

# 4 Introduction to probability theory

<b>4</b>	<b>Introduction to probability theory</b>	<b>1</b>
4.1	Axiomatic probability theory	2
4.1.1	Sample space	2
4.1.2	Axiomatic foundations: Sigma Algebra	3
4.1.3	Axiomatic foundation: Kolmogorov axioms	5
4.1.4	Sample space having equally likely outcomes	9
4.1.5	Interpretations of probability	10
4.2	Conditional probability	11
4.2.1	Properties of conditional probability	11
4.2.2	Bayes' theorem	12
4.2.3	Two independent events	14
4.2.4	Three independent events	15
4.2.5	Birthday paradox	16
4.2.6	Two probability paradoxes	17
4.3	Vibe Coding Project attempts	19
4.3.1	Vibe Coding Project 1: Dice simulator	19
4.3.2	Vibe Coding Project 2: Card drawing simulator	21
4.3.3	Vibe Coding Project 3: Visualizing birthday paradox	26
4.3.4	Vibe Coding Project 4: Monty Hall Problem simulator	27

I Reference. TBC

In this note, we introduce a formal axiomatic treatment of probability theory. For simplicity, we will not employ any measure theory in this note. A rigorous treatment with measure theory will be done later. In this note let's stick to Riemann integrals.

## 4.1 Axiomatic probability theory

### 4.1.1 Sample space

**Definition 4.1.** The set  $\Omega$  of all possible outcomes of a particular experiment is called the **sample space** of the experiment.

**Example 1.** In an experiment of coin tossing, we have two possible outcomes: head ( $H$ ) or tail ( $T$ ) [we do not accept a standing coin in this experiment]. Thus, the sample space is  $\Omega = \{H, T\}$ .



**Example 2.** In an experiment of throwing a standard six-faced die, the sample space is  $\Omega = \{1, 2, 3, 4, 5, 6\}$ .

**Remark.** *Dice* is the plural, and *die* is its singular form. The word **dices** is wrong!

**Example 3.** In an experiment of drawing a card from a standard 52-card deck,

- If we focus on its suit, then the possible outcomes are heart  $\heartsuit$  ( $H$ ), diamond  $\diamondsuit$  ( $D$ ), spade  $\spadesuit$  ( $S$ ) and club  $\clubsuit$  ( $C$ ). Then the sample space is  $\Omega = \{H, D, S, C\}$ .
- If we focus on its color, then the possible outcomes are red ( $R$ ) (includes heart  $\heartsuit$  and diamond  $\diamondsuit$ ) and black ( $B$ ) (includes spade  $\spadesuit$  and club  $\clubsuit$ ). Then the sample space is  $\Omega = \{R, B\}$ .
- If we focus on its index, then sample space is  $\Omega = \{A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K\}$

**Example 4.** In an experiment of measuring the reaction time to a certain stimulus, the possible outcomes can be any positive real number, so the sample space is  $\Omega = (0, \infty)$ .

**Example 5.** In an experiment of counting the total amount of banknotes (in HKD) in one's wallet, the possible outcomes can be any positive integral multiple of 10, so the sample space is  $\Omega = 10\mathbb{N} = \{10, 20, 30, \dots\}$ .

In the future we often need to partition the sample space. The definition of partition is recapped as follows:

**Definition 4.2.** Let  $\Omega$  be a set and  $\Pi \subset \mathcal{P}(\Omega)$ .  $\Pi$  is said to be a **partition** of  $\Omega$  if

- (1)  $\emptyset \notin \Pi$
- (2) Pairwise disjoint: If  $S, T \in \Pi$  are s.t.  $S \neq T$ , then  $S \cap T = \emptyset$ .
- (3)  $\Omega = \bigcup \Pi$

**Remark.** Recall the notation of union of a family: if  $\Pi$  is a family of sets, then  $\bigcup \Pi := \bigcup_{T \in \Pi} T$ .

**Example 6.** Consider  $\Omega = [0, \infty)$ .

- $\Pi = \{[0, a) : a \in \Omega\}$  is not a partition of  $\Omega$ . Even though

$$\bigcup \Pi = \bigcup_{a \in \Omega} [0, a) = \Omega,$$

they are not pairwise disjoint because  $0 \in [0, a) \forall a \in \Omega$ .

- $\Pi = \{[n, n+1) : n \in \omega\}$  is a partition of  $\Omega$ . Note that they are pairwise disjoint, and

$$\bigcup \Pi = \bigcup_{n \in \omega} [n, n+1) = \Omega.$$

Probability theory is all about random experiments, so it would be convenient if we could use programs to stimulate the experiments many times at once, for example 100,000 times in a run. Therefore in this note I will include a number of ‘Vibe Coding Projects’ that suggest some interesting simulations. Since the writer is learning Vibe Coding, these chunks are called this way. Readers of course may choose to write the programs themselves without the aid of AI. For readers who are interested in the writer’s attempt, you may go to the last subsection of the note.

#### Vibe Coding Projects 1. >>> dice\_simulator.py

- Task: Write a program to simulate rolling a fair die for  $k$  times, where  $k$  is our input, and then display the frequency of obtaining each outcome.

### 4.1.2 Axiomatic foundations: Sigma Algebra

We can intuitively define the probability of an event  $E \subset \Omega$  as follows: Repeat the experiment for  $n$  times, and let  $n(E)$  be the number of times that  $E$  occurs. Then we can define

$$\mathbb{P}(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n}.$$

However, this approach has two serious issues:

1. Why must the limit exist?
2. Even if this limit exists, why is it independent of the experiment?

Therefore it is desirable to have another way to define probability. As usual, we define it by a set of axioms. Before defining probability, we need to define the notion of **events**. Sometimes, we do not want some of the subsets of  $\Omega$  to be considered as events, because

1. sometimes  $\Omega$  can be really complicated, and to make things simpler, we restrict our attention to a particular class of subsets of  $\Omega$ .
2. sometimes not all subsets of  $\Omega$  can be assigned a probability, and we only consider those that we can assign a probability to as an event.

We will use a structure called **sigma algebra** to avoid these weird subsets:

**Definition 4.3.**  $\mathcal{B} \subset \mathcal{P}(\Omega)$  is a **sigma algebra** on  $\Omega$  if

- (a)  $\emptyset \in \mathcal{B}$
- (b) Closed under complement: If  $A \in \mathcal{B}$  then  $A^c \in \mathcal{B}$ .
- (c) Closed under countable union: If  $\{A_i\}_{i=1}^{\infty} \subset \mathcal{B}$ , then

$$\bigcup_{i=1}^{\infty} A_i \in \mathcal{B}.$$

**Lemma 4.4.** If  $\mathcal{B}$  is a sigma algebra on  $\Omega$ , then  $\Omega \in \mathcal{B}$ .

**Proof.** By (a),  $\emptyset \in \mathcal{B}$ . By (b),  $\Omega = \Omega \setminus \emptyset = \emptyset^c \in \mathcal{B}$ . ■

**Proposition 4.5** (Closed under countable intersection). If  $\mathcal{B}$  is a sigma algebra on  $\Omega$  and  $\{A_i\}_{i=1}^{\infty} \subset \mathcal{B}$ , then

$$\bigcap_{i=1}^{\infty} A_i \in \mathcal{B}.$$

**Proof.** De Morgan’s law states that

$$\bigcap_{i=1}^{\infty} A_i = \left( \bigcup_{i=1}^{\infty} A_i^c \right)^c.$$

By (b),  $A_i^c \in \mathcal{B} \forall i \in \mathbb{N}$ . By (c),  $\bigcup_{i=1}^{\infty} A_i^c \in \mathcal{B}$ . By (b) again,  $\bigcap_{i=1}^{\infty} A_i = \left( \bigcup_{i=1}^{\infty} A_i^c \right)^c \in \mathcal{B}$ . ■

**Corollary 4.6.** If  $\mathcal{B}$  is a sigma algebra on  $\Omega$  and  $A, B \in \mathcal{B}$ , then  $A \setminus B \in \mathcal{B}$ .

**Proof.** Observe  $A \setminus B = A \cap B^c$ . Thus by (b) and the proposition,  $A \setminus B \in \mathcal{B}$  as desired. ■

There can be many different sigma algebras on  $\Omega$ . Here we only study a few important examples. For a thorough treatment of sigma algebra, readers should refer to books on measure theory.

**Example 7.** Two trivial examples of sigma algebra:

- (1)  $\{\emptyset, \Omega\}$ .
- (2)  $\{\emptyset, A, A^c, \Omega\}$  where  $A \subset \Omega$ .

**Theorem 4.7.**  $\mathcal{P}(\Omega)$  is a sigma algebra on  $\Omega$ .

**Proof.**

- (a) By definition of power set,  $\emptyset \in \mathcal{P}(\Omega)$ .
- (b) Suppose  $A \in \mathcal{P}(\Omega)$ . Then

$$A^c = \Omega \setminus A = \{x \in \Omega : x \notin A\} \subset \Omega$$

and so  $A^c \in \mathcal{P}(\Omega)$ .

- (c) Suppose  $\{A_i\}_{i=1}^\infty \subset \mathcal{P}(\Omega)$ . We need to show  $\bigcup_{i=1}^\infty A_i \in \mathcal{P}(\Omega)$ , or equivalently  $\bigcup_{i=1}^\infty A_i \subset \Omega$ . Pick any  $x \in \bigcup_{i=1}^\infty A_i$ . Then  $x \in A_i$  for some  $i \in \mathbb{N}$ . Since  $A_i \in \mathcal{P}(\Omega)$ , we know  $A_i \subset \Omega$ , so  $x \in \Omega$  as desired. ■

This is the sigma algebra we will use in the case that  $\Omega$  is countable (possibly finite or infinite). When  $\Omega = \mathbb{R}$  which is uncountable, we would need a smaller sigma algebra to get rid of weird subsets, but large enough to contain all intervals.

**Theorem 4.8.** For any  $\mathcal{C} \subset \mathcal{P}(\Omega)$ , there exists a unique smallest sigma algebra on  $\Omega$  containing  $\mathcal{C}$ . This sigma algebra is said to be **generated by**  $\mathcal{C}$ , denoted by  $\sigma(\mathcal{C})$

**Proof.** Let  $\mathcal{F}$  be the family of all sigma algebras on  $\Omega$  containing  $\mathcal{C}$ . Our goal is to show that  $\bigcap \mathcal{F}$  is a sigma algebra; minimality is immediate as an intersection.

- (a) Since every  $\mathcal{B} \in \mathcal{F}$  is a sigma algebra,  $\emptyset \in \mathcal{B} \forall \mathcal{B} \in \mathcal{F}$ , and hence  $\emptyset \in \bigcap \mathcal{F}$ .
- (b) Suppose  $A \in \bigcap \mathcal{F}$ . Then  $A \in \mathcal{B} \forall \mathcal{B} \in \mathcal{F}$ . Since every  $\mathcal{B} \in \mathcal{F}$  is a sigma algebra,  $A^c \in \mathcal{B} \forall \mathcal{B} \in \mathcal{F}$ , and hence  $A^c \in \bigcap \mathcal{F}$ .
- (c) Suppose  $\{A_i\}_{i=1}^\infty \subset \bigcap \mathcal{F}$ . Then  $\{A_i\}_{i=1}^\infty \subset \mathcal{B} \forall \mathcal{B} \in \mathcal{F}$ . Since every  $\mathcal{B} \in \mathcal{F}$  is a sigma algebra,  $\bigcup_{i=1}^\infty A_i \in \mathcal{B} \forall \mathcal{B} \in \mathcal{F}$ , and hence  $\bigcup_{i=1}^\infty A_i \in \bigcap \mathcal{F}$ . ■

**Definition 4.9.** The sigma algebra on  $\mathbb{R}$  generated by  $\tau := \{G \subset \mathbb{R} : G \text{ is open}\}$  is called the **Borel sigma algebra** of  $\mathbb{R}$ , and we denote it by  $\sigma(\tau)$ .

**Lemma 4.10.**  $\sigma(\tau)$  contains all intervals.

**Proof.** We check every cases one-by-one.

- $\emptyset, \mathbb{R}, (-\infty, a), (a, b), (a, +\infty) \in \tau \subset \sigma(\tau)$ .
- $(-\infty, a] = (a, +\infty)^c \in \sigma(\tau)$  and  $[a, +\infty) = (-\infty, a)^c \in \sigma(\tau)$
- $[a, b] = (-\infty, b] \cap [a, +\infty) \in \sigma(\tau)$ .
- $(a, b] = (-\infty, b] \cap (a, +\infty) \in \sigma(\tau)$ .
- $[a, b) = (-\infty, b) \cap [a, +\infty) \in \sigma(\tau)$ .

**Theorem 4.11.**  $\sigma(\tau)$  is the sigma algebra generated by the open intervals.

**Proof.** Let  $\mathcal{B}$  be the sigma algebra generated by the open intervals. By the structure theorem of open sets, every nonempty open set in  $\mathbb{R}$  can be written as a disjoint union of open intervals. Thus  $\tau \subset \mathcal{B}$ . By minimality of  $\sigma(\tau)$ , we know  $\sigma(\tau) \subset \mathcal{B}$ . On the other hand, since every open interval is an open set, by minimality of  $\mathcal{B}$  we know  $\mathcal{B} \subset \sigma(\tau)$ . Therefore  $\mathcal{B} = \sigma(\tau)$ . ■

This is the sigma algebra we will always use on  $\mathbb{R}$ .

### 4.1.3 Axiomatic foundation: Kolmogorov axioms

We are in a position to define probability.

**Definition 4.12** (Kolmogorov axioms). Given a sample space  $\Omega$  and a sigma algebra  $\mathcal{B}$  on  $\Omega$ , a **probability measure** on  $\mathcal{B}$  is a function

$$\mathbb{P} : \mathcal{B} \rightarrow [0, 1]$$

that satisfies

- (a)  $\mathbb{P}(\Omega) = 1$
- (b) Countable additivity: If  $\{A_i\}_{i=1}^{\infty}$  is a sequence of pairwise disjoint sets in  $\mathcal{B}$ , then

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$$

The triple  $(\Omega, \mathcal{B}, \mathbb{P})$  is called a **probability space**, and the sets in  $\mathcal{B}$  are called **events**.

**Theorem 4.13.** If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space, then

- (i)  $\mathbb{P}(\emptyset) = 0$
- (ii) Finite additivity: If  $\{A_i\}_{i=1}^n$  is a finite sequence of pairwise disjoint sets in  $\mathcal{B}$ , then

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

**Proof.**

- (i) Setting  $A_i = \emptyset \forall i \in \mathbb{N}$ , by (b) we have

$$\mathbb{P}(\emptyset) = \sum_{i=1}^{\infty} \mathbb{P}(\emptyset)$$

which is possible only if  $\mathbb{P}(\emptyset) = 0$

- (ii) Setting  $A_i = \emptyset \forall i > n$ , by (i) and (b),

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

■

**Corollary 4.14.** If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $A \in \mathcal{B}$ , then  $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$ .

**Proof.**  $\{A, A^c\}$  is a partition of  $\Omega$ . So by (ii) and (a),

$$\mathbb{P}(A) + \mathbb{P}(A^c) = \mathbb{P}(A \sqcup A^c) = \mathbb{P}(\Omega) = 1.$$

■

**Proposition 4.15** (Monotonicity of probability). If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $A, B \in \mathcal{B}$  such that  $A \subset B$ , then  $\mathbb{P}(A) \leq \mathbb{P}(B)$ .

**Proof.** Note that  $B = A \cup (B \setminus A)$ . Since  $A \cap (B \setminus A) = \emptyset$ , so by (ii),

$$\mathbb{P}(B) = \mathbb{P}(A) + \mathbb{P}(B \setminus A) \geq \mathbb{P}(A).$$

■

**Lemma 4.16.** If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $A, B \in \mathcal{B}$ , then  $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$ .

**Proof.** Note that  $A \cup B = A \cup (B \setminus A)$ . Note that  $A \cap (B \setminus A) = \emptyset$ , so by (ii),

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B \setminus A).$$

Now note that  $B = (B \setminus A) \cup (B \cap A)$  where  $(B \setminus A) \cap (B \cap A) = \emptyset$ , by (ii) again, we have

$$\mathbb{P}(B) = \mathbb{P}(B \setminus A) + \mathbb{P}(B \cap A).$$

Combining these equations yields the desired identity. ■

**Theorem 4.17** (Principle of inclusion-exclusion for probability). If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $\{A_i\}_{i=1}^n \subset \mathcal{B}$ , then

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n \mathbb{P}(A_i) - \sum_{1 \leq i_1 < i_2 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2}) + \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2} \cap A_{i_3}) - \cdots + (-1)^{n+1} \mathbb{P}\left(\bigcap_{i=1}^n A_i\right)$$

**Proof.** We proceed by induction on  $n$ .

(i) The case  $n = 2$  follows from the lemma.

(ii) Suppose the formula holds for  $n$  events. Then

$$\begin{aligned} \mathbb{P}\left(\bigcup_{i=1}^{n+1} A_i\right) &= \mathbb{P}\left(A_{n+1} \cup \bigcup_{i=1}^n A_i\right) \\ &= \mathbb{P}(A_{n+1}) + \mathbb{P}\left(\bigcup_{i=1}^n A_i\right) - \mathbb{P}\left(A_{n+1} \cap \bigcup_{i=1}^n A_i\right) \\ &= \mathbb{P}(A_{n+1}) + \mathbb{P}\left(\bigcup_{i=1}^n A_i\right) - \mathbb{P}\left(\bigcup_{i=1}^n (A_i \cap A_{n+1})\right) \end{aligned}$$

Here we apply the inductive hypothesis to the two probabilities of unions to yield

$$\begin{aligned} \mathbb{P}\left(\bigcup_{i=1}^{n+1} A_i\right) &= \mathbb{P}(A_{n+1}) \\ &\quad + \sum_{i=1}^n \mathbb{P}(A_i) - \sum_{1 \leq i_1 < i_2 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2}) + \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2} \cap A_{i_3}) \\ &\quad - \cdots + (-1)^{n+1} \mathbb{P}\left(\bigcap_{i=1}^n A_i\right) \\ &\quad - \sum_{i=1}^n \mathbb{P}(A_i \cap A_{n+1}) + \sum_{1 \leq i_1 < i_2 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2} \cap A_{n+1}) - \sum_{1 \leq i_1 < i_2 < i_3 \leq n} \mathbb{P}(A_{i_1} \cap A_{i_2} \cap A_{i_3} \cap A_{n+1}) \\ &\quad + \cdots - (-1)^{n+1} \mathbb{P}\left(\bigcap_{i=1}^{n+1} A_i\right) \\ &= \sum_{i=1}^{n+1} \mathbb{P}(A_i) - \sum_{1 \leq i_1 < i_2 \leq n+1} \mathbb{P}(A_{i_1} \cap A_{i_2}) + \sum_{1 \leq i_1 < i_2 < i_3 \leq n+1} \mathbb{P}(A_{i_1} \cap A_{i_2} \cap A_{i_3}) \\ &\quad - \cdots + (-1)^{n+2} \mathbb{P}\left(\bigcap_{i=1}^{n+1} A_i\right) \end{aligned}$$

as desired. ■

**Corollary 4.18** (Bonferroni's inequality). If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $A, B \in \mathcal{B}$ , then

$$\mathbb{P}(A \cap B) \geq \mathbb{P}(A) + \mathbb{P}(B) - 1.$$

**Proof.** By the lemma, we have

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cup B) \geq \mathbb{P}(A) + \mathbb{P}(B) - 1.$$

■

**Example 8.** Let  $A, B$  be two events. Suppose  $\mathbb{P}(A) = 0.8$  and  $\mathbb{P}(B) = 0.9$ .

- (a) Find a lower bound for  $p = \mathbb{P}(A \cap B)$ .
- (b) Find an upper bound for the probability that exactly one of  $A$  or  $B$  occurs.

**Solution.**

- (a) By Bonferroni's inequality,

$$p \geq \mathbb{P}(A) + \mathbb{P}(B) - 1 = 0.8 + 0.9 - 1 = 0.7$$

- (b) Let  $G$  be the event that exactly one of  $A$  or  $B$  occurs, i.e.

$$G = (A \setminus B) \sqcup (B \setminus A).$$

Note that  $A = (A \cap B) \sqcup (A \setminus B)$  and  $B = (A \cap B) \sqcup (B \setminus A)$ , so

$$\mathbb{P}(A \setminus B) = \mathbb{P}(A) - \mathbb{P}(A \cap B) = 0.8 - p$$

$$\mathbb{P}(B \setminus A) = \mathbb{P}(B) - \mathbb{P}(A \cap B) = 0.9 - p$$

$$\mathbb{P}(G) = \mathbb{P}(A \setminus B) + \mathbb{P}(B \setminus A) = 1.7 - 2p \leq 0.3$$

■

**Theorem 4.19** (Countable subadditivity). If  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $\{A_i\}_{i=1}^{\infty} \subset \mathcal{B}$ , then

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

**Proof.** We write  $\bigcup_{i=1}^{\infty} A_i$  as a union of pairwise disjoint events by letting

$$B_1 = A_1 \quad \text{and} \quad B_{n+1} = A_{n+1} \setminus \left\{ \bigcup_{i=1}^n A_i \right\}$$

Then  $B_i$  are pairwise disjoint,  $B_i \subset A_i$  and

- $\bigcup_{i=1}^n A_i = \bigsqcup_{i=1}^n B_i \quad \forall n \in \mathbb{N}$ , and
- $\bigcup_{i=1}^{\infty} A_i = \bigsqcup_{i=1}^{\infty} B_i$ .

Therefore

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \mathbb{P}\left(\bigsqcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(B_i) \leq \sum_{i=1}^{\infty} \mathbb{P}(A_i).$$

■

**Theorem 4.20** (Continuity of probability). Suppose  $(\Omega, \mathcal{B}, \mathbb{P})$  is a probability space and  $\{A_i\}_{i=1}^{\infty} \subset \mathcal{B}$ .

(a) If  $\{A_i\}_{i=1}^{\infty}$  is an increasing sequence, i.e.  $A_1 \subset A_2 \subset \cdots \subset A_n \subset A_{n+1} \subset \cdots$ , then

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \lim_{n \rightarrow \infty} \mathbb{P}(A_n).$$

(b) If  $\{A_i\}_{i=1}^{\infty}$  is a decreasing sequence, i.e.  $A_1 \supset A_2 \supset \cdots \supset A_n \supset A_{n+1} \supset \cdots$ , then

$$\mathbb{P}\left(\bigcap_{i=1}^{\infty} A_i\right) = \lim_{n \rightarrow \infty} \mathbb{P}(A_n).$$

**Proof.**

(a) Again, we write  $\bigcup_{i=1}^{\infty} A_i$  as a union of pairwise disjoint events by letting

$$B_1 = A_1 \quad \text{and} \quad B_{n+1} = A_{n+1} \setminus \left\{ \bigcup_{i=1}^n A_i \right\}$$

Recall that  $B_i$  are pairwise disjoint,  $B_i \subset A_i$  and

- $A_n = \bigcup_{i=1}^n A_i = \bigsqcup_{i=1}^n B_i \quad \forall n \in \mathbb{N}$ , and
- $\bigcup_{i=1}^{\infty} A_i = \bigsqcup_{i=1}^{\infty} B_i$ .

Therefore

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \mathbb{P}\left(\bigsqcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(B_i) = \lim_{n \rightarrow \infty} \sum_{i=1}^n \mathbb{P}(B_i) = \lim_{n \rightarrow \infty} \mathbb{P}\left(\bigsqcup_{i=1}^n B_i\right) = \lim_{n \rightarrow \infty} \mathbb{P}(A_n)$$

(b) Note that  $A_1 \supset A_2 \supset \cdots \supset A_n \supset A_{n+1} \supset \cdots$  implies

$$A_1^c \subset A_2^c \subset \cdots \subset A_n^c \subset A_{n+1}^c \subset \cdots,$$

so (a) implies

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i^c\right) = \lim_{n \rightarrow \infty} \mathbb{P}(A_n^c).$$

By De Morgan's law,

$$\bigcup_{i=1}^{\infty} A_i^c = \left(\bigcap_{i=1}^{\infty} A_i\right)^c$$

so  $\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i^c\right) = 1 - \mathbb{P}\left(\bigcap_{i=1}^{\infty} A_i\right)$ . Also,  $\mathbb{P}(A_n^c) = 1 - \mathbb{P}(A_n)$ . This implies that

$$\mathbb{P}\left(\bigcap_{i=1}^{\infty} A_i\right) = \lim_{n \rightarrow \infty} \mathbb{P}(A_n).$$

■



#### 4.1.4 Sample space having equally likely outcomes

**Remark.** From now on we write  $\mathbb{P}\{s\}$  instead of  $\mathbb{P}(\{s\})$  for convenience.

Consider a probability space  $(\Omega, \mathcal{P}(\Omega), \mathbb{P})$  where  $\Omega = \{s_1, s_2, \dots, s_n\}$  is finite. Sometimes it is natural to make an assumption that all outcomes are **equally likely**, i.e.

$$\mathbb{P}\{s_1\} = \mathbb{P}\{s_2\} = \dots = \mathbb{P}\{s_n\}.$$

In this case, by the axioms we can deduce that  $\mathbb{P}\{s_i\} = \frac{1}{n} \forall i$ , and hence for any  $A \subset \Omega$ , we have

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

This is the **classical interpretation of probability**, which we encountered when we were in high school.

**Example 9.** If two fair 6-sided dice are rolled, what is the probability that the sum of the two outcomes is 8?

**Solution.** The sample space is  $\Omega := \{(i, j) \in \mathbb{Z}^2 : 1 \leq i, j \leq 6\}$ . Also, let  $A$  be the event that the sum of the two outcomes is 8. Then

$$A = \{(i, j) \in \Omega : i + j = 8\} = \{(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)\}.$$

So  $\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{5}{6^2} = \frac{5}{36}.$  ■

**Example 10.** A committee of 5 is to be selected from a group of 6 men and 9 women. If the selection is made randomly, what is the probability that the committee consists of 3 men and 2 women?

**Solution.** Let  $\Omega$  be the sample space, and let  $A$  be the event that the committee consists of 3 men and 2 women. Then

$$|\Omega| = \binom{6+9}{5} = \binom{15}{5} \quad \text{and} \quad |A| = \binom{6}{3} \binom{9}{2}.$$

So  $\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{\binom{6}{3} \binom{9}{2}}{\binom{15}{5}} = \frac{240}{1001}.$  ■

**Example 11.** In the game of bridge, the entire deck of 52 cards is dealt out to the 4 players.

- (a) What is the probability that one of the players receives all 13 spades?
- (b) What is the probability that every player receives a 2?

**Solution.** Let  $\Omega$  be the sample space. Then  $|\Omega| = \binom{52}{13} \binom{39}{13} \binom{26}{13} \binom{13}{13}.$

- (a) Let  $A$  be the event that one of the players receives all 13 spades. Then

$$|A| = \binom{39}{13} \binom{26}{13} \binom{13}{13} \times 4.$$

$$\text{Therefore } \mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{\binom{39}{13} \binom{26}{13} \binom{13}{13} \times 4}{\binom{52}{13} \binom{39}{13} \binom{26}{13} \binom{13}{13}} = \frac{4}{\binom{52}{13}} \approx 6.2991 \times 10^{-12}. \quad B$$

- (b) Let  $A$  be the event that every player receives a 2. Then

$$|B| = \binom{4}{1} \binom{48}{12} \times \binom{3}{1} \binom{36}{12} \times \binom{2}{1} \binom{24}{12} \times \binom{1}{1} \binom{12}{12}.$$

Therefore

$$\mathbb{P}(B) = \frac{|B|}{|\Omega|} = \frac{\binom{4}{1} \binom{48}{12} \times \binom{3}{1} \binom{36}{12} \times \binom{2}{1} \binom{24}{12} \times \binom{1}{1} \binom{12}{12}}{\binom{52}{13} \binom{39}{13} \binom{26}{13} \binom{13}{13}} = \frac{2197}{20825}.$$
 ■

**Example 12.** A deck of 52 cards is dealt out. What is the probability that the first 2 occurs in the 14<sup>th</sup> card?

**Proof.** Let  $\Omega$  be the sample space and  $A$  be the event that the first 2 occurs in the 14<sup>th</sup> card. Then

$$|\Omega| = 52! \text{ and } |A| = 48 \times 47 \times \cdots \times 36 \times 4 \times 38!.$$

$$\text{So } \mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{48 \times 47 \times \cdots \times 36 \times 4 \times 38!}{52!} = \frac{38 \times 37 \times 36 \times 4}{52 \times 51 \times 50 \times 49} = \frac{8436}{270725}.$$

### 4.1.5 Interpretations of probability

Here we introduce three common ways to think about probability. None of them fully explain the meaning of probability, but they all highlight some key features of probability.

#### 1. Classical interpretation.

This is the one we discussed in the previous subsection, where if  $\Omega$  is finite and we assume all outcomes are equally likely, then

$$\mathbb{P}(A) = \frac{|A|}{|\Omega|} \quad \forall A \subset \Omega$$

We can easily see two drawbacks of this approach:

- This interpretation does not work if the sample space is infinite, or if the outcomes are not equally likely.
- It is circular, because it relies on the notion of equally likely outcomes, which means the outcomes have the same probability. So we need to define probability in another way beforehand to avoid this logical fallacy.

#### 2. Frequentist interpretation.

In the frequentist approach, we define probability as the long-run proportion of times an event occurs. In other words, if  $n_0$  denoted the number of occurrence of event  $A$  in  $n$  trials of the experiment, then

$$\mathbb{P}(A) = \lim_{n \rightarrow \infty} \frac{n_0}{n}.$$

This is very intuitive, but it still has some drawbacks:

- It assumes that all experiments are repeatable, which is not true.
- It is in some sense circular, because it depends on the **law of large numbers**, which requires a definition of probability to prove.

#### 3. Bayesian interpretation.

In Bayesian approach, probabilities are **measures of belief** of an individual assessing the uncertainty of a particular situation. A common example is that in weather reports, the scientists may declare that ‘there is a 40% chance of rain this afternoon’. This approach carries a significant drawback:

- Different individuals may have different opinions, so they may assign different probabilities to the same event.

Although having this significant drawback, we can actually remediate it by updating our beliefs about something based on new evidence. Therefore, this interpretation is powerful when we are making decisions.

**Example 13.** Suppose we predict that a certain stock has a 60% chance of growing. However, a major scandal of the CEO of that company has been revealed. Under this evidence, we reduce our confidence and predict that the stock has only 40% chance of growing, for example.

In this note, we followed an axiomatic approach to define probability. This approach is free of interpretation. In other words, we can choose the interpretation we need based on the scenario.

#### Vibe Coding Projects 2. >>> card\_simulation.py

- Task: Write a program to simulate example 12 by the frequentist approach: when I input the total number of trials, the program returns a graph where the  $x$ -axis is the number of trials  $n$ , and the  $y$ -axis is the ratio 
$$\frac{\text{number of occurrence of event in the } n \text{ trials}}{n}.$$

## 4.2 Conditional probability

In example 13 about Bayesian interpretation of probability, the 40% chance of growth has a precondition of the scandal. This is an example of **conditional probability**. The formal definition of conditional probability is as follows:

**Definition 4.21.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space. Let  $A, B \in \mathcal{B}$  where  $\mathbb{P}(B) > 0$ . Then the **conditional probability** of  $A$  given  $B$ , denoted by  $\mathbb{P}(A|B)$ , is

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

In this notation, the 40% chance of growth should be written as

$$\mathbb{P}(\text{the stock grows} | \text{A major scandal of the CEO has been revealed}) = 40\%.$$

**Example 14.** Two dice are rolled. Given that the first die is a 3, what is the probability that the sum of the two outcomes is 8?

**Proof.** Let  $\Omega = \{(i, j) \in \mathbb{Z}^2 : 1 \leq i, j \leq 6\}$  be the sample space,  $A$  be the event that the first die is a 3, and  $B$  be the event that the sum of the two outcomes is 8. In other words,

$$A = \{(3, j) \in \Omega : 1 \leq j \leq 6\} = \{(3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6)\}$$

$$B = \{(i, j) \in \Omega : i + j = 8\} = \{(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)\}.$$

Thus  $\mathbb{P}(A) = \frac{6}{36} = \frac{1}{6}$  and  $\mathbb{P}(A \cap B) = \frac{1}{36}$ . Thus

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{1/36}{1/6} = \frac{1}{6}.$$

### 4.2.1 Properties of conditional probability

One should note that  $A \mapsto \mathbb{P}(A|B)$  is a probability measure. More precisely,

**Theorem 4.22.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space and  $B \in \mathcal{B}$  where  $\mathbb{P}(B) > 0$ . Then the function

$$\mathbb{Q} : \mathcal{B} \rightarrow [0, 1]$$

$$A \mapsto \mathbb{P}(A|B)$$

defines a probability measure on  $\mathcal{B}$ .

**Proof.**

(a) We need to show  $\mathbb{Q}(\Omega) = 1$ . Indeed,

$$\mathbb{Q}(\Omega) = \mathbb{P}(\Omega|B) = \frac{\mathbb{P}(\Omega \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(B)}{\mathbb{P}(B)} = 1.$$

(b) Countable additivity: Suppose  $\{A_i\}_{i=1}^{\infty}$  is a sequence of pairwise disjoint sets in  $\mathcal{B}$ . Then

$$\begin{aligned} \mathbb{Q}\left(\bigcup_{i=1}^{\infty} A_i\right) &= \mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i \mid B\right) = \frac{\mathbb{P}\left(B \cap \bigcup_{i=1}^{\infty} A_i\right)}{\mathbb{P}(B)} \\ &= \frac{\mathbb{P}\left(\bigcup_{i=1}^{\infty} (A_i \cap B)\right)}{\mathbb{P}(B)} \\ &= \sum_{i=1}^{\infty} \frac{\mathbb{P}(A_i \cap B)}{\mathbb{P}(B)} \\ &= \sum_{i=1}^{\infty} \mathbb{P}(A_i|B) \\ &= \sum_{i=1}^{\infty} \mathbb{Q}(A_i) \end{aligned}$$

So  $\mathbb{Q}$  satisfies the Kolmogorov axioms, and thus is a probability measure on  $\mathcal{B}$ .

**Example 15.** Let  $A, B$  be two events with  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . Suppose  $\mathbb{P}(A|B) = \frac{3}{4}$  and  $\mathbb{P}(B|A) = \frac{3}{8}$ . Let  $\mathbb{P}(A) = a$ .

(a) Express  $\mathbb{P}(A \cap B)$  and  $\mathbb{P}(B)$  in terms of  $a$ .

(b) Suppose  $\mathbb{P}(A^c \cap B^c) = \frac{7}{16}$ . Find  $a$ .

**Solution.**

$$(a) \mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B|A) = \frac{3}{8}a, \text{ and } \mathbb{P}(B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A|B)} = \frac{a}{2}.$$

$$(b) \frac{7}{16} = \mathbb{P}(A^c \cap B^c) = \mathbb{P}(A \cup B)^c = 1 - \mathbb{P}(A \cup B) \implies \mathbb{P}(A \cup B) = \frac{9}{16}. \text{ Hence by principle of inclusion-exclusion,}$$

$$\frac{9}{16} = \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B) = a + \frac{a}{2} - \frac{3}{8}a \implies a = \frac{1}{2}. \quad \blacksquare$$

By rewriting the definition, we get  $\mathbb{P}(A \cap B) = \mathbb{P}(B)\mathbb{P}(A|B)$ . This is a baby case of the **multiplicative rule**:

**Proposition 4.23** (Multiplicative rule). Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space. Let  $\{A_i\}_{i=1}^n \subset \mathcal{B}$  s.t.  $\mathbb{P}\left(\bigcap_{i=1}^{n-1} A_i\right) > 0$ .

Then

$$\mathbb{P}\left(\bigcap_{i=1}^n A_i\right) = \mathbb{P}(A_1)\mathbb{P}(A_2|A_1)\mathbb{P}(A_3|A_1 \cap A_2) \cdots \mathbb{P}(A_n|A_1 \cap A_2 \cap \cdots \cap A_{n-1}).$$

**Proof.**

$$\begin{aligned} & \mathbb{P}(A_1)\mathbb{P}(A_2|A_1)\mathbb{P}(A_3|A_1 \cap A_2) \cdots \mathbb{P}(A_n|A_1 \cap A_2 \cap \cdots \cap A_{n-1}) \\ &= \mathbb{P}(A_1) \times \frac{\mathbb{P}(A_1 \cap A_2)}{\mathbb{P}(A_1)} \times \frac{\mathbb{P}(A_1 \cap A_2 \cap A_3)}{\mathbb{P}(A_1 \cap A_2)} \times \cdots \times \frac{\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{n-1} \cap A_n)}{\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{n-1})} \\ &= \mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{n-1} \cap A_n) \end{aligned} \quad \blacksquare$$

## 4.2.2 Bayes' theorem

Note that by rewriting the definition, we get  $\mathbb{P}(A \cap B) = \mathbb{P}(B)\mathbb{P}(A|B) = \mathbb{P}(A)\mathbb{P}(B|A)$ . In other words,

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

This is the basis of the famous **Bayes' theorem**:

**Theorem 4.24** (Bayes' theorem). Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space,  $\{A_i\}$  be a countable partition of  $\Omega$  for which  $\mathbb{P}(A_i) > 0 \forall i$ , and  $B \in \mathcal{B}$  s.t.  $\mathbb{P}(B) > 0$ . Then for any  $j$ , it holds

$$\mathbb{P}(A_j|B) = \frac{\mathbb{P}(B|A_j)\mathbb{P}(A_j)}{\sum_i \mathbb{P}(B|A_i)\mathbb{P}(A_i)}$$

**Remark.** Recall that in this note, a countable set may be finite or countably infinite.

Note that  $\mathbb{P}(A_i) > 0$  is important here, even though we assumed  $A_i \neq \emptyset$  in the definition of partition. This is because a nonempty set may still have a zero probability, which we will explain later.

Bayes' theorem follows immediately from the following law:

**Theorem 4.25** (Law of total probability). Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space,  $\{A_i\}$  be a countable partition of  $\Omega$  for which  $\mathbb{P}(A_i) > 0 \forall i$ , and  $B \in \mathcal{B}$  s.t.  $\mathbb{P}(B) > 0$ . Then

$$\mathbb{P}(B) = \sum_i \mathbb{P}(B|A_i)\mathbb{P}(A_i)$$

**Proof.** Observe that

$$B = B \cap \Omega = B \cap \bigcup_i A_i = \bigcup_i (B \cap A_i).$$

$$\text{Thus } \mathbb{P}(B) = \mathbb{P}\left(\bigcup_i (B \cap A_i)\right) = \sum_i \mathbb{P}(B \cap A_i) = \sum_i \mathbb{P}(B|A_i)\mathbb{P}(A_i). \quad \blacksquare$$

Bayes' theorem is quite simple but it has profound consequences in probability and statistics, which we will illustrate below:

**Example 16.** Donated blood is screened for AIDS. Suppose the test has 99% accuracy, i.e. given that one has AIDS, there is a 99% chance that the test is positive. Suppose 1 in 10,000 people in the city have AIDS. The test also has a 5% false positive rating, i.e. given that one has no AIDS, there is a 5% chance that the test is positive. Now suppose one has a positive test result. What is the probability that he has AIDS?

**Proof.** Define the events:

- $S = \{\text{the test result is positive}\}$ ,  $S^c = \{\text{the test result is negative}\}$ .
- $T = \{\text{the person has AIDS}\}$ ,  $T^c = \{\text{the person has no AIDS}\}$ .

It is given that  $\mathbb{P}(S|T) = 99\%$ ,  $\mathbb{P}(T) = \frac{1}{10,000}$  and  $\mathbb{P}(S|T^c) = 5\%$ , and we need to find  $\mathbb{P}(T|S)$ . By Bayes' theorem,

$$\mathbb{P}(T|S) = \frac{\mathbb{P}(S|T)\mathbb{P}(T)}{\mathbb{P}(S|T)\mathbb{P}(T) + \mathbb{P}(S|T^c)\mathbb{P}(T^c)} = \frac{99\% \times \frac{1}{10000}}{99\% \times \frac{1}{10000} + 5\% \times \frac{9999}{10000}} \approx 1.9763 \times 10^{-3}.$$

■

**Example 17.** When coded messages are sent, sometimes errors in transmission may occur. In particular, Morse code uses 'dots' and 'dashes', which are known to occur in the proportion of 3 : 4. Suppose there is interference on the transmission line, for which there is a  $\frac{1}{8}$  probability that a dot is mistakenly received as a dash, and a  $\frac{1}{8}$  probability that a dash is mistakenly received as a dot. If we received a dot, how much can we be sure that a dot was sent?

**Proof.** Define the events:

- $S = \{\text{a dot is sent}\}$ ,  $S^c = \{\text{a dash is sent}\}$ .
- $T = \{\text{a dot is received}\}$ ,  $T^c = \{\text{a dash is received}\}$ .

It is given that  $\mathbb{P}(S) = \frac{3}{7}$ ,  $\mathbb{P}(S^c) = \frac{4}{7}$  and  $\mathbb{P}(T^c|S) = \mathbb{P}(T|S^c) = \frac{1}{8}$ , and we need to find  $\mathbb{P}(S|T)$ . Since  $\mathbb{P}(\cdot|S)$  defines a probability measure, we have

$$\mathbb{P}(T|S) = 1 - \mathbb{P}(T^c|S) = \frac{7}{8}.$$

Hence by Bayes' theorem, we have

$$\mathbb{P}(S|T) = \frac{\mathbb{P}(T|S)\mathbb{P}(S)}{\mathbb{P}(T|S)\mathbb{P}(S) + \mathbb{P}(T|S^c)\mathbb{P}(S^c)} = \frac{\frac{7}{8} \times \frac{3}{7}}{\frac{7}{8} \times \frac{3}{7} + \frac{1}{8} \times \frac{4}{7}} = \frac{21}{25}.$$

■

### 4.2.3 Two independent events

Let  $A, B \in \mathcal{B}$  be two events with  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . In general, knowing that  $B$  has occurred changes the chance that  $A$  occurs, that is  $\mathbb{P}(A|B) \neq \mathbb{P}(A)$ . There is an exception case:

**Definition 4.26.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space and  $A, B \in \mathcal{B}$  with  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . If  $\mathbb{P}(A|B) = \mathbb{P}(A)$ , we say that  $A$  is **independent** of  $B$ , or that  $A$  and  $B$  are **independent events**.

**Proposition 4.27.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space and  $A, B \in \mathcal{B}$  with  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . TFAE:

- (a)  $A$  and  $B$  are independent events,
- (b)  $\mathbb{P}(A|B) = \mathbb{P}(A)$ ,
- (c)  $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$ , and
- (d)  $\mathbb{P}(B|A) = \mathbb{P}(B)$ .

**Proof.**

$$\begin{aligned}
 A \text{ and } B \text{ are independent events} &\iff \mathbb{P}(A|B) = \mathbb{P}(A) \\
 &\iff \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \mathbb{P}(A) \\
 &\iff \mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B) \\
 &\iff \mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} = \mathbb{P}(B)
 \end{aligned}$$

■

In practice, (c) is more commonly used.

**Example 18.** Suppose two fair six-faced dice are randomly tossed. Define the events:

- $A = \{\text{the sum of the dice is 6}\}$
- $B = \{\text{the sum of the dice is 7}\}$
- $C = \{\text{the first die is 4}\}$

Are  $A$  and  $C$  independent? What about  $B$  and  $C$ ?

**Solution.** We evaluate that

$$A = \{(1, 5), (2, 3), (3, 3), (4, 2), (5, 1)\}$$

$$B = \{(1, 6), (2, 4), (3, 4), (4, 3), (5, 2), (6, 1)\}$$

$$C = \{(4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6)\}$$

and so  $\mathbb{P}(A) = \frac{5}{36}$ ,  $\mathbb{P}(B) = \mathbb{P}(C) = \frac{1}{6}$ , and  $\mathbb{P}(A \cap C) = \mathbb{P}(B \cap C) = \frac{1}{36}$ . Compute that

$$\mathbb{P}(A)\mathbb{P}(C) = \frac{5}{36} \times \frac{1}{6} = \frac{5}{216} \neq \mathbb{P}(A \cap C),$$

so  $A$  and  $C$  are dependent. On the other hand,

$$\mathbb{P}(B)\mathbb{P}(C) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36} = \mathbb{P}(B \cap C),$$

so  $B$  and  $C$  are independent.

■

Why are  $A$  and  $C$  dependent while  $B$  and  $C$  independent? The main idea is that if the first die is 6, then the sum is impossible to be 6, while no matter which outcome the first die is, the sum can still be 7. Thus, in the former situation, the first die matters, while in the latter situation, the first die does not matter. In other words, in the former situation, having information about the first die updates the probability, while in the latter case it does not. This example shall clarify the concept of independence and illustrate the importance of conditional probability in Bayesian interpretation of probability.

**Proposition 4.28.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space and  $A, B \in \mathcal{B}$  with  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . Suppose  $A$  and  $B$  are independent. Then

- (a)  $A^c$  and  $B$  are independent.
- (b)  $A$  and  $B^c$  are independent.
- (c)  $A^c$  and  $B^c$  are independent.

**Proof.** (a) and (b) are equivalent due to the reflexivity of independence, so we only prove (b) and (c) here.

(b) Since  $A$  and  $B$  are independent, we have  $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$ . Now note that  $A = (A \cap B) \sqcup (A \cap B^c)$ , so

$$\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap B^c) = \mathbb{P}(A)\mathbb{P}(B) + \mathbb{P}(A \cap B^c).$$

Thus  $\mathbb{P}(A \cap B^c) = \mathbb{P}(A)[1 - \mathbb{P}(B)] = \mathbb{P}(A)\mathbb{P}(B^c)$ , so  $A$  and  $B^c$  are independent.

(c) By (b),  $A$  and  $B^c$  are independent. Apply (a) to this fact implies  $A^c$  and  $B^c$  are independent. ■

**Example 19.** Let  $A, B$  be two events where  $\mathbb{P}(A), \mathbb{P}(B) > 0$ . If  $\mathbb{P}(A^c|B) = 5\mathbb{P}(A|B)$  and  $\mathbb{P}(A \cap B^c) = \mathbb{P}(A \cap B) + 0.45$ , are  $A$  and  $B$  independent?

**Solution.**  $\mathbb{P}(A^c|B) = 5\mathbb{P}(A|B)$  implies  $\frac{\mathbb{P}(A^c \cap B)}{\mathbb{P}(B)} = 5 \times \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$  or equivalently  $\mathbb{P}(A^c \cap B) = 5\mathbb{P}(A \cap B)$ . Recall  $B = (A \cap B) \sqcup (A^c \cap B)$ , so

$$\mathbb{P}(B) = \mathbb{P}(A \cap B) + \mathbb{P}(A^c \cap B) = 6\mathbb{P}(A \cap B).$$

Also, since  $A = (A \cap B) \sqcup (A \cap B^c)$ , we have

$$\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap B^c) = 2\mathbb{P}(A \cap B) + 0.45 = \frac{1}{3}\mathbb{P}(B) + 0.45$$

Write  $p = \mathbb{P}(B)$  and suppose  $A$  and  $B$  are independent. Then

$$\begin{aligned} \mathbb{P}(A \cap B) &= \mathbb{P}(A)\mathbb{P}(B) \\ \implies \frac{p}{6} &= p \left( \frac{p}{3} + 0.45 \right) \\ \implies p &= p(2p + 2.7) \\ \implies 2p + 2.7 &= 1 \\ \implies p &= -0.85 < 0 \end{aligned}$$

which is nonsense. So  $A$  and  $B$  are dependent. ■

#### 4.2.4 Three independent events

**Definition 4.29.** Let  $(\Omega, \mathcal{B}, \mathbb{P})$  be a probability space and  $A, B, C \in \mathcal{B}$  with  $\mathbb{P}(A), \mathbb{P}(B), \mathbb{P}(C) > 0$ . We say  $A, B, C$  are **mutually independent** if the following conditions are satisfied:

- (i) Triple independent:  $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C)$
- (ii) Pairwise independent:  $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$ ,  $\mathbb{P}(B \cap C) = \mathbb{P}(B)\mathbb{P}(C)$  and  $\mathbb{P}(C \cap A) = \mathbb{P}(C)\mathbb{P}(A)$ .

We can see these two conditions are necessary:

**Example 20.** Suppose two fair six-faced dice are randomly tossed. Define the events:

- $A = \{\text{the two dice have the same outcome}\}$
- $B = \{\text{the sum is between 7 and 10 inclusive}\}$
- $C = \{\text{the sum is 2, 7 or 8}\}$

It is easy to see that  $\mathbb{P}(A) = \frac{1}{6}$ ,  $\mathbb{P}(B) = \frac{1}{2}$  and  $\mathbb{P}(C) = \frac{1}{3}$ . Furthermore,

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}\{(4, 4)\} = \frac{1}{36} = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C),$$

so  $A, B$  and  $C$  are triple independent. However we can compute that

$$\mathbb{P}(B \cap C) = \mathbb{P}\{\text{the sum is 7 or 8}\} = \frac{11}{36} \neq \frac{1}{6} = \mathbb{P}(B)\mathbb{P}(C),$$

so  $B$  and  $C$  are not independent.

**Example 21.** Suppose we are randomly picking a number from 1, 2, 3, 4. Define the events:

- $A = \{1, 2\}$
- $B = \{1, 3\}$
- $C = \{1, 4\}$

We also suppose the outcomes are equally likely, so  $\mathbb{P}(A) = \mathbb{P}(B) = \mathbb{P}(C) = \frac{1}{2}$ . Furthermore,

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

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

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

That means  $A$ ,  $B$  and  $C$  are pairwise independent. However,

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}\{1\} = \frac{1}{4} \neq \frac{1}{8} = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C),$$

so  $A$ ,  $B$  and  $C$  are not triple independent.

## 4.2.5 Birthday paradox

The problem of interest in this subsection is as follows:

- Let  $p(n)$  be the probability that, in a class of  $n$  randomly chosen students, at least two will share the same birthday. What is the smallest value of  $n$  s.t.  $p(n) > 0.5$ ? For simplicity, we assume that
  - there are 365 possible birthdays, and
  - each person's birthday is equally likely to be any of these days, independent of the other people in the group, regardless of leap years, twins, selection bias, seasonal and weekly variations, etc.

It is convenient to consider  $\tilde{p}(n)$ , which is the probability that, in a class of  $n$  randomly chosen students, no two will share the same birthday.  $p(n)$  and  $\tilde{p}(n)$  are related by  $p(n) = 1 - \tilde{p}(n)$ , so it is equivalent to find the minimal  $n$  s.t.

$$\tilde{p}(n) < 0.5.$$

To formulate  $\tilde{p}(n)$ , we investigate the birthdays of the students one by one.

- The first student can have any birthday.
- The second student cannot have the same birthday as the first student, so s/he has  $365 - 1 = 364$  possible birthdays.
- The third student cannot have the same birthday as the previous students, so s/he has  $365 - 2 = 363$  possible birthdays.
- In the same fashion, the  $k^{\text{th}}$  student cannot have the same birthday as the previous students, so s/he has  $365 - (k - 1)$  possible birthdays.

Therefore we have

$$\tilde{p}(n) = \prod_{k=1}^n \frac{365 - (k - 1)}{365} = \prod_{k=0}^{n-1} \frac{365 - k}{365} = \prod_{k=0}^{n-1} \frac{365 - k}{365} = \prod_{k=0}^{n-1} \left(1 - \frac{k}{365}\right) = \prod_{k=1}^{n-1} \left(1 - \frac{k}{365}\right).$$

The argument below follows from Paul Halmos. Recall the exponential bound  $e^x \geq 1 + x \forall x \in \mathbb{R}$ , so we have

$$\tilde{p}(n) = \prod_{k=1}^{n-1} \left(1 - \frac{k}{365}\right) \leq \prod_{k=1}^{n-1} \exp\left\{-\frac{k}{365}\right\} = \exp\left\{-\sum_{k=1}^{n-1} \frac{k}{365}\right\} = \exp\left\{-\frac{n(n-1)}{730}\right\}$$

If  $n$  satisfies  $\exp\left\{-\frac{n(n-1)}{730}\right\} < 0.5$ , then it would also satisfy  $\tilde{p}(n) < 0.5$ . Hence we solve the former inequality as follows:

$$\exp\left\{-\frac{n(n-1)}{730}\right\} < 0.5 \implies n(n-1) > -730 \log(0.5) = 730 \log(2).$$

Now note that  $730 \log(2) \approx 505.997$ , which is barely below 506, the value of  $n^2 - n$  attained when  $n = 23$  (or  $n = -22$  which is rejected). Therefore

$$n \leq 23.$$

Note that this approach only gives us an upper bound; it leaves open the possibility that  $n$  is 22 or less could also work. However, using a scientific calculator, we can calculate that

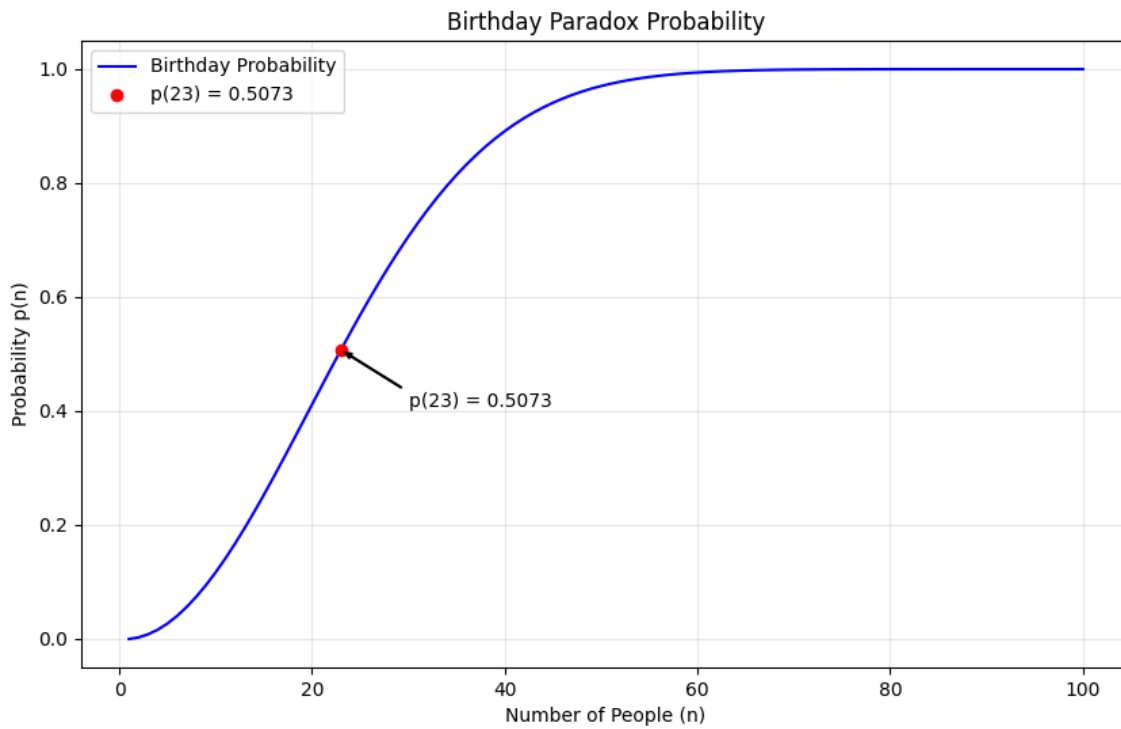
$$\tilde{p}(22) \approx 0.5243 > 0.5 \quad \text{and} \quad \tilde{p}(23) \approx 0.4927 < 0.5,$$

so the desired number is  $n = 23$ . It seems to be surprising at first glance, but it is true. This is an example of a veridical paradox.



### Vibe Coding Projects 3. >>> birthday\_probability.py

- Task: Plot a graph of  $p(n)$  against  $n$ .



#### 4.2.6 Two probability paradoxes

To further clarify the concepts of conditional probability and independence, we introduce two more famous probability paradoxes as follows. These paradoxes are marvellous concept sharpeners.

##### Example 22 (The boy or girl paradox).

- Mr. Jones has two children. The older child is a girl. What is the probability that both children are girls?
- Mr. Smith has two children. At least one of them is a boy. What is the probability that both children are boys?

In this problem, take for granted the following assumptions: Each child is either male or female, and has the same chance of being male as of being female; the sex of each child is independent of the sex of the other.

**Solution.** Let the sample space be  $\Omega = \{GG, GB, BG, BB\}$  where each element consists of two letters, the first letter is the gender of the older child, and the second letter is the gender of the younger child, in which  $G$  stands for a girl and  $B$  stands for a boy.

(a) Define the events:

- $A_1 = \{\text{the older child is a girl}\} = \{GG, GB\}$
- $A_2 = \{\text{both children are girls}\} = \{GG\}$

so the desired probability is

$$\mathbb{P}(A_2|A_1) = \frac{\mathbb{P}(A_1 \cap A_2)}{\mathbb{P}(A_1)} = \frac{1}{2}.$$

(b) Define the events:

- $B_1 = \{\text{at least one of the children is a boy}\} = \{GB, BG, BB\}$
- $B_2 = \{\text{both children are boys}\} = \{BB\}$

so the desired probability is

$$\mathbb{P}(B_2|B_1) = \frac{\mathbb{P}(B_1 \cap B_2)}{\mathbb{P}(B_1)} = \frac{1}{3}.$$

**Example 23 (Monty Hall Problem).** Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice? In this problem, the role of the host is as follows:

1. The host must always open a door that was not selected by the contestant.
2. The host must always open a door to reveal a goat and never the car. When there are two unchosen closed doors with a goat behind, the host opens one of them randomly.
3. The host must always offer the chance to switch between the chosen door and the remaining closed door.

**Solution.** Define the events:

- $A = \{\text{the car is behind door No. 2}\}$
- $B = \{\text{the host opens door No. 3}\}$

The desired probability is  $\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$ . We can compute it by drawing a table:

Case	What's behind...			The door the host opens	What if...		Probability of this case
	door 1	door 2	door 3		I stay	I switch	
1	Goat	Goat	<b>Car</b>	2	Win a goat	Win a goat	1/3
2	Goat	<b>Car</b>	Goat	3	Win a goat	<b>Win a car</b>	1/3
3a	<b>Car</b>	Goat	Goat	2	<b>Win a car</b>	Win a goat	1/6
3b				3	<b>Win a car</b>	Win a goat	1/6

Thus

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\frac{1}{3}}{\frac{1}{3} + \frac{1}{6}} = \frac{2}{3},$$

so it is more advantageous to switch the door. ■

**Vibe Coding Projects 4.** >>> `monty_hall.py`

- Task: Write a Monty Hall Problem simulator with two working modes: an interactive mode, where the user can interact with the program to feel how the game works; and a simulation mode, where I input the number of trials and the door I choose in the first step, then the program returns the probabilities of winning a car when my strategy is to stay or to switch.

## 4.3 Vibe Coding Project attempts

In the era of GenAI, vibe coding is growing very rapidly. In the following, we will make use of **Cursor**, a program designed specifically for vibe coding, to write some interesting code. So, what is vibe coding? This is a term coined by the computer scientist Andrej Karpathy, a co-founder of OpenAI and former AI leader at Tesla, which states the following:

*There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.*

Knowing the core philosophy of vibe coding, I begin my first hands-on experience. In the following, I will stick to the AI model **claude-3.5-sonnet** and the programming language **Python**. One thing to bear in mind is that I am using AI, so it is nearly impossible for AI to give you the same response as what AI gave me. Thus these examples only serve as a case study.

### 4.3.1 Vibe Coding Project 1: Dice simulator

- Task: Write a program to simulate rolling a fair die for  $k$  times, where  $k$  is our input, and then display the frequency of obtaining each outcome.

To do this task, I write the following prompt in **Cursor**:

**Prompt.** Can you write me a program in Python, where if I input a positive integer, the program simulates rolling a fair 6-sided die for the given number of times, and then the program should count the frequency of each number obtained afterwards; while if the input is not a positive integer, print an error message.

Then the AI gives me the following code:

```
1 import random
2
3 def simulate_dice_rolls(num_rolls):
4     # Check if input is a positive integer
5     if not isinstance(num_rolls, int) or num_rolls <= 0:
6         return "Error: Please enter a positive integer."
7
8     # Initialize a dictionary to store frequencies
9     frequencies = {i: 0 for i in range(1, 7)}
10
11     # Simulate dice rolls
12     for _ in range(num_rolls):
13         roll = random.randint(1, 6)
14         frequencies[roll] += 1
15
16     # Format and return results
17     result = f"\nResults of {num_rolls} dice rolls:\n"
18     for number, count in frequencies.items():
19         percentage = (count / num_rolls) * 100
20         result += f"Number {number}: {count} times ({percentage:.2f}%) \n"
21
```

```

22     return result
23
24 def main():
25     try:
26         # Get input from user
27         rolls = input("Enter the number of times to roll the dice: ")
28         rolls = int(rolls)
29
30         # Run simulation and print results
31         print(simulate_dice_rolls(rolls))
32
33     except ValueError:
34         print("Error: Please enter a valid integer.")
35
36 if __name__ == "__main__":
37     main()

```

Now I have the code of the program already. How can I run this code? A convenient way to run Python code is through Visual Studio Code with Python preinstalled. Open the .py file created by Cursor in Visual Studio Code, and then click 'Run' in the top right corner. A terminal appears below and says

```

1 PS D:\cursor projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/
  python3.11.exe "d:/cursor projects/dice_simulator.py"
2 Enter the number of times to roll the dice:

```

Enter the number of times to roll the dice and then press Enter. For example, I entered 100,000, and this is what the program returned:

```

1 PS D:\cursor projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/
  python3.11.exe "d:/cursor projects/dice_simulator.py"
2 Enter the number of times to roll the dice: 100000
3
4 Results of 100000 dice rolls:
5 Number 1: 16878 times (16.88%)
6 Number 2: 16777 times (16.78%)
7 Number 3: 16684 times (16.68%)
8 Number 4: 16392 times (16.39%)
9 Number 5: 16745 times (16.74%)
10 Number 6: 16524 times (16.52%)

```

And then I run the program for the second time. This is what I obtained this time:

```

1 PS D:\cursor projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/
  python3.11.exe "d:/cursor projects/dice_simulator.py"
2 Enter the number of times to roll the dice: 100000
3
4 Results of 100000 dice rolls:
5 Number 1: 16739 times (16.74%)
6 Number 2: 16614 times (16.61%)
7 Number 3: 16667 times (16.67%)
8 Number 4: 16797 times (16.80%)
9 Number 5: 16458 times (16.46%)
10 Number 6: 16725 times (16.73%)

```

The results are different. Thus, I can proudly conclude that the code satisfied our task.

### 4.3.2 Vibe Coding Project 2: Card drawing simulator

- Task: Task: Write a program to simulate example 12 by the frequentist approach: when I input the total number of trials, the program returns a graph where the  $x$ -axis is the number of trials  $n$ , and the  $y$ -axis is the ratio  $\frac{\text{number of occurrence of event in the } n \text{ trials}}{n}$ .

I convert the task scenario into a prompt:

**Prompt.** Can you write me a new Python program to simulate drawing cards from a standard deck, when I input an integer, you repeat simulation for the respective number of times, and then generate a graph with x-axis as the number of trials, and y-axis as the ratio of trials that the 14th card is the first card of 2 over the number of trials.

Then AI write me the following code:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from typing import List
4
5 def simulate_card_draw() -> bool:
6     """
7     Simulates drawing cards from a standard deck and checks if the first 2
8     appears at position 14.
9     Returns True if the condition is met, False otherwise.
10    """
11    # Create a deck with 4 of each value (1-13)
12    deck = np.repeat(np.arange(1, 14), 4)
13    # Shuffle the deck
14    np.random.shuffle(deck)
15    # Check if the first 2 appears at position 14 (index 13)
16    return np.where(deck == 2)[0][0] == 13
17
18 def run_simulations(num_trials: int) -> List[float]:
19    """
20    Runs multiple simulations and tracks the running ratio of successful trials.
21
22    Args:
23        num_trials: Number of trials to run
24
25    Returns:
26        List of running ratios for each trial number
27    """
28    successes = 0
29    ratios = []
30
31    for i in range(1, num_trials + 1):
32        if simulate_card_draw():
33            successes += 1
34            ratios.append(successes / i)
35
36    return ratios
37
38 def plot_results(ratios: List[float], num_trials: int):
39    """
40    Plots the results of the simulation.
```

```

40
41 Args:
42     ratios: List of running ratios
43     num_trials: Total number of trials
44     """
45     plt.figure(figsize=(10, 6))
46     plt.plot(range(1, num_trials + 1), ratios, 'b-', label='Running ratio')
47     plt.axhline(y=1/52, color='r', linestyle='--', label='Expected probability
(1/52)')
48     plt.xlabel('Number of Trials')
49     plt.ylabel('Ratio of Successful Trials')
50     plt.title('Card Drawing Simulation Results')
51     plt.legend()
52     plt.grid(True)
53     plt.show()
54
55 def main():
56     while True:
57         try:
58             num_trials = int(input("Enter the number of trials to run (or 0 to
exit): "))
59             if num_trials == 0:
60                 break
61             if num_trials < 1:
62                 print("Please enter a positive number.")
63                 continue
64
65             print(f"Running {num_trials} simulations...")
66             ratios = run_simulations(num_trials)
67             plot_results(ratios, num_trials)
68
69             # Print final probability
70             print(f"Final probability after {num_trials} trials: {ratios[-1]:.4f}
")
71
72             print(f"Expected probability: {1/52:.4f}")
73
74         except ValueError:
75             print("Please enter a valid integer.")
76         except KeyboardInterrupt:
77             print("\nExiting...")
78             break
79
80 if __name__ == "__main__":
81     main()

```

I try to run this program in Visual Studio Code, but then the following message pops out:

A virtual environment is not currently selected for your Python interpreter. Would you like to create a virtual environment?

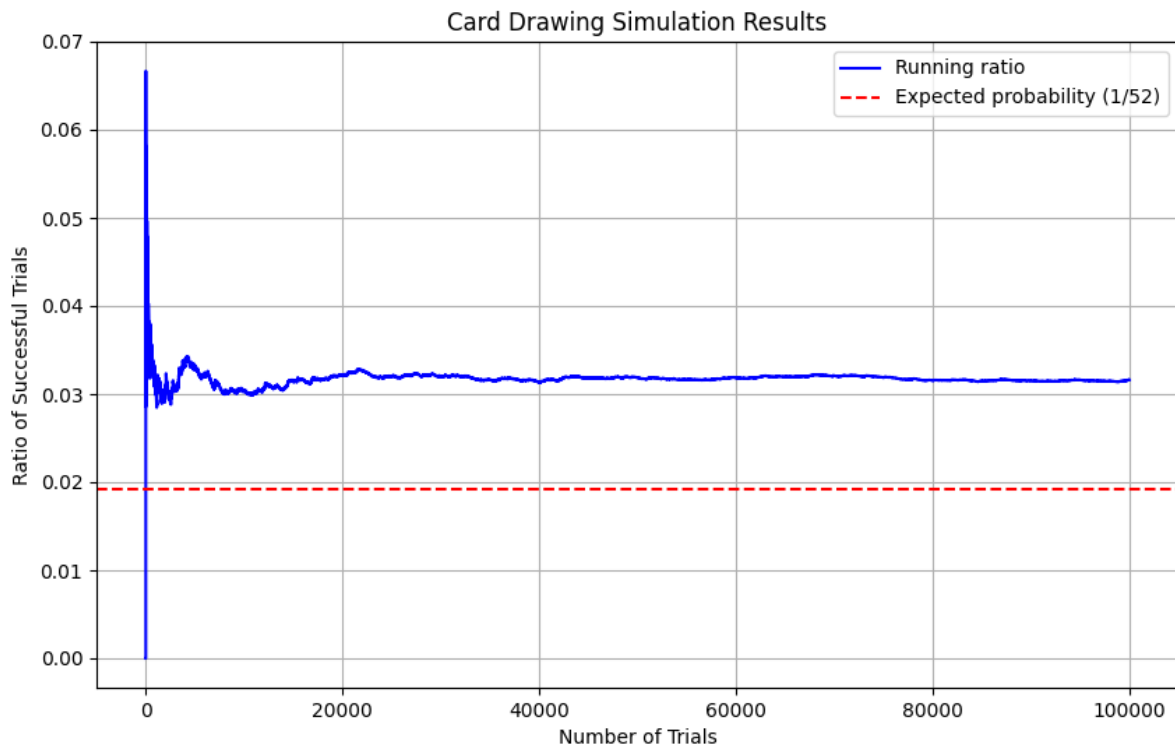
Press 'Create' and choose 'Venv', and then follow the instructions there. Upon completion, the following message pops out:

The following environment is selected: d:\cursor projects\.venv\Scripts\python.exe

Now I can run the program again. The following message appeared in the terminal:

```
1 PS D:\cursor projects> & "D:/cursor projects/.venv/Scripts/python.exe" "d:/cursor
  projects/prob/vce2/card_simulation.py"
2 Enter the number of trials to run (or 0 to exit):
```

Enter the number of trials and then press Enter. For example, I entered 100,000, and the program return me the following graph:



We have calculated that the probability is  $\frac{8436}{270725} \approx 0.0312$ , and the blue line is roughly there. But then I notice an unnecessary and incorrect red line of expected probability appears. I return to Cursor and ask AI to remove this line:

**Prompt.** I don't need the line of expected probability. Can you remove it for me?

And then AI generates a new code:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from typing import List
4
5 def simulate_card_draw() -> bool:
6     """
7     Simulates drawing cards from a standard deck and checks if the first 2
8     appears at position 14.
9     Returns True if the condition is met, False otherwise.
10    """
11    # Create a deck with 4 of each value (1-13)
12    deck = np.repeat(np.arange(1, 14), 4)
13    # Shuffle the deck
14    np.random.shuffle(deck)
15    # Check if the first 2 appears at position 14 (index 13)
16    return np.where(deck == 2)[0][0] == 13
17
18 def run_simulations(num_trials: int) -> List[float]:
```

```

18     """
19     Runs multiple simulations and tracks the running ratio of successful trials.
20
21     Args:
22         num_trials: Number of trials to run
23
24     Returns:
25         List of running ratios for each trial number
26     """
27     successes = 0
28     ratios = []
29
30     for i in range(1, num_trials + 1):
31         if simulate_card_draw():
32             successes += 1
33             ratios.append(successes / i)
34
35     return ratios
36
37 def plot_results(ratios: List[float], num_trials: int):
38     """
39     Plots the results of the simulation.
40
41     Args:
42         ratios: List of running ratios
43         num_trials: Total number of trials
44     """
45     plt.figure(figsize=(10, 6))
46     plt.plot(range(1, num_trials + 1), ratios, 'b-', label='Running ratio')
47     plt.xlabel('Number of Trials')
48     plt.ylabel('Ratio of Successful Trials')
49     plt.title('Card Drawing Simulation Results')
50     plt.legend()
51     plt.grid(True)
52     plt.show()
53
54 def main():
55     while True:
56         try:
57             num_trials = int(input("Enter the number of trials to run (or 0 to
58 exit): "))
59             if num_trials == 0:
60                 break
61             if num_trials < 1:
62                 print("Please enter a positive number.")
63                 continue
64
65             print(f"Running {num_trials} simulations...")
66             ratios = run_simulations(num_trials)
67             plot_results(ratios, num_trials)
68
69             # Print final probability
70             print(f"Final probability after {num_trials} trials: {ratios[-1]:.4f}")

```

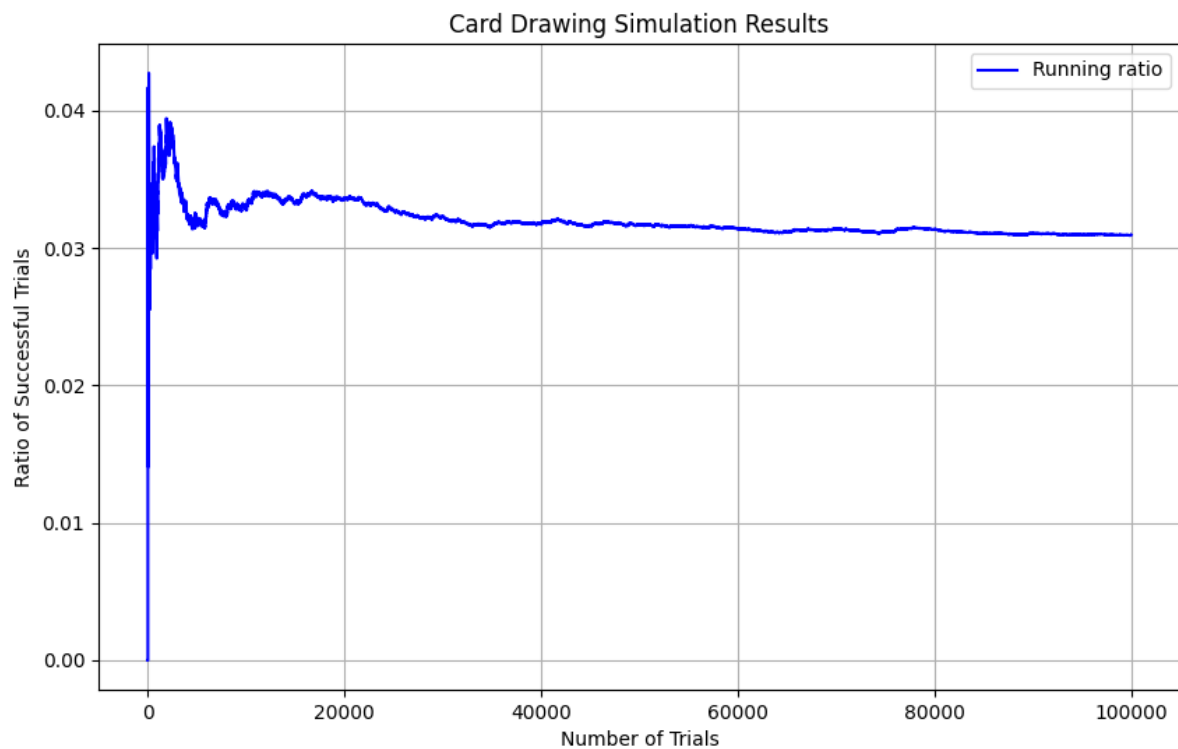


```

70     ")
71     except ValueError:
72         print("Please enter a valid integer.")
73     except KeyboardInterrupt:
74         print("\nExiting...")
75         break
76
77 if __name__ == "__main__":
78     main()

```

I run the program with input 100,000 again, and we yield the following new graph:



### 4.3.3 Vibe Coding Project 3: Visualizing birthday paradox

- Task: Plot a graph of  $p(n)$  against  $n$ .

I take advantage that AI knows how to read  $\text{\LaTeX}$  code and input the following prompt:

**Prompt.** Can you write me a python program to plot the function  $p(n)=1-\prod_{k=1}^{n-1} \left(1-\frac{k}{365}\right)$ , and display the value of  $p(23)$  in the graph?

And this is what AI gives me:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def birthday_probability(n):
5     if n <= 1:
6         return 0
7     probability = 1
8     for k in range(1, n):
9         probability *= (1 - k/365)
10    return 1 - probability
11
12 # Generate x values from 1 to 100
13 x = np.arange(1, 101)
14 # Calculate probabilities for each n
15 y = [birthday_probability(n) for n in x]
16
17 # Create the plot
18 plt.figure(figsize=(10, 6))
19 plt.plot(x, y, 'b-', label='Birthday Probability')
20
21 # Highlight p(23)
22 p_23 = birthday_probability(23)
23 plt.plot(23, p_23, 'ro', label=f'p(23) = {p_23:.4f}')
24
25 # Add labels and title
26 plt.xlabel('Number of People (n)')
27 plt.ylabel('Probability p(n)')
28 plt.title('Birthday Paradox Probability')
29 plt.grid(True, alpha=0.3)
30 plt.legend()
31
32 # Add text annotation for p(23) with an arrow
33 plt.annotate(f'p(23) = {p_23:.4f}',
34             xy=(23, p_23),
35             xytext=(30, p_23-0.1),
36             arrowprops=dict(facecolor='black',
37                             shrink=0.05,
38                             width=0.5,
39                             headwidth=3,
40                             headlength=4))
41
42 plt.show()
```

Now I run the program in Visual Studio Code, which gives us the graph in the text. Here I realize the AI is more powerful than I think - given the function  $p(n)$ , the AI realizes the function is related to the birthday paradox.

### 4.3.4 Vibe Coding Project 4: Monty Hall Problem simulator

- Task: Write a Monty Hall Problem simulator with two working modes: an interactive mode, where the user can interact with the program to feel how the game works; and a simulation mode, where I input the number of trials and the door I choose in the first step, then the program returns the probabilities of winning a car when my strategy is to stay or to switch.

After Vibe Coding Project 3, I realise the AI is more powerful than I think - given the function  $p(n)$ , the AI realizes the function is related to the birthday paradox. So without much thinking I directly copy the task above as the prompt:

**Prompt.** Write a Monty Hall Problem simulator with two working modes: an interactive mode, where the user can interact with the program to feel how the game works; and a simulation mode, where I input the number of trials and the door I choose in the first step, then the program returns the probabilities of winning a car when my strategy is to stay or to switch.

And this is what AI gives me:

```
1 import random
2 from typing import Tuple, List
3 import sys
4
5 def setup_game() -> List[str]:
6     """Set up the initial game state with one car and two goats."""
7     doors = ['goat', 'goat', 'car']
8     random.shuffle(doors)
9     return doors
10
11 def host_opens_door(doors: List[str], player_choice: int) -> int:
12     """Host opens a door with a goat, different from the player's choice."""
13     possible_doors = [i for i in range(3)
14                       if i != player_choice and doors[i] == 'goat']
15     return random.choice(possible_doors)
16
17 def play_interactive_game():
18     """Run an interactive version of the Monty Hall game."""
19     print("\nWelcome to the Monty Hall Game!")
20     print("Behind two doors are goats, and behind one door is a car.")
21
22     # Initial setup
23     doors = setup_game()
24
25     # Player's first choice
26     while True:
27         try:
28             choice = int(input("\nChoose a door (1, 2, or 3): ")) - 1
29             if 0 <= choice <= 2:
30                 break
31             print("Please enter 1, 2, or 3.")
32         except ValueError:
33             print("Please enter a valid number.")
34
35     # Host opens a door
36     opened_door = host_opens_door(doors, choice)
37     print(f"\nI'll open door {opened_door + 1}, which has a goat.")
38
39     # Player's final decision
```

```

40 while True:
41     switch = input("\nWould you like to switch doors? (yes/no): ").lower()
42     if switch in ['yes', 'no']:
43         break
44     print("Please answer 'yes' or 'no'.")
45
46 # Calculate final choice and result
47 if switch == 'yes':
48     final_choice = [i for i in range(3)
49                     if i != choice and i != opened_door][0]
50     print(f"\nYou switched to door {final_choice + 1}.")
51 else:
52     final_choice = choice
53     print("\nYou stayed with your original choice.")
54
55 # Reveal result
56 print(f"\nYou got a {doors[final_choice]}!")
57 print("\nFinal door contents:")
58 for i, item in enumerate(doors):
59     print(f"Door {i + 1}: {item}")
60
61 def run_simulation(trials: int, initial_choice: int) -> Tuple[float, float]:
62     """Run multiple trials and return win probabilities for stay/switch
63     strategies."""
64     stay_wins = 0
65     switch_wins = 0
66
67     for _ in range(trials):
68         doors = setup_game()
69         opened_door = host_opens_door(doors, initial_choice)
70         switch_choice = [i for i in range(3)
71                         if i != initial_choice and i != opened_door][0]
72
73         # Count wins for both strategies
74         if doors[initial_choice] == 'car':
75             stay_wins += 1
76         if doors[switch_choice] == 'car':
77             switch_wins += 1
78
79     return (stay_wins / trials, switch_wins / trials)
80
81 def main():
82     print("Welcome to the Monty Hall Problem Simulator!")
83     print("\nChoose a mode:")
84     print("1. Interactive Game")
85     print("2. Simulation Mode")
86
87     while True:
88         try:
89             mode = int(input("\nEnter mode (1 or 2): "))
90             if mode in [1, 2]:
91                 break
92             print("Please enter 1 or 2.")

```

```

92         except ValueError:
93             print("Please enter a valid number.")
94
95     if mode == 1:
96         play_interactive_game()
97     else:
98         # Get simulation parameters
99         while True:
100             try:
101                 trials = int(input("\nEnter number of trials: "))
102                 if trials > 0:
103                     break
104                 print("Please enter a positive number.")
105             except ValueError:
106                 print("Please enter a valid number.")
107
108         while True:
109             try:
110                 door = int(input("Enter your initial door choice (1, 2, or 3): "))
111                 door = door - 1
112                 if 0 <= door <= 2:
113                     break
114                 print("Please enter 1, 2, or 3.")
115             except ValueError:
116                 print("Please enter a valid number.")
117
118         # Run simulation and display results
119         stay_prob, switch_prob = run_simulation(trials, door)
120         print(f"\nResults after {trials} trials:")
121         print(f"Probability of winning if you stay: {stay_prob:.2%}")
122         print(f"Probability of winning if you switch: {switch_prob:.2%}")
123
124 if __name__ == "__main__":
125     main()

```

I am trying the simulator now. I first try the interactive mode:

```

1 PS D:\cursor projects> & "D:/cursor projects/.venv/Scripts/python.exe" "d:/cursor
  projects/monty_hall.py"
2 Welcome to the Monty Hall Problem Simulator!
3
4 Choose a mode:
5 1. Interactive Game
6 2. Simulation Mode
7
8 Enter mode (1 or 2): 1
9
10 Welcome to the Monty Hall Game!
11 Behind two doors are goats, and behind one door is a car.
12
13 Choose a door (1, 2, or 3): 1
14
15 I'll open door 3, which has a goat.
16
17 Would you like to switch doors? (yes/no): yes

```

```
18
19 You switched to door 2.
20
21 You got a goat!
22
23 Final door contents:
24 Door 1: car
25 Door 2: goat
26 Door 3: goat
```

At this stage I am worried about two things:

1. What if AI fools me - the 'car - goat - goat' case is the only case in the simulator?
2. What if I type something other than 'yes' or 'no' when the program asks me to switch the doors?

To eliminate my concerns, I run the program again with the same set of inputs, and this is the result:

```
1 ...
2
3 You got a car!
4
5 Final door contents:
6 Door 1: goat
7 Door 2: car
8 Door 3: goat
```

So the first concern is eliminated. Now we move to eliminate the second concern:

```
1 ...
2
3 Would you like to switch doors? (yes/no): nigga
4 Please answer 'yes' or 'no'.
5
6 Would you like to switch doors? (yes/no):
```

Cool, everything in the interactive mode works so smoothly. Now I move to try the simulator mode:

```
1 PS D:\cursor projects> & "D:/cursor projects/.venv/Scripts/python.exe" "d:/cursor
  projects/monty_hall.py"
2 Welcome to the Monty Hall Problem Simulator!
3
4 Choose a mode:
5 1. Interactive Game
6 2. Simulation Mode
7
8 Enter mode (1 or 2): 2
9
10 Enter number of trials: 100000
11 Enter your initial door choice (1, 2, or 3): 1
12
13 Results after 100000 trials:
14 Probability of winning if you stay: 33.26%
15 Probability of winning if you switch: 66.74%
```

A little takeaway after eliminating the concerns: I feel so stupid to check them every time, so I spend a very little amount of time to look at what AI generated to me.