# Six Packages, One Estimator

Do They Agree?

A Cross-Package Specification Curve Audit

of Callaway & Sant'Anna (2021) Implementations

Scott Cunningham
Baylor University

March 2026

## Claude Code wrote every script in three languages simultaneously

A cross-language audit of 6 packages requires fluency in R, Stata, **and** Python — and identical specifications in each.

That used to mean assembling a team of polyglot programmers.

Claude Code wrote all 96 scripts, built every figure, diagnosed the numerical failures, and compiled this deck.

**Three goals today:**

1. Show what AI-assisted empirical research looks like
2. Expose hidden implementation differences in CS (2021)
3. Make concrete recommendations for applied researchers

## We trust econometric packages implicitly

A researcher types:

```
att_gt(yname, tname, idname, gname,
       xformla = ~X, est_method = "dr")
```

and trusts the output.

But should they?

## This experiment removes all analyst discretion

Same data. Same estimator. Same covariates.

Same control group. Same base period.

Same event-study window. Same seed.

The only thing that varies is the package.

Unlike many-analysts designs, there is **zero** researcher discretion.
Any variation in ATT estimates is purely due to implementation differences.

## Six packages, three languages, different internal machinery

| Package | Lang. | Authors | Stars | PS Model | OR Model |
|---|---|---|---|---|---|
| `did` | R | Callaway & Sant'Anna | 383 | logit (MLE) | OLS |
| `ddml` | R | Wiemann | 21 | ML (cross-fit) | ML (cross-fit) |
| `csdid` | Stata | Rios-Avila et al. | — | logit (IRLS) | WLS |
| `csdid2` | Stata | Rios-Avila | 11 | logit (Mata) | WLS |
| `differences` | Python | Dionisi | 108 | logit (MLE) | OLS |
| `diff-diff` | Python | Gerber | 110 | logit (BFGS) | OLS |

All implement CS (2021) doubly robust estimation. Same estimator in theory.
Different numerical implementations in practice.

## CS (2021) has 10,750 citations in six years

**10,750**

Google Scholar citations since 2019

Six independent teams built software to implement it.

At this adoption scale, implementation differences
affect thousands of applied papers.

## The Experiment

Design, data, and a quick review of CS (2021)

## CAPS mental health centers and Brazilian homicide

### Setting

Staggered rollout of CAPS (mental health centers) across Brazilian municipalities, 2002–2016.

Outcome: homicide rate (sim_agressao)

14 treatment cohorts (2003–2016)

### Estimation Framework

Callaway & Sant'Anna (2021)

Doubly robust (DR)

Not-yet-treated controls

Universal base period

Balanced event study: $e \in [-4, +4]$

Bootstrap: 1000 iterations

Common seed: 1

## The DR estimator combines two models per 2×2 cell

### Two models per cell

**Propensity score:**
$\hat{p}(X) = \Pr(G{=}g \mid X)$
Who gets treated when?

**Outcome regression:**
$\hat{m}(X) = E[Y \mid X, G{\neq}g]$
Counterfactual from untreated

DR is consistent if **either** model is correctly specified.

### The DR formula

$\widehat{\mathsf{ATT}}(g,t) =$

$$E\left[\left(\frac{G}{E[G]} - \frac{\hat{p}(X)(1{-}G)}{(1{-}\hat{p}(X))E[G]}\right)\Delta Y\right]$$

where $\Delta Y = Y_t - Y_0 - \hat{m}_{t,0}(X)$

Both models require fitting $\mathbf{X}'\mathbf{X}$ inversions internally.

## `dripw` and `drimp` are different estimators

### `dripw` (traditional DR)

Default in R's `did`

Logit via MLE for $\hat{p}(X)$
OLS for $\hat{m}(X)$

Sant'Anna & Zhao (2020)
Theorem 1

### `drimp` (improved DR)

Default in Stata's `csdid`/`csdid2`

Inverse probability tilting for $\hat{p}(X)$
WLS for $\hat{m}(X)$

Sant'Anna & Zhao (2020)
Theorem 3

Both are doubly robust. Both are semiparametrically efficient.

But they use different optimizers and weighting schemes
— and handle near-separation differently.

## 29 covariates produce 536 million possible specifications

The Dias & Fontes (2024) dataset includes **29 covariates**:

Demographics, economics, health system capacity,
urbanization, education, and population trends.

$$\sum_{k=1}^{29} \binom{29}{k} = 2^{29} - 1 = \mathbf{536{,}870{,}911}$$

non-empty covariate subsets

We select **16 specifications** that sample this space.

But the dataset has 5,172 municipalities — and individual cohorts
have as few as 47 treated units. More covariates, fewer comparable units.

## High-dimensional covariates break common support in finite samples

### The problem

Each 2×2 cell compares one cohort (e.g., 47 treated) to ∼4,000 controls.

As $k$ grows, some covariate combinations **perfectly predict** treatment status.

$\hat{p}(X) \to 0$ or $1$ for those units.

### The consequence

The IPW weight is $\hat{p}/(1-\hat{p})$.

When $\hat{p} \to 1$, this weight $\to \infty$.

**Common support fails:**
no comparable untreated unit exists.

Each package handles this failure differently — which is what produces different ATTs.

## OLS extrapolates beyond the data; propensity scores cannot

### TWFE regression

$Y = \alpha + \delta D + \beta X + \varepsilon$

OLS fits a line and **extrapolates** beyond the support of the data.

If no control looks like a treated unit, OLS imputes the counterfactual anyway.

Wrong if the relationship is nonlinear, but it always produces a number.

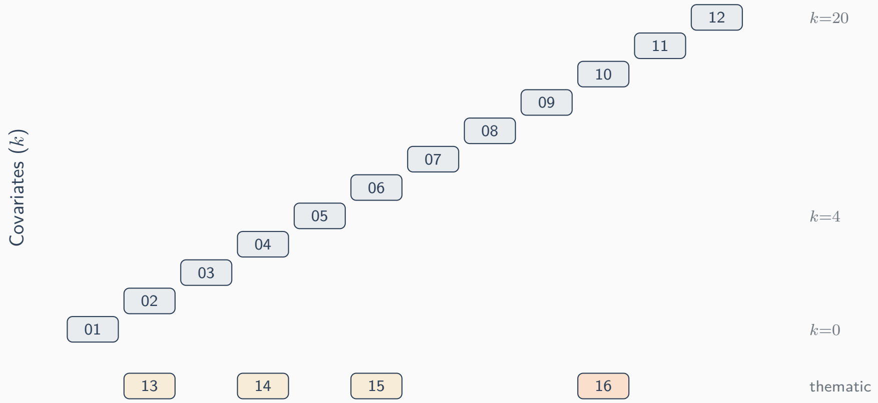
### CS (2021) doubly robust

The propensity score **reweights** actual observations: $\hat{p}/(1-\hat{p})$.

If no control looks like a treated unit, there is nothing to reweight.

The method needs overlap in $X$-space.
No overlap $\rightarrow$ no estimate (or garbage).

## Sixteen specifications build a covariate ladder



Specs 01–12: cumulative ladder. Specs 13–15: thematic subsets (no `poptotaltrend`). Spec 16: kitchen sink ($k$=29).

## With no covariates, all six packages agree

### Spec 01: $k = 0$ (unconditional)

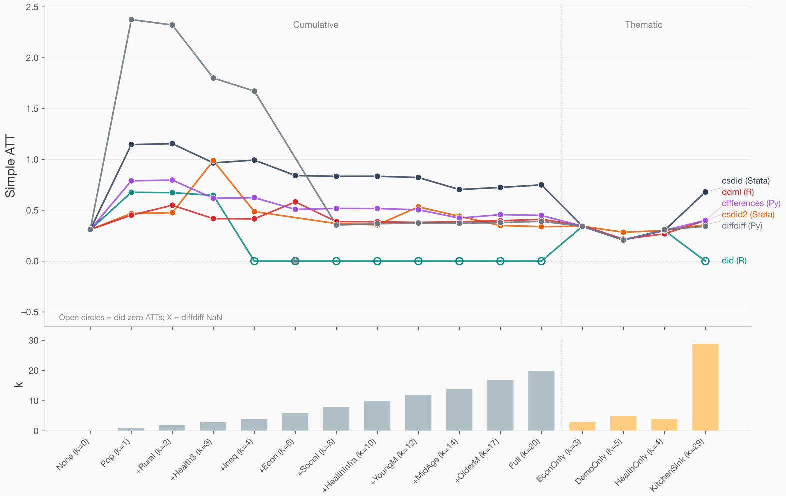
| Package | Simple ATT |
|---|---|
| `did` (R) | 0.3105 |
| `ddml` (R) | 0.3105 |
| `csdid` (Stata) | 0.3107 |
| `csdid2` (Stata) | 0.3105 |
| `differences` (Python) | 0.3105 |
| `diff-diff` (Python) | 0.3122 |

Range: 0.0017. All packages agree to within rounding.
The baseline works. The data are the same. The estimator is the same.

## The Divergence

What happens when you add covariates

## Adding covariates causes packages to diverge



## R's `did` package silently returns zeros

**What the user sees**

For specs 05–12, 16 ($k \geq 4$ with `poptotaltrend`):

```
ATT = 0.00000000
SE = NA
```

All 14 group ATTs: zero. All event-study estimates: zero.

No warning. No error message.

**What happened inside**

`att_gt()` checks: `rcond(X'X) < .Machine$double.eps`

When this fails, `did` aborts the ATT$(g,t)$ cell.

With `poptotaltrend` ($\sim 10^{10}$), $\kappa > 1/\varepsilon$. All 210 cells abort $\rightarrow$ all zeros.

## A sixth package produced unusable estimates

We tested a sixth package (`diff-diff`, Python).

Its logistic regression silently failed on most specifications — converging to wrong coefficients or falling back to unconditional estimation without warning.

We are working with the developer on the issues we identified.

The remaining analysis focuses on the five packages above.

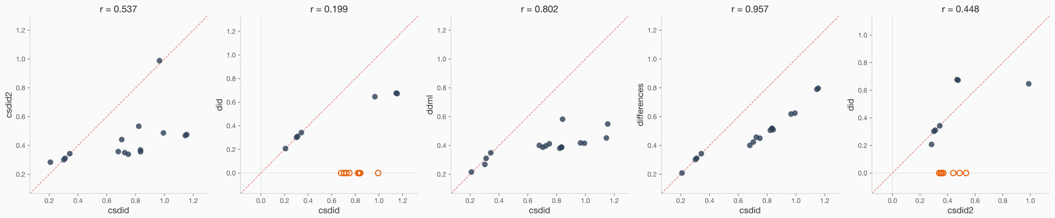
## `csdid` consistently estimates ATTs twice as large

Selected specifications where all packages produce non-zero estimates

| | Simple ATT | | | | |
|---|---|---|---|---|---|
| **Spec** | `did` | `ddml` | `differences` | `csdid` | `csdid2` |
| 02 ($k$=1) | 0.68 | 0.45 | 0.79 | **1.15** | 0.47 |
| 03 ($k$=2) | 0.67 | 0.55 | 0.80 | **1.15** | 0.48 |
| 04 ($k$=3) | 0.65 | 0.42 | 0.62 | **0.97** | 0.99 |
| 13 ($k$=3) | 0.34 | 0.35 | 0.34 | 0.34 | 0.34 |
| 14 ($k$=5) | 0.21 | 0.22 | 0.21 | 0.21 | 0.28 |

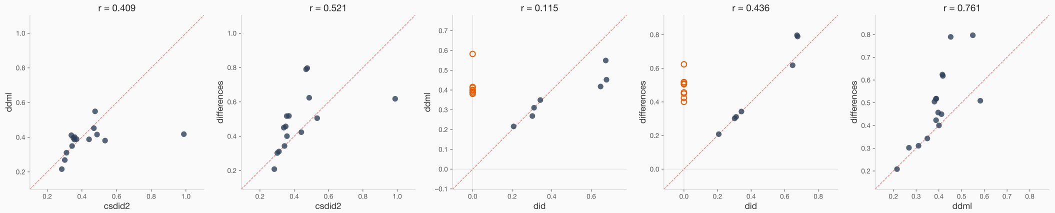
Specs 13–14 (no `poptotaltrend`): packages converge (range $< 0.07$).
Specs 02–04 (with `poptotaltrend`): `csdid` produces the largest ATTs — up to 2.5× `ddml`.

# Pairwise ATT agreement: csdid family and did



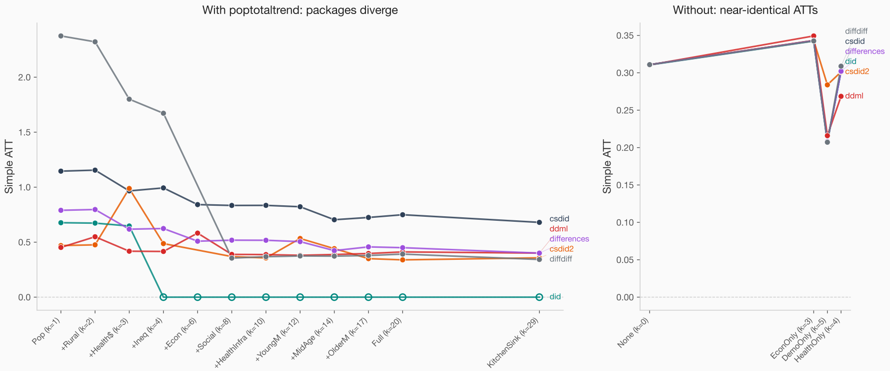
Each point is one specification. On the 45-degree line = agreement. Open circles = zero-ATT failures.

## Pairwise ATT agreement: cross-language pairs



High correlation masks systematic level shifts. `csdid` estimates consistently above the 45-degree line.

# Remove `poptotaltrend` and packages agree



Specs 13–15 exclude `poptotaltrend`. The between-package gap drops from >0.57 to <0.08 (spec 13: 0.007).

## How each package handles near-singularity drives the divergence

### The evidence

Spec 13 (no `poptotaltrend`): all five packages agree to 0.007.

Specs 02–04 (with `poptotaltrend`): range exceeds 0.57.

**Average gap is 12× larger with `poptotaltrend`.**

OR models differ in every spec (OLS, WLS, ML) — yet packages agree when `poptotaltrend` is absent.

### The logic

OR differences are constant across specs. If OR drove divergence, **all** specs would disagree.

**Only specs with `poptotaltrend` diverge.**

`poptotaltrend` creates near-singularity in the **propensity score**. Each package handles that singularity differently.

## Why This Happens

Inside the estimator you never see

## The DR estimator inverts a matrix you never see

What most users think happens:

```
att_gt(..., xformla = ~X) ⟶ ATT
```

What actually happens inside:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

The doubly robust estimator fits both a propensity score model and an outcome regression. Both require inverting $\mathbf{X}'\mathbf{X}$.

If this matrix is **numerically singular**, the inverse is garbage.

## Computers store numbers with finite precision

Every number in R, Stata, and Python is a 64-bit "double."

It stores $\sim$15–16 significant digits. After that, it rounds.

**Machine epsilon:** $\varepsilon \approx 2.22 \times 10^{-16}$

The smallest number where $1 + \varepsilon \neq 1$.

When two columns of $\mathbf{X}$ differ by a factor of $10^{10}$,
the computer cannot distinguish the small column from rounding error.

## The condition number measures how close a matrix is to breaking

$$\kappa(\mathbf{X}) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Ratio of largest to smallest singular value of $\mathbf{X}$

**The rule:**

When $\kappa(\mathbf{X}'\mathbf{X}) > 1/\varepsilon \approx 4.5 \times 10^{15}$,

the computer cannot reliably invert the matrix.

If $\kappa$ is too large, the propensity scores are garbage
— and the package may not tell you.

## `poptotaltrend` is 10 billion; `rural` is 0.4

Condition number of the design matrix $\mathbf{X}$

| Scenario | $\kappa(\mathbf{X})$ | Status |
|---|---:|---|
| Raw (with `poptotaltrend`) | 4,100,000,000 | **SINGULAR** |
| Divide by 1,000 | 4,100,000 | OK |
| Z-score standardize | $\sim 1$ | **Perfect** |
| Drop `poptotaltrend` | 121,000 | OK |

$\kappa(\mathbf{X}'\mathbf{X}) = \kappa(\mathbf{X})^2 \approx 10^{19} \gg 1/\varepsilon \approx 4.5 \times 10^{15}$. The inverse is numerically meaningless. High $k$ alone is **not** the problem — the scale mismatch is.

## Each package handles singularity differently

| **Package** | **Check** | **Result** |
|---|---|---|
| `did` (R) | `rcond(X'X)` $< \varepsilon$ | Aborts cell $\rightarrow$ 0/NA. Silent. |
| `differences` (Py) | None | Pseudoinverse; DR carries estimate. |
| `diff-diff` (Py) | Catches exceptions | Unconditional fallback. Silent. |
| `csdid` (Stata) | Unknown | Estimates $\sim 2\times$ larger. No diagnostic. |

None of the six packages warns the user about covariate scaling.

## Variance Decomposition

How much variation comes from package choice?

## Three sources of variation — and one of them shouldn't exist

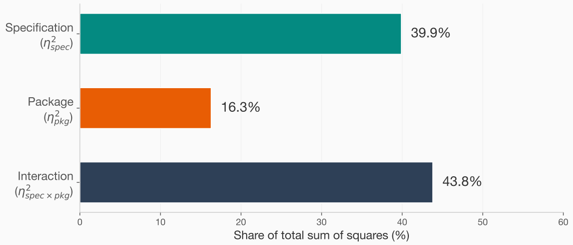
Total variation in ATT estimates across specifications and packages:

$$\mathrm{SS}_{\text{total}} = \mathrm{SS}_{\text{spec}} + \mathrm{SS}_{\text{pkg}} + \mathrm{SS}_{\text{interaction}}$$

| | |
|---|---|
| $\mathrm{SS}_{\text{spec}}$ | Variation due to specification choice (changing covariates, holding package fixed) |
| $\mathrm{SS}_{\text{pkg}}$ | Variation due to package choice (changing package, holding specification fixed) |
| $\mathrm{SS}_{\text{interaction}}$ | Variation from packages responding *differently* to the same specification change |

Effect size: $\eta^2 = \mathrm{SS}_{\text{source}} \,/\, \mathrm{SS}_{\text{total}}$

# 44% of variation is interaction



| **Source** | $\eta^2$ |
|---|---|
| Specification | 40% |
| Package | 16% |
| Interaction | 44% |

The large interaction term means packages **break at different specifications in different ways.**

This is not random noise.
It is systematic disagreement.

(All 6 tested packages.)

## Package choice is a substantive decision

Package choice is not a convenience.

It determines whether your ATT is 0.00, 0.45, 1.15, or 2.38.

Yet no journal requires reporting which package was used.

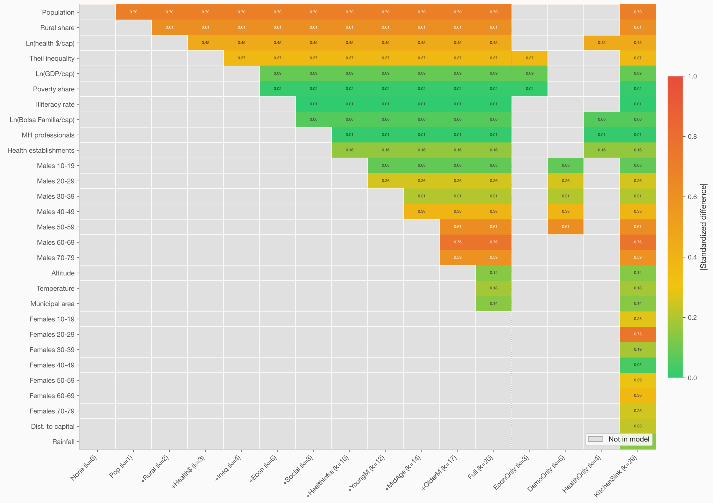
No referee asks whether the covariates were standardized.

No replication protocol tests multiple implementations.

## Balance & the Hidden First Stage

What standardized differences mask

# Balance looks fine — but the estimator doesn't care about balance



## Balance normalizes for scale — but the estimator does not

Standardized mean differences (SDM) divide by
the pooled standard deviation.

A covariate with mean $= 10^{10}$ and SD $= 10^9$
can have SDM $=$ 0.05 (“balanced”)

while making $\mathbf{X}'\mathbf{X}$ numerically singular.

Balance diagnostics report covariate overlap.
They say nothing about the numerical stability
of the matrix inversion happening inside the estimator.

## Round 4a Results

Did standardization work?

## Standardization fixed the numerical singularity

### Round 4a design

All 96 scripts re-run with:

1. Z-score standardized covariates (mean $=$ 0, sd $=$ 1)
2. No external probit trimming
3. Packages estimate their own internal propensity scores **Result:**

**96/96 cells populated.**

Zero NaN. Zero missing.

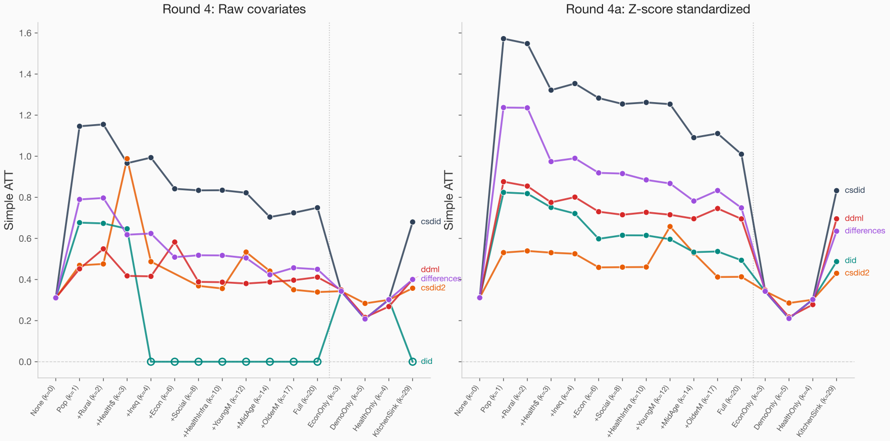
### `did` zero-recovery

| Spec | R4 | R4a |
|---|---|---|
| 05 ($k$=4) | 0.000 | 0.721 |
| 06 ($k$=6) | 0.000 | 0.598 |
| 07 ($k$=8) | 0.000 | 0.615 |
| 08 ($k$=10) | 0.000 | 0.615 |
| 12 ($k$=20) | 0.000 | 0.494 |
| 16 ($k$=29) | 0.000 | 0.487 |

All 9 zero-valued specs recovered.

The `rcond` check now passes.

# But packages still diverge after standardization



5 packages (excl. diffdiff). Left: R4 with `did` zeros and wide fans. Right: R4a fills the zeros but the fan persists.

## Near-separation, not condition numbers, drives the residual gap

### What standardization fixed

Condition number $\to 1$.
`rcond` check passes.
No more silent zeros.

All packages now produce real ATT estimates for every specification.

### What it cannot fix

`poptotaltrend` creates extreme outlier municipalities with near-degenerate propensity scores.

Z-scoring preserves outlier structure.

Different logit optimizers handle near-separation differently $\to$ different $\hat{p} \to$ different ATTs.

## Why `csdid` diverges: inverse probability tilting amplifies extreme weights

**MLE logit** (`did`, `differences`)

Maximizes the likelihood of the observed treatment assignments.

Near-separation $\rightarrow$ coefficients grow large but the optimizer stops at a finite maximum.

Result: extreme $\hat{p}$ values, but bounded by the optimizer's tolerance.

**IPT** (`csdid`, `csdid2`)

Finds weights that **exactly balance** covariates between treated and controls.

Near-separation $\rightarrow$ IPT assigns extreme weights to achieve exact balance, even when MLE logit would not.

Result: more aggressive reweighting $\rightarrow$ larger ATT estimates.

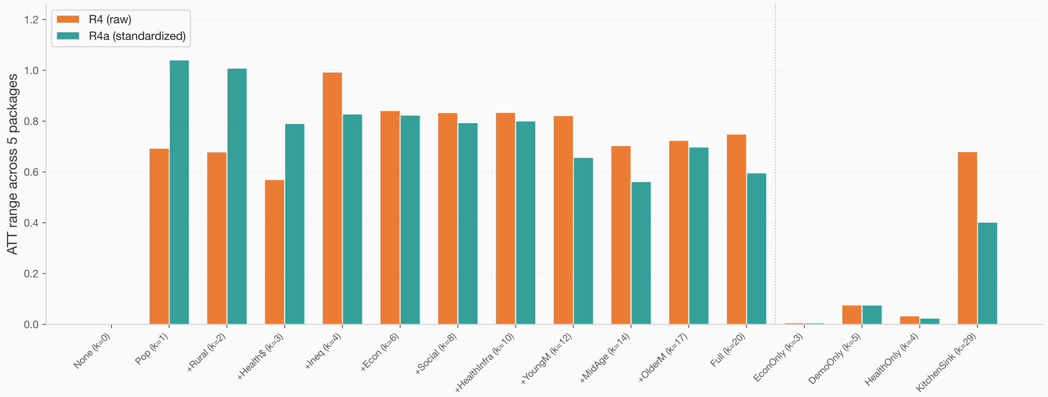
## Interaction variance fell by a third — packages now move together

Two-way ANOVA: $\eta^2$ decomposition (5 packages, excluding diffdiff)

| **Source** | **R4 (raw)** | **R4a (z-scored)** | **Change** |
|---|---|---|---|
| Specification ($\eta^2_{\text{spec}}$) | 49% | **53%** | $\approx$ |
| Package ($\eta^2_{\text{pkg}}$) | 30% | 33% | $\approx$ |
| Interaction ($\eta^2_{\text{spec}\times\text{pkg}}$) | 21% | **15%** | $\downarrow$ |

Packages still disagree on **levels**, but they now respond more uniformly to specification changes.

## Where packages converge and where they don't



Specs 05–16: convergence (range narrowed). Specs 02–04: divergence grew because `csdid` inflated further. Specs 13–15: near-perfect agreement in both rounds.

## Three requirements for applied DiD researchers

**1. Report the package and version.** Even after standardization, package choice produces $2\times$ different estimates.

**2. Standardize covariates before estimation.** It eliminates numerical singularity and reduces interaction variance by a third.

**3. Run a $k=0$ baseline.** Confirm packages agree unconditionally before adding covariates.

## AI agents are built for exactly this kind of problem

### Why cross-package audits don't happen

Writing identical specifications in three languages is tedious.

Diagnosing numerical edge cases requires reading source code.

The whole exercise is high-value but brutally time-consuming.

### Why AI agents should do it

**High time cost.** Months of work for a human team.

**High stakes.** Wrong estimates propagate through literatures and feed publication bias.

**Easy to get wrong.** One mismatched flag across languages invalidates the comparison.

Unknown between-package variation is a previously undocumented source of publication bias. AI agents can help us find it — across every estimator, not just CS (2021).

## The packages didn’t agree.

And none of them told you.