Chapter 4: Random Number Generation
Inverse Transform & Rejection Sampling

Roadmap for today

1. Warm-up: truly random vs pseudo-random number generators
2. Inverse transform sampling
   - intuition via CDF
   - inverse distribution function (quantile function)
   - theorem and proof idea
   - worked example: Exponential(λ)
   - how this appears in R
3. Rejection (envelope) sampling
   - intuition (accept/reject)
   - algorithm and condition
   - efficiency discussion
   - proof sketch

Warm-up: True vs pseudo-random

Question (from last session): Which of these are *truly* random number generators, and which are *pseudo*-random?

- Physical process / hardware entropy (e.g. thermal noise, radioactive decay)
- Coin flips / dice rolls (physical randomness)
- Linear congruential generator (LCG)
- `runif()` and friends in R

Key idea:

- True RNG uses a *physical* source of entropy.
- Pseudo RNG is an *algorithm* (deterministic given its seed), designed to *look* random.

Answer & takeaway

Generator	True / Pseudo?	Why
Physical process	**True**	randomness from physics
Coin flips/dice (real)	**True**	physical unpredictability
Linear congruential (LCG)	**Pseudo**	deterministic recurrence; periodicity
`runif()` in R	**Pseudo**	algorithmic PRNG under the hood

Practical note: In statistics/computation, pseudo-random numbers are usually what we use—as long as the generator is high quality and we manage seeds carefully.

Chapter 4: Two big tools

We can often generate **Uniform**(0,1) random numbers efficiently.

Goal: Use Uniform$(0, 1)$ samples to generate samples from more "exotic" distributions.

Two methods:

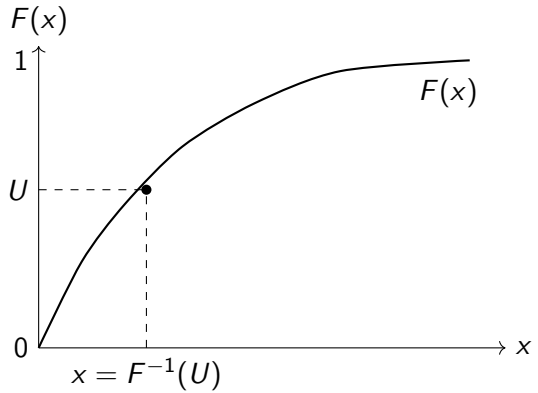
1. Inverse transform sampling (works when we can work with the CDF and its inverse)
2. Rejection (envelope) sampling (works when inverse CDF is unavailable)

Inverse transform sampling: the picture

Suppose we want samples from a continuous distribution with CDF F.

Idea:

- draw $U \sim \mathrm{Unif}(0, 1)$ (a vertical coordinate on the CDF scale)
- map it to the corresponding x via $x = F^{-1}(U)$



Definition: inverse distribution function (quantile function)

Let X be a real-valued random variable with **distribution function** (CDF) F.

The **inverse distribution function** F^{-1} is defined for $u \in (0,1)$ by

$$F^{-1}(u) \;=\; \inf\{x \in \mathbb{R} : F(x) \geq u\}.$$

Why this definition?

- It works cleanly even when F has flat regions or jumps.
- It is the right object for proving inverse transform sampling.

Theorem: inverse transform theorem

Let $F : \mathbb{R} \to [0, 1]$ be a **continuous** distribution function. Let $U \sim \mathrm{Unif}(0, 1)$ and define

$$Y = F^{-1}(U).$$

Then Y has distribution function F (i.e. $Y \sim F$).

Translation:

$$U \sim \mathrm{Unif}(0, 1) \quad \Longrightarrow \quad F^{-1}(U) \text{ is a sample from the target distribution.}$$

Proof idea (step-by-step)

Fix any real number a. We want to show:

$$\mathbb{P}(Y \leq a) = F(a).$$

Start from the definition $Y = F^{-1}(U)$:

$$\mathbb{P}(Y \leq a) = \mathbb{P}(F^{-1}(U) \leq a).$$

Using the inverse definition,

$$F^{-1}(U) \leq a \quad \Longleftrightarrow \quad U \leq F(a) \qquad \text{(when } F \text{ is continuous)}.$$

So

$$\mathbb{P}(Y \leq a) = \mathbb{P}(U \leq F(a)) = F(a),$$

because $U \sim \mathrm{Unif}(0,1)$ implies $\mathbb{P}(U \leq t) = t$ for $t \in [0,1]$.

Inverse transform: the three-step recipe

To sample from a continuous distribution with density $\pi(x)$:

1. Compute the CDF:
$$F(x) = \int_{-\infty}^{x} \pi(t)\, dt.$$
2. Compute the inverse CDF (quantile function) $F^{-1}(u)$.
3. Generate $U \sim \mathrm{Unif}(0, 1)$ and output
$$X = F^{-1}(U).$$

Bottleneck: Many distributions have CDFs/inverses that are not available in closed form.

Worked example: Exponential(λ)

Let $X \sim \mathrm{Exp}(\lambda)$ with density

$$\pi(x \mid \lambda) = \lambda e^{-\lambda x}, \qquad x \geq 0,$$

and $\pi(x) = 0$ otherwise.

We will apply the 3-step recipe to derive an explicit sampling formula.

Step 1: Compute the CDF

For $x \geq 0$,

$$F(x) = \mathbb{P}(X \leq x) = \int_0^x \lambda e^{-\lambda t}\, dt = \Big[-e^{-\lambda t} \Big]_0^x = 1 - e^{-\lambda x}.$$

So

$$F(x) = \begin{cases} 0, & x < 0, \\ 1 - e^{-\lambda x}, & x \geq 0. \end{cases}$$

Step 2: Invert the CDF

Let $u \in (0, 1)$ and solve $u = 1 - e^{-\lambda x}$ for x:

$$u = 1 - e^{-\lambda x} \implies e^{-\lambda x} = 1 - u \implies -\lambda x = \log(1 - u)$$

hence

$$F^{-1}(u) = -\frac{1}{\lambda} \log(1 - u).$$

Note: $1 - U \sim \mathrm{Unif}(0, 1)$ as well, so many implementations use $-\frac{1}{\lambda} \log(U)$.

Step 3: Plug in $U \sim \mathrm{Unif}(0,1)$

If $U \sim \mathrm{Unif}(0,1)$, define

$$X = -\frac{1}{\lambda}\log(1-U).$$

Then

$$X \sim \mathrm{Exp}(\lambda).$$

Final sampling rule (Exponential):

$$\boxed{U \sim \mathrm{Unif}(0,1) \quad \Rightarrow \quad -\frac{1}{\lambda}\log(1-U) \sim \mathrm{Exp}(\lambda).}$$

Implementation in R

Example: generate n samples from $\mathrm{Exp}(\lambda)$ using inverse transform.

```
U <- runif(n)  U
~ Unif(0,1) X <- -(1/lambda) * log(1 - U)  X ~ Exp(lambda)
Compare with built-in:  X_builtin <- rexp(n, rate = lambda)
```

What `rexp()` does conceptually: generate Uniform$(0, 1)$, then apply a fast implementation of $\log(\cdot)$ and the inverse-CDF formula.

Where inverse transform breaks down

Inverse transform sampling requires:

$$\text{(i) } F(x) \text{ is tractable} \qquad \text{and} \qquad \text{(ii) } F^{-1}(u) \text{ is tractable.}$$

Many common families fail one (or both) of these in closed form:

- Normal: F involves a non-elementary integral; F^{-1} not closed form
- Beta/Gamma: F uses incomplete beta/gamma functions; F^{-1} generally numerical

So we need a second tool: rejection (envelope) sampling.

Rejection sampling: intuition

Setting: we want samples from a target density $\pi(x)$, but cannot sample from it directly.

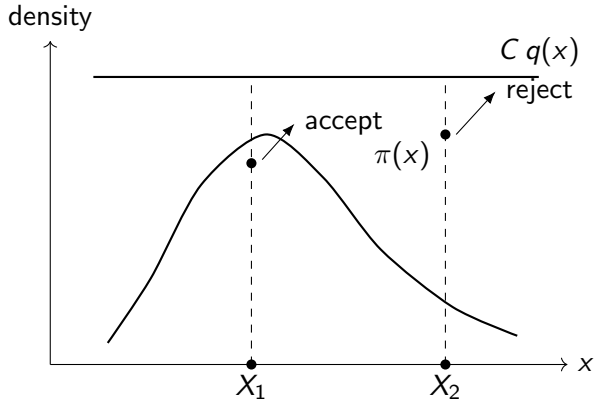
We choose a **proposal/envelope** density $q(x)$ that *we can* sample from, and a constant $C > 0$ such that

$$\frac{\pi(x)}{q(x)} \leq C \quad \text{for all } x.$$

Geometric picture:

- sample $X \sim q$
- generate a uniform "height" and accept if it falls under the target curve
- rejected samples are thrown away

Rejection sampling: the picture



Acceptance is more likely where $\pi(x)$ is large relative to the envelope.

Rejection sampling: formal condition

We want to sample from target density $\pi(x)$.

Assume we can sample from proposal density $q(x)$ and that there exists $C > 0$ such that

$$\frac{\pi(x)}{q(x)} \leq C \quad \text{for all } x \text{ where } q(x) > 0.$$

Equivalently:

$$\pi(x) \leq C\, q(x) \quad \text{for all } x.$$

Interpretation: Cq is an *envelope* that sits above π everywhere.

Rejection sampling: algorithm (step-by-step)

Inputs: target π, proposal q, constant C with $\pi(x) \leq Cq(x)$.

Repeat until you accept:

1. Sample $X \sim q(x)$.
2. Sample $U \sim \mathrm{Unif}(0, 1)$.
3. Compute acceptance probability

$$\alpha(X) = \frac{\pi(X)}{C\, q(X)}.$$

4. **Accept** X if $U \leq \alpha(X)$; otherwise **reject** and try again.

The accepted X is a valid sample from π.

Why the constant C matters

C scales the proposal: Cq must dominate π.

Bigger $C \Rightarrow$ easier to satisfy $\pi \leq Cq$ but:

$$\alpha(X) = \frac{\pi(X)}{Cq(X)} \text{ gets smaller } \Rightarrow \text{ more rejections.}$$

Smaller C (closer envelope) $\Rightarrow$ higher acceptance rate, but might fail the dominance condition.

Design principle: choose q and C so that Cq tightly "hugs" π.

Efficiency: three qualitative cases

Think of Cq as a shape over π.

- **Very inefficient:** envelope is far above π (large wasted area)
- **Reasonably efficient:** envelope just touches the maximum of π
- **Very efficient:** envelope closely matches π across its support

Rule of thumb: acceptance rate is roughly

$$\text{Acc} \approx \frac{\text{area under } \pi}{\text{area under } Cq}.$$

Since π is a density, $\int \pi(x)\,dx = 1$, so

$$\text{Acc} = \frac{1}{C} \quad \text{if } q \text{ is a normalized density and } C \text{ is valid.}$$

(We will formalize expected acceptance and expected trial counts later.)

Proof sketch that rejection sampling works

Let $X \sim q$ and $U \sim \mathrm{Unif}(0,1)$ independent. We accept when

$$U \leq \frac{\pi(X)}{Cq(X)}.$$

Consider the (unnormalized) density of an accepted draw at value x:

$$\mathbb{P}(X \in dx \text{ and accept}) = q(x)\,dx \cdot \mathbb{P}\left(U \leq \frac{\pi(x)}{Cq(x)}\right).$$

Since $U \sim \mathrm{Unif}(0,1)$,

$$\mathbb{P}\left(U \leq \frac{\pi(x)}{Cq(x)}\right) = \frac{\pi(x)}{Cq(x)}.$$

Therefore,

$$\mathbb{P}(X \in dx \text{ and accept}) = q(x)\,dx \cdot \frac{\pi(x)}{Cq(x)} = \frac{1}{C}\,\pi(x)\,dx.$$

Conditioning on acceptance removes the factor $1/C$, so the accepted X has density $\pi(x)$.

Putting it together: what you should be able to do

After this lecture (and lab), you should be able to:

- classify generators as true vs pseudo-random (and explain why)
- use inverse transform sampling when you can compute F and F^{-1}
- derive the inverse-CDF sampler for $\mathrm{Exp}(\lambda)$
- explain why inverse transform fails for many common distributions
- state the rejection sampling condition $\pi(x) \leq Cq(x)$
- implement accept/reject logic and discuss efficiency qualitatively

Quick checks (for discussion)

1. If $U \sim \mathrm{Unif}(0,1)$, why is $1-U$ also Uniform$(0,1)$?
2. For $\mathrm{Exp}(\lambda)$, show that using $-\frac{1}{\lambda}\log(U)$ also works.
3. In rejection sampling, what happens if $q(x) = 0$ somewhere that $\pi(x) > 0$?
4. Why is a "tight" envelope good computationally?

Lab preview: what you will code

Inverse transform sampling lab tasks typically look like:

- derive F and F^{-1} for a given distribution (when possible)
- generate many samples using $F^{-1}(U)$
- validate: histogram vs theoretical density, sample mean/variance checks

Rejection sampling lab tasks typically look like:

- choose a proposal q and constant C
- implement accept/reject
- estimate acceptance rate empirically
- compare different envelopes for efficiency

Pseudocode: Inverse transform sampling

Given: continuous CDF F and quantile function F^{-1}.

Algorithm

1. Draw $U \sim \mathrm{Unif}(0, 1)$.
2. Return $X = F^{-1}(U)$.

Guarantee: X has CDF F.

Pseudocode: Rejection sampling

Given: target density π, proposal density q, constant C such that $\pi(x) \leq Cq(x)$.

Algorithm

1. Repeat:
   1. Sample $X \sim q$.
   2. Sample $U \sim \mathrm{Unif}(0, 1)$.
   3. If $U \leq \pi(X)/(Cq(X))$, **accept** and output X.

Guarantee: accepted X has density π.