

Using Pre-Trained AI

Shifting gears from building models to
using them

Setting the Scene

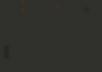
LLMs: Large Language Models

- These are models like **ChatGPT (OpenAI)**, **Claude (Anthropic)**, **Gemini (Google)**, or **Llama (Meta)**
- "Large" because they are trained on vast swathes of data (most of the internet!), and contain billions of parameters
- At the most basic level, they **take some text as input**, and then successively **predict the next word** as their output
- Turns out that this basic operation is enough to allow sophisticated reasoning, problem-solving, code generation, data analysis, and more!



* Back at it, Sandy

How can I help you today?



Opus 4.5



</> Code

Create

Learn

Write

Life stuff

openai.com

Research

Safety

For Business

For Developers

ChatGPT

Sora

Stories

Company

News

Introducing ChatGPT Go, now available worldwide
Product 3 min read

The new ChatGPT Images is here
Product 7 min read

Introducing GPT-5.2-Codex
Product 5 min read

Log in

anthropic.com

ANTHROPIC

Research Economic Futures Commitments Learn News Try Claude

AI research and products that put safety at the frontier

AI will have a vast impact on the world. Anthropic is a public benefit corporation dedicated to securing its benefits and mitigating its risks.

Claude Opus 4.5

Introducing the best model in the world for coding, agents, computer use, and enterprise workflows.

Learn more →

Introducing Claude Opus 4.5 →

Advanced tool use on the Claude →

Try Gemini

Ask Gemini

Discover cities with history and art scenes

Create an imaginative aerial landscape

Challenge Gemini to guess what you're drawing

Speak your app idea existence

help with your golf swing

Create vertical videos

Create a custom quiz from audio input

Transform a sketch into a realistic image

Terminology

Tokens: inputs (text, images, etc.) are broken up into ***t*okens** which are smaller chunks of data that the model can use, this is ***tokenisation***

Parameter/model weights: these are set during the model's learning/training and determine how the model acts/behave

Context window: this is the amount of input and output data that the model can "see" at any one time before it forgets earlier information

platform.openai.com/tokenizer

OpenAI Platform

Docs API reference Log in Sign up

Look again at that dot. That's here. That's home. That's us. On it everyone you love, everyone you know, everyone you ever heard of, every human being who ever was, lived out their lives. The aggregate of our joy and suffering, thousands of confident religions, ideologies, and economic doctrines, every hunter and forager, every hero and coward, every creator and destroyer of civilization, every king and peasant, every young couple in love, every mother and father, hopeful child, inventor and explorer, every teacher of morals, every corrupt politician, every "superstar," every "supreme leader," every saint and sinner in the history of our species lived there--on a mote of dust suspended in a sunbeam.

Clear Show example

Tokens Characters
150 708

Look again at that dot. That's here. That's home. That's us. On it everyone you love, everyone you know, everyone you ever heard of, every human being who ever was, lived out their lives. The aggregate of our joy and suffering, thousands of confident religions, ideologies, and economic doctrines, every hunter and forager, every hero and coward, every creator and destroyer of civilization, every king and peasant, every young couple in love, every mother and father, hopeful child, inventor and explorer, every teacher of morals, every corrupt politician, every "superstar," every "supreme leader," every saint and sinner in the history of our species lived there--on a mote of dust suspended in a sunbeam.

Text Token IDs

Terminology

Training: learning from data to build new model versions; expensive and amortised across all the model's users

Inference: actually using the trained model to generate output; costs the user per token (input and output)

Prompt: an instruction given to the model that it acts upon

RAG (Retrieval Augmented Generation): extra sources of information (documents, web content) given to the model at inference time

Timeline - Foundation Years

"Attention is all you need" paper published

2018



GPT-2 launches,
"too dangerous to release"

2017

GPT-1
launches

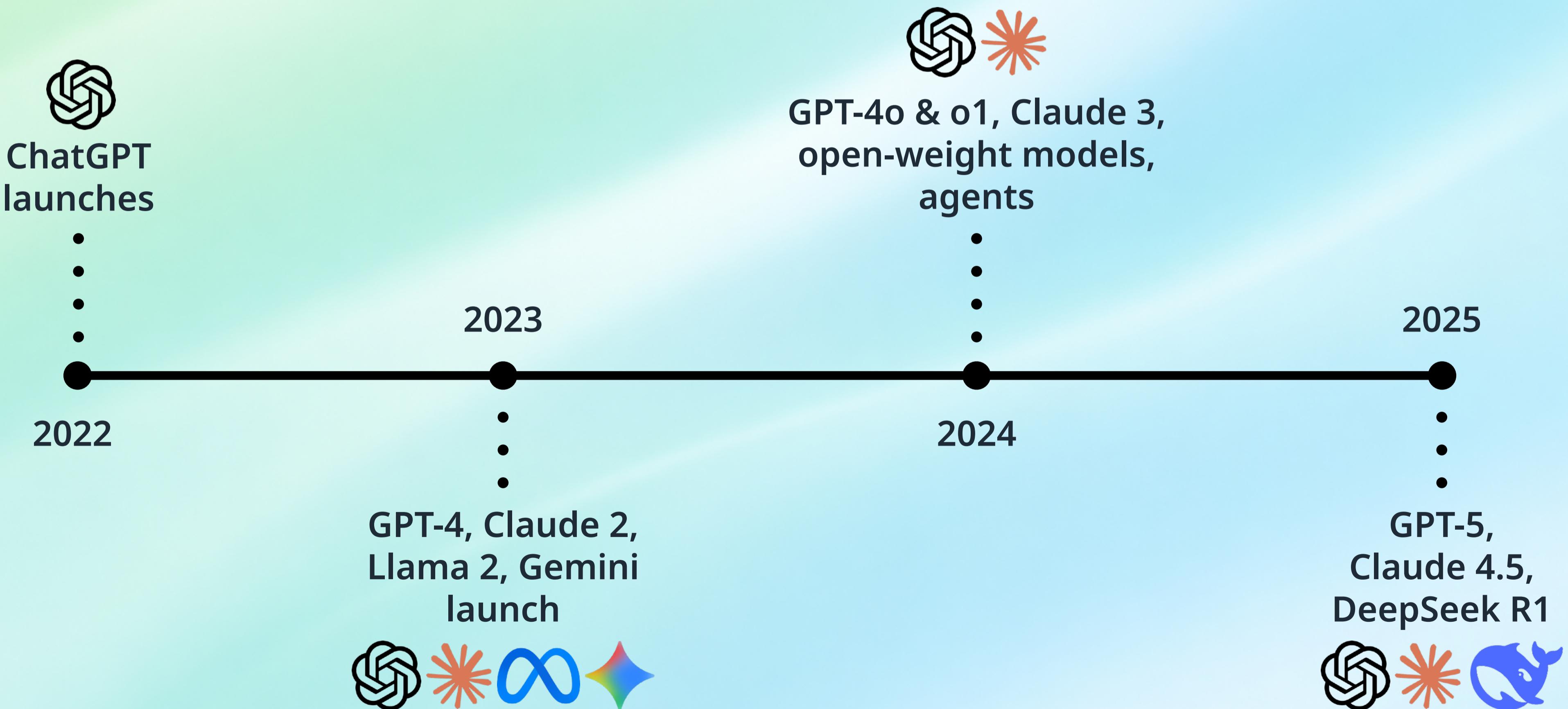


2020

GPT-3
launches



Timeline - The AI Explosion



LLM Architecture

System prompt: defines behaviour, model's persona, constraints

Context/memory: previous conversation, memories

Hidden from user

Tools: web searches, code writing/execution, file creation

Guardrails: prompt refusal, content filtering

.....

User prompt: what you type into the chat

Differences Between Platforms*

WHICH SHOULD I PICK, AND HOW DO THEY DIFFER?

	ChatGPT 5.2	Claude Opus 4.5	Gemini 2.5 Pro	Gemma3 1B (local)
Context window	400K tokens	200K tokens	1M tokens	128K tokens
Model size	Likely 300B to 2T	Not disclosed	Not disclosed	1 billion
Agents/tools	Computer use, web search, code exec.	Computer use, MCP/APIs, Claude Code	Computer use, web search, code exec.	Limited
Reasoning?	Yes	Yes	Yes	No
Privacy	Data sent to cloud	Data sent to cloud	Data sent to cloud	No data sent
Image generation	Yes	No	Yes	No
Cost (first paid tier)	\$20/month	\$20/month	\$20/month	Free

* Data accurate at end of January 2026

Questions:

Which LLMs have you used?

What did you like/dislike?

Local Models

Why bother?

Privacy: no data sent to the cloud, can be completely air-gapped

Compliance: if being used for sensitive data or patentable purposes

Cost: free, unlimited use

Learning: good for experimentation



The Capability Gap

Gemma3 1B



Llama 8B



Claude Sonnet



Claude Opus/GPT-5



** for illustration only, not to scale*

Bigger models are better for complex reasoning tasks, but not always necessary for basic jobs e.g. text formatting or basic questions

Which Model to Choose?

- Simple task → Smaller, local model may suffice
- Complex reasoning needed → Frontier model like ChatGPT
- Sensitive data → Local model or locked-down enterprise model like Gemini
- Up-to-date information needed → Frontier model like ChatGPT
- Coding with a codebase → Claude Code, or IDE integration in VS Code or Cursor

Best Practices

The Art of Good Prompting

"Garbage in, garbage out"

Analogy is asking someone to build a piece of IKEA furniture but not giving them the instructions. They'll struggle, or fail to successfully build it.

- Be specific
- Provide context
- Give examples, if possible

Structuring a Prompt

Context: "I am trying to analyse some RNA-seq data from an experiment studying cell cycle regulation in mice"

Task: "I would like to pull out the top 50 differentially expressed genes with a log² fold change threshold of +/- 2 and an adjusted p value threshold of < 0.05"

Constraints: "Just give me the canonical gene names and Ensembl gene IDs back in CSV format"

Example: "For example: Ciz1,ENSMUSG00000039205"

Question:

*How could you make this prompt
better?*

"Tell me about gene expression"

Strong Points for LLMs

- Explaining or summarising topics
- Reformatting, changing tone, or summarising of written text
- Translating written text or spoken word
- Generating code
- Idea generation / brainstorming

LLMs do well with topics for which there is a large corpus of information, or with well-defined tasks. They struggle more with problem-solving or lateral-thinking.



✳️ Good afternoon, Sandy

Can you give me a summary of what a C56BL/6 mouse is?

+



Opus 4.5



Sticking Points for LLMs

- ***Hallucinations:*** LLMs will occasionally just tell you something that is not true, or has been invented. **Always sanity check and fact check!**
- ***Sycophancy:*** depending on the LLM, it will sometimes go out of its way to please you by giving you an answer even if it is not certain. **You can tailor this through your system prompt.**
- ***Knowledge lag:*** the model of LLM you use will not encompass recent events, so bear in mind that information may be slightly or significantly out of date. **You can enable web searching/research to improve this.**
- ***Context limit:*** the model you use has a limited context, meaning that very long "chats" will start to forget earlier context. **Choosing a model with a large context window can help here.**

Question:

You ask your LLM to give you five references on a particular topic. It gives you back five authors, titles, and journal references.

What should you do? How could you avoid mistakes?

Comparing LLMs

LLM Tasks

ChatGPT 5.2, Claude Opus 4.5, Gemini 3, and Gemma3 1B asked the same three prompts, and we'll compare their output:

1. *"Explain the difference between Type I and Type II errors to someone with basic statistics knowledge. Use a biological research example."*
2. *"Write a Python function that calculates Benjamini-Hochberg adjusted p-values for a list of p-values."*
3. *"I'm designing an experiment to test whether a new antibiotic is effective against E. coli biofilms. What controls should I include?"*

Prompt 1:

"Explain the difference between Type I and Type II errors to someone with basic statistics knowledge. Use a biological research example."

chatgpt.com/c/697b7080-70a4-8326-aa32-50702f676de8

ChatGPT 5.2

Share

Explain the difference between Type I and Type II errors to someone with basic statistics knowledge. Use a biological research example.

Here's a clean, intuitive way to think about it, using a biological research example.

The setup (hypothesis testing)

Suppose you're testing a new drug to see whether it **reduces tumour size** compared to a placebo.

- **Null hypothesis (H_0):** The drug has *no effect* on tumour size.
- **Alternative hypothesis (H_1):** The drug *does reduce* tumour size.

Now, two kinds of mistakes are possible.

Type I error (false positive)

You conclude there *is* an effect when there actually isn't.

In our example:

You find a statistically significant reduction in tumour size and claim the drug works — **but in reality, it doesn't**. The observed effect was just random variation.

- Probability of a Type I error = **α (alpha)**, often set at **0.05**
- Interpretation: "*I said the drug works when it  really doesn't.*"

This is especially serious in biology and medicine because it can lead to:

+ Ask anything

ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#).

ChatGPT 5.2 response

Type I and Type II errors in biological research

Explain the difference between Type I and Type II errors to someone with basic statistics knowledge. Use a biological research example.

Type I and Type II Errors

Imagine you're testing whether a new drug reduces inflammation in mice compared to a control group.

Type I error (false positive) occurs when you conclude the drug works, but it actually doesn't. You've detected an effect that isn't real — perhaps by chance, your treated mice happened to show lower inflammation, leading you to reject a true null hypothesis. The probability of making this error is your significance level (α), typically set at 0.05.

Type II error (false negative) occurs when you conclude the drug doesn't work, but it actually does. You've missed a real effect — perhaps your sample size was too small to detect a modest but genuine reduction in inflammation. The probability of making this error is β , and your statistical power ($1 - \beta$) represents your ability to detect true effects.

A useful way to think about the trade-off: if you're very strict about avoiding false positives (using $\alpha = 0.01$ instead of 0.05), you become more likely to miss real effects. Conversely, if you're lenient to avoid missing anything, you'll end up with more false leads.

In practice, which error matters more depends on context. For a preliminary screen of drug candidates, false negatives might be more costly — you don't want to discard something promising. For a confirmatory clinical trial, false positives are the bigger concern — you don't want to approve an ineffective treatment.

Reply...

Opus 4.5

Claude is AI and can make mistakes. Please double-check responses.

Claude Opus 4.5 response

gemini.google.com/app/a7c809fbe9037d01

Gemini

Understanding Statistical Errors: Type I & II

UNIVERSITY OF YORK

W¹²

Think of statistical errors as the two ways a researcher can be "fooled" by their own data. Even with a perfect experiment, randomness can occasionally make you see things that aren't there—or miss things that are.

In the world of hypothesis testing, we usually start with a **Null Hypothesis (H_0)**, which is the "boring" assumption that nothing interesting is happening (e.g., a drug doesn't work).

The Decision Matrix

Type I Error: The "False Positive"

A Type I error occurs when you reject the null hypothesis, but the null hypothesis was actually true. In plain English: **you found an effect that isn't really there.**

- **Statistical Symbol:** α (Alpha)
- **The Vibe:** You're being a bit too "over-eager." You think you've made a breakthrough, but it was just a statistical fluke.

Type II Error: The "False Negative"

A Type II error occurs when you fail to reject the null hypothesis, but it was actually false. In plain English: **you missed an effect that actually exists.**

Enter a prompt for Gemini

Tools

Thinking

Your University of York chats aren't used to improve our models. Gemini can make mistakes, including about people, so double-check it.
[Your privacy and Gemini](#)

Gemini 3 response

The screenshot shows a dark-themed messaging application window. At the top left are three colored window control buttons (red, yellow, green). Below them is a toolbar with a 'New Chat' button and a 'Today' button. A message input field at the bottom left contains the placeholder text 'Explain the difference betwee...'. A large, semi-transparent callout bubble is positioned in the upper right area, containing the following text:
Explain the difference between Type I and Type II errors
to someone with basic statistics knowledge. Use a
biological research example.

Below the input field, the message history starts with a response from the bot: 'Okay, let's break down Type I and Type II errors in statistics, using a biological example to make it easier to understand.'

A section titled 'The Basic Idea' follows, with the text: 'In statistics, we're trying to make decisions about something – like whether a drug is effective or not, or whether a particular gene is linked to a disease. We often get a "best guess" based on data. However, these guesses aren't always perfect. There's a risk of making a mistake. Type I and Type II errors are two ways these mistakes can happen.'

Under the heading '1. Type I Error (False Positive)', there is a bulleted list of three items:

- **What it is:** A Type I error means you *incorrectly conclude* that a treatment or hypothesis is effective when it actually isn't. Essentially, you're saying there's an effect when there isn't one.
- **Think of it like:** A medical test that says a patient *has a disease*, when they actually *don't*. You're falsely claiming a positive result.
- **Example in Biology:** Let's say a researcher is testing a new drug to treat a specific type of cancer. They conduct a clinical trial and find that a group of patients treated with the new drug shows significantly better results than a control group. *However*, because of random chance, they might also find a *slightly* better result in the group treated with the new drug. They've made a Type I error – they *incorrectly* concluded the drug is effective. They've falsely reported

At the bottom of the window, there is a 'Send a message' input field and a footer bar with three circular icons: a plus sign, the name 'gemma3:1b' with a dropdown arrow, and an upward arrow.

Gemma3-1B response

Prompt 2:

*"Write a Python function
that calculates Benjamini-
Hochberg adjusted p-values
for a list of p-values."*

ChatGPT 5.2

Write a Python function that calculates Benjamini-Hochberg adjusted p-values for a list of p-values.

```
python
from __future__ import annotations

from typing import Iterable, List, Sequence, Union
import math

Number = Union[int, float]

def benjamini_hochberg(pvalues: Sequence[Number]) -> List[float]:
    """
    Compute Benjamini-Hochberg (BH/FDR) adjusted p-values (q-values).

    Parameters
    -----
    pvalues : sequence of numbers
        Raw p-values. Must be finite and in [0, 1]. NaNs/infs raise ValueError.

    Returns
    -----
    list of float
    """

    sorted_pvalues = sorted(pvalues)
    n = len(sorted_pvalues)
    adjusted_pvalues = []
    for i, pvalue in enumerate(sorted_pvalues):
        adjusted_pvalue = pvalue * (n + 1 - i) / n
        adjusted_pvalues.append(adjusted_pvalue)

    return adjusted_pvalues
```

+ Ask anything

0

SA

ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#).

ChatGPT 5.2 response

claude.ai/chat/a2b36078-6d49-4113-9a4d-50fb7d43b8a3

Write a Python function that c...

Share

python

```
def benjamini_hochberg(p_values):
    """
    Calculate Benjamini-Hochberg adjusted p-values (FDR correction).

    Parameters
    -----
    p_values : list or array-like
        Raw p-values to adjust.

    Returns
    -----
    list
        Adjusted p-values in the original order.
    """
    n = len(p_values)
    if n == 0:
        return []
```

↓

Reply...

Opus 4.5

SM

indexed.sort(key=lambda x: x[0])

Claude is AI and can make mistakes support.anthropic.com/en/articles/8525154-claude-is...g-incorrect-or-misleading-responses-what-s-going-on

Claude Opus 4.5 response

gemini.google.com/app/9ccaf93cd7d34c27

Gemini Benjamini-Hochberg Adjusted P-Values

Python

```
import numpy as np

def benjamini_hochberg(p_values):
    """
    Calculates BH adjusted p-values (False Discovery Rate).

    Args:
        p_values (list or np.array): List of raw p-values.

    Returns:
        np.array: Adjusted p-values, mapped back to original order.
    """
    p_values = np.asarray(p_values)
    m = len(p_values)

    # Get indices that would sort the array
    sort_idx = np.argsort(p_values)
    reverse_idx = np.argsort(sort_idx)

    # Sort p-values
    sorted_p = p_values[sort_idx]

    # Calculate (p * m) / i
    # Ranks are 1-based, so we use np.arange(1, m + 1)
```

Enter a prompt for Gemini

Tools

Your University of York chats aren't used to improve our models. Gemini is AI and can make mistakes, including about people. [Your privacy and Gemini](#)

Gemini 3 response



Write a Python function that calculates Benjamini-Hochberg adjusted p-values for a list of p-values.

```
python Copy
from scipy import stats

def benjamini_Hochberg_adjusted_p_values(p_values, critical_value=0.05):
    """
    Calculates Benjamini-Hochberg adjusted p-values for a list of p-values.

    Args:
        p_values: A list of p-values to adjust.
        critical_value: The significance level to use for adjusting p-values. Default is 0.05.

    Returns:
        A list of adjusted p-values, sorted from smallest to largest.
    """

    adjusted_p_values = []
    for p_val in p_values:
        z = abs(p_val - 0.5) # Standardize p-value to the mean of the p-values
        adjusted_p_val = stats.norm.cdf(z)
        adjusted_p_values.append(adjusted_p_val)

    adjusted_p_values.sort(key=lambda x: abs(x - critical_value)) # Sort by adjusted p-value
    return adjusted_p_values

if __name__ == '__main__':
    # Example Usage
    p_values = [0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009]
    print(benjamini_Hochberg_adjusted_p_values(p_values))
```

Send a message

+ gemma3:1b ↕

Gemma3-1B response

Prompt 3:

*"I'm designing an experiment
to test whether a new
antibiotic is effective against
E. coli biofilms. What
controls should I include?"*

ChatGPT 5.2

I'm designing an experiment to test whether a new antibiotic is effective against *E. coli* biofilms. What controls should I include?

For an ***E. coli* biofilm antibiotic efficacy** experiment, you want controls that (1) prove biofilms formed, (2) prove your assay can detect killing/biomass change, (3) separate antibiotic effects from solvents/handling, and (4) catch contamination and plate/reader artifacts.

Core controls (minimum set)

- **Untreated biofilm control (negative treatment control)**
 - Biofilm grown normally, then exposed to the same treatment conditions but **no drug**.
 - This is your baseline for biomass/viability.
- **Vehicle/solvent control**
 - If the antibiotic is dissolved in DMSO/EtOH/etc., treat biofilms with **the same final solvent %** but no antibiotic.
 - Ensures any effect isn't from the solvent.
- **No-biofilm / background control**
 - Wells with **media only** that go through staining/reading steps.
 - (Optional but useful) Wells inoculated but **biofilm prevented** (e.g., very short incubation) to define non-adherent background, depending on your setup.
 - Lets you subtract background absorbance/fluorescence and detect reagent artifacts.
- **Positive control antibiotic (known anti-biofilm-active comparator)**
 - Use a standard antibiotic with known activity against *E. coli* in your system (pick one appropriate to your strain and assay). 
 - Confirms your assay and workflow can detect a "real" effect.

+ | Ask anything 

SA ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#).

ChatGPT 5.2 response

I'm designing an experiment to... Share

I'm designing an experiment to test whether a new antibiotic is effective against *E. coli* biofilms. What controls should I include?

Great question—biofilm experiments need careful controls to distinguish genuine antibiotic effects from artefacts. Here's what I'd recommend:

Negative controls

- Untreated biofilms (growth medium only) to establish your baseline biomass/viability
- Vehicle control if your antibiotic is dissolved in something other than water (DMSO, ethanol, etc.) at the same concentration used in treatments

Positive controls

- A known effective anti-biofilm agent or antibiotic with established activity against *E. coli* biofilms (ciprofloxacin is commonly used)
- If available, a biofilm-disrupting agent like DNase I can help validate your detection method is actually measuring biofilm matrix

Technical controls

- Sterile medium wells to check for contamination and establish your assay background
- Planktonic MIC in parallel—this lets you compare biofilm vs planktonic susceptibility

Reply... Opus 4.5

SM

Biological considerations

Claude is AI and can make mistakes. Please double-check responses.

Claude Opus 4.5 response

gemini.google.com/app/fded0f21d1b18414

Gemini Biofilm Antibiotic Efficacy Controls

I'm designing an experiment to test whether a new antibiotic is effective against *E. coli* biofilms. What controls should I include?

Designing an experiment for biofilms is a bit more complex than standard "planktonic" (free-floating) bacteria tests because biofilms are notoriously hardy—essentially the biological equivalent of a fortified bunker.

To ensure your results are actually due to the antibiotic and not just a fluke of the setup, you need a robust set of controls.

1. The Negative Control (Growth Control)

This is your baseline. You need to prove that, under normal conditions without any treatment, your *E. coli* actually forms a healthy, robust biofilm.

- Setup:** *E. coli* + Growth Media + **No Antibiotic**.
- Purpose:** To show what "100% growth" looks like. If this doesn't grow well, your experimental results are meaningless.

Enter a prompt for Gemini

Tools

Fast

Your University of York chats aren't used to improve our models. Gemini is AI and can make mistakes, including about people. [Your privacy and Gemini](#)

Gemini 3 response



I'm designing an experiment to test whether a new antibiotic is effective against *E. coli* biofilms. What controls should I include?

Okay, that's a really important and complex challenge! Testing the effectiveness of a new antibiotic against *E. coli* biofilms requires careful consideration of controls. Here's a breakdown of appropriate controls for your experiment, categorized by their purpose and rationale, along with explanations of why they're crucial:

1. Negative Controls – Establishing Baseline & Ruling Out Interference

- **Unexposed Controls (Control Group):** This is *essential*. You need a group that *doesn't* receive the new antibiotic. This allows you to see what would happen naturally under the same conditions. The choice of the negative control depends on your experimental setup – could be a standard culture, or a preparation mimicking the biofilm environment.
- **Control with a Known Standard:** Include a control group treated with a known standard antibiotic (e.g., a known antibiotic that inhibits biofilm formation) to demonstrate that the new antibiotic *doesn't* have a similar effect. This strengthens your claim about efficacy.
- **Untreated Control:** A truly negative control – no antibiotic administered. This is key to demonstrate that the observed effects are due to the antibiotic, not something else.

2. Positive Controls – Validating the Experimental Setup

- Positive Control – Established Standard: A known biofilm-forming bacterium (*e.g.*, *E. coli*)

|Send a message

+ gemma3:1b ↑

Gemma3-1B response

Beyond the Browser

IDEs and LLM plugins

Most of the popular IDEs will have plugins for the main LLMs like **ChatGPT**, **Claude**, **Gemini**, etc.

For example, Visual Studio Code has the **OpenAI Codex plugin** which allows you to use ChatGPT to write/edit/fix/check your code or codebase.

They can usually be used in a **supervised** (checking each suggestion) or **unsupervised** (completely hands-off) mode.

Add your credentials to the plugin to link it to ChatGPT.

The screenshot shows the Visual Studio Code interface with the OpenAI Codex plugin installed. The main editor window displays a Python script named `codex-example.py`. The code defines a function `sphere_volume_mass` that calculates the volume and mass of a sphere given its diameter and density. It includes error handling for negative inputs and command-line argument parsing. The status bar at the bottom indicates the file has been changed (+37 -1).

```
import math
from typing import Tuple

def sphere_volume_mass(diameter: float, density: float) -> Tuple[float, float]:
    """Return (volume, mass) of a sphere given diameter and density.

    - diameter: length units
    - density: mass per cubic length unit

    Raises ValueError for negative inputs.
    """
    if diameter < 0:
        raise ValueError("diameter must be non-negative")
    if density < 0:
        raise ValueError("density must be non-negative")
    r = diameter / 2.0
    volume = (4.0 / 3.0) * math.pi * r**3
    mass = density * volume
    return volume, mass

if __name__ == "__main__":
    import sys
    try:
        diameter = float(sys.argv[1]) if len(sys.argv) > 1 else 10
        density = float(sys.argv[2]) if len(sys.argv) > 2 else 1000
    except ValueError:
        print("Usage: python codex-example.py [diameter] [density]")
        sys.exit(1)

    vol, m = sphere_volume_mass(diameter, density)
    print(f"Diameter: {diameter}")
    print(f"Volume: {vol}")
    print(f"Mass: {m}
```

The Chat panel on the right shows a conversation with the AI. The user asks for a Python function to calculate the volume and mass of a sphere. The AI responds by creating a short TODO plan and providing a concise function and usage example. The AI's progress is tracked with a checklist.

CHAT

← PYTHON FUNCTION FOR SPHERE VO...

Can you write me a Python function that calculates the volume and mass of a sphere given its diameter and its density?

✓ Optimizing tool selection...

✓ Refined final code output and implemented function updates

I'll create a short TODO plan for this task.

✓ Executed manage_todo_list

Progress: Plan created — now writing the Python function.

Here's a concise function and a small usage example:

Todos (1/2)

- Add `sphere_volume_mass()` to `codex-example.py`
- Run `codex-example.py` to verify output

> 1 file changed +37 -1 Keep Undo

Describe what to build next

Ln 37, Col 1 Spaces: 4 UTF-8 {} Python 3.14.0

Visual Studio Code OpenAI Codex plugin

LLM-centric IDEs

Some IDEs are centred around LLMs, and designed specifically to use them.

One of the best-known is **Cursor**, a fork of Visual Studio Code, but is a paid-for option.

Cursor is a bit more clever about picking which LLM to use for what, and uses multiple different LLMs under the hood.

OpenAI now have their own standalone **Codex IDE**.

There are others like **Windsurf**.

The screenshot shows the Cursor IDE interface with a Python file named `codex-example.py` open. The code defines two functions: `sphere_volume_mass` and `cylinder_volume_mass`. The `sphere_volume_mass` function takes diameter and density as parameters and returns a tuple of volume and mass. It includes validation for non-negative inputs. The `cylinder_volume_mass` function adds height as a parameter and follows a similar structure. A sidebar on the right provides context and completion suggestions for adding a cylinder volume and mass calculation.

```
import math
from typing import Tuple

def sphere_volume_mass(diameter: float, density: float) -> Tuple[float, float]:
    """Return (volume, mass) of a sphere given diameter and density.

    - diameter: length units
    - density: mass per cubic length unit

    Raises ValueError for negative inputs.
    """
    if diameter < 0:
        raise ValueError("diameter must be non-negative")
    if density < 0:
        raise ValueError("density must be non-negative")
    r = diameter / 2.0
    volume = (4.0 / 3.0) * math.pi * r**3
    mass = density * volume
    return volume, mass

def cylinder_volume_mass(diameter: float, height: float, density: float) -> Tuple[float, float]:
    """Return (volume, mass) of a cylinder given diameter, height, and density.

    - diameter: length units
    - height: length units
    - density: mass per cubic length unit

    Raises ValueError for negative inputs.
    """
    if diameter < 0:
        raise ValueError("diameter must be non-negative")
```

Cursor IDE

Command Line Tools

There are command line (terminal) tools available to use LLMs in your terminal.

These usually have terminal UIs (TUIs) which behave much like the web, standalone, and IDE plugins: you type commands and the LLM either replies and/or acts on your behalf.

These often use optimised models specifically suited to coding.

Claude Code is probably the best-known of these.

OpenCode is increasingly popular, open-source, and lets you use multiple LLMs.

OC | Cube volume and mass calculation func...

```
# Cube volume and mass calculation function in codex-example.py          9,942 2% ($0.00) v1.1.53

Add a function to the codex-example.py file to calculate the volume and mass of a cube given the length of a side and the density
```

* Glob "**/*" in . (1 match)
→ Read codex-example.py

```
← Patched Users/ajm513/Desktop/codex-example/codex-example.py
```

39 mass = density * volume 40 return volume, mass 41 42	39 mass = density * volume 40 return volume, mass 41 42 43 + def cube_volume_mass(side_length: float, density: float) → Tuple[float, float]: 44 + """Return (volume, mass) of a cube given side length and density. 45 + 46 + - side_length: length units 47 + - density: mass per cubic length unit 48 +
--	--

```
Build GPT-5.3 Codex OpenAI
```

ctrl+t variants tab agents ctrl+p commands

Benefits of LLMs in the Terminal

- More easily scriptable
- More portable
- Can run on remote, headless machines
- Can be customised, e.g. Claude Code Skills, Hooks, MCP (Model Context Protocol)
- Many programmers and data scientists now work exclusively in the terminal

LLM APIs

APIs (Application Programming Interfaces) are completely programmatic ways of interacting with software and services.

The most popular LLMs like ChatGPT and Claude have APIs that you can use, to integrate their LLMs into your own code or apps.

These are *very flexible* and powerful, but can be **expensive** if you are making lots of API calls!

Essentially *pay as you go*

Using LLMs in Research



Use cases for LLMs in research

Things to Bear in Mind

Always check your University/Institution's policies on using LLMs

For example, the University of York mandates use of Google Gemini, and it is set up specifically not to collect/store data

Check any journals' policies on use of AI, and acknowledging use of AI

ALWAYS SANITY CHECK ANSWERS/RESULTS FROM LLMs!

Question:

*Thinking about your recent
research, how could you have used
an LLM to help with it?*

*If you've used one recently, what
was the use case? Did it work well?*