Agentic AI and Empirical Research

What I delegate, what I check, and what I decide

Pedro H. C. Sant'Anna

Emory University

November 12, 2026

The workflow I use in my own research

claude-code-my-workflow is my public template for using Claude Code on research and teaching projects. You can copy it from GitHub and adapt it to your work.

- **Skills** give Claude procedures for specific tasks, such as reviewing a paper.
- **Shared rules** carry practices across tasks, such as reproducing a result before extending it.
- **Automatic checks** can flag results for review when analysis code changes.

> This is a powerful workflow for getting more done. Even with these checks, errors can slip through. I still verify the results I use.

Skills, shared rules, and automatic checks

A repository of reusable help for research and teaching.

- **61 skills** give instructions for particular tasks, such as reviewing a paper, checking R code, or preparing a lecture.
- **37 rules** carry instructions across tasks, such as planning substantial work, using consistent notation, and replicating results before extending them.
- **11 hooks** run automatically at specific moments. A new session can load saved notes. A code change can flag linked results for rechecking.
- **18 specialized agents** take on focused jobs, such as reviewing R code, checking a lesson's clarity, or fixing slides.

Workflow repository

How I use it in my research

Example request for `overdid`, the R package for our Efficient DiD paper (Chen, Sant'Anna, and Xie 2025)

In `overdid`, is estimating an event study directly better than estimating each group's effect and combining them?

Use the paper and appendix, both R implementations, and archived simulations. Review arguments and code; compare the same target, groups, and periods.

Check small samples, long panels, and clustering. Report bias, RMSE, 95% interval coverage, runtime, memory, and Monte Carlo uncertainty. Keep the criteria registered before the runs; flag coverage below 90%.

Deliver R scripts, an R-exported table, and a two-page recommendation. Require a clean rerun and evidence for every claim. Explain theory/code gaps; verify Oracle findings independently.

Skills Claude can select: paper review, R code review, and reproducibility audit.

Audit finding: faster computation came with statistical trade-offs.

Why use the same simulated data? Both methods face the same problem, and the known effect gives us a benchmark for accuracy and interval coverage.

How I use the workflow for teaching

- **Minimum wages.** New Jersey versus Pennsylvania in Card & Krueger (1994).
- **Bank deregulation.** States reform at different dates (Jayaratne & Strahan, 1996).
- **Democracy and growth.** Countries democratize and sometimes revert (Acemoglu et al., 2019).

An example of the request I would give

Create a 15-minute lesson on Card & Krueger (1994) for students who know supply and demand, not DiD. Use Table 3, rows 1–3, and my Lecture 1 HTML style.

Four slides: policy; NJ/PA table; DiD and parallel trends; discussion question (answer in notes).

Use R outputs for every number in slides and notes. Import the table exported directly by R; never retype values. Check Table 3 within 0.02 full-time-equivalent jobs and explain rounding.

Follow AEA data/code guidance. Deliver QMD, HTML, master R script, check report, and README with data sources, software versions, and run instructions. Cite Table 3; check overflow.

Why have R generate every number? The script is the source for the table. One rerun updates it, and the published table gives us a check.

How can you use the Repo?

Getting started

1. Fork and clone the repository and follow the guide's setup steps.
2. Open Claude Code in your project folder.
3. Fill in your project details and review the proposed setup. Copy the starter prompt

> For each task, explain the question and audience, name the sources to check against, define what counts as success, and specify the files you want back.
>
> "Every number in the slides must come from the R script; one command must regenerate the table and slides."

Prompt shaping is a shared rule that helps clarify informal requests. `/interview-me` is a skill that asks focused questions to turn an idea into a written plan.

How I cross-check the work

- From Claude or Codex, I use Oracle to send a focused question and relevant files to another model.
- For difficult questions, I've found **GPT-6 Pro** useful for reviewing proofs, methods, and implementations.
- I ask for concrete errors, counterexamples, or failing cases, then check each finding against the argument, source, or calculation.

> "Find a counterexample to this lemma, or identify the exact assumptions needed for the argument to hold."

Two models can share a mistake. I'm responsible for the conclusions I use. Oracle on GitHub

From the template to a project

What I delegate, what I verify, and what I record.

What the agent does inside a project

My request. “Reproduce one number in this paper from the authors’ code.”

1. The agent reads the paper and the instructions for running the code.
2. It runs the program and reads what the program reports.
3. It can fix a path and try again, or stop and tell me what is missing.
4. It returns the number, the command, and where I can check the output.

I keep a record on GitHub. Commits show what changed. Issues record bugs, decisions, and follow-up checks so I can find them in a later session.

What I put in the project folder

In my public template	What it does
`CLAUDE.md`	Instructions the agent reads at the start. One rule is **“Plan first.”**
Replication protocol	Reproduce the authors’ result before extending it. Keep a record of discrepancies.
Data analysis skill	A reusable recipe for writing and running an analysis and saving its outputs.
Reproducibility audit	Trace a number in the manuscript to the output that produced it.

A real job starts with one result

> Reproduce Table 2, column 1 of the authors' paper. Use the sample and coding rules in their replication files. Return the estimate, standard error, sample count, and command. If a choice would change the sample or policy coding, bring it to me first.

Ground truth for this check. The published values tell us what the replay should produce. I compare the estimate, standard error, and sample count with the saved output. That verification is what makes the replication reliable.

Where my judgment enters

I decide	The agent can do
The question and the comparison that would answer it	Locate files and run the agreed analysis
Which checks matter	Show sample counts, treatment dates, and discrepancies
What the evidence lets me claim	Rebuild the table or try a comparison I approve

> If I have to sign my name to the result, I need to be able to explain how it was made.

Case 1 · Juvenile arrests

What happens when we reproduce a published result and then change the method?

Why I started with a published paper

Marcelo Ortiz-Villavicencio and I were revising a paper about **how to estimate the effect of a policy when places adopt it at different times**.

We wanted to see whether the choice of method mattered in real studies. We started with Ashna Arora's study of juvenile justice (2023), which has a published table and code we could rerun.

> First, reproduce the author's result. Then ask what changes when we apply our method on a common sample.

Arora's question about juvenile arrests

Several states raised the age for adult criminal court from 17 to 18. **Why might arrests rise even for 16-year-olds, who stay in juvenile court?**

- **Future incentives.** An extra year of lighter punishment can make gaining experience in crime today more worthwhile.
- **Peer spillovers.** If lighter sanctions lead more 17-year-olds to offend, younger peers may find it easier or more rewarding to join them.
- **The causal challenge.** Without reform, the younger–older arrest gap must have followed similar trends across states. Arrests also depend on policing and reporting.

Arora, Ashna. 2023. "Juvenile Crime and Anticipated Punishment." *American Economic Journal: Economic Policy* 15 (4), 522–550.

What the arrest data look like

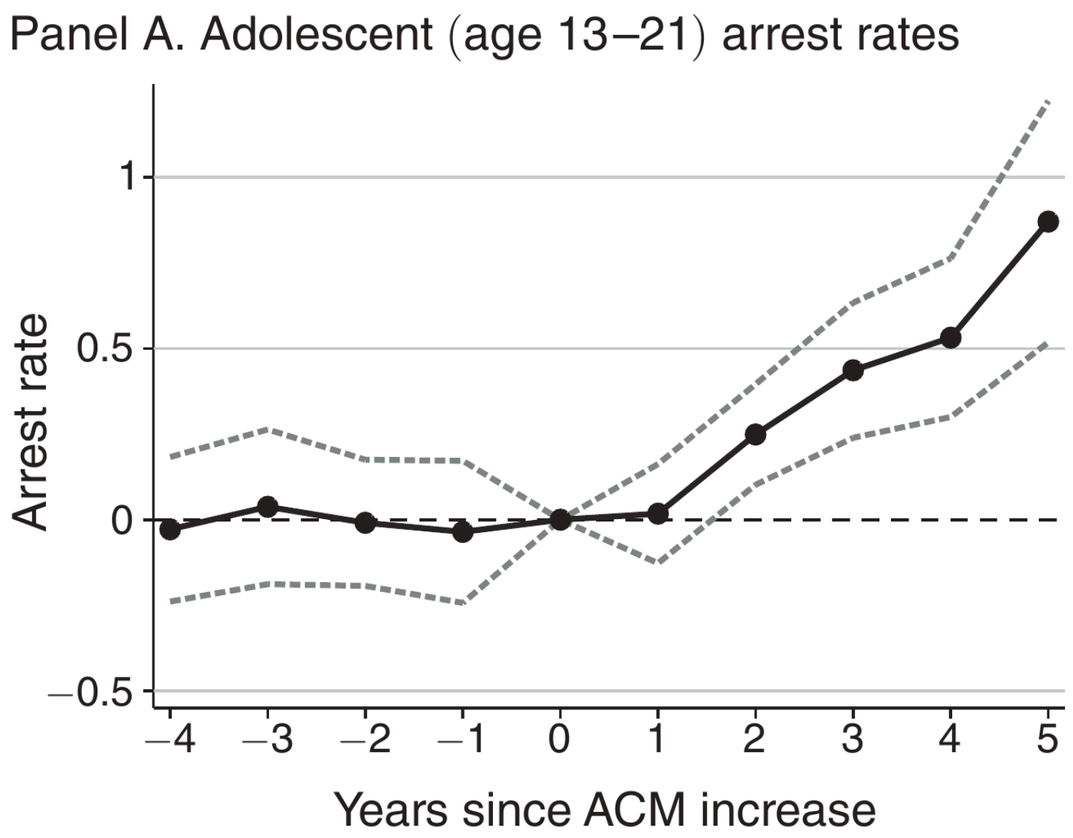
One row records an agency, an age band, and a month. The outcome is a monthly arrest rate per 100,000 residents.

State and agency ID	Age	Month	Arrest rate	Study exposure
Connecticut, agency 7	17	June 2012	0.000	No
Connecticut, agency 7	17	July 2012	2.054	Yes
Connecticut, agency 7	22	June 2012	2.739	No
Connecticut, agency 7	22	July 2012	2.739	No

Four actual rows for Bridgeport, Connecticut (agency 7), from the analysis file built by Arora's replication code. Rates are rounded to three decimals. Replication package

The policy calendar dates the reform. Here the arrest rate rises by **2.054 per 100,000** at age 17 and stays flat at age 22, so the younger–older gap rises by **2.054**. Subtract the corresponding change in a comparison state to get the triple difference.

What the original paper showed



About 12% higher

The original average estimate was **+0.222** arrests per 100,000 residents in a month, against a comparison mean of **1.900**.

ACM is the age of criminal majority. Read across for years relative to reform and up for changes in arrests. Dashed lines show uncertainty.

Arora (2023), Table 2, column 1 and Figure 5, panel A

The first handoff matched the paper

Table 2, column 1	Published	Reproduced
Estimated increase	0.222	0.222
Standard error	0.067	0.067
Agency × age-band × month rows	1,332,432	1,332,432

The agent helped trace the author’s input and code and compare the output with the printed table.

> We now had a benchmark to come back to when the new analysis moved.

The new run had to decide what to do with Rhode Island

Rhode Island **lowered** its age threshold in 2007 and **restored** it four months later. An early version of our code gave it the same label as states that never changed their law.

> Our method expects one switch that stays in place. I flagged Rhode Island because its law changed twice.

A label of “never changed” can let those four policy months enter the comparison used to estimate what would have happened without the policy.

The policy calendar settled the sample decision

State	What the policy record showed
Connecticut, Massachusetts, New Hampshire	Raised the threshold in July 2012, September 2013, and July 2015, respectively
New York, Vermont	No switch in this window
Rhode Island	Temporary change in 2007, then a reversal

My decision. Keep Arora's six-state benchmark. For the new comparison, drop Rhode Island **for both methods** before assigning treatment groups.

I asked what else the agent had changed

For each reform state, we compare how the arrest gap between younger and older people changes with how that gap changes in states that have yet to reform. We give the comparisons weights and use the final pre-law year as the reference.

Procedure	What the code did
Our established method	Combine the comparisons, then subtract the average estimate from the pre-law year.
Agent's proposed main result	Set the pre-law reference inside each comparison, then choose weights and combine.

What triggered the check. The agent also volunteered a result using all the years before the law. I asked what else had changed in the replication code.

We held the sample fixed and ran both recipes

Five-state triple-difference average	Estimate
Agent’s draft procedure	0.232
Our restored procedure	0.243

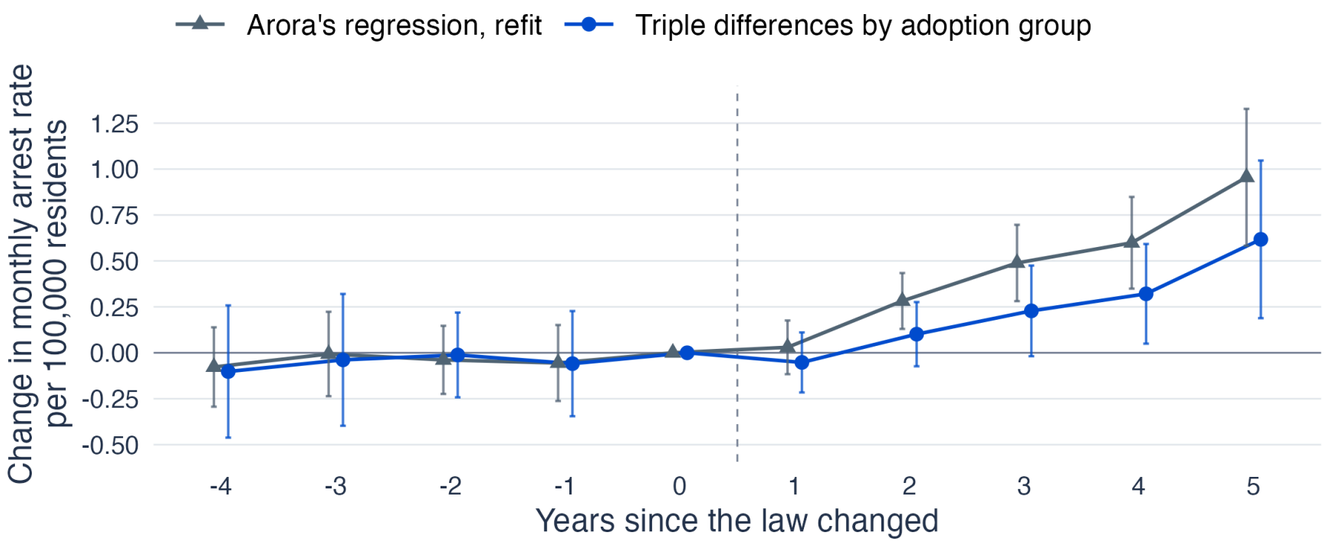
Same data and same year weights for the annual average. Units are **monthly arrests per 100,000 residents**.

Each year keeps its weight in the final average. Changing when we set the baseline can change the weights on comparison states within each year.

At year 4: the draft’s 95% interval was **[−0.001, 0.537]**. The restored interval was **[0.050, 0.592]**. The baseline choice changed whether the interval included zero.

Arora’s Table 2 reports **+0.222** from one regression on six states. Here we average five annual DDD estimates from the same five states and change only how we set the baseline.

The revised plot gave us a new question



Blue shows our DDD estimates. We use the estimator from our *Triple Differences* paper. Gray refits Arora's regression on the same five states.

The last point looks like the others. Its label tells us the year, while the underlying data tell us which states and months contribute.

Five-state reanalysis. Year 1 is the first 12 months after the law changes.

The last point was six months from Connecticut

How we summarized the five-state triple differences	Result
Equal weight on the first four years	0.150
Equal weight on all five available years	0.243

The last year supplied **3.85% of the treated observations** and **20% of the equal-year average**.

Equal weight answers a question about the average across **years**. Only Connecticut supplies treated observations in the last year. Comparison states can still contribute to its estimate.

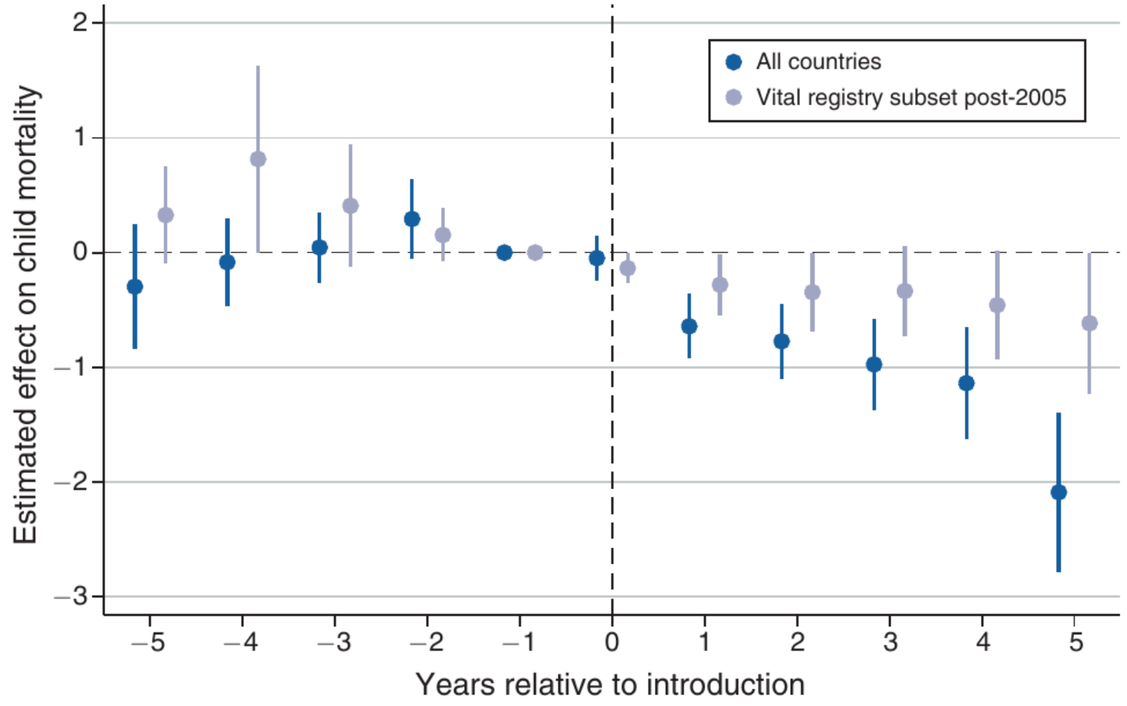
Case 2 · Vaccine aid

Did introducing Gavi-funded vaccines reduce deaths among children under five from the diseases those vaccines target?

The answer helps funders judge whether expanding vaccine access saves lives.

Why the authors show two samples

Shastry and Tortorice check whether the result holds across data sources and samples.



Dark blue tests whether the result extends to a broader sample, including countries with modeled causes of death.

Gray uses official death records after 2005 to check the result using causes recorded directly.

Our replication matched Table 6 at **−1.19 deaths per 1,000 live births**.

Shastry and Tortorice (2025), Figure 5 and Table 6, panel A, column 2

What counts as one observation in Gavi?

One observation is a country, a year, and a cause of death. Each number below is deaths per 1,000 live births.

Country	Cause	2011	2012	2013	Diarrhea vaccine launch
Armenia	Diarrhea	0.447	0.402	0.364	2012
Armenia	Injuries	1.383	1.326	1.276	Untargeted
Kazakhstan	Diarrhea	0.465	0.385	0.313	No Gavi launch
Kazakhstan	Injuries	1.215	1.178	1.123	Untargeted

Compare diarrhea with injuries in each country. Diarrhea deaths also fell in Kazakhstan, where the data show no Gavi launch.

Actual observations from Shastry and Tortorice's prepared analysis data. Replication package

How we define the treatment date matters

For Solomon Islands, the source records put vaccine introduction in **2008**. The study's rule makes **2009** the first treated year. The early code used **2012**, when usable mortality records first appeared.

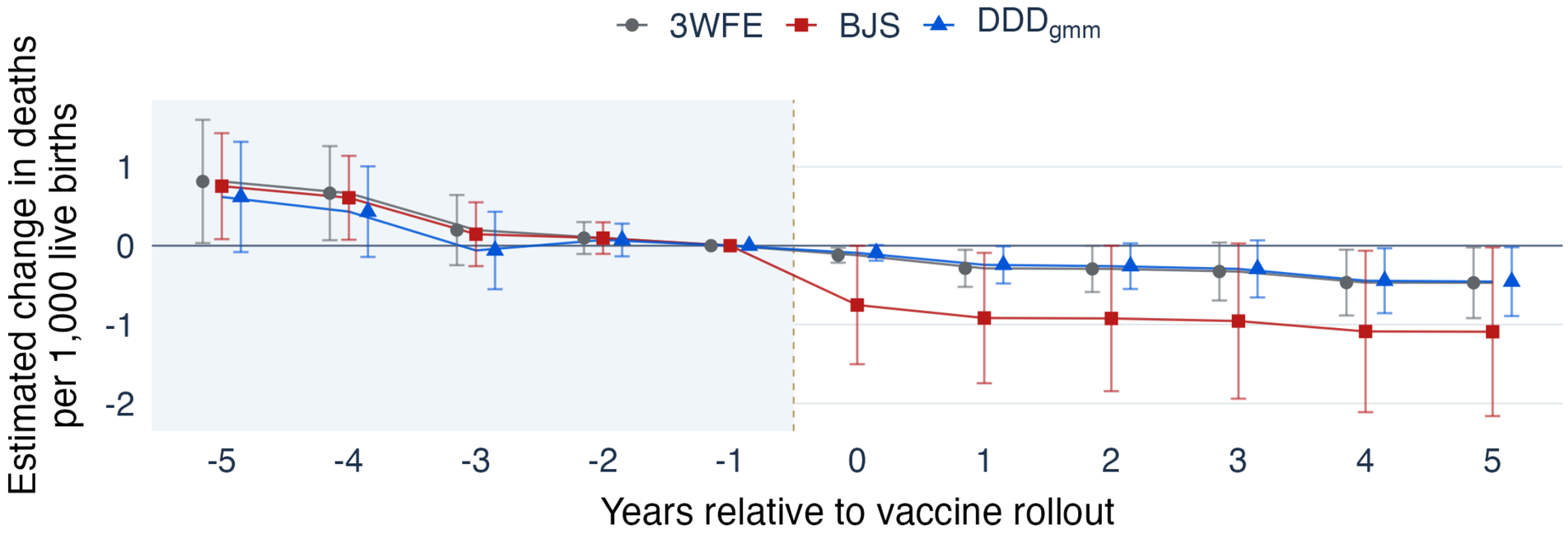
Solomon Islands	2008	2009	2011	2012
After introduction under the study's rule	No	Yes	Yes	Yes
Mortality rows we kept for the analysis	0	0	0	4

The study dates exposure from the vaccine rollout. The first usable mortality record tells us when we can measure the outcome.

A 2009 treatment date requires a 2008 baseline. With no usable 2008 mortality records, Solomon Islands drops out. The rerun corrected both dates and which countries had data to support the comparison.

After correcting dates and support, the triple difference moved from **−0.079 to −0.141 deaths per 1,000 live births**, close to the corrected regression estimate of **−0.142**.

The figure ran. The sentence did not fit it.



Our reanalysis, Figure 6. The two benchmarks are three-way fixed effects (3WFE) and Borusyak, Jaravel, and Spiess imputation (BJS).

The AI draft said "We estimate both benchmarks on the four targeted causes." I checked the inputs to the figure and table separately.

The benchmarks use **13 causes in this plot** and **4 in the summary table**.

Differences between the figure and table can reflect which causes were included. The draft sentence obscured that difference.

Testing our own methods

With simulated data, we know the true effect and can see which runs fail.

A handoff said every simulation returned a result

We simulated data with a known policy effect to check our uncertainty intervals in *Efficient Difference-in-Differences and Event Study Estimators* (Chen, Sant'Anna, and Xie 2025).

Planned draws	Fits returned	Intervals available	Intervals that covered
1,000	997	994	991

Claude's handoff said every run returned a result. Codex hit a missing value, and the run records showed six missing intervals.

- **991 / 994** is coverage among the intervals we could check.
- **991 / 1,000** is the share of planned runs that delivered an interval covering the true effect.

The six runs without usable intervals still count when we evaluate how often the whole procedure delivers a covering interval.

A faster method still needed a reliability check

I gave Claude two R implementations from **overdid**, our Efficient DiD package. One estimates the average policy effect directly. The other estimates it for each group and averages across groups. I asked Claude to run both on the same simulated data, where we set the true effect in advance.

We asked	The audit showed
How fast does it run?	Direct fitting took about one-third as long on the main simulation grid.
How often does an interval cover the known answer?	In one design, intervals intended to cover 95% covered only 85%.

You can use the same test whenever you speed up an analysis. Check the accuracy of the estimates and how often their intervals cover the known answer.

Checks I carry into the next project

What should we decide before the run? What evidence should we save afterward?

What I now ask for before any analysis

The examples show what can change after an agent starts. I now write down the key decisions before the first run.

In the research brief	How it would have helped in Arora
What is the question and target?	Reproduce one published result, then compare methods on the same five states.
Which choices do we fix?	State list, policy dates, age groups, reference period, and averaging rule.
What requires my decision?	Changes to the sample, policy coding, baseline, or interpretation.
What would convince us the task is done?	Show the benchmark, treatment calendar, and command for the revised result.

An error can become a check for the next project

I also work at Amazon, and I borrow some of its practices for checking research.

What Amazon Web Services (AWS) does. When an incident exposes a failure, teams turn it into a question that future teams can check before release.

Our version. Print every state's policy changes before fitting the model. Flag any state with more than one switch.

Give the check	It should return
A state with one adoption that stays in place	Pass
Rhode Island's change and reversal	Flag, with the two dates

A bad data join. Two workers earn $100 and $400. Their mean is $250. A join duplicates the $100 record and the mean falls to $200. Require one earnings record per worker for the quarter. The clean data should pass and the duplicate should fail. Log the duplicate key, the fix, and the test in GitHub.

Five different questions about the same result

I want to know	A check that helps
Can I regenerate it?	Replay the published table from the inputs.
Did we do what we agreed?	Trace the baseline and weights in the code.
Do the data measure what I say?	Match the treatment dates to the legal record.
Does this method behave as intended?	Test it on simulated data with a known answer.
Can I read this as a causal effect?	Explain why the comparison describes what would happen without the policy.

The Arora replay answers the first question. The policy calendar checks what treatment means. The causal question still needs an argument about what would have happened without the law.

What goes beside a number I report

Once I've checked a result, I save the evidence with the claim so someone can retrace it later.

Claim. The last point in our five-state arrest plot uses six months from Connecticut.

Evidence. The support file shows one state and 3,216 treated observations at that horizon.

Scope. The point describes the observations available at that horizon. That last point supplies one-fifth of the equal-year average.

> Save the data version, command, output, and reason for any change beside the claim.

How I cut slop before it reaches the paper

A draft sentence	A sentence I can check
“The results are robust.”	“The published estimate, standard error, and sample size reproduced to the printed precision.”
“Both methods use four causes.”	“The benchmarks use four causes in Table 5 and all 13 in Figure 6.”

Give the agent the actual exhibit. Ask for the exact source of each claim and require it to preserve the numbers and units. Cut sentences that repeat the result or make it sound stronger.

A second model as a reviewer

Ask it to find a specific failure, then test the finding.

How I cross-check with two providers

1. **Claude (Anthropic)** runs the agreed analysis and saves the input, code, and output.
2. **Codex or ChatGPT (OpenAI)** gets the original sources and the same question in a fresh session. I ask for its own calculation before sharing Claude's answer.
3. **I compare the two trails.** When they disagree, I trace the files or run a check that can settle the question.

Oracle packages selected files and a question for a separate model session. Oracle on GitHub

A second model can inherit the same mistake

What happened in overdid	What it meant
We told Oracle the default fitted curve changed how it could bend at the middle of the data.	The prompt gave the reviewer a factual premise about the code.
Oracle found a failure that depended on that feature.	The objection made sense if the premise held.
We checked the code. The default had no such breakpoint.	The failure applied to a different setting.

> Ask the reviewer to verify your premises. Settle disagreements with the original evidence.

Two prompts that ask a reviewer to find a failure

Audit the analysis. “Try to find one state and month where the treatment code disagrees with the original study. Show the rule, the expected and observed values, and the smallest check that reproduces the discrepancy.”

Audit the claim. “Point to the table cell or source passage that supports each sentence. Try to falsify the strongest claim. If you cannot settle it with the evidence provided, say what is missing.”

For the first prompt, supply the policy rule, calendar, and code. For the second, supply the draft paragraph, its tables or figures, and their notes.

The expertise I still have to bring

Before the run. Know enough about the study to name the comparison and the decisions that can change it.

During the run. Recognize when a result could be right for a different sample or method.

After the run. Decide what the result supports and be able to defend it to someone who asks.

In Arora, knowing that a law could change twice told me to check the policy calendar before relying on the treatment code.

How I’d start the next replication

- **Delegate.** Rebuild the original table and return the code and outputs.
- **Check.** Match the published numbers and verify the treatment dates against the policy calendar.
- **Decide.** Approve the comparison sample and the claim it can support.

Put those three instructions in the research brief before the first run.

What I want you to take away

- **Treat AI-generated analysis and code as unverified.** Reproduce a known result or check the output against an independent calculation. Code can run and still answer the wrong question.
- **Build the skills to spot mistakes.** Learn the method, read the code, inspect the data, and work through small examples by hand.
- **Give the agent a harness.** Specify the task, the benchmark, the checks, and the evidence you expect back.
- **Keep refining the workflow.** Turn a mistake you catch into a check the next run can use.
- **Log decisions in GitHub.** Use issues for what went wrong, why you made a choice, and what to check next. Use commits to track what changed.

Sources and further reading

Workflow and safeguards

- My public Claude workflow and Claude Mini
- AWS Operational Readiness Reviews

Research

- Ortiz-Villavicencio and Sant'Anna, Better Understanding Triple Differences Estimators
- Arora (2023), Juvenile Crime and Anticipated Punishment
- Shastry and Tortorice (2025), Effective Health Aid
- Chen, Sant'Anna, and Xie (2025), Efficient Difference-in-Differences and Event Study Estimators