AI for the Scientist in a Hurry

or "Running Notes and Reflections About AI"

George G. Vega Yon, Ph.D.

2026-02-18

Table of contents

1 Preface

> **AI version**
>
> This version of the chapter hasn't been edited by a human. It was generated with the help of AI by providing the core ideas and asking the AI to help with writing and editing. Once this chapter is reviewed by a human, this note will be removed.

The idea for this book emerged gradually, shaped by multiple discussions and recent talks I have given in a variety of settings. These conversations have taken place in conference halls, internal institutional meetings, invited lectures, and informal chats with friends. They have also unfolded in my classroom, where I teach advanced programming, and during my time working at the CDC, where AI is slowly being adopted into everyday workflows. Even at home, I find myself trying to explain to my children what AI is—and just as importantly, what it is not. Across all these spaces, the same themes, questions, and misunderstandings continue to surface, and this book is my attempt to bring some coherence to them.

With the current level of hype surrounding AI, I felt the need to create something that helps me—and hopefully others—stay grounded and up to date. There is an overwhelming amount of information about what these systems can do, but less clarity about what is actually useful, what constitutes a good example of meaningful application, and what mistakes we should be careful not to repeat. This book grows out of that need for orientation: a way to step back, take stock of what is out there, and think critically about how we engage with it.

This project is also closely tied to my personal experience. I consider myself an advocate and early adopter of AI, and I have been genuinely excited about its potential. At the same time, I have become increasingly concerned about some of the pitfalls—particularly how easily these tools can make us less careful thinkers if we rely on them uncritically. That tension troubles me. As a colleague once put it, the goal should be to have AI enhance us, not replace us. Part of this book is therefore reflective: an effort to articulate recommendations and principles that help ensure these tools strengthen, rather than erode, our intellectual habits.

Although I am an advocate for AI, I recognized early on that I am not an AI scientist in the sense of building foundational models or competing with the multibillion-dollar efforts of private companies developing them. That is not my aim. Instead, I see my role as someone who tries to think ahead about how best to use these tools—how to integrate them thoughtfully into scientific work, teaching, and daily problem-solving. My focus is not on building the models themselves, but on understanding how to work with them wisely.

Like the other books I am currently working on, this one will evolve over time. I am building it primarily as a resource for myself and for those who work closely with me—other scientists and young scholars navigating this rapidly changing landscape. Of course, I would be glad if it proves useful to a broader audience. But at its core, this book is an ongoing effort to think clearly about AI in practice: what it is, what it can do, and how we can use it responsibly and effectively.

1.1 Spanish Version Available

A Spanish translation of this book is available at es/. The Spanish version was created using AI translation and may require human review for technical accuracy.

1.2 About the Author

I am a Research Assistant Professor at the **University of Utah's Division of Epidemiology**, where I work on studying Complex Systems using Statistical Computing. I was born and raised in Chile. I have over fifteen years of experience developing scientific software focusing on high-performance computing, data visualization, and social network analysis. My training is in Public Policy (M.A. UAI, 2011), Economics (M.Sc. Caltech, 2015), and Biostatistics (Ph.D. USC, 2020).

I obtained my Ph.D. in Biostatistics under the supervision of **Prof. Paul Marjoram** and **Prof. Kayla de la Haye**, with my dissertation titled "*Essays on Bioinformatics and Social Network Analysis: Statistical and Computational Methods for Complex Systems.*"

If you'd like to learn more about me, please visit my website at https://ggvy.cl.

1.3 AI Disclosure

Since the beginning of this project, I have been using AI to help me write this book. Mainly, I use a combination of ChatGPT, GitHub co-pilot, which aids with code and text, and Grammarly, which aids with grammar and style. AI's role has been to help me write faster, including editing and proofreading, but the core conceptualization and ideas are my own. **Since I do use AI to assist in writing, I will be transparent about it indicating which pieces of the book are mostly AI-generated and unreviewed by me.**

2 Introduction

> **AI version**
>
> This version of the chapter hasn't been edited by a human. It was generated with the help of AI by providing the core ideas and asking the AI to help with writing and editing. Once this chapter is reviewed by a human, this note will be removed.

Artificial intelligence is here, and whether we welcome it or resist it, it is steadily weaving itself into nearly every aspect of our daily lives. That statement can sound alarmist to some, unsettling to others, and deeply encouraging to many. I prefer to treat it as something simpler: a fact. AI is no longer a futuristic concept or a niche tool used by a handful of specialists. It is becoming part of the infrastructure of how we work, write, code, search, communicate, and make decisions.

When most people think about AI today, they think about large language models—LLMs—and the ability to "chat" with systems like ChatGPT or Gemini. But AI is not just LLMs. It is a broader combination of methods, architectures, and systems. In my view, the most important recent innovation is not merely the possibility of conversing with an LLM. It is the development of agentic AI.

By agentic AI, I mean AI systems that are embedded in workflows or other software environments in ways that allow them to interact with other systems. These systems do not just respond to prompts; they can take actions, retrieve information, write files, modify code, and connect tools together. That shift—from a chatbot in a window to an agent embedded in a workflow—is, in my opinion, the real game changer.

It is through agentic AI that tools like ChatGPT, Gemini, and others are becoming genuinely useful beyond experimentation. A recent example of this shift is the purchase of OpenClaw by OpenAI, the company behind ChatGPT. OpenClaw represents what many of us had in mind when we first began imagining AI assistants: systems that could go beyond answering questions and instead carry out tasks on our behalf. Although I have not used OpenClaw directly—partly because it is still new and there are ongoing security concerns—I believe this type of product will have a significant impact across the economy.

Before speculating too much about what the future of AI might hold, it is worth clarifying the purpose of this book. As I mentioned in the preface, I see this project as a way to keep myself—and my collaborators—up to date with the evolving landscape of AI tools, as well as the practical dos and don'ts of using them responsibly and effectively.

With that goal in mind, the book is organized as follows.

Chapter One provides an overview of the most important concepts and how they connect to current products and innovations. It discusses what an LLM is, how these systems work, why context matters, and what the prevailing architectures behind many of these tools look like. It also introduces agentic AI and its relationship to Model Context Protocols (MCPs), along with references to other useful sources for readers who want to go deeper.

Chapter Two focuses on applications of AI as a writing assistant. It covers best practices, practical guidance around prompt engineering, and concrete examples of usage.

Chapter Three explores how to use AI for programming, with particular attention to AI agents integrated into tools such as VS Code, Positron, and RStudio (by Posit). This chapter also takes a deeper dive into context in AI and how to make the most of it in technical workflows.

Chapter Four is dedicated to agentic AI using Claude and GitHub Copilot, which, in my personal opinion, are among the most transformative tools currently available. This chapter includes live examples drawn from my own projects, including cases where things did not work as expected.

Chapter Five broadens the scope with additional examples of AI in other contexts, including how to leverage AI to evaluate your own writing—for example, reviewing something as high-stakes as an NIH grant.

Finally, **Chapter Six** introduces more advanced applications, focusing on how to use AI programmatically through tools such as Ollama and other open-source frameworks.

This structure reflects both the rapid evolution of the field and my own ongoing effort to understand it. The goal is not to predict the future, but to engage with the present—carefully, critically, and practically.

3 Overview of AI

AI version

This version of the chapter hasn't been edited by a human. It was generated with the help of AI by providing the core ideas and asking the AI to help with writing and editing. Once this chapter is reviewed by a human, this note will be removed.

Artificial Intelligence is a broad and, at times, overloaded term. From a technical perspective, AI refers to computational systems that perform tasks that resemble decision-making. Under this definition, AI is not limited to humanoid robots or conversational systems. A regression model that estimates the probability of an event is AI. A neural network that classifies an image as a dog or a cat is AI. And what most people now associate with AI—"smart chats" that generate text responses—is also AI. The difference lies not in whether they are intelligent, but in how they are built and what they are designed to do.

In recent years, however, one particular family of models has dominated the conversation: Large Language Models.

3.1 Large Language Models

The most visible breakthrough in contemporary AI has come from Large Language Models (LLMs). These are probabilistic models built on a specific neural network architecture—transformers—and trained on extremely large datasets. The core principle behind them is conceptually simple: given a sequence of text `x`, what is the most likely next word?

This next-word prediction task, repeated billions of times during training, leads to surprisingly powerful behavior. By learning statistical patterns in language at scale, LLMs acquire the ability to summarize, translate, write code, answer questions, and simulate reasoning. Importantly, they are not retrieving fixed answers from a database. They are generating responses token by token, sampling from probability distributions conditioned on the input.

Not all LLMs are equal. Beyond differences in training data and model size, one major distinction is how much *context* they can handle. Context refers to how much information—previous conversation, documents, instructions—the model can take into account at once when generating a response.

3.1.1 The Role of Context

The quality of an LLM's output depends critically on the input it receives. Think of it this way: how much would you expect a complete stranger to correctly answer a technical question if you provide almost no background information? The same logic applies to LLMs. The more relevant context you provide, the better the response is likely to be.

This is why users are often advised to specify roles or constraints. For example:

> "You are an expert statistician specializing in randomized controlled trials."

By doing this, you are shaping the distribution of possible answers. Without such guidance, the model's response will reflect a blend of patterns learned across its entire training data—an average over many domains and styles.

Another useful way to think about LLMs is through a Bayesian analogy. Imagine that an LLM has an extremely broad and fuzzy prior—essentially encompassing large portions of recorded human knowledge. When you provide a prompt, you are conditioning on new evidence. If your prompt is vague, the posterior distribution remains diffuse, and the model will produce a generic, "average" response. If you provide specific context, you narrow the posterior and constrain the sampling space.

This raises a natural question: how much context should you provide?

The answer depends on your goal. If you want precision, provide as much relevant context as possible. The narrower the conditioning information, the more targeted and reliable the output tends to be. However, if your objective is creativity—using the LLM as a sounding board for new ideas—over-constraining the model can reduce novelty. In Bayesian terms, an overly tight likelihood function limits exploration. A slightly broader search space can produce more unexpected, and sometimes more interesting, results.

Understanding this trade-off between precision and creativity is essential to using LLMs effectively.

3.2 Agentic AI

While LLMs transformed how we interact with AI, the next major shift has come from what is often called *Agentic AI*. If LLMs made AI conversational, agentic systems are making AI actionable.

At the beginning of the recent AI surge, many users turned to models like ChatGPT for coding help. The results were mixed at best. Unless the task was very simple, the generated code often failed—sometimes syntactically, sometimes logically, and occasionally in more subtle ways. Worse, the models would confidently invent functionality that did not exist.

I experienced this directly. A user of the `netdiffuseR` R package once wrote to me saying that ChatGPT had informed them of an algorithm published by certain authors and claimed that my package included a vignette demonstrating it. The user could not find it. The reason was simple: it did not exist. The model had fabricated a plausible but false reference.

This behavior—often referred to as "hallucination"—was a fundamental limitation of early LLM deployments. The model could generate text that *sounded* correct, but it had no mechanism to verify its claims.

3.2.1 From Static Models to Acting Agents

With the introduction of frameworks such as Anthropic's Model Context Protocol (MCP), LLMs gained the ability to move beyond the chat window. Instead of merely proposing code, an agent could execute it, observe the result, and iterate if it failed. The idea is straightforward but powerful: give the model a tool to test its own output.

If the code does not run, the agent receives feedback and revises it. This loop—propose, execute, evaluate, revise—dramatically improves performance. In practice, this transformed coding assistants from occasionally helpful text generators into tools that can produce working scripts.

The same principle applies beyond coding. Another major advance came with enabling models to search the web. Before tool integration, LLMs were constrained to their training data. Once connected to search and browsing tools via MCPs, they could retrieve up-to-date information.

From a research perspective, this was transformative. When asking about recent papers, models can now return actual manuscripts rather than fabricated citations. That said, a caveat remains: while the paper itself may be real, summaries or interpretations can still be inaccurate. Tool access reduces hallucination but does not eliminate the need for verification.

3.2.2 Interacting with the World

The evolution of agentic AI has continued with systems that can interact directly with local environments. Open-source projects such as OpenClaw extended the MCP idea further by allowing agents to run locally rather than exclusively in the cloud. Through mechanisms like defined "agent skills" (e.g., `skills.md`), these systems provide structured ways for models to interact with email, calendars, files, and other computer resources.

This shift—from passive text generation to active system interaction—marks a qualitative change. AI systems are no longer confined to answering questions; they can now perform tasks, coordinate workflows, and integrate with personal and professional environments.

In summary, today's AI landscape is shaped by two intertwined developments: Large Language Models that generate remarkably fluent and flexible outputs, and agentic frameworks that allow those models to act, test, retrieve, and iterate. Understanding both is essential. LLMs explain why AI can speak. Agentic systems explain why AI can now do.

Together, they define the current frontier of artificial intelligence.

4 AI as a Writing Assistant

> **AI version**
>
> This version of the chapter hasn't been edited by a human. It was generated with the help of AI by providing the core ideas and asking the AI to help with writing and editing. Once this chapter is reviewed by a human, this note will be removed.

4.1 Creating Original Text

One of the most widespread uses of AI today is writing. For better or worse, AI-generated text is now everywhere—even in chapters like this one. The real question is not whether AI was involved, but how much it influenced the writing. Did the AI generate an entire chapter from a prompt such as "write me a chapter about writing with AI"? Or did it simply organize ideas that were already clearly formulated by the author, as I am attempting to do here?

> **Note**
>
> You can see the original discussion I had with the AI to generate this chapter here: https://chatgpt.com/share/6996ba6a-6eb8-800d-a408-36ea28620eed. You can also see it in 02-writing-assistant-draft.md.

Large language models (LLMs) are designed to generate text. That is their core function. The quality of that text, however, is a separate matter. Over time, the grammar and fluency of AI-generated writing have improved significantly. What has not improved to the same degree is the reliability of the content. AI systems can still produce statements that are inaccurate or misleading. For that reason, it is essential that users carefully review anything an AI produces. The responsibility for accuracy remains with the author.

Personally, I believe it is acceptable to use AI for writing anything. The ethical boundary, however, lies in authorship. The core ideas, the intellectual contribution, and the main arguments should originate from the human author. In addition, transparency about the role of AI in the writing process is important. This naturally leads to a practical concern: how can we ensure that AI is not generating new ideas on our behalf?

In my experience, the solution is surprisingly simple. You explicitly instruct the AI not to create new ideas. For every text-generation task in which I use AI, I ask it to organize my ideas rather than expand upon them. This usually works well. I also begin the interaction by providing context and often end my prompt with a clear instruction such as, "Don't do anything just yet." This framing helps establish boundaries before the AI starts generating text.

Problems tend to arise when users provide only a short, vague prompt. Ironically, those cases are often the easiest to detect. Text generated from minimal supervision frequently shares recognizable characteristics. It tends to be overly long, as AI systems attempt to integrate across multiple domains and err on the side of verbosity. When not written in a discursive style, it may rely excessively on bullet points. It may include emojis or unusual formatting. It often adopts an overly polished or excessively polite tone. When clear guardrails are set—when the AI is told precisely what it can and cannot do—these symptoms are largely absent.

4.2 Reviewing Text

Beyond drafting, AI can be extremely useful for reviewing text. For grammar specifically, my personal preference is Grammarly, a company based in Ukraine that provides an AI-powered writing assistant integrated into desktop and mobile platforms. Whether you are a native English speaker or an ESL writer, there is something to learn from it. I once discussed this with my PhD committee chair, Dr. Paul Marjoram. Paul, who was born and raised in the UK, already had excellent writing. Yet when he passed a short piece of text through Grammarly, the system identified improvements that even he found impressive.

Context is always critical when using AI for review. Grammarly allows you to specify the intended audience and purpose of your text. With systems like ChatGPT, this context is provided through natural language. The more explicit you are about your goals, the better the output tends to be. For example, you might write: "The following text is oriented for a scientific journal in the field of social networks. You are an expert in exponential-family random graph models." Clear instructions shape the quality and relevance of the feedback.

When reviewing longer texts, there are different strategies. In my own work, I typically submit paragraphs or sections rather than an entire document at once. This keeps me in control of the revision process and limits the scope of potential edits. In some cases, however—such as grant proposals—it may be appropriate to provide the full set of materials. I have even asked AI to score a grant proposal. When preparing an NIH R21 grant (currently under review), I first asked three colleagues to provide feedback. After receiving their comments, I submitted the same material to ChatGPT, along with context about the funding call. I instructed it to assume the role of an NIH review panel and to be objective rather than polite. The strengths and weaknesses it identified were the same as those independently noted by my colleagues.

Another particularly effective use of large language models is summarization. Although sometimes overlooked, this function can serve as a test of whether the AI truly understands what it is reviewing. Asking the system to provide a brief summary before offering feedback is, in my view, essential. This approach also applies to programming tasks, as will be discussed later. The key is again to be explicit: "Before you go on, please provide a summary of the text I am asking you to review." If the summary is accurate, you can proceed with greater confidence.

4.3 Bottom Line

Using AI to create text can be appropriate and helpful, provided it serves to organize and clarify your thoughts rather than replace them. With sufficient context and clear instructions, AI can also be a powerful tool for reviewing and summarizing written work. The responsibility, however, remains with the author to ensure originality, accuracy, and transparency.

5 AI for Programming

6 Agentic AI with Claude and GitHub Copilot

7 AI in Other Contexts

8 Advanced Applications

References

A News

A.1 Version 2026.02.18

First public version of *AI for the Scientist in a Hurry*. The book is still a work in progress, but the overall structure is in place. Current chapters include:

- **Introduction**: Motivation and scope of the book.
- **Overview of AI**: A conceptual introduction to Large Language Models and agentic AI, including the role of context and the shift from static models to acting agents.
- **AI as a Writing Assistant**: How to use LLMs to support scientific writing tasks.
- **Programming with AI**: Using AI coding assistants in environments like VS Code, Positron, and RStudio.
- **Agentic AI**: A deeper look at agentic frameworks and the Model Context Protocol (MCP).
- **Evaluating Writing**: Using AI to review high-stakes documents such as grant applications.
- **Programmatic AI Usage**: Working with AI tools programmatically, including Ollama and related libraries.

A Spanish translation is being developed in parallel under the `es/` directory.