undid: A Stata package for difference-in-differences with unpoolable data

Eric B. Jamieson Nichole Austin Sunny Karim Erin Strumpf
Matthew D. Webb

2025 Canadian Stata Conference — Poster Presentation

Motivation

Existing difference-in-differences (DiD) models implicitly assume that data is poolable. However, there are many research questions that necessitate working with private or confidential data which is not allowed to be exported outside of its respective silo.

Examples of data which is often unpoolable include:

- health care data from different insurers, provinces, or countries
- restricted use micro-data in secure research environments

Methodology

- For the simple computation of the average treatment effect on the treated (ATT) with two periods and two silos, note that it is possible to compute each difference locally at each silo:

$$\text{ATT} = \underbrace{\left(\overline{Y}_{T,1} - \overline{Y}_{T,0}\right)}_{\lambda_{\text{Treated},\,1-0}} - \underbrace{\left(\overline{Y}_{C,1} - \overline{Y}_{C,0}\right)}_{\lambda_{\text{Control},\,1-0}}$$

- To allow for covariates and staggered adoption, $\lambda_{silo,post-pre}$ is calculated by the regression:

$$Y_{i,t} = \alpha + \lambda D_t + x_i^{'\beta+\epsilon}$$

- where $t \in t_0, t_1$ with $t_0$being the before treatment period and t_1 the relevant posttreatment period.
- D_t is equal to 1 if the observation is post treatment time, and 0 otherwise, and $x_i^{'}$ are the vectors of covariate values

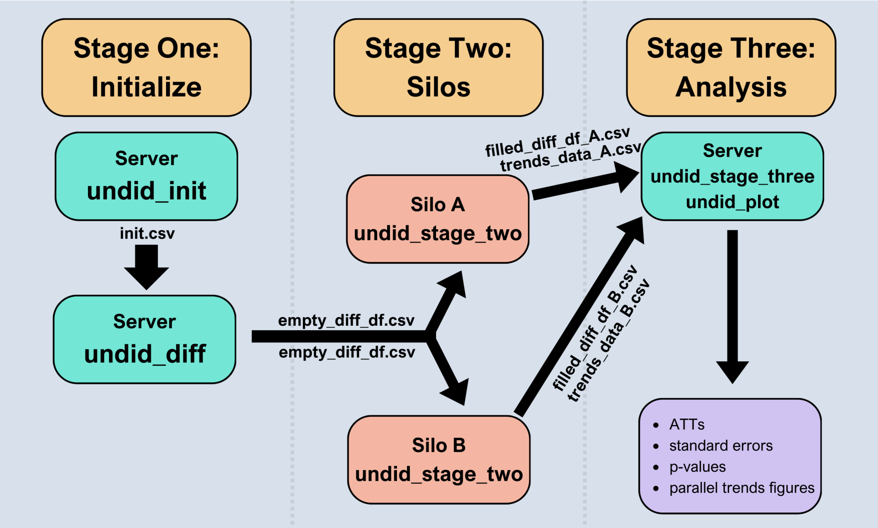
Methodology - continued

- For staggered adoption, every relevant value of λ is calculated such that each post-treatment period is compared to the pre-treatment period.
- The sub-aggregate ATTs are then computed via a regression such as the following, where *treat* is a binary variable indicating if the λ value comes from a treated silo:

$$\lambda_{silo,g,t} = \alpha + \text{ATT}\, treat_{silo,g,t} + epsilon \text{ if } G = g \text{ and } T = t$$

- where G is the treatment time, and T is the specified posttreatment period
- Finally, the aggregate ATT is computed as a weighted mean of the $\text{ATT}_{g,t}$ terms.

undid flowchart



Stage One: Initialize

- **undid_init:** Researchers specify the start and end dates for the analysis, the silos, the treatment times, and any covariates. Creates a CSV file.
- **undid_diff**: The CSV file created by undid_init is transformed into a larger CSV file with embedded instructions telling each silo the time periods for which λ values must be calculated, and what covariates, if any, should be included.

Stage Two: Silos

- **undid_stage_two:** With the local silo's data loaded, users call this command to produce two CSV files: one with the λ values; and the other with the mean of the outcome variable (and its residualized analog, if covariates were specified) by period.

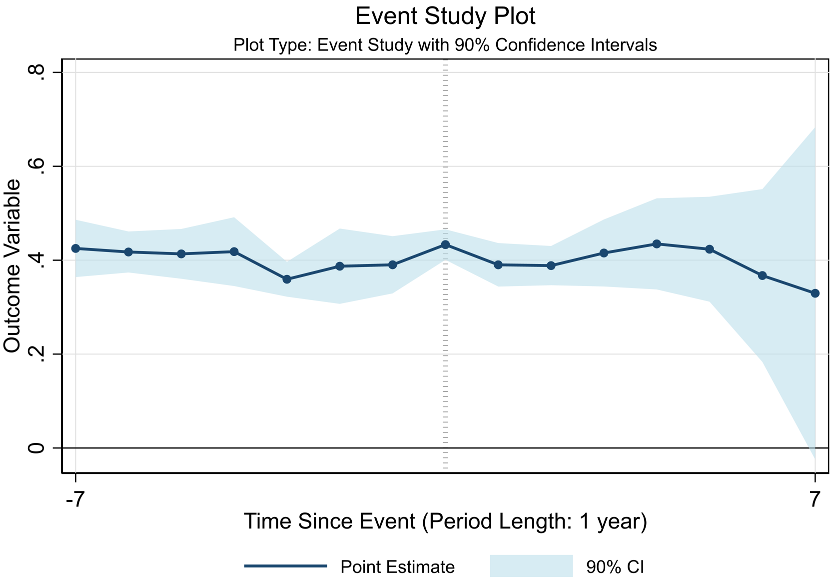
Stage Three: Analysis

- **undid_stage_three:** Researchers specify a path to the folder containing all the CSV files with the computed λ values from every silo. Sub-aggregate ATTs, the aggregate ATTs and all related standard errors and p-values are returned.
- **undid_plot**: Given a path to the folder with the trends data CSV files, produces parallel trends and event study plots.

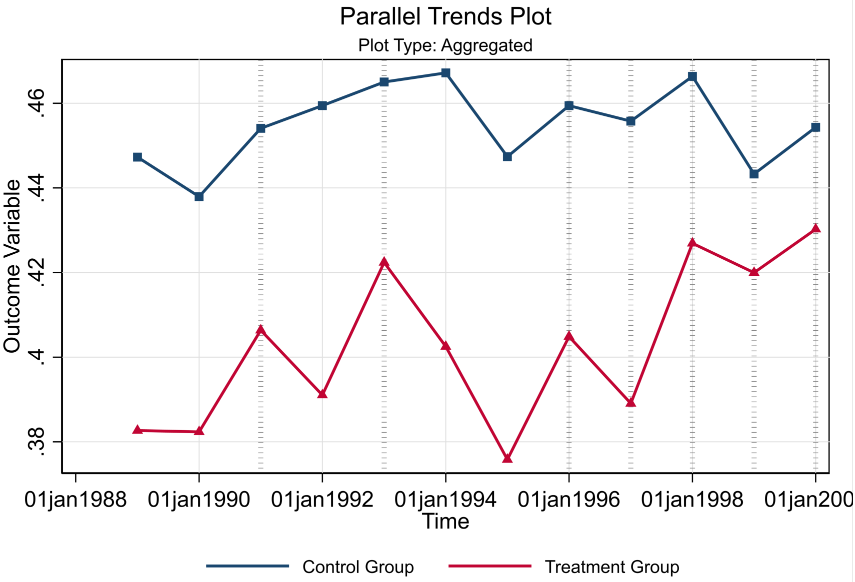
```
. undid_stage_three, dir_path("test_csv_files\stage_three\staggered")
-----------------------------------------------------------------------------------------------------
                                   undid: Sub-Aggregate Results
-----------------------------------------------------------------------------------------------------
Sub-Aggregate Group       | ATT               | SE       | p-val    | JKNIFE SE    | JKNIFE p-val   | RI p-val
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1991                      |0.0529100          | 0.022    | 0.017    | 0.024        | 0.030          |0.524     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1993                      |0.0235928          | 0.017    | 0.155    | 0.019        | 0.204          |0.685     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1996                      |0.0564351          | 0.024    | 0.021    | 0.029        | 0.057          |0.504     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1997                      |0.0711167          | 0.023    | 0.002    | 0.027        | 0.009          |0.208     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1998                      |0.0485436          | 0.033    | 0.143    | 0.039        | 0.213          |0.485     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
1999                      |0.0120440          | 0.015    | 0.424    | 0.021        | 0.561          |0.867     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|
2000                      |-0.0330623         | 0.032    | 0.308    | 0.096        | 0.732          |0.687     |
--------------------------|-----------------|--------|--------|------------|--------------|---------|


------------------------------
   undid: Aggregate Results
------------------------------
Aggregation: g
Weighting: both
Aggregate ATT: .04582252
Standard error: .01159691
p-value: .00752668
Jackknife SE: .01329357
Jackknife p-value: .01368368
RI p-value: .146
Permutations: 1000
```

```
undid_plot, dir_path("filepath") plot("event")
ci(0.90) event_window(-7 7)
```



```
undid_plot, dir_path("filepath")
```



Data Harmonization

- As written, undid assumes that all silos have the same covariate names
- If covariates were specified in the first stage, but are not present when running `undid_stage_two`, a warning is displayed suggesting the user rename covariates using the spellings from the first stage
- Additionally, any sample selection or data cleaning routines must be done before running `undid_stage_two`
- Future work will explore whether different codings, for control variables, across silos is valid

Package Availability

- The `Stata` package is available here: `https://github.com/ebjamieson97/undid`.
- Also available on SSC
  `ssc install undid`

Other Languages

- Available in `R` on CRAN
- see `undidr` Jamieson [2025]
- The `Julia` program can be found here: `https://github.com/ebjamieson97/Undid.jl`.
- The `Python` version works as a wrapper to call the `Julia` program, and can be found here: `https://github.com/ebjamieson97/undidPyjl`.

Related Papers

- The undid package implements the new estimator described in Karim, Webb, Austin, and Strumpf [2024]
- Karim, Webb, Austin, and Strumpf [2025] explores cluster robust inference, and state-level policy effects using undid
- undid can handle some forms of CCC violations described in Karim and Webb [2024]
- Clustering is handled either through Randomization Inference (RI-β) based on MacKinnon and Webb [2020] or the jackknife [MacKinnon, Nielsen, and Webb, 2023]

Contact Info

- Eric B. Jamieson — `ericbrucejamieson@gmail.com`
- Nichole Austin — `nichole.austin@dal.ca`
- Sunny Karim — `SunnyKarim@cmail.carleton.ca`
- Erin Strumpf — `erin.strumpf@mcgill.ca`
- Matthew D. Webb — `matt.webb@carleton.ca`

References I

Eric Jamieson. *undidR: Difference-in-Differences with Unpoolable Data*, 2025. URL https://CRAN.R-project.org/package=undidR. R package version 1.0.2.

Sunny Karim and Matthew D Webb. Good controls gone bad: Difference-in-differences with covariates. *arXiv preprint arXiv:2412.14447*, 2024.

Sunny Karim, Matthew D Webb, Nichole Austin, and Erin Strumpf. Difference-in-differences with unpoolable data. *arXiv preprint arXiv:2403.15910*, 2024.

Sunny Karim, Matthew D. Webb, Nichole Austin, and Erin Strumpf. Which policy works and where? estimation and inference of state level treatment effects using difference-in-differences. Mimeo, 2025.

James G. MacKinnon, Morten Ø. Nielsen, and Matthew D. Webb. Fast and reliable jackknife and bootstrap methods for cluster-robust inference. *Journal of Applied Econometrics*, 38:671–694, 2023.

References II

James G MacKinnon and Matthew D Webb. Randomization inference for difference-in-differences with few treated clusters. *Journal of Econometrics*, 218(2):435–450, 2020.