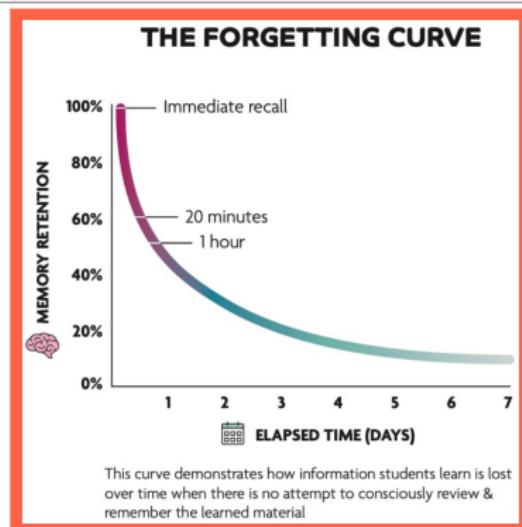


## AI for the working mathematician - part 5

---

Or: What is...reproducibility with AI?

# Why reproducibility is hard in practice



- ▶ **Human memory** Forgetting curve:  $\sim 50\%$  lost after 1 hour,  $\sim 70\%$  after 24 hours, up to 90% after a week (without review)
- ▶ **Reproducibility crisis** Surveys:  $>70\%$  of researchers failed to reproduce others' results;  $\sim 60\%$  failed to reproduce their own results after a few years
- ▶ **For us** Our “experiments” are scripts and notebooks; reproducibility means future-you can re-run them and get the same numbers and proofs

# Why is AI so good at this?

## Metrics used to measure performance

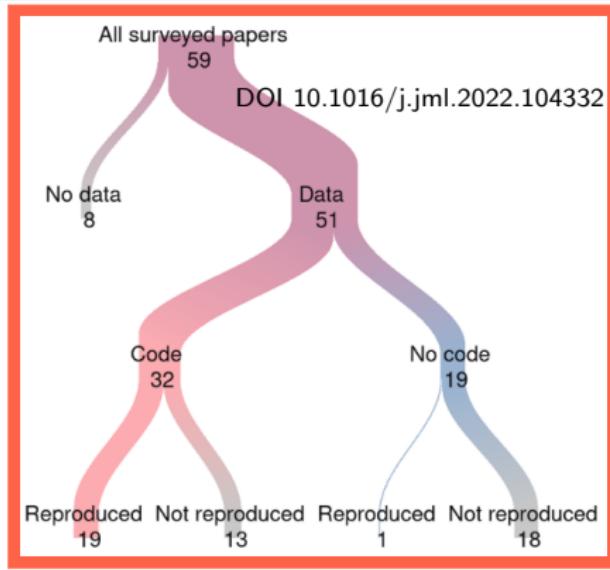
Top responses shown, N=500



<https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

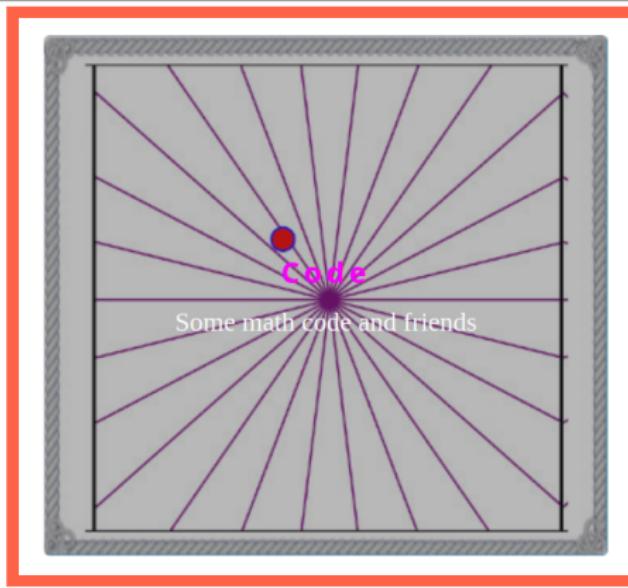
- ▶ **Pattern machines** Most developers now use AI coding tools; models are trained on huge corpora of real code and config files
- ▶ **Used for repetition** Surveys and case studies: people mainly use AI for repetitive work: docs, comments, tests, wiring up scripts
- ▶ **Reproducibility angle** READMEs, comments etc. for future-you

# Studies, studies, studies



- ▶ **Data vs. code** Case study of 59 papers: only about 34–56% were computationally reproducible; with data but *no code*, this drops to ~37%, but jumps to ~80% when scripts are provided
- ▶ **Main barriers** Surveys: top blockers are e.g. incomplete documentation
- ▶ **Moral** Exactly the stuff AI can help you write and standardize

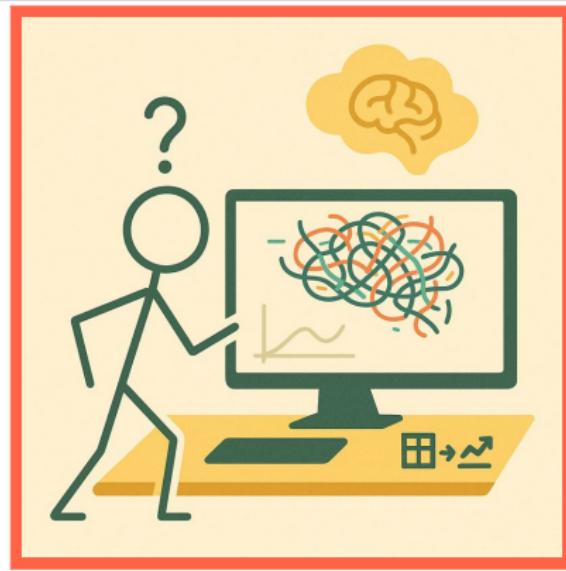
## An example from my own code



- ▶ **Reality check** I have been coding for years; my documentation is terrible: lots of one-off scripts, almost no comments, future-me is lost
- ▶ **What I do now** I feed a messy script or notebook to AI and ask: “Summarize what this does; add comments, docstrings, and a single driver function.”
- ▶ **Result** In a few rounds (you still edit!) I get something I can actually reuse

# Summary

---



- ▶ **Why this matters** Memory fades fast, projects live for years; without a bit of structure, even you cannot reproduce your own work
- ▶ **Minimal kit** For each serious experiment: one folder, one main script/notebook, a short README, dependencies, etc.
- ▶ **Role of AI** AI can cheaply generate the boring bits

**Thank you for your attention!**

---

I hope that was of some help.