

Lecture 3

Generative AI in Economics

Pierre Biscaye
Université Clermont Auvergne

Data Science for Economics

Note: Materials for this lecture are drawn partly from Anton Korinek's Generative AI for Economics [website](#) and materials

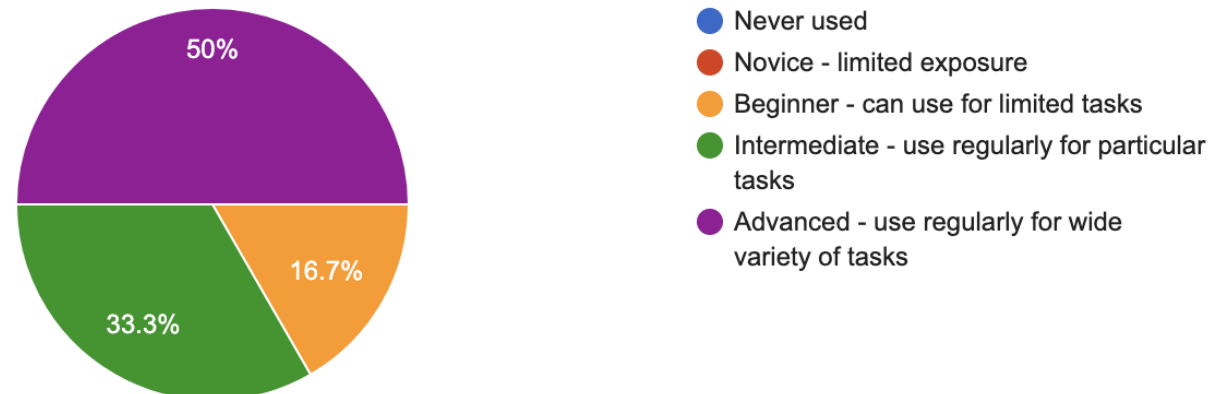
Agenda

1. What is generative AI?
2. Large Language Models/Chatbots
3. Applications of generative AI in economics
 1. Emphasis on Chatbots

What is your level of experience with generative AI like ChatGPT?

12 responses

 [Copy chart](#)



1. What is Generative AI?

Generative AI refers to artificial intelligence models designed to create new content, such as text, images, audio, or code, by learning patterns and structures from existing data.

- How It Works:
 - Trains on large datasets to understand and mimic data patterns.
 - Generates outputs based on prompts or input data, often using advanced neural networks like transformers.
- Applications:
 - Text generation, image synthesis, coding assistance, and more.

Examples of current Generative AI tools (changing quickly!)

- **Text Generation:**

- OpenAI's ChatGPT, GPT-4
- Google Bard
- Anthropic's Claude

- **Image and Video Generation:**

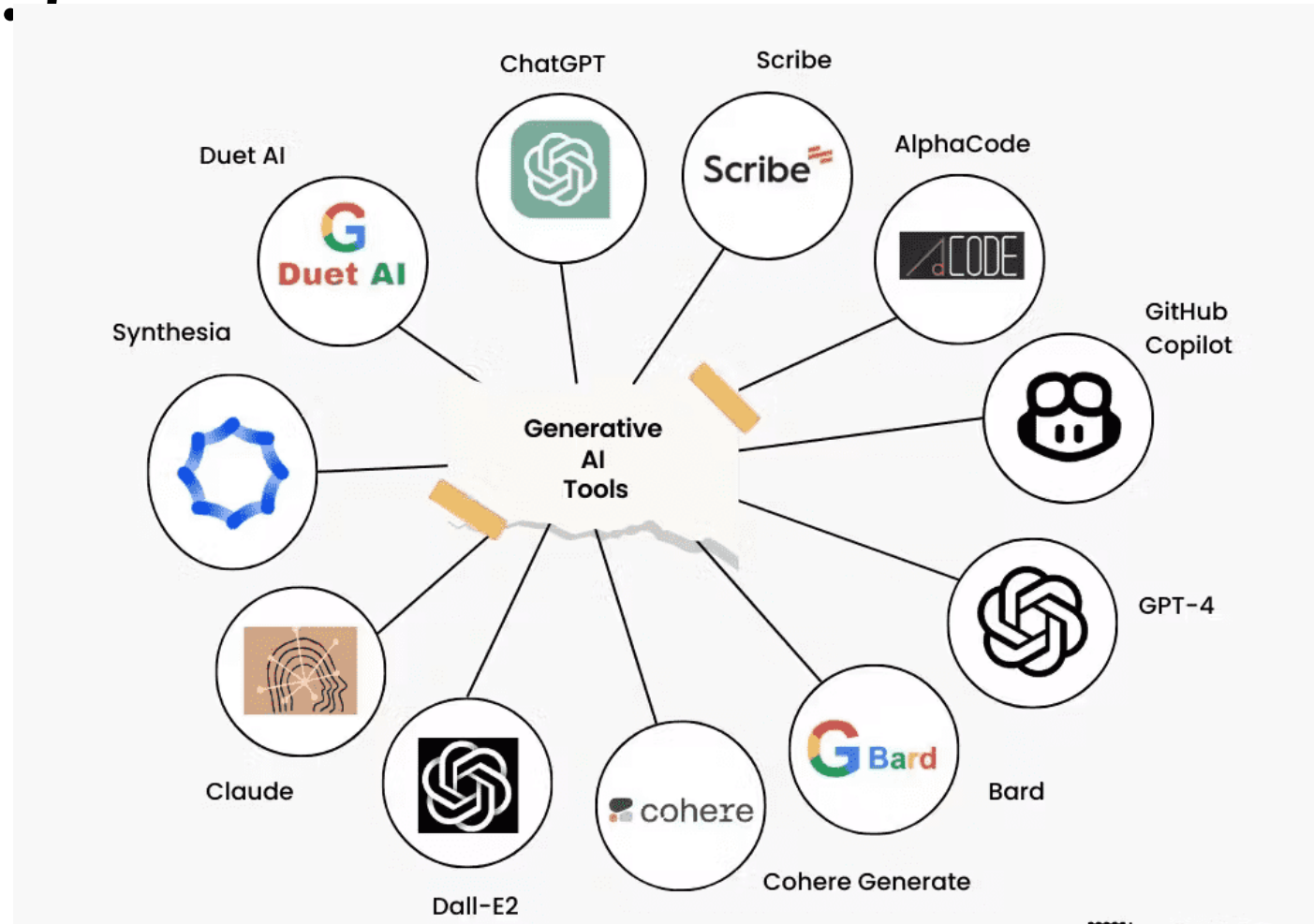
- DALL·E, Sora (OpenAI)
- MidJourney
- RunwayML (for video)

- **Code Assistance:**

- GitHub Copilot
- Tabnine
- Claude Code

- **Audio and Music Generation:**

- OpenAI's Jukebox
- AIVA (AI music composition)



[Source](#)

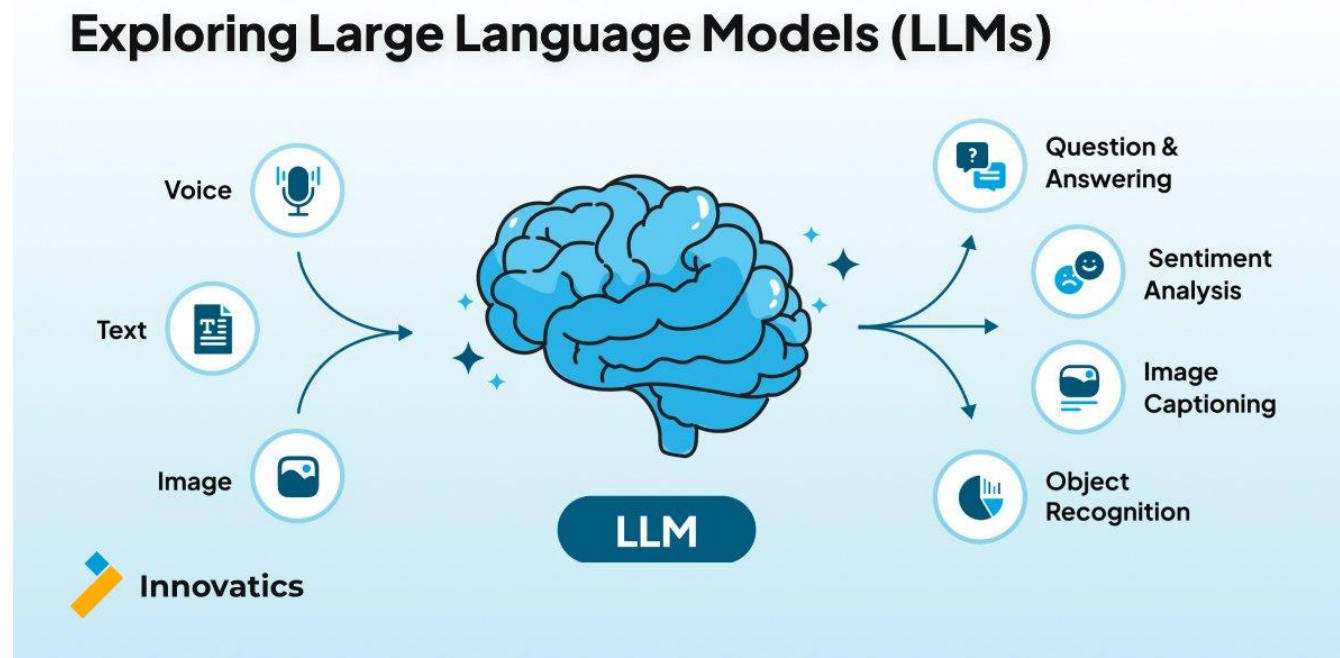
Many applications for researchers

- Writing code
- Editing writing
- Generating hypotheses
- Background research
- Summarizing and repackaging results
- Producing podcasts
- And more!

Focus today on use of text generation tools: LLMs.

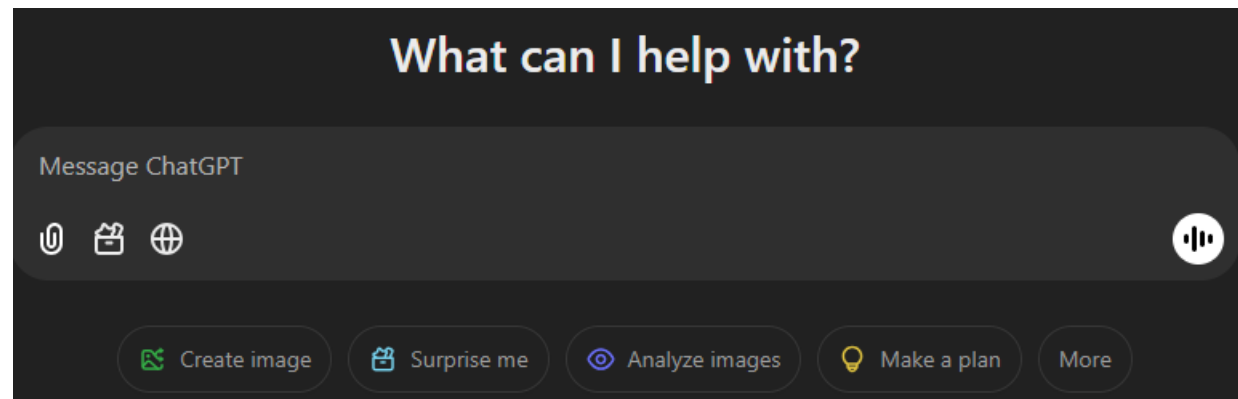
2. Large Language Models (LLMs)

- LLMs are advanced AI systems trained on vast amounts of text data to understand, generate, and interact in natural language.
- Examples: ChatGPT (OpenAI), Claude (Anthropic), LLaMA (Meta), Gemini (Google)



Main ways to access LLMs

- Chatbots (main way)
- Interactive voice assistant (app on phone or desktop)
- Interactive workspaces (Canvas, Artifact, Copilot)
- Web-based experimentation platforms



Getting started with Chatbots

- Sign up a for an account
 - Free account gives access to basic models with limits on uploadsaccess
 - Paid account gives access to latest models and other tools and plug-ins; around \$20/month
 - For example, ChatGPT Advanced Data Analysis
 - ChatGPT Plus: upload files and perform data processing tasks on them
- Start chatting
 - Different chatbots can take text, voice, images, and files as inputs

Using Chatbots effectively

- **Refine your prompts** to get the best output.
 - Garbage in -> garbage out
- **Repetition** is a powerful way to achieve the desired output.
 - Explain like you would to an overeager but skilled brand-new research assistant.
- **Provide feedback**/ask the chatbot to change something – it can refer to previous messages.
 - Tell it that it did something wrong, ask it to explain something to you.
- Make sure can **understand the output**. It is *your responsibility* to check for accuracy.

What are some of the risks of heavy
Chatbot use?

Being careful with Chatbots

- **Security:**
 - Privacy and consent issues may arise from chatbot interactions.
 - *Solution:* Be careful about any information you share and consider whether there are implications of making that information public.
- **Data accuracy/hallucination:**
 - Chatbots may provide outdated or incorrect information.
 - *Solution:* Cross-check data with authoritative sources and verify claims.
 - Chatbots may misinterpret complex questions or context.
 - *Solution:* Provide clear, specific queries, and verify chatbot responses in context.
- **Biases:**
 - Chatbots can inherit biases from their training data.
 - *Solution:* Evaluate responses critically, using diverse perspectives and external validation.

Efficient research vs outsourcing thinking

Risks of over-reliance

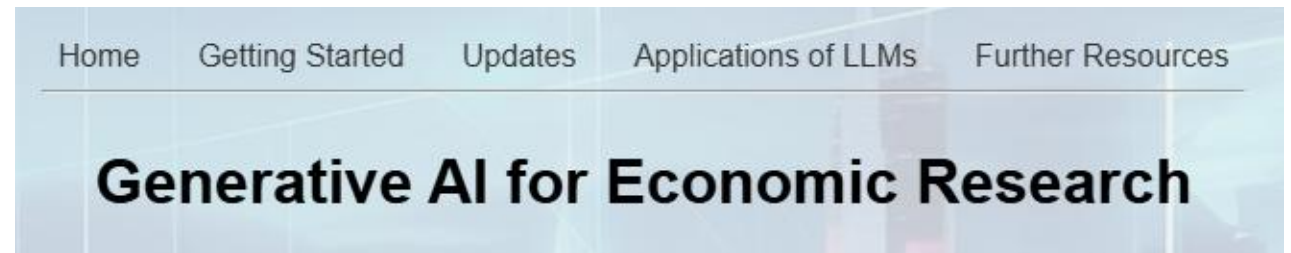
- The **illusion of efficiency**: Debugging AI errors can often take longer than writing the code from scratch, especially if you are not capable of giving clear instructions and feedback.
- **Erosion of problem-solving**: If AI is always the first response, it can be hard to troubleshoot when you encounter unique edge cases in real-world data.
- The **Black Box effect**: You may produce a correct result without understanding why a specific approach was the right choice, leading to an inability to defend or explain your methods.
- **Stagnant learning curve**: Relying on AI for basic syntax prevents "muscle memory" development, making it difficult to pass technical interviews or live-coding assessments.

Using AI to complement, not replace

- The **sounding board**: Use AI to explain complex concepts (e.g., "Explain the difference between a Left Join and an Inner Join in simple terms") rather than just asking for the code.
- **Syntax reference**, not full code: Use AI to look up specific function arguments or to flesh out your ideas, but design the structure of the code yourself.
 - If you can't come up with the structure/outline of what needs to happen, you don't actually know what you are trying to do.
- **Code documentation & cleanup**: Write your own logic first, then ask the AI to suggest comments or identify ways to make your code more "Pythonic" and readable.
- **Struggle first**: Try to work through a coding challenge on your own first before seeking AI assistance.
 - Build up your skills and intuition!

3. Generative AI for Economic Research

- Anton Korinek published [paper](#) in JEL in 2023
- Publishing regular [updates](#)
- Website hosts introduction to using generative AI for economists
- Proposes 6 general **categories of capabilities**
 1. Ideation and feedback
 2. Writing
 3. Background research
 4. Coding
 5. Data analysis
 6. Math



[Source](#)



3.1. Ideation and feedback

- Ask general questions for brainstorming
- Can feed in text or a paper and ask for suggestions to improve it
- Evaluate pros and cons of hypotheses or research plans
- Provide counterarguments

Iteration is critical

- Engage in discussion with the chatbot to follow up and refine ideas.

Sample ideation questions

- How can economic growth increase conflict?
- How does climate change affect economic development in Africa?
- What are the challenges in conducting research on climate change and economic development?

Any thoughts about these prompts?

More refined ideation questions -> More useful output

- Can you brainstorm 10 channels through which economic growth may increase conflict? Limit your response to 10 words for each point.
 - Follow-up: For which channels is there more empirical evidence? For which channels is there little evidence?
- I am an economist working on climate change and economic development in rural Africa. Can you brainstorm 5 potential paper topics and describe each in one sentence?
 - My particular skills include using applied microeconomic theory, econometric tools, and remote sensing. Can you come up with how I could work on each of the topics you proposed using these tools, in a rough outline of 2-3 paragraphs per topic? Please note potential data sources and econometric approaches I would use.



3.2. Writing

- Generating catchy titles/headlines, writing abstracts, writing a blog, writing a tweet
- Write a paragraph based on an outline/suggestions
- Editing/grammar, style advice
- Rewrite text in a way accessible to a different audience (eg non-economists)
- Generate slide content (including in Latex)

Example: Paper on locusts and conflict

- Upload current draft of paper
- Prompts:
 - Based on the introduction, can you generate a 100 word abstract that would be suitable to a top economics journal?
 - Can you summarize the main contributions of the paper in 4 bullet points?
 - Can you generate a Latex slide based on your last response?
 - Could you draft a paragraph summarizing contributions of the paper to a literature on econometrics by showing the sensitivity of estimate impacts using panel data depending on assumptions about whether treatment impacts are transitory or persistent?
 - Can you suggest ways to improve this paper and make it more suitable for a top economics journal?

3.3. Background research

- Summarizing text/papers
 - Adept at extracting most relevant content from a passage.
- Researching literature
 - Limited (but improving) usefulness; beware hallucinations
 - Abstract representation of training data involves loss of information
- Formatting references
- Translating
- Explaining concepts
 - Beware mistakes!



Example: Project on flood mapping

- Using bullet points, what are the most common ways flood incidence is measured in economics papers, and what are some citations of published papers using each method?
 - Are the papers real?
- Upload Patel (2024); In 3 bullet points, what are the main takeaways from this paper for how researchers should approach measuring flood incidence?
- In 3 bullet points, can you explain the main differences between identifying flood incidence using satellite imagery and satellite radar?

AI and lit reviews (Bolanos et al 2024)

- Many existing lit review tools
 - Powerful tools for screening and extracting content from literature
- Limitations
 - “Fall short on usability and user-friendliness”
 - Many depend on outdated AI methodologies
 - Limited transparency of methods
 - Inconsistent outputs, hallucinations
- Emerging AI tools
 - Bibliographic search engine chatbots (Scopus, Dimensions, CORE)
 - Search engine tools: EvidenceHunt, Scite, Scispace, Elicit, Textero, MirrorThink, Consensus, Perplexity
 - Different bibliographic databases indexed
 - Still in their infancy

Table 1 The 21 SLR tools analysed in this survey

ID	Tool	Stage SLR	Mode	OS	References
1	Abstrackr	Screening	Web	No	Wallace et al. (2012)
2	ASReview	Screening	Desktop	Yes	Van De Schoot et al. (2021)
3	Colandr	Screening	Web	Yes	Cheng et al. (2018), Cheng and Augustin (2021)
4	Covidence	Screening	Web	No	–
5	DistillerSR	Screening	Web	No	–
6	EPPI-Reviewer	Screening	Web	No	Thomas et al. (2010), Machine Learning Functionality in EPPI-Reviewer (2019)
7	FAST2	Screening	Web	Yes	Yu and Menzies (2019)
8	LitSuggest	Screening	Web	No	Allot et al. (2021)
9	Nested Knowledge	Screening	Web	No	–
10	PICOPortal	Screening	Web	No	Agai (2020), Minion et al. (2021)
11	Pitts.ai	Screening	Web	No	–
12	Rayyan	Screening	Web	No	Ouzzani et al. (2016)
13	Research Screener	Screening	Web	No	Chai et al. (2021)
14	RobotAnalyst	Screening	Web	No	Przybyla et al. (2018)
15	SWIFT-Active Screener	Screening	Web	No	Howard et al. (2020)
16	SWIFT-Review	Screening	Desktop	No	Howard et al. (2016)
17	SysRev	Screening	Web	No	Bozada et al. (2021)
18	Dextr	Extraction	Web	No	Walker et al. (2022)
19	ExaCT	Extraction	Web	No	Kiritchenko et al. (2010)
20	Iris.ai	Both	Web	No	–
21	RobotReviewer/RobotSearch	Both	Web	Yes	Marshall et al. (2017, 2018)

OS open source

3.4. Coding

- Explain coding concepts
- Provide code to accomplish tasks
- Identify errors in code
- Convert code across languages

Probably the main use case for this course.



[Source](#)

Example coding questions

- How do I generate summary statistics for a given variable 'var1' in a data frame 'df' in Python? Can I turn this into a nicely formatted table for export to Latex?
- Can you refine the describe() method to include the 90th and 99th percentiles?
- I wrote this code and am getting an error. Can you help me identify it? `summary_stats = df['var1'].describe(percentiles=[0.1, 0.5, 0.9, 0.99])`
 - Can also copy in text of error messages
- How would I write that code in Stata?
- What does the following function do? [Code]

Coding assistants

- Example: Claude Code - integrated directly into your codebase.

Code onboarding

Claude Code maps and explains entire codebases in a few seconds. It uses agentic search to understand project structure and dependencies without you having to manually select context files.

Turn issues into PRs

Stop bouncing between tools. Claude Code integrates with GitHub, GitLab, and your command line tools to handle the entire workflow—reading issues, writing code, running tests, and submitting PRs—all from your terminal.

Make powerful edits

Claude Code's understanding of your codebase and dependencies enables it to make powerful, multi-file edits that work.



Arthur Spirling

@arthur_spirling



Asked Claude Code to “help me make the most significant contribution I can to human knowledge. Take a look at what I’m working on and take action”

Took an hour but it...

- deleted all my files
- emailed all my coauthors apologizing
- locked me out of my PC

Very impressive

8:14 PM · Jan 4, 2026 · **69.9K** Views

Vibe coding

- The original idea: Type natural language commands instead of code, accept all AI suggestions *without reading them*, copy-paste error messages until things work.
- Vibe coding vs. AI-assisted programming
 - Difference between careful, supervised use and “accept all” approach
- Risks of “flying blind”
 - Code that works... until it doesn't
 - Solutions that are almost right... but not quite
 - Introduces subtle bugs that take hours to debug
- As a student: undermines your ability to learn to code

[Source](#)

Vibe Coding vs Real Coding

	VIBE CODING	REAL CODING
Method	Guesswork, AI prompts, or copying code	Logic, planning, understanding
Speed	Quick to start, slow/little long-term use	Slow to start, faster and reliable long-term use
Understanding	Often unclear	Clear and deliberate
Usefulness	Good for prototyping or experimenting	Best for complex and maintainable software
Risk	Higher risk: bugs, security issues, little scalability	Lower risk: more stable, easier to debug and build on

[Source](#)



Jeffrey Cisyk
@jeff_cisyk

The only two true outcomes of vibe-coding...



LLM: Oracle vs overconfident intern

- Shift your mindset: LLMs can be wrong and fail in subtle ways
 - Repeating the same prompt different times can give different results -> not giving `answers` but statistical guesses
- Hallucinations: invent a plausible sounding method like *df.summarize_distribution()*
 - Usually throws an error -> you know it's wrong
- Silent killers: correct syntax but scientifically wrong
 - For example, non-vectorized loops that work on small sample but crash on full data, or applying a method to full dataset before splitting into train/test in ML exercise (leads to incorrect accuracy)
 - You may not detect these issues from just looking at the code or running it at first.

Example: RA code vs my code

- Task: scrape daily rasters of flood incidence from the VIIRS flood mapping archive and process into raster of annual days of flooding by year
- RA approach: basically vibe coding
 - Unable to explain what the code does or its structure
- My approach:
 - Break down into logical steps, outline the structure
 - Ask LLM for help writing code for each step
 - Test with subsample
 - Iterate with feedback
 - Target simple, clear, and efficient code
- Let's compare!

3.5. Data analysis

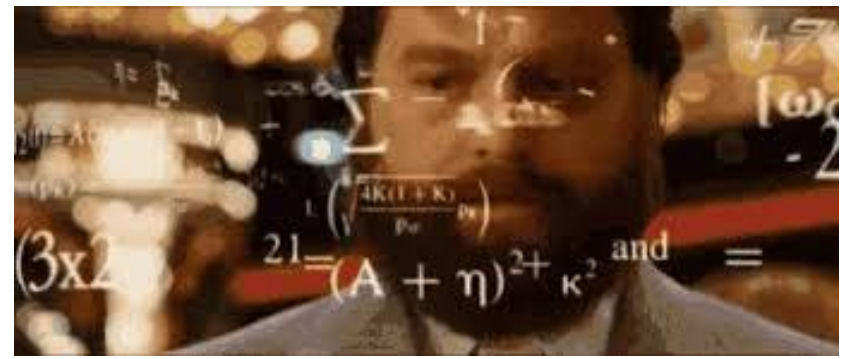
- Create figures
 - Particularly useful with paid subscriptions
 - E.g., ChatGPT Advanced Data Analysis
- Extract data from text
- Classifying text, extracting sentiment
- Suggest sources



Example prompts

- Paul got an 10 in econ and a 12 in math. Marie got a 13 in both econ and math. Eloise got a 15 in econ and an 11 in math. Based on this information, can you generate code to create a dataframe with information on student name, econ grade, and math grade in python?
- Can you rewrite that as a Latex table?
- Is the following FOMC statement from December 2024 hawkish or dovish?
<https://www.federalreserve.gov/newsevents/pressreleases/monetary20241218a.htm>
- What data sources could I use to analyze electricity access in Senegal?

3.6. Math



- Derivations, integrals, etc.
 - Beware mistakes
- Turning words into equations
- Explaining models
- Converting images of equations or tables to Latex
- Examples [here](#)

Conclusion: Yes, do use Generative AI

- It can be quicker and easier to get a relevant response than Googling or using built in help or documentation.
 - But the effort of Googling can also help with learning.
- You can iterate to dig deeper or refine the response to help you solve your issue.
- It can deepen your understanding and help you troubleshoot.

Don't rely exclusively on Chatbots for coding (or anything)

- Try and code something yourself first.
 - Refer to and adapt the code from class, or Google for examples.
- If you are stuck somewhere, think hard about the problem.
 - Break down the structure and components, isolate aspects to work on and troubleshoot.
- **Don't** blindly copy and paste output!!
- **Do** carefully read the output, try to understand it, and ask follow-up questions about anything you don't understand.
 - Think: is there something unclear? Something that seems inefficient?

Chatbots should *complement* but not *replace* your own knowledge and skills.

BREAK

- Then, wrap up Python basics.