**1. The Addition Rule of Probability**

The addition rule is used to find the probability of one event *or* another event occurring.

**a. Mutually Exclusive Events**

Two events are mutually exclusive if they cannot happen at the same time. The formula for the probability of A or B occurring is:

P(A∪B)=P(A)+P(B)**Error! Filename not specified.**

**Example:** What is the probability of rolling a 1 or a 6 on a single roll of a fair six-sided die?

* P(rolling a 1)=61​**Error! Filename not specified.**
* P(rolling a 6)=61​**Error! Filename not specified.**
* These are mutually exclusive events, so: P(1 or 6)=P(1)+P(6)=61​+61​=62​=31​**Error! Filename not specified.**

**b. Non-Mutually Exclusive Events**

Two events are not mutually exclusive if they can happen at the same time. The formula must account for the overlap to avoid double-counting.

P(A∪B)=P(A)+P(B)−P(A∩B)**Error! Filename not specified.**

**Example:** What is the probability of drawing a Queen or a Spade from a standard 52-card deck?

* P(Queen)=524​ (There are 4 Queens in the deck)
* P(Spade)=5213​ (There are 13 Spades in the deck)
* The events are not mutually exclusive because there is one Queen of Spades.
* P(Queen∩Spade)=521​**Error! Filename not specified.**
* P(Queen or Spade)=P(Queen)+P(Spade)−P(Queen∩Spade)=524​+5213​−521​=5216​=134​**Error! Filename not specified.**

/\*\*

\* Java program to demonstrate the Addition Rule of Probability.

\* This example calculates the probability of drawing a Queen or a Spade from a standard deck of cards.

\* This is a non-mutually exclusive event, as the Queen of Spades can satisfy both conditions.

\*/

public class AdditionRule {

public static void main(String[] args) {

// Total number of cards in a standard deck

double totalCards = 52.0;

// Number of Queens

double numQueens = 4.0;

// Number of Spades

double numSpades = 13.0;

// Number of cards that are both a Queen and a Spade (The Queen of Spades)

double numQueenAndSpade = 1.0;

// Calculate the individual probabilities

double probQueen = numQueens / totalCards;

double probSpade = numSpades / totalCards;

double probQueenAndSpade = numQueenAndSpade / totalCards;

// Apply the addition rule for non-mutually exclusive events

// P(A or B) = P(A) + P(B) - P(A and B)

double probQueenOrSpade = probQueen + probSpade - probQueenAndSpade;

System.out.println("Probability of drawing a Queen: " + probQueen);

System.out.println("Probability of drawing a Spade: " + probSpade);

System.out.println("Probability of drawing a Queen and a Spade: " + probQueenAndSpade);

System.out.println("----------------------------------------------");

System.out.println("Probability of drawing a Queen OR a Spade: " + probQueenOrSpade);

}

}

```

\*\*Code Output:\*\*

```

Probability of drawing a Queen: 0.07692307692307693

Probability of drawing a Spade: 0.25

Probability of drawing a Queen and a Spade: 0.019230769230769232

----------------------------------------------

Probability of drawing a Queen OR a Spade: 0.3076923076923077

**2. The Multiplication Rule of Probability**

The multiplication rule is used to find the probability of two or more events occurring in sequence.

**a. Independent Events**

Two events are independent if the occurrence of one does not affect the probability of the other. The formula is:

P(A∩B)=P(A)∗P(B)**Error! Filename not specified.**

**Example:** What is the probability of flipping a coin and getting heads, and then rolling a die and getting a 4?

* P(Heads)=21​**Error! Filename not specified.**
* P(Rolling a 4)=61​**Error! Filename not specified.**
* P(Heads and 4)=P(Heads)∗P(4)=21​∗61​=121​**Error! Filename not specified.**

**b. Dependent Events (Conditional Probability)**

Two events are dependent if the outcome of the first event affects the probability of the second. The formula uses conditional probability, P(B∣A), which is the probability of event B occurring given that event A has already occurred.

P(A∩B)=P(A)∗P(B∣A)**Error! Filename not specified.**

**Example:** A bag contains 3 red marbles and 7 blue marbles (10 total). What is the probability of drawing a red marble, then drawing a second red marble *without replacement*?

* P(1st red)=103​**Error! Filename not specified.**
* After the first red marble is drawn, there are 2 red marbles and 7 blue marbles left (9 total).
* P(2nd red∣1st red)=92​**Error! Filename not specified.**
* P(1st red and 2nd red)=P(1st red)∗P(2nd red∣1st red)=103​∗92​=906​=151​**Error! Filename not specified.**

\*\*

\* Java program to demonstrate the Multiplication Rule of Probability.

\* This example calculates the probability of drawing two red marbles in a row

\* from a bag without replacement. This is a dependent event.

\*/

public class MultiplicationRule {

public static void main(String[] args) {

// Initial number of marbles in the bag

double initialTotalMarbles = 10.0;

double initialRedMarbles = 3.0;

// Step 1: Probability of the first event (drawing a red marble)

double probFirstRed = initialRedMarbles / initialTotalMarbles;

// After drawing one red marble, the state of the bag changes.

// New number of total marbles and red marbles for the second draw.

double newTotalMarbles = initialTotalMarbles - 1.0;

double newRedMarbles = initialRedMarbles - 1.0;

// Step 2: Conditional probability of the second event (drawing a second red marble),

// given that the first was red.

double probSecondRedGivenFirstRed = newRedMarbles / newTotalMarbles;

// Apply the multiplication rule for dependent events

// P(A and B) = P(A) \* P(B|A)

double probBothRed = probFirstRed \* probSecondRedGivenFirstRed;

System.out.println("Initial Probability of drawing a red marble: " + probFirstRed);

System.out.println("Conditional Probability of drawing a second red marble: " + probSecondRedGivenFirstRed);

System.out.println("----------------------------------------------");

System.out.println("Probability of drawing two red marbles in a row: " + probBothRed);

}

}

```

\*\*Code Output:\*\*

```

Initial Probability of drawing a red marble: 0.3

Conditional Probability of drawing a second red marble: 0.2222222222222222

----------------------------------------------

Probability of drawing two red marbles in a row: 0.06666666666666667

**3. Bayes' Theorem**

Bayes' theorem is a powerful tool for calculating conditional probability. It describes how to update the probability of a hypothesis based on new evidence.

The formula is:

P(A∣B)=P(B)P(B∣A)∗P(A)​**Error! Filename not specified.**

Where:

* P(A∣B) is the posterior probability: the probability of A given B has occurred.
* P(B∣A) is the likelihood: the probability of B given A has occurred.
* P(A) is the prior probability: the initial probability of A.
* P(B) is the marginal probability: the total probability of B.

**Example:** A company's factory has two machines, A and B, which produce all the company's output. Machine A produces 60% of the output and Machine B produces 40%. 2% of the items from Machine A are defective, while 3% from Machine B are defective. An item is selected at random and found to be defective. What is the probability that it was produced by Machine A?

* P(A)=0.60 (Probability an item is from Machine A)
* P(B)=0.40 (Probability an item is from Machine B)
* P(Defective∣A)=0.02 (Probability an item is defective, given it's from Machine A)
* P(Defective∣B)=0.03 (Probability an item is defective, given it's from Machine B)

We need to find P(A∣Defective). First, we need to find P(Defective). The total probability of an item being defective is the sum of the probabilities of being defective from each machine:

P(Defective)=P(Defective∣A)P(A)+P(Defective∣B)P(B) P(Defective)=(0.02∗0.60)+(0.03∗0.40)=0.012+0.012=0.024**Error! Filename not specified.**

Now we can apply Bayes' Theorem:

P(A∣Defective)=P(Defective)P(Defective∣A)∗P(A)​=0.0240.02∗0.60​=0.0240.012​=0.5**Error! Filename not specified.**

So, there is a 50% probability that a defective item was produced by Machine A.

/\*\*

\* Java program to demonstrate Bayes' Theorem.

\* This example calculates the probability that a defective item was produced

\* by Machine A, given the item is defective.

\*/

public class BayesTheorem {

public static void main(String[] args) {

// Prior probabilities of an item being from each machine

double probMachineA = 0.60;

double probMachineB = 0.40;

// Likelihoods (conditional probabilities of being defective from each machine)

double probDefectiveGivenA = 0.02;

double probDefectiveGivenB = 0.03;

// Calculate the marginal probability of a random item being defective

// P(Defective) = P(Defective|A) \* P(A) + P(Defective|B) \* P(B)

double probDefective = (probDefectiveGivenA \* probMachineA) + (probDefectiveGivenB \* probMachineB);

// Apply Bayes' Theorem to find the posterior probability

// P(A|Defective) = [P(Defective|A) \* P(A)] / P(Defective)

double probAGivenDefective = (probDefectiveGivenA \* probMachineA) / probDefective;

System.out.println("Prior Probability of Machine A: " + probMachineA);

System.out.println("Prior Probability of Machine B: " + probMachineB);

System.out.println("Probability of defective item given it's from A: " + probDefectiveGivenA);

System.out.println("Probability of defective item given it's from B: " + probDefectiveGivenB);

System.out.println("----------------------------------------------");

System.out.println("Total Probability of an item being defective: " + probDefective);

System.out.println("Probability that a defective item came from Machine A: " + probAGivenDefective);

}

}

``

\*\*Code Output:\*\*

```

Prior Probability of Machine A: 0.6

Prior Probability of Machine B: 0.4

Probability of defective item given it's from A: 0.02

Probability of defective item given it's from B: 0.03

----------------------------------------------

Total Probability of an item being defective: 0.024

Probability that a defective item came from Machine A: 0.5