Seminar 1 : Probabilistic Models

Carlos Buitrago

Table of contents

1 Probability Space

A **probability space** is a triple $(\Omega, \mathcal{F}, P)$ consisting of:

1. A **sample space** Ω, the set of all possible outcomes.
2. A σ**-algebra** $\mathcal{F}$ of subsets of Ω, whose elements are called events.
3. A **probability measure** $P : \mathcal{F} \to [0, 1]$.

A collection $\mathcal{F} \subseteq 2^{\Omega}$ is called a σ**-algebra** if:

- $\Omega \in \mathcal{F}$;
- $A \in \mathcal{F} \implies A^c \in \mathcal{F}$;
- if $A_1, A_2, ... \in \mathcal{F}$, then $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$.

A function $P : \mathcal{F} \to [0, 1]$ is called a **probability measure** if:

- $P(\Omega) = 1$;
- for every pairwise disjoint sequence $A_1, A_2, ... \in \mathcal{F}$,

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n).$$

From the axioms one obtains the basic properties

$$P(A^c) = 1 - P(A),$$

$$A \subseteq B \quad \implies \quad P(A) \leq P(B),$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

1.0.1 Continuity of Probability

If $A_1 \subseteq A_2 \subseteq A_3 \subseteq \cdots$, then

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \lim_{n\to\infty} P(A_n).$$

Similarly, if $A_1 \supseteq A_2 \supseteq A_3 \supseteq \cdots$, then

$$P\left(\bigcap_{n=1}^{\infty} A_n\right) = \lim_{n\to\infty} P(A_n).$$

These properties are known as **continuity from below** and **continuity from above**.

1.0.2 Inclusion–Exclusion Principle

One of the most important formulas in probability theory is the inclusion–exclusion principle.

For events $A_1, \ldots, A_n \in \mathcal{F}$,

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i) - \sum_{1\leq i<j\leq n} P(A_i \cap A_j) + \sum_{1\leq i<j<k\leq n} P(A_i \cap A_j \cap A_k) - \cdots + (-1)^{n-1} P(A_1 \cap \cdots \cap A_n).$$

---

2 Union Bound

A useful consequence of the inclusion–exclusion principle is the following estimate.

For any events $A_1, \dots, A_n \in \mathcal{F}$,

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i).$$

This inequality is known as the **union bound** or **Boole's inequality**.

Although simple, it is one of the most powerful tools in probability theory and appears throughout the course.

3 Classical Probability Model

The **classical model** is defined by the following assumptions:

- $\Omega = \{\omega_1, \dots, \omega_n\}$ is a finite sample space;
- $\mathcal{F} = 2^{\Omega}$;
- all elementary outcomes are equally likely:

$$P(\{\omega_i\}) = \frac{1}{n}, \qquad i = 1, \dots, n.$$

Consequently, for every event $A \subseteq \Omega$,

$$P(A) = \frac{|A|}{|\Omega|}.$$

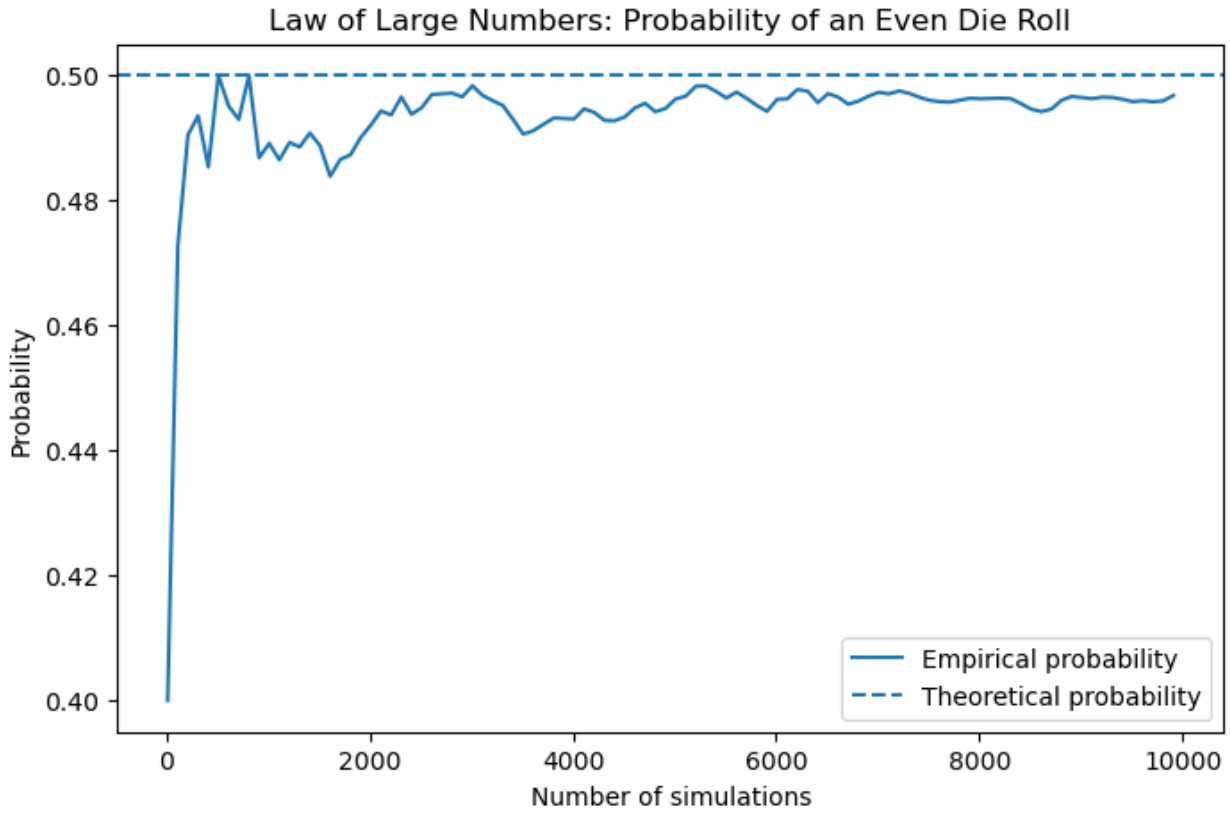
More generally, if Ω is finite or countable and $\mathcal{F} = 2^{\Omega}$, then a probability measure is completely determined by the probabilities of the elementary outcomes $P(\{\omega\}), \omega \in \Omega$.

3.1 Example: Die Roll

Consider one fair die roll. Then $\Omega = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{F} = 2^{\Omega}$, and $P(\{\omega\}) = 1/6$ for every $\omega \in \Omega$.

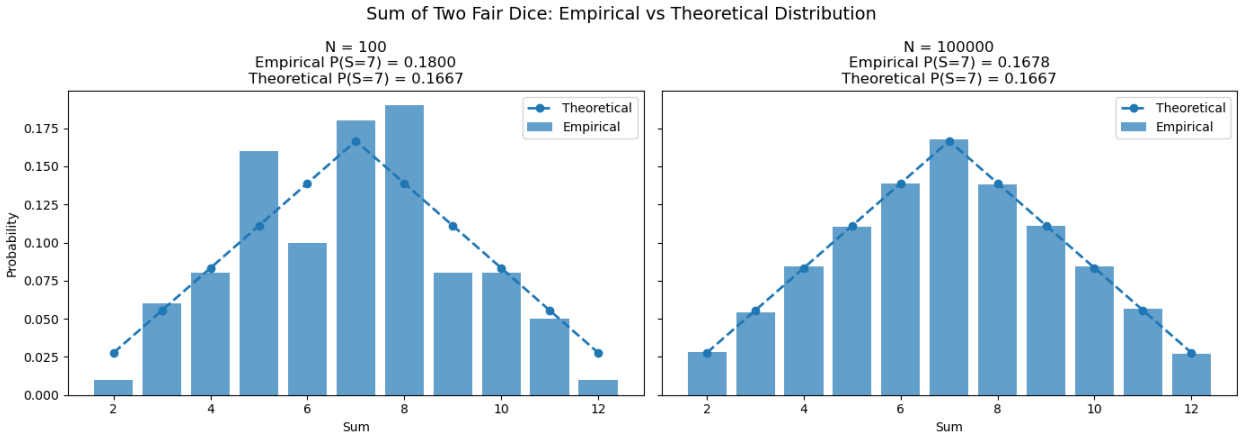
Let $A = \{\text{the result is even}\} = \{2, 4, 6\}$. Then $P(A) = |A|/|\Omega| = 3/6 = 1/2$.

```
Empirical probability: 0.4962
Theoretical probability: 0.5
```



3.2 Example: Two Dice

Roll two fair dice. Then $\Omega = \{1, \dots, 6\}^2$ and $|\Omega| = 36$. Let $A = \{\text{the sum is } 7\}$. The favorable outcomes are $(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)$, hence $|A| = 6$ and $P(A) = 6/36 = 1/6$.




```
Theoretical P(S=7) = 0.166667
N=   100: Empirical P(S=7) = 0.180000
N=100000: Empirical P(S=7) = 0.167760
```

3.3 Example: Cards

Suppose we draw cards from a standard deck of 52 cards. The probability of drawing an ace is $4/52 = 1/13$.

If two cards are drawn without replacement, then the probability that both cards are aces is

$$\frac{4}{52} \cdot \frac{3}{51}.$$

```
Empirical probability: 0.00475
Theoretical probability: 0.004524886877828055
```

4 Geometric Probability Model

Let $\Omega \subseteq \mathbb{R}^d$ be a measurable set with finite Lebesgue measure $\mu(\Omega) < \infty$. The **geometric probability model** is defined by

$$\mathcal{F} = \{A \subseteq \Omega : \ A \text{ is measurable}\},$$

and

$$P(A) = \frac{\mu(A)}{\mu(\Omega)}, \qquad A \in \mathcal{F}.$$

Thus, probabilities are proportional to geometric quantities such as length, area, or volume. Informally, a random point is chosen uniformly from Ω, and the probability of an event is the fraction of the total measure occupied by the corresponding region.

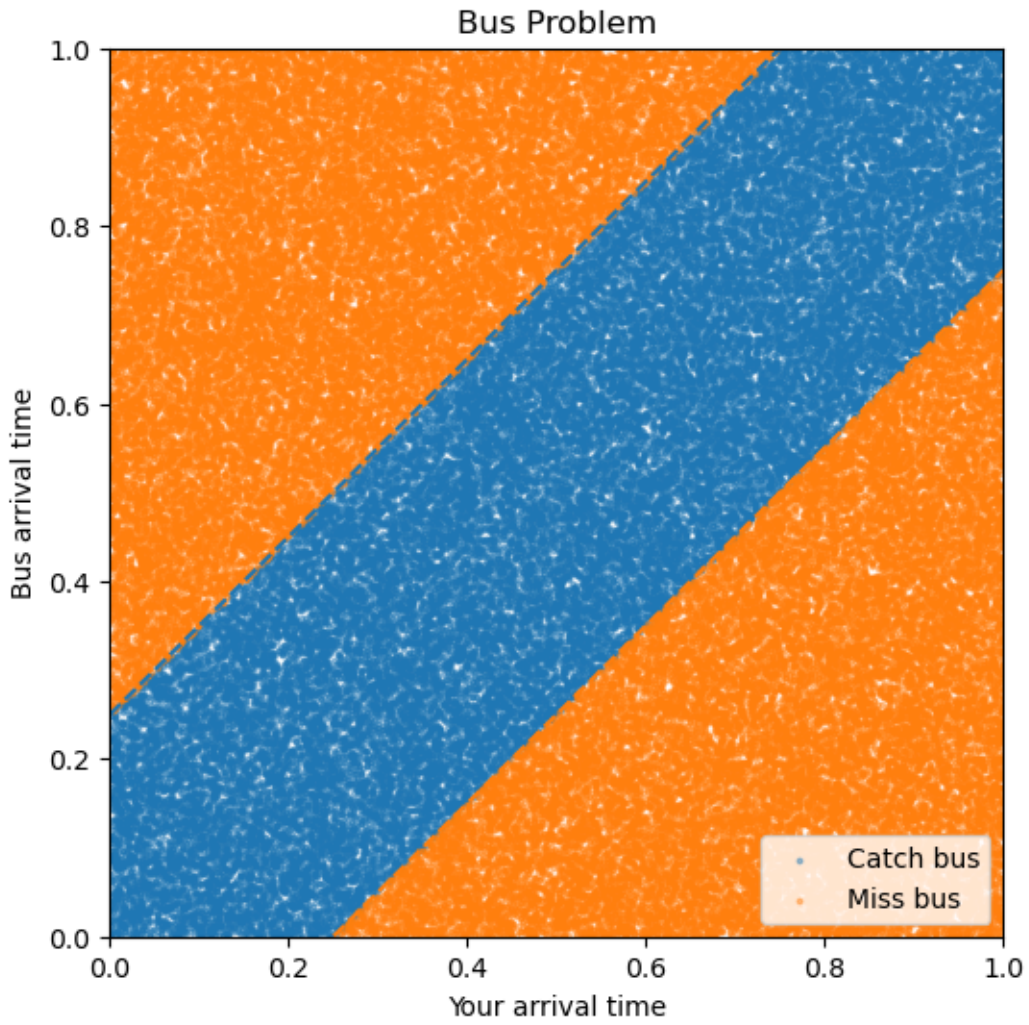
4.1 Example: Bus Problem

Suppose you and a bus arrive independently and uniformly at random between $12:00$ and $13:00$. The bus waits for 15 minutes before leaving, and you also wait 15 minutes if the bus has not arrived yet.

After rescaling time to $[0, 1]$, 15 minutes corresponds to $1/4$. You catch the bus if $|X - Y| \leq 1/4$. The bad region consists of two triangles with total area $2 \cdot (3/4)^2/2 = 9/16$. Therefore,

$$P(\text{catch the bus}) = 1 - \frac{9}{16} = \frac{7}{16}.$$

```
Empirical probability: 0.43745
Theoretical probability: 0.4375
```



4.2 Bertrand's Paradox

Consider a circle of radius R. We choose a chord at random and ask

$$P(\text{the chord is longer than the side of an inscribed equilateral triangle}) = ?$$

The side length of an inscribed equilateral triangle is $\sqrt{3}R$. If d is the distance from the center of the circle to the chord, then the chord length is $L = 2\sqrt{R^2 - d^2}$. Hence

$$L > \sqrt{3}R \quad \Longleftrightarrow \quad d < \frac{R}{2}.$$

The ambiguity is that the phrase "choose a chord at random" can mean different things.

Method 1: random endpoints. Choose two points independently and uniformly on the circumference and join them by a chord. This gives

$$P\left(L > \sqrt{3}R\right) = \frac{1}{3}.$$

Method 2: random distance from the center. Choose the distance d uniformly from $[0, R]$, then choose the direction of the chord uniformly at random. Since the chord is long exactly when $d < R/2$, this gives

$$P\left(L > \sqrt{3}R\right) = \frac{1}{2}.$$

Method 3: random midpoint. Choose the midpoint of the chord uniformly from the disk. The chord is long exactly when the midpoint lies inside the disk of radius $R/2$, so this gives

$$P\left(L > \sqrt{3}R\right) = \frac{\pi(R/2)^2}{\pi R^2} = \frac{1}{4}.$$

Thus the same informal question gives three different answers:

$$\frac{1}{3}, \qquad \frac{1}{2}, \qquad \frac{1}{4}.$$

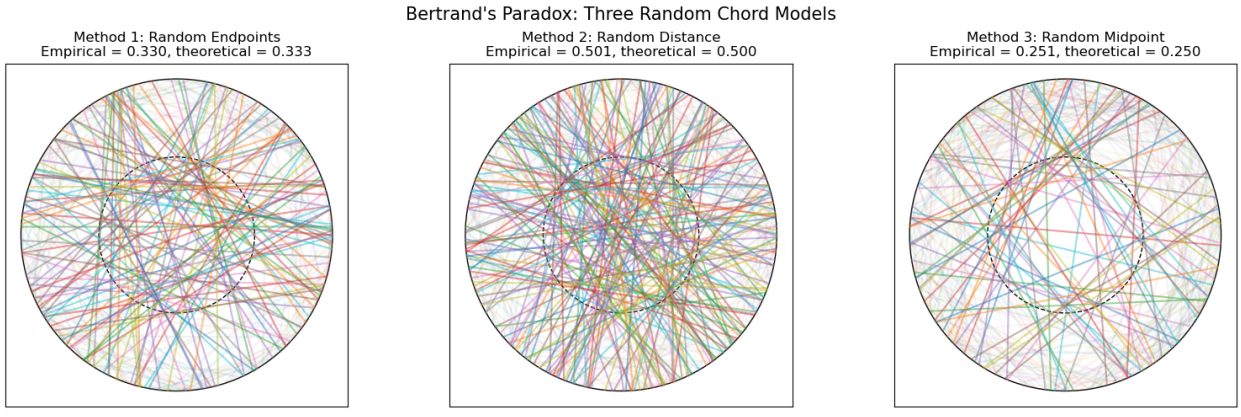
This is Bertrand's paradox. It shows that a probability problem is not well-defined until the probability space, or equivalently the random mechanism, is precisely specified.

```
Bertrand's Paradox

Method 1: Random endpoints
Empirical probability:   0.3299
Theoretical probability: 0.3333

Method 2: Random distance from center
Empirical probability:   0.5006
Theoretical probability: 0.5000

Method 3: Random midpoint in disk
Empirical probability:   0.2509
Theoretical probability: 0.2500
```



5 Bernoulli Scheme

Fix $n \in \mathbb{N}$ and $p \in (0, 1)$. The **Bernoulli scheme** consists of n independent trials, each resulting in success (1) with probability p and failure (0) with probability $1 - p$.

The sample space is

$$\Omega = \{(x_1, \dots, x_n) : x_i \in \{0, 1\}\},$$

so that $|\Omega| = 2^n$.

For an outcome $\omega = (x_1, \dots, x_n)$,

$$P(\{\omega\}) = p^{\sum_{i=1}^n x_i}(1-p)^{n-\sum_{i=1}^n x_i}.$$

The random variable

$$S_n = \sum_{i=1}^n x_i$$

counts the number of successes.

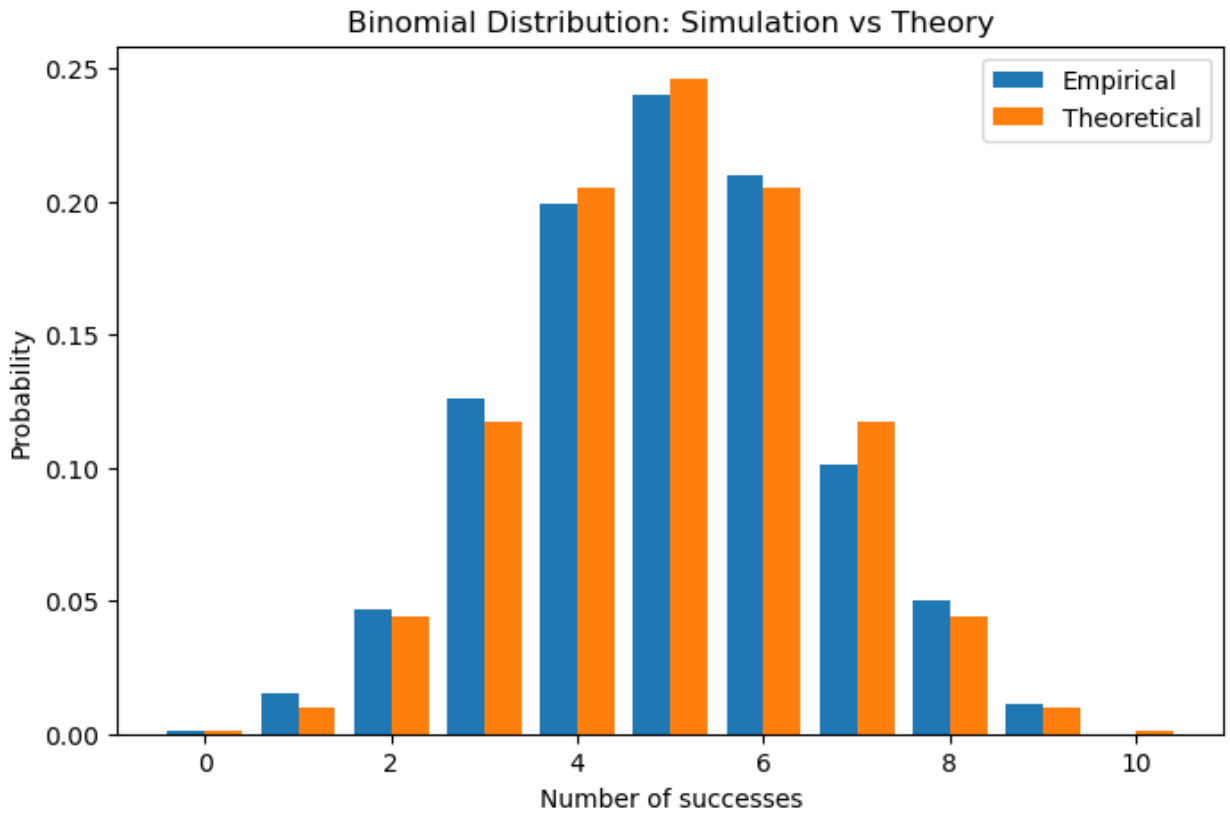
5.1 Example: Binomial Distribution

Let S_n be the number of successes in n independent Bernoulli trials with success probability p. Then

$$P(S_n = m) = \binom{n}{m} p^m (1-p)^{n-m}, \qquad m = 0, 1, \dots, n.$$

For example, if a fair coin is tossed 10 times, then the probability of exactly 5 heads is

$$P(S_{10} = 5) = \binom{10}{5} \left(\frac{1}{2}\right)^{10}.$$



5.2 Example: At Least One Success

If $S_n \sim \text{Bin}(n, p)$, then

$$P(S_n \geq 1) = 1 - P(S_n = 0) = 1 - (1-p)^n.$$

This formula is often easier than adding many probabilities.

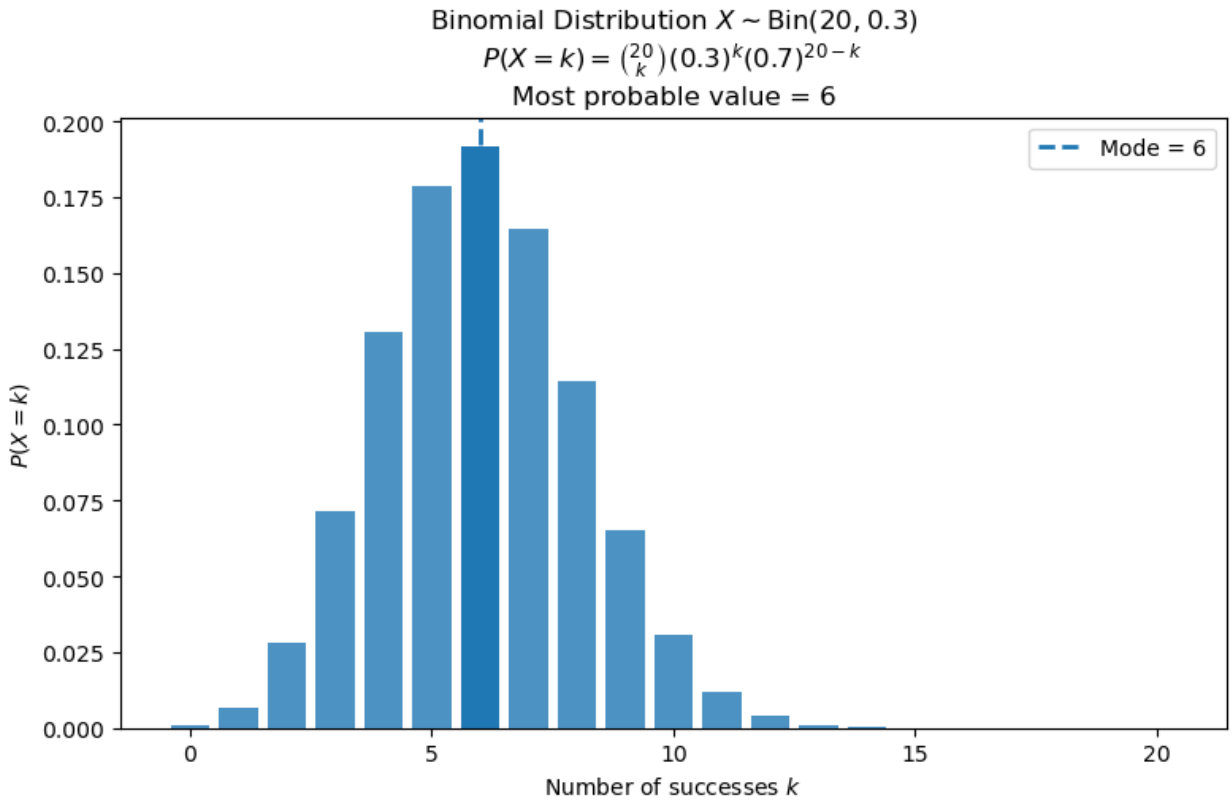
5.3 Example: Most Probable Number of Successes

If $S_n \sim \text{Bin}(n, p)$, then a most probable value is

$$m = \lfloor (n+1)p \rfloor.$$

If $(n+1)p$ is an integer, then both $(n+1)p - 1$ and $(n+1)p$ are modes.

```
Mode by formula: 6
Mode by computation: 6
```



6 Independence of Events

Let $A, B \in \mathcal{F}$. The events A and B are called **independent** if

$$P(A \cap B) = P(A)P(B).$$

Equivalently, whenever $P(B) > 0$,

$$P(A \mid B) = P(A).$$

A collection of events $A_1, \dots, A_n$ is called **pairwise independent** if

$$P(A_i \cap A_j) = P(A_i)P(A_j), \qquad i \neq j.$$

The events $A_1, \dots, A_n$ are called **mutually independent** if for every subset $I \subseteq \{1, \dots, n\}$ with $|I| \geq 2$,

$$P\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} P(A_i).$$

Mutual independence implies pairwise independence, but the converse is generally false.

6.1 Example: Dependent Events

Draw two cards without replacement from a standard deck. Let $A = \{\text{first card is an ace}\}$ and $B = \{\text{second card is an ace}\}$. Then

$$P(A) = \frac{4}{52}, \qquad P(B) = \frac{4}{52},$$

but

$$P(A \cap B) = \frac{4}{52} \cdot \frac{3}{51} \neq \frac{4}{52} \cdot \frac{4}{52}.$$

Therefore, A and B are not independent.

```
P(A): 0.07821
P(B): 0.07597
P(A and B): 0.00465
P(A)P(B): 0.0059416137
```

6.2 Example: Pairwise but Not Mutual Independence

Toss two fair coins and write the outcomes as $X, Y \in \{0, 1\}$. Define three events

$$A = \{X = 0\}, \qquad B = \{Y = 0\}, \qquad C = \{X = Y\}.$$

Then A, B, and C are pairwise independent. However,

$$P(A \cap B \cap C) = \frac{1}{4},$$

while

$$P(A)P(B)P(C) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}.$$

Therefore, the events are pairwise independent but not mutually independent.

7 How Are Random Numbers Generated by a Computer?

Throughout this course we will frequently use Python functions such as

```
np.random.random()
np.random.randint()
np.random.binomial()
```

to simulate random experiments. This naturally raises the following question:

> How can a computer generate random numbers if a computer is a deterministic machine?

7.1 True Randomness vs Pseudorandomness

A computer executes a sequence of deterministic instructions. If we start a program twice with exactly the same initial conditions, it will perform exactly the same calculations and produce exactly the same output.

For this reason, most "random" numbers generated by a computer are not truly random. Instead, they are **pseudorandom numbers**.

A pseudorandom number generator (PRNG) is a deterministic algorithm that produces a sequence of numbers that behaves *as if* it were random.

Given an initial value called a **seed**, the generator produces

$$x_0, x_1, x_2, ...$$

according to a deterministic rule

$$x_{n+1} = f(x_n).$$

If the same seed is used again, exactly the same sequence will be generated.

For example:

```
import numpy as np

np.random.seed(42)

print(np.random.random())
print(np.random.random())
print(np.random.random())
```

Running this code twice produces exactly the same numbers.

7.2 Why Does This Work?

Although pseudorandom numbers are generated deterministically, good generators are designed so that their outputs satisfy many statistical properties expected from genuinely random sequences.

For practical purposes, the sequence appears random:

- frequencies are approximately correct;
- correlations are extremely small;
- patterns are difficult to detect;
- many statistical tests cannot distinguish the sequence from true randomness.

As a result, pseudorandom generators are sufficient for most scientific simulations, machine learning algorithms, statistical experiments, computer graphics, and video games.

7.3 How Difficult Is It to Generate Truly Random Numbers?

Generating *true* randomness is surprisingly difficult.

A computer by itself cannot create randomness from nothing because its internal operations are deterministic.

To obtain genuine randomness, we must rely on unpredictable physical phenomena such as:

- radioactive decay;
- thermal noise in electronic circuits;
- atmospheric noise;
- photon arrival times;
- quantum measurements.

Such devices are called **hardware random number generators (HRNGs)**.

For example, some modern processors contain dedicated hardware that measures microscopic physical noise and converts it into random bits.

7.4 Why Is True Randomness Important?

For many simulations, pseudorandom numbers are perfectly adequate.

However, some applications require unpredictability rather than merely statistical randomness.

Examples include:

- cryptography;
- secure communications;
- online banking;
- digital signatures;
- password generation;
- blockchain technologies;
- military systems.

Suppose a cryptographic system generates passwords using a predictable pseudorandom generator. If an attacker discovers the algorithm and the seed, every future password can be predicted.

For this reason, cryptographic systems often combine pseudorandom generators with physical sources of randomness.

7.5 Randomness and Monte Carlo Methods

One of the most important applications of random numbers is **Monte Carlo simulation**.

The basic idea is simple:

1. Generate many random outcomes.
2. Compute the quantity of interest for each outcome.
3. Average the results.

For example, to estimate

$$P(A),$$

we can simulate the experiment many times and compute

$$\hat{P}(A) = \frac{\text{number of times } A \text{ occurs}}{\text{number of simulations}}.$$

The Law of Large Numbers guarantees that

$$\hat{P}(A) \to P(A) \qquad \text{as} \qquad N \to \infty.$$

This principle is used throughout science and engineering.

7.6 Random Numbers in Real Life

Random number generators play a crucial role in many modern technologies:

- weather forecasting;
- machine learning;
- artificial intelligence;
- computer graphics and animation;
- risk analysis in finance;
- epidemiological modeling;
- nuclear physics;
- cryptography and cybersecurity;
- simulation of communication networks;
- reliability analysis of complex systems.

Many scientific problems are so complicated that exact calculations are impossible, while Monte Carlo simulations using random numbers provide accurate approximations.

7.7 Historical Remark

During the 1940s, scientists working on the Manhattan Project needed large quantities of random numbers to simulate neutron transport inside nuclear reactors.

This led to the development of the **Monte Carlo method**, named after the famous casino in Monaco.

Today Monte Carlo methods are among the most powerful computational tools in mathematics, statistics, physics, engineering, finance, and data science.