02-680 Module 1

Essentials of Mathematics and Statistics

Aidan Jan

August 28, 2025

Logistics

Grading

- 40% Homework
- 10% Attendance, participation, quiz
- 50% Exams (25% each)

Quizzes and Exams

- Quizzes are distributed and taken via Canvas before class starts. Extended quizzes may be present during class.
- There are two exams, and they are not cumulative.

Homework

- Collaboration is encouraged, but answers must be your own.
- Generative AI is allowed.

Modules

1. Fundamentals (Logic, Sets, Tuples, Graphs and Trees)
2. Linear Algebra (Vectors, Matrices and Operations, Solving Linear Equations, Vector Spaces, Linear Independence, Eigenvalues and Eigenvectors)
3. Calculus (Calculus, Matrix Calculus)
4. Probability (Independence, Conditional Probability, Bayes' Theorem)
5. Statistics (Random Variables, PDF, CDF)
6. Applications of Statistics (T-test, Categorical Data, Multiple Testing, Confidence Intervals, Sampling, Linear Regression)

Fundamentals - Logic

Logic forms the **essential building block of mathematical reasoning**, allowing us to formalize statements, analyze arguments, and lay the groundwork for more advanced topics in linear algebra, probability, and statistics.

- *Propositional logic* often includes variables **without assigned values**, so we can't know if the whole statement is true or false until the variables are defined.
- *Predicate logic* specifies **what the variables represent**, allowing the full statement to be evaluated; it extends propositional logic by adding more detail.

Propositional Logic

There are three types of propositions:

- An *atomic proposition* is one that is conceptually indivisible; it cannot be broken down into simpler statements. Atomic propositions are also sometimes called Boolean variables.
  - Usually represented using lower-case letters.
  - e.g., $p \leftarrow$ "$2 + 2 = 4$", $q \leftarrow$ "Bear is CMU mascot"
  - p is true, q is false.
- A *negate proposition* negates a statement.
$$\neg q$$
  - This is read as "not q", or "Bear is not CMU mascot."
- A *compound proposition* is one formed by combining simpler propositions using logical connectives. A compound proposition that contains boolean variables $p_1, p_2, \ldots, p_k$ is sometijmes called a boolean expression or boolean formula over $p_1, p_2, \ldots, p_k$.
  - A compound proposition can take into account multiple atomic propositions to create a single statement:
$$p \wedge q$$
  - The statement does not necessarily be true. We read the above as "p and q, and for the whole statement to be true, both p and q must be true.

Logical Connectives

There are multiple types of logical connectives, or links that combine simple propositions into more complex compound propositions.

- Negation [not, $\neg$]: mentioned before, negates the following statement
- Conjunction [and, $\wedge$]: returns true if the two statements on both sides are true
- Disjunction [or, $\vee$]: returns true if at least one of the two statements are true
- Implication [if/then, $\Rightarrow$]: $p \Rightarrow q$ means if p is true, then q is also true. It is false only when p is true and q is false.
  - In the $p \Rightarrow q$ structure, p is called the *antecedent* or *hypothesis*. q is called the *consequent* or *conclusion*.
  - p does not necessarily cause q! p being true implies q is true. However, p being false does not necessarily mean q is false.
  - If p is true, then q must be true. If q turns out to be false, then the entire implication is false since the promise of the implication was violated.

- If and only If [$\Leftrightarrow$]: returns true when p and q are both true, or both false. returns false otherwise
  - Sometimes called a *biconditional* or *mutual implication*.
- Exclusive Or [xor, $\oplus$]: returns true when either p or q is true, but not both.

The logical connectives have a precedence in the order they are applied, or an **order of operations.** The highest precedence "binds the tightest", and the lowest precedence "binds the loosest". These rules tell us when to include parentheses on an expression to make it mean what we want it to mean, and when parentheses are optional.

1. Highest Precedence: Negation $\neg$
2. Medium Precedence (Equal importance): Conjunction $\wedge$, Disjunction $\vee$, Exclusive Or $\oplus$
3. Lower Precedence: Implication $\Rightarrow$
4. Lowest Precedence: Biconditional $\Leftrightarrow$

For example, the following two statements are identical:

$$p \vee q \Rightarrow \neg r \wedge l$$
$$(p \vee q) \Rightarrow (\neg r \wedge l)$$

De Morgan's Laws

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$
$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

(here, $\equiv$ means logical equivalence.) These laws can be proven using a truth table.

Associativity and Commutativity

Just like in addition and multiplication, most logical operators (specifically conjunction[$\wedge$], disjunction[$\vee$], exclusive or[$\oplus$], and mutual implication[$\Leftrightarrow$]) are associative and commutative. For example,

$$(p \vee q) \vee r \equiv v \vee (q \vee r) \qquad \text{(associativity)}$$
$$p \wedge q \equiv q \wedge p \qquad \text{(commutativity)}$$

Negation and implication are not associative or commutative.

Additional Logically Equivalent Propositions

Distribution of $\wedge$ over $\vee$:

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

Distribution of $\vee$ over $\wedge$:

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

Mutual Implication:

$$(p \Rightarrow q) \wedge (q \Rightarrow p) \equiv p \Leftrightarrow q$$

Bound and Free Variables

While all the stuff above we know if p and q are always true or false, often a proposition looks more like

$$r \leftarrow \text{"}2 + x = 4\text{"}$$

Without a definition of x we can't say if the statement is true or not.

Truth tables

A truth table shows all possible truth assignments for a proposition. Each row corresponds to one trutth assignment. The last column gives the truth value of the whole proposition under that assignment. Example: Defining $\wedge$

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- The first two columns list every possible combination of p and q. Since there are two variables, we need four rows. (TT, TF, FT, FF).
- The third column shows the result of the compound proposition.

Truth tables for the basic logical connectives

p	q	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \oplus q$	$p \Leftrightarrow q$
T	T	T	T	T	T	T
T	F	F	T	F	T	F
F	T	F	T	T	T	F
F	F	F	F	T	F	T

Predicate Logic

Simple propositional logic has a significant limitation: it can't deal with *properties of individual objects within a collection.* For handling such properties, we need predicate logic.

Universal Quantifier

If we want to say something like: "*for any* integer x, the value of $x \cdot 0$ is 0." We would write that using the **universal quantifier**.

$$\forall x \in \mathbb{Z} \, : \, x \cdot 0 = 0$$

We say $\forall$ as "for all". Notice that "$x \cdot 0 = 0$" is a proposition!

- Let P be a predicate over S. The proposition $\forall x \in S \, : \, P(x)$ is true if, for every possible $x \in S$, we have that $P(x)$ is true.

Existential Quantifier

But sometimes we want to not talk about all items, but one (or more) in particular, in which case we can use the **existential quantifier**.

$$\exists y \in \mathbb{Z} \, : \, y^2 = |y + y|$$

We say $\exists$ as "there exists". In this case there exists more than one y that satisfies the proposition, so the statement is true.

- Let P be a predicate over S. The proposition $\exists x \in S \, : \, P(x)$ is true if, for at least one possible $x \in S$, we have that $P(x)$ is true.

Quantifier Negation

- We may also need to negate expressions with quantifiers.
- Just like with propositional logic it is not trivial. Once again De Morgan's Theorem comes to the rescue:

$$\neg[\forall x \ : \ P(x)] \Leftrightarrow [\exists x \ : \ \neg P(x)]$$
$$\neg[\forall x \ : \ P(x)] \Leftrightarrow [\forall x \ : \ \neg P(x)]$$

- Notice that whats left to do ($\neg P(x)$) is just the negation of a proposition, which was described earlier.

Nested Quantifiers

In predicate logic, we can combine multiple quantifiers (like $\forall$ and $\exists$) in one expression. This is called **nesting**.

- Changing the **order** can completely change the meaning of the statement.

For example, consider

$$\forall x \in \mathbb{Z} \ : \ [\exists y \in \mathbb{Z} \ : \ x = -y]$$

"For every integer x, there exists an integer y such that $x = -y$."

- The **outer quantifier** is $\forall x \in \mathbb{Z}$.
- The **inner quantifier** is $\exists y \in \mathbb{Z}$.
- The **predicate** is $x = -y$.

If we write this statement incorrectly (in reverse order) we find a statement that is unsatisfiable.

$$\exists x \in \mathbb{Z} \ : \ [\forall y \in \mathbb{Z} \ : \ x = -y]$$

"There exists an integer y such that for every integer x, $x = -y$."

- This is **false** because no single y can produce all possible x values by $x = -y$.

Negating Nested Quantifiers

When negating nested quantifiers, it is basically a *recursive* procedure.

- Start with the outermost quantifier.
- Apply negation one layer at a time, moving inward.
- Flip each quantifier according to De Morgan's Laws:

$$\neg[\forall x \ : \ P(x)] \Leftrightarrow [\exists x \ : \ \neg P(x)]$$
$$\neg[\forall x \ : \ P(x)] \Leftrightarrow [\forall x \ : \ \neg P(x)]$$

For example:

$$\neg[\exists x \ : \ [\forall x \ : \ P(x)]] \Leftrightarrow [\forall x \ : \ \neg[\forall x \ : \ P(x)]] \Leftrightarrow [\forall x \ : \ [\exists x \ : \ \neg P(x)]]$$