The AI Bootcamp

19

Why AI Gets Numbers Wrong

In this chapter you'll learn

The difference between a number it computed and a number it guessed, and one rule for every figure someone else will see

The AI Bootcamp

THE PROBLEM

Most AI education is a pile of disconnected tips. A LinkedIn carousel here, a YouTube tutorial there, a $2,000 cohort course that goes out of date before the refund window closes. None of it is built to be read in order, so nothing compounds. You finish twelve videos and can recite twelve facts, not perform one new skill.

THE FIX

The AI Bootcamp is built the other way. 90 skills, packed 2 to 3 per chapter, 36 chapters, 3 tracks. Every chapter states what it needs from you and what it hands to the chapter after it, so skill 47 only works because skill 12 already landed. Read it in order and by the end you are not collecting facts about AI, you are running work through it.

LETTER FROM THE EDITOR

We built this because we kept meeting smart people who felt behind on AI for no reason other than nobody had organized the material for them.

This book does not assume you are technical, and it does not water anything down either. Every chapter earns its place: a real skill, a real exercise, a one-page companion you can flip back to in month three when you have forgotten the details but still need the answer.

If a chapter loses you, or a chapter changes how you work, I want to hear about it either way.

Kris

Editor, AI Central

Write to me at **editor@thecentral.ai**

THE 3 TRACKS

AI 101

Get running

Pick a tool, set it up, write a clear instruction, use it safely at work.

AI 202

Prompt properly

The framework behind every good instruction, and catching AI when it is wrong.

AI 303

Build without watching it

Connect AI to your tools, and automations that run while you are not looking.

IN THIS LESSON YOU'LL LEARN

By the end of this chapter you can tell whether your assistant computed a number or predicted it, make it compute on demand, read the two lines of its working that actually matter, and apply a single rule to every figure that leaves your desk.

Numbers are what you are held accountable for. A clumsy sentence in a board pack is embarrassing. A wrong figure is a different category of event, and this particular failure is unusually quiet: the wrong number arrives formatted, rounded sensibly, and surrounded by prose that is completely correct.

CONTENTS

19.1 There are two ways it can give you a number

19.2 Look for the working, not for a warning

19.3 The workbook it answered about is not always the one you attached

19.4 Any number someone else will see arrives with its working

TOPICS REQUIRED

Chapter 18. You should already be comfortable putting a long document in front of your assistant and pulling structure out of it without losing what matters. Bring two things: one real spreadsheet from your own work, and one number you have already sent to somebody.

AFTER THIS CHAPTER

Chapter 20 assumes you can tell the difference between a figure your assistant produced and a figure it retrieved, because research raises the identical question about sources.

19.1 There are two ways it can give you a number

It can predict the number, the way it predicts the next word of a sentence. Or it can write a small program, run it, and tell you what came back. Those are not two grades of the same skill.

Predicted

The number is generated the same way the sentence around it is: as the most plausible thing to say next.

TRAITS

· Appears instantly, inside the prose
· No code, no formula, no steps you can open
· Reads exactly as confident when wrong as when right

PROS

+ Fine for an estimate you are about to sanity-check
+ Fast enough to think out loud with

CONS

– Ask twice and you can get two answers
– Degrades sharply on long multiplication and multi-step sums

Computed

The assistant writes code, runs it in a sandbox, and reports what the code returned.

TRAITS

· A block of code on screen that you can open and read
· Same inputs, same answer, every time
· The arithmetic is done by a computer, not by a model

PROS

+ The arithmetic itself is effectively reliable
+ You can check the method, not just the result

CONS

– It can run perfectly correct code on the wrong rows
– The sandbox is offline, so no live rates or prices

Fig 19.1 The same answer, arrived at two ways. Notice that the cons on the right are not about arithmetic: once a computer does the sums, what is left to go wrong is which rows you pointed it at.

19.1

Predicted. The most plausible-looking number, produced the way a plausible sentence is.

Computed. Code ran. The number is what the code returned, and you can read the code.

Both look identical. Formatted the same, rounded the same, stated with the same confidence.

When an assistant gives you a number it has done one of two completely different things. It has predicted the number, the way it predicts the next word of a sentence, or it has written a small program, run that program somewhere real, and reported what came back. The output looks the same either way. The reliability does not.

A predicted number is the most plausible-looking number given everything that came before it. On something like 12 times 8 that works, because the answer appears everywhere in what the model learned. On long multiplication it falls apart. A 2025 study presented at the AIME-Con conference gave four models thirty five-digit multiplications each: GPT-4o got 2 of 30 right, while o1, a model that reasons step by step before answering, got 30 of 30. Same lab, same question, and neither of them touched a calculator. The paper also found that the errors were overwhelmingly procedural rather than conceptual, which is the honest version of "it is bad at maths": it knows what to do and slips while doing it.

Since then all four assistants have grown the second machine. Claude, ChatGPT and Gemini can each write code, run it in a sandboxed container and report the result. Anthropic is blunt about why, describing the point as answers that are mathematically precise and reproducible rather than merely well reasoned. Microsoft made running Python from Copilot inside Excel generally available in August 2026, and its Analyst agent writes and runs Python across several files at once.

That changes the reliability picture completely, and it is the single most important thing in this chapter. It also comes with a condition that almost nobody states out loud: reaching for the tool is a decision the assistant makes, question by question. A piece of arithmetic sitting in the middle of a paragraph often does not look, to the model, like it warrants spinning up a sandbox. So it answers from prediction, in the same voice, in the same reply, with no marker of any kind.

None of this is an argument for keeping numbers away from your assistant. It is an argument for knowing which of the two machines answered you. Before you read on, ask yours to multiply two five-digit numbers off the top of your head, and watch what appears on the way to the answer.

19.2 Look for the working, not for a warning

Nothing on screen announces a guess. There is no red flag, no hedge, no "I estimated this". What there is, when computation actually happened, is an artefact you can open.

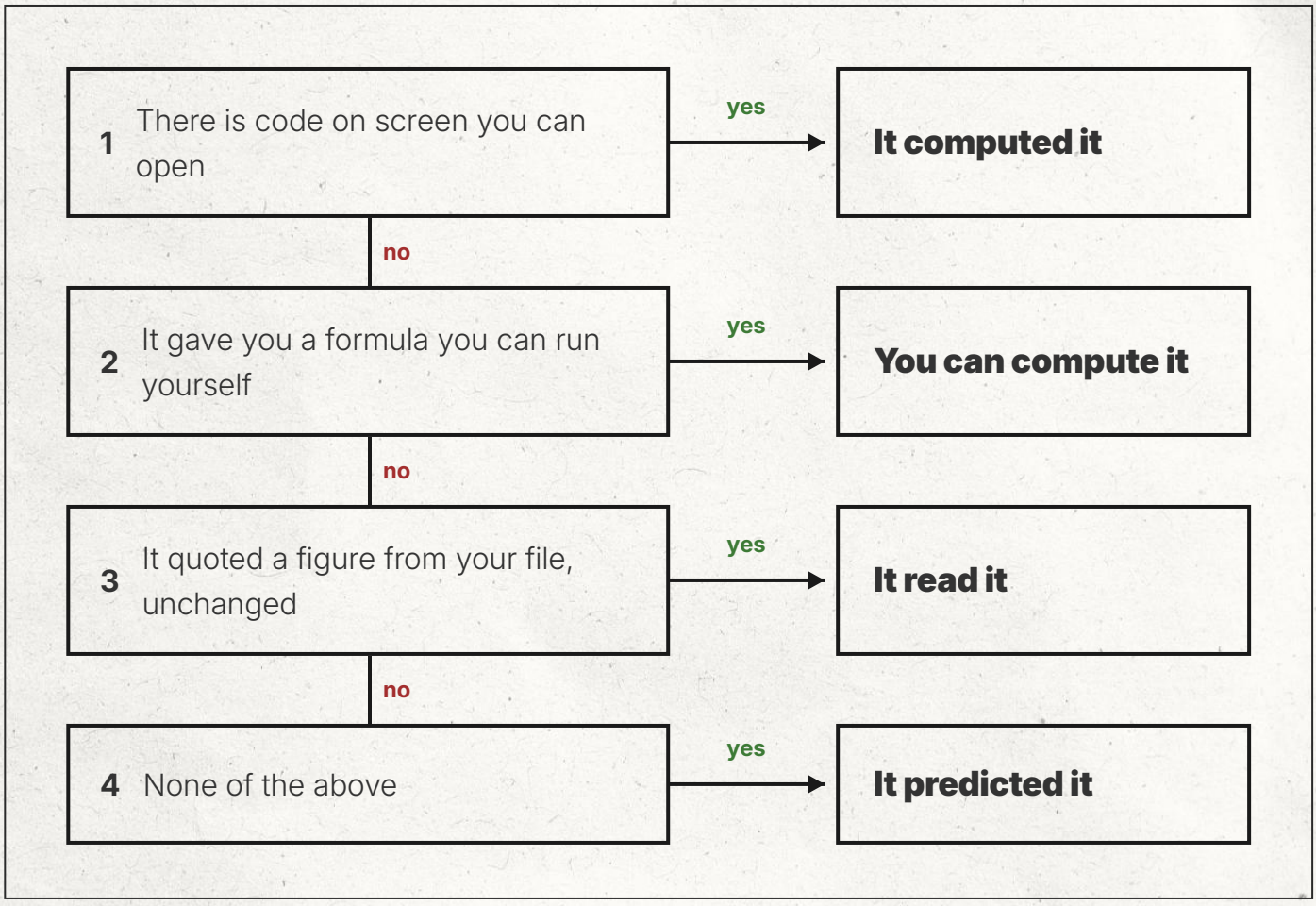

Fig 19.2 Read down until a line is true. The fourth line catches more questions than people expect, and a number that lands there is not wrong on purpose: it is a plausible number, made the way a plausible sentence is made.

19.2

ChatGPT Python sandbox. The analysis step expands so you can read the code it ran.

Claude Writes and runs code in the conversation. Its container is offline by design.

Gemini Python sandbox that takes your file as input and shows the code it wrote.

Copilot Runs Python inside Excel, and the Analyst agent shows the code live. Needs a Microsoft 365 Copilot licence.

Here is the problem this section exists to fix. You cannot tell a computed number from a predicted one by looking at the number, and no assistant labels the difference for you. What you can do is look one layer up, at whether an artefact appeared.

The reliable tell is code. If a block of code appeared and you can open it, something ran. The wording differs by product and changes every few months: ChatGPT collapses it into an analysis step you expand, Claude puts the code and its output into the conversation, Gemini shows the Python it wrote, Copilot lets you watch the Analyst agent work in real time. Ignore the labels and look for the artefact. No code anywhere means no computation anywhere.

A second tell is cheaper and works when you are not sure what you are looking at. Ask the same question again in a fresh chat. A computed answer comes back identical, because the same code ran on the same inputs. A predicted one tends to wobble, and two different answers to one arithmetic question is all the evidence you need.

You can also just insist. "Compute this by running code, and show me the code" is understood by all four, and it is worth making a habit rather than a special occasion. Two caveats. Free tiers cap how often the analysis tool can run, and when you hit that cap you can still ask the question and still get a number, so check for the code rather than waiting to be told. And on Copilot, the Python route lives behind a Microsoft 365 Copilot licence, so on free consumer Copilot assume prediction unless you can see otherwise.

One last thing looks like computation and is not: a number carried forward from earlier in the same conversation. The whole thread is context. A figure you superseded ten messages ago is still sitting there, as visible to your assistant as the corrected one, and it will sometimes pick up the dead version and do flawless arithmetic on top of it. When a key number changes mid-thread, do not correct it in place. Start a new chat containing only the numbers that survived.

19.3 The workbook it answered about is not always the one you attached

Attaching a file feels like handing someone the file. It is closer to handing them a photocopy of one page, taken at an angle, with the formulas frozen at whatever they said the last time anyone saved.

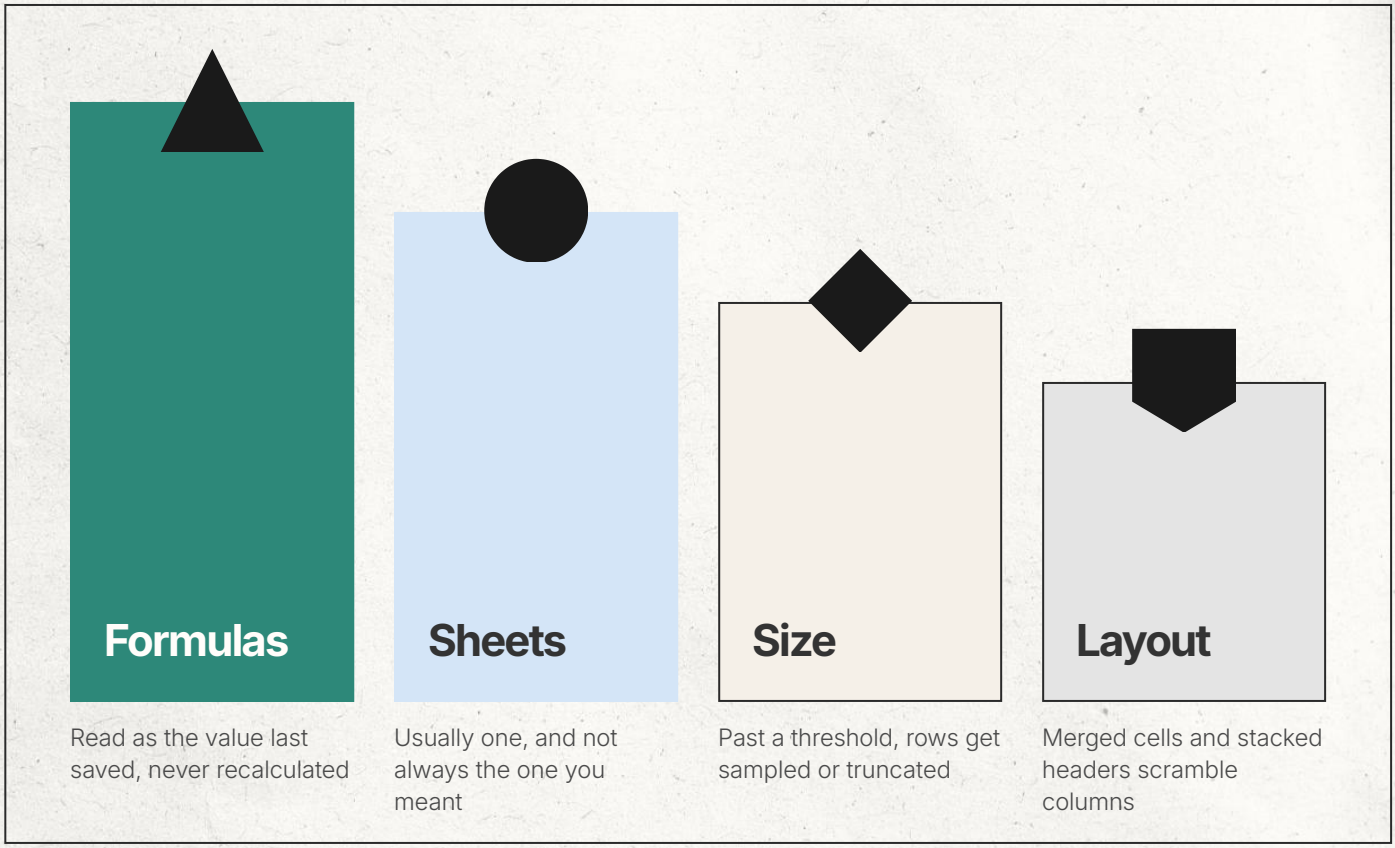

Fig 19.3 Four things that happen to a workbook between your desktop and the answer. The bar heights are visual rhythm, not a ranking or a frequency: the first one is simply the one that surprises people most, because it is invisible from inside the file.

19.3

Formulas. An .xlsx stores the formula and the last value Excel calculated. Your assistant gets the second one.

Name the sheet. And the range. Left to itself it picks, usually the first visible tab.

No live data. The sandbox is offline. Any exchange rate or share price in the output came from memory.

You attach the workbook, ask your question, get an answer in nine seconds, and the answer is about a slightly different workbook than the one you attached. This is the second half of the numbers problem, and unlike the first half it is not about arithmetic at all.

Start with formulas, because this one has teeth. A cell containing a formula stores two separate things: the formula itself, and the value your spreadsheet program calculated the last time it opened the file. Every standard tool for reading these files outside Excel returns the second. The documentation for openpyxl, the Python library sitting inside these assistants' sandboxes, says so in as many words: what you get back is the value stored the last time Excel read the sheet. So if the workbook came out of an export, or was last touched by something that did not recalculate, your assistant reads a stale number and cannot possibly know it. From inside the file, a stale value and a fresh one are identical.

Then sheets. Left to itself it will work on one tab, typically the first visible one, and quietly answer as though that were the whole file. Naming the sheet and the range in your prompt costs six words and removes the entire class of problem: "in the sheet named Balance, rows 4 to 380."

Then size. The published limits differ and move around, and they are the wrong thing to memorise. Gemini currently takes 10 files per prompt at up to 100 MB each; the others publish their own numbers and change them. What matters is the behaviour near the limit rather than the limit itself. Rows get sampled or truncated, and a total computed on a sample looks exactly like a total computed on the file. If your file is large, ask how many rows it actually read, and check that number against the one at the bottom of your screen.

Then layout, which is the most human failure of the four. A title in A1, two rows of stacked headers, merged cells across a quarter, a totals row sitting inside the data. You read past all of that without noticing. A parser does not. A totals row swept into an average is one of the most common ways a spreadsheet answer goes wrong without looking wrong, and it is why the numbers can be internally consistent and still not reconcile with the report you already published.

One more, and it matters more than its size suggests. The code sandbox has no internet: Anthropic's documentation says outbound access is completely disabled, and the others are built the same way for the same reason. So if a currency conversion, a share price or an inflation figure appears in your output, that rate did not come from a source. It came from the model's memory of a rate. Supply the rate yourself, in the prompt, with its date.

19.4 Any number someone else will see arrives with its working

One rule, phrased so it survives being applied at speed. If a figure is going to be read by anyone but you, it comes with its working attached, and you have looked at the working.

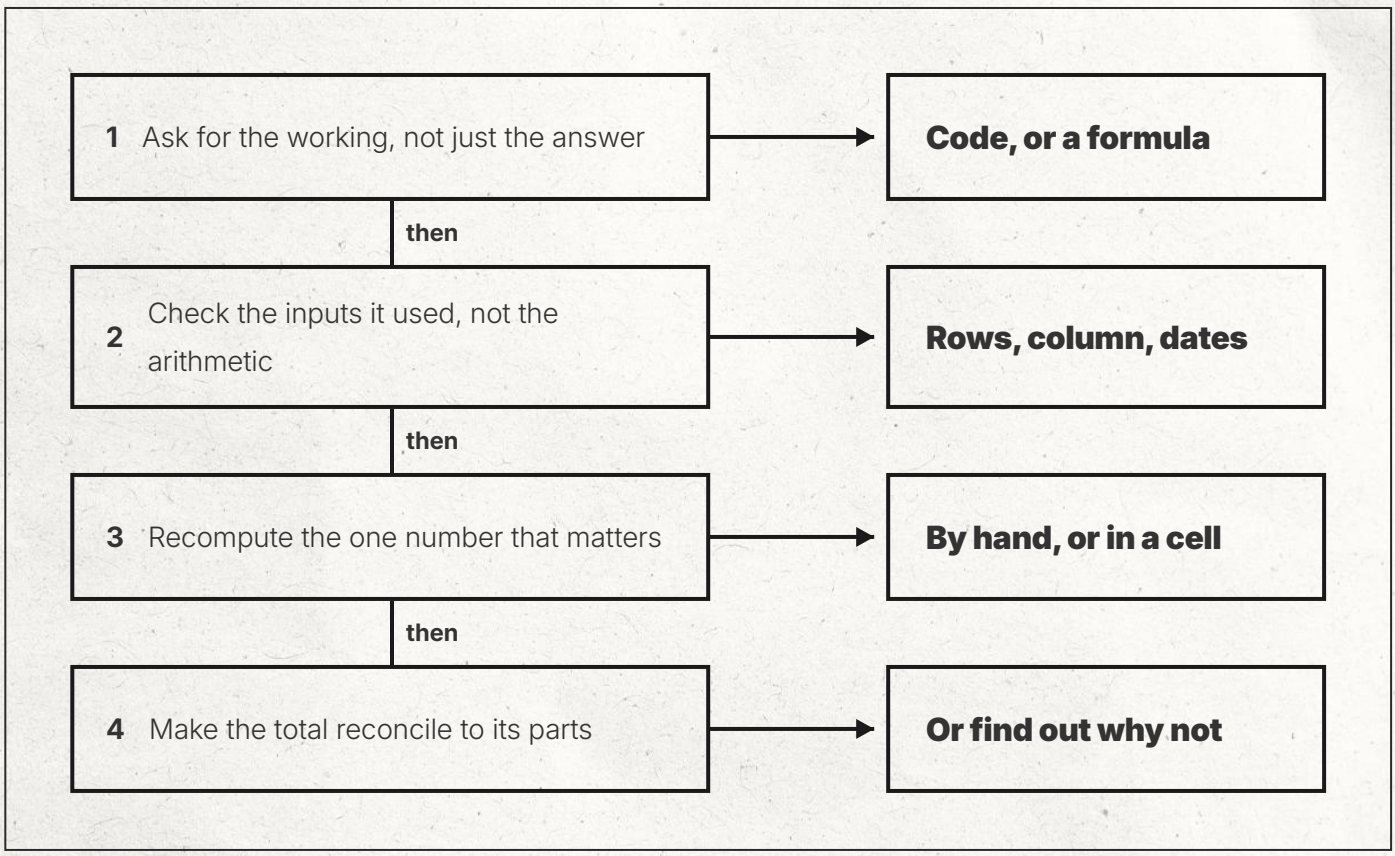

Fig 19.4 Four steps, in this order, on any figure leaving your desk. Step 2 catches more real errors than step 3 does: once code runs, the arithmetic is a computer's arithmetic, and what is left to go wrong is what you pointed it at.

19.4

The rule. Seen by someone else means the working comes with it, and you have read the working.

Units. Percent and percentage points are different units. Say which one you want back.

Dates. "Last 12 months" depends on today. Give it the range, not the phrase.

Here is the rule, and it is short on purpose so that it still works at 6pm on a Thursday. Any number that will be seen by someone else arrives with its working attached, and you have looked at the working. Numbers you are only thinking with are exempt. The moment one gets pasted into a deck, an email or a board pack, it is not exempt any more.

Working means one of three things. Code you can open. A formula you can paste into a cell and run. Or the steps written out with the inputs named, so you can follow them. What it does not mean is your assistant confirming its own answer. Asking it whether a number is right gets you a second prediction from the same machine, usually agreeing with the first, and curing that agreeableness is Chapter 21's whole subject.

When you read the working, read the top of it, not the middle. The two lines that matter are the ones that load the file and filter the rows: which sheet, which column, which date range, which definition of "active". You do not need to read Python to see whether it filtered to 2026 or to 2025, or whether the column it summed is the one called revenue or the one called revenue_gross. That is where the errors now live. The arithmetic underneath is not the thing that will embarrass you.

Two units will get you if you let them. Percent and percentage points are different quantities, and your assistant will mix them exactly as readily as your colleagues do: a move from 4% to 6% is a rise of 2 percentage points and an increase of 50%, and both sentences are true. Growth rates across periods do not average either, so three years of 10%, 10% and minus 20% does not net to zero. State the unit you want back in the prompt, and state it in the answer too. The same discipline applies to currencies and to fiscal versus calendar years.

Now the useful reframe, because this chapter is not really an argument for doing your own sums. Your assistant is weak at one narrow slice of quantitative work, the arithmetic, and that slice is the one you can make verifiable. It is genuinely strong at everything wrapped around it: structuring an analysis before you build it, writing the commentary under the table, naming the column that should be in this report and is not, explaining why a variance moved, and listing what would have to be true for the number to be wrong. None of that is arithmetic, and all of it is the part that takes you an afternoon. Give it that work. Make the arithmetic show its working, and keep it.

READING ENDS HERE

Everything before was reading. Everything after is doing.

3 exercises, About 15 minutes and one real decision that opens Chapter 20.

Do these before Chapter 20

3 exercises. About 15 minutes

Ask your assistant to multiply two 5-digit numbers taken from your own work. Write down whether any code appeared on screen, and what it answered. Then ask the identical question in a brand new chat and write down whether the answer matched.

Take one number you sent to somebody last month. Fill in both boxes from memory, without opening the file.

WHERE THE NUMBER CAME FROM	HOW YOU WOULD PROVE IT IN 60 SECONDS

Attach a real workbook and ask one question you already know the answer to. Write down which sheet it used and how many rows it says it read. You are not testing the arithmetic. You are testing whether it read the file you meant.

Chapter 19 on one page

PREDICTED VS COMPUTED	A predicted number is the most plausible-looking number. A computed one is what code returned. Only the second is arithmetic
THE TELL	Code you can open. No code, no computation. Force it with: "compute this by running code, and show me the code"
THE WORKING RULE	Any number someone else will see arrives with its working attached, and you have looked at the working
WHERE THE RISK SITS NOW	Not the sums. Read the top of the working: which sheet, which rows, which column, which dates, which currency
SPREADSHEET TRAPS	Formulas read as last-saved values · one sheet unless you name it · large files get sampled · the sandbox is offline, so no live rates

BRING TO CHAPTER 20

The workbook you tested and the one number you checked. Chapter 20 moves from numbers you supplied to facts your assistant goes out and finds.

BOTTOM LINE

You can tell a computed number from a predicted one, make it show the working, and you have one rule for every figure that leaves your desk.

NEXT CHAPTER

How to Tell Real Research From Recall

Chapter 20 turns to the other half of the trust problem: what happens when your assistant goes looking for information instead of producing it. What it can actually see on the web, what a citation is worth, and how to tell a real source from a plausible one.

BRING WITH YOU

- ☐ The working rule from this chapter, ready to point at facts instead of figures
- ☐ One question you would normally answer with 20 minutes of searching

GO FURTHER

The Ultimate AI Library

Every guide, prompt pack and playbook AI Central has published, in one place. The reference shelf this course keeps pointing you to.

explore the library ↗



WHO MAKES THIS

AI CENTRAL

AI Central Media is a London-based editorial team. Our flagship publication covers practical AI for senior professionals, and 100+ AI companies, SaaS platforms and growth teams have advertised with us since 2023.

100+

AI companies, SaaS platforms, education brands and growth teams have advertised with us since 2023

613K

Accounts reached every month, across the 7 channels we operate

151

Countries our readers are in, plus all 50 US states

PROUD PARTNER OF

GAMMA · IIElevenLabs · guidde. · Outskill · Luma AI

HubSpot · attio · Synthflow · Notion · Replit

We turn attention into pipeline.

visit thecentral.ai ↗