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

Ciaran Evans

Previously

- Methods to generate $U \sim Uniform(0, 1)$:
  - Linear congruential generator
  - Mersenne twister
  - lots of other variants and alternatives
- Methods to generate other random variables:
  - inverse transform method
    - Pros: easy and efficient if the inverse cdf is easy to find
    - Cons: requires the inverse cdf to be tractable
  - Acceptance-rejection sampling:
    - Pros: works for any continuous distribution if you can find a good candidate g
    - Cons: can be slow/inefficient if we don't choose a good candidate density
    - Project 1: Making acceptance-rejection sampling more efficient with an *adaptive* candidate density

Today: How else can we generate random variables?

Generating a Normal random variable

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

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

$$F_X(t) = \text{?}$$

Box-Muller Transformation

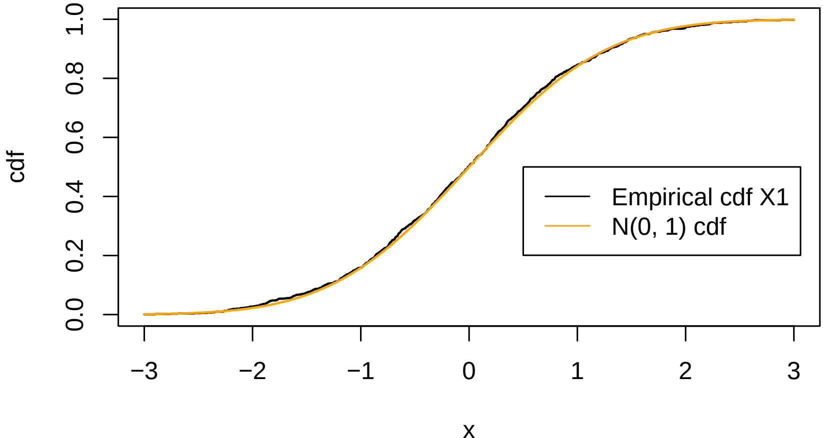
Box-Muller in practice

```
# 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)
```

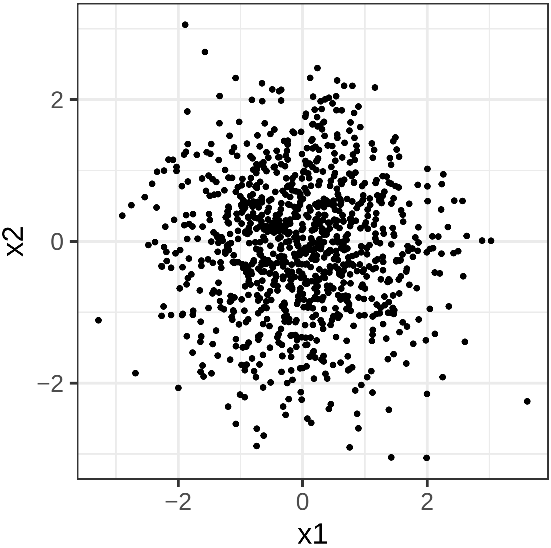
Question: How can I check that the samples match the desired $N(0,1)$ distribution?

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)
```



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

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