Lecture 9: Generating random variables – transformations and wrap-up

Ciaran Evans

Recap: acceptance-rejection sampling

- Want to sample continuous r.v. $X \sim f$
- Can easily sample from a different density: $Y \sim g$, such that $\frac{f(t)}{g(t)} \leq c$

1. Sample $Y \sim g$
2. Sample $U \sim Uniform(0, 1)$
3. If $U \leq \dfrac{f(Y)}{cg(Y)}$, set $X = Y$. Otherwise, return to step 1.

Question: What are some potential downsides to the acceptance-rejection sampling method?

Inefficiency in acceptance-rejection sampling

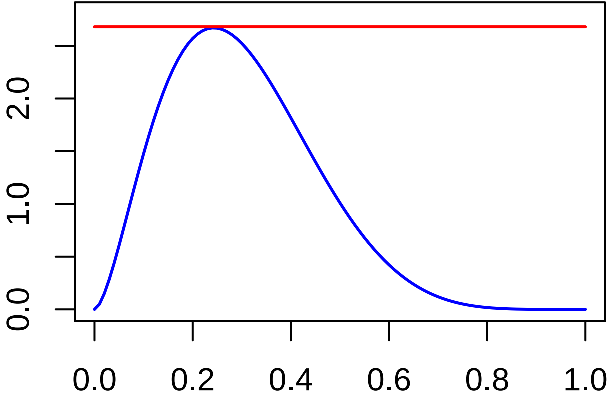
1. Sample $Y \sim g$
2. Sample $U \sim Uniform(0, 1)$
3. If $U \leq \frac{f(Y)}{cg(Y)}$, set $X = Y$. Otherwise, return to step 1.

$$P(\text{accept } Y|Y = y) = P\left(U \leq \frac{f(y)}{cg(y)}\right) = ??$$

$$P(\text{accept } Y) = ??$$

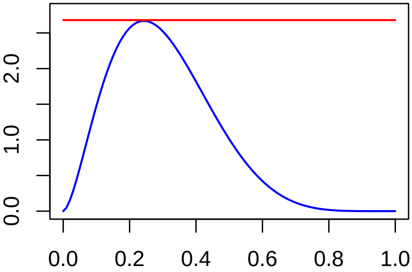
Inefficiency in acceptance-rejection sampling

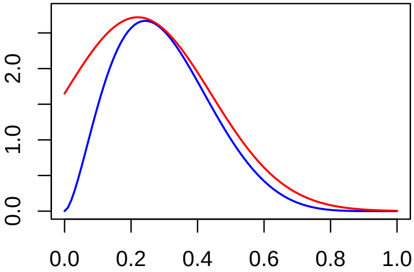
$Beta(2.7, 6.3)$ example from class activity:



Here $c = 2.7$. About many samples from g would I need to get 1000 samples from f?

Inefficiency in acceptance-rejection sampling





Question: Which of these two candidate densities g would you prefer?

Drawbacks of acceptance-rejection sampling

1. Need to find a suitable candidate g
2. Requires more samples $Y \sim g$ than we get from target f (because we reject some samples)
   - Want g to be as close as possible to f, to accept as many samples as possible
3. Calculating $f(Y)$ for candidate draws $Y \sim g$ may be expensive for some distributions

Project 1: Modifying the acceptance-rejection method to address these drawbacks

Today: Another approach to generating random variables

Generating a Normal random variable

Suppose we want to simulate $X \sim N(0, 1)$

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}$$

$$F_X(t) = \ ?$$

Box-Muller Transformation

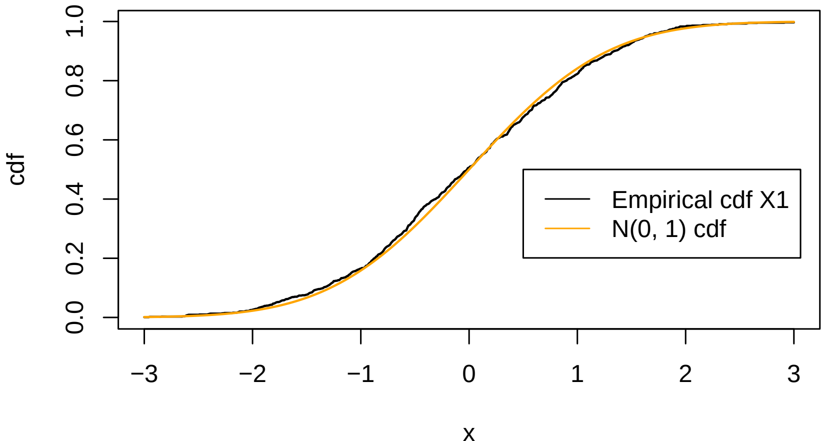
Box-Muller in practice

```r
# generate 1000 samples
u1 <- runif(1000)
u2 <- runif(1000)
x1 <- sqrt(-2*log(u1)) * cos(2*pi*u2)
x2 <- sqrt(-2*log(u1)) * sin(2*pi*u2)
```

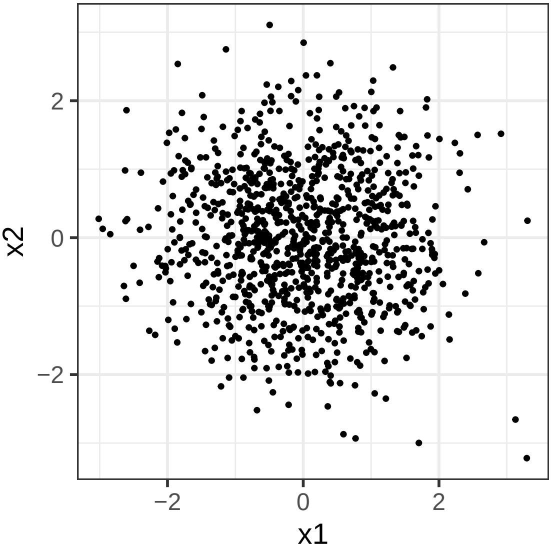
Question: How can I check that the samples match the desired $N(0, 1)$ distribution?

Box-Muller in practice

```
# generate 1000 samples
u1 <- runif(1000)
u2 <- runif(1000)
x1 <- sqrt(-2*log(u1)) * cos(2*pi*u2)
x2 <- sqrt(-2*log(u1)) * sin(2*pi*u2)
```



Box-Muller in practice



Other Normals

Suppose that $Z \sim N(0, 1)$. How do I get $X \sim N(\mu, \sigma^2)$?

A few other transformations

- If $X \sim N(\mu, \sigma^2)$, then $e^X \sim Lognormal(\mu, \sigma^2)$
- If $Z_1, ..., Z_k \overset{iid}{\sim} N(0, 1)$, then

$$\sum_{i=1}^{k} Z_i^2 \sim ?$$

- If $V_1 \sim \chi^2_{d_1}$ and $V_2 \sim \chi^2_{d_2}$ are independent, then

$$\frac{V_1/d_1}{V_2/d_2} \sim ?$$

- If $Y_1 \sim Gamma(\alpha, \theta)$ and $Y_2 \sim Gamma(\beta, \theta)$ are independent, then

$$\frac{Y_1}{Y_1 + Y_2} \sim ?$$

Summary (so far)

Methods to generate random variables, in rough order of preference:

1. Use inverse transform method (if inverse cdf is tractable)
2. Find a different transformation (if possible)
3. Acceptance-rejection sampling (perhaps with modifications)

Homework 3

https://sta379-s25.github.io/homework/hw3.html

- Practice generating random variables
- Accept and submit coding portion of assignment on GitHub Classroom
- Collaboration encouraged on homework, but everyone must submit their own work and acknowledge collaborators