Quantitative Methods

Human Sciences, 2020–2021

Elias Nosrati

Lecture 1: 15 October 2020

General setup

General setup

- Statistics and data science in **R**.

General setup

- Statistics and data science in **R**.
- Building on first-year introductory statistics course.

General setup

- Statistics and data science in **R**.
- Building on first-year introductory statistics course.
- Assessment: one take-home assignment (Hilary) and one online open-book exam paper (Trinity).

General setup

- Statistics and data science in **R**.
- Building on first-year introductory statistics course.
- Assessment: one take-home assignment (Hilary) and one online open-book exam paper (Trinity).
- Weekly problem sheets to be submitted day before teaching.

General setup

- Statistics and data science in **R**.
- Building on first-year introductory statistics course.
- Assessment: one take-home assignment (Hilary) and one online open-book exam paper (Trinity).
- Weekly problem sheets to be submitted day before teaching.
- All materials available on Canvas and on eliasnosrati.github.io.

Syllabus

Syllabus

- Probability theory.

Syllabus

- Probability theory.
- Theories of statistical inference.

Syllabus

- Probability theory.
- Theories of statistical inference.
- Counterfactual inference (causality and prediction).

Syllabus

- Probability theory.
- Theories of statistical inference.
- Counterfactual inference (causality and prediction).
- Descriptive inference (data discovery and pattern recognition).

Today

Today

- Some mathematical preliminaries.

Today

- Some mathematical preliminaries.
- Probability theory.

Today

- Some mathematical preliminaries.
- Probability theory.
- Introduction to **R** (tutorial).

Why probability theory?

Why probability theory?

- Foundation of statistics.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.
- Physics: quantum mechanics, statistical mechanics.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.
- Physics: quantum mechanics, statistical mechanics.
- Computer science: randomised algorithms, artificial intelligence.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.
- Physics: quantum mechanics, statistical mechanics.
- Computer science: randomised algorithms, artificial intelligence.
- Meteorology: weather forecasting.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.
- Physics: quantum mechanics, statistical mechanics.
- Computer science: randomised algorithms, artificial intelligence.
- Meteorology: weather forecasting.
- Social and political science: policy impacts, elections.

Why probability theory?

- Foundation of statistics.
- Biology: inheritance of genes, mutations.
- Medicine: clinical trials, epidemiology.
- Physics: quantum mechanics, statistical mechanics.
- Computer science: randomised algorithms, artificial intelligence.
- Meteorology: weather forecasting.
- Social and political science: policy impacts, elections.
- And much more!

Set theory: definition of a set

Set theory: definition of a set

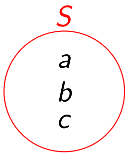
Definition

A *set* is a collection of objects.

Set theory: definition of a set

Definition

A *set* is a collection of objects.



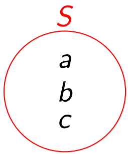
Example

- $S = \{a, b, c\}$ is a set composed of the first three letters of the English alphabet.

Set theory: definition of a set

Definition

A *set* is a collection of objects.



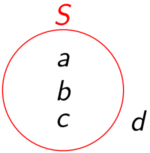
Example

- $S = \{a, b, c\}$ is a set composed of the first three letters of the English alphabet.
- $a \in S$: a is a member (or element) of S.

Set theory: definition of a set

Definition

A *set* is a collection of objects.



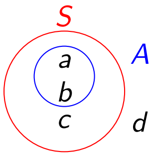
Example

- $S = \{a, b, c\}$ is a set composed of the first three letters of the English alphabet.
- $a \in S$: a is a member (or element) of S.
- $d \notin S$: d is not a member (or element) of S.

Set theory: definition of a set

Definition

A *set* is a collection of objects.



Example

- $S = \{a, b, c\}$ is a set composed of the first three letters of the English alphabet.
- $a \in S$: a is a member (or element) of S.
- $d \notin S$: d is not a member (or element) of S.
- If $A = \{a, b\}$, then $A \subset S$: A is a *subset* of S.

Set theory: definition of a subset

Definition

A set A is a *subset* of a set B, or $A \subset B$, if every member of A is also a member of B:

$$(x \in A) \Rightarrow (x \in B)$$

for all elements x.

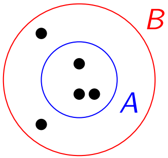
Set theory: definition of a subset

Definition

A set A is a *subset* of a set B, or $A \subset B$, if every member of A is also a member of B:

$$(x \in A) \Rightarrow (x \in B)$$

for all elements x.



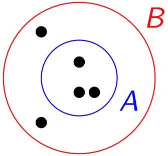
Set theory: definition of a subset

Definition

A set A is a *subset* of a set B, or $A \subset B$, if every member of A is also a member of B:

$$(x \in A) \Rightarrow (x \in B)$$

for all elements x.



Remark

If $A \subset B$ and $A \neq B$, then A is said to be a *proper subset* of B. Otherwise, we often write $A \subseteq B$.

Set theory: some important sets

Set theory: some important sets

- The set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

Set theory: some important sets

- The set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

- The set of integers

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}.$$

Set theory: some important sets

- The set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

- The set of integers

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}.$$

- The set of rational numbers

$$\mathbb{Q} = \{p/q : p, q \in \mathbb{Z} \text{ and } q \neq 0\}.$$

Set theory: some important sets

- The set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

- The set of integers

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}.$$

- The set of rational numbers

$$\mathbb{Q} = \{p/q : p, q \in \mathbb{Z} \text{ and } q \neq 0\}.$$

- The set of real numbers $\mathbb{R}$ (think of the number line).

Set theory: some more important sets

For $a, b \in \mathbb{R}$ and $a < b$, we define the following:

Set theory: some more important sets

For $a, b \in \mathbb{R}$ and $a < b$, we define the following:

- The open interval

$$(a, b) = \{x \in \mathbb{R} : a < x < b\}.$$

Set theory: some more important sets

For $a, b \in \mathbb{R}$ and $a < b$, we define the following:

- The open interval

$$(a, b) = \{x \in \mathbb{R} : a < x < b\}.$$

- The half open (closed) interval

$$(a, b] = \{x \in \mathbb{R} : a < x \leq b\}.$$

- The half closed (open) interval

$$[a, b) = \{x \in \mathbb{R} : a \leq x < b\}.$$

Set theory: some more important sets

For $a, b \in \mathbb{R}$ and $a < b$, we define the following:

- The open interval

$$(a, b) = \{x \in \mathbb{R} : a < x < b\}.$$

- The half open (closed) interval

$$(a, b] = \{x \in \mathbb{R} : a < x \leq b\}.$$

- The half closed (open) interval

$$[a, b) = \{x \in \mathbb{R} : a \leq x < b\}.$$

- The closed interval

$$[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}.$$

Set theory: empty set and power set

Definition

The *empty set*, denoted by $\varnothing = \{\}$, is a unique set with no members.

Set theory: empty set and power set

Definition

The *empty set*, denoted by $\varnothing = \{\}$, is a unique set with no members.

Definition

For a set A, the *power set* of A is the set of all subsets of A:

$$\mathcal{P}(A) = \{X : X \text{ is a subset of } A\},$$

Set theory: empty set and power set

Definition

The *empty set*, denoted by $\varnothing = \{\}$, is a unique set with no members.

Definition

For a set A, the *power set* of A is the set of all subsets of A:

$$\mathcal{P}(A) = \{X : X \text{ is a subset of } A\},$$

Example

Let $A = \{1, 2, 3\}$. Then

$$\mathcal{P}(A) = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \varnothing\}.$$

Set theory: empty set and power set

Definition

The *empty set*, denoted by $\varnothing = \{\}$, is a unique set with no members.

Definition

For a set A, the *power set* of A is the set of all subsets of A:

$$\mathcal{P}(A) = \{X : X \text{ is a subset of } A\},$$

Example

Let $A = \{1, 2, 3\}$. Then

$$\mathcal{P}(A) = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \varnothing\}.$$

Note that for any set S, $S \subseteq S$ and $\varnothing \subseteq S$. Hence $S \in \mathcal{P}(S)$ and $\varnothing \in \mathcal{P}(S)$ for all S. Note also for any set with n elements, its power set has 2^n elements (why?).

Set theory: union and intersection

Set theory: union and intersection

Definition

For two sets A and B, their *union* is defined as the set of elements contained in either A or B (or both):

$$A \cup B := \{x : x \in A \text{ or } x \in B\}.$$

Set theory: union and intersection

Definition

For two sets A and B, their *union* is defined as the set of elements contained in either A or B (or both):

$$A \cup B := \{x : x \in A \text{ or } x \in B\}.$$

Definition

The *intersection* of two sets A and B is defined as the set of elements A and B have in common:

$$A \cap B = \{x : x \in A \text{ and } x \in B\}.$$

Set theory: union and intersection

Definition

For two sets A and B, their *union* is defined as the set of elements contained in either A or B (or both):

$$A \cup B := \{x : x \in A \text{ or } x \in B\}.$$

Definition

The *intersection* of two sets A and B is defined as the set of elements A and B have in common:

$$A \cap B = \{x : x \in A \text{ and } x \in B\}.$$

Remark

We say that A and B are *disjoint* (mutually exclusive) is they have no elements in common, i.e., if $A \cap B = \varnothing$.

Set theory: examples of union and intersection

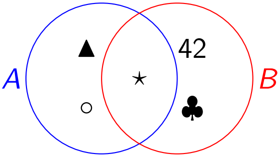
Example

Let $A = \{\star, \circ, \blacktriangle\}$ and $B = \{42, \clubsuit, \star\}$.

Set theory: examples of union and intersection

Example

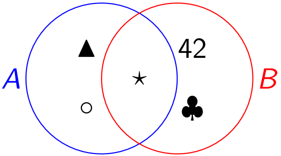
Let $A = \{\star, \circ, \blacktriangle\}$ and $B = \{42, \clubsuit, \star\}$.



Set theory: examples of union and intersection

Example

Let $A = \{\star, \circ, \blacktriangle\}$ and $B = \{42, \clubsuit, \star\}$.

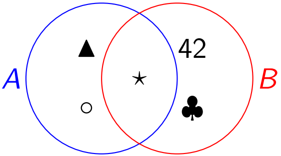


Then $A \cup B = \{\star, \circ, \blacktriangle, 42, \clubsuit\}$ and $A \cap B = \{\star\}$.

Set theory: examples of union and intersection

Example

Let $A = \{\star, \circ, \blacktriangle\}$ and $B = \{42, \clubsuit, \star\}$.



Then $A \cup B = \{\star, \circ, \blacktriangle, 42, \clubsuit\}$ and $A \cap B = \{\star\}$.

Example

Let $A = \{\text{Denmark, Norway, Sweden}\}$ and
$B = \{\text{Botswana, Namibia, Zimbabwe}\}$. Then $A \cap B = \{\} = \varnothing$.

Set theory: complement and set difference

Definition

The *complement* of a set A, denoted by A^c, is the set of elements that are not members of A:

$$A^{\mathsf{c}} = \{x : x \notin A\}.$$

Set theory: complement and set difference

Definition

The *complement* of a set A, denoted by A^c, is the set of elements that are not members of A:

$$A^{\mathrm{c}} = \{x : x \notin A\}.$$

Definition

The *set difference* $A - B$ denotes the complement of a set B relative to A:

$$A - B = \{x \in A : x \notin B\}.$$

Set theory: complement and set difference

Definition

The *complement* of a set A, denoted by A^c, is the set of elements that are not members of A:

$$A^{\mathrm{c}} = \{x : x \notin A\}.$$

Definition

The *set difference* $A - B$ denotes the complement of a set B relative to A:

$$A - B = \{x \in A : x \notin B\}.$$

Example

Let A be the set of all individuals named Henry and let B be the set of all individuals with brown hair. Then A^c is the set of all people whose name is not Henry and $A - B$ is the set of all people named Henry who do not have brown hair.

Sample space

What does all this have to do with probability?

Sample space

What does all this have to do with probability?

An experiment is performed, the outcome of which is uncertain. We can mathematically express all possible outcomes of the experiment as a set:

Sample space

What does all this have to do with probability?

An experiment is performed, the outcome of which is uncertain. We can mathematically express all possible outcomes of the experiment as a set:

Definition

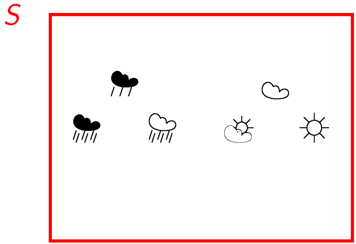
The *sample space* S of an experiment is the set of all possible outcomes of the experiment. An *event* A is a subset of this sample space, and if the actual outcome is an element of A, we say that A *occurred*.

Sample space: weather example

Let S be the space of possible weather outcomes, and let A denote the event that it rains.

Sample space: weather example

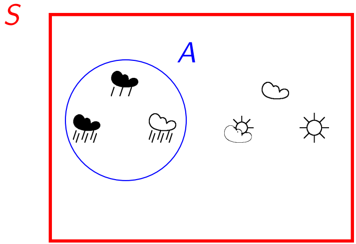
Let S be the space of possible weather outcomes, and let A denote the event that it rains.



According to this definition of the event A, we do not distinguish between a light drizzle or torrential rain: either A happens or it doesn't. Performing an experiment amounts to randomly selecting one outcome.

Sample space: weather example

Let S be the space of possible weather outcomes, and let A denote the event that it rains.



According to this definition of the event A, we do not distinguish between a light drizzle or torrential rain: either A happens or it doesn't. Performing an experiment amounts to randomly selecting one outcome.

Sample space: coin toss example

A coin is flipped ten times.

Sample space: coin toss example

A coin is flipped ten times. We encode the outcome 'Heads' as H and the outcome 'Tails' as T.

Sample space: coin toss example

A coin is flipped ten times. We encode the outcome 'Heads' as H and the outcome 'Tails' as T. Then the sample space is the set of all possible strings of H's and T's of length 10:

$$S = \{(s_1, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 1 \leq i \leq 10\}.$$

Sample space: coin toss example

A coin is flipped ten times. We encode the outcome 'Heads' as H and the outcome 'Tails' as T. Then the sample space is the set of all possible strings of H's and T's of length 10:

$$S = \{(s_1, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 1 \leq i \leq 10\}.$$

Now let $A_1 \subset S$ be the event that the outcome of the first flip of the coin is T, i.e., $s_1 = T$:

$$A_1 = \{(T, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 2 \leq i \leq 10\}.$$

Sample space: coin toss example

A coin is flipped ten times. We encode the outcome ‘Heads’ as H and the outcome ‘Tails’ as T. Then the sample space is the set of all possible strings of H's and T's of length 10:

$$S = \{(s_1, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 1 \leq i \leq 10\}.$$

Now let $A_1 \subset S$ be the event that the outcome of the first flip of the coin is T, i.e., $s_1 = T$:

$$A_1 = \{(T, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 2 \leq i \leq 10\}.$$

Quiz: Let A_i be the event that the ith flip is T and let B be the event that we obtain T at least once. How is B denoted in set-theoretical notation?

Sample space: coin toss example

A coin is flipped ten times. We encode the outcome 'Heads' as H and the outcome 'Tails' as T. Then the sample space is the set of all possible strings of H's and T's of length 10:

$$S = \{(s_1, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 1 \leq i \leq 10\}.$$

Now let $A_1 \subset S$ be the event that the outcome of the first flip of the coin is T, i.e., $s_1 = T$:

$$A_1 = \{(T, \ldots, s_{10}) : s_i \in \{H, T\} \text{ for } 2 \leq i \leq 10\}.$$

Quiz: Let A_i be the event that the ith flip is T and let B be the event that we obtain T at least once. How is B denoted in set-theoretical notation?

Answer:

$$B = A_1 \cup \cdots \cup A_{10} = \bigcup_{i=1}^{10} A_i.$$

Sample space: coin toss example (cont.)

Another quiz: Let D be the event that we obtain at least two consecutive T's. How can we express this in set-theoretical terms?

Sample space: coin toss example (cont.)

Another quiz: Let D be the event that we obtain at least two consecutive T's. How can we express this in set-theoretical terms?

Answer:

$$D = (A_1 \cap A_2) \cup \cdots \cup (A_9 \cap A_{10}) = \bigcup_{i=1}^{9} (A_i \cap A_{i+1}).$$

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards.

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards. Let A denote the event that your card is an Ace:

$A = \{$Ace of Clubs, Ace of Diamonds, Ace of Hearts, Ace of Spades$\}$.

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards. Let A denote the event that your card is an Ace:

$A = \{$Ace of Clubs, Ace of Diamonds, Ace of Hearts, Ace of Spades$\}$.

Let H be the event that you card is a Heart and let B be the event that the card has a black suit.

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards. Let A denote the event that your card is an Ace:

$A = \{$Ace of Clubs, Ace of Diamonds, Ace of Hearts, Ace of Spades$\}$.

Let H be the event that you card is a Heart and let B be the event that the card has a black suit.

What event does $A \cap H$ denote?

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards. Let A denote the event that your card is an Ace:

$A = \{$Ace of Clubs, Ace of Diamonds, Ace of Hearts, Ace of Spades$\}$.

Let H be the event that you card is a Heart and let B be the event that the card has a black suit.

What event does $A \cap H$ denote?

What event does $(A \cup B)^c$ denote?

Sample space: card deck example

Pick a card from a standard deck of 52 cards. The sample space is the set of all 52 cards. Let A denote the event that your card is an Ace:

$A = \{$Ace of Clubs, Ace of Diamonds, Ace of Hearts, Ace of Spades$\}$.

Let H be the event that you card is a Heart and let B be the event that the card has a black suit.

What event does $A \cap H$ denote?

What event does $(A \cup B)^c$ denote?

Remark

Note that $(A \cup B)^c = A^c \cap B^c$. This is one of *De Morgan's laws*. The other is $(A \cap B)^c = A^c \cup B^c$.

Translating between English and set theory

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \dots A_n$ occurs $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \dots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \dots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \dots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \dots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \dots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \dots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \dots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \dots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$ $A \subseteq B$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$ $A \subseteq B$.
- A and B are mutually exclusive $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$ $A \subseteq B$.
- A and B are mutually exclusive $\leftrightsquigarrow$ $A \cap B = \varnothing$.

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$ $A \subseteq B$.
- A and B are mutually exclusive $\leftrightsquigarrow$ $A \cap B = \varnothing$.
- $A_1, \ldots A_n$ are a partition of S $\leftrightsquigarrow$

Translating between English and set theory

- Sample space $\leftrightsquigarrow$ S.
- s is a possible outcome $\leftrightsquigarrow$ $s \in S$.
- A is an event $\leftrightsquigarrow$ $A \subseteq S$.
- At least one of A and B occurs $\leftrightsquigarrow$ $A \cup B$.
- At least one of $A_1, \ldots A_n$ occurs $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n$.
- Both A and B occur $\leftrightsquigarrow$ $A \cap B$.
- All of $A_1, \ldots A_n$ occur $\leftrightsquigarrow$ $A_1 \cap \cdots \cap A_n$.
- A does not occur $\leftrightsquigarrow$ A^c.
- A or B, but not both $\leftrightsquigarrow$ $(A \cap B^c) \cup (A^c \cap B)$.
- A implies B $\leftrightsquigarrow$ $A \subseteq B$.
- A and B are mutually exclusive $\leftrightsquigarrow$ $A \cap B = \varnothing$.
- $A_1, \ldots A_n$ are a partition of S $\leftrightsquigarrow$ $A_1 \cup \cdots \cup A_n = S$, $A_i \cap A_j = \varnothing$ for $i \neq j$.

Functions

Functions

Definition

Let X and Y be sets. A *function* f from X to Y is a mapping from the *domain* X to the *codomain* Y that assigns a value $f(x) \in Y$ to each $x \in X$:

$$f : X \to Y \ \text{ such that } \ x \mapsto y = f(x).$$

Functions

Definition

Let X and Y be sets. A *function* f from X to Y is a mapping from the *domain* X to the *codomain* Y that assigns a value $f(x) \in Y$ to each $x \in X$:

$$f : X \to Y \;\text{ such that }\; x \mapsto y = f(x).$$

The definition of a function encompasses assignment, domain, and codomain.

Functions

Definition

Let X and Y be sets. A *function* f from X to Y is a mapping from the *domain* X to the *codomain* Y that assigns a value $f(x) \in Y$ to each $x \in X$:

$$f : X \to Y \text{ such that } x \mapsto y = f(x).$$

The definition of a function encompasses assignment, domain, and codomain. As such, the function

$$f_1 : \mathbb{R} \to \mathbb{R} \text{ given by } f_1(x) = x^2$$

is *not* the same function as

$$f_2 : [0, \infty) \to \mathbb{R} \text{ given by } f_2(x) = x^2.$$

Functions

Definition

Let X and Y be sets. A *function* f from X to Y is a mapping from the *domain* X to the *codomain* Y that assigns a value $f(x) \in Y$ to each $x \in X$:

$$f : X \to Y \text{ such that } x \mapsto y = f(x).$$

The definition of a function encompasses assignment, domain, and codomain. As such, the function

$$f_1 : \mathbb{R} \to \mathbb{R} \text{ given by } f_1(x) = x^2$$

is *not* the same function as

$$f_2 : [0, \infty) \to \mathbb{R} \text{ given by } f_2(x) = x^2.$$

(Why not?)

Probability: a definition

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Example

We flip a fair coin once and observe the outcome.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Example

We flip a fair coin once and observe the outcome. As before, we encode Heads as H and Tails as T.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Example

We flip a fair coin once and observe the outcome. As before, we encode Heads as H and Tails as T. $S = \{H, T\}$.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Example

We flip a fair coin once and observe the outcome. As before, we encode Heads as H and Tails as T. $S = \{H, T\}$. Let $A = \{H\} \subset S$ be the event of obtaining Heads.

Probability: a definition

A probability is a way of assigning a measure of uncertainty to each possible outcome of an experiment.

Definition

A *probability space* consists of a sample space S, a class of events defined on S, and a *probability function*

$$\mathbb{P} : S \to [0, 1]$$

which takes an event $A \subseteq S$ as input and returns $\mathbb{P}(A)$, a real number between 0 and 1, as output.

Example

We flip a fair coin once and observe the outcome. As before, we encode Heads as H and Tails as T. $S = \{H, T\}$. Let $A = \{H\} \subset S$ be the event of obtaining Heads. Then $\mathbb{P}(A) = 0.5$.

The three axioms of probability

The three axioms of probability

1. $\mathbb{P}(A) \geq 0$ for any event A.

The three axioms of probability

1. $\mathbb{P}(A) \geq 0$ for any event A.
2. $\mathbb{P}(\varnothing) = 0$ and $\mathbb{P}(S) = 1$ (something happens!).

The three axioms of probability

1. $\mathbb{P}(A) \geq 0$ for any event A.
2. $\mathbb{P}(\varnothing) = 0$ and $\mathbb{P}(S) = 1$ (something happens!).
3. If $A_1, \ldots, A_n$ are disjoint events, then

$$\mathbb{P}(A_1 \cup \cdots \cup A_n) = \mathbb{P}(A_1) + \cdots + \mathbb{P}(A_n).$$

The three axioms of probability

1. $\mathbb{P}(A) \geq 0$ for any event A.
2. $\mathbb{P}(\varnothing) = 0$ and $\mathbb{P}(S) = 1$ (something happens!).
3. If $A_1, \dots, A_n$ are disjoint events, then

   $$\mathbb{P}(A_1 \cup \cdots \cup A_n) = \mathbb{P}(A_1) + \cdots + \mathbb{P}(A_n).$$

   Equivalently:

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

Properties of probability

Properties of probability

- $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.

Properties of probability

- $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
- If $A \subseteq B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$.

Properties of probability

- $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
- If $A \subseteq B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$.
- $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.

Properties of probability

- $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$.
- If $A \subseteq B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$.
- $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.

Exercise: Can your prove these properties (using only the three axioms of probability)?

The Law of Total Probability

The Law of Total Probability

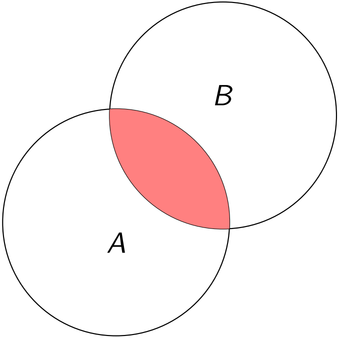
For any two events A and B, the *Law of Total Probability* states that

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

The Law of Total Probability

For any two events A and B, the *Law of Total Probability* states that

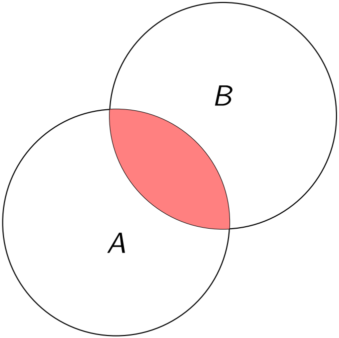
$$\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap B^c).$$

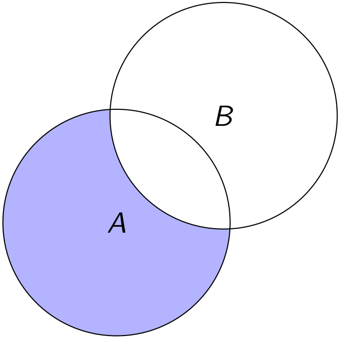


The Law of Total Probability

For any two events A and B, the *Law of Total Probability* states that

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





Counting

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6.

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } 1 \leq i \leq 3\}.$$

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } 1 \leq i \leq 3\}.$$

Let $A \subset S$ be the event that $s_2 = 4$:

$$A = \{(s_1, 4, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } i \in \{1, 3\}\}.$$

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1,2,3,4,5,6\} \text{ for } 1 \leq i \leq 3\}.$$

Let $A \subset S$ be the event that $s_2 = 4$:

$$A = \{(s_1, 4, s_3) : s_i \in \{1,2,3,4,5,6\} \text{ for } i \in \{1,3\}\}.$$

Then

$\mathbb{P}(A) =$

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } 1 \leq i \leq 3\}.$$

Let $A \subset S$ be the event that $s_2 = 4$:

$$A = \{(s_1, 4, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } i \in \{1, 3\}\}.$$

Then

$$\mathbb{P}(A) = \frac{\#\text{ of outcomes in which } s_2 = 4}{\text{total }\#\text{ of outcomes in } S}$$

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } 1 \leq i \leq 3\}.$$

Let $A \subset S$ be the event that $s_2 = 4$:

$$A = \{(s_1, 4, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } i \in \{1, 3\}\}.$$

Then

$$\mathbb{P}(A) = \frac{\#\text{ of outcomes in which } s_2 = 4}{\text{total } \#\text{ of outcomes in } S} = \frac{6 \times 1 \times 6}{6 \times 6 \times 6}$$

Counting

Many probability problems require us to count all the possible outcomes of an experiment.

Example

We throw a fair die three times and record the resulting ordered triple of numbers between 1 and 6. We have

$$S = \{(s_1, s_2, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } 1 \leq i \leq 3\}.$$

Let $A \subset S$ be the event that $s_2 = 4$:

$$A = \{(s_1, 4, s_3) : s_i \in \{1, 2, 3, 4, 5, 6\} \text{ for } i \in \{1, 3\}\}.$$

Then

$$\mathbb{P}(A) = \frac{\#\text{ of outcomes in which } s_2 = 4}{\text{total }\#\text{ of outcomes in } S} = \frac{6 \times 1 \times 6}{6 \times 6 \times 6} = \frac{36}{216} \approx 17\%.$$

Permutations

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Example

Imagine a jar with ten marbles, labelled from 1 to 10.

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Example

Imagine a jar with ten marbles, labelled from 1 to 10. We sample a marble one at a time, record the label, return the marble to the jar, and repeat the process five times.

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Example

Imagine a jar with ten marbles, labelled from 1 to 10. We sample a marble one at a time, record the label, return the marble to the jar, and repeat the process five times. Each sampled marble is a sub-experiment with $n = 10$ possible outcomes, and there are $k = 5$ sub-experiments.

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Example

Imagine a jar with ten marbles, labelled from 1 to 10. We sample a marble one at a time, record the label, return the marble to the jar, and repeat the process five times. Each sampled marble is a sub-experiment with $n = 10$ possible outcomes, and there are $k = 5$ sub-experiments. The number of ways to obtain a sample of size 5 is $P_k^n = 10!/(10-5)! = 30240$.

Permutations

Definition

The *permutation* of n objects is an arrangement of these objects in a specific order. The number of ways of permuting k objects out of n unique objects is given by

$$P_k^n = n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}.$$

(Recall that $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$.)

Example

Imagine a jar with ten marbles, labelled from 1 to 10. We sample a marble one at a time, record the label, return the marble to the jar, and repeat the process five times. Each sampled marble is a sub-experiment with $n = 10$ possible outcomes, and there are $k = 5$ sub-experiments. The number of ways to obtain a sample of size 5 is $P_k^n = 10!/(10-5)! = 30240$. This is an example of *sampling with replacement*.

Permutations: queuing example

How many ways are there in which n people can queue to buy ice cream?

Permutations: queuing example

How many ways are there in which n people can queue to buy ice cream?

Each queue is a permuation of n unique elements (people) and hence $n = k$ and $P_k^n = n!$.

Permutations: queuing example

How many ways are there in which n people can queue to buy ice cream?

Each queue is a permuation of n unique elements (people) and hence $n = k$ and $P_k^n = n!$.

This is an example of *sampling without replacement*.

Combinations: the binomial coefficient

Combinations: the binomial coefficient

Unlike a permutation, a combination is a way of choosing k out of n objects *without* regard to order. Formally, this is known as a binomial coefficient:

Combinations: the binomial coefficient

Unlike a permutation, a combination is a way of choosing k out of n objects *without* regard to order. Formally, this is known as a binomial coefficient:

Definition

A *binomial coefficient* counts the number of subsets of size k for a set of size n.

Combinations: the binomial coefficient

Unlike a permutation, a combination is a way of choosing k out of n objects *without* regard to order. Formally, this is known as a binomial coefficient:

Definition

A *binomial coefficient* counts the number of subsets of size k for a set of size n. For any non-negative integers k and n, the binomial coefficient, read as "n choose k", is defined as

$$C_k^n = \binom{n}{k} = \frac{P_k^n}{k!} = \frac{n!}{k!(n-k)!}.$$

Combinations: the binomial coefficient

Unlike a permutation, a combination is a way of choosing k out of n objects *without* regard to order. Formally, this is known as a binomial coefficient:

Definition

A *binomial coefficient* counts the number of subsets of size k for a set of size n. For any non-negative integers k and n, the binomial coefficient, read as "n choose k", is defined as

$$C_k^n = \binom{n}{k} = \frac{P_k^n}{k!} = \frac{n!}{k!(n-k)!}.$$

Since sets are unordered (i.e., $\{x, y\} = \{y, x\}$), this amounts to choosing k out of n objects, without replacement and without distinguishing the order in which they are chosen.

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

$$\binom{4}{2} = 6.$$

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

$$\binom{4}{2} = 6.$$

How many ways are there to break the group into two teams of two?

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

$$\binom{4}{2} = 6.$$

How many ways are there to break the group into two teams of two?

$$\binom{4}{2} = 6?$$

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

$$\binom{4}{2} = 6.$$

How many ways are there to break the group into two teams of two?

$$\binom{4}{2} = 6?$$

No, because this overcounts by a factor of 2 (picking A and B to be a team is equivalent to picking C and D to be a team)!

Combinations: committee example

Consider a group of four people. How many ways are there to choose a two-person committee?

$$\binom{4}{2} = 6.$$

How many ways are there to break the group into two teams of two?

$$\binom{4}{2} = 6?$$

No, because this overcounts by a factor of 2 (picking A and B to be a team is equivalent to picking C and D to be a team)!

$$\frac{\binom{4}{2}}{2} = 3.$$

Some important identities (with story proofs)

- Choosing the complement:

$$\binom{n}{k} = \binom{n}{n-k}.$$

Some important identities (with story proofs)

- Choosing the complement:

$$\binom{n}{k} = \binom{n}{n-k}.$$

- Choosing a team captain:

$$n\binom{n-1}{k-1} = \binom{n}{k}k.$$

Some important identities (with story proofs)

- Choosing the complement:

$$\binom{n}{k} = \binom{n}{n-k}.$$

- Choosing a team captain:

$$n\binom{n-1}{k-1} = \binom{n}{k}k.$$

- Choosing junior and senior committee members:

$$\binom{m+n}{k} = \sum_{j=0}^{k}\binom{m}{j}\binom{n}{k-j}.$$

Some important identities (with story proofs)

- Choosing the complement:

$$\binom{n}{k} = \binom{n}{n-k}.$$

- Choosing a team captain:

$$n\binom{n-1}{k-1} = \binom{n}{k}k.$$

- Choosing junior and senior committee members:

$$\binom{m+n}{k} = \sum_{j=0}^{k} \binom{m}{j}\binom{n}{k-j}.$$

(*Vandermonde's identity*.)

Introduction to **R**

- **R** is an open-source statistical programming environment.

Introduction to **R**

- **R** is an open-source statistical programming environment.
- Freely available from the Comprehensive **R** Archive Network (CRAN).

Introduction to **R**

- **R** is an open-source statistical programming environment.
- Freely available from the Comprehensive **R** Archive Network (CRAN).
- A powerful tool for statistics and data analysis.

Introduction to **R**

- **R** is an open-source statistical programming environment.
- Freely available from the Comprehensive **R** Archive Network (CRAN).
- A powerful tool for statistics and data analysis.
- Learning a foreign language.

Introduction to **R**

- **R** is an open-source statistical programming environment.
- Freely available from the Comprehensive **R** Archive Network (CRAN).
- A powerful tool for statistics and data analysis.
- Learning a foreign language.
- **R** or **RStudio**?

Basic operations

```r
4 + 2
```

```
## [1] 6
```

Basic operations

```
4 + 2
```

```
## [1] 6
```

```
4 - 2
```

```
## [1] 2
```

Basic operations

```r
4 + 2
```

```
## [1] 6
```

```r
4 - 2
```

```
## [1] 2
```

```r
4 * 2
```

```
## [1] 8
```

Basic operations (cont.)

```
4 * (5 - 3)

## [1] 8
```

Basic operations (cont.)

```
4 * (5 - 3)

## [1] 8
```

```
4 / 2

## [1] 2
```

Basic operations (cont.)

```
4 * (5 - 3)
```

```
## [1] 8
```

```
4 / 2
```

```
## [1] 2
```

```
4 ^ 2
```

```
## [1] 16
```

Basic operations (cont.)

```
4 * (5 - 3)

## [1] 8
```

```
4 / 2

## [1] 2
```

```
4 ^ 2

## [1] 16
```

```
sqrt(4)

## [1] 2
```

Creating objects

Create an object `x` that saves information:

```
x <- 4 * 2
```

Creating objects

Create an object `x` that saves information:

```r
x <- 4 * 2
```

View object:

```r
x

## [1] 8

(x <- 4 * 2)

## [1] 8
```

Numeric objects: double and integer

```
(x1 <- 2.0)

## [1] 2

typeof(x1)

## [1] "double"
```

Numeric objects: double and integer

```
(x1 <- 2.0)

## [1] 2

typeof(x1)

## [1] "double"
```

```
(x2 <- 2L)

## [1] 2

typeof(x2)

## [1] "integer"
```

Character objects

```
(instructor <- "Elias")

## [1] "Elias"
```

Character objects

```r
(instructor <- "Elias")

## [1] "Elias"
```

```r
(instructor <- "Elias Nosrati")

## [1] "Elias Nosrati"
```

Character objects

```
(instructor <- "Elias")

## [1] "Elias"
```

```
(instructor <- "Elias Nosrati")

## [1] "Elias Nosrati"
```

```
typeof(instructor)

## [1] "character"
```

Character objects

```r
(instructor <- "Elias")

## [1] "Elias"
```

```r
(instructor <- "Elias Nosrati")

## [1] "Elias Nosrati"
```

```r
typeof(instructor)

## [1] "character"
```

```r
number_as_character <- "10"
number_as_character / 2

## Error in number_as_character/2:  non-numeric
argument to binary operator
```

Logical objects

```
5 > 2
```

```
## [1] TRUE
```

Logical objects

```
5 > 2

## [1] TRUE
```

```
x <- 5
x > 2

## [1] TRUE
```

Logical objects

```
5 > 2

## [1] TRUE
```

```
x <- 5
x > 2

## [1] TRUE
```

```
y <- x > 2
typeof(y)

## [1] "logical"
```

Data structures: vectors

```r
x <- c(1, 2, 3)
x

## [1] 1 2 3
```

Data structures: vectors

```r
x <- c(1, 2, 3)
x

## [1] 1 2 3
```

```r
a <- c("a", 2, FALSE)
b <- c("z", 47L)
c <- c(a, b)
c

## [1] "a"     "2"     "FALSE" "z"     "47"
```

Data structures: vectors (cont.)

```
(x <- c(1, 2, 3, 4, 5))

## [1] 1 2 3 4 5
```

Data structures: vectors (cont.)

```
(x <- c(1, 2, 3, 4, 5))

## [1] 1 2 3 4 5
```

```
(x <- 1:5)

## [1] 1 2 3 4 5
```

Data structures: vectors (cont.)

```r
(x <- c(1, 2, 3, 4, 5))

## [1] 1 2 3 4 5
```

```r
(x <- 1:5)

## [1] 1 2 3 4 5
```

```r
(x <- seq(1, 5))

## [1] 1 2 3 4 5
```

Data structures: vectors (cont.)

```r
(x <- c(1, 2, 3, 4, 5))

## [1] 1 2 3 4 5
```

```r
(x <- 1:5)

## [1] 1 2 3 4 5
```

```r
(x <- seq(1, 5))

## [1] 1 2 3 4 5
```

```r
(x <- seq(10, 50, by = 10))

## [1] 10 20 30 40 50
```

Data structures: matrices

```
matrix(1:4, nrow = 2)

##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

Data structures: matrices

```
matrix(1:4, nrow = 2)

##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
matrix(1:4, nrow = 2, byrow = TRUE)

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

Data structures: lists

```r
list(1:3, c("a", "b"), TRUE, 44.7, " ")

## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] "a" "b"
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] 44.7
##
## [[5]]
## [1] " "
```

Data structures: data frame

```
data.frame("Var_1" = 1:3, "Var_2" = 4:6)

##   Var_1 Var_2
## 1     1     4
## 2     2     5
## 3     3     6
```

Data structures: data frame

```
data.frame("Var_1" = 1:3, "Var_2" = 4:6)

##   Var_1 Var_2
## 1     1     4
## 2     2     5
## 3     3     6
```

```
df <- data.frame("Var_1" = 1:3, "Var_2" = 4:6)
names(df) <- c("Variable_1", "Variable_2")
df

##   Variable_1 Variable_2
## 1          1          4
## 2          2          5
## 3          3          6
```

Converting between data structures

```
m <- matrix(1:100, nrow = 10)
as.data.frame(m)

##    V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
## 1   1 11 21 31 41 51 61 71 81  91
## 2   2 12 22 32 42 52 62 72 82  92
## 3   3 13 23 33 43 53 63 73 83  93
## 4   4 14 24 34 44 54 64 74 84  94
## 5   5 15 25 35 45 55 65 75 85  95
## 6   6 16 26 36 46 56 66 76 86  96
## 7   7 17 27 37 47 57 67 77 87  97
## 8   8 18 28 38 48 58 68 78 88  98
## 9   9 19 29 39 49 59 69 79 89  99
## 10 10 20 30 40 50 60 70 80 90 100
```

Indexing

```r
x <- 1:10
x[5]

## [1] 5
```

Indexing

```r
x <- 1:10
x[5]

## [1] 5
```

```r
x[2:7]

## [1] 2 3 4 5 6 7
```

Indexing

```
x <- 1:10
x[5]

## [1] 5
```

```
x[2:7]

## [1] 2 3 4 5 6 7
```

```
x[c(2, 7)]

## [1] 2 7
```

Indexing

```
x <- 1:10
x[5]

## [1] 5
```

```
x[2:7]

## [1] 2 3 4 5 6 7
```

```
x[c(2, 7)]

## [1] 2 7
```

```
x[-c(2, 7)]

## [1]  1  3  4  5  6  8  9 10
```

Indexing (cont.)

```
df <- data.frame("V1" = 1:3, "V2" = 4:6)
df[2]

##   V2
## 1  4
## 2  5
## 3  6
```

Indexing (cont.)

```
df <- data.frame("V1" = 1:3, "V2" = 4:6)
df[2]

##   V2
## 1  4
## 2  5
## 3  6
```

```
df$V2

## [1] 4 5 6
```

Indexing (cont.)

```
df[3, 2]

## [1] 6
```

Indexing (cont.)

```
df[3, 2]

## [1] 6
```

```
subset(df, V1 == 1 & V2 == 4)

##   V1 V2
## 1  1  4
```

Indexing (cont.)

```
df[3, 2]

## [1] 6
```

```
subset(df, V1 == 1 & V2 == 4)

##   V1 V2
## 1  1  4
```

```
subset(df, V1 == 2 | V2 != 4)

##   V1 V2
## 2  2  5
## 3  3  6
```

Functions

We have already seen several functions: `c()`, `class()`, `data.frame()`, etc.

Functions

We have already seen several functions: `c()`, `class()`, `data.frame()`, etc.

```
my_function <- function(input) {

  # Define output using input here

  return(output)
}
```

Functions: constructing $f(x) = x^2 + 4$

```r
my_function <- function(x) { # Function takes input x
  y <- x^2 + 4 # Expression for f(x)
  return(y) # Output
}
```

Functions: constructing $f(x) = x^2 + 4$

```r
my_function <- function(x) { # Function takes input x
  y <- x^2 + 4 # Expression for f(x)
  return(y) # Output
}
```

```r
my_function(2)

## [1] 8
```

Functions: creating a summary function

```
my_summary <- function(x) { # Input
  s_out <- sum(x) # Sum
  l_out <- length(x) # Length
  m_out <- s_out / l_out # Mean
  out <- c(s_out, l_out, m_out) # Define output
  names(out) <- c("Sum", "Length", "Mean") # Labels
  return(out) # End function by calling output
}
```

Functions: creating a summary function

```
my_summary <- function(x) { # Input
  s_out <- sum(x) # Sum
  l_out <- length(x) # Length
  m_out <- s_out / l_out # Mean
  out <- c(s_out, l_out, m_out) # Define output
  names(out) <- c("Sum", "Length", "Mean") # Labels
  return(out) # End function by calling output
}
```

```
input <- 1:10
```

Functions: creating a summary function

```r
my_summary <- function(x) { # Input
  s_out <- sum(x) # Sum
  l_out <- length(x) # Length
  m_out <- s_out / l_out # Mean
  out <- c(s_out, l_out, m_out) # Define output
  names(out) <- c("Sum", "Length", "Mean") # Labels
  return(out) # End function by calling output
}
```

```r
input <- 1:10
```

```r
my_summary(input)

##    Sum Length   Mean
##   55.0   10.0    5.5
```

Packages

- In **R**, the fundamental unit of shareable code is the package.

Packages

- In **R**, the fundamental unit of shareable code is the package.
- A package bundles together code, data, documentation, and tests, and is easy to share with others.

Packages

- In **R**, the fundamental unit of shareable code is the package.
- A package bundles together code, data, documentation, and tests, and is easy to share with others.
- You install them from CRAN with `install.packages("name_of_package")`.

Packages

- In **R**, the fundamental unit of shareable code is the package.
- A package bundles together code, data, documentation, and tests, and is easy to share with others.
- You install them from CRAN with `install.packages("name_of_package")`.
- You load them into **R** by typing `library(name_of_package)`.

Packages

- In **R**, the fundamental unit of shareable code is the package.
- A package bundles together code, data, documentation, and tests, and is easy to share with others.
- You install them from CRAN with `install.packages("name_of_package")`.
- You load them into **R** by typing `library(name_of_package)`.
- To get help on usage, type `package?name_of_package` and `help(package = "name_of_package")`.

Scripts

- To save and replicate extended chunks of code, use a text editor.

Scripts

- To save and replicate extended chunks of code, use a text editor.
- Text editor included in **R** and **RStudio**.

Scripts

- To save and replicate extended chunks of code, use a text editor.
- Text editor included in **R** and **RStudio**.
- Write and execute code with editor and save text file with .R file extension.

Data and directories

- You can read and upload various types of data files into **R**.

Data and directories

- You can read and upload various types of data files into **R**.
- We will mostly use *comma-separated values* (CSV) files.

Data and directories

- You can read and upload various types of data files into **R**.
- We will mostly use *comma-separated values* (CSV) files.
- Use `read.csv("file_name.csv")` to load file.

Data and directories

- You can read and upload various types of data files into **R**.
- We will mostly use *comma-separated values* (CSV) files.
- Use `read.csv("file_name.csv")` to load file.
- Store external files in your *working directory*.

Data and directories

- You can read and upload various types of data files into **R**.
- We will mostly use *comma-separated values* (CSV) files.
- Use `read.csv("file_name.csv")` to load file.
- Store external files in your *working directory*.
- View directory: `getwd()`.

Data and directories

- You can read and upload various types of data files into **R**.
- We will mostly use *comma-separated values* (CSV) files.
- Use `read.csv("file_name.csv")` to load file.
- Store external files in your *working directory*.
- View directory: `getwd()`.
- Change directory: `setwd("new_directory")`.

Some hands-on exercises

- Install and load the `swirl` package.

Some hands-on exercises

- Install and load the `swirl` package.
- Then type `install_course_github("kosukeimai", "qss-swirl")`.

Some hands-on exercises

- Install and load the `swirl` package.
- Then type `install_course_github("kosukeimai", "qss-swirl")`.
- Type `swirl()` to start the first exercise.

Homework

- Chapters 2 and 3 of Grolemund's *Hands-On Programming with R* (available online).

Homework

- Chapters 2 and 3 of Grolemund's *Hands-On Programming with R* (available online).
- Chapters 3–5 of Wickham and Grolemund's *R for Data Science* (available online).

Homework

- Chapters 2 and 3 of Grolemund's *Hands-On Programming with R* (available online).
- Chapters 3–5 of Wickham and Grolemund's *R for Data Science* (available online).
- Don't just read – type in all the commands yourself and try the exercises.

Homework

- Chapters 2 and 3 of Grolemund's *Hands-On Programming with R* (available online).
- Chapters 3–5 of Wickham and Grolemund's *R for Data Science* (available online).
- Don't just read – type in all the commands yourself and try the exercises.
- Complete Problem Sheet 1 and submit your **R** scripts by email at least 24h before the next lecture.