda)$ let $F(x) = \mathbb{P}(X \leq x)$ and $p(x) = \mathbb{P}(X = x)$. Show that the probability function satisfies

$$p(x+1) = \frac{\lambda}{x+1} p(x).$$

   Using this write a function to calculate $p(0), p(1), \ldots, p(x)$ and $F(x) = p(0) + p(1) + \cdots + p(x)$.

   If $X \in \mathbb{Z}_+$ is a random variable and `F(x)` is a function that returns the cdf F of X, then you can simulate X using the following program:

```
F.rand <- function () {
  u <- runif(1)
  x <- 0
  while (F(x) < u) {
    x <- x + 1
  }
  return(x)
}
```

   In the case of the Poisson distribution, this program can be made more efficient by calculating F just once, instead of recalculating it every time you call the function `F(x)`. By using two new variables, `p.x` and `F.x` for $p(x)$ and $F(x)$ respectively, modify this program so that instead of using

the function `F(x)` it updates `p.x` and `F.x` within the `while` loop. Your program should have the form

```
F.rand <- function(lambda) {
  u <- runif(1)
  x <- 0
  p.x <- ?
  F.x <- ?
  while (F.x < u) {
    x <- x + 1
    p.x <- ?
    F.x <- ?
  }
  return(x)
}
```

You should ensure that at the start of the `while` loop you always have `p.x` equal to $p(x)$ and `F.x` equal to $F(x)$.

7. This exercise asks you to verify the function `F.rand` from Exercise 6. The idea is to use `F.rand` to estimate the Poisson probability mass function, and compare the estimates with known values. Let $X_1, \ldots, X_n$ be independent and identically distributed (iid) pois(λ) random variables, then we estimate $p_\lambda(x) = \mathbb{P}(X_1 = x)$ using

$$\hat{p}_\lambda(x) = \frac{|\{X_i = x\}|}{n}.$$

Write a program `F.rand.test(n, lambda)` that simulates n pois(λ) random variables and then calculates $\hat{p}_\lambda(x)$ for $x = 0, 1, \ldots, k$, for some chosen k. Have your program print a table giving $p_\lambda(x)$, $\hat{p}_\lambda(x)$ and a 95% confidence interval for $p_\lambda(x)$, for $x = 0, 1, \ldots, k$.

Finally, modify your program `F.rand.test` so that it also draws a graph of $\hat{p}$ and p, with confidence intervals, similar to Figure 18.2.

8. Suppose that X takes on values in the countable set $\{\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots\}$, with probabilities $\{\ldots, p_{-2}, p_{-1}, p_0, p_1, p_2, \ldots\}$. Suppose also that you are given that $\sum_{i=0}^{\infty} p_i = p$, then write an algorithm for simulating X.

Hint: first decide whether or not $X \in \{a_0, a_1, \ldots\}$, which occurs with probability p.

9. Suppose that X and Y are independent rv's taking values in $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ and let $Z = X + Y$.

   (a). Suppose that you are given functions `X.sim()` and `Y.sim()`, which simulate X and Y. Using these, write a function in R to estimate $\mathbb{P}(Z = z)$ for a given z.

   (b). Suppose that instead of `X.sim()` and `Y.sim()` you are given `X.pmf(x)` and `Y.pmf(y)`, which calculate $\mathbb{P}(X = x)$ and $\mathbb{P}(Y = y)$ respectively.

Using these, write a function `Z.pmf(z)` to calculate $\mathbb{P}(Z = z)$ for a given z.

(c). Given `Z.pmf(z)` write a function in R to calculate $\mathbb{E}Z$.

Note that we may have $\mathbb{P}(Z = z) > 0$ for all $z \geq 0$. To approximate $\mu = \mathbb{E}Z$ numerically we use $\mu_n^{trunc} = \sum_{z=0}^{n-1} z\mathbb{P}(Z = z) + n\mathbb{P}(Z \geq n) = n - \sum_{z=0}^{n-1}(n - z)\mathbb{P}(Z = z)$. How can we decide how large n needs to be to get a good approximation?

Do you think this method of approximating $\mathbb{E}Z$ is better or worse than simulation?

10. Consider the following program, which performs a simulation experiment. The function `X.sim()` simulates some random variable X, and we wish to estimate $\mathbb{E}X$.

```
# set.seed(7)
# seed position 1
mu <- rep(0, 6)
for (i in 1:6) {
  # set.seed(7)
  # seed position 2
  X <- rep(0, 1000)
  for (j in 1:1000) {
    # set.seed(7)
    # seed position 3
    X[j] <- X.sim()
  }
  mu[i] <- mean(X)
}
spread <- max(mu) - min(mu)
mu.estimate <- mean(mu)
```

(a). What is the value of `spread` used for?

(b). If we uncomment the command `set.seed(7)` at seed position 3, then what is `spread`?

(c). If we uncomment the command `set.seed(7)` at seed position 2 (only), then what is `spread`?

(d). If we uncomment the command `set.seed(7)` at seed position 1 (only), then what is `spread`?

(e). At which position should we set the seed?

11. (a). Here is some code for simulating a discrete random variable Y. What is the probability mass function (pmf) of Y?

```
Y.sim <- function() {
  U <- runif(1)
  Y <- 1
  while (U > 1 - 1/(1+Y)) {
    Y <- Y + 1
```

```
  }
  return(Y)
}
```

Let N be the number of times you go around the while loop when `Y.sim()` is called. What is $\mathbb{E}N$ and thus what is the expected time taken for this function to run?

(b). Here is some code for simulating a discrete random variable Z. Show that Z has the same pmf as Y

```
Z.sim <- function() {
  Z <- ceiling(1/runif(1)) - 1
  return(Z)
}
```

Will this function be faster or slower that `Y.sim()`?

12. People arrive at a shoe store at random. Each person then looks at a random number of shoes before deciding which to buy.

(a). Let N be the number of people that arrive in an hour. Given that $\mathbb{E}N = 10$, what would be a good distribution for N?

(b). Customer i tries on X_i pairs of shoes they do not like before finding a pair they like and then purchase ($X_i \in \{0, 1, \ldots\}$). Suppose that the chance they like a given pair of shoes in 0.8, independently of the other shoes they have looked at. What is the distribution of X_i?

(c). Let Y be the total number of shoes that have been tried on, excluding those purchased. Supposing that each customer acts independently of other customers, give an expression for Y in terms of N and the X_i, then write functions for simulating N, X_i, and Y.

(d). What is $\mathbb{P}(Y = 0)$?

Use your simulation of Y to estimate $\mathbb{P}(Y = 0)$. If your confidence interval includes the true value, then you have some circumstantial evidence that your simulation is correct.

13. Consider the continuous random variable with pdf given by:

$$f(x) = \begin{cases} 2(x-1)^2 & \text{for } 1 < x \leq 2, \\ 0 & \text{otherwise.} \end{cases}$$

Plot the cdf for this random variable.

Show how to simulate a rv with this cdf using the inversion method.

14. Consider the continuous random variable X with pdf given by:

$$f_X(x) = \frac{\exp(-x)}{(1+\exp(-x))^2} \quad -\infty < x < \infty.$$

X is said to have a standard logistic distribution. Find the cdf for this random variable. Show how to simulate a rv with this cdf using the inversion method.

15. Let $U \sim U(0,1)$ and let $Y = 1 - U$. Derive an expression for the cdf $F_Y(y)$ of Y in terms of the cdf of U and hence show that $Y \sim U(0,1)$.

16. For a given u, adapt the bisection method from Chapter 10 to write a program to find the root of the function $\Phi(x) - u$ where $\Phi(x)$ is the cdf of the standard normal distribution. (You can evaluate Φ using numerical integration or by using the built-in R function.) Notice that the root satisfies $x = \Phi^{-1}(u)$.

    Using the inversion method, write a program to generate observations on a standard normal distribution. Compare the proportion of your observations that fall within the interval $(-1, 1)$ with the theoretical value of 68.3%.

17. The continuous random variable X has the following probability density function (pdf), for some positive constant c,

$$f(x) = \frac{3}{(1+x)^3} \text{ for } 0 \leq x \leq c.$$

    (a). Prove that $c = \sqrt{3} - 1$.
    (b). What is $\mathbb{E}X$? (Hint: $\mathbb{E}X = \mathbb{E}(X+1) - 1$.)
    (c). What is $\operatorname{Var} X$? (Hint: start with $\mathbb{E}(X+1)^2$.)
    (d). Using the inversion method, write a function that simulates X.

18. The Cauchy distribution with parameter α has pdf

$$f_X(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)} \quad -\infty < x < \infty.$$

    Write a program to simulate from the Cauchy distribution using the inversion method.

    Now consider using a Cauchy envelope to generate a standard normal random variable using the rejection method. Find the values for α and the scaling constant k that minimise the probability of rejection. Write an R program to implement the algorithm.