DiD Toolbox Documentation: Tools for Difference-in-Differences Analysis

Dr. **Ralf Elsas-Nicolle (LMU Munich, Germany, Version as of 10/07/2025)**

Table of Contents

I General Overview of Purpose, Structure, and Implementation....2
Purpose....2
Disclaimer....2
Overview Codebase and Theoretical Foundations....2
Usage Illustrations....3
II Description of Different Tools for Analysis....3
II.0 General....3
II.1 Simulating data....4
Syntax....4
Output....6
II.2 Data Description Utility....6
Syntax....6
Output....7
II.3 Plotting Utility....7
Syntax....7
Example....7
II.3 Diagnostic Tool: Bacon Decomposition....8
Role of Negative Weights....8
Syntax Example....9
III Estimator Descriptions and Syntax....9
III.1 Two-Way Fixed Effects (TWFE)....9
Estimation Strategy....9
Use Case & Caveat 10
Syntax Example 10
Output....10
III.2 Flexible Estimator: Wooldridge (TWM)....10
Estimation Strategy....10
Syntax:....11
Output....11
III.3 Robust/Efficient Estimator: Borusyak, Jaravel, Spiess (BJS)....12
Estimation Strategy 12
Syntax....12
Output....12
III.4 Robust Estimator: de Chaisemartin/D'Haultfœuille Estimator (DID_M)....13
Estimation Strategy 13
Diagnostic....13
Syntax 13
Output....13
III.5 Callaway / Sant'Anna 2021 (CS)....14
Estimation Strategy....14
Aggregations....14
Models with covariates (Sant'Anna/Xu 2025)....15
Syntax....16
Output....16
III.VI Interaction Weighted (IW)....17
Estimation Strategy....17
Syntax....18

Output....18
IV Usage Guidelines.... 18
V References.... 19

I General Overview of Purpose, Structure, and Implementation

Purpose

The DiD Toolbox is a set of Matlab tools designed for applied statisticians and econometricians to conduct Difference-in-Differences (DiD) analyses, particularly focusing on designs involving staggered treatment timing. The traditional Two-Way Fixed Effects (TWFE) estimator, while common, yields estimates that may be biased and difficult to interpret when treatment effects are heterogeneous across groups or vary over time. The primary goal of this toolbox is to address these methodological challenges by providing modern, robust estimators that yield valid causal estimates in complex multi-period, staggered adoption scenarios.

Disclaimer

- The toolbox is under development, and needs more internal structure. Do use with care and be aware that the current state may not be "production ready" yet.
- Most of the code base was developed with the help of ChatGPT 5 Thinking. The documentation was developed also using features of NotebookLM by Google.
- However, I backtested all of the estimators and corresponding results using simulated data with Stata 18, utilizing implementations by Stata Corp. itself and some of the packages designed by the underlying paper's authors. Also see the provided example file, which has a Stata log-file for comparison.

If researchers with interest in the topic and OOP skills in Matlab are interested in joining this project, I'd very much appreciate collaboration.

Overview Codebase and Theoretical Foundations

The methodologies of the toolbox are grounded in foundational and recent academic literature. The software implementation uses Object-Oriented Programming (OOP) within Matlab.

Theoretical Papers: The toolbox implements methods developed or discussed in the following key papers:

- Goodman-Bacon (2021): Provides the mathematical foundation for the decomposition of the TWFE estimator, explaining how it operates as a weighted average of DiD estimates and identifying the source of bias from "negative weights" due to treatment effect heterogeneity.
- Wooldridge (2021): Establishes the algebraic equivalence between the Two-Way Fixed Effects (TWFE) estimator and the Two-Way Mundlak (TWM) regression, enabling flexible implementation using pooled OLS.
- Borusyak, Jaravel, and Spiess (2024): Derives the efficient and robust imputation estimator (BJS) for staggered DiD, which estimates counterfactual outcomes using only untreated observations to calculate heterogeneous causal effects, providing efficiency and avoiding spurious identification.

- de Chaisemartin and D'Haultfœuille (2020): Proposes the DID_M estimator, which estimates a robust Average Treatment Effect across switching cells (δ_S) and introduces robustness measures for assessing TWFE bias, particularly in designs where weights may be negative. And the unique feature is that the estimator can handle an on/off treatment, while all other estimators assume that treatment is an absorptive state.
- Callaway / Sant'Anna (2021): CS focus on cohort/time based analyses, looking at $ATT(g,k)$ with a focus on taking covariates into account. They derive identification, estimation, and inference strategies assuming parametric nuisance models (i.e. linear outcome regression, logit for propensity scores) that admit standard regularity conditions. In an extension, the interaction weighted estimator by Sun/Abraham (2021) builds on this work, showing how get unbiased event studies in this setting.

Codebase Structure: The Matlab source code utilizes OOP principles under the +did/ package namespace:

- *Core Classes and Functions*: fit.m, Model.m (orchestration), getEstimator.m, Dataset.m (data management).
- *Utilities*: descData.m (descriptive statistics for DiD data), genDIDdata.m (simulation utility).
- *Estimation Modules*: Specific estimators are located in the +did/+estimators/ folder, including implementations for the *de Chaisemartin/d'Haultfœuille* estimator (ch_estimator.m), B*orusyak, Jaravel, and Spiess* (BJS) estimator (bjs_imputation.m), *Wooldridge* (TWM) estimator (wooldridge_TB.m) as well as *Callaway/Sant'Anna* (cs_estimator.m) with the extension of interaction weighing by *Sun/Abraham* (iw_estimator.m).
- Covariance Estimation: The +did/+vcov/ package handles various variance-covariance strategies, including WildBootstrap.m and Clustered.m.

Usage Illustrations

The toolbox provides an illustration on how to use it, available as a Matlab mlx-file and pdf in the folder "Examples":

- RealWorld_Example_DiDToolbox.mlx (with a complementary Stata 18 log-file, analyzing the same data for comparison).

II Description of Different Tools for Analysis

II.0 General

Throughout the toolbox, data is described by four key variables:

The function requires a panel data table (T) and mandatory name-value arguments for identifying necessary items for conducting DiD analyses (*idVar*, *timeVar*, *dVar*).

- Input data table (*T*)
- *idVar*: Identifier for units in the panel data
- *timeVar*: Time indicator
- *dVar*: Identifier per individual and time point whether treated or not (dummy)

When using commands, these are the names of input field (except for *T*), where you provide in the function call the actual (and potentially different) names of items in your data (e.g. timeVar="t").

You should start your analysis of data T by instantiating a validated data object, similar to the following:

ds = did.Dataset.fromTable(T, idVar="id", timeVar="time", dVar="D", yVar="y",describe=true);

The typical command to conduct estimations then is called using *did.fit(ds,options)*

All functionality can also be called by using the full syntax similar to the following:

resWool = did.fit("Wooldridge", T, idVar="id", timeVar="time", dVar="D", yVar="y",options);

but the recommended proceeding is by using a data object.

There are three exceptions:

- Generating simulated panel data with or without different forms of treatment using *did.genDIDdata(.)*
- *did.Dataset.fromTable(.) i*s the first initiation of the data object and thus requires the definition of the data table and the key variables' names.
- The only command that is estimator specific is a plotting utility. It needs to be called based on an estimator's output (instead of the data object *ds*),:

d*id.did_plot(resWool, "event");   % event-study*

II.1 Simulating data

For understanding and backtesting the estimators in this toolbox (or DiD settings in general), genDIDdata.m generates data suitable for DiD analyses. It generates the outcome variable by combining components such as the base outcome (`y_base`), factoring in influences from time-varying covariates (`yX`), the true treatment effect (`tau_true`), and an error term (`epsilon`). You can set the number of individuals, how many are treated and what are the treatment effects and their time patterns.

Syntax

Mandatory: *numPeriods*, *numIds* and *percTreated* are mandatory input arguments and shall be directly provided in this order.

The function supports different characteristics of the treatment effects:

treatType="constant"|"constantTime"|"timeIncrease"|"onOff"

- Type "*constant*": The canonical DiD type, where all treatments have the same magnitude and happen at the same time (over time and over cohorts).
- Type "*constantTime*": Treatment is staggered, numCohort is the number of cohorts (with different start of treatment), *CohortIncrease* is the difference in ATT per cohort / first treatment time.
- Type " *timeIncrease*" increases the treatment effect within cohort over time after first treatment (*dynEffect* gives the change per period)

- Type "*onOff*" generates non-absorptive treatment, ending in *endPeriod* (a vector with length equal to *numCohorts*)

Cohorts can be explicitly set w.r.t *CohortTimes* (timing of treatment start), *CohortSize* (absolute size of each cohort) or *CohortProbs* (fraction of each cohort (vector length of *cohortProbs* must equal number of cohorts)

cohortTimes	explicit adoption times to use
numCohorts	if set, auto-pick this many evenly spaced times
cohortProbs	length == numel(cohortTimes) probabilities
cohortSize	length == numel(cohortTimes) exact counts; sum == #treated
treatedNum	optional: force exact number treated

The function supports various options for **pre-trend**s, including `"none"`, `"unitLinear"`, `"unitQuadratic"`, or `"shock" (not tested yet)`.

Example

```
    clear;
    T = did.genDIDdata(12, 500, 0.6, ...
        treatType="constantTime", startPeriod=4, ...
        cohortTimes=[4 7 10],xNum=2,                ... % creates x1, x2
        betaX=[0.6, -0.4],          ... % effect of x1,x2 on y
        xUnitStd=1, xTimeStd=1, xShockStd=1, CohortIncrease=1.5);

    % Instantiate data object:
    ds = did.Dataset.fromTable(T, idVar="id", timeVar="time", dVar="D",
yVar="y",describe=true);
```

```
[DataDesc] Units = 500 | Periods = 12 | Nobs = 6000 | Balanced Units = 500 (100.0%)
[DataDesc] Ever-treated = 279 (55.8%) | Never-treated = 221 (44.2%) | Leavers = 0 (0.0% of ever)
Cohorts (first 6 rows):
    cohort_label    N_units    Share    N_leavers    ShareLeaversAmongCohort    FirstTreat_time
    ____________    _______    _____    _________    _______________________    _______________

        "4"             85      0.17        0                   0                      4
        "7"            101     0.202        0                   0                      7
        "10"            93     0.186        0                   0                     10

Outcome Y (overall) - N/Mean/SD/Min/P25/Median/P75/Max:
     N       Mean       SD        Min        P25        Median      P75        Max
    ____    ______    ______    _______    ________    ______    ______    ______

    6000    1.8214    3.1707    -9.0163    -0.38658    1.693     3.8883    13.131

Outcome Y by cohort (including 'never'):
    cohort_int    cohort_time    cohort_label     N       Mean        SD        Min         P25         Median
    __________    ___________    ____________    ____    _______    ______    _______    _________    _______

         0            NaN          "never"       2652    0.84289    2.7249    -9.0163     -1.0002     0.86612
         4              4          "4"           1020     2.4959    2.7356    -5.6815      0.5654      2.5105
         7              7          "7"           1212     2.8253    3.4262    -7.5883     0.44748      2.8503
        10             10          "10"          1116     2.4399    3.5603    -6.2428    -0.092498     2.0881
```

Note: In this example, the true treatment effects are set to 2, 3.5 and 5 for the cohorts treated at time 4,7, and 10, respectively. As we know the true $ATT(g,t)$ in the simulated data, this allows to compare the estimated coefficients between different methods under the same degrees of data characteristics (like covariates, error term variance etc.).

Output

Key variables in the generated data table are the identifier items *id* (idVar), *time* (timeVar), *y* (yVar) and *D* (dVar), where the latter is the indicator variable for actual treatment.

Output Field	Type	Description
ATT (Output)	vector	The true average treatment effect used in the simulated data.
eventTime (Output)	vector	Event time (periods since treatment).
g (Output)	vector	Cohort start time.
everTreated (Output)	vector	Indicator for ever-treated status.
x1 ... (Output)	vector	Additional covariates.
i_FE	vector	Fixed effect per individual
t_FE	vector	Time fixed-effects

II.2 Data Description Utility

The *did.dataDesc* function provides a quick overview of the panel dataset, summarizing essential characteristics such as unit counts, time periods, treatment assignment patterns, and outcome variable descriptives.

This utility calculates descriptive statistics by identifying the panel structure, determining treatment cohorts based on their first treatment date, and classifying units as ever-treated, never-treated, or "leavers" (units whose treatment turned off after being on). If an outcome variable is provided (yVar), overall and cohort-stratified descriptive statistics are computed.

Note that it is automatically called when instantiating the data object with the option *describe=true*

Syntax

The function requires the panel data table (T) and mandatory name-value arguments for identifying columns (*idVar*, t*imeVar*, *dVar*).

Example

```
descStat = did.dataDesc(ds);
```

```
[DataDesc] Units = 500 | Periods = 12 | Nobs = 6000 | Balanced Units = 500 (100.0%)
[DataDesc] Ever-treated = 279 (55.8%) | Never-treated = 221 (44.2%) | Leavers = 0 (0.0% of ever)
Cohorts (first 6 rows):
    cohort_label    N_units    Share    N_leavers    ShareLeaversAmongCohort    FirstTreat_time
    ____________    _______    _____    _________    _______________________    _______________

        "4"            85       0.17        0                   0                      4
        "7"           101      0.202        0                   0                      7
        "10"           93      0.186        0                   0                     10

Outcome Y (overall) – N/Mean/SD/Min/P25/Median/P75/Max:
     N       Mean      SD        Min         P25       Median     P75       Max  
    ____    ______    ______    _______    ________    ______    ______    ______

    6000    1.8214    3.1707    -9.0163    -0.38658    1.693     3.8883    13.131
```

```
  Outcome Y by cohort (including 'never'):
    cohort_int    cohort_time    cohort_label     N        Mean        SD        Min          P25         Median 
    __________    ___________    ____________    ____    _______    ______    _______    _________    _______

         0            NaN          "never"       2652    0.84289    2.7249    -9.0163      -1.0002    0.86612
         4              4          "4"           1020     2.4959    2.7356    -5.6815       0.5654     2.5105
         7              7          "7"           1212     2.8253    3.4262    -7.5883      0.44748     2.8503
        10             10          "10"          1116     2.4399    3.5603    -6.2428    -0.092498     2.0881
```

Output

- *Overview*(table): Summarizes time range, #periods, #units, N_{obs}, balance, EverTreated, NeverTreated, and Leavers.
- *Cohorts* (table): Details on treatment cohorts (first treat time), including unit counts and leaver statistics per cohort.
- *PeriodStats* (table): Statistics by time period (units observed, treated, not-yet, never, off-after-on).
- *Y_overall* (table): If yVar is provided, overall summary statistics (N, Mean, SD, Min, Median, Max).
- *Y_by_cohort* (table): If yVar is provided, summary statistics broken down by treatment cohort (including the "never" treated group) are included.
- Options (table): Identifiers provided

II.3 Plotting Utility

For convenience, a plotting utility *did_plot* is provided. After running an estimator of the toolbox (see the next section), one can easily make graphs of treatment effects, either aggregated over cohorts ("cohort") or over event time ("event").

Syntax

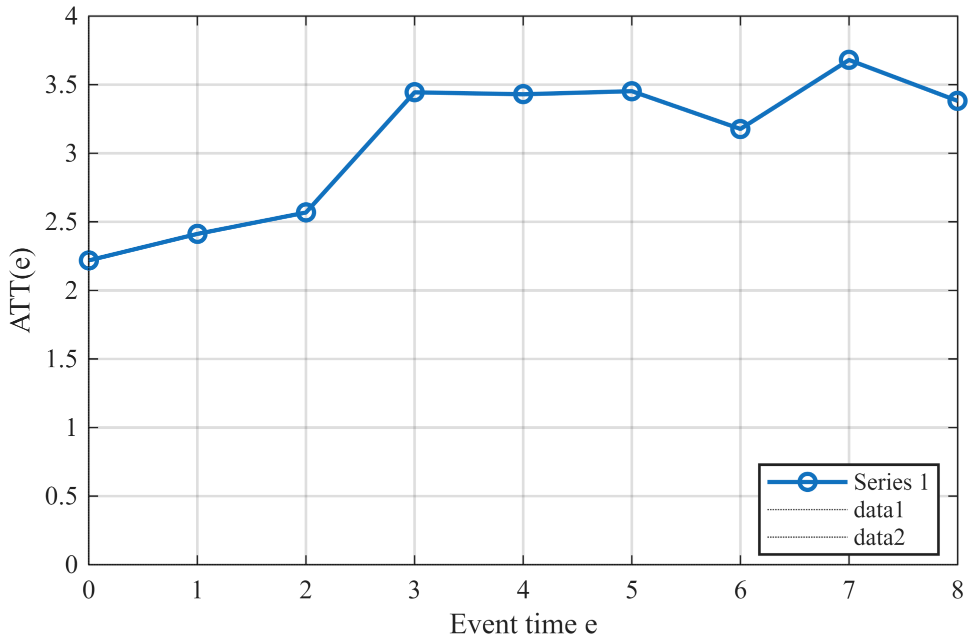
did.did_plot(resBJS, "cohort")

Example

```
resWool =  did.fit("Wooldridge", ds, Covariates=["x1","x2"]);
```

```
  [wooldridge] ATT by cohort (mean of cohorts coefficients):
    Cohort    #TimePeriods    ATT(k)       SE        tStat       pValue   
    ______    ____________    ______    _______    ______    ___________

      4            9          1.8265    0.12728     14.35     2.4185e-39
      7            6          3.4369    0.11167    30.776    2.2245e-117
     10            3          5.1427     0.1314    39.137     3.401e-154

  [wooldridge] Overall ATT (cohort-share weighted):
    Estimate       SE       tStat       pValue   
    ________    ________    ______    ___________

     3.5149     0.080312    43.765    5.8718e-173

  [wooldridge] ATT by event time (mean/median across cohorts):
    k    ATT_hat_mean    ATT_hat_median    nCells
    _    ____________    ______________    ______
    0       2.2184          1.8371          3
    1       2.4119          2.0661          3
    2       2.5679          2.1408          3
    3       3.4442          3.4442          2
    4         3.43            3.43          2
    5       3.4526          3.4526          2
    6       3.1773          3.1773          1
    7       3.6802          3.6802          1
    8       3.3819          3.3819          1
```

```
did.did_plot(resWool,"event");
```



II.3 Diagnostic Tool: Bacon Decomposition

The Bacon Decomposition, based on the work of Goodman-Bacon (2021), is a necessary diagnostic tool in staggered DiD settings. The standard TWFE estimator $\widehat{\beta}_{DD}$ is shown to be a weighted average of all possible Difference-in-Differences (DD) estimators based on the cohorts in the data. The decomposition identifies the contribution of four types of comparisons:

1. Treated vs. Untreated: Compares a treated cohort to a never-treated control group.
2. Early vs. Late Treated (Admissible Timing): Earlier-treated groups act as treatment, later-treated groups act as controls during their pre-period.
3. Late vs. Early Treated (Forbidden Comparison): Later-treated groups act as treatment, already-treated earlier groups act as controls.

Role of Negative Weights

When Average Treatment Effects (ATT) vary over time ($\Delta ATT \neq 0$), the "Late vs. Early" (forbidden) comparisons are the primary source of negative weights. This occurs because the TWFE specification implicitly uses already-treated units as controls for later-treated units, causing changes in the outcomes of the already-treated group (which include changes in their treatment effects) to be subtracted, biasing the estimate away from the desired causal parameter (VWATT). If treatment effects are constant, weights are positive.

Syntax Example

res = did.fit("bacon", T, idVar="id",timeVar="time",yVar="y",dVar="D");

Output comprises

- *res.ByType* (aggregated decomposition) and
- *res.Pairs* (pairwise comparisons)

Example:

```
res = did.fit("bacon", ds);
```

```
---- Bacon: Overall (TWFE) -----
    3.5270

Bacon: ---- Comparisons -----
               type               group1    group2     weight      estimate
    __________________________    ______    ______    ________    ________

    "Treated vs Never Treated"       4         0       0.19244     1.8216
    "Treated vs Never Treated"       7         0       0.30488     3.4466
    "Treated vs Never Treated"      10         0       0.21055     5.1427
    "Treated earlier vs later"       4         7      0.029315     2.0519
    "Treated earlier vs later"       4        10      0.053986     1.8516
    "Treated earlier vs later"       7        10      0.064148     3.3786
    "Treated later vs earlier"       7         4       0.05863     3.7377
    "Treated later vs earlier"      10         4      0.053986      5.264
    "Treated later vs earlier"      10         7      0.032074     5.0722

Bacon: ---- Comparisons agg. by Type ----
               Type               Weight     Estimate
    __________________________    _______    ________

    "Treated earlier vs later"    0.14745     2.5557
    "Treated later vs earlier"    0.14469      4.603
    "Treated vs Never Treated"    0.70786     3.5093
```

III Estimator Descriptions and Syntax

The toolbox provides implementations for the standard TWFE estimator and five robust estimators designed for staggered adoption settings.

Default for **standard errors** is always *idVar*, unless the specific estimator requires a specific method. If you want other clustering options, you have to set "vcov"= *clusterMethod*. If clustering on one or two specific variables is desired, set "clusters" (e.g. ["id", "time"]).

III.1 Two-Way Fixed Effects (TWFE)

Estimation Strategy

This method fits a linear regression model using Ordinary Least Squares (OLS) including full sets of unit fixed effects (α_i) and time period fixed effects (β_t): $Y_{it} = \alpha_i + \beta_t + \tau D_{it} + \epsilon_{it}$

The estimate $\hat{\tau}_{FE}$ is numerically the same as the coefficient obtained from the Two-Way Mundlak (TWM) regression without covariates.

Use Case & Caveat

TWFE was until recently the standard for DiD analyses, but in staggered designs with heterogeneous effects, $\hat{\tau}_{FE}$ identifies a variance-weighted average treatment effect on the treated plus bias terms. If negative weights occur, or weights of "forbidden" comparisons (with already treated groups) are substantial, the estimate can be misleading.

Quite generally speaking, if treatment is staggered, potentially differs between cohorts or is an on/off treatment, use the Bacon decomposition to analyze heterogeneous treatment effects of cohorts, and if it is indicated, use BJS, Wooldridge or DID_M estimator (the choice depends on your use case, see the estimators' descriptions).

Syntax Example

res = did.fit("twfe", T, idVar="id",timeVar="time",yVar="y",dVar="D", vcov="clustered", clusters=["id","time"]);

Output

The function returns only the coefficient for the interaction isTreatmentGroup * isTreated and coefficients on covariates, if provided.

Example

```
res = did.fit("twfe", ds, Covariates=["x1","x2"],...
                vcov="clustered", clusters=["id","time"]);
```

```
[TWFE] ATT(D) = 3.551297
[TWFE] Standard errors clustered by id, time
[TWFE]
  3×5 table

    Name    Estimate       SE        tStat       pValue  
    ____    ________    ________    _______    __________

    "D"       3.5513     0.33485     10.606    4.0953e-07
    "x1"     0.61429    0.028508     21.548    2.3984e-10
    "x2"    -0.40942    0.018823    -21.752    2.1676e-10
```

III.2 Flexible Estimator: Wooldridge (TWM)

Estimation Strategy

This estimator uses the algebraic equivalence of the TWFE estimator and the Two-Way Mundlak (TWM) regression. The TWM regression is a pooled OLS regression that includes the time-varying covariates, unit-specific time averages ($\overline{xi.}$) and period-specific cross-sectional averages ($\overline{x}.t$). The key is that the two-way fixed effects estimate ($\widehat{\beta}_{FE}$) is identical to the coefficient on x_{it} obtained from the TWM regression ($\widehat{\beta}_M$).

This equivalence simplifies modeling complex heterogeneity (e.g., cohort-specific TEs, time-varying effects, and flexible covariate interactions), allowing the user to implement highly flexible versions of TWFE using the computational simplicity of pooled OLS. In the staggered case, the approach can be used to identify cohort/calendar time combinations that have causal interpretations.

Standard errors are clustered by the input "clusters". Stata uses clustering on the cohort level (then use a variable that identifies the cohorts, e.g. clusters="gvar"). Alternatively, one could use clustering on the individuals level (i.e. use e.g. clusters="id").

The control group for the treated observations are the never treated observations. If these don't exist, the last treated cohort is omitted and used as control for the other cohorts.

Syntax:

res = did.fit("wooldridge", T, idVar="id",timeVar="time",yVar="y",dVar="D", vcov="clustered", clusters="id");

Output

- Overall ATT
- ATT by time
- ATT by event time
- If you set the option *details,* the full regression output with the pairwise estimates is shown

Example

```
res = did.fit("wooldridge", ds, Covariates=["x1","x2"],vcov="clustered",
clusters=["id"]);
```

```
[wooldridge] ATT by cohort (mean of cohorts coefficients):
    Cohort    #TimePeriods    ATT(k)       SE          tStat       pValue
    ______    ____________    ______    __________    ______    __________

      4            9          1.8265    0.00054515    3350.4    0.00019001
      7            6          3.4369    0.00035139    9780.7    6.5089e-05
     10            3          5.1427     0.0002919     17618    3.6134e-05

[wooldridge] Overall ATT (cohort-share weighted):
    Estimate        SE        tStat       pValue
    ________    __________    ______    __________

     3.5149     0.00039059    8998.9    7.0744e-05

[wooldridge] ATT by event time (mean/median across cohorts):
    k    ATT_hat_mean    ATT_hat_median    nCells
    _    ____________    ______________    ______

    0       2.2184           1.8371          3
    1       2.4119           2.0661          3
    2       2.5679           2.1408          3
    3       3.4442           3.4442          2
    4         3.43             3.43          2
    5       3.4526           3.4526          2
    6       3.1773           3.1773          1
    7       3.6802          3.6802          1
    8       3.3819          3.3819          1
```

III.3 Robust/Efficient Estimator: Borusyak, Jaravel, Spiess (BJS)

Estimation Strategy

BJS provides the efficient linear unbiased estimator for estimating pre-specified weighted sums of treatment effects (τ_w) in staggered designs, allowing for unrestricted heterogeneity ($\mathbf{\Gamma} = \mathbf{I}$). The procedure is intuitive:

1. Fixed Effect Estimation: Estimate unit ($\widehat{\lambda}_i$) and period fixed effects ($\widehat{\delta}$) using OLS applied only to the untreated observations ($i, t \in \mathscr{I}_0$).

2. Imputation: Impute the counterfactual untreated potential outcome ($\widehat{Y}_{it}(0) = \widehat{\lambda}_i^* + \widehat{\delta}^*$) for treated observations ($i, t \in \mathscr{I}_1$).

3. Effect Calculation: The estimated treatment effect for each observation is $\widehat{\tau}it^* = Yit - \widehat{Y}_{it}(0)$, which are then aggregated by the chosen weights to estimate the target $\widehat{\tau}_w^*$.

This method is robust, avoids "forbidden comparisons," and is characterized as efficient under spherical errors. It explicitly avoids the spurious identification of long-run causal effects that plagues conventional dynamic TWFE specifications.

- Standard errors are estimated via the jackknife ("leave-one-out", LOO) method.
- If you have many individuals in the panel, you might want to set the optional parameter *useParallel* equal to the number of processors to be used with parallel processing.

Syntax

res = did.fit("BJS", T, idVar="id", timeVar="time", yVar="y", dVAr="D",BootReps=0);

Output

The output consists of

- the overall weighted ATT
- one table that shows ATT per cohort, weighted by number of obs cohrts (and s.e., nobs, CI).
- More output is returned but not printed.

Example:

```
res = did.fit("BJS",ds, Covariates=["x1","x2"],useParallel=8);
```

```
Starting parallel pool (parpool) using the 'Processes' profile ...
Connected to parallel pool with 1 workers.

[BJS] Untreated-only fit on ALL D==0: N_untreated = 4350 | Units = 500 | Periods = 12
[BJS] Treated identified: 1650/1650 (100.0%)
[BJS] Overall ATT = 3.0142 (SE=0.1033; 95% CI [2.8118, 3.2167])
ATT by Cohort
    cohort    ATT_obs    SD_obs    N_obs     SE_obs     CI_lo_obs    CI_hi_obs
    ______    _______    ______    _____    ________    _________    _________

      4        1.864     1.6371     765     0.059188     1.7478       1.9802
```

```
     7      3.4985     1.6036     606     0.065143     3.3706       3.6265
    10      5.1161     1.5903     279     0.095212     4.9287       5.3035
```

III.4 Robust Estimator: de Chaisemartin/D'Haultfœuille Estimator (DID_M)

Estimation Strategy

The DID_M estimator targets δ_S, the Average Treatment Effect (ATE) across all "switching cells"—groups whose treatment status changes between $t-1$ and t. This is calculated as a weighted average of two specific DiD terms:

◦ $DID_{+,t}$: Compares joiners (treatmen t$0 \rightarrow 1$) to groups remaining untreated.

◦ $DID_{-,t}$: Compares leavers (treatment $1 \rightarrow 0$) to groups remaining treated.

DID_M is valid under heterogeneous treatment effects and inherently avoids the source of negative weighting bias by strictly contrasting switching groups against stable groups. In staggered adoption designs, where treatment is typically irreversible, DID_M is a weighted average only of the $DID_{+,t}$ estimators.

Note

- Contrary to the TWFE, Wooldridge and BJS estimators, DID_M measures treatment effects only at the period of entry and exit, i.e. not over all periods of treatment.
- DID_M is the only estimator (to my knowledge) which can handle on/off-treatments. i.e. not assuming that treatment is an absorptive state.
- As the standard errors are determined by a jackknife bootstrap ("leave one out", LOO), this can take some time depending on the group size of the clustering variable. If the parallel toolbox is available, you can calculations speed up by setting the option *"useParallel"=true*.

Diagnostic

The Placebo DID_M ($DID_{M,pl}$) is computed, comparing pre-switch outcome trends ($t-2$ to $t-1$) for switching groups versus stable groups. A significant placebo estimate suggests the parallel trends assumption (Assumptions 4, 5, 9, and 10 in the paper) is violated.

Syntax

resDIDM = did.fit("DID_M", T, idVar="id", timeVar="time", yVar="y", dVAr="D");

Output

The function returns

- an overall weighted estimate of the ATT (labeled DID_M) and separate coefficient estimates for "Joiner_Only" and "Leavers_only". I addition, the weighted overall estimate for the placebo test is provided (Placebo_DIDI_M).
- the coefficients for each cohort, differentiated by the period of entry (DID_plus) and end of treatment (DID_minus).

Example:

```
res = did.fit("DID_M", ds,Covariates=["x1","x2"]);
```

```
=== DID_M summary (table) ===
            Effect                 Estimate      SE          t          p     
    ___________________________    ________    _______    _______    __________

    "DID_M"                          3.4596    0.17522     19.744    1.4985e-64
    "Joiners-only"                   3.4596    0.17522     19.744    1.4985e-64
    "Leavers-only"                        0          0        NaN           NaN
    "Overall (cohort-weighted)"      3.4596    0.16565     20.885    4.5119e-70
    "Placebo DID_M"                 0.13817    0.16158    0.85511        0.3929


=== DID_M by Cohort (table) ===
    t     DID_plus    DID_minus
    __    ________    _________

     4     2.0329        NaN   
     7     3.2642        NaN   
    10     4.9758        NaN   
```

III.5 Callaway / Sant'Anna 2021 (CS)

Estimation Strategy

In panels where units adopt at different times (or never), Callaway/Sant’Anna (CS) compute treatment effects for each cohort–time cell, written ATT(g,t): “the average effect at calendar time *t* for the cohort first treated in *g*.” This avoids the mixing issues of classic TWFE when effects vary over time or across cohorts.

- Groups of comparison are "never" or "notyet" treated.
- As outcomes can move before formal adoption, you can set Delta = δ ≥ 0 to exclude the δ periods before adoption from the pre-treatment baseline (*Delta*). The baseline becomes r = g − δ − 1 (instead of g − 1).
- Standard errors are estimated by a wild bootstrap type of method (e.g. *SEMethod="multiplier"*) or clustered on the unit level (SEMethod="clustered"). If there is another time-invariant source of heterogeneity between groups of data, a second cluster variable can be used.

Four possible estimation approaches are:

- *unconditional*: no covariates are taken into account (*Approach="unconditional"*).
- *OR*: Outcome regression, not taking control variables into account (*Approach="OR"*).
- *IPW*: Uses inverse probability weighting for handling the covariates**.** Estimates propensity Pr(G=g | X), reweights controls to match treated covariates, and compares mean changes (*Approach="IPW"*).
- *DR*: Combine OR and IPW (doubly robust), i.e. adjust treated by an outcome model and weight control residuals by propensities. If only one of the two models are corrected, ATT estimates will be unbiased (*Approach="DR"*).

Note that with covariates, DR is the recommended ioption, as Sat'Anna/Zaho (2020) show that even if none of the two approaches are correct, the still perform better than either of the single model approaches.

Aggregations

In a recent paper, Deb et al. (2025) show that Stata's implementation of the CS estimator uses weights for aggregation based on

- the number of treated observations in cohort c at treated time t (t>c), and
- the number of pre-period reference count for each post cell

This Deb et al. (2025) show that this can distort estimation of the overall ATET for repeated cross-sections (and was not suggested by CS):

- *Balanced panels***:** no problem - the two formulas coincide because $Nct = Nc, c - 1 N_{ct} = N_{c,c-1} Nct = Nc, c - 1$ for each cohort and time.
- *Repeated cross-sections with varying cell sizes***:** differences arise because N_{ct} varies over t. The Stata weight adds a time-invariant pre-period count to each post cell, changing relative weights across (c, t)
- *Big pre-period relative to post-period sizes***:** the distortion is largest when the pre-period sample for a cohort is large relative to its post-period sizes. (In their simulation, this pushes more weight onto later post periods and raises the aggregated ATET.)
- *Heterogeneous effects over cohorts/time***:** the more treatment effects vary across (c, t), the more harmful the wrong weights become; if all ATTs were equal, weighting wouldn’t matter.
- *Declining treated sample sizes over time*: particularly prone to bigger gaps between the correct and Stata’s aggregation.
- *Not just “overall”***:** the same weighting issue also affects event-time, by-cohort, and calendar-time averages.

The DiD Toolbox does not use the counts of pre-period observations for weighting. Instead it offers the options

- *Weighting="treatedObs",* which is the preferred weighting aschem by Deb et al. (2025) and the default, or alternatively
- *Weighting="cohortShare"*, cohort-share weights that are constant across a cohort’s post-treatment periods.

Models with covariates (Sant'Anna/Xu 2025)

When using Outcome Regression (OR), Inverse Probability Weighting (IPW), or Doubly Robust (DR), one is fitting nuisance models:

- OR: regression of ΔY on X for controls
- IPW: logistic regression for Pr(G=g|X)
- DR: both

If one uses the same sample to both estimate the nuisance functions (regressions / propensity scores) and to plug them into the ATT formula, ones risks overfitting. This bias can be especially bad when:

- X is high-dimensional
- The nuisance models are complex (ML, flexible nonparametrics)

The input parameters *CrossFit* (together with *Kfolds*) uses the kFold estimation by (Chernozhukov et al. 2018, "Double/Debiased ML") to split the sample into k folds, where one subsample is held out of the estimation. In the end, ATTs are constructed from recombining the estimates. Setting *StratifyFoldsBy="cohort"* keeps a similar share of each cohort in every fold.

The default for CrossFit is *false*, however, as the logistic regression is well behaved in most cases.

Syntax

```
    resCS = did.fit("cs", ds, ...
               Comparison="never", Delta=0, Approach="dr", 
Covariates=["x1","x2"], ...
                Seed=42, SEMethod="multiplier");
```

```
[CS] R=18 cells; Approach=dr; Comp=never; δ=0; SE=multiplier; W=treatedObs. Bootstrap B=100, crit(95%)=2.899
[CS] Overall ATT
  1×6 table

    Estimate      SE       tStat     pValue      LB        UB  
    ________    _______    ______    ______    ______    ______

     3.022      0.14829    20.379      0       2.7195    3.3245

[CS] Estimates
  18×11 table

            Name              Effect       Estimate    g     t       SE       tStat     pValue    gYear    tYear
    _________________     __________    ________    __    __    _______    ______    ______    _____    _____

    "ATT(g=4, t=4)"       "ATT(g,t)"     2.1062     4     4     0.28186    7.4727      0         4        4  
    "ATT(g=4, t=5)"       "ATT(g,t)"     2.1191     4     5     0.31839    6.6557      0         4        5  
    "ATT(g=4, t=6)"       "ATT(g,t)"     2.4465     4     6     0.28711    8.5211      0         4        6  
    "ATT(g=4, t=7)"       "ATT(g,t)"     1.6406     4     7     0.31297     5.242      0         4        7  
    "ATT(g=4, t=8)"       "ATT(g,t)"     1.8837     4     8     0.27282    6.9045      0         4        8  
    "ATT(g=4, t=9)"       "ATT(g,t)"     2.2371     4     9     0.28395    7.8783      0         4        9  
    "ATT(g=4, t=10)"      "ATT(g,t)"     1.9854     4    10     0.32449    6.1187      0         4       10  
    "ATT(g=4, t=11)"      "ATT(g,t)"     1.8336     4    11     0.27363    6.7011      0         4       11  
    "ATT(g=4, t=12)"      "ATT(g,t)"     2.0198     4    12     0.31267    6.4601      0         4       12  
    "ATT(g=7, t=7)"       "ATT(g,t)"     3.1395     7     7     0.24213    12.966      0         7        7  
    "ATT(g=7, t=8)"       "ATT(g,t)"     3.5075     7     8     0.26961     13.01      0         7        8  
    "ATT(g=7, t=9)"       "ATT(g,t)"     3.3204     7     9     0.28289    11.737      0         7        9  
    "ATT(g=7, t=10)"      "ATT(g,t)"     3.3229     7    10     0.24274    13.689      0         7       10  
    "ATT(g=7, t=11)"      "ATT(g,t)"     3.3316     7    11     0.25359    13.138      0         7       11  
    "ATT(g=7, t=12)"      "ATT(g,t)"     3.4323     7    12     0.27704    12.389      0         7       12  
    "ATT(g=10, t=10)"     "ATT(g,t)"     4.9727    10    10      0.2417    20.573      0        10       10  
    "ATT(g=10, t=11)"     "ATT(g,t)"     5.0646    10    11     0.29213    17.337      0        10       11  
    "ATT(g=10, t=12)"     "ATT(g,t)"     5.0992    10    12     0.25684    19.853      0        10       12  

[CS] By Cohort
  3×8 table

    g     Estimate       SE       tStat     pValue       LB         UB      gYear
    __    ________    _______    ______    ______    _______    ______    _____

     4     2.0302     0.28186    7.2031      0       -1.0377    5.0982      4  
     7     3.3424     0.21703      15.4      0       0.98005    5.7047      7  
    10     5.0455     0.17868    28.238      0        3.1006    6.9904     10  
```

Output

```
  resCS2 = did.fit("cs",ds, ...
                 Comparison="never", Delta=0, Approach="dr",
Covariates=["x1","x2"], ...
                  Seed=42, SEMethod="multiplier");
```

```
[CS] R=18 cells; Approach=dr; Comp=never; δ=0; SE=multiplier; W=treatedObs. Bootstrap B=100, crit(95%)=2.899
[CS] Overall ATT
  1×6 table

    Estimate      SE       tStat     pValue      LB        UB  
    ________    _______    ______    ______    ______    ______

     3.022      0.14829    20.379      0       2.7195    3.3245

[CS] Estimates
  18×11 table

           Name               Effect       Estimate    g     t        SE        tStat     pValue    gYear    tYear
    _________________      __________      ________    __    __    _______      ______    ______    _____    _____

    "ATT(g=4, t=4)"        "ATT(g,t)"       2.1062      4     4    0.28186      7.4727      0         4        4  
    "ATT(g=4, t=5)"        "ATT(g,t)"       2.1191      4     5    0.31839      6.6557      0         4        5  
    "ATT(g=4, t=6)"        "ATT(g,t)"       2.4465      4     6    0.28711      8.5211      0         4        6  
    "ATT(g=4, t=7)"        "ATT(g,t)"       1.6406      4     7    0.31297       5.242      0         4        7  
    "ATT(g=4, t=8)"        "ATT(g,t)"       1.8837      4     8    0.27282      6.9045      0         4        8  
    "ATT(g=4, t=9)"        "ATT(g,t)"       2.2371      4     9    0.28395      7.8783      0         4        9  
    "ATT(g=4, t=10)"       "ATT(g,t)"       1.9854      4    10    0.32449      6.1187      0         4       10  
    "ATT(g=4, t=11)"       "ATT(g,t)"       1.8336      4    11    0.27363      6.7011      0         4       11  
    "ATT(g=4, t=12)"       "ATT(g,t)"       2.0198      4    12    0.31267      6.4601      0         4       12  
    "ATT(g=7, t=7)"        "ATT(g,t)"       3.1395      7     7    0.24213      12.966      0         7        7  
    "ATT(g=7, t=8)"        "ATT(g,t)"       3.5075      7     8    0.26961       13.01      0         7        8  
    "ATT(g=7, t=9)"        "ATT(g,t)"       3.3204      7     9    0.28289      11.737      0         7        9  
    "ATT(g=7, t=10)"       "ATT(g,t)"       3.3229      7    10    0.24274      13.689      0         7       10  
    "ATT(g=7, t=11)"       "ATT(g,t)"       3.3316      7    11    0.25359      13.138      0         7       11  
    "ATT(g=7, t=12)"       "ATT(g,t)"       3.4323      7    12    0.27704      12.389      0         7       12  
    "ATT(g=10, t=10)"      "ATT(g,t)"       4.9727     10    10     0.2417      20.573      0        10       10  
    "ATT(g=10, t=11)"      "ATT(g,t)"       5.0646     10    11    0.29213      17.337      0        10       11  
    "ATT(g=10, t=12)"      "ATT(g,t)"       5.0992     10    12    0.25684      19.853      0        10       12  

[CS] By Cohort
  3×8 table

    g     Estimate       SE       tStat     pValue       LB         UB      gYear
    __    ________    _______    ______    ______    _______    ______    _____

     4     2.0302     0.28186    7.2031      0       -1.0377    5.0982      4  
     7     3.3424     0.21703      15.4      0       0.98005    5.7047      7  
    10     5.0455     0.17868    28.238      0        3.1006    6.9904     10  
```

III.VI Interaction Weighted (IW)

Estimation Strategy

The IW estimator was suggested by Sun/Abraham (2021) and is similar to Callaway/Sant'Anna in separating essentially ATT(g,t) estimation to avoid "forbidden" comparisons between treated and untreated observations. It can be interpreted as a clean event-study aggregator built on cohort-specific DiD contrasts ($ATT(g,t)$) that respects heterogeneity and avoids TWFE pitfalls.

In the toolbox, it is identical to and builds on the CS estimation (*Approach="unconditional", Comparison="notyet", δ chosen*), but it uses a special weighting scheme for the event time aggregation. This *interaction-weighted aggregation (IW)* aggregates across cohorts at each event time using cohort weights (e.g., cohort shares among treated), producing the event-study series θ(e)\theta(e)θ(e). This is the “IW” part: one weighs cohort×event-time interactions, not the raw outcomes.

IW does not use covariates. If this is relevant, use CS with the OR, IPW or DR option.

Syntax

As the IW estimator builds on CS, options are similar, in particular the choice for clustered or bootstrapped standard errors.

resIW = did.fit("iw", T, idVar="id", timeVar="time", yVar="y", dVar="D", Comparison="notyet", Delta=0, SEMethod="multiplier", B=999, Seed=42,   Multiplier="mammen");

Output

```
resIW = did.fit("iw", ds, ...
                    Comparison="notyet", Delta=0, SEMethod="multiplier", ...
                    B=999, Seed=42, Multiplier="mammen", Weighting="cohortShare");
```

```
[IW] IW (notyet; δ=0, SE=multiplier; W=cohortShare)
[IW] K=9 event-time points.
    e     Estimate      SE         tStat      pValue      LB        UB
    _     ________    _______     ______      ______    ______    ______

    0      3.4095     0.16262     20.966        0       2.9652    3.8538
    1      3.7013     0.16574     22.332        0       3.2485    4.1541
    2      3.6106     0.16457     21.939        0        3.161    4.0602
    3      2.5501     0.20596     12.382        0       1.9874    3.1128
    4       2.713     0.19347     14.023        0       2.1845    3.2416
    5      2.9015     0.19831     14.632        0       2.3598    3.4433
    6       2.091     0.31296     6.6814        0        1.236    2.9461
    7      1.9426     0.29332     6.6227        0       1.1412     2.744
    8      2.0043     0.30243     6.6275        0       1.1781    2.8306
```

IV Usage Guidelines

Estimator	Recommended Use Case	Caveats/Conditions
Bacon Decomposition	Diagnostic: Always run first in staggered designs to assess the contamination from forbidden comparisons.	High negative weights or high variation in components signal likely TWFE bias.
TWFE	Should only be used when treatment effects are credibly assumed to be strictly homogeneous across time and groups.	Risk of non-interpretable results if effects are heterogeneous or dynamic.
DID_M	When dealing with general designs where treatment status may turn on or off (joiners and leavers). Valid if the target estimand is the average effect precisely at the time of switching.	Requires stable (never/always treated) or not-yet-treated groups for comparison. Uses Placebo test for common trends.
BJS	Default Robust Choice: Recommended for staggered event studies and dynamic effect estimation where efficiency is desired and identification is restricted to identified estimands.	Relies on Assumptions 1' and 2 (Generalized Parallel Trends and No Anticipation).

Wooldridge	When analyzing rich models requiring flexible incorporation of covariates,heterogeneous trends, or complex interaction terms, using the computational simplicity of OLS.	Requires centering covariates correctly for interpretable ATT coefficients. Allows explicit testing and relaxation of common trends.
Callaway/ Sant'Anna (CS)	Specifically suitable when covariates are relevant and potential anticipation of treatment is of relevance. The DR option (doubly robust) only requires one of IPW (inverse prob weighting) or OR (outcome regression) to be valid.	Parallel trends (unconditional or conditional on X, matching your approach). No anticipation (with Delta = δ, no effects before g − δ). Sufficient independent units/clusters for inference.
IW (Sun/ Abraham)	Builts on Callaway/Sant'Anna and introduces with interaction weighing an unbiased estimator for dynamic effects under parallel trends and no anticipation.	At the moment, no extension for covariates implemented. The base model is CS "unconditional" approach.

V References

- Baker, A. C., Larcker, D. F., & Wang, C. C. Y. (2022). How Much Should We Trust Staggered Difference-In-Differences Estimates?, Journal of Financial Economics, 144, 370–395.
- Borusyak, K., Jaravel, X., & Spiess, J. (2024). Revisiting Event-Study Designs: Robust and Efficient Estimation, Review of Economic Studies, 00, 1–33.
- Deb, P. / Norton, E. / Wooldridge, J. / Zabel, J. E. (2025) :Aggregating Average Treatment Effects on the Treated in Difference-in-Differences Models, NBER Working Paper No. 34331, October 2025.
- Callaway, B. / Sant'Anna, P. (2021). Difference-in-Differences with Multiple Time Periods. Journal of Econometrics, 225(2), 200–230.
- de Chaisemartin, C., & D'Haultfœuille, X. (2020). Two-Way Fixed Effects Estimators with Heterogeneous Treatment Effects, American Economic Review, 110(9), 2964–2996.
- Chernozhukov et al. (2018): Double/debiased machine learning for treatment and structural parameters, Econometrics Journal 21, 1-68.
- Goodman-Bacon, A. (2021). Difference-in-differences with variation in treatment timing, Journal of Econometrics, 225(2), 254–277.
- Sant’Anna, P. / Xu, Q. (2025): Difference-in-Differences with Compositional Changes, Arxiv 2304.13925.
- Sant'Anna, P. / Zhao, J. (2020): Doubly robust difference-in-differences estimators, *Journal of Econometrics*, *219*(1), 101-122.
- Sun, L. / Abraham, S. (2021): Estimating dynamic treatment effects in event studies with heterogeneous treatment effects, Journal of Econometrics 225, 175-199.
- Wooldridge, J. (2021). Two-Way Fixed Effects, the Two-Way Mundlak Regression, and Event Study Estimators. Working Paper.