# Probability, Computation and Simulation | Homework 1

Brian Cervantes Alvarez
October 6, 2024

## Problem 2

Write a computer program that, when given a probability mass function $\{p_j, j = 1, \dots, n\}$ as an input, gives as an output the value of a random variable having this mass function.

### Pseudocode

1. Enter a vector of probabilities $p$.
2. Generate a random number $u$ from a $Uniform(0, 1)$.
3. Compute the cdf of $p$.
4. Find the index $j$ such that the cumulative sum of $p$ is greater than or equal to $u$.
5. Output the value $j$.

### Putting it in practice

```r
set.seed(202425)
p <- c(0.1, 0.3, 0.6)
u <- runif(1)
j <- which.max(cumsum(p) >= u)
j
```

```
[1] 3
```

## Problem 4

A deck of 100 cards—numbered $1, 2, \dots, 100$—is shuffled and then turned over one card at a time. Say that a "hit" occurs whenever card $i$ is the $i$-th card to be turned over, $i = 1, \dots, 100$. Write a simulation program to estimate the expectation and variance of the total number of hits.

**Pseudocode**

1. Let $N$ be the number of simulations
2. Initialize a counter for hits.
3. For each simulation:
    - Shuffle the deck.
    - Count the number of hits where card $i$ is in position $i$.
4. Compute the expectation and variance of the total number of hits.

**Putting it in practice**

```r
set.seed(202425)
N <- 10000
hits <- replicate(N, {
  deck <- sample(1:100)
  sum(deck == 1:100)
})
mean(hits)
```

```
[1] 1.0043
```

```r
var(hits)
```

```
[1] 0.9929808
```

## Problem 7

A pair of fair dice are to be continually rolled until all the possible outcomes $2, 3, \dots, 12$ have occurred at least once. Develop a simulation study to estimate the expected number of dice rolls.

**Pseudocode**

1. Let $N$ be the number of simulations
2. For each simulation:
    - Initialize an empty set of sums.
    - Repeatedly roll the dice until all sums from 2 to 12 are collected.
    - Count the number of rolls.
3. Compute the expected number of rolls.

**Putting it in practice**

```
set.seed(202425)
N <- 10000
rolls <- replicate(N, {
  outcomes <- integer(0)
  count <- 0
  while(length(unique(outcomes)) < 11) {
    outcome <- sum(sample(1:6, 2, replace = TRUE))
    outcomes <- c(outcomes, outcome)
    count <- count + 1
  }
  count
})
mean(rolls)
```

```
[1] 60.5079
```

## Problem 13

Give two methods for generating a random variable $X$ such that

$$P(X = i) = \frac{e^{-\lambda}\lambda^i/i!}{\sum_{j=0}^{k} e^{-\lambda}\lambda^j/j!}, \quad i = 0, \ldots, k$$

### Method 1: Inverse Transform

#### Pseudocode

1. Compute the cdf for the given probabilities.
2. Generate a random number $u$ from a $Uniform(0, 1)$.
3. Find the smallest $i$ such that the CDF at $i$ is greater than or equal to $u$.

#### Putting it in practice

```
set.seed(202425)
lambda <- 3
k <- 5
p <- dpois(0:k, lambda)
u <- runif(1)
i <- which.max(cumsum(p) >= u)
i
```

```
[1] 6
```

## Problem 13

Give two methods for generating a random variable $X$ such that

$$P(X = i) = \frac{e^{-\lambda}\lambda^i/i!}{\sum_{j=0}^{k} e^{-\lambda}\lambda^j/j!}, \quad i = 0, \dots, k$$

## Method 2: Acceptance-Rejection

### Pseudocode

1. Generate $i$ from the example distribution.
2. Compute the acceptance probability for $i$.
3. Accept the probability proportional to the target distribution. Otherwise, repeat.

### Putting it in practice

```r
set.seed(202425)
accept <- FALSE
while(!accept) {
  i <- sample(0:k, 1)
  u <- runif(1)
  if(u <= dpois(i, lambda) / max(dpois(0:k, lambda))) {
    accept <- TRUE
  }
}
i
```

```
[1] 2
```

# Problem 14

Let $X$ be a binomial random variable with parameters $n$ and $p$. Suppose that we want to generate a random variable $Y$ whose probability mass function is the same as the conditional mass function of $X$ given that $X \geq k$, for some $k \leq n$. Let $\alpha = P(X \geq k)$ and suppose that the value of $\alpha$ has been computed.

## Part A: Inverse Transform Method

### Pseudocode

1. Compute the conditional CDF for $X \geq k$.
2. Generate a random number $u$ from a uniform distribution.
3. Find the smallest $i$ such that the CDF at $i$ is greater than or equal to $u$.

### Putting it in practice

```r
set.seed(202425)
n <- 10
p <- 0.5
k <- 5
alpha <- pbinom(n, size = n, prob = p) - pbinom(k - 1, size = n, prob = p)
p_cond <- dbinom(k:n, size = n, prob = p) / alpha
u <- runif(1)
i <- which.max(cumsum(p_cond) >= u) + k - 1
i
```

```
[1] 7
```

## Problem 14

Let $X$ be a binomial random variable with parameters $n$ and $p$. Suppose that we want to generate a random variable $Y$ whose probability mass function is the same as the conditional mass function of $X$ given that $X \geq k$, for some $k \leq n$. Let $\alpha = P(X \geq k)$ and suppose that the value of $\alpha$ has been computed.

### Part B: Rejection Sampling Method

#### Pseudocode

1. Generate a value $X$ from a binomial distribution.
2. If $X \geq k$, accept $X$. Otherwise, repeat.

#### Putting it in practice

```r
set.seed(202425)
accept <- FALSE
while(!accept) {
  X <- rbinom(1, size = n, prob = p)
  if(X >= k) {
    accept <- TRUE
  }
}
X
```

```
[1] 7
```