Exercises 4.2–4.5: Inverse Transform & Rejection Sampling

Big picture

- **Inverse transform sampling**: derive F and compute F^{-1}, then set $X = F^{-1}(U)$ with $U \sim \text{Unif}(0, 1)$.
- **Rejection sampling**: choose easy proposal q, find c such that $\pi(x) \leq cq(x)$, accept with probability $\pi(y)/(cq(y))$.

Exercise 4.2: normalise and invert

Given:

$$\pi(x) = ax^2, \quad x \in [0,1].$$

Normalisation:

$$1 = \int_0^1 ax^2\,dx = a \cdot \frac{1}{3} \quad \Rightarrow \quad a = 3.$$

CDF:

$$F(x) = \begin{cases} 0, & x < 0, \\ x^3, & 0 \le x \le 1, \\ 1, & x > 1. \end{cases} \qquad F^{-1}(u) = u^{1/3}.$$

Exercise 4.2 (R code, part 1/3): comments + simulate

```
#Integrating over [0, 1] and setting the result equal to 1, gives a = 3.
#The CDF is 0 for x < 0, x^3 for 0 <= x <= 1 and 1 for x > 1.
#The inverse function for the inverse transform method is u^1/3.

#Generate random uniform samples
u <- runif(10000)

#Generate samples from required distribution
x <- u^(1/3)
```

Exercise 4.2 (R code, part 2/3): histogram + density overlay

```
#Plot results
X <- seq(0, 1, 0.01)
hist(x, freq = FALSE, main = "", breaks = 20)
lines(X, 3*X^2)
```

Exercise 4.2: what you should see

Histogram should match the curve $\pi(x) = 3x^2$ on $[0, 1]$:

- few samples near 0,
- many samples near 1,
- density increases like x^2.

Exercise 4.3: inverse transform

Given:

$$\pi(x) = \frac{1}{\theta} x^{\frac{1-\theta}{\theta}} = \frac{1}{\theta} x^{\frac{1}{\theta}-1}, \qquad x \in [0, 1].$$

CDF:

$$F(x) = x^{1/\theta} \quad \Rightarrow \quad F^{-1}(u) = u^{\theta}.$$

We generate $10{,}000$ samples for $\theta \in \{1, 5, 10\}$.

Exercise 4.3 (R code, part 1/4): comments + function

```
#The inverse function is F^-1(x) = x^\theta for x \in [0, 1].

# Inverse Transform Function ---------------------------------------------

#This function take in a vector of U[0, 1] numbers and a value for theta
#and returns samples from the distributions 1/theta x^{(1-theta)/theta}.

inverse.transform.function <- function(u, theta){

  #Transform using inverse distribution function
  x <- u^theta

  return(x)
}
```

Exercise 4.3 (R code, part 2/4): generate samples

```
#Generate random uniform samples
u <- runif(10000)

#Generate samples from required distribution
x1 <- inverse.transform.function(u, 1)
x5 <- inverse.transform.function(u, 5)
x10 <- inverse.transform.function(u, 10)
```

Exercise 4.3 (R code, part 3/4): theta = 1 and theta = 5

```
#Plot results
X <- seq(0, 1, 0.01)
hist(x1, freq = FALSE, main = "", breaks = 20)
lines(X, rep(1, length(X)))

#Plot results
X <- seq(0, 1, 0.01)
hist(x5, freq = FALSE, main = "", breaks = 100)
lines(X, 0.2*X^-0.8)
```

Exercise 4.3 (R code, part 4/4): theta = 10

```
#Plot results
X <- seq(0, 1, 0.01)
hist(x10, freq = FALSE, main = "", breaks = 100)
lines(X, 0.1*X^-0.9)
```

Exercise 4.4: rejection sampling on $[0, 1]$

Target: $\pi(x) = 3x^2$ on $[0, 1]$.
Use proposal $q(x) = 1$ on $[0, 1]$ (Uniform).

$$\text{Need } c \geq \sup_{x\in[0,1]} \frac{\pi(x)}{q(x)} = \sup 3x^2 = 3.$$

So $c = 3$ is optimal; your code uses $c = 4$ (still valid, slightly less efficient).

Exercise 4.4 (R code, part 1/4): visualise + choose c

```
#It helps to plot the distribution so we can get an idea of what value c to use.
#It looks like a u[0, 1] distribution with c >=3 will work here. I'll use c=4, but 3 is the most
     efficient.
X <- seq(0, 1, 0.01)
plot(X, 3*X^2, type = 'l')

c <- 4
```

Exercise 4.4 (R code, part 2/4): initialise

```
x <- rep(0,10000)
count <- 1
```

Exercise 4.4 (R code, part 3/4): rejection loop

```
while (count < 10001){
  #Generate from Q (uniform)
  y <- runif(1)

  # Calculate the probability density function values
  f <- 3*y^2

  #Generate acceptance probabilities
  k <- f/(c*1)

  # Accept samples based on the acceptance/rejection criterion
  u <- runif(1)
  if (u < k){
    x[count] <- y
    count <- count +1
  }
}
```

Exercise 4.4 (R code, part 4/4): check vs true density

```
#Plot results
hist(x, freq = FALSE, main = "", breaks = 20)
lines(X, 3*X^2)
```

Exercise 4.5: half-normal via Exp(1) proposal

Target (half-normal):

$$\pi(x) = \frac{2}{\sqrt{2\pi}} \exp(-x^2/2), \qquad x \geq 0.$$

Proposal: $q(x) = \lambda e^{-\lambda x}$.
For $\lambda = 1$, the envelope constant is

$$c = \sup_{x \geq 0} \frac{\pi(x)}{q(x)} = \frac{2}{\sqrt{2\pi}} e^{1/2}.$$

(That is the value used in your code.)

Exercise 4.5 (R code, part 1/4): plot target + envelope

```
#Let us use lambda=1
#It helps to plot the distribution so we know that our value of c is correct
X <- seq(0, 5, 0.01)
plot(X, 2/(sqrt(2*pi))*exp(-X^2/2), type = 'l')
lines(X, 2/sqrt(2*pi)*exp(1/2)*dexp(X, 1), col = 2)
```

Exercise 4.5 (R code, part 2/4): propose + compute f

```
#Generate from Q (uniform)
y <- rexp(10000, 1)

# Calculate the probability density function values
f <- 2/(sqrt(2*pi))*exp(-y^2/2)
```

Exercise 4.5 (R code, part 3/4): accept/reject

```
#Generate acceptance probabilities
c <- 2/sqrt(2*pi)*exp(1/2)
k <- f/(c*dexp(y, 1)) #q(y) = 1

# Generate random samples from a uniform distribution for acceptance/rejection
u <- runif(10000)

# Accept samples based on the acceptance/rejection criterion
x <- y[u < k]
```

Exercise 4.5 (R code, part 4/4): check vs true density

```
#Plot results
hist(x, freq = FALSE, main = "", breaks = 20)
lines(X, 2/(sqrt(2*pi))*exp(-X^2/2))
```

Summary

- Ex 4.2: normalise to get $a = 3$, then $X = U^{1/3}$.
- Ex 4.3: $F^{-1}(u) = u^\theta$ for $\theta \in \{1, 5, 10\}$.
- Ex 4.4: rejection on $[0, 1]$ with Uniform proposal; $c = 3$ optimal (your code uses $c = 4$).
- Ex 4.5: rejection with Exp(1) proposal; envelope constant $c = \frac{2}{\sqrt{2\pi}} e^{1/2}$.