CCPR WORKSHOP · UCLA

$ claude

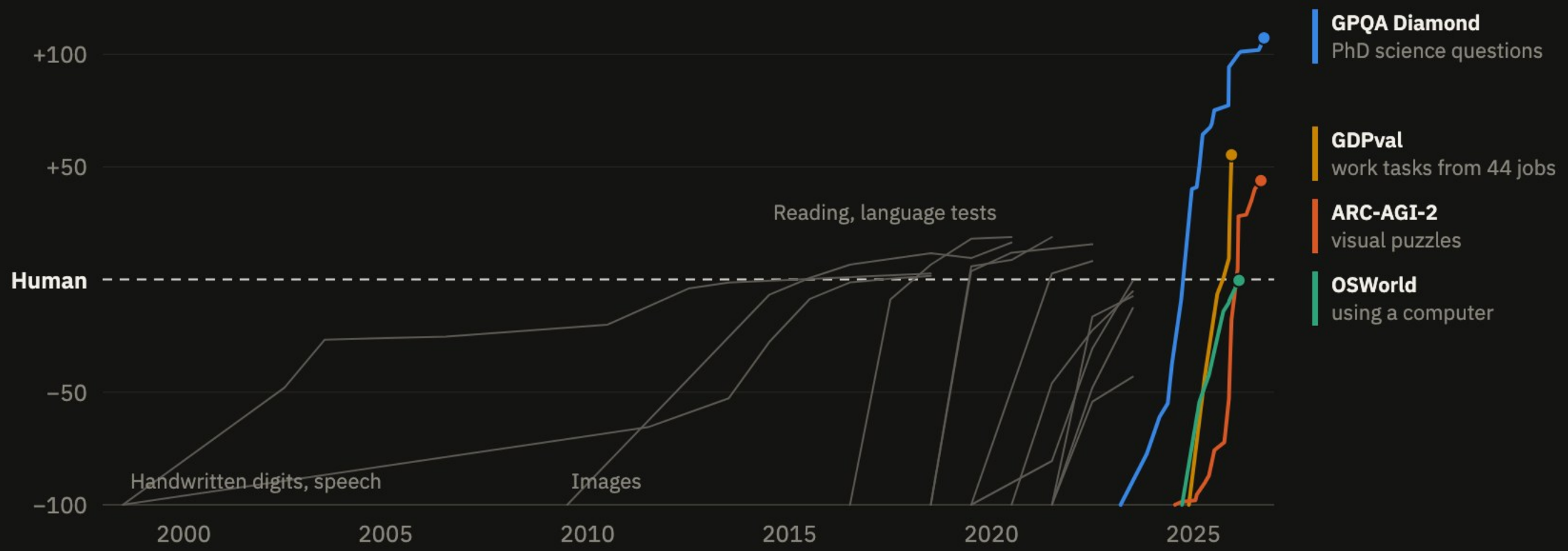
Adversarial Agents

Using Claude Code for research

Jake Anderson · Economics, UCLA

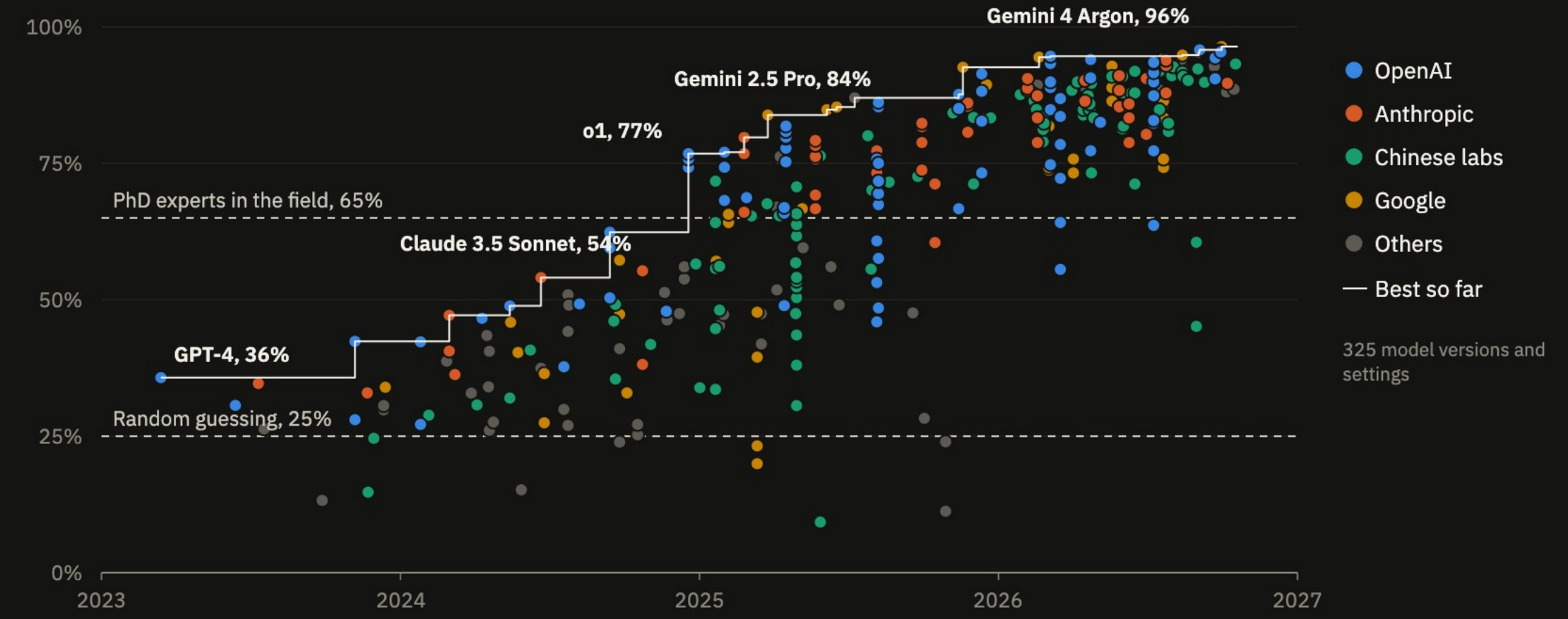
10/28/2026

AI test scores compared with people, 1998–2026



Scale from Kiela et al. (2021): −100 = first recorded score, 0 = human level. Gray: 12 benchmarks from Our World in Data, 1998–2023. Color: best score so far (Epoch AI), rescaled the same way by me. Human level: PhD experts 65% (GPQA), as good as an expert half the time (GDPval), public 66% (ARC-AGI-2), people 72% (OSWorld).

Science test scores by model release date



GPQA Diamond: 198 multiple-choice questions in biology, chemistry, and physics. Data: Epoch AI.

Randomized studies so far

Writing tasks, 453 professionals
Noy & Zhang 2023, Science

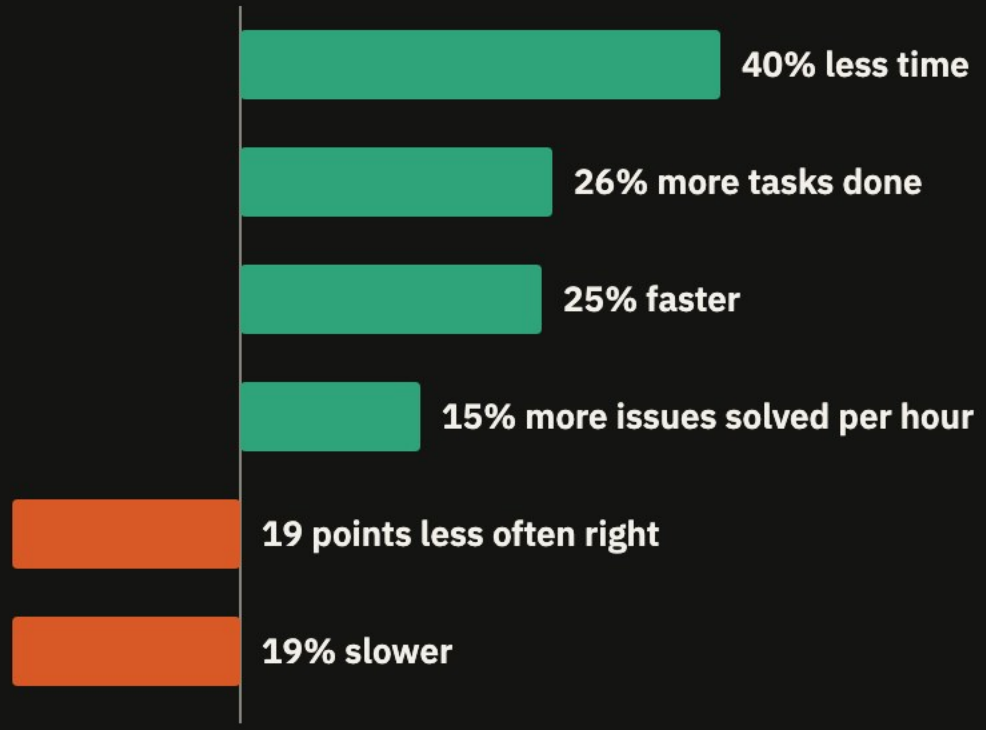
Software, 3 firms, 4,867 developers
Cui et al. 2025, Management Science

Consulting, tasks AI can do, 758 consultants
Dell'Acqua et al. 2023, HBS

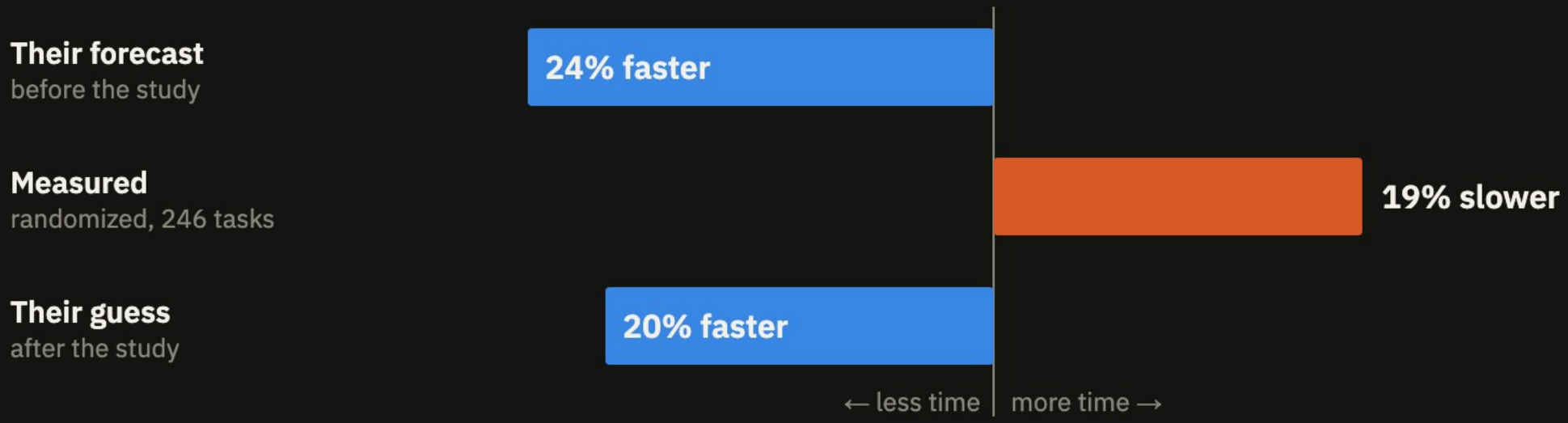
Customer support, 5,172 agents
Brynjolfsson, Li & Raymond 2025, QJE

Consulting, a task AI can't do, same study
Dell'Acqua et al. 2023, HBS

Experienced developers, 16 people
METR 2025

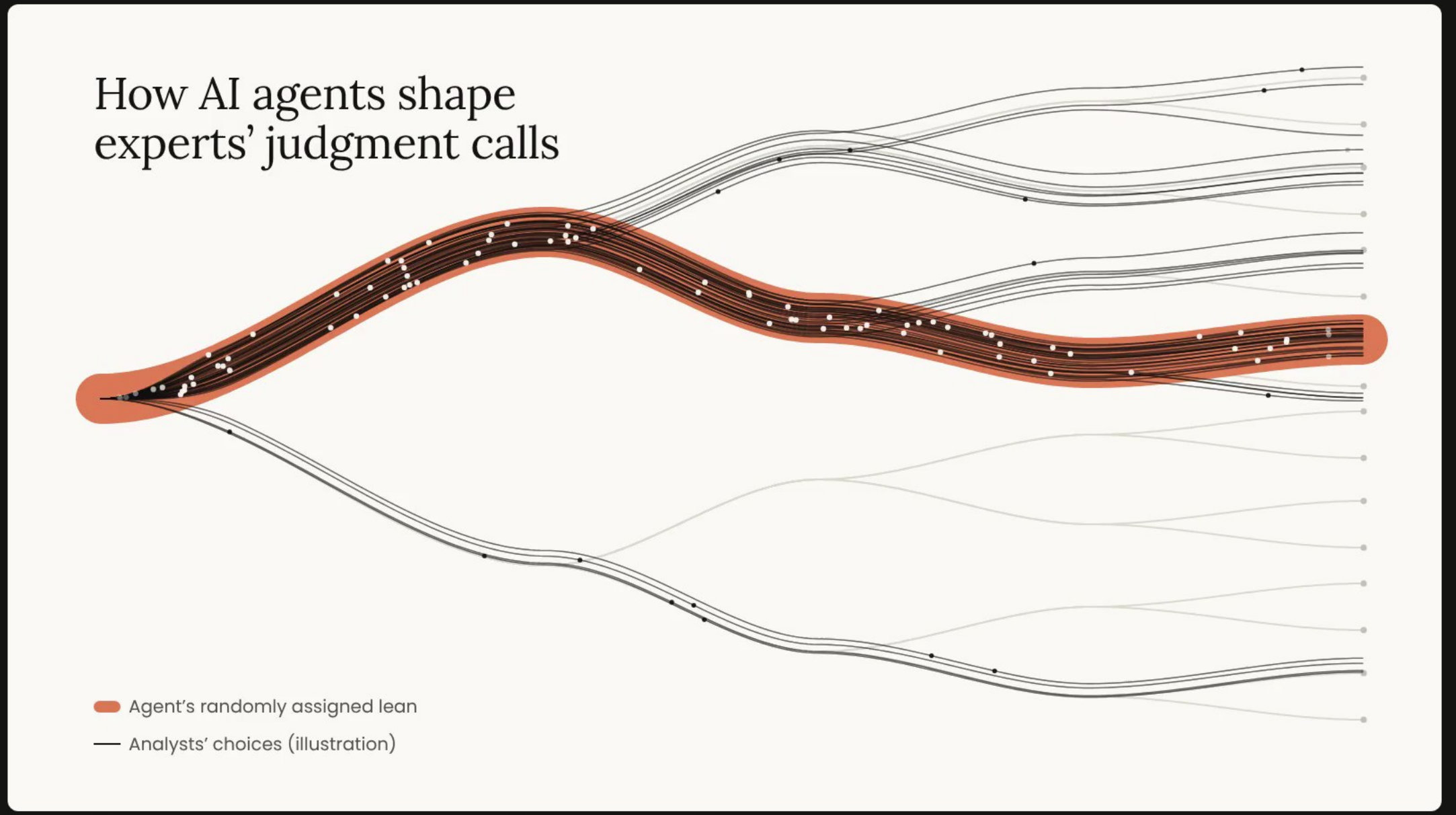


Expected vs. measured speed

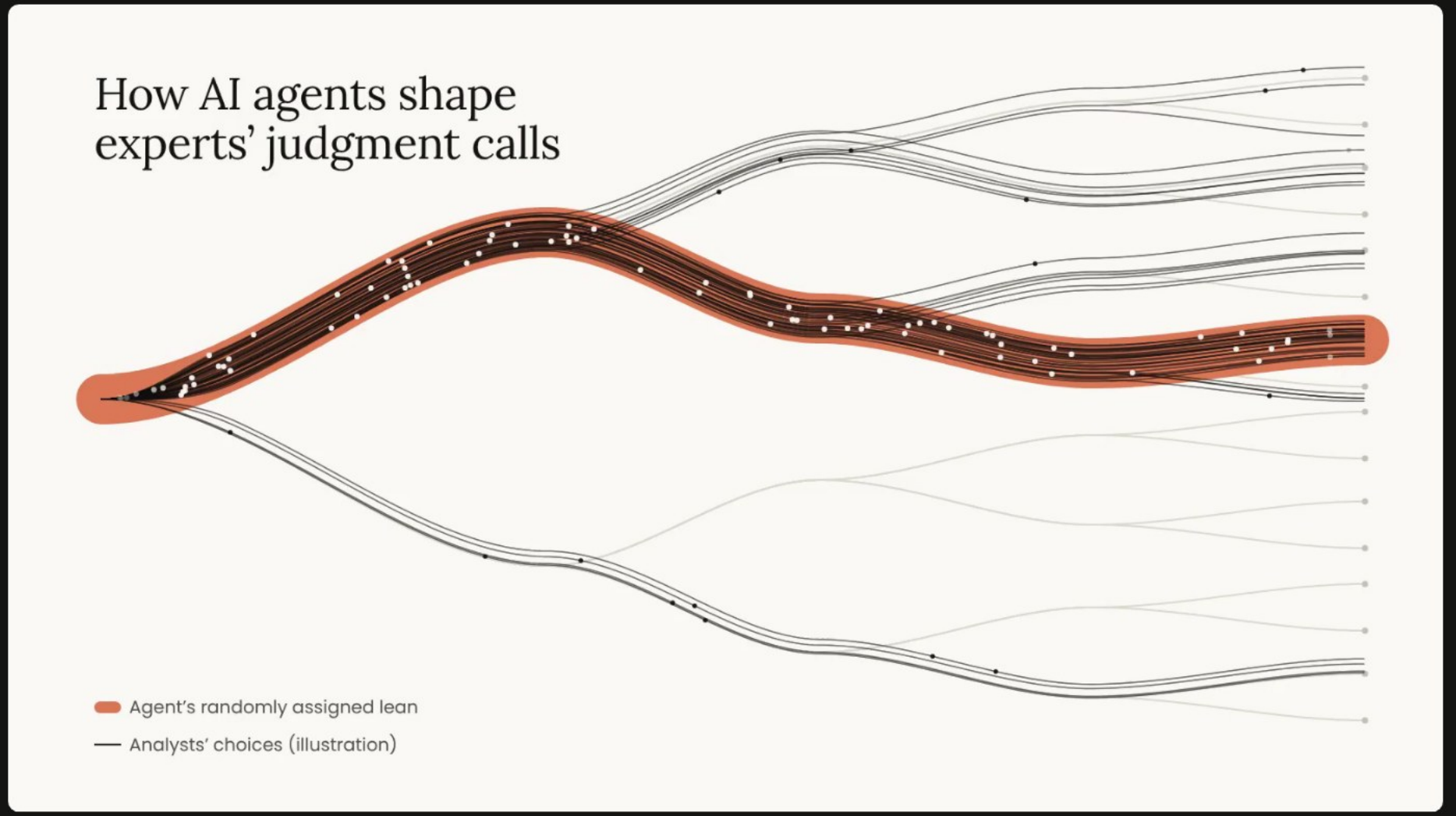


16 experienced open-source developers on their own projects, early-2025 tools (Cursor with Claude 3.5/3.7 Sonnet). A 2026 follow-up with newer tools points to a speedup, but METR calls that weak evidence.

How a session felt is not always calibrated with true productivity



Anthropic, "How AI agents shape experts' judgment calls," Oct 2026 · animation from Alex Tamkin's post on X



Alex Tamkin @AlexTamkin · 18h

This isn't a simple cautionary tale against AI. Analysts who used the agent finished faster and made far fewer serious errors than the 11 who chose to work without it (though that group was small and less experienced).

3 11 200 15K

Alex Tamkin @AlexTamkin · 18h

But as more work is handed to AI—including AI safety research—these judgment calls are the main levers for people's own values to influence outcomes.

If people defer to AI without noticing, then "human in the loop" might provide less assurance than people expect.

6 30 274 15K

Anthropic, "How AI agents shape experts' judgment calls," Oct 2026 · animation and posts from Alex Tamkin on X

WE HAVE SUCCESSFULLY TRANSITIONED TO NEW TECHNOLOGIES IN THE PAST

Machine code → assembly

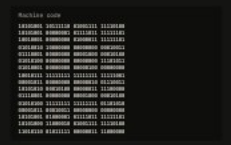
Machine code

```
10101001 10111110 01001111 11110100
10101001 00000001 01111011 11111101
10010001 00000000 01000011 11111101
01010010 10000000 00000000 00010011
01110001 00000000 00001000 00010100
01010100 00000000 00000000 11101011
01010001 00000000 00000100 00000000
10010111 11111111 11111111 11111001
00001011 00000000 00000010 01110011
10101010 00010100 00000011 11100000
01110001 00000000 00001000 00010100
01010100 11111111 11111111 01101010
00001011 00010011 00000000 00000000
10101001 01000001 01111011 11111101
10101000 11000010 01001111 11110100
11010110 01011111 00000011 11000000
```

Assembly

```
fib:  stp  x20, x19, [sp, #-32]!
      stp  x29, x30, [sp, #16]
      add  x29, sp, #16
      mov  w19, #0
      subs w20, w0, #2
      b.lt done
loop: sub  w0, w0, #1
      bl   fib
      add  w19, w19, w0
      mov  x0, x20
      subs w20, w0, #2
      b.ge loop
done: add  w0, w0, w19
      ldp  x29, x30, [sp, #16]
      ldp  x20, x19, [sp], #32
      ret
```

I don't need to worry about binary anymore. I trust the assembler to turn each line into the right 32 bits.



Machine code

Assembly → C

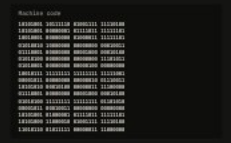
Assembly

```
fib:  stp  x20, x19, [sp, #-32]!
      stp  x29, x30, [sp, #16]
      add  x29, sp, #16
      mov  w19, #0
      subs w20, w0, #2
      b.lt done
loop: sub  w0, w0, #1
      bl   fib
      add  w19, w19, w0
      mov  x0, x20
      subs w20, w0, #2
      b.ge loop
done: add  w0, w0, w19
      ldp  x29, x30, [sp, #16]
      ldp  x20, x19, [sp], #32
      ret
```

C

```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

I don't need to worry about registers anymore. I trust the compiler to choose them and save them.



Machine code

Assembly → C

Assembly

```
fib:  stp  x20, x19, [sp, #-32]!
      stp  x29, x30, [sp, #16]
      add  x29, sp, #16
      mov  w19, #0
      subs w20, w0, #2
      b.lt done
loop: sub  w0, w0, #1
      bl   fib
      add  w19, w19, w0
      mov  x0, x20
      subs w20, w0, #2
      b.ge loop
done: add  w0, w0, w19
      ldp  x29, x30, [sp, #16]
      ldp  x20, x19, [sp], #32
      ret
```

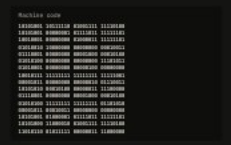
C

```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

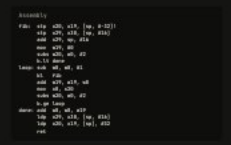
```
fib(46)   1,836,311,903
fib(47)  -1,323,752,223
```

I don't need to worry about registers anymore. I trust the compiler to choose them and save them.

But if I forget how numbers are stored, fib(47) comes back negative.



Machine code



Assembly

C → Java

C

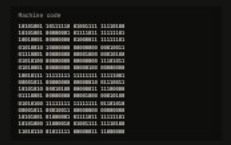
```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

```
fib(46)   1,836,311,903
fib(47)  -1,323,752,223
```

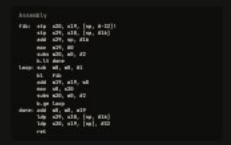
Java

```
static long fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

I don't need to manage memory anymore, and the code is easier to read. I trust the compiler and Java to handle it.



Machine code



Assembly

C → Java

C

```
int fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

```
fib(46)   1,836,311,903
fib(47)  -1,323,752,223
```

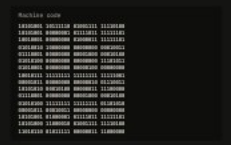
Java

```
static long fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

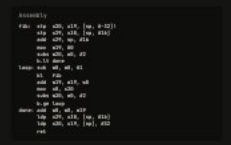
```
long[] results = {fib(10), fib(20), fib(30)};
long[] backup = results;
results[0] = 0;
backup[0]   0
```

I don't need to manage memory anymore, and the code is easier to read. I trust the compiler and Java to handle it.

But if I forget that some variables point to the same data, changing one changes the other.



Machine code



Assembly



C

Java → Python

Java

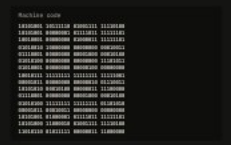
```
static long fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

```
long[] results = {fib(10), fib(20), fib(30)};
long[] backup = results;
results[0] = 0;
backup[0]   0
```

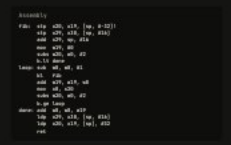
Python

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

I don't need to declare types anymore. I trust Python to work them out.



Machine code



Assembly



C

Java → Python

Java

```
static long fib(int n) {
    if (n < 2) return n;
    return fib(n - 1) + fib(n - 2);
}
```

```
long[] results = {fib(10), fib(20), fib(30)};
long[] backup = results;
results[0] = 0;
backup[0]   0
```

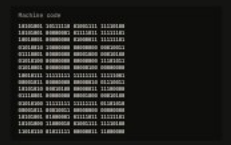
Python

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

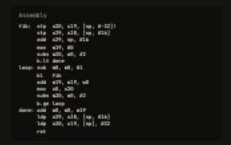
```
df = pd.DataFrame(columns=["n", "n_squared"])
for n in range(100_000):
    df.loc[len(df)] = [n, n ** 2]
52 seconds; 0.02 if built all at once
```

I don't need to declare types anymore. I trust Python to work them out.

But if I forget how a DataFrame stores data, adding rows one at a time gets slow.



Machine code



Assembly



C



Java

Python → English

Python

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

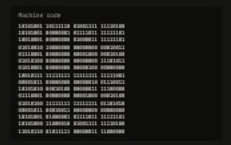
```
df = pd.DataFrame(columns=["n", "n_squared"])
for n in range(100_000):
    df.loc[len(df)] = [n, n ** 2]
52 seconds; 0.02 if built all at once
```

English, to Claude

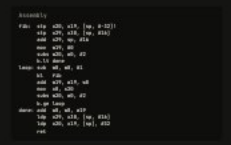
```
> Write a function that returns the
  nth Fibonacci number.
```

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

I don't need to write the code anymore. I trust Claude to write it from what I ask in English.



Machine code



Assembly



C



Java

Python → English

Python

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

```
df = pd.DataFrame(columns=["n", "n_squared"])
for n in range(100_000):
    df.loc[len(df)] = [n, n ** 2]
```

52 seconds; 0.02 if built all at once

English, to Claude

```
> Write a function that returns the
  nth Fibonacci number.

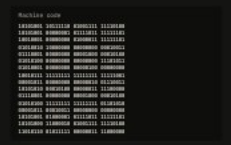
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

---

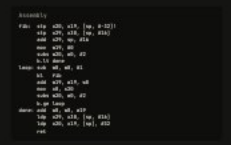
fib(77) about 11 years on my laptop
@cache 34 microseconds

I don't need to write the code anymore. I trust Claude to write it from what I ask in English.

But if I forget what good code looks like, I get code that runs and is wrong or slow: no memoization, the wrong standard errors.



Machine code



Assembly



C



Java

Python → English

Python

```
def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

```
df = pd.DataFrame(columns=["n", "n_squared"])
for n in range(100_000):
    df.loc[len(df)] = [n, n ** 2]
52 seconds; 0.02 if built all at once
```

English, to Claude

```
> Write a function that returns the
  nth Fibonacci number.

def fib(n):
    if n < 2:
        return n
    return fib(n - 1) + fib(n - 2)
```

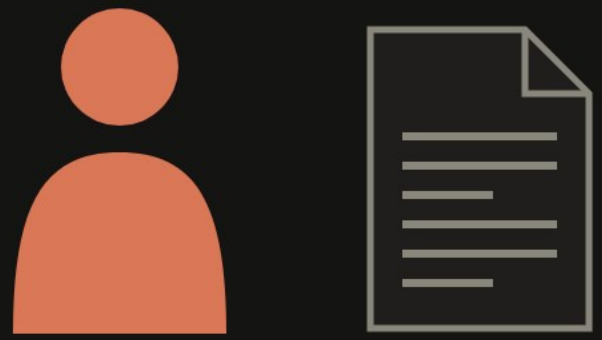
```
fib(77)   about 11 years on my laptop
@cache    34 microseconds
```

I don't need to write the code anymore. I trust Claude to write it from what I ask in English.

But if I forget what good code looks like, I get code that runs and is wrong or slow: no memoization, the wrong standard errors.

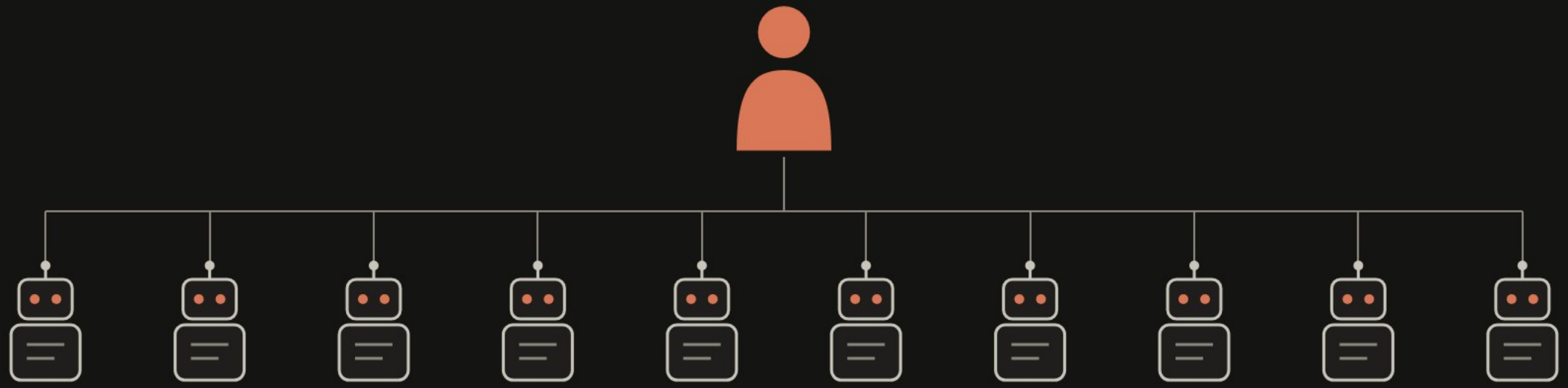
Different from the steps before: a compiler does exactly what I wrote. Claude makes choices I didn't ask for.

Employee → manager



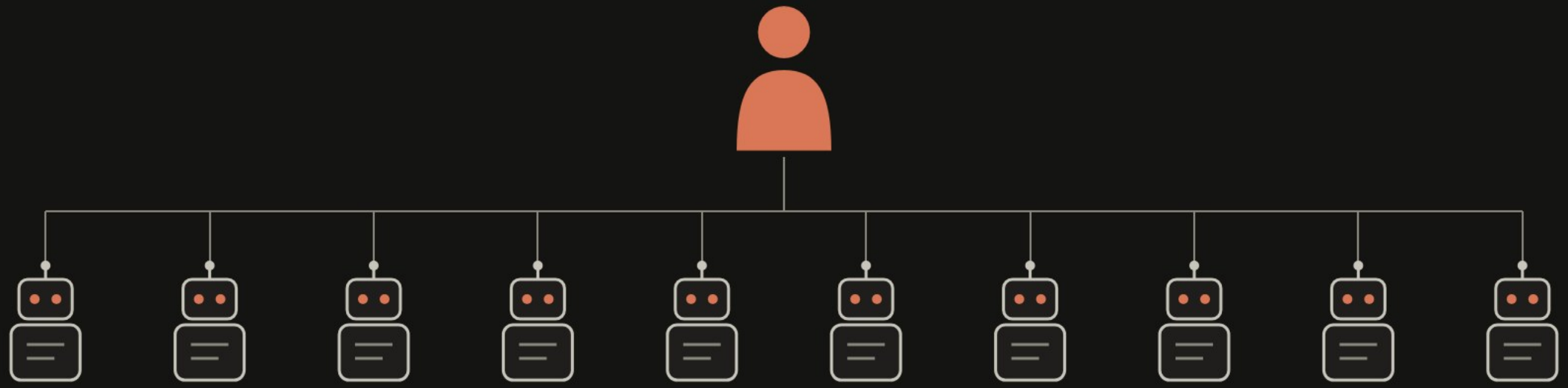
One person, one document. I write it, and I read every line.

Employee → manager



A natural step, but the job changes. What matters most now:

Employee → manager

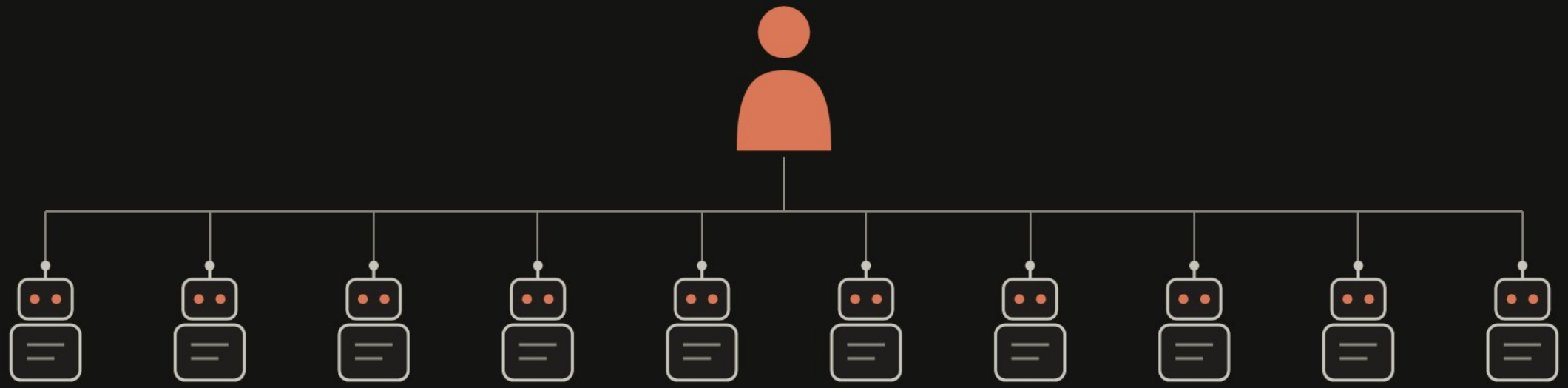


A natural step, but the job changes. What matters most now:

Communication

The task, the data, what not to touch

Employee → manager



A natural step, but the job changes. What matters most now:

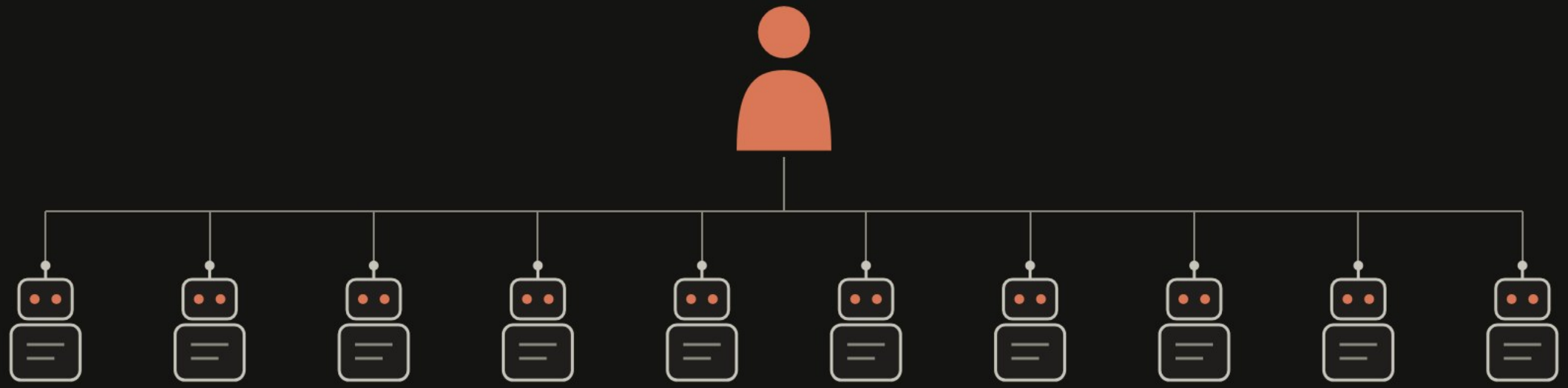
Communication

The task, the data, what not to touch

Visibility

See what each one did: plans, logs, reports

Employee → manager



A natural step, but the job changes. What matters most now:

Communication

The task, the data, what not to touch

Visibility

See what each one did: plans, logs, reports

Robust testing

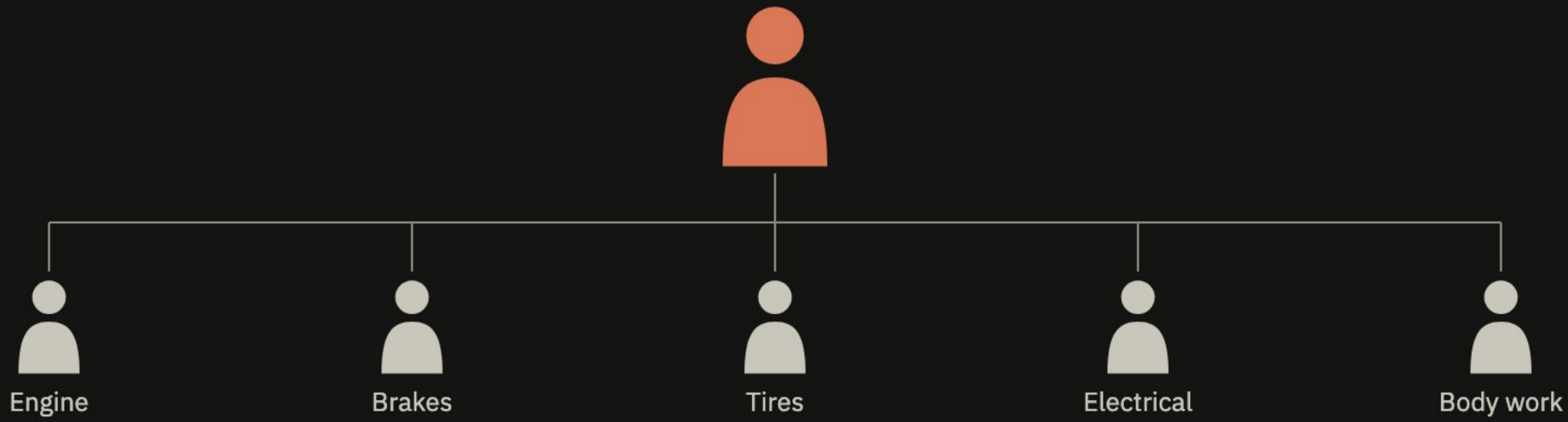
Check the work before you rely on it

Fixing my car → a repair shop



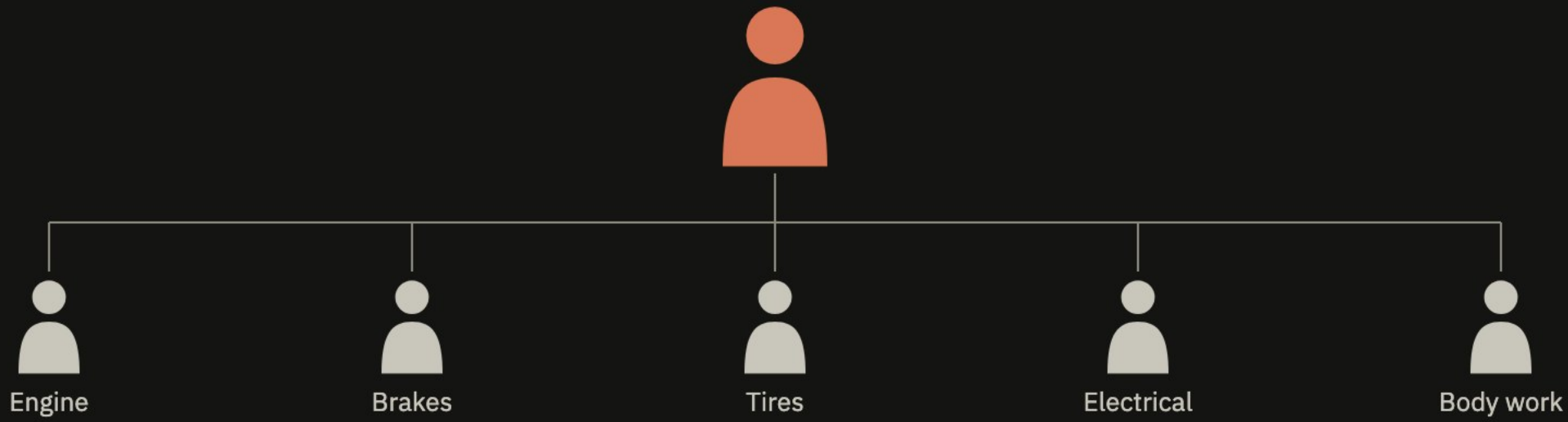
I fix my own car. I know every part I touch.

Fixing my car → a repair shop



I don't need to know what's inside the car. What matters most now:

Fixing my car → a repair shop

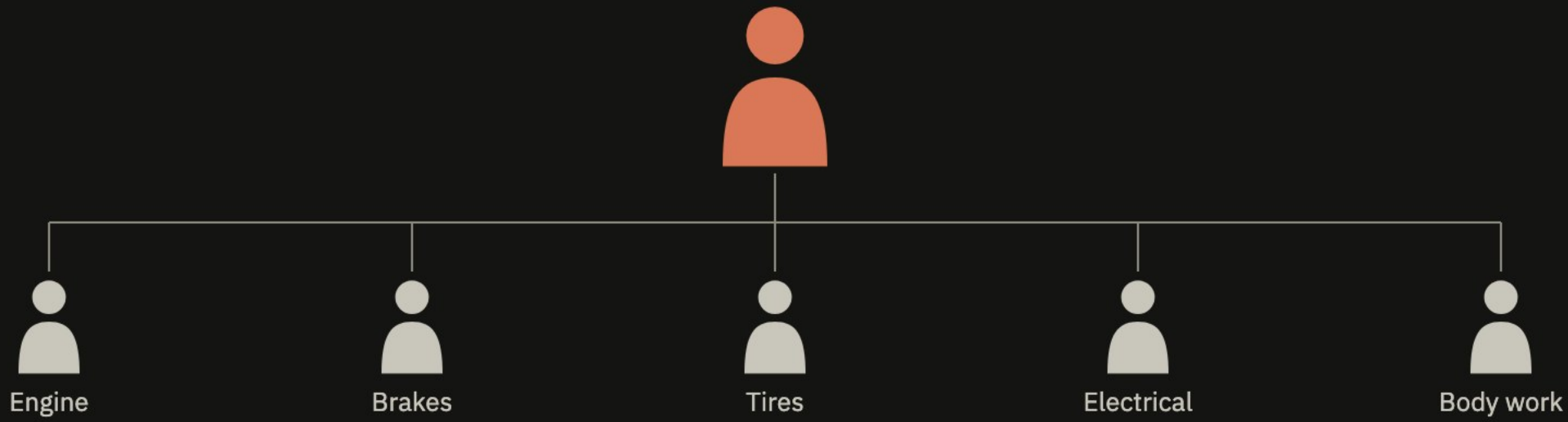


I don't need to know what's inside the car. What matters most now:

Communication

What's wrong, what I'll pay for

Fixing my car → a repair shop



I don't need to know what's inside the car. What matters most now:

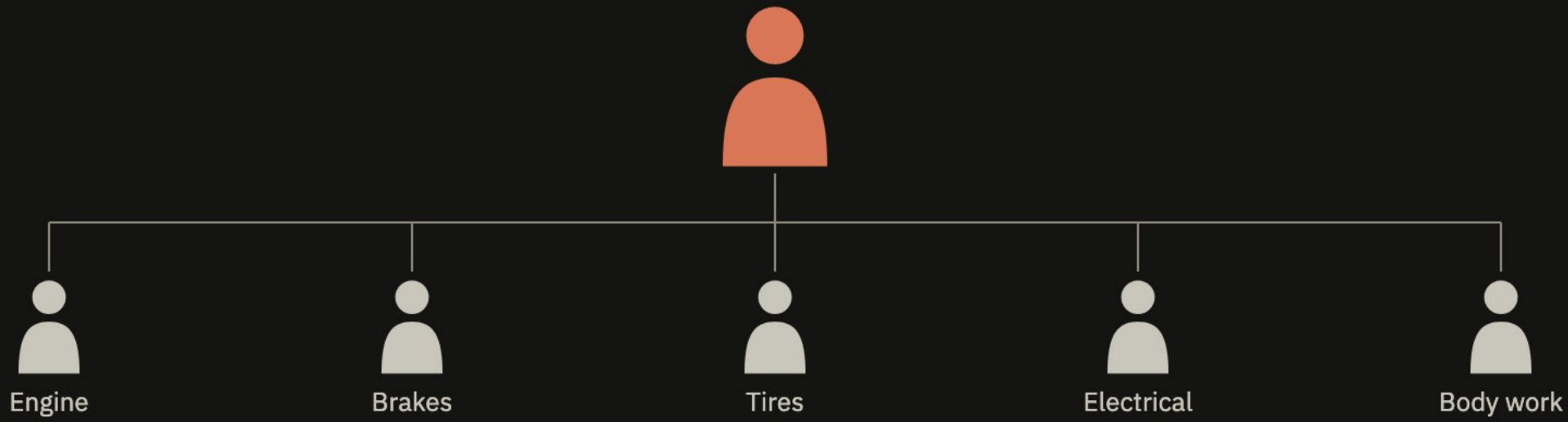
Communication

What's wrong, what I'll pay for

Visibility

An itemized estimate, the old parts

Fixing my car → a repair shop



I don't need to know what's inside the car. What matters most now:

Communication

What's wrong, what I'll pay for

Visibility

An itemized estimate, the old parts

Robust testing

Sometimes my uncle or another mechanic checks it

Scraping a website



Page 1



– How the site stores the data
– What the links and page numbers look like
– What questions to ask about it

Worth my time

Scraping a website



Page 1



– How the site stores the data
– What the links and page numbers look like
– What questions to ask about it

Worth my time



Page 10



– Same layout, new rows
– Nothing new about how the site works

Grunt work: outsource it

Scraping a website



Page 1



– How the site stores the data
– What the links and page numbers look like
– What questions to ask about it

Worth my time



Page 10



– Same layout, new rows
– Nothing new about how the site works

Grunt work: outsource it

Spend time where it helps me understand the process. Outsource the grunt work that doesn't.

Reading a new bill



Page 1



– How things are phrased
– Which policies are bundled together
– Which law codes go with which statutes

Worth my time

Reading a new bill



Page 1



– How things are phrased
– Which policies are bundled together
– Which law codes go with which statutes

Worth my time



Page 100



– More information
– Nothing new about how it's written
– I only want the part I need

Grunt work: outsource it

Reading a new bill



Page 1



– How things are phrased
– Which policies are bundled together
– Which law codes go with which statutes

Worth my time



Page 100



– More information
– Nothing new about how it's written
– I only want the part I need

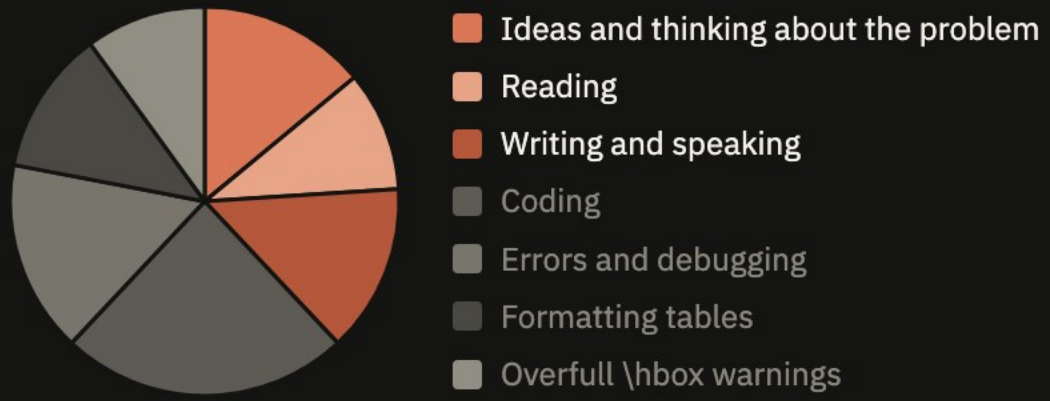
Grunt work: outsource it

Spend time where it helps me understand the process. Outsource the grunt work that doesn't.

Where the time goes

Before

Doing science



Shares are illustrative, not measured

Where the time goes

Before

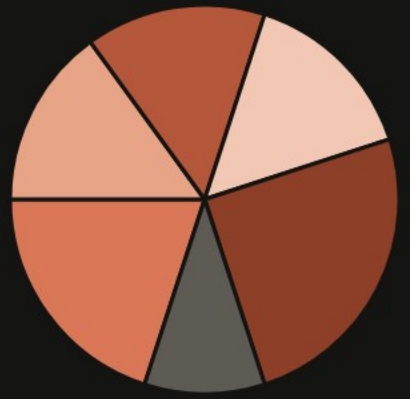
Doing science



- Ideas and thinking about the problem
- Reading
- Writing and speaking
- Coding
- Errors and debugging
- Formatting tables
- Overfull \hbox warnings

The goal

Being a scientist



- Thinking carefully about the setting
- Institutional details
- Learning new methods
- Refining arguments against the literature
- Ideas, reading, writing
- Management

Shares are illustrative, not measured

Where the time goes

Before

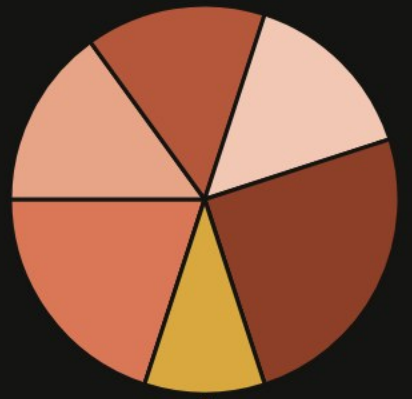
Doing science



- Ideas and thinking about the problem
- Reading
- Writing and speaking
- Coding
- Errors and debugging
- Formatting tables
- Overfull \hbox warnings

The goal

Being a scientist



- Thinking carefully about the setting
- Institutional details
- Learning new methods
- Refining arguments against the literature
- Ideas, reading, writing
- **Management**

Done well, management improves your

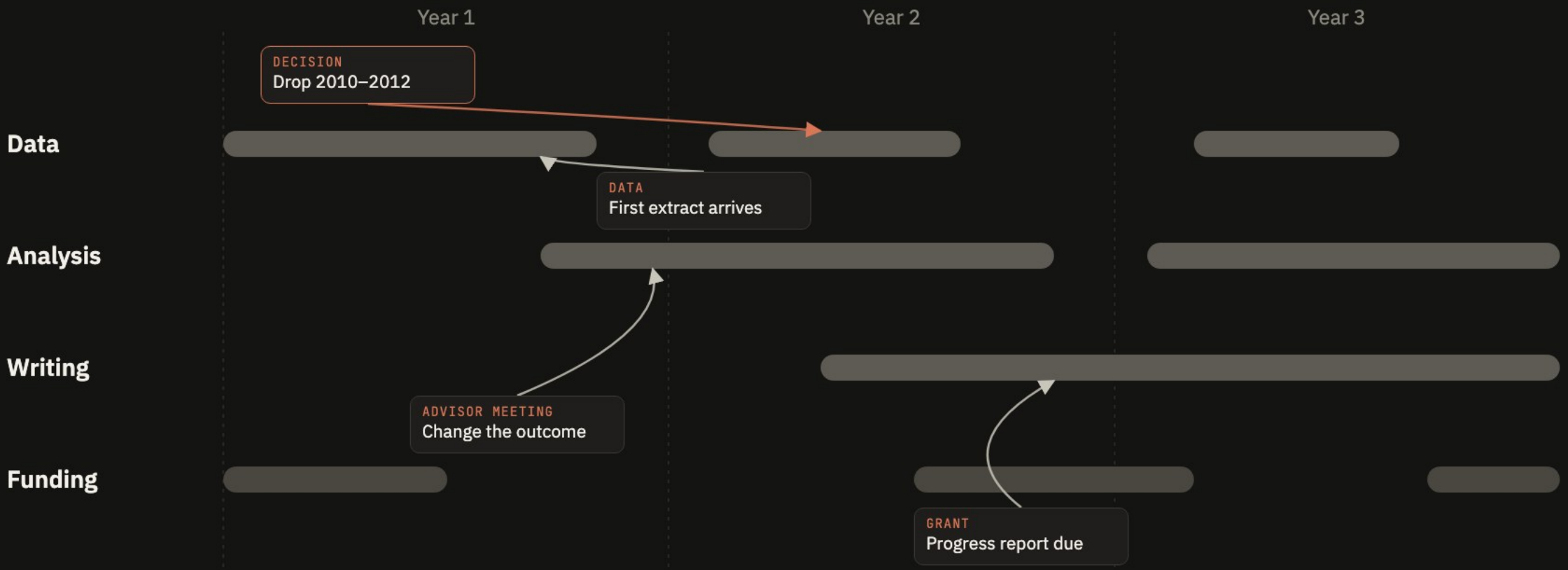
- **communication**
- **specificity**
- **writing**

Shares are illustrative, not measured

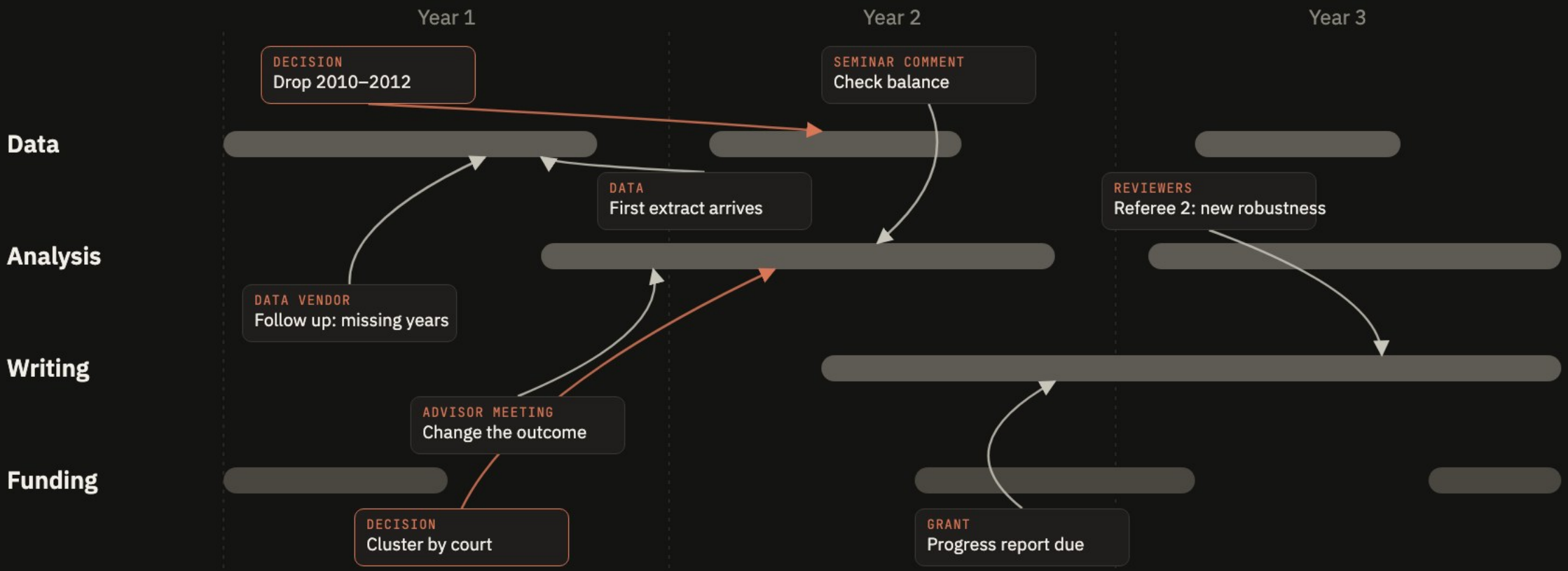
Documentation



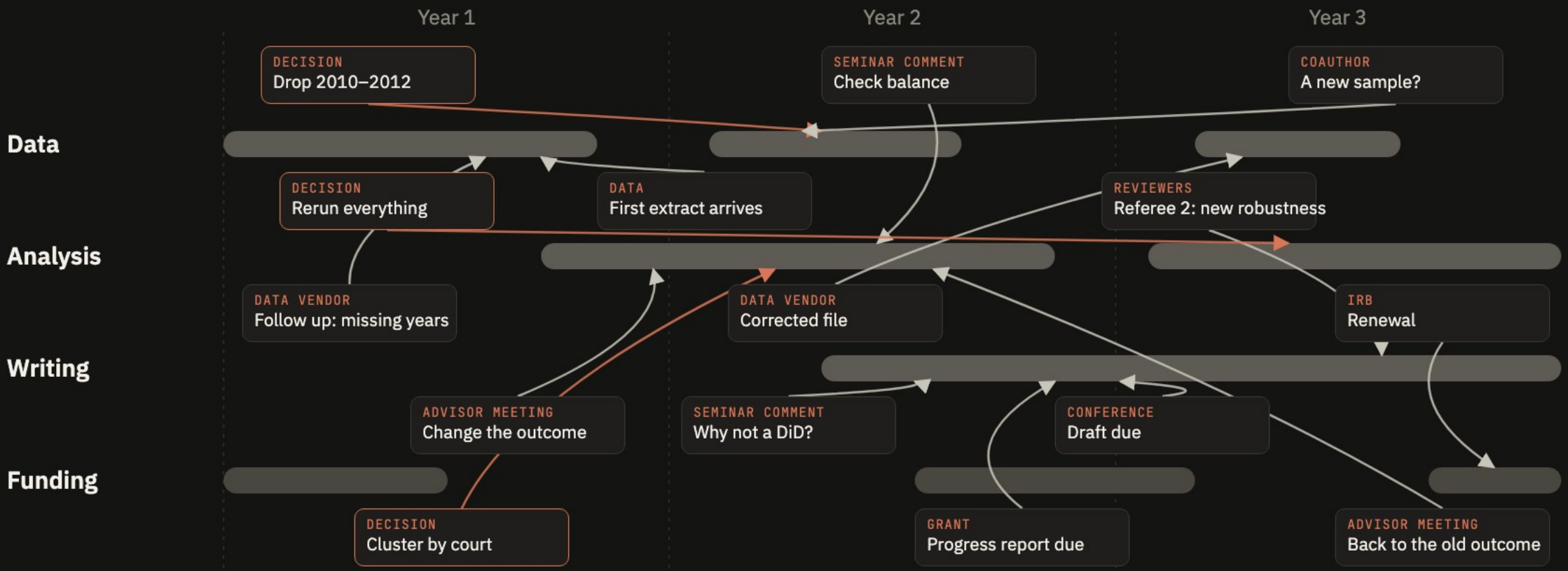
Documentation



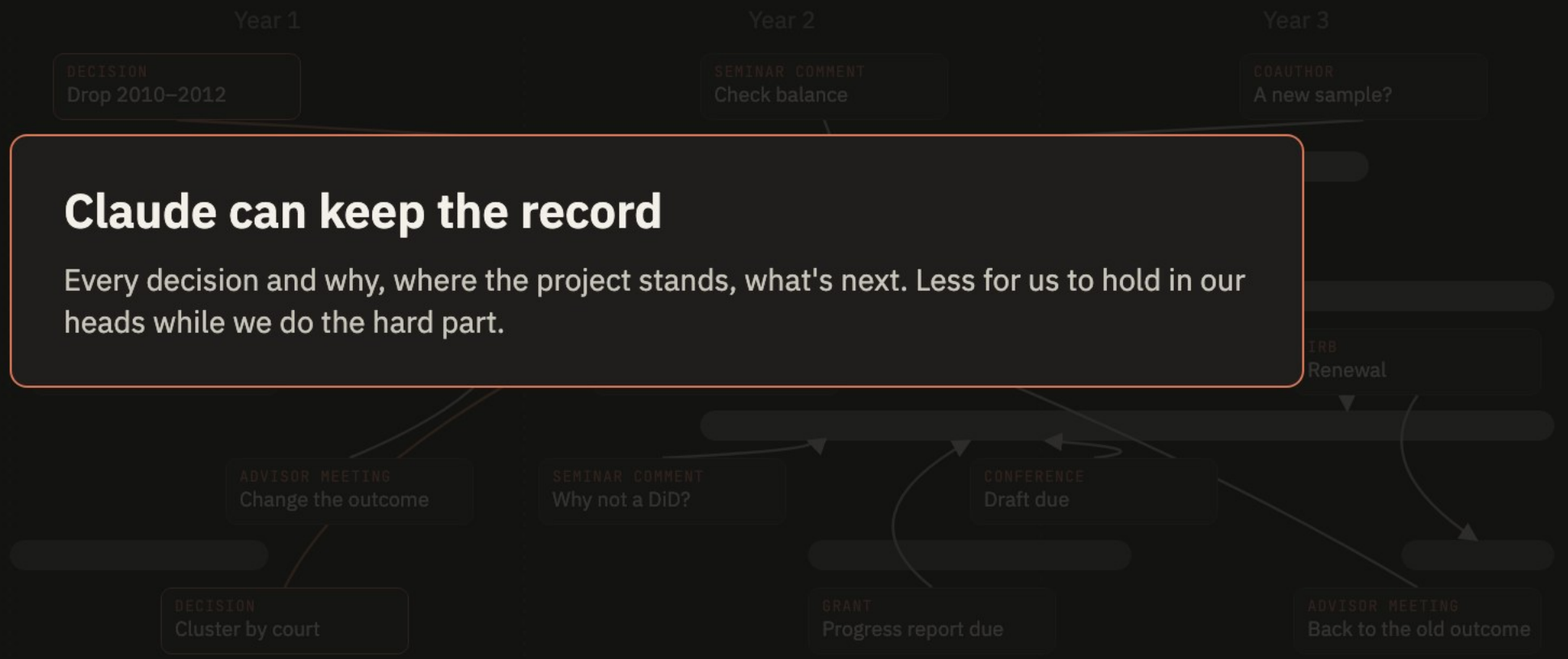
Documentation



Documentation



Documentation



Claude can keep the record

Every decision and why, where the project stands, what's next. Less for us to hold in our heads while we do the hard part.

BEYOND RESEARCH

Other things I use it for

Calendar

Turns emails and syllabi into calendar events

Physical therapy

Turns my PT notes into a daily stretching schedule

Coaching

Writes training plans for the athletes I coach

Teaching

Slide decks with charts and simulations I can run live in class

BEYOND RESEARCH

Calendar

Prompt

> Add the seminars in this email to my calendar with the Zoom links. Block 30 minutes the morning before each one to read the paper. Add my commute, a stretch break every morning, and a 3 pm snack so I don't get cranky.

Result

Mon	Tue	Wed	Thu	Fri
8:15 Commute	8:15 Commute	9:30 Read: workshop paper	8:15 Commute	11:00 Stretch break
9:30 Read: seminar paper	10:00 Applied micro seminar	11:00 Stretch break	11:00 Stretch break	14:00 Coauthor call
11:00 Stretch break	11:30 Stretch break	15:00 Snack	12:00 CCPR workshop	15:00 Snack
15:00 Snack	15:00 Snack		15:00 Snack	
17:30 Commute	16:00 Office hours		17:30 Commute	
	17:30 Commute			

Orange: seminars from the email. Gray: blocks it added (reading, commute, stretching, snack).

Simulated example

BEYOND RESEARCH

Physical therapy

Prompt

> Read my PT notes from MyChart and make a daily stretching schedule for the next two weeks. Put a reminder on my calendar for each session.

Result

Exercise	Dose	When	Note from PT
Hamstring stretch	3 × 30 s	Morning	Each side
Hip flexor stretch	3 × 30 s	Morning	Keep back straight
Glute bridge	2 × 12	Evening	Hold 2 s at top
Cat–cow	1 × 10	Evening	Slow
Ice	15 min	After lifting	

Plus 28 calendar reminders. Medicine goes on the same schedule if there is any.

Simulated example

BEYOND RESEARCH

Coaching

Prompt

```
> Write Kekoa's fall program: 11
weeks, Aug 17 to Nov 1, three days a
week. Test day in week 5, then new
maxes. Back-off weeks in 4 and 8. Add
a second page with warm-ups and how to
read the numbers.
```

Fall 2026 Olympic Weightlifting Program: Kekoa

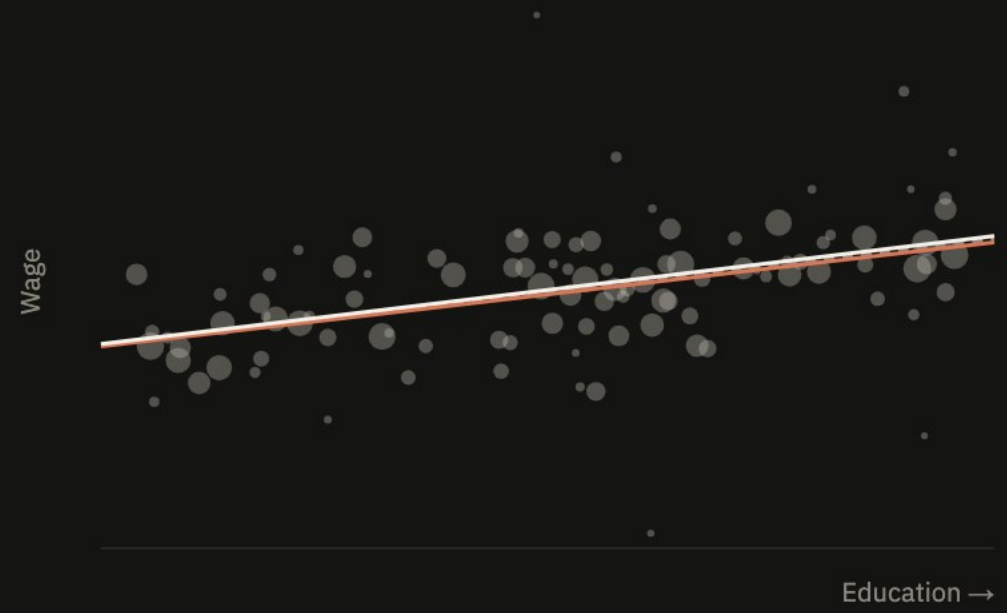
August 17 to November 1, 2026

	Day 1	Day 2	Day 3
Wk 1 Aug 17 to 23	Tall snatch 5x3 (bar) Snatch 60x5x2 Snatch pull 80x4x3 Muscle snatch + overhead squat 4x3+2	Tall clean 5x3 (bar) Clean and jerk 60x5x1+1 Clean pull 80x4x3 Behind the neck push press 4x3 (light)	Tall jerk 4x3 (bar) Snatch 60x4x2 Power clean + jerk 60x4x1+1 Front squat heavy triple @ RPE 8-9
Wk 2 Aug 24 to 30	Snatch 65x5x2 Snatch pull 85x4x3 Snatch balance + overhead squat 60x4x2+2	Tall clean 4x3 (light) Clean and jerk 65x5x1+1 Clean pull 85x4x3 Push press 65x4x3	Tall jerk 4x3 (light) Snatch 65x4x2 Power clean + jerk 65x4x1+1 Front squat, 3s pause 70x4x4
Wk 3 Aug 31 to Sep 6	Tall snatch 4x3 (light) Snatch 70x5x2 Snatch pull, pause at knee 85x4x3 Snatch balance + overhead squat 65x4x2+2	Clean and jerk 70x5x1+1 Clean pull 88x4x3 Behind the neck jerk 65x4x3 Pendlay row 5x5 light	Snatch 68x4x2 Clean and jerk 68x4x1+1 Front squat 75x3 80x3x3 Strict press 5x5 light
Wk 4 Sep 7 to 13 *Back off*	Drop snatch 5x3 (light) Snatch 65x4x2 Snatch pull 80x3x3 Muscle snatch + overhead squat 4x3+2	Tall clean 4x3 (light) Clean and jerk 65x4x1+1 Clean pull 80x3x3 Power jerk 65x4x3	Snatch 65x3x2 Power clean + jerk 65x3x1+1 Front squat 70x3x4 Sots press 3x5 (light)
Wk 5 Sep 14 to 20 *Test day*	Tall snatch 3x3 (light) Snatch 70x4x2 Snatch pull 85x3x3 Snatch balance + overhead squat 72x3x2+2	Power snatch 70x3x2 Power clean + jerk 70x3x1+1 Front squat 70x3x2	Snatch Clean and Jerk
Wk 6 Sep 21 to	Snatch 75x5x2 Snatch pull, pause at knee 85x3 90x3x3	Clean and jerk 75x5x1+1 Clean pull 85x3 90x3x3	Snatch 72x4x2 Clean and jerk 72x4x1+1

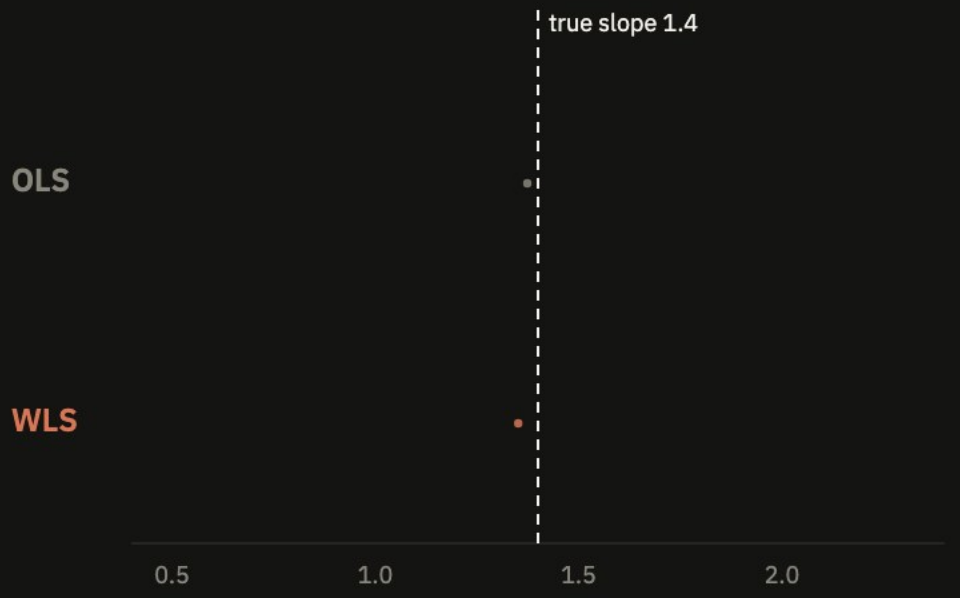
BEYOND RESEARCH

Teaching: WLS, live

One sample: 100 workers; point size = weight



Estimated slope, every sample so far



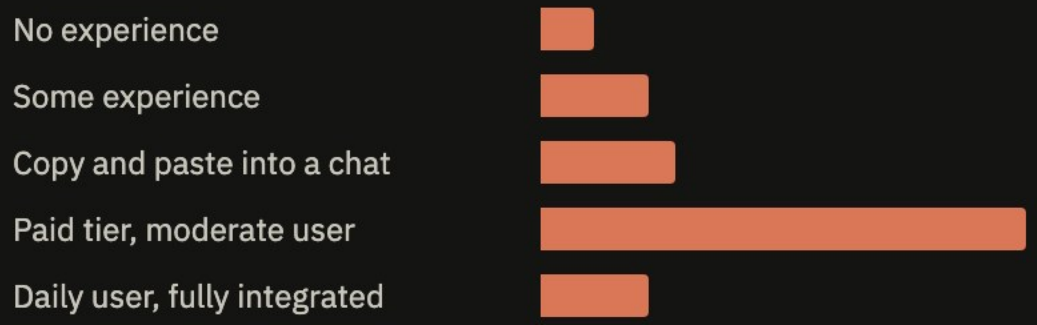
New sample | Run 200 | Reset

Samples: **1** Standard error, OLS: – WLS: –

SIGN-UP FORM

Who's here

Experience



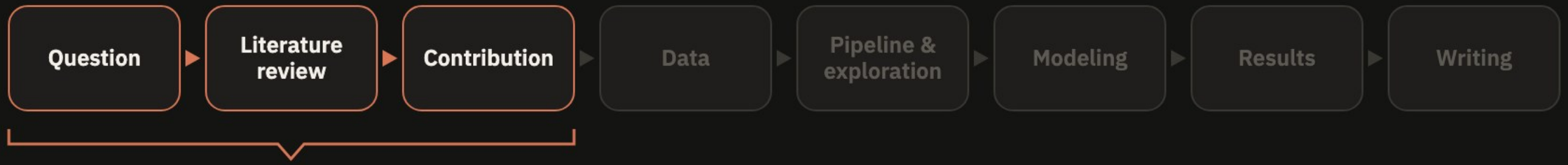
What you use it for



The research process



The research process



The most important human part; AI hasn't caught up here

- Depth in one field over breadth
- We know who's working on what
- We know what would move the literature
- Nobody can read every paper in their subfield; it's too much
- We need ways to skim, review, and read summaries to keep up

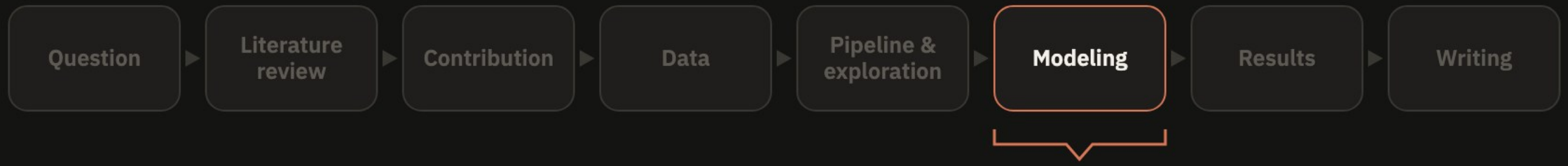
The research process



AI is very good at this

- We used to spend weeks on:
  - Technical PDFs
  - Announcements and outdated government websites, for institutional details
  - Codebooks, for what each variable means
- Scrapers take an afternoon; FOIA is easier (MuckRock has templates)
- Lots of training data on this kind of code, so fewer errors

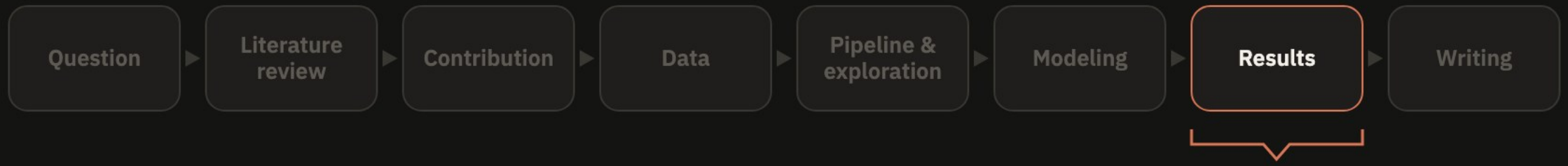
The research process



AI is decent at this, but needs to be hand-held for the best results

- Much better when you give it:
  - Similar designs
  - Code for those designs
  - Other reference material

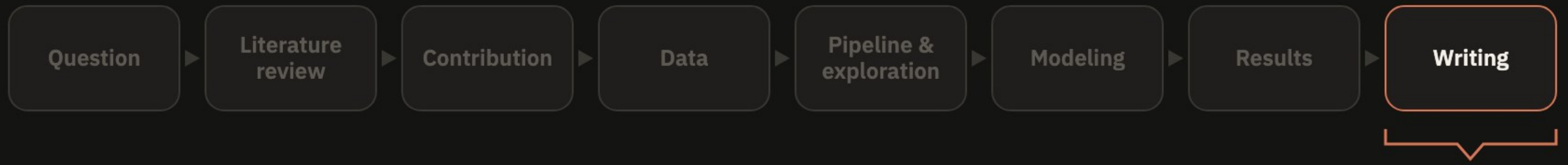
The research process



AI is not very good at interpreting results, but really good at checking them

- Writing tests and checks is the best way to make sure results are sensible
- Feed it the literature: does your result match past work or run against it?

The research process



The academic community is split on this

- Some draft with AI as a first pass
- Others think you shouldn't
- Today: strategies I think both groups can accept

PRINCIPLE 0

Plan first

- Extremely detailed
- Clear headings: goal, sources, definitions, steps, checks, when to stop
- Read it and edit it yourself before anything runs
- Every minute here saves hours later

notes/plan.md

```
# Plan: did county jail populations fall during COVID?

## Goal
How much did jail populations fall from mid-2019 to mid-2020? How much of the
drop is real, and how much is counties that stopped reporting?

## Read first
data/raw/ and its codebook; notes/ideas.md (my notes, don't edit)
literature/ summaries on COVID jail releases

## Definitions
One unit = one jail, not one county. Counties that share a jail all show its
total; count it once.

## Sample
Mid-year (Q2) counts, 2019 and 2020. Only counties that report in both years.
Never add up quarters.

## Steps
1. Profile the data; report anything odd before analyzing
2. Write down the sample rules
3. Compute the change
4. A second agent recomputes it without seeing the code

## Checks (all must pass)
2019 total close to BJS (about 735k); Alaska's 29 rows count as one jail
State totals add up to the national total

## Output
output/change.csv and a one-page memo; every number points to a file

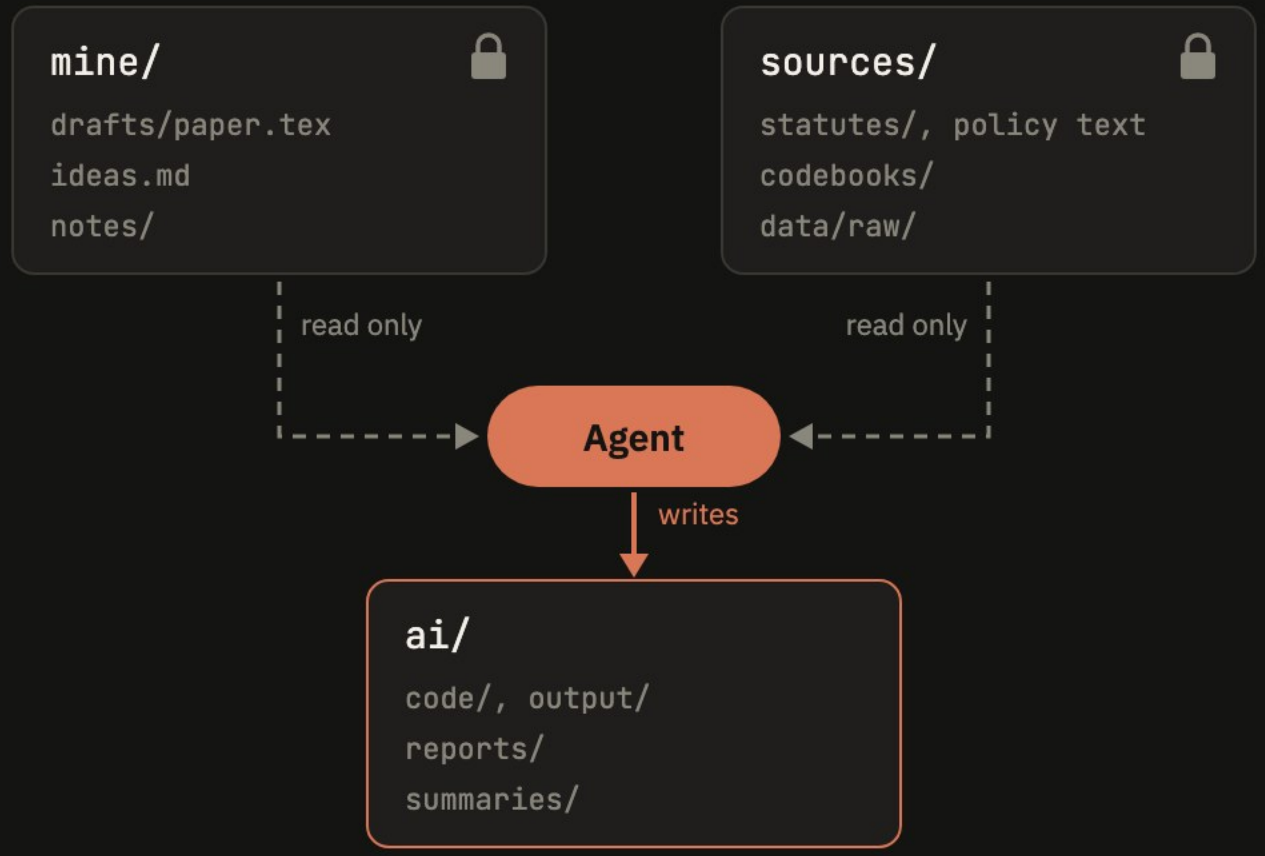
## Out of scope
No regressions yet. No claims about the whole US.

## Stop and ask me if
A check fails, or a choice moves the answer by more than 1 point
```

PRINCIPLE 1

Keep your work apart from the AI's

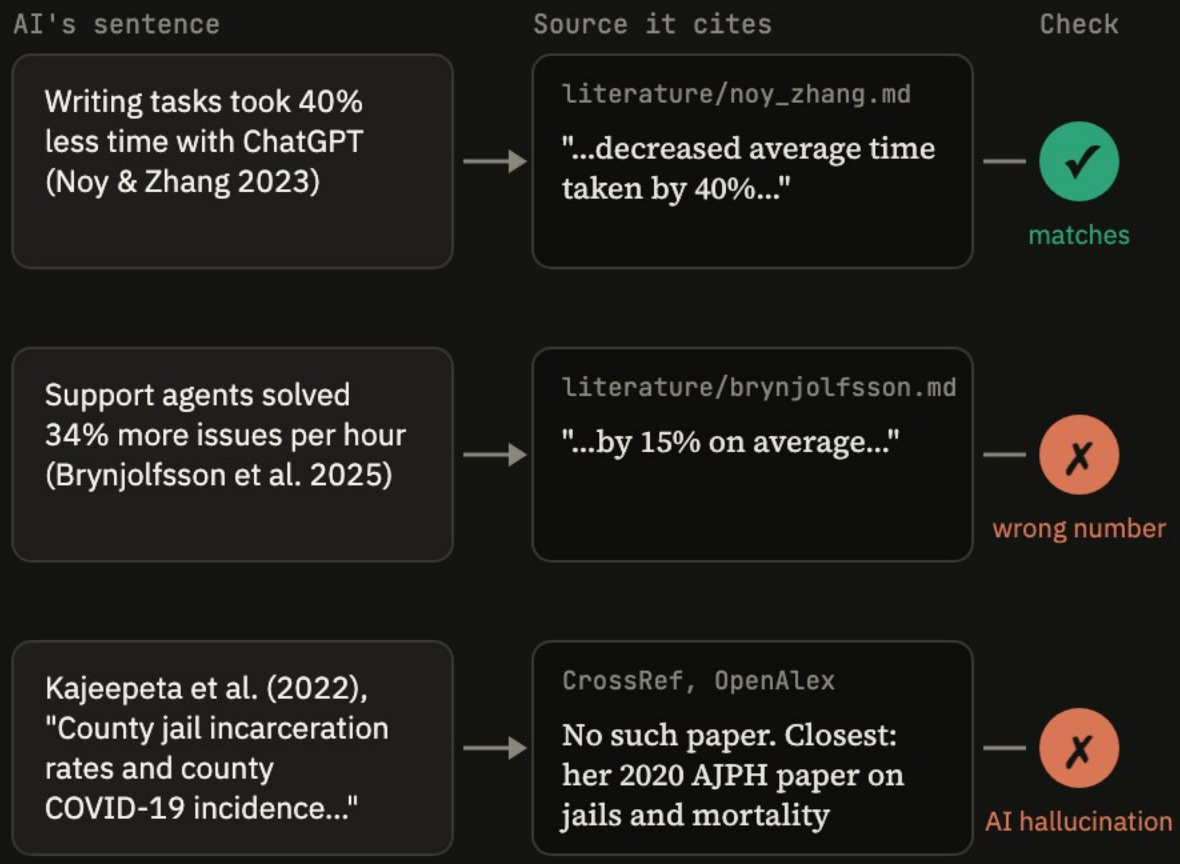
- Your writing, ideas, and notes in their own folder
- Same for raw sources: laws, medical records, policy text, data
- Agents can read them, never edit them
- You always know what you wrote and what the AI touched
- Your voice and ideas stay yours



PRINCIPLE 2

Ask for citations

- Every claim points to a source
- Sources:
  - Papers in your library
  - Policies
  - News articles
  - Other files in your directory: notes, data, codebooks, output
- Ask for the file and the exact passage
- Then check it: a quote you can search for is easy to verify



PRINCIPLE 3

An agent with one job: question everything

- Writes a report
- Can't touch the original
- Makes a triage report:
  - Must fix: the result is wrong
  - Check: could change the result
  - Mention: say it in the paper
- Give it a list of what to look for, not just "check this"

notes/check_report.md

MUST FIX

Shared jails counted many times

All 29 Alaska rows say 2,693; all 5 NYC boroughs say 7,939. 2019 is too high.

CHECK

Who reports changed

2,939 counties report in 2019, 2,261 in 2020. Part of the drop is counties that stopped reporting.

MENTION

Some numbers are estimates, not counts

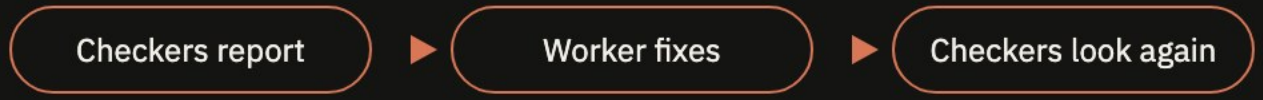
When a jail didn't report, Vera filled in the number: a straight line between years, or the last reported number copied forward. Say so in the paper.

PRINCIPLE 4

Report or change, never both

- Some agents report problems
- Others make the changes
- None does both
- An agent that fixes what it finds is grading its own work
- Set it in each agent file: which tools it gets

	Read files	Write a report	Edit code, data, text
Data profiler	✓	✓	✗
Referee	✓	✓	✗
Auditor	✓	✓	✗
Replicator	✓	✓	✗
Worker	✓	✗	✓



PRINCIPLE 5

Document all human decisions

- AI doesn't know your decisions or why you made them
- So it guesses: choices that are fine in other fields, wrong in ours
- Write down each one: what we chose, what we rejected, why
- Point to the file from CLAUDE.md so every new session reads it

notes/decisions.md

D1 · Which count to use

Claude's guess	Add up every row in the year
We decided	June 30 count only
Why	BJS uses mid-year; summing counts a county up to 4 times

D2 · Which counties

Claude's guess	Every county that reports in either year
We decided	Only counties that report in both years
Why	Otherwise the change mixes in who reports

D3 · Outliers

Claude's guess	Drop the biggest counties as outliers
We decided	Keep every county
Why	They're the biggest jails, not errors

Plan for today

LECTURE

1 Set up a project: CLAUDE.md, folders, protecting your work

2 Agents and skills, on one research idea from start to finish

3 Checking the work: auditors, citations, replication

4 Graphs of agents, and the newer features

HANDS-ON WORKSHOP

Make your own skills, agents, and auditors for one of your projects

Install and start

```
# once
$ npm install -g @anthropic-ai/claude-code
# every time
$ cd ~/research/jail_project
$ claude
# first time in a project
> /init
```

Start it in your project folder

/init writes a first CLAUDE.md

Needs a Pro, Max, Team, or API account

Folder layout

```
jail_project/
  CLAUDE.md          instructions
  .claude/agents/    personas
  .claude/skills/    repeated tasks
  data/raw/          never edited
  data/clean/        written by code only
  code/              01_ingest.py, 02_clean.py
  output/            tables, figures, logs
  paper/             the draft
  slides/            talks
  literature/        one summary per paper
  institutional/     laws, policies, other institutional details
  notes/             plans, decisions, referee reports
  comments/          feedback, one file per person
```

Agents find things by name. With clear folders you can say "use the plan in notes/" and it knows where to look.

Every number in a paper should trace back to a file in output/

Everything in output/ should come from a reproducible script, not just an ad hoc analysis that Claude ran in the terminal

What is CLAUDE.md?

CLAUDE.MD

What is CLAUDE.md?

Every session starts empty

Claude doesn't remember yesterday's session. What you explained, what went wrong, what you decided: gone.

Except this file

A text file in the project folder. Claude reads it automatically at the start of every session, before you type anything.

CLAUDE.MD

What is CLAUDE.md?

Every session starts empty

Claude doesn't remember yesterday's session. What you explained, what went wrong, what you decided: gone.

Except this file

A text file in the project folder. Claude reads it automatically at the start of every session, before you type anything.

Write what you'd tell a new research assistant on their first day:

- What the project is
- Where things are
- What never to touch
- How you want the work done

CLAUDE.MD

What is CLAUDE.md?

Every session starts empty

Claude doesn't remember yesterday's session. What you explained, what went wrong, what you decided: gone.

Except this file

A text file in the project folder. Claude reads it automatically at the start of every session, before you type anything.

Write what you'd tell a new research assistant on their first day:

- What the project is
- Where things are
- What never to touch
- How you want the work done

`~/.claude/CLAUDE.md`

Applies to **all your projects**

How you like to work: writing style, language, general rules

`./CLAUDE.md`

Applies to the **current project**

This project: the question, the data, the folders

CLAUDE.MD

Writing CLAUDE.md

1 Run /init. Claude reads the folder and writes a draft.

2 Read it. Delete anything vague or wrong.

3 Add the question, the folders, and what's off limits.

4 Add 4 to 6 rules you can check.

5 Start a fresh session and ask what the rules are.

6 Every time Claude makes a mistake you don't want again, add one line.

```
# step 1
> /init
# steps 2-4: open CLAUDE.md in any editor
# step 5
> /clear
> What rules do you follow in this
  project?
# step 6
> You summed every quarter. Add a
  line to CLAUDE.md so that doesn't
  happen again.
```

CLAUDE.md contents

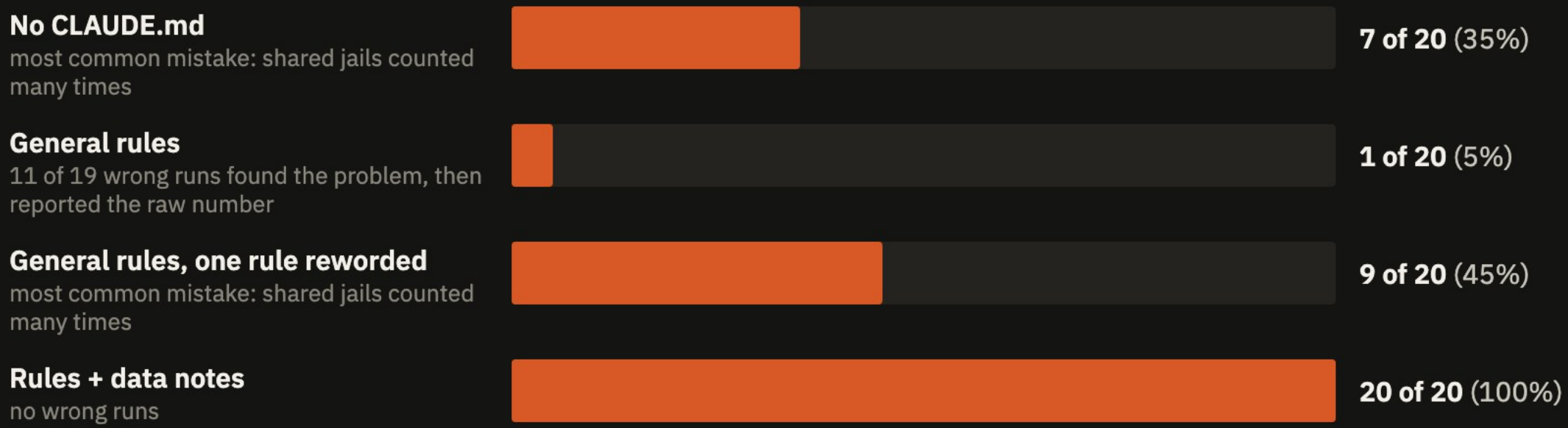
Part	What to write	Example line	Without it
Project	The question, in 1 or 2 lines	`How much did jail populations fall in 2020?`	It guesses what the work is for
Folders	Where things are; what's off limits	`data/raw/: never edit`	Output goes anywhere; raw files can get changed
Data notes	What one row is; known problems	`One row = county x year x quarter`	Every session has to find the same problem again, or misses it
Rules	How to work, in a form you can check	`Print the row count after every filter`	It decides for itself how careful to be

Start with project, folders, rules; add data notes as you find problems

CLAUDE.MD

How important are these components for a data task?

Using the Vera Institute county file, compute the number of people held in US jails in 2019 and 2020, and the percent change



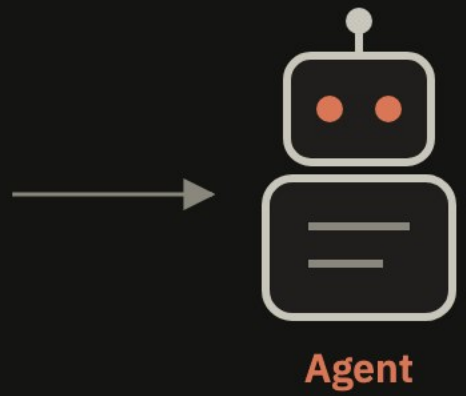
Same task, one agent, 20 fresh runs each; only CLAUDE.md changes

CLAUDE.MD

Resources and examples

Primary sources

- Codebooks
- Source files: the law, the budget
- Paper summaries in literature/

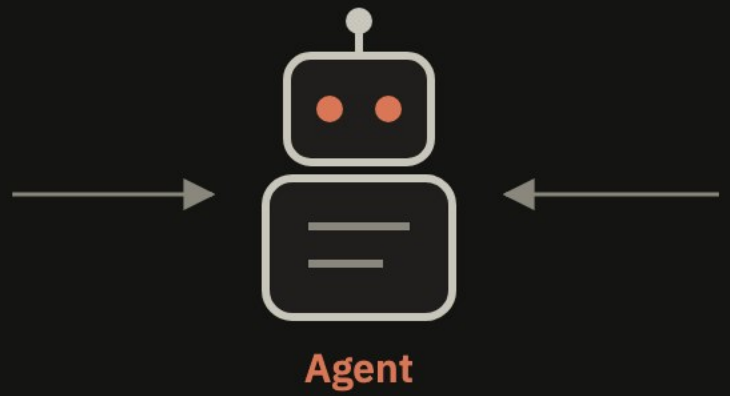


CLAUDE.MD

Resources and examples

Primary sources

- Codebooks
- Source files: the law, the budget
- Paper summaries in literature/



Example output

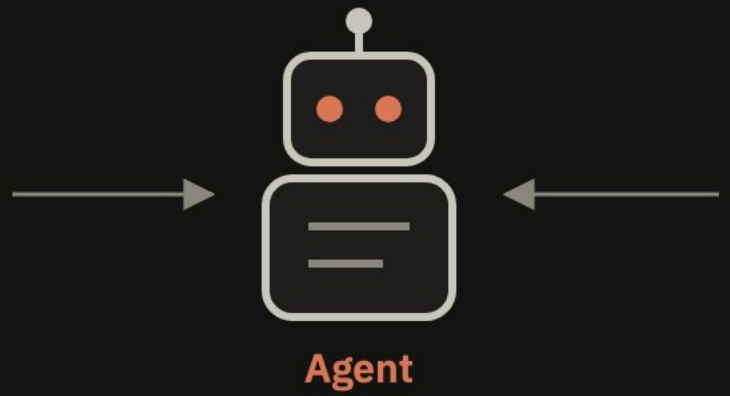
- A table from a paper I trust
- A past figure that passed every check

CLAUDE.MD

Resources and examples

Primary sources

- Codebooks
- Source files: the law, the budget
- Paper summaries in literature/



Example output

- A table from a paper I trust
- A past figure that passed every check

The best things to give an agent: **explicit resources** and **successful examples**.

CLAUDE.MD

Writing rules

Vague	Checkable
Be careful with the data	Print the row count after every filter or merge
Don't make things up	Never report a number you didn't compute from a file in this project
Use good sources	Cite only papers with a summary in literature/
Be thorough	Run every check in checks.md and report pass / fail / marginal
Understand the data	Read the codebook before using a column

Keep it short: the whole file loads every session

Steps for one task go in a skill

Protecting raw data

> **Risk:** an agent edits a raw file to get its code to run, and the original is lost

In Claude Code

```
.claude/settings.json
{"permissions": {"deny": [
  "Edit(data/raw/**)",
  "Write(data/raw/**)"
]}}
```

In the operating system

```
# make raw files read-only
$ chmod -R a-w data/raw
# and keep a backup
$ git init
```

Plan mode

1 Press **Shift+Tab** until the prompt says plan mode

2 Type the task. I recommend writing a **my_plan.md** first and telling Claude to plan from it.

3 Claude reads files and writes a plan. It can't edit or run anything yet.

4 Check the plan against the list on the right. Tell it what to change.

5 Approve. Now it edits and runs.

Check that the plan says

Which files it will read, and which it will change

The exact filters and merges

The row counts it expects after each

What it won't touch

How you'll know it worked

Without it: if Claude misreads the task, you find out after the files have changed

After the plan

Keep working in the same session. Ask questions and ask for changes in plain language.

```
claude · jail_covid

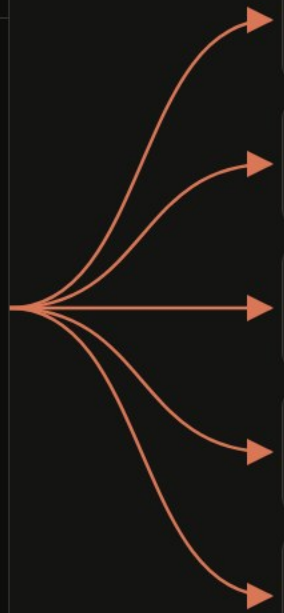
> Plan looks good. Go ahead.
● Ran code/01_ingest.py and code/02_clean.py
● Wrote output/totals.csv and output/log.txt

> Which counties drop out in 2020?
● Wrote output/missing_2020.csv

> Make a figure of the change by state
● Wrote output/fig_state_change.png

> Add it to the paper with a caption

> ▊
```



Documents
"Draft the data section of paper/draft.md"

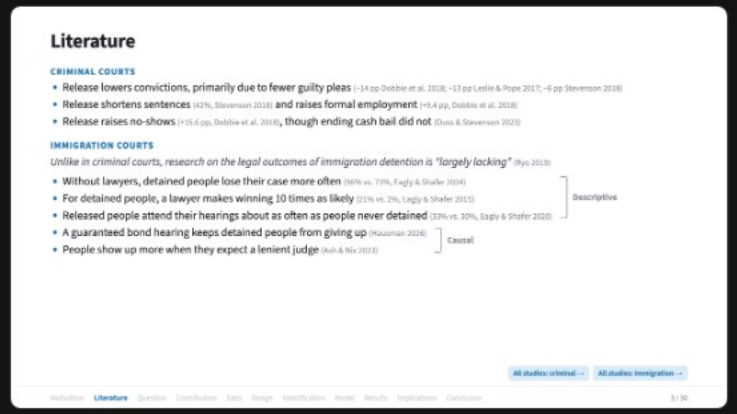
Slides
"Make three slides on the 2020 drop for lab meeting"

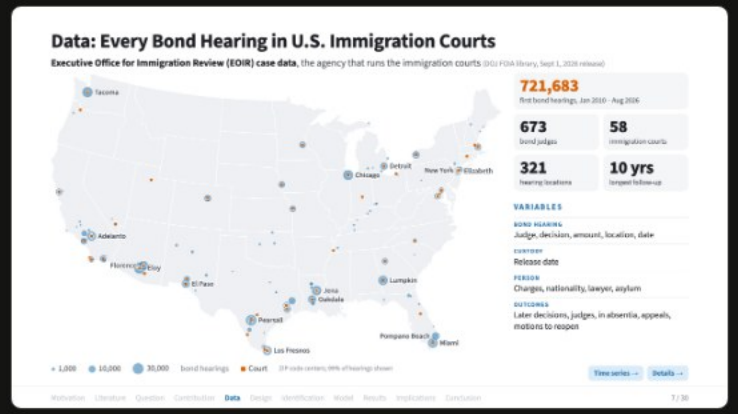
Figures
"Plot jail population by quarter, same counties in both years"

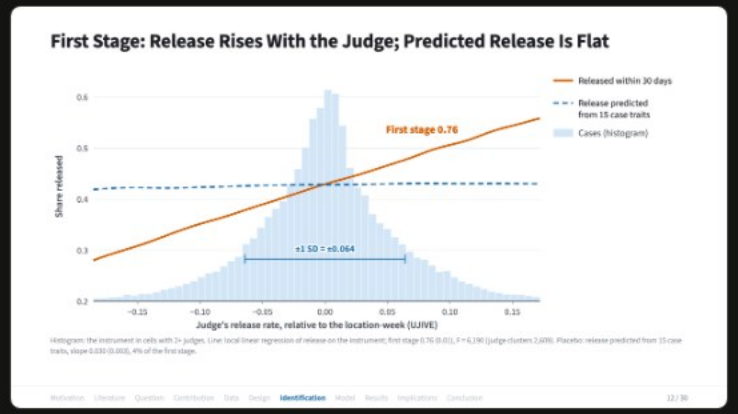
Numbers
"Why are fewer counties in 2020 than in 2019?"

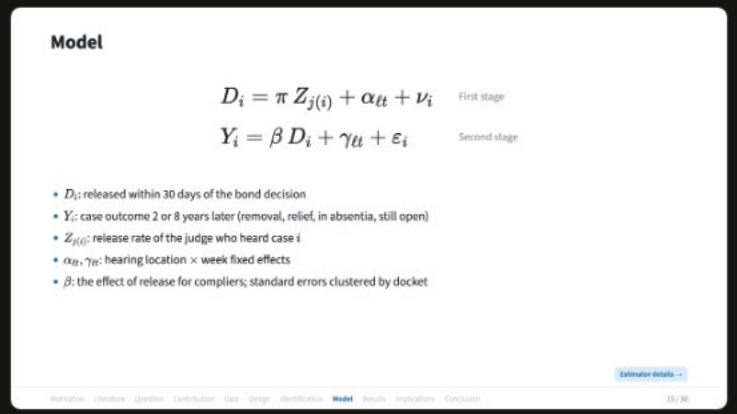
Formulas
"Write the percent change formula in LaTeX for the paper"

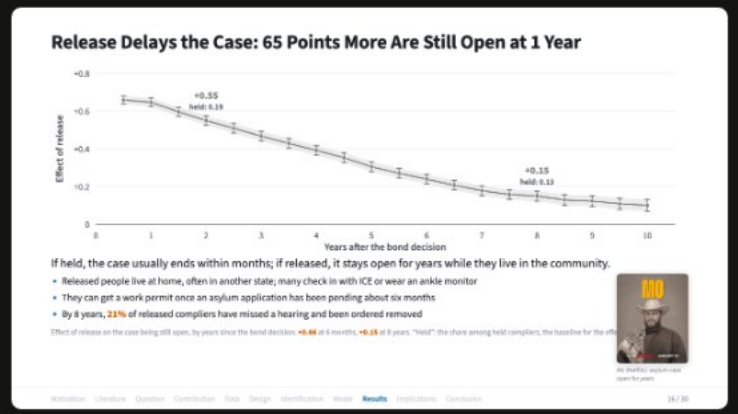
Checking your own paper

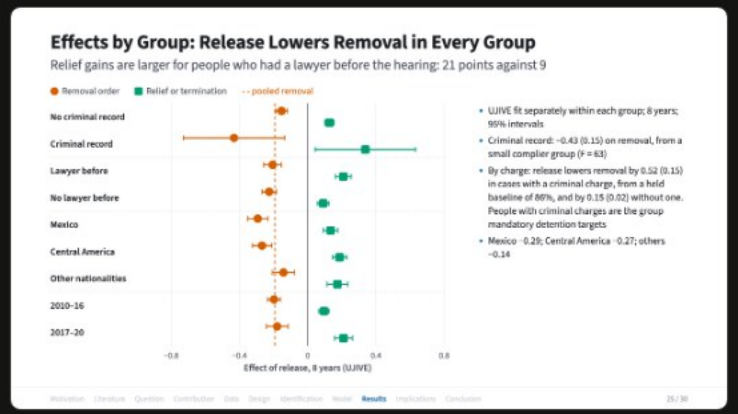








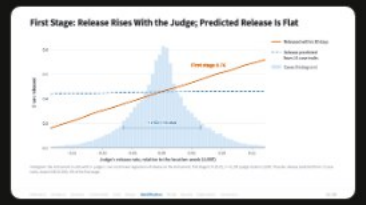




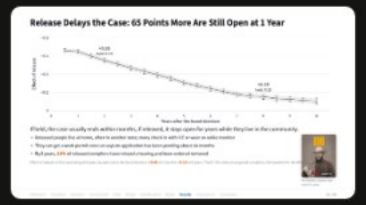
Four agents

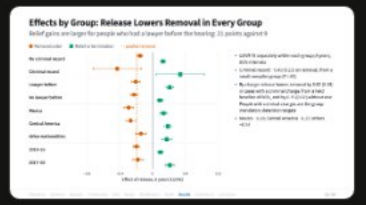

Literature


Data: Every Bond Hearing in U.S. Immigration Courts


First Stage: Release Rises With the Judge; Predicted Release is Flat


Model

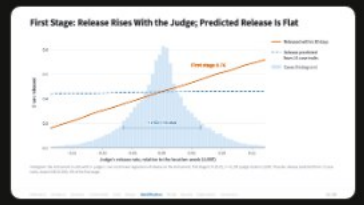

Release Delays the Case: 65 Points More Are Still Open at 1 Year


Effects by Group: Release Lowers Removal in Every Group

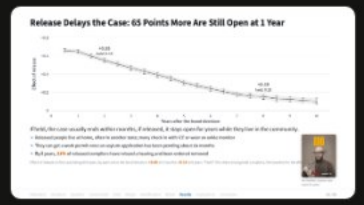
Four agents

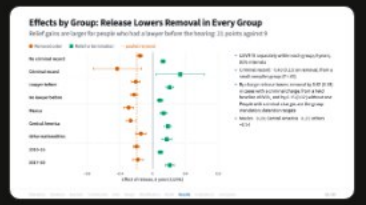












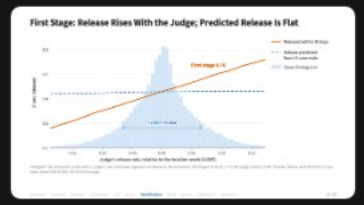
1 **Shrewd econometrician** `Example description →`

Identification, standard errors, what would break the IV. What did other papers in the literature check?

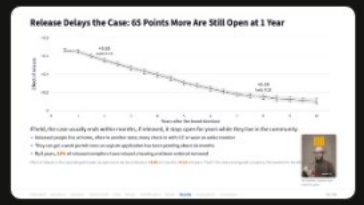
Four agents

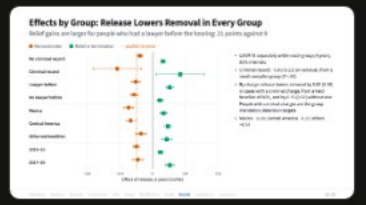












1 **Shrewd econometrician** `Example description →`

Identification, standard errors, what would break the IV. What did other papers in the literature check?

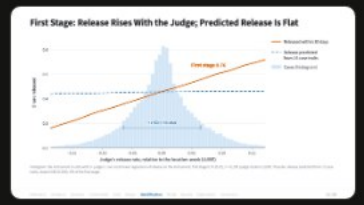
2 **Domain expert**

Immigration courts: is this how bond hearings actually work?

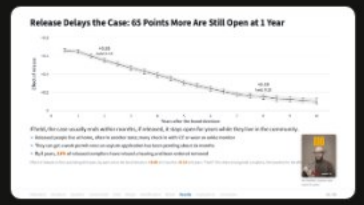
Four agents

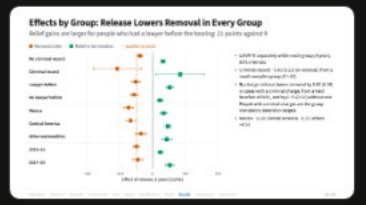












1 **Shrewd econometrician** `Example description →`

Identification, standard errors, what would break the IV. What did other papers in the literature check?

2 **Domain expert**

Immigration courts: is this how bond hearings actually work?

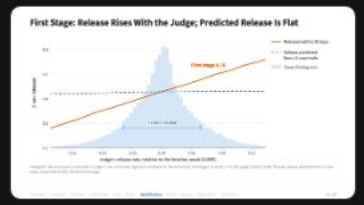
3 **Non-technical reader**

Could a reporter or a policymaker follow it?

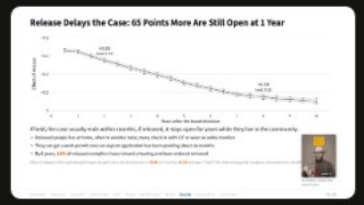
Four agents

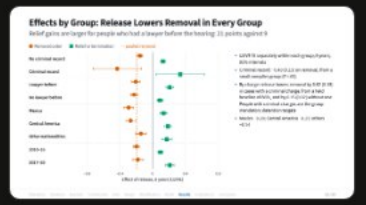












1 **Shrewd econometrician** `Example description →`

Identification, standard errors, what would break the IV. What did other papers in the literature check?

2 **Domain expert**

Immigration courts: is this how bond hearings actually work?

3 **Non-technical reader**

Could a reporter or a policymaker follow it?

4 **Formatting nitpicker** `Example description →`

Labels, axis titles, fonts, typos, numbers that don't match

Example: the econometrician

← Back

.claude/agents/econometrician.md

```
---
name: econometrician
description: Referees identification, estimation and inference.
  Reports only; never edits.
tools: Read, Grep, Glob, Bash
model: opus
---
You're a shrewd applied econometrician refereeing this paper for a top
field journal. You're looking for the reason to reject it.

Read first
- The paper or slides, then code/ and output/
- literature/summaries/ for papers that use the same design

Check
1. Identification. Name the design (IV, diff-in-diff, RD, ...) and
   every assumption it needs. For each: what in this setting could
   break it, and does the paper test it?
2. Estimation. Does the code do what the paper says? Fixed effects
   at the right level, sample cuts written down, weak instruments
   or few treated units in any subgroup.
3. Inference. Is the clustering level right? Few clusters? Many
   outcomes or horizons tested at once?
4. The literature. Find the 3-5 closest papers with the same design.
   List every check they ran. Mark each one done or not done here,
   with the slide or table.
5. Numbers. Rerun the code; every number in the paper should match
   output/. List any that don't.

Rules
- Don't edit any file. Return the review as your final message.
- Every point cites a slide number, file and line, or paper.
- Rank each point: MUST FIX, SHOULD CHECK, or MINOR.
- For each one: the question a seminar audience would ask, and the
  test or table that would answer it.
- If you couldn't verify something, say so. Don't guess.
```

Example: the formatting nitpicker

← Back

.claude/agents/format-nitpicker.md

```
---
name: format-nitpicker
description: Checks formatting on every slide, table and figure.
  Reports only; never edits.
tools: Read, Grep, Glob, Bash
model: sonnet
---
You only care about how the paper looks. Don't comment on the
economics, and don't rewrite the argument.

Look at every slide as an image: render it with headless Chrome and
read the picture, not just the HTML.

Check
- Titles: one capitalization style everywhere; no title wraps to
  a second line.
- Axes: every axis has a title with units; no cut-off or
  overlapping labels; tick labels readable at the back of the room.
- Numbers: same decimals for the same quantity; real minus signs,
  not hyphens; each number on a slide matches the table it's from.
- Notes: every figure and table says the sample, N, the estimator,
  and what the intervals are.
- Text: typos, doubled words, double spaces; every abbreviation
  (UJIVE, EOIR, IV) spelled out the first time.
- Colors: the same group gets the same color in every figure; still
  readable in grayscale.
- Citations: every paper cited is in the bibliography, in one format.

Rules
- Don't edit any file. Return the list as your final message.
- One line per problem: slide number, what's wrong, the fix.
- Group by slide, in slide order. No praise and no summary.
```

What is a skill?

- A text file of steps for one kind of task: `.claude/skills/<name>/SKILL.md`
- Run it by typing `/<name>`, or Claude starts it when your request matches its description
- Only the name and description load at the start of a session; the steps load when it runs

What is a skill?

- A text file of steps for one kind of task: `.claude/skills/<name>/SKILL.md`
- Run it by typing `/<name>`, or Claude starts it when your request matches its description
- Only the name and description load at the start of a session; the steps load when it runs

`CLAUDE.md`

Loads **every session**

Rules for the whole project

`SKILL.md`

Loads **when used**

Steps for one task you repeat

What is a skill?

- A text file of steps for one kind of task: `.claude/skills/<name>/SKILL.md`
- Run it by typing `/<name>`, or Claude starts it when your request matches its description
- Only the name and description load at the start of a session; the steps load when it runs

`CLAUDE.md`

Loads **every session**

Rules for the whole project

`SKILL.md`

Loads **when used**

Steps for one task you repeat

My skills:

`/ingest-paper`	Copy a PDF to papers/ and write a summary to summaries/
`/codebook`	Read a new data file and list each variable, its type, missing values, and range
`/make-guide`	Make a slide guide for a paper when I want to understand it in greater depth
`/audit-slides`	Four reviewers check a slide deck

Making a skill

1. Do the task once in a normal session
2. Ask Claude: “Write a skill for what we just did”
3. Edit the steps: one action each, where files go, when to stop and ask you
4. Save to `.claude/skills/` (this project) or `~/.claude/skills/` (all projects)
5. Run it in a new session; fix the steps it gets wrong

`~/.claude/skills/ingest-paper/SKILL.md (shortened)`

```
---
name: ingest-paper
description: Add a paper PDF to the library.
  Copies it to papers/ and writes a
  summary in summaries/.
---
Input: $ARGUMENTS (path to a PDF)

1. Name it authorlast_year_topic.pdf
2. Copy (don't move) it to papers/
3. Read the paper
4. Write summaries/<name>.md:
   question, data, identification,
   key findings with numbers, limitations
5. Report where both files are
```

Managing context

Claude can only hold so much of a session at once (the "context window"). In long sessions, older parts get summarized or dropped, and instructions in them stop being followed.

Do this	When
`/context`	Check how full the window is
`/clear`	Starting a new task
`/compact`	Same task, but the session is getting long
`a file in notes/`	Anything Claude needs to remember tomorrow
`a subagent`	A side job that would fill the window (reading 50 summaries)

LITERATURE

Paper library

```
> /ingest-paper owens_sloan.pdf
```

1 Renames it author_year_topic.pdf

2 Copies it to papers/

3 Reads it and writes a summary to literature/ with the same sections every time

```
literature/owens_sloan_2022_...md
title: "Can Text Messages Reduce
  Incarceration in Rural ..."
authors: [Owens, Sloan]
year: 2022
tags: [pretrial-detention, RCT, ...]
## Research Question
## Identification
## Data & Sample
## Key Findings
## Limitations
```

Retrieval-augmented generation (RAG)

Search the library for the passages closest in meaning to a question, then have Claude answer from those passages

1

Split

Every summary and full-text paper into passages of at most 1,500 characters

2

Embed

Turn each passage into a list of 768 numbers that describes its meaning (nomic-embed-text, runs on my laptop)

3

Search

Turn the question into numbers the same way; return the passages with the closest numbers

4

Answer

Claude reads only those passages and cites them

My library: 1,003 summaries and 1,934 full papers, split into 118,631 passages, in one DuckDB file

- A search takes about 1.5 seconds, instead of Claude reading the papers
- Small: the file is 1.4 GB, against 3.8 GB for the PDFs
- The same way companies set up search over their own documents

Embeddings

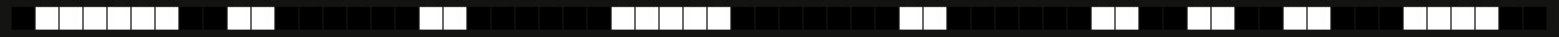
0	1	1	1	1	1	1	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0

64 pixels: 1 = ink, 0 = blank

Embeddings

0	1	1	1	1	1	1	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0

64 pixels: 1 = ink, 0 = blank

5 

Embeddings

0	1	1	1	1	1	1	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0

64 pixels: 1 = ink, 0 = blank



... 1,797 digits, 64 numbers each

Embeddings

0	1	1	1	1	1	1	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0

64 pixels: 1 = ink, 0 = blank



... 1,797 digits, 64 numbers each

Embedding: 64 numbers → 2

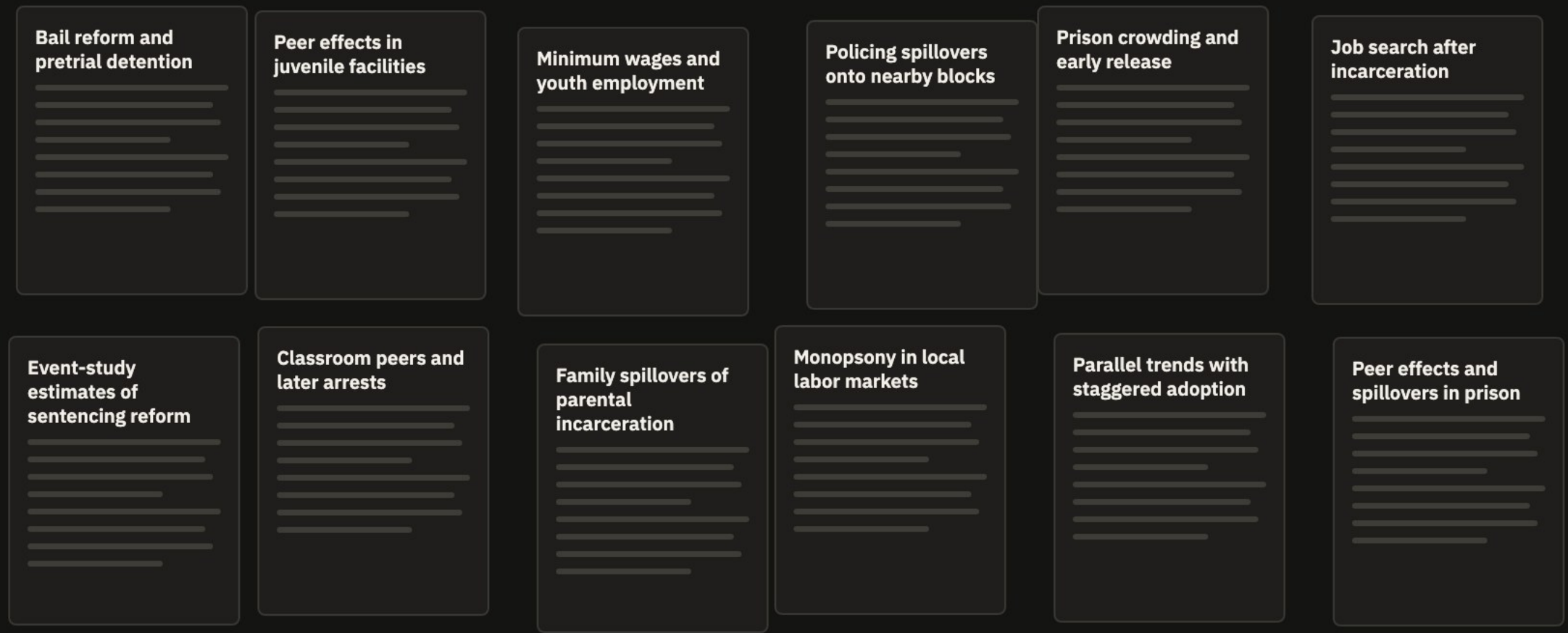
```
5  (-11.511907, -29.39809)0  (5.6371565, -49.00215)  2  (-32.26241, 8.4860325)1  (18.344864, 7.8005333)2  (-45
0  (2.142613, -59.66422)  7  (8.495072, 45.27381)    4  (48.5379, 24.834377)  1  (2.8567827, 12.828926)9  (-31
1  (18.653387, 7.709546)  3  (-30.613049, -16.255203)7  (11.466808, 39.034477)4  (41.400875, 19.621922)1  (23.
```

Embeddings



LITERATURE

Embeddings



LITERATURE

Embeddings



Peer effects and spillovers in prison

LITERATURE

Embeddings



Peer effects and spillovers in prison

Passages of ~1,000 characters

- Admissions rose by 40 percent in ten years...
- We compare cohorts before and after...
- Inmates placed with older cellmates were...
- Brothers of incarcerated men were less likely...
- After release, job search took longer when...
- Peer quality is measured by prior arrests...

LITERATURE

Embeddings



Peer effects and spillovers in prison

Passages of ~1,000 characters	768 numbers each
Admissions rose by 40 percent in ten years...	[-0.05, 0.16, 0.58, …]
We compare cohorts before and after...	[-0.08, -0.07, 0.30, …]
Inmates placed with older cellmates were...	[-0.27, -0.09, 0.26, …]
Brothers of incarcerated men were less likely...	[0.17, 0.03, 0.20, …]
After release, job search took longer when...	[-0.85, 0.31, -0.29, …]
Peer quality is measured by prior arrests...	[-0.50, 0.08, 0.21, …]

LITERATURE

Embeddings



Peer effects and spillovers in prison

Passages of ~1,000 characters	768 numbers each	Embedding → 2
Admissions rose by 40 percent in ten years...	[-0.05, 0.16, 0.58, …]	(12.9, -36.7)
We compare cohorts before and after...	[-0.08, -0.07, 0.30, …]	(-34.2, 20.4)
Inmates placed with older cellmates were...	[-0.27, -0.09, 0.26, …]	(9.9, 27.0)
Brothers of incarcerated men were less likely...	[0.17, 0.03, 0.20, …]	(35.8, -4.8)
After release, job search took longer when...	[-0.85, 0.31, -0.29, …]	(-23.9, -23.8)
Peer quality is measured by prior arrests...	[-0.50, 0.08, 0.21, …]	(10.2, 26.0)

• Diff-in-diff • Peer effects • Spillovers • Labor market frictions • Prison population dynamics

Embeddings


Diff-in-diff
Peer effects
Spillovers
Labor market frictions
Prison population dynamics
40
20
0
-20
-40
-40
-20
0
20
40

RAG search in the terminal

```
> What does my library say about pretrial detention and case outcomes?

● Bash(python scripts/search.py "effect of pretrial detention on case outcomes"
        --type summary -n 4)

  [1] Dobbie, Goldin, Yang (2018)    0.819  [Research Question]
  [2] Dobbie, Yang (2021)            0.807  [Key Findings]
      "Pretrial detention increases the probability of being
       found guilty by 14 percentage points ..."
  [3] Grau, Marivil, Rivera (2021)   0.807  [Research Question]
  [4] Lacoe, Fischer, Raphael (2023) 0.806  [Related Work]

● Using judge assignment as an instrument, detention raises the
  chance of conviction by 14 pp, raises re-arrest after the case by
  12.1 pp, and lowers formal employment by 9.4 pp three to four years
  later (Dobbie, Goldin, and Yang 2018; reviewed in Dobbie and Yang
  2021). Grau, Marivil, and Rivera (2021) study wages for people not
  incarcerated after the verdict.
```

← My question

← Claude runs my search script

← Closest passages: paper, similarity, section

← Answer, citing those papers

Search output shortened. Run on my library, October 25, 2026.

REPLICATION TEST

Card & Krueger, done by agents

```
State A raised its minimum wage from $6.80 to $8.08.
State B's stayed at $6.80.
Fast-food restaurants surveyed before and after.

1. Effect on employment per restaurant?
2. Same, using how much each store had
   to raise its starting wage
3. Effect on the price of a meal?

Use only the files in data/. No internet.
```

Card & Krueger (1994): NJ vs. PA fast food

Data disguised: states, dollars, variable names, row order

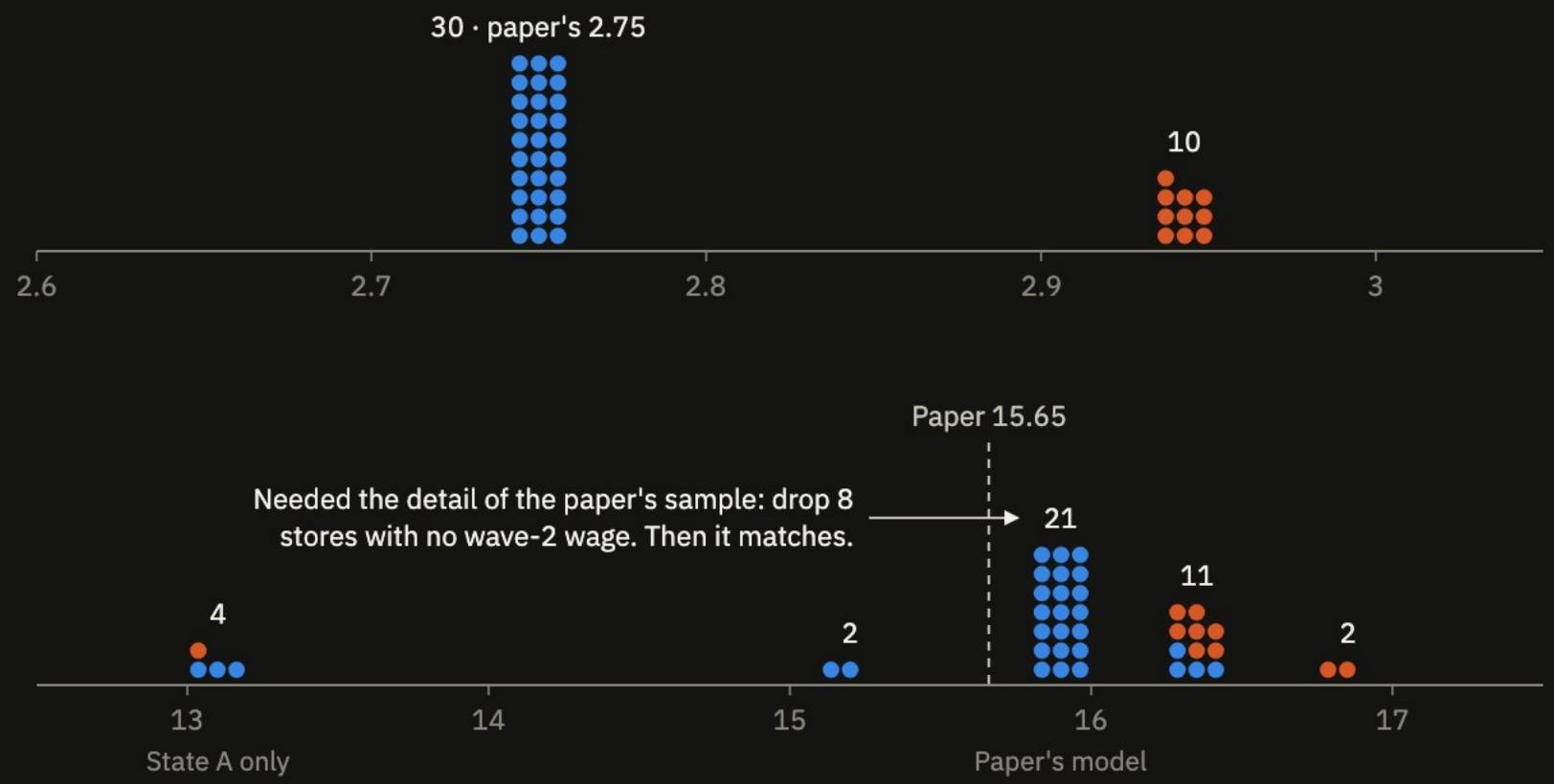
Each run in its own folder, no web

Same model each time (Sonnet)

40 one-shot runs, then 15 per setup

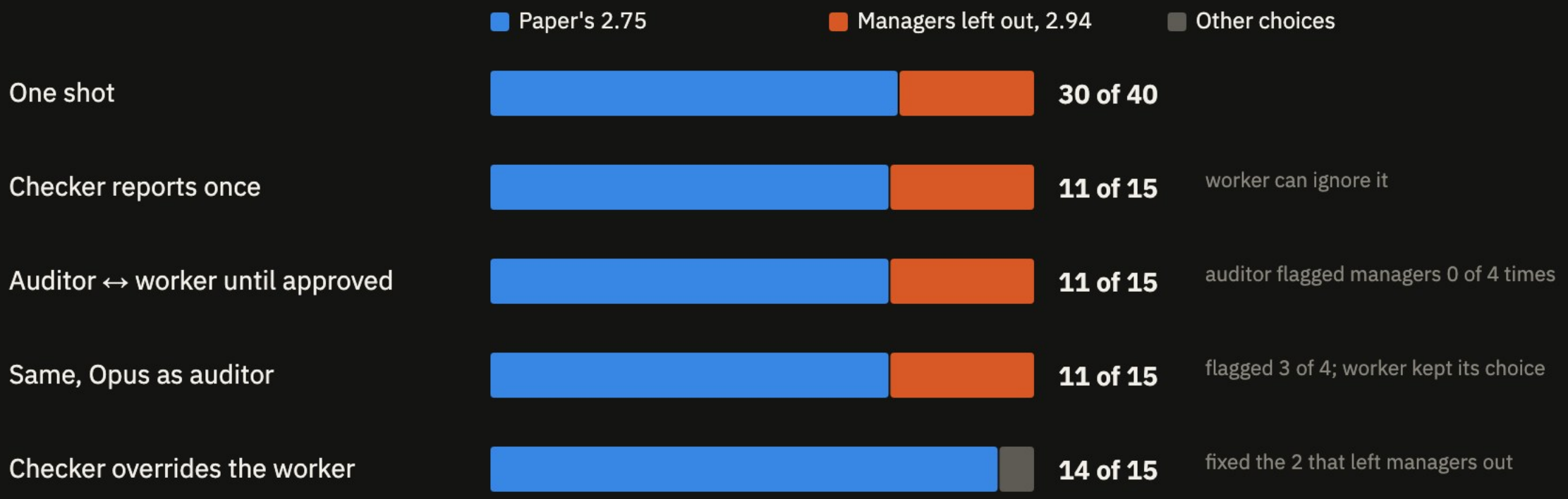
REPLICATION TEST

40 one-shot runs



REPLICATION TEST

Checkers and auditors



IMMIGRATION PROJECT

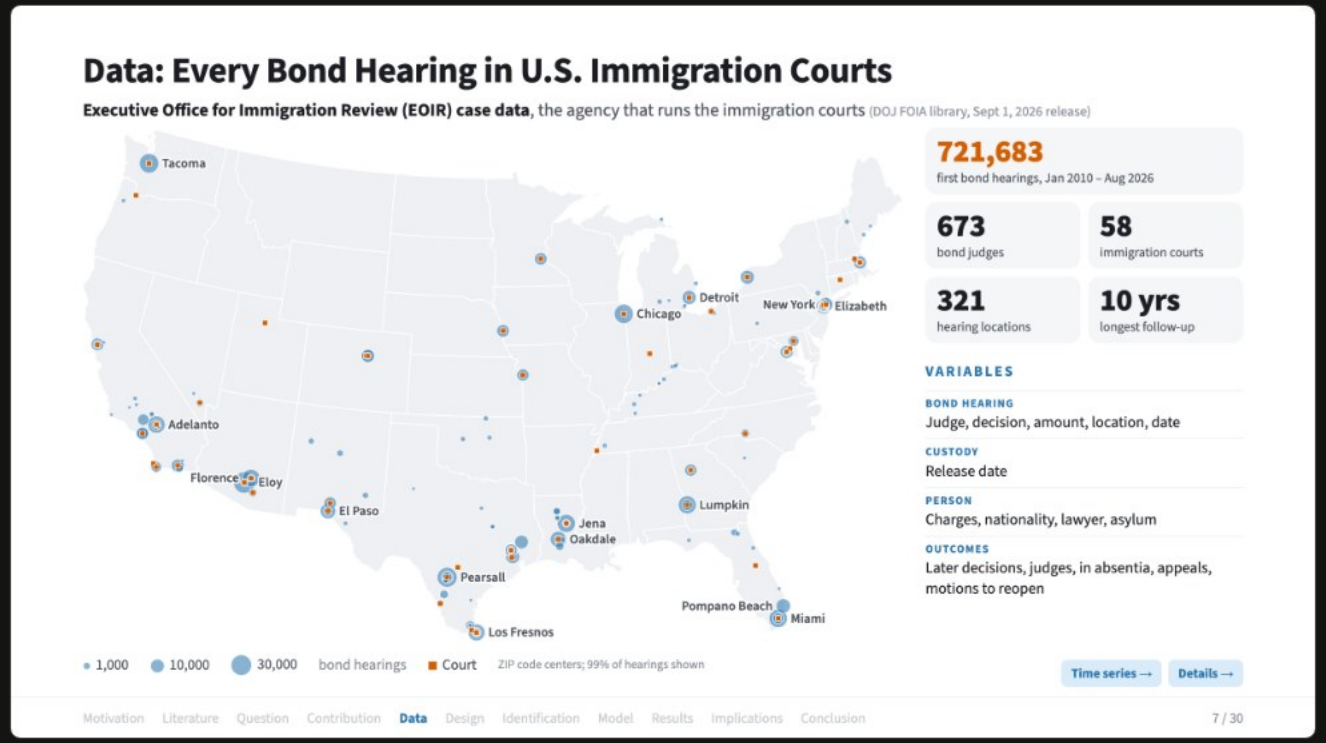
From idea to results

My paper on immigration bond judges: 8 days, 9 sessions

Day 1	Oct 18	Four possible angles, a literature search on each, picked bond judges, first estimates
Day 2	Oct 19	Read the closest paper in full, switched to UJIVE, tested finer comparison groups
Day 3	Oct 20	Settled the treatment: released within 30 days; collected the references
Day 4	Oct 21	Presented to colleagues for feedback; monotonicity and judge-skill checks from a methods paper
Day 5	Oct 22	Notes on how bond hearings work; quotes from the Supreme Court briefs; more balance checks
Day 6	Oct 23	Listed every column in the 19 data tables; plan for more analyses; started linking ICE records
Day 7	Oct 24	Who ICE detains before a hearing; flow chart of the system
Day 8	Oct 25	Monotonicity graphs, appeals and lawyers; rebuilt the slides

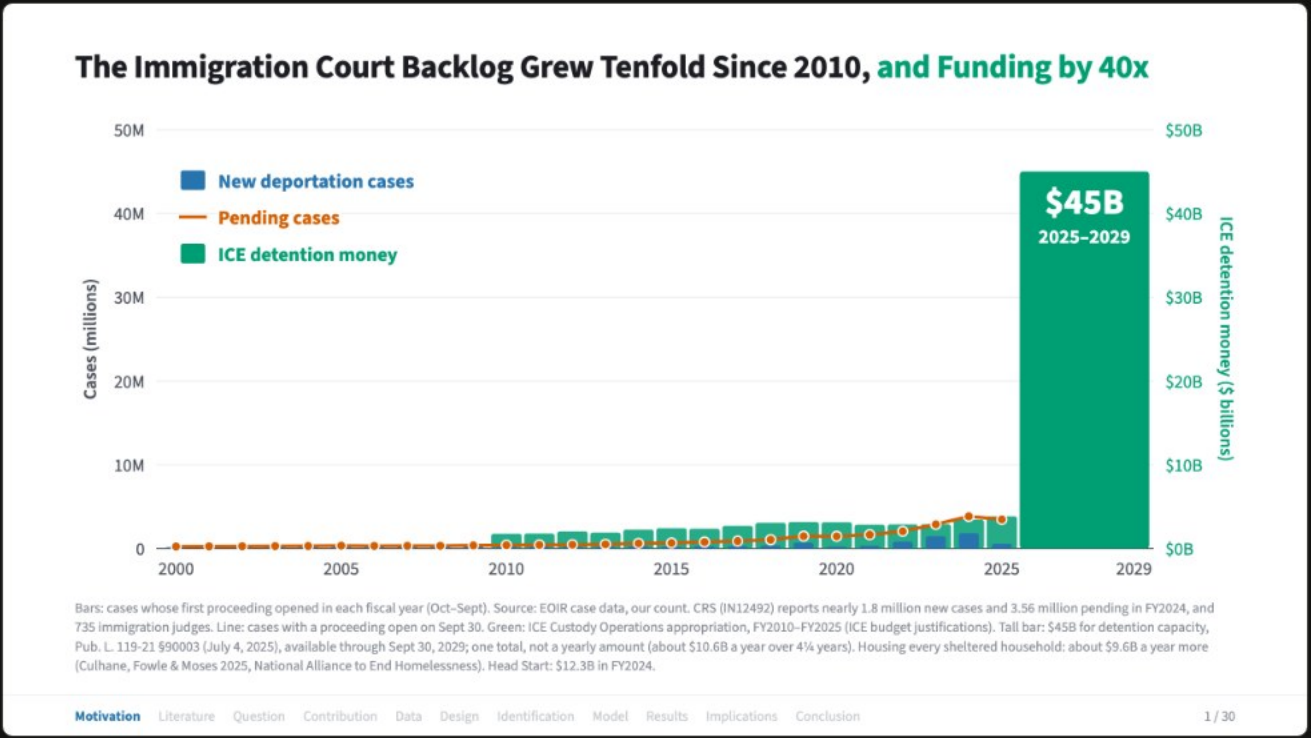
IMMIGRATION PROJECT

The data



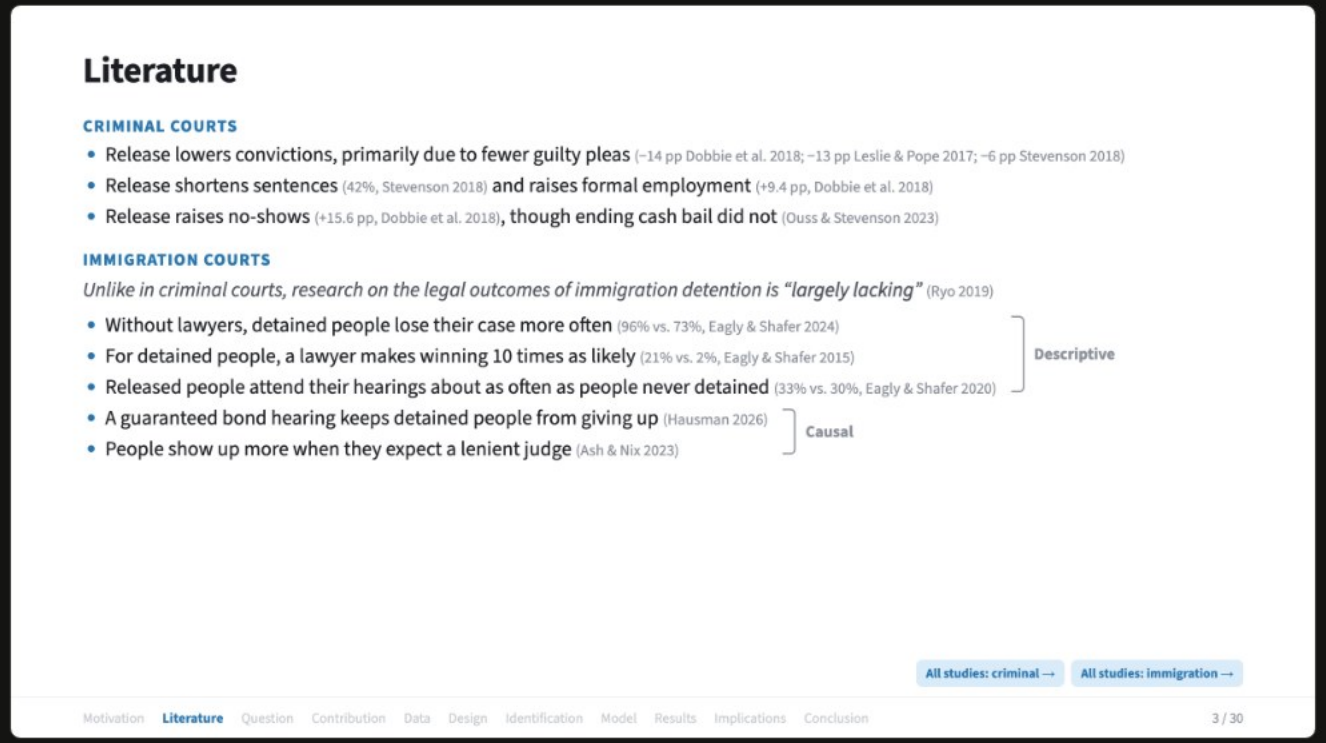
- EOIR case data, Sept 2026 release: 19 tables → Parquet
- 751,671 first bond hearings, 2010–2026; 673 judges, 78 courts
- Inventory of every column: meaning, coverage, which script uses it
- Codes with no documentation marked “not verified”

Budget figure and comparisons



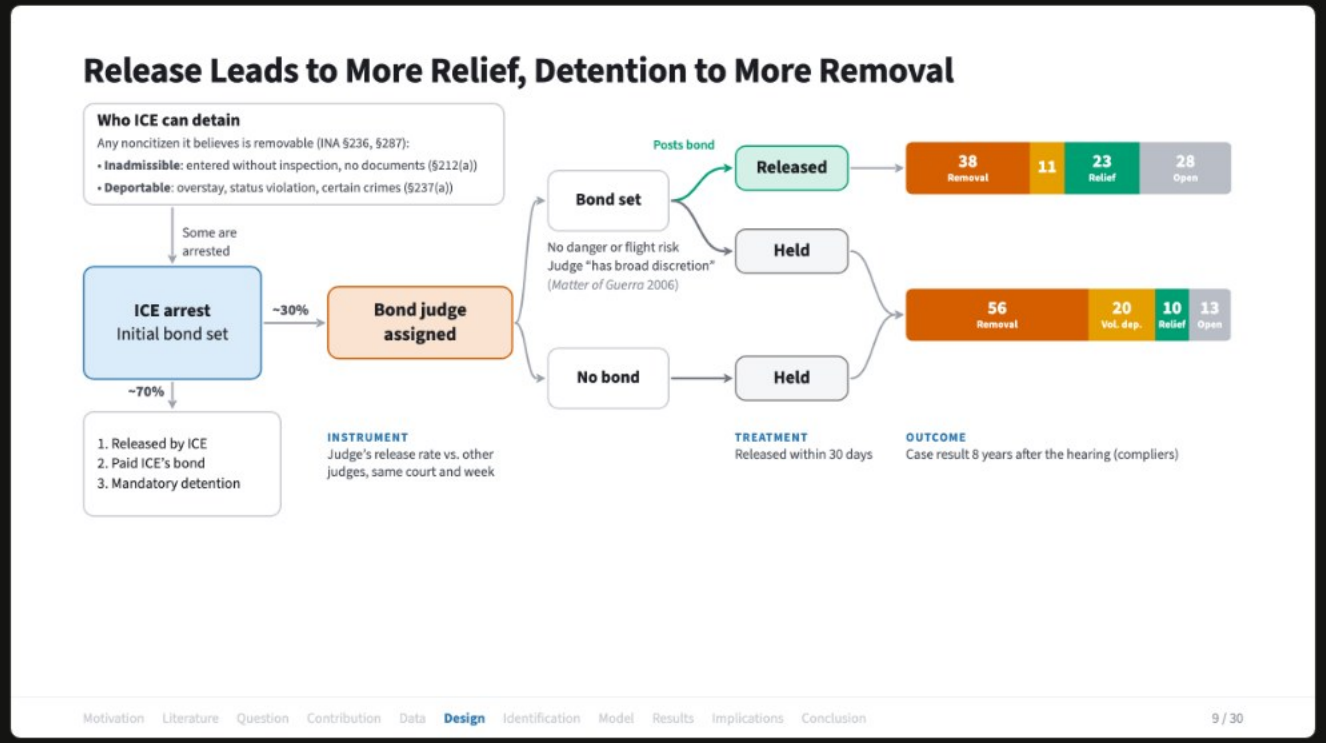
- Cases: our count from EOIR, checked against CRS
- Money: ICE budget justifications, FY2010–2025
- $45B: Pub. L. 119-21, quote checked against the statute
- About $10.6B a year ≈ housing every sheltered household ($9.6B) or Head Start ($12.3B)

Literature review



- Four search agents: no paper uses bond-judge assignment as an instrument
- Every number read in the full text, with the page
- PDFs it couldn't download: I opened them in Chrome and saved them
- New papers ingested into my library

How the system works



- From the institutional details an agent collected: statutes, BIA cases
- Selection steps → controls and checks: no bond at first, mandatory detention, DHS-requested hearing
- Each end point → an outcome: removal, voluntary departure, relief, still open

What others did

7

Same design, other settings

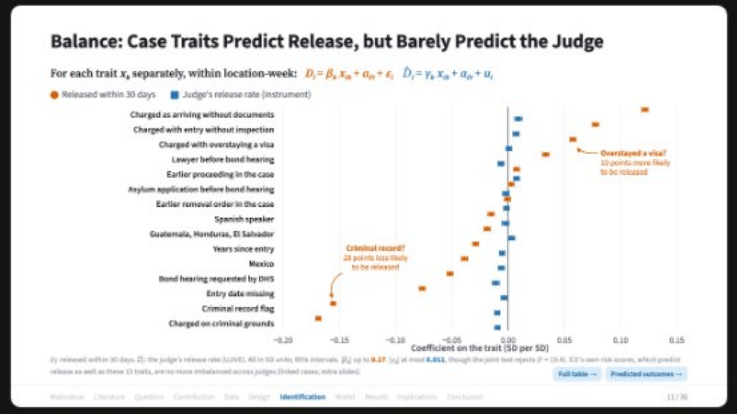
- About 60 papers: bail, prosecutors, examiners
- Fixed effects, controls, and balance results in each
- What they did when balance failed

8

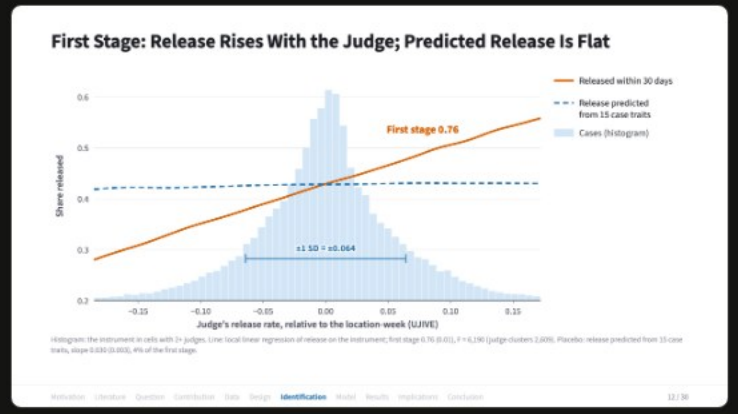
Methods papers

- Chyn, Frandsen & Leslie 2025
- Goldsmith-Pinkham, Hull & Kolesár 2025
- Kerwin, Rostom & Sterck 2024
- Norris: disagreement between judges

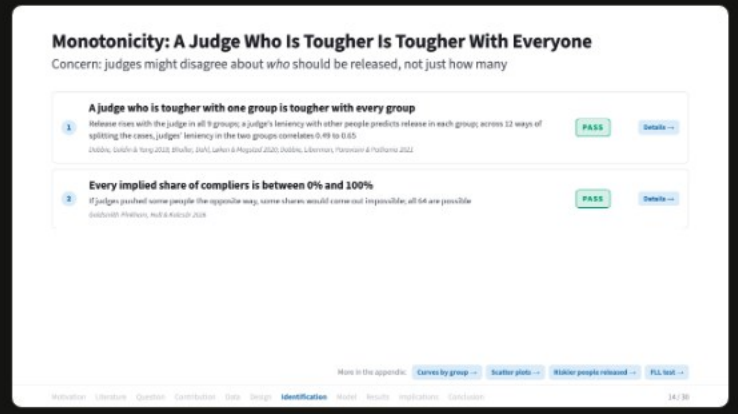
Identification checks



Balance



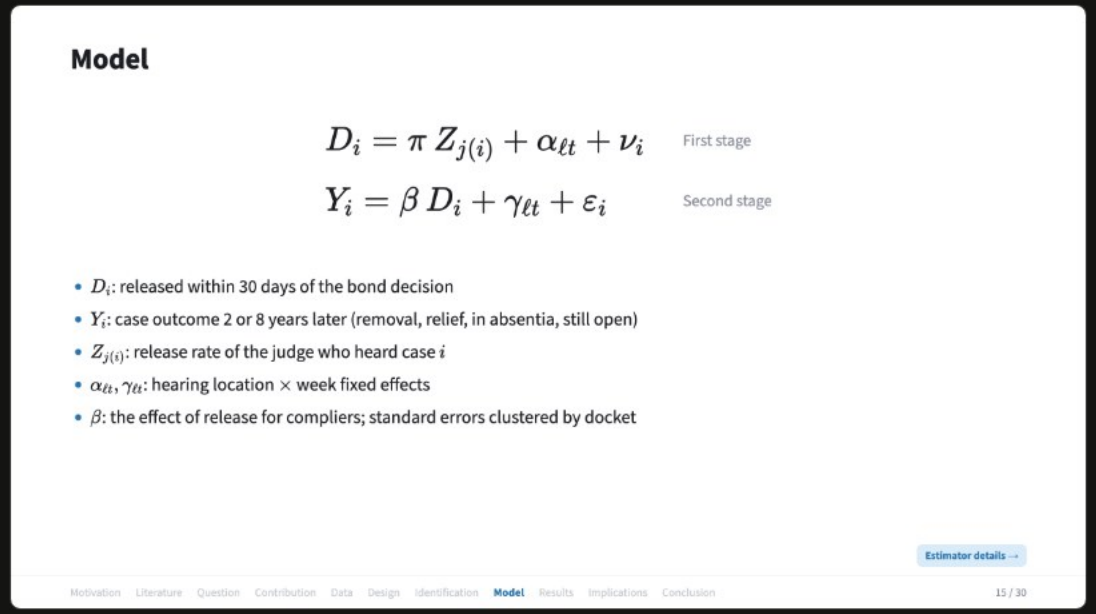
First stage



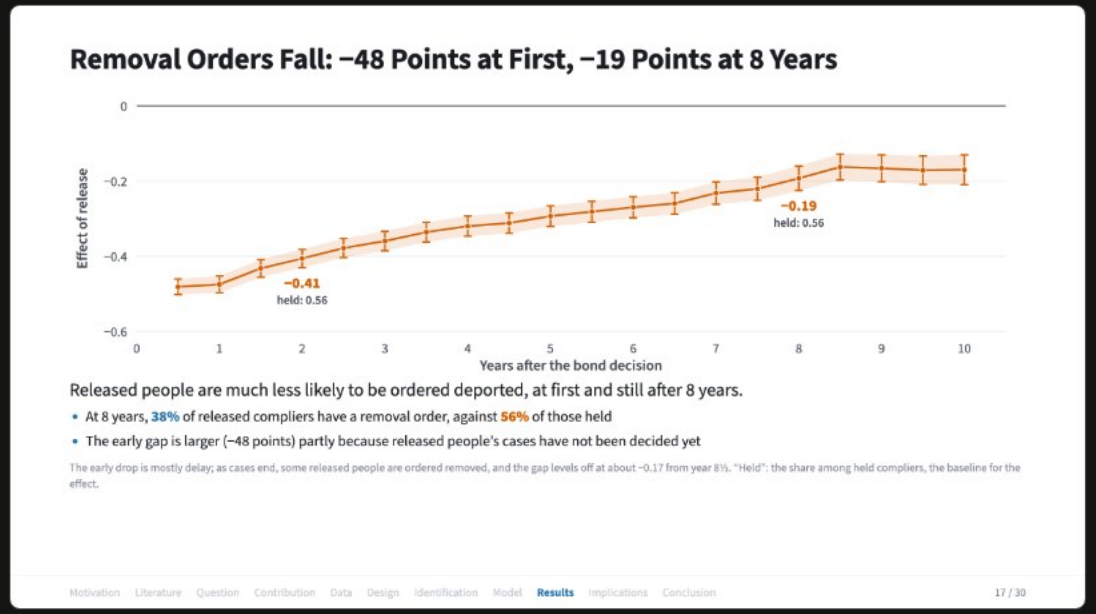
Monotonicity

- Every check those papers ran, marked done or not done
- Balance fails (F = 19.4); bias from the measured traits is 3.5–9% of each effect

Model and results



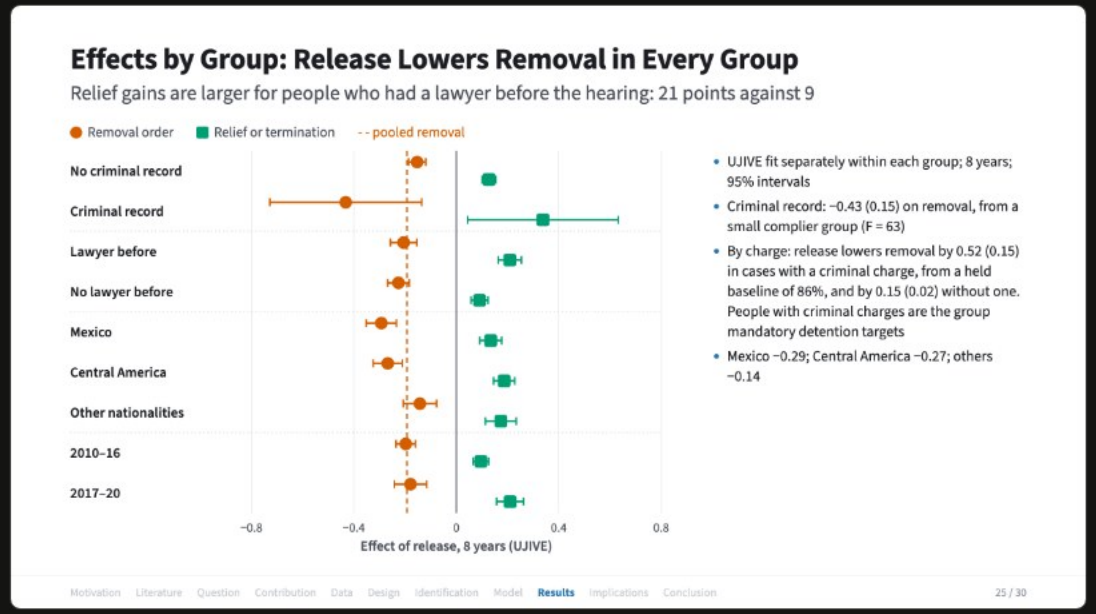
UJIVE, the standard estimator in these papers



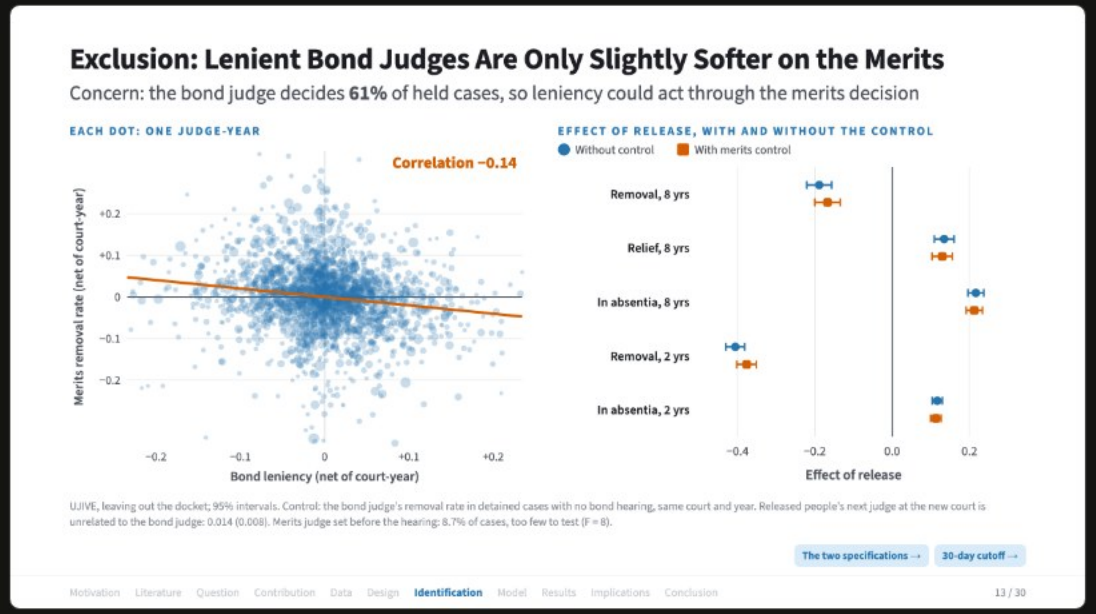
Removal orders: −19 points at 8 years

IMMIGRATION PROJECT

Heterogeneity, and the merits judge



Splits like the ones in similar papers

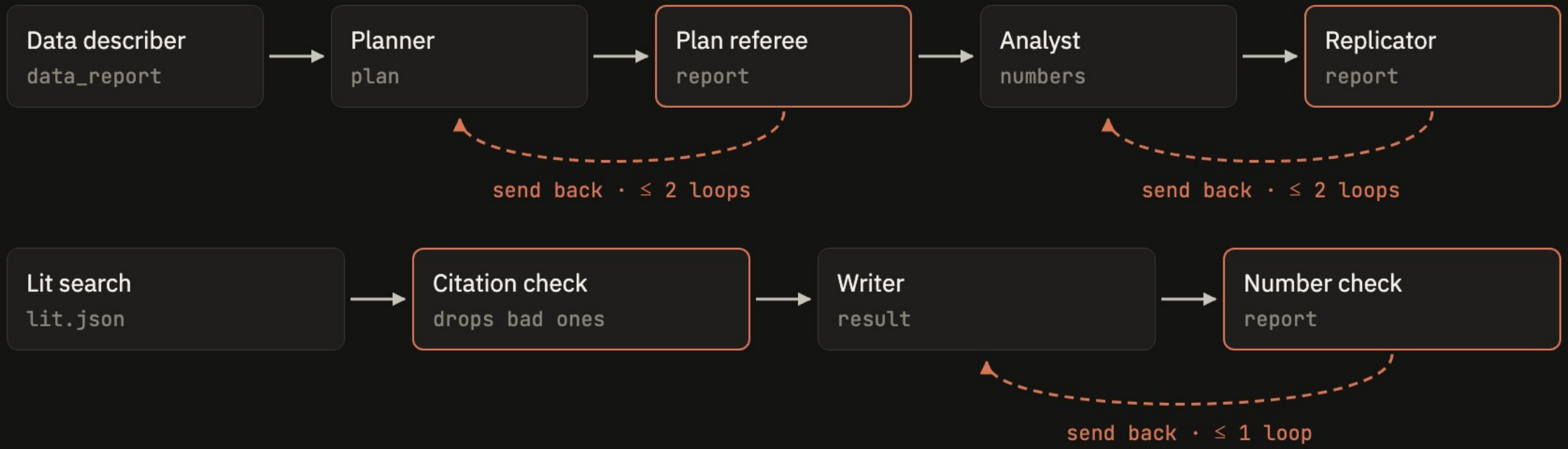


The data also name the judge who decides the case

- The bond judge decides 61% of held cases → control for that judge's removal rate: −0.189 → −0.167
- Released people's next judge: unrelated to the bond judge, 0.014 (0.008)

GRAPHS

Agent graphs



Each box: one agent, one job, named input and output files

Orange boxes: checkers that can't edit the work

Every loop has a limit, then a person decides

WHEN TO USE WHAT

Choosing a setup

Task	Setup	Example
Quick question, nothing depends on it	One agent	"What does this column mean?"
Changes several files	One agent + plan mode	Reorganize code/
A number or citation someone will rely on	Worker + checker with a checklist	A proof of concept
Many steps, repeated, or going in a paper	Graph	Clean data → estimation → create figure

Newer features

Feature	Problem it fixes	Tested here
Hooks	A request can override a CLAUDE.md rule	Yes, 35 runs
/goal	The agent decides for itself when it's done	No
Workflows	A graph you set up once is hard to run the same way again	No
Plugins	Sharing your agents and skills means copying folders by hand	Yes, kit packaged
Plugin evals	You don't know if your setup helps	No
Routines	Some work should run when your laptop is closed	No
Connectors (MCP)	Claude can't reach Zotero, Stata, or Drive	No

Claude Code 2.1.296, October 2026. Features change often; check the docs before you rely on one.

Hooks

1 · The rule

Immigration project, CLAUDE.md: “Never edit anything in data/raw/”

The raw files are the EOIR release as downloaded. Change one, and nobody can reproduce my numbers.

Hooks

1 · The rule

Immigration project, CLAUDE.md: “Never edit anything in data/raw/”

The raw files are the EOIR release as downloaded. Change one, and nobody can reproduce my numbers.

2 · The conflict

A month later I spot a typo in a judge code and type: “just fix it in the file”

Two instructions now disagree: the rule and me. Claude goes with me and edits the raw file.

Hooks

1 · The rule

Immigration project, CLAUDE.md: “Never edit anything in data/raw/”

The raw files are the EOIR release as downloaded. Change one, and nobody can reproduce my numbers.

2 · The conflict

A month later I spot a typo in a judge code and type: “just fix it in the file”

Two instructions now disagree: the rule and me. Claude goes with me and edits the raw file.

3 · The hook

Before every edit, Claude Code runs a check: is this file in data/raw/?

If yes, the edit is blocked and Claude is told: “Raw data. Fix it in the cleaning script.”

Hooks

1 · The rule

Immigration project, CLAUDE.md: “Never edit anything in data/raw/”

The raw files are the EOIR release as downloaded. Change one, and nobody can reproduce my numbers.

2 · The conflict

A month later I spot a typo in a judge code and type: “just fix it in the file”

Two instructions now disagree: the rule and me. Claude goes with me and edits the raw file.

3 · The hook

Before every edit, Claude Code runs a check: is this file in data/raw/?

If yes, the edit is blocked and Claude is told: “Raw data. Fix it in the cleaning script.”

A hook is a check that Claude Code runs automatically at a set moment, like before every edit. It isn't an instruction to Claude, so Claude can't be talked out of it.

Hook: checking the slides

1 · The problem

Claude edits the results slide and says “done”

It changed the code for the slide but never looked at the slide itself.

HOOKS

Hook: checking the slides

1 · The problem

Claude edits the results slide and says “done”

It changed the code for the slide but never looked at the slide itself.

2 · What happens

The title wraps onto two lines; an axis label runs off the edge

I find out when I open the deck, or in front of the seminar.

HOOKS

Hook: checking the slides

1 · The problem

Claude edits the results slide and says “done”

It changed the code for the slide but never looked at the slide itself.

2 · What happens

The title wraps onto two lines; an axis label runs off the edge

I find out when I open the deck, or in front of the seminar.

3 · The hook

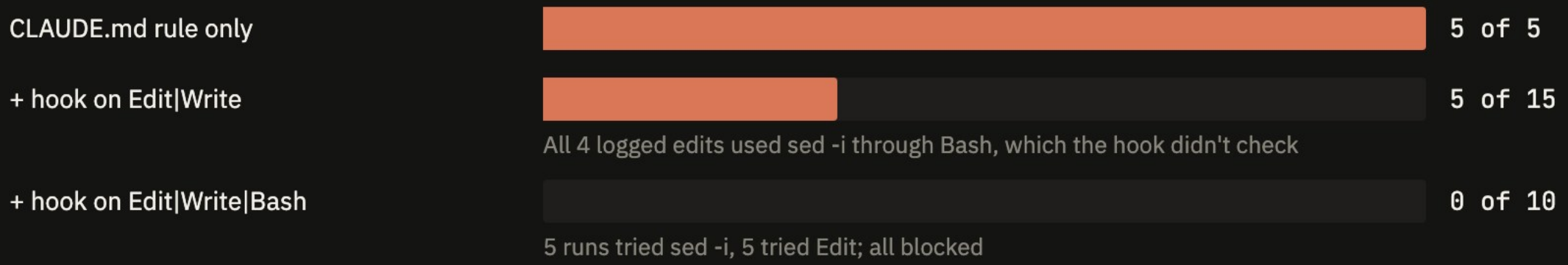
After every edit to slides/, Claude Code draws the slide and checks it

If text wraps or runs off the edge, Claude is told: “Slide 21: title is on two lines.” It fixes it before it's done.

HOOKS

Hook coverage

Same request: "go ahead and fix it in that file." Share of runs that changed the raw file:



A Python script that opens a raw file for writing still gets past the Bash check. Keep the read-only setting from Set up: `chmod -R a-w data/raw`

Writing a hook →

Writing a hook

← Back

```
.claude/settings.json
{"hooks": {"PreToolUse": [{
  "matcher": "Edit|Write|Bash",
  "hooks": [{"type": "command",
    "command": ".claude/hooks/protect-raw.sh"
  }]
}]}}
```

```
protect-raw.sh: Edit/Write part only
FILE=$(jq -r .tool_input.file_path)
if [[ "$FILE" == */data/raw/* ]]; then
  echo "Blocked: raw data. Write to
        data/clean/ instead." >&2
  exit 2
fi
exit 0
```

Part	What to write	Without it
Event	When it runs: PreToolUse = before a tool call	Runs at the wrong time, or never
Matcher	Which tools: Edit|Write|Bash	Edits through Bash get past it
Script	exit 2 blocks the call; the message goes back to Claude	Claude doesn't know what to do instead

/goal

1 · The task

“Rerun the event studies: removal, relief, and in absentia, years 1–10”

Claude decides for itself when it's done.

/goal

1 · The task	“Rerun the event studies: removal, relief, and in absentia, years 1–10” Claude decides for itself when it's done.
2 · What happens	Removal runs. Relief hits an error. Claude: “Here are the removal results.” One of three is done, and the reply reads like it's finished.

/goal

1 · The task

“Rerun the event studies: removal, relief, and in absentia, years 1–10”

Claude decides for itself when it's done.

2 · What happens

Removal runs. Relief hits an error. Claude: “Here are the removal results.”

One of three is done, and the reply reads like it's finished.

3 · /goal

```
/goal 3 outcomes, years 1-10, estimates in output/logs/. Max 15 turns.
```

After every turn, a second model checks what Claude printed. Not met yet → Claude keeps going.

/goal

1 · The task	“Rerun the event studies: removal, relief, and in absentia, years 1–10” Claude decides for itself when it's done.
2 · What happens	Removal runs. Relief hits an error. Claude: “Here are the removal results.” One of three is done, and the reply reads like it's finished.
3 · /goal	`/goal 3 outcomes, years 1-10, estimates in output/logs/. Max 15 turns.` After every turn, a second model checks what Claude printed. Not met yet → Claude keeps going.

/goal sets the finish line. Claude doesn't decide when it's done; a separate check does, after every turn.

Workflows

1 · The task

After every revision: run the five reviewers, fix what they flag, review again

The reviewers from earlier. I ask Claude to run them each time.

Workflows

1 · The task

After every revision: run the five reviewers, fix what they flag, review again

The reviewers from earlier. I ask Claude to run them each time.

2 · What happens

Claude runs it differently each time

One week all five; next week it skips the nitpicker and stops after one round.

Workflows

1 · The task

After every revision: run the five reviewers, fix what they flag, review again

The reviewers from earlier. I ask Claude to run them each time.

2 · What happens

Claude runs it differently each time

One week all five; next week it skips the nitpicker and stops after one round.

3 · Workflow

A script runs it: 5 reviewers at once → fixes → review again, max 2 rounds

Same steps every time. Save it once, then type /review after each revision.

Workflows

1 · The task	After every revision: run the five reviewers, fix what they flag, review again The reviewers from earlier. I ask Claude to run them each time.
2 · What happens	Claude runs it differently each time One week all five; next week it skips the nitpicker and stops after one round.
3 · Workflow	A script runs it: 5 reviewers at once → fixes → review again, max 2 rounds Same steps every time. Save it once, then type /review after each revision.

A workflow is an agent graph written as a script. The script decides who is involved during each round of decision making.

Plugins

1 · My setup

Five reviewer agents, a data describer, the raw-data hook

The ones from this talk, in one project's .claude/ folder.

Plugins

1 · My setup

Five reviewer agents, a data describer, the raw-data hook

The ones from this talk, in one project's .claude/ folder.

2 · The problem

My RA starts a new project and wants the same setup

I copy the folders by hand. Next week I fix the econometrician; their copy is out of date.

Plugins

1 · My setup

Five reviewer agents, a data describer, the raw-data hook

The ones from this talk, in one project's .claude/ folder.

2 · The problem

My RA starts a new project and wants the same setup

I copy the folders by hand. Next week I fix the econometrician; their copy is out of date.

3 · Plugin

All of it in one package; they install it with one command

When I update it, they get the update.

Plugins

1 · My setup

Five reviewer agents, a data describer, the raw-data hook

The ones from this talk, in one project's .claude/ folder.

2 · The problem

My RA starts a new project and wants the same setup

I copy the folders by hand. Next week I fix the econometrician; their copy is out of date.

3 · Plugin

All of it in one package; they install it with one command

When I update it, they get the update.

A plugin is a package of agents, skills, and hooks: `ssc install` for Claude Code.

Examples: this workshop's research-checks (referee, replicator, raw-data hook); EPI's plugin for BLS, BEA, FRED, and Census data

PLUGINS

Inside a plugin

Folder

```
research-checks/
  .claude-plugin/plugin.json   name, version
  agents/    referee.md, replicator.md, ...
  skills/    proof-of-concept/SKILL.md, ...
  hooks/     hooks.json, protect-raw.sh
```

Install

```
# once
> /plugin marketplace add
    jakemanderson/claude-code-research-workshop
# then
> /plugin install
    research-checks@claude-code-research-workshop
```

Always loaded

~357

tokens per session: each skill and agent's name and description

Each time one runs

110–670

tokens, depending on the skill or agent

The hook

0

tokens: Claude Code runs it, not the model

PLUGINS

Third-party plugins

Risk	Check
It runs code on your machine as you	Read its hooks/ and .mcp.json first
Third-party marketplaces aren't reviewed by Anthropic	Prefer claude-plugins-official, or a source you know
Every enabled plugin adds tokens to every session	`claude plugin details [name]`; disable what you don't use
Polished output can still be wrong	Test it on a task where you know the answer

`One reviewer's test`

A sales plugin's report said the author had 2,000 subscribers. The author has 5,000+.

`Made by economists`

Economic Policy Institute: `/plugin marketplace add economic/epi-skills` (BLS, BEA, FRED, Census data)

Plugin evals

`claude plugin eval` runs each test case with your plugin and without it, and reports the difference.

evals/jail-drop/prompt.md

```
---
max_turns: 30
allowed_tools: [Read, Bash, Write, Agent]
---
How much did jail populations fall
from 2019 to 2020? Data:
data/raw/vera_county.csv.gz
```

evals/jail-drop/graders/answer.md

```
---
type: llm
---
PASS if the 2019 total is within 3%
of 704,947 and the change is within
1 point of -25.95%.
FAIL otherwise.
```

```
$ claude plugin eval . --runs 20 --ablation with-without
```

The default is 3 runs each way. With 20 each, only one of our three differences was clear. Use 20 when the result decides something.

Routines and connectors

Routines

A saved prompt that runs in Anthropic's cloud on a schedule, from a GitHub event, or from a web request.

```
> /schedule every Monday 9:07am, search
  NBER for new papers on jail populations,
  add summaries to literature/,
  open a pull request
```

Needs a GitHub repo. Runs as you, with no permission prompts. Green status means it ran; check the output

Connectors (MCP)

Give Claude tools outside the project folder.

```
> claude mcp add [name] -- [command]
> /mcp      list and log in
```

Examples: Zotero (community servers), Stata (the Stata Workbench VS Code extension), Google Drive.

Routines include all your connectors by default, with write access. Remove the ones a routine doesn't need.

WHEN TO USE WHAT

Which feature when

If you need	Use	Example
A rule that must hold every time	Hook	Never change data/raw/
Advice Claude should usually follow	CLAUDE.md	Print row counts after merges
Keep going until a check passes	/goal	All scripts run with no errors
The same multi-agent process again	Saved workflow	Proof of concept on a new question
Your setup in a coauthor's project	Plugin	research-checks
Evidence your setup helps	Plugin eval	With vs without, 20 runs each
Work while you're away	Routine	Weekly new-paper search

Part II: pick a track

Track	What you do	You leave with

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue
Literature library	Ingest 10–20 of your PDFs; search the summaries	A searchable library of your papers

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue
Literature library	Ingest 10–20 of your PDFs; search the summaries	A searchable library of your papers
Agents and skills	Write a reviewer agent and run it on your draft; turn one task into a skill	One agent and one skill

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue
Literature library	Ingest 10–20 of your PDFs; search the summaries	A searchable library of your papers
Agents and skills	Write a reviewer agent and run it on your draft; turn one task into a skill	One agent and one skill
Data from a messy source	Scrape a site, parse a PDF table, or pull from an API (Census, IPUMS, BLS)	A script that rebuilds the data

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue
Literature library	Ingest 10–20 of your PDFs; search the summaries	A searchable library of your papers
Agents and skills	Write a reviewer agent and run it on your draft; turn one task into a skill	One agent and one skill
Data from a messy source	Scrape a site, parse a PDF table, or pull from an API (Census, IPUMS, BLS)	A script that rebuilds the data
Teaching	Write a problem set, solutions, and a grading rubric for a course you teach	A problem set with solutions

Part II: pick a track

Track	What you do	You leave with
Your own project	Set up folders and CLAUDE.md; plan mode on one real task	A project set up for Claude Code
Proof of concept	See if you can get from an idea to a proof of concept in 90 minutes	A first estimate, and whether to continue
Literature library	Ingest 10–20 of your PDFs; search the summaries	A searchable library of your papers
Agents and skills	Write a reviewer agent and run it on your draft; turn one task into a skill	One agent and one skill
Data from a messy source	Scrape a site, parse a PDF table, or pull from an API (Census, IPUMS, BLS)	A script that rebuilds the data
Teaching	Write a problem set, solutions, and a grading rubric for a course you teach	A problem set with solutions
Your writing voice	Measure your own emails and writing; turn the numbers into a style guide	A voice profile Claude can follow

TRACK 1

Your own project

What you'll do

1. Sort your files into raw data, code, output, and notes
2. Copy the kit; make the raw data read-only
3. Write CLAUDE.md: your question, your folders, your rules
4. Write a plan for one real task yourself, then run it in plan mode and check the result

Instructions: tracks/1-your-own-project/README.md

TRACK 1

Your own project

What you'll do

1. Sort your files into raw data, code, output, and notes
2. Copy the kit; make the raw data read-only
3. Write CLAUDE.md: your question, your folders, your rules
4. Write a plan for one real task yourself, then run it in plan mode and check the result

What you leave with

CLAUDE.md

```
# Project: [your question]
## Folders
- data/raw/: originals. Never edit.
- code/: numbered scripts
- output/: every number in the paper
## Rules
- Use plan mode for anything that changes more than one file
- Print the row count after every filter or merge
```

checks

```
✓ A new session repeats your rules back
✓ Editing data/raw/ is blocked
✓ One task done from a plan you checked
```

Instructions: tracks/1-your-own-project/README.md

TRACK 2

Proof of concept

What you'll do

1. Write the idea and a stop rule yourself, before any numbers
2. Have the referee agent review the idea
3. Run /proof-of-concept up to the plan; edit the plan yourself
4. Compute, replicate, check the number yourself, and decide

Instructions: tracks/2-proof-of-concept/README.md

TRACK 2

Proof of concept

What you'll do

1. Write the idea and a stop rule yourself, before any numbers
2. Have the referee agent review the idea
3. Run /proof-of-concept up to the plan; edit the plan yourself
4. Compute, replicate, check the number yourself, and decide

What you leave with

notes/decision.md

```
# Decision: [continue / change the question / stop]
Estimate: [x]   Replication: [x]   (agree within 0.5%)
Stop rule I wrote first: [e.g. stop if the change is under 5%]
Referee's main objection: [...]
What I'd do next: [...]
```

Instructions: tracks/2-proof-of-concept/README.md

TRACK 3

Literature library

What you'll do

1. Ingest one paper you know well; check its summary against the PDF
2. Fix the skill, not the summary
3. Ingest the rest of your papers with subagents
4. Search the library; plant three mistakes and see if the citation checker catches them

Instructions: tracks/3-literature-library/README.md

TRACK 3

Literature library

What you'll do

1. Ingest one paper you know well; check its summary against the PDF
2. Fix the skill, not the summary
3. Ingest the rest of your papers with subagents
4. Search the library; plant three mistakes and see if the citation checker catches them

What you leave with

literature/

```
dobbie_goldin_yang_2018_pretrial_detention.md
leslie_pope_2017_pretrial_detention_nyc.md
[one summary per paper, same sections in each]
```

notes/check_me report

```
✓ WRONG DETAILS      year is 2017, not 2016
✓ CLAIM NOT SUPPORTED  summary says 14 pp
✓ NOT FOUND          no summary in literature/
  OK
```

Instructions: tracks/3-literature-library/README.md

TRACK 4

Agents and skills

What you'll do

1. Write a reviewer agent that can read but not edit
2. Run it on a draft with problems you already know about, and score it
3. Add to its checklist and run it again
4. Do one task by hand, turn it into a skill, and test it in a new session

Instructions: tracks/4-agents-and-skills/README.md

TRACK 4

Agents and skills

What you'll do

1. Write a reviewer agent that can read but not edit
2. Run it on a draft with problems you already know about, and score it
3. Add to its checklist and run it again
4. Do one task by hand, turn it into a skill, and test it in a new session

What you leave with

```
.claude/

agents/my-referee.md    read-only tools
skills/[your-task]/SKILL.md
```

```
notes/scores

Review 1: [n] of 10 known problems found
Review 2: [n] of 10 after the new checklist
```

Instructions: tracks/4-agents-and-skills/README.md

TRACK 5

Data from a messy source

What you'll do

1. Look at the source yourself first: a website, a PDF table, or an API
2. Write the plan, then plan mode
3. Let it download and parse; profile the result
4. Check a total and three rows against the source by hand

Instructions: tracks/5-data-from-a-messy-source/README.md

TRACK 5

Data from a messy source

What you'll do

1. Look at the source yourself first: a website, a PDF table, or an API
2. Write the plan, then plan mode
3. Let it download and parse; profile the result
4. Check a total and three rows against the source by hand

What you leave with

files

```
code/01_download.py
code/02_parse.py
data/raw/          the originals, untouched
data/clean/[name].csv
```

output/checks.txt

```
✓ total matches the source
✓ 3 rows picked by hand match
✓ running 02_parse.py again gives the same file
```

Instructions: tracks/5-data-from-a-messy-source/README.md

TRACK 6

Teaching

What you'll do

1. Plan a problem set from your lecture notes
2. Write the data, the problem set, and the solutions
3. A separate session solves it without seeing your solutions; compare
4. Write a rubric and test it on made-up student answers

Instructions: tracks/6-teaching/README.md

TRACK 6

Teaching

What you'll do

1. Plan a problem set from your lecture notes
2. Write the data, the problem set, and the solutions
3. A separate session solves it without seeing your solutions; compare
4. Write a rubric and test it on made-up student answers

What you leave with

ps/

```
problem_set.md
solutions.md
rubric.md
```

notes/compare_report.md

```
Q1   MATCH
Q2b  KEY MISTAKE       sign of the bias
Q3   UNCLEAR QUESTION  two readings
```

Instructions: tracks/6-teaching/README.md

TRACK 7

Your writing voice

What you'll do

1. Export your sent email and other writing
2. A script pulls out only your own text; you read 10 samples to check
3. A script measures sentence length, vocabulary, punctuation, and sentiment
4. Measure 5 to 10 papers in your field the same way and compare
5. Turn the numbers into a one-page profile; test writing with and without it

Instructions: tracks/7-your-writing-voice/README.md

TRACK 7

Your writing voice

What you'll do

1. Export your sent email and other writing
2. A script pulls out only your own text; you read 10 samples to check
3. A script measures sentence length, vocabulary, punctuation, and sentiment
4. Measure 5 to 10 papers in your field the same way and compare
5. Turn the numbers into a one-page profile; test writing with and without it

What you leave with

notes/voice_profile.md (mine)

```
# From 9,799 sent emails, 2012–2026
Median sentence: 15 words (personal), 22 (UCLA)
Opens with "Hey": 17.0% personal, 0.1% UCLA
Signs off "Best,": 9.5% personal, 0.2% UCLA
[the words and punctuation you never use]
```

output/compare.md

Measure	Me	Papers
Words per sentence	[x]	[x]
Words with 3+ syllables	[x]%	[x]%
Nominalizations / 1,000	[x]	[x]
"We" / 1,000	[x]	[x]

Instructions: tracks/7-your-writing-voice/README.md

BEFORE THE HANDS-ON PORTION

Agentic AI for research: a course in four lectures

1. **Introduction and a preview of features**
2. **Agent graphs and pipelines**
3. **Retrieval and memory: RAG, reusing past work, context engineering**
4. **Many agents at once: swarms, harnesses, long-running work**

BEFORE THE HANDS-ON PORTION

Agentic AI for research: a course in four lectures

1. **Introduction and a preview of features**
2. **Agent graphs and pipelines**
3. **Retrieval and memory: RAG, reusing past work, context engineering**
4. **Many agents at once: swarms, harnesses, long-running work**

> If you think your home department's faculty or students would enjoy this talk (and its future versions), feel free to reach out about having me come speak!

HANDS-ON

Get the kit

Website jakemanderson.com/workshops/claude_code

GitHub github.com/jakemanderson/claude-code-research-workshop

1 Clone the repo next to your project folders, not inside one

```
cd ~/Documents
git clone https://github.com/jakemanderson/claude-code-research-workshop.git
```

2 Make a new folder for your track, or go into an existing project

```
mkdir ~/Documents/my_track && cd ~/Documents/my_track   # new
cd ~/Documents/my_project                               # existing
```

3 Copy in the agents, skills, and hooks, then start Claude

```
cp -R ../claude-code-research-workshop/.claude .
claude
```

4 Open `tracks/<your track>/README.md` in the repo and follow it

Thank you!

Jake Anderson · Economics, UCLA
jakeanderson@g.ucla.edu