On Synthetic Difference-in-Differences and Related Estimation Methods in Stata

Damian Clarke
Department of Economics
University of Chile
dclarke@fen.uchile.cl

Daniel Pailañir
Department of Economics, University of Chile
Ministry of Economics, Development and Tourism
dpailanir@fen.uchile.cl

Susan Athey
Graduate School of Business
Stanford University
athey@stanford.edu

Guido Imbens
Graduate School of Business
Stanford University
imbens@stanford.edu

Abstract. In this paper, we describe a computational implementation of the Synthetic difference-in-differences (SDID) estimator of Arkhangelsky et al. (2021) for Stata. Synthetic difference-in-differences can be used in a wide class of circumstances where treatment effects on some particular policy or event are desired, and repeated observations on treated and untreated units are available over time. We lay out the theory underlying SDID, both when there is a single treatment adoption date and when adoption is staggered over time, and discuss estimation and inference in each of these cases. We introduce the `sdid` command which implements these methods in Stata, and provide a number of examples of use, discussing estimation, inference, and visualization of results. Along with SDID, the `sdid` command allows for the implementation of standard synthetic control and difference-in-differences methods in an identical framework, permitting estimation, inference, and the generation of graphical output in a computationally efficient way.

Keywords: synthetic difference-in-differences, synthetic control, difference-in-differences, estimation, inference, visualization.

1 Introduction

There has been an explosion in recent advances in econometric methods for policy analysis. A particularly active area is that applied to estimating the impact of exposure to some particular event or policy, when observations are available in a panel or repeated cross section of groups and time (see for example recent surveys by de Chaisemartin and D'Haultfœuille (2022); Roth et al. (2022); Arkhangelsky and Imbens (2023) for reviews of these methods). A modelling challenge in this setting is in determining what would have happened to exposed units had they been left unexposed. Should such a counterfactual be estimable from underlying data, causal inference can be conducted by comparing outcomes in treated units to those in theoretical counterfactual untreated states, under the potential outcome framework (Holland 1986; Rubin 2005).

A substantial number of empirical studies in economics and the social sciences more generally seek to estimate effects in this setting using difference-in-difference (hereafter DID) style designs. Here impacts are inferred by comparing treated to control units, where time-invariant level differences between units are permitted as well as general common trends. However, the drawing of causal inferences requires a parallel trend assumption, which states that in the absence of treatment, treated units would have followed parallel paths to untreated units. Whether this assumption is reasonable in a particular context is an empirical issue. Recently, a number of methodologies have sought to loosen this assumption. These include procedures in which counterfactual trends can be assumed to deviate from parallel, leading to partial identification (Manski and Pepper 2018; Rambachan and Roth 2023), flexible procedures to adequately control for any prevailing differences between treated and control units (Bilinski and Hatfield 2018), often based on pre-treatment periods only (Bhuller et al. 2013; Goodman-Bacon 2021) and IV-style methods which explicitly consider dynamics in pre-treatment periods (Freyaldenhoven et al. 2019).

In many cases, parallel trends may be a questionable modelling assumption. One particular solution to the challenge has been the application of synthetic control (SC) methods. Early work in synthetic control explores the setting of comparative case studies, where a single treated unit is observed, and one wishes to construct a matched synthetic control from a larger number of potential donor units (Abadie and Gardeazabal 2003; Abadie et al. 2010, 2015). These methods seek to generate a single synthetic control from a unique convex weighting of underlying control units, such that this synthetic control is as closely matched as possible to the treated unit in pre-treatment outcomes, and potentially other covariates. These weights are optimally generated and fixed over time, potentially assigning zero weight to certain control units, and larger weights to others. This has attracted considerable attention in both empirical applications and theoretical extensions, with recent advances including debiasing procedures (Ben-Michael et al. 2021) which can additionally house multiple treatment units (Abadie and L'Hour 2021), more flexible weighting schemes, or constant fixed differences between treated and synthetic control units (Doudchenko and Imbens 2016; Ferman and Pinto 2021).

A recent particularly flexible modelling option which can be applied in panel data settings seeks to bridge the DID and SC procedures. Arkhangelsky et al. (2021) propose the Synthetic Difference-in-Differences (SDID) estimator, which brings in strengths from both the DID and SC methods. Like DID models, SDID allows for treated and control units to be trending on entirely different levels prior to a reform of interest. And like SC methods, SDID seeks to optimally generate a matched control unit which considerably loosens the need for parallel trend assumptions. Correspondingly, SDID avoids common pitfalls in standard DID and SC methods – namely an inability to estimate causal relationships if parallel trends are not met in aggregate data in the case of DID, and a requirement that the treated unit be housed within a "convex hull" of control units in the case of SC. Arkhangelsky et al. (2021) propose estimation and inference procedures, formally proving consistency and asymptotic normality of the proposed estimator. What's more, the authors briefly discuss a number of important applied points such as how their estimator can incorporate covariates, and how their estimator can be applied to both multiple treatment units, and even multiple treatment units which adopt treatment in different time periods.

In this paper we describe the `sdid` command (available for download as Pailañir and Clarke (2022)) which implements the SDID estimator in Stata.[1] This command allows for the simple implementation of the SDID estimator provided that a strongly balanced panel or repeated cross section of data is available covering groups and time periods. The command, written principally in Mata, seamlessly incorporates cases where there is a single treated unit, multiple treatment units, and multiple treatment periods. It reports treatment effects laid out in Arkhangelsky et al. (2021), additionally implementing their proposed bootstrap, jackknife and placebo inference procedures. A number of graphical output options are provided to examine the generation of the SDID estimator and the underlying optimal weight matrices. While principally written to conduct SDID estimation, the `sdid` command (and the SDID method) nests as possible estimation procedures SC and DID, which can be easily generated to allow comparison of estimation procedures and estimates.[2] The `sdid` command is written in a computationally efficient way, largely implemented in Mata and adopting effective optimization algorithms. For this reason, although its main use is likely in implementing SDID methods, it is potentially a useful tool for standard synthetic control estimation and inference.

Stata provides a rich environment for panel-based analysis in DID and SC settings, and it is useful to understand how `sdid` both compares to and differs from the tools currently available. From the side of synthetic control methods, a range of powerful packages exist for implementation. This includes the original `synth` package (Abadie et al. 2011), as well as `synth_runner` (Galiani and Quistorff 2017) which allows for the additional implementation of placebo tests permuting treatment assignment, `synth2` which simplifies syntax for permutation tests including placebo tests permuting treatment time (Yan and Chen 2023), and recent extensions to synthetic control methods such as `scul` (Greathouse 2022) which implements synthetic control with a LASSO selector and `allsynth` which implements bias correction methods (Wiltshire 2022). What `sdid` shares with these packages for standard synthetic control implementation is a manner to very simply implement synthetic control methods, and additionally an ability to conduct permutation inference. What `sdid` does not provide out of the box is an ability to conduct placebo inference based on permutation of treatment times, or recent modeling extensions implemented in `allsynth` and `scul`. From the side of difference-in-differences methods, many recent innovations exist including Stata's native `didregress`, `xtdidregress`, and `hdidregress` commands, and user-written commands which implement a range of heterogeneity-robust DID based estimators including `did_multiplegt` (de Chaisemartin et al. 2019), `csdid` (Rios-Avila et al.

[1] This code can be downloaded from the SSC as `ssc install sdid`. Additionally, we maintain an open source repository on Github, located at https://github.com/Daniel-Pailanir/sdid. Further examples of use can be found at this repository.

[2] Code from the original paper was provided in R (Hirshberg undated), which can do many of the procedures which `sdid` implements, and indeed, abstracting from differences in pseudo-random number generation, give identical results in cases where similar procedures are possible. A number of useful extensions are available in `sdid`, such as the implementation of estimates in cases where treatment occurs in multiple periods, and alternative manners to include covariates.

2021), `did_imputation` (Borusyak 2021), `eventstudyinteract` (Sun 2021), xtevent (Freyaldenhoven et al. 2021), `stackedev` (Bleiberg 2021) and `jwdid` (Rios-Avila 2022) as well as traditional event-study designs such as `eventdd` Clarke and Tapia-Schythe (2021). What `sdid` shares with these packages for difference-in-differences estimation is an ability to estimate parameters and conduct inference accounting for time-varying roll-out in the presence of heterogenous treatment effects in a valid way. What `sdid` does not provide and many of these routines do provide is a way to generate *a single set* of dynamic estimates and confidence intervals which are robust to time-varying events and temporally heterogeneous treatment effects. Beyond its functionality for implementing both SC and DID, `sdid`'s principal contribution is that it allows for estimation, inference and visualisation for SDID, which is not available elsewhere.

In introducing the `sdid` command in what remains of this paper, we first provide a primer on the core methodological points of SDID (as well as comparisons to DID and SC), and then describe how these procedures extend to a setting where treatment adoption occurs over multiple time periods. We then lay out the command syntax of `sdid`, as well as the elements which are returned to the user. We provide a number of examples to illustrate the use of the SDID method in Stata, both based upon a well-known example of California's passage of Proposition 99, an anti-smoking measure previously presented in Abadie et al. (2010); Arkhangelsky et al. (2021) in which a single state adopts a treatment at a given time, as well as an example where exposure to a policy occurs at mutiple periods: the case of parliamentary gender quotas studied by Bhalotra et al. (2023). We conclude by making a number of practical points on the computational implementation of this estimator, and briefly discussing potential future areas of work.

2 Methods

2.1 The Canonical Synthetic Difference-in-Differences Procedure

The synthetic DID procedure, hereafter SDID, is developed in Arkhangelsky et al. (2021), and we lay out principal details here. As input, SDID requires a balanced panel of N units or groups, observed over T time periods. An outcome, denoted Y_{it}, is observed for each unit i in each period t. Some, but not all, of these observations are treated with a specific binary variable of interest, denoted W_{it}. This treatment variable $W_{it} = 1$ if observation i is treated by time t, otherwise, $W_{it} = 0$ indicates that unit i is untreated at time t. Here, we assume that there is a single adoption period for treated units, which Arkhangelsky et al. (2021) refer to as a 'block treatment assignment'. In section 2.3, we extend this to a 'staggered adoption design' (Athey and Imbens 2022), where treated units adopt treatment at varying points. A key element of both of these designs is that once treated, units are assumed to remain exposed to treatment forever thereafter. In the particular setting of SDID, no always treated units can be included in estimation. For estimation to proceed, we require at least two pre-treatment periods off of which to determine control units. In practice, for generating good matched controls, a larger number of pre-treatment periods are required (refer to discussion in Section 5).

The goal of SDID is to consistently estimate the causal effect of receipt of policy or treatment W_{it}, (an average treatment effect on the treated, or ATT) even if we do not believe in the parallel trends assumption between all treatment and control units on average. Estimation of the ATT proceeds as follows:

$$\left(\widehat{\tau}^{sdid}, \widehat{\mu}, \widehat{\alpha}, \widehat{\beta}\right) = \underset{\tau,\mu,\alpha,\beta}{\arg\min} \left\{ \sum_{i=1}^{N} \sum_{t=1}^{T} (Y_{it} - \mu - \alpha_i - \beta_t - W_{it}\tau)^2 \widehat{\omega}_i^{sdid} \widehat{\lambda}_t^{sdid} \right\} \tag{1}$$

where the estimand is the ATT, generated from a two-way fixed effect regression, with optimally chosen weights $\widehat{\omega}_i^{sdid}$ and $\widehat{\lambda}_t^{sdid}$ discussed below. Note that here, this procedure flexibly allows for shared temporal aggregate factors given the estimation of time fixed effects β_t and time invariant unit-specific factors given the estimation of unit fixed effects α_i. As is standard in fully saturated fixed-effect models, one α_i and one β_t fixed effect are normalized to zero to avoid multi-colinearity. The presence of unit-fixed effects implies that SDID will simply seek to match treated and control units on pre-treatment trends, and *not* necessarily on both pre-treatment trends and levels, allowing for a constant difference between treatment and control units.

In this setting, it is illustrative to consider how the SDID procedure compares to the traditional synthetic control method of Abadie et al. (2010), as well as the baseline DID procedure. The standard DID procedure consists of precisely the same two-way fixed effect OLS procedure, simply assigning equal weights to all time periods and groups:

$$\left(\widehat{\tau}^{did}, \widehat{\mu}, \widehat{\alpha}, \widehat{\beta}\right) = \underset{\tau,\mu,\alpha,\beta}{\arg\min} \left\{ \sum_{i=1}^{N} \sum_{t=1}^{T} (Y_{it} - \mu - \alpha_i - \beta_t - W_{it}\tau)^2 \right\}. \tag{2}$$

The synthetic control, on the other hand, maintains optimally chosen unit-specific weights ω (as laid out below), however does not seek to optimally consider time periods via time weights, and omits unit fixed effects α_i implying that the synthetic control and treated units should maintain approximately equivalent pre-treatment levels, as well as trends.

$$\left(\widehat{\tau}^{sc}, \widehat{\mu}, \widehat{\beta}\right) = \underset{\tau,\mu,\beta}{\arg\min} \left\{ \sum_{i=1}^{N} \sum_{t=1}^{T} (Y_{it} - \mu - \beta_t - W_{it}\tau)^2 \widehat{\omega}_i^{sc} \right\} \tag{3}$$

From (2)-(3) it is clear that the SDID procedure offers greater flexibility than both the DID and SC procedures; in the case of DID by permitting a violation of parallel trends in aggregate data, and in the case of SC, by both additionally seeking to optimally weight time periods when considering counterfactual outcomes, and allowing for level differences between treatment and control groups.

The selection of unit weights, ω, as inputs to (1) (and (3)) seeks to ensure that comparison is made between treated units and controls which were approximately following parallel trends prior to the adoption of treatments. The selection of time weights, λ in the case of SDID seeks to draw more weight from pre-treatment periods which are more similar to post-treatment periods, in the sense of finding a constant difference between each control unit's post treatment average, and pre-treatment weighted averages across all selected controls. Specifically, as laid out in Arkhangelsky et al. (2021), unit-specific weights are found by resolving:

$$\left(\widehat{\omega}_0, \widehat{\omega}^{sdid}\right) = \underset{\omega_0 \in \mathbb{R}, \omega \in \Omega}{\arg\min} \sum_{t=1}^{T_{pre}} \left(\omega_0 + \sum_{i=1}^{N_{co}} \omega_i Y_{it} - \frac{1}{N_{tr}} \sum_{i=N_{co}+1}^{N} Y_{it} \right)^2 + \zeta^2 T_{pre} ||\omega||_2^2 \tag{4}$$

$$\text{where } \Omega = \left\{ \omega \in \mathbb{R}_+^N, \text{ with } \sum_{i=1}^{N_{co}} \omega_i = 1 \text{ and } \omega_i = \frac{1}{N_{tr}} \text{ for all } i = N_{co}+1, \ldots, N \right\},$$

$||\omega||_2$ refers to the Euclidean norm and ζ is a regularization parameter laid out in Arkhangelsky et al. (2021, pp. 4091-4092). For the sake of completion, this regularization parameter is calculated as $\zeta = (N_{tr} \times T_{post})^{1/4} \widehat{\sigma}$, where:

$$\widehat{\sigma}^2 = \frac{1}{N_{co}(T_{pre}-1)} \sum_{i=1}^{N_{co}} \sum_{t=1}^{T_{pre}-1} (\Delta_{it} - \bar{\Delta})^2, \ \Delta_{it} = Y_{i,(t+1)} - Y_{it}, \tag{5}$$

$$\text{and } \bar{\Delta} = \frac{1}{N_{co}(T_{pre}-1)} \sum_{i=1}^{N_{co}} \sum_{t=1}^{T_{pre}-1} \Delta_{it}.$$

This regularization parameter is shown to be theoretically motivated and have good performance in (Arkhangelsky et al. 2021), and as such is adopted by default in `sdid`, but options allow for this parameter to be varied.

This procedure leads to a vector of N_{co} non-negative weights plus an intercept ω_0. The weights ω_i for all $i \in \{1, \ldots, N_{co}\}$ imply that absolute difference between control and treatment trends units should be minimized over all pre-treatment periods, while ω_0 initially allows for a constant difference between treatment and controls over time. Together, these imply that units will follow parallel pre-trends, though provided $\omega_0 \neq 0$, not *identical* pre-trends.

In the case of time weights, a similar procedure is followed, finding weights which minimize the following objective function:

$$\left(\widehat{\lambda}_0, \widehat{\lambda}^{sdid}\right) = \underset{\lambda_0 \in \mathbb{R}, \lambda \in \Lambda}{\arg\min} \sum_{i=1}^{N_{co}} \left(\lambda_0 + \sum_{t=1}^{T_{pre}} \lambda_t Y_{it} - \frac{1}{T_{post}} \sum_{t=T_{pre}+1}^{T} Y_{it} \right)^2 + \zeta^2 N_{co} ||\lambda||^2 \tag{6}$$

$$\text{where } \Lambda = \left\{ \lambda \in \mathbb{R}_+^T, \text{ with } \sum_{t=1}^{T_{pre}} \lambda_t = 1 \text{ and } \lambda_t = \frac{1}{T_{post}} \text{ for all } t = T_{pre}+1, \ldots, T \right\},$$

where the final term in (6) is a very small regularization term to ensure uniqueness of time weights, where $\zeta = 1 \times 10^{-6} \widehat{\sigma}$, and $\widehat{\sigma}$ is defined as in (5). Again, this parameter is adopted by default in computational implementations, but can be adjusted.

This estimation procedure is summarized in Arkhangelsky et al. (2021, algorithm 1), reproduced in Appendix A1 here for ease of access. Arkhangelsky et al. (2021) also prove that the estimator is asymptotically normal, suggesting that confidence intervals on τ can be constructed as:

$$\widehat{\tau}^{sdid} \pm z_{\alpha/2}\sqrt{\widehat{V}_\tau},$$

where $z_{\alpha/2}$ refers to the inverse normal density function at percentile $\alpha/2$ should one wish to compute 1-α confidence intervals. These confidence intervals thus simply require an estimate of the variance of τ, $\widehat{V}_\tau$. Arkhangelsky et al. (2021) propose three specific procedures to estimate this variance: a block bootstrap, a jackknife, or a permutation-based approach.

The block (also known as clustered) bootstrap approach, consists of taking some large number, B, of bootstrap resamples over units, where units i are the resampled blocks in the block bootstrap procedure. Provided that a given resample does not consist entirely of treated, or entirely of control units, the quantity $\widehat{\tau}^{sdid}$ is re-estimated, and denoted as $\widehat{\tau}^{sdid}_{(b)}$ for each bootstrap resample. The bootstrap variance $\widehat{V}^{(b)}_\tau$ is then calculated as the variance of resampled estimates $\widehat{\tau}^{sdid}_{(b)}$ across all B resamples. The bootstrap algorithm is defined in Arkhangelsky et al. (2021, algorithm 2), reproduced here in Appendix A1. This bootstrap procedure is observed in simulation to have particularly good properties, but has two particular drawbacks, justifying alternative inference procedures. The first is that it may be computationally costly, given that in each bootstrap resample the entire synthetic DID procedure is re-estimated, including the estimation of optimal weights. This is especially computationally expensive in cases where working with large samples, or where covariates are included, as discussed at more length below. The second, is that formal proofs of asymptotic normality rely on the number of treated units being large, and as such, estimated variance, and confidence intervals, may be unreliable when the number of treated units is small.

An alternative estimator which significantly reduces the computational burden inherent in the bootstrap is estimating a jackknife variance for $\widehat{\tau}^{sdid}$. This procedure consists of iterating over all units in the data, in each iteration removing the given unit, and recalculating $\widehat{\tau}^{sdid}$, denoted $\widehat{\tau}^{sdid}_{(-i)}$, *maintaining fixed* the optimal weights for ω and λ calculated in the original SDID estimate. The jackknife variance, $\widehat{V}^{(jack)}_\tau$ is then calculated based on the variance of all $\widehat{\tau}^{sdid}_{(-i)}$ estimates, following Arkhangelsky et al. (2021, Algorithm 3) (refer to Appendix A1 here). In this case, each iteration saves on recalculating optimal weights, and as documented by Arkhangelsky et al. (2021), provide a variance leading to conservative confidence intervals, without the computational burden imposed by the bootstrap. Once again, asymptotic normality relies on there being a large number of treated units, and in particular if only 1 treated unit is observed – as is often the case in comparative case studies – the jackknife will not even be defined given that a $\widehat{\tau}^{sdid}_{(-i)}$ term will be undefined when removing the single treated unit.

Given limits to these inference options when the number of treated units is small, an alternative placebo, or permutation-based, inference procedure is proposed. This consists of, first, conserving just the control units, and then randomly assigning the same treatment structure to these control units, as a placebo treatment. Based on this placebo treatment, we then re-estimate $\widehat{\tau}^{sdid}$, denoted $\widehat{\tau}^{sdid}_{(p)}$. This procedure is repeated many times, giving rise to a vector of estimates $\widehat{\tau}^{sdid}_{(p)}$, and the placebo variance, $\widehat{V}^{(placebo)}_{\tau}$, can be estimated as the variance of this vector. This is formally defined in Arkhangelsky et al. (2021, algorithm 4), and Appendix A1 here. It is important to note that in the case of this placebo-based variance, homoskedasticity across units is required, given that the variance is based off placebo assignments of treatment made only within the control group.

2.2 Conditioning on Covariates

So far, we have limited exposition to cases where one wishes to study outcomes Y_{it}, and their evolution in treated and synthetic control units. However, in certain settings, it may be of relevance to condition on exogenous time-varying covariates X_{it}. Arkhangelsky et al. (2021) note that in this case, we can proceed by applying the SDID algorithm to the residuals calculated as:

$$Y^{res}_{it} = Y_{it} - X_{it}\widehat{\beta}, \tag{7}$$

where $\widehat{\beta}$ comes from regression of Y_{it} on X_{it}. This procedure, in which the SDID process will be applied to the residuals Y^{res}_{it}, is different to the logic of synthetic controls following Abadie et al. (2010). In Abadie et al.'s conception, when covariates are included the synthetic control is chosen to ensure that these covariates are as closely matched as possible between treated and synthetic control units. However in the SDID conception, covariate adjustment is viewed as a pre-processing task, which removes the impact of changes in covariates from the outcome Y_{it} prior to calculating the synthetic control. Along with their paper, Arkhangelsky et al. (2021) provide an implementation of their algorithm in R (Hirshberg undated), and in practice they condition out these variables X_{it} by finding $\widehat{\beta}$ within an optimization procedure which additionally allows for the efficient calculation of optimal weights ω and λ. In the `sdid` code described below, we follow Hirshberg (undated) in implementing this efficient optimization procedure (the Frank and Wolfe (1956) solver), however there are a number of potential complications which can arise in this manner of dealing with covariates, and as such, alternative procedures are also available.

A first potential issue is purely numerical. In the Frank-Wolfe solver discussed above, a minimum point is assumed to be found when successive iterations of the solver lead to arbitrarily small changes in all parameters estimated.[3] Where these parameters include coefficients on covariates, in extreme cases the solution found for (1) can be sensitive to the scaling of covariates. This occurs in particular when covariates have very large magnitudes and variances. In such cases, the inclusion of covariates in (1) can cause optimization routines to suggest solutions which are not actually globally optimal, given that successive movements in $\widehat{\beta}$ can be very small. In extreme cases, this can imply that when multiplying all variables X_{it} by a large constant values, the estimated treatment effect can vary. While this issue can be addressed by using smaller tolerances for defining

[3] In the case of convex functions such as that in (1), the Frank-Wolfe solver finds a global minima, see for example Lawphongpanich (2009).

stopping rules in the optimization procedure, it can be addressed in a more simple way if all covariates are first re-standardized as Z-scores, implying that no very-high-variance variables are included, while capturing the same underlying variation in covariates.

A second, potentially more complicated point is described by Kranz (2022). He notes that in certain settings, specifically where the relationship between covariates and the outcome vary over time differentially in treatment and control groups, the procedure described above may fail to capture the true relationship between covariates and the outcome of interest, and may subsequently lead to bias in estimated treatment effects. He proposes a slightly altered methodology of controlling for covariates. Namely, his suggestion is to first estimate a two-way fixed effect regression of the dependent variable on covariates (plus time and unit fixed effects), using *only* observations which are not exposed, or not exposed yet, to treatment, i.e. sub-setting to observations for which $W_{it} = 0$. Based on the regression $Y_{it} = X_{it}\beta + \mu_i + \lambda_t + \varepsilon_{it}$ for units with $W_{it} = 0$, coefficients $\widehat{\beta}$ estimated by OLS can be used to follow the procedure in (7), and SDID can then be conducted. An additional benefit of this method is that unlike the optimized method described previously, $\widehat{\beta}$ is calculated in a single step via OLS rather than an interative optimization procedure, which often leads to substantial speed ups in computation time.

We document these methods based on a number of empirical example in the following section. It is worth noting that regardless of which procedure one uses, the implementation of SDID follows the suggestion laid out in Arkhangelsky et al. (2021, footnote 4) and (7) above. What varies between the former and the latter procedures discussed here is the manner with which one estimates coefficients on covariates, $\widehat{\beta}$, in a pre-processing step.

2.3 The Staggered Adoption Design

The design discussed up to this point assumes block assignment, or that all units are either controls, or are treated in a single unit of time. However, in Arkhangelsky et al. (2021, Appendix A), they note that this procedure can be extended to a staggered adoption design, where treated units adopt treatment at varying moments of time. Here we lay out the formal details related to the staggered adoption design, focusing first on estimation of an aggregate treatment effect, and then on extending the three inference procedures laid out previously into a staggered adoption setting. This proposal is one potential way to deal with staggered adoption settings, though there are other possible manners to proceed – see for example Ben-Michael et al. (2021), or Arkhangelsky et al. (2021, Appendix A). In cases where the number of pure control units is small, this proposal may not necessarily be very effective given challenges in finding appropriate counterfactuals for each adoption-specific period.

Estimation Unlike the block assignment case where a single pre- versus post-treatment date can be used to conduct estimation, in the staggered adoption design, multiple adoption dates are observed. Consider for example the treatment matrix below, consisting of 8 units, 2 of which (1 and 2) are untreated, while the other 6 are treated, however at varying points.

$$\mathbf{W} = \left(\begin{array}{ccccccccc} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 5 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 6 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 7 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 8 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

This single staggered treatment matrix $\mathbf{W}$ can be broken down into adoption date specific matrices, $\mathbf{W}^1$, $\mathbf{W}^2$ and $\mathbf{W}^3$, or generically, $\mathbf{W}^1, \ldots, \mathbf{W}^A$, where A indicates the number of distinct adoption dates. Additionally, a row vector $\mathbf{A}$ consisting of A elements contains these distinct adoption periods. In this specific setting where units first adopt treatment in period t_3 (units 7 and 8), t_4 (units 5 and 6), and t_7 (units 3 and 4), the adoption date vector consists simple of periods 3, 4 and 7.

$$\mathbf{A} = \begin{pmatrix} 3 & 4 & 7 \end{pmatrix}$$

Finally, adoption-specific matrices $\mathbf{W}^1$-$\mathbf{W}^3$ simply consist of pure treated units, and units which adopt in this specific period, as below:

$$\mathbf{W}^1 = \left(\begin{array}{ccccccccc} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right),$$

$$\mathbf{W}^2 = \left(\begin{array}{ccccccccc} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 6 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right),$$

$$\mathbf{W}^3 = \left(\begin{array}{ccccccccc} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 8 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right).$$

As laid out in Arkhangelsky et al. (2021, Appendix A), the average treatment effect on the treated can then be calculated by applying the synthetic DID estimator to each of these 3 adoption-specific samples, and calculating a weighted average of the three estimators, where weights are assigned based on the relative number of treated units and time periods in each adoption group. Generically, this ATT is calculated based on adoption-specific SDID estimates as:

$$\widehat{ATT} = \sum_{\text{for } a \in \mathbf{A}} \frac{T^a_{post}}{T_{post}} \times \hat{\tau}^{sdid}_a \tag{8}$$

where T_{post} refers to the total number of post-treatment periods observed in treated units. This estimation procedures is laid out formally in Algorithm 1 below.

Note that in this case, while the parameter interest is likely the treatment effect ATT or adoption specific τ^{sdid}_a parameters, each adoption period is associated with an optimal unit and time weight vector ω^{sdid}_a and λ^{sdid}_a, which can be returned following estimation.

```
Algorithm 1: Estimation of the ATT with staggered adoption
  Data: $\mathbf{Y}$, $\mathbf{W}$, $\mathbf{A}$.
  Result: Point estimate $\widehat{ATT}$ and adoption-specific values $\hat{\tau}^{sdid}_a$, $\hat{\omega}^{sdid}_a$ and $\hat{\lambda}^{sdid}_a$
   for all a $\in$ $\mathbf{A}$.

  for a $\in$ A do
      1. Subset Y and W to units who are pure controls, and who first adopt
       treatment in period $t = a$. ;
      2. Compute regularization parameter $\zeta$ ;
      3. Compute unit weights $\hat{\omega}^{sdid}_a$ ;
      4. Compute time weights $\hat{\lambda}^{sdid}_a$ ;
      5. Compute the SDID estimator via the weighted DID regression ;

      $\left(\hat{\tau}^{sdid}_a, \hat{\mu}_a, \hat{\alpha}_a, \hat{\beta}_a\right) = \underset{\tau,\mu,\alpha,\beta}{\arg\min} \left\{ \sum_{i=1}^{N} \sum_{t=1}^{T} (Y_{it} - \mu - \alpha_i - \beta_t - W_{it}\tau)^2 \widehat{\omega}^{sdid}_{a,i} \hat{\lambda}^{sdid}_{a,t} \right\}$
  end

  6. Compute ATT across adoption-specific SDID estimates

      $\widehat{ATT} = \sum_{\text{for } a \in \mathbf{A}} \frac{T^a_{post}}{T_{post}} \times \hat{\tau}^{sdid}_a$
```

Inference In the staggered adoption design, estimated treatment effects are simply a multi-period extension of the underlying SDID algorithm, in each case working with the relevant pure control and treated sub-sample. Thus, inference can be conducted in the staggered adoption design under similar resample or placebo procedures. Here we discuss inference following each of the bootstrap, jackknife, or placebo procedures laid out in Arkhangelsky et al. (2021), applied to a staggered adoption setting. We note that in this design, it is likely the case that one wishes to conduct inference on the treatment effect, ATT from (8). Thus in the below, we propose inference details for this estimand, additionally noting that standard errors and confidence intervals on adoption-specific SDID parameters τ_a^{sdid} come built-in as part of these procedures.

Consider first the case of bootstrap inference. Suppose that one wishes to estimate standard errors or generate confidence intervals on the global treatment effect ATT. A bootstrap procedure can be conducted based on many clustered bootstrap resamples over the entire initial dataset, where in each case, a resampled ATT estimate $\widehat{ATT}^b$ is generated, following Algorithm 1. Based on many such resampled estimates, the bootstrap variance can be calculated as the variance of these resamples. We lay out the bootstrap variance estimate below in Algorithm 2. Note that as in the block treatment design, this bootstrap procedure requires the number of treated units to grow with N within each adoption period. As such, if a very small number of treated units exist for certain adoption periods, placebo inference is likely preferable. Similarly, as laid out in the block treatment design, the bootstrap procedure re-estimates optimal weight matrices in each resample, and can be computationally expensive in cases where samples are large.

```
Algorithm 2: Bootstrap inference in the staggered adoption setting
  Data: $\mathbf{Y}$, $\mathbf{W}$, $\mathbf{A}$, $B$.
  Result: Variance estimator $\widehat{V}^{cb}_{ATT}$. Additionally, variance for each adoption
    specific estimate $\widehat{V}^{cb}_{\tau_a}$ for all $a \in \mathbf{A}$.

  for $b \leftarrow 1$ to $B$ do
      1. Construct a bootstrap dataset $(\mathbf{Y}^{(b)}, \mathbf{W}^{(b)}, \mathbf{A}^{(b)})$ by sampling $N$ rows of
         $(\mathbf{Y}, \mathbf{W})$ with replacement, and generating $\mathbf{A}$ as the corresponding adoption
         vector ;
      2. if the bootstrap sample has no treated units or no control units then
             Discard resample and go to 1 ;
         end
      3. Compute SDID estimate $ATT^{(b)}$ following Algorithm 1 based on
         $(\mathbf{Y}^{(b)}, \mathbf{W}^{(b)}, \mathbf{A}^{(b)})$. Similarly, generate a vector of adoption-date specific
         resampled SDID estimates $\tau_a^{(b)}$ for all $a \in \mathbf{A}^{(b)}$ ;
  end

  4. Define $\widehat{V}^{cb}_{ATT} = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{ATT^{(b)}} - \frac{1}{B}\sum_{b=1}^{B}\widehat{ATT^{(b)}}\right)^2$. Similarly, estimate
    adoption-date specific variances for each $\tau_a^{sdid}$ estimate as the variance over
    each $\tau_a^{(b)}$;
```

An alternative inference procedure, which is less computationally intensive but similarly based on asymptotic arguments with a large number of states, and many treated units, is based on the jackknife. Here, optimal weight matrices calculated for each adoption-specific estimate τ_a^{sdid} in Algorithm 1 are treated as fixed, and provided as inputs to a jackknife procedure, described below in Algorithm 3. Below, these matrices, which consist of weights for each adoption period $a \in \mathbf{A}$, are denoted as $\boldsymbol{\omega}$, $\boldsymbol{\lambda}$.[4] Note that in Algorithm 3, notation $(-i)$ refers to a standard jackknife estimator, removing a single state (i) in each iteration. In cases where i refers to a treated unit, the ATT will be calculated removing this particular treated unit. For this reason, the jackknife estimator will not be defined in cases where any single adoption period has only one treated unit, as in this case, $\widehat{\tau}_a^{(-i)}$ will not be defined.

Algorithm 3: Jackknife inference in the staggered adoption setting

```
Inputs: $\mathbf{Y}$, $\mathbf{W}$, $\mathbf{A}$, $\widehat{\boldsymbol{\omega}}$, $\widehat{\boldsymbol{\lambda}}$, $\widehat{ATT}$.
Result: Variance estimator $\widehat{V}_{ATT}$

for $i \leftarrow 1$ to $N$ do
    1. Compute SDID estimate $ATT^{(-i)}$ following Algorithm 1 based on
       $(\mathbf{Y}^{(-i)}, \mathbf{W}^{(-i)}, \mathbf{A})$. Similarly, generate a vector of adoption-date specific
       resampled SDID estimates $\tau_a^{(-i)}$ for all $a \in \mathbf{A}$;
end

2. Compute $\widehat{V}_{ATT}^{jack} = (N-1)N^{-1}\sum_{i=1}^{N}\left(\widehat{ATT}^{(-i)} - \widehat{ATT}\right)^2$;
```

Finally, in cases where the number of treated units is small, and concerns related to the validity of the previous variance estimators exists, the placebo inference procedure defined in algorithm 4 can be used. Here, this is defined for the staggered adoption case, generalising Algorithm 4 of Arkhangelsky et al. (2021). To conduct this procedure, placebo treatments are randomly assigned based on the actual treatment structure, however *only* to the control units. Based on these placebo assignments, placebo values for ATT are generated, which can be used to calculate the variance as laid out in Algorithm 4. It is important to note that such a procedure will only be feasible in cases where the number of control units is strictly larger than the number of treated units (or hence placebo assignments will not be feasible), and, as laid out in Arkhangelsky et al. (2021); Conley and Taber (2011), such a procedure relies on homoskedasticity across units, as otherwise the variance of the treatment effect on the treated could not be inferred from variation in assignment of placebo treatments to control units.

[4] In the case of $\boldsymbol{\omega}$, this is an $A \times N_{co}$ matrix, where for each row a, the weight assigned to each control unit in this particular adoption period is recorded. In the case of $\boldsymbol{\lambda}$, this is a matrix containing A columns, where each column consists of as many rows as the number of pre-treatment periods to this specific adoption date. In each cell, the weight assigned to a particular pre-treatment year for each adoption period is recorded.

Algorithm 4: Placebo inference in the staggered adoption

```
Inputs: $\mathbf{Y}_{co}$, $N_{tr}$, $B$.
Result: Variance estimator $\widehat{V}_{ATT}^{placebo}$

for $b \leftarrow$ 1 to $B$ do
    1. Sample $N_{tr}$ out of the $N_{co}$ control units without replacment to ‘receive
       the placebo’ ;
    2. Construct a placebo treatment matrix $\mathbf{W}_{co}^{(b)}$, for the controls ;
    3. Compute SDID estimate $ATT^{(b)}$ following algorithm 1 based on
       $(\mathbf{Y}_{co}, \mathbf{W}_{co}^{(b)}, \mathbf{A}^{(b)})$ ;
end

4. Define $\widehat{V}_{ATT}^{placebo} = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{ATT^{(b)}} - \frac{1}{B}\sum_{b=1}^{B}\widehat{ATT^{(b)}}\right)^2$ ;
```

3 The sdid command

Synthetic Difference-in-Differences requires for data to be in Stata’s long format, organised with a single line per group and time period. If data is in wide rather than long format, it should be reshaped using Stata’s `reshape` command prior to implementing `sdid`. Once data is in long format Synthetic Difference-in-Differences can be implemented in Stata using the following command syntax:

`sdid` *depvar groupvar timevar treatment* [*if*] [*in*] `, vce(`*type*`)` [
`covariates(`*varlist, type*`)` `seed(`#`)` `reps(`#`)` `method(`*type*`)` `zeta_lambda(`#`)`
`zeta_omega(`#`)` `min_dec(`#`)` `max_iter(`#`)` `level(`#`)` `graph` `g1on`
`g1_opt(`*string*`)` `g2_opt(`*string*`)` `graph_export(`*string, type*`)`
`msize(`*markersizestyle*`)` `unstandardized` `mattitles` `verbose` `returnweights`
`generate(`*string*`)`]

where *depvar* describes the dependent variable in a balanced panel of units (*groupvar*) and time periods (*timevar*). The variable which indicates units which are treated at each time period, which accumulates over time, is indicated as *treatment*. Note that here, it is not necessary for users to specify whether the design is a block or staggered adoption design, as this will be inferred based off the data structure. Optionally, *if* and *in* can be specified, provided that this does not result in imbalance in the panel. Required and permitted options are discussed below, followed by a description of objects returned by `sdid`.

Options

`vce(`*type*`)` is a required option. This must be one of bootstrap, jackknife, placebo or noinference, where in each case inference proceeds following the specified method. In the case of bootstrap, this is only permitted if greater than one unit is treated. In the case of jackknife, this is only permitted if greater than one unit is treated in each treatment period (if multiple treatment periods are considered). In the case of placebo, this requires at least one more control than treated unit to allow for permutations to be constructed. In each case, inference follows the specific algorithm laid out in Arkhangelsky et al. (2021). We allow the no inference option (noinference) should one wish to simply generate the point estimator. This is useful if you wish to plot outcome trends without the added computational time associated with inference procedures.

`covariates(`*varlist, type*`)` Covariates should be included as a varlist, and if specified, treatment and control units will be adjusted based on covariates in the synthetic difference-in-differences procedure. Optionally, type may be specified, which indicates how covariate adjustment will occur. If the type is indicated as "optimized" (the default) this will follow the method described in Arkhangelsky et al. (2021), footnote 4, where SDID is applied to the residuals of all units after regression adjustment. However, this has been observed to be problematic at times (refer to Kranz (2022)), and is also sensitive to optimization if covariates have high dispersion. Thus, an alternative type is implemented ("projected"), which consists of conducting regression adjustment based on parameters estimated only in untreated units. This type follows the procedure proposed by Kranz (2022) (xsynth in R), and is observed to be more stable in some implementations (and at times, considerably faster). `sdid` will run simple checks on the covariates indicated and return an error if covariates are constant, to avoid multicolineality. However, prior to running sdid, you are encouraged to ensure that covariates are not perfectly multicolinear with other covariates and state and year fixed effects, in a simple two-way fixed effect regression. If perfectly multicolinear covariates are included sdid will execute without errors, however where type is "optimized", the procedure may be sensitive to the inclusion of redundant covariates.

`seed(`#`)` Define the seed for pseudo-random numbers.

`reps(`#`)` Set the number of repetitions used in the calculation of bootstrap and placebo standard errors. Default is 50 repetitions. Larger values should be preferred where possible.

`method(`*type*`)` this option allows you to change the estimation method. The type must be one of sdid, did or sc, where sdid refers to synthetic difference-in-differences, sc refers to synthetic control, and did refers to difference-in-differences. By default, sdid is enabled.

`zeta_omega(`#`)` Value used when defining the regularization term for unit weight calculations. This value is the quantity prior to the $\widehat{\sigma}$ term used to calculate ζ in (4)). Default is $(N_{tr}T_{post})^{1/4}$ as discussed in Section 2 when `method(sdid)` is implemented. For other methods, default value is 1e-6.

`zeta_lambda(`*#*`)` Value used when defining the regularization term for time weight calculations. This value is the scalar prior to the $\widehat{\sigma}$ term used to calculate ζ in (6)). Default is 1e-6 as discussed in Section 2. This is only relevant when `method(sdid)` is used, as otherwise time weights are not used.

`min_dec(`*#*`)` Estimation of optimal weights occurs iteratively until a sequential stopping rule is met. By default, a minimum is assumed when consecutive iterations move by no more than the value indicated in `min_dec`. By default, this value is set at 1e-5.

`max_iter(`*#*`)` Defines the maximum number of iterations to be performed when calculating optimal weights. By default, a maximum of 10,000 iterations will be performed. Larger values can be set to ensure that a minimum is reached.

`graph` if this option is specified, graphs will be displayed showing unit and time weights as well as outcome trends as per Figure 1 from Arkhangelsky et al. (2021). If `graph` is indicated, graphs will be produced and displayed on screen for all versions of Stata except Stata(console); if additionally graphs should be saved to disk, the `graph_export` option should be used.

`g1on` this option activates the unit-specific weight graph. By default g1 is off to avoid slow processing in cases where many groups exist.

`g1_opt(`*string*`)` option to modify the appearance of the unit-specific weight graph. These options adjust the underlying scatter plot, so should be consistent with twoway scatter plots.

`g2_opt(`*string*`)` option to modify the appearance of the outcome trend graphs. These options adjust the underlying line plot, so should be consistent with twoway line plots.

`graph_export(`*string, type*`)` Graphs will be saved as weightsYYYY and trendsYYYY for each of the unit-specific weights and outcome trends respectively, where YYYY refers to each treatment adoption period. Two graphs will be generated for each treatment adoption period provided that `g1on` is specified, otherwise a single graph will be generated for each adoption period. If this option is specified, type must be specified, which refers to a valid Stata graph type (eg ".eps", ".pdf", or any other options permitted by `graph_export`.). Additionally, if `type` is specified as ".gph" the graph is saved on disk in Stata's ".gph" format which permits editing of the graph. Optionally, a stub can be specified, in which case this will be prepended to exported graph names.

`level(`*#*`)` specifies the confidence level, as a percentage, for confidence intervals. The default is the level set by `set level` (which by default is `level(95)`).

`msize(`*markersizestyle*`)` allows you to modify the size of the marker for graph 1.

`unstandardized` if controls are included and the "optimized" method is specified, controls will be standardized as Z-scores prior to finding optimal weights. This avoids problems with optimization when control variables have very high dispersion. If unstandardized is specified, controls will simply be entered in their original units. This option should be used with care.

`mattitles` requests labels to be added to the returned `e(omega)` weight matrix providing names (in string) for the unit variables which generate the synthetic control group in each case. If mattitles is not indicated, the returned weight matrix (`e(omega)`) will store these weights with a final column providing the numerical ID of units, where this numerical ID is either taken from the unit variable (if this variable is a numerical format), or arranged in alphabetical order based on the unit variable, if this variable is in string format.

`verbose` Requests additional output, such as warnings messages if the number of iterations indicated in `max_iter` is reached.

`returnweights` Indicates that estimated weights $\widehat{\omega}$ and $\widehat{\lambda}$ should be returned directly in the dataset corresponding to each unit. By default, these will be returned as variables named `omegaYYYY` and `lambdaYYYY` where `YYYY` is replaced by treatment adoption periods.

`generate(`*string*`)` Specifies that the variables containing omega and lambda weights returned if the `returnweights` option is indicated should be named starting with *string*. If `returnweights` is indicated but `generate` is not indicated, variables will simply follow default naming.

Returned Objects

`sdid` stores the following in `e()`:

Scalars:

`e(ATT)`	Average Treatment Effect on the Treated
`e(ATT_l)`	Left-hand point of confidence interval on ATT (based on `level()`)
`e(ATT_r)`	Right-hand point of confidence interval on ATT (based on `level()`)
`e(se)`	Standard error for the ATT
`e(reps)`	Number of bootstrap/placebo replications
`e(N_clust)`	Number of units (groups) observed in the original panel used for `sdid`

Macros:

`e(cmd)`	Returns command name (sdid)
`e(cmdline)`	Returns command as typed
`e(depvar)`	Lists name of dependent variable
`e(vce)`	Provide vcetype specified in `vce()` (placebo, bootstrap, jackknife or noinference)
`e(clustvar)`	Provides the name of the unit (group) variable

Matrices:

`e(tau)` tau estimate for each adoption time-period, along with its standard error

`e(lambda)` lambda weights (time-specific weights)

`e(omega)` omega weights (unit-specific weights)

`e(adoption)` A vector containing the list of all treatment adoption times

`e(beta)` A vector corresponding to coefficients estimated on covariates included as controls (only returned if the `covariates` options is used)

`e(series)` control and treatment series containing time series trends of outcome means over time

`e(difference)` Difference between treatment and control series over time

`e(b)` Coefficient estimate returned for ATT

`e(V)` Variance estimate returned for ATT

The matrices `e(b)` and `e(V)` are included to facilite the exportation of results from `sdid` with routines such as `estout`.

4 Examples based on an Empirical Application

In the sections below we provide a number of illustrations of the usage of, and performance of, the `sdid` command, which operationalizes the Synthetic Difference-in-Differences estimator in Stata. We consider both a block treatment design (with a single adopting state), and a staggered adoption design, noting a number of points covering estimation, inference, and visualization.

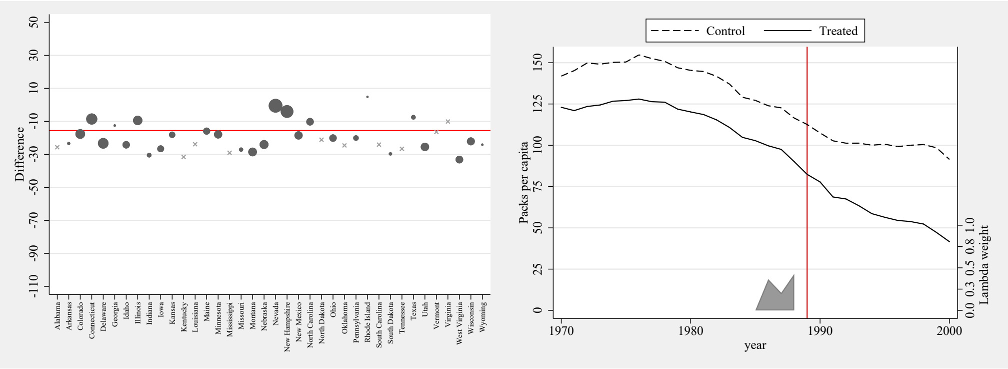
4.1 A Block Design

In the first case, we consider the well-known example, also presented in Arkhangelsky et al. (2021), of California's "Proposition 99" tobacco control measure. This example, based on the context described in Abadie et al. (2010) and data of Orzechowski and Walker (2005), is frequently used to illustrate synthetic control style methods. Proposition 99, which was passed by California in 1989, increased the taxes paid on a packet of cigarettes by 25 cents. The impact of this reform is sought to be estimated by comparing the evolution of sales of cigarettes in packs per capita in California (the treated state) with those in 38 untreated states, which did not significantly increase cigarette taxes during the study period.

The data used in analysis cover each of these 39 states over the period of 1970–2000, with a single observation for each state and year. Adoption occurs in California in 1989, implying $T_{pre} = 19$ pre-treatment periods and $T_{post} = 12$ post-treatment periods. There are $N_{co} = 38$ control and a single treated state, hence $N_{tr} = 1$. Using the `sdid` command, we replicate the results from Arkhangelsky et al. (2021). In the below code example, we first download the data, and then conduct the Synthetic Difference-in-Differences implementation using a placebo inference procedure with (a default) 50 placebo iterations.

```
. webuse set www.damianclarke.net/stata/

. webuse prop99_example.dta, clear

. sdid packspercapita state year treated, vce(placebo) seed(1213)
Placebo replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................     50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
packsperca~a |     ATT      Std. Err.     t      P>|t|     [95% Conf. Interval]
-------------+---------------------------------------------------------------
     treated | -15.60383    9.87941    -1.58    0.114    -34.96712     3.75946
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

The third line of this code excerpt quite simply implements the synthetic difference-in-differences estimator, returning identical point estimates to those documented in Table 1 of Arkhangelsky et al. (2021). Standard errors are slightly different, as these are based on pseudo-random placebo reshuffling, though can be replicated as presented here provided that the same seed is set in the `seed` option. Note that in this case, given that a small number (1) of treated units is present, placebo inference is the only appropriate procedure, as indicated in the `vce()` option.



(a) Unit-Specific Weights

(b) Outcome Trends and Time-Specific Weights

Figure 1: Proposition 99, example from Abadie et al. (2010); Arkhangelsky et al. (2021)

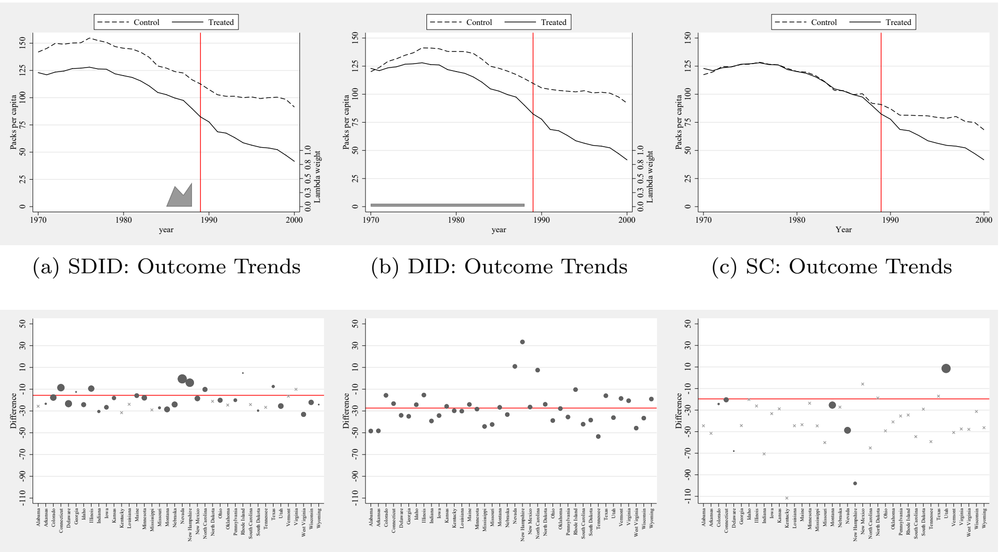
Should we wish to generate the same graphs as in Arkhangelsky et al. (2021), summarizing both (a) unit specific weights, and (b) treatment and synthetic control outcome trends along with time specific weights, this can be requested with the addition of the `graph` option. This is displayed below, where we additionally modify plot aesthetics via the `g1_opt()` and `g2_opt()` options for weight graphs (Figure 1(a)), and trend graphs (Figure 1(b)) respectively. Finally, generated graphs can be saved to disk using the `graph_export()` option, with a graph type (`.eps` below), and optionally a pre-pended plot name. Output corresponding to the below command is provided in Figure 1.

```
. sdid packspercapita state year treated, vce(placebo) seed(1213) graph g1on
>     g2_opt(ylabel(0(25)150) ytitle("Packs per capita") scheme(sj))
>     g1_opt(xtitle("") scheme(sj)) graph_export(sdid_, .eps)
```

It is illustrative to compare the output of SDID estimation procedures with those of standard synthetic control methods of Abadie et al. (2010), and unweighted difference-in-difference estimates. By using the `method()` option one can request a standard difference-in-differences output, requested with `method(did)`, or synthetic control output, requested with `method(sc)`. In the interests of completeness, `method(sdid)` is also accepted, although this is the default behaviour when `method` is not included in command syntax. In each case, resulting graphs document matched treated and control/synthetic control trends, as well as weights received by each unit and time period. These are displayed in Figure 2, with plots corresponding to each of the three calls to `sdid` displayed below. In the left-hand panel, identical SDID plots are provided as those noted above. In the middle plot, corresponding to `method(did)`, a difference-in-difference setting is displayed. Here, in the top panel, outcomes for California are displayed as a solid line, while mean outcomes for all control states are documented as a dashed line, where a clear divergence is observed in the pre-treatment period. The bottom panel shows that in this case, each control unit receives an identical weight, while time weights indicated at the base of the top plot note that each period is weighted identically. Finally, in the case of synthetic control, output from the third call to `sdid` is provided in the right-hand panel. In this case, treated and synthetic control units are observed to overlap nearly exactly, with weights in figure (f) noted to be more sparse, and placing relatively more weight on fewer control states. We note that in each case, the `vce(noinference)` option is used, as here we are simply interested in observing exported graphs, not the entire command output displaying aggregate estimates, standard errors and confidence intervals.

```
sdid packspercapita state year treated, method(sdid) vce(noinference) graph
>     g1_opt(ylabel(-110(20)50) xtitle("") scheme(sj)) g1on
>     g2_opt(ylabel(0(25)150) ytitle("Packs per capita") scheme(sj))
>     graph_export(sdid_, .eps)

sdid packspercapita state year treated, method(did) vce(noinference) graph msize(small)
>     g1_opt(ylabel(-110(20)50) xtitle("") scheme(sj)) g1on
>     g2_opt(ylabel(0(25)150) ytitle("Packs per capita") scheme(sj))
>     graph_export(did_, .eps)

sdid packspercapita state year treated, method(sc) vce(noinference) graph
>     g1_opt(ylabel(-110(20)50) xtitle("") scheme(sj)) g1on
>     g2_opt(ylabel(0(25)150) ytitle("Packs per capita") scheme(sj))
>     graph_export(sc_, .eps)
```

The `sdid` command returns multiple matrices containing treatment and control outcome trends, weights, and other elements. These elements can be accessed simply for use in post-estimation procedures or graphing. As a simple example, the following code excerpt accesses treatment and synthetic control outcome trends (stored in



(a) SDID: Outcome Trends (b) DID: Outcome Trends (c) SC: Outcome Trends

(d) SDID: Unit-Specific Weights (e) DID: Unit-Specific Weights (f) SC: Unit-Specific Weights

Figure 2: Comparison of estimators

`e(series)`, and time weights (stored in `e(lambda)`) and uses these elements to replicate the plot presented in Figure 1b. The resulting graphical output is presented as Figure 3. Figure 3 is virtually identical to Figure 1b however omits the second y-axis, in line with the plots originally presented in Arkhangelsky et al. (2021). In this way, if one wishes to have further control over the precise nature of plotting, beyond that provided in the graphing options available in `sdid`'s command syntax, one can simply work with elements returned in the ereturn list. In Appendix A2, we show that with slightly more effort, returned elements can be used to construct the unit-specific weight plot from Figure 1a.

```
. preserve
. clear
. matrix series=e(series)
. matrix lambda=e(lambda)
. qui svmat series, names(col)
. qui svmat lambda

. tw line Yco1989 year, yaxis(1) ||
>    line Ytr1989 year, yaxis(1) ||
>    area lambda1 year if year<=1988, yaxis(2) ylabel(0(1)5, axis(2)) yscale(off axis(2))
>    xline(1989, lc(red)) legend(order(1 "Control" 2 "Treated") pos(12) col(2)) scheme(sj)
. graph export sdid_replicate.eps, replace
. restore
```

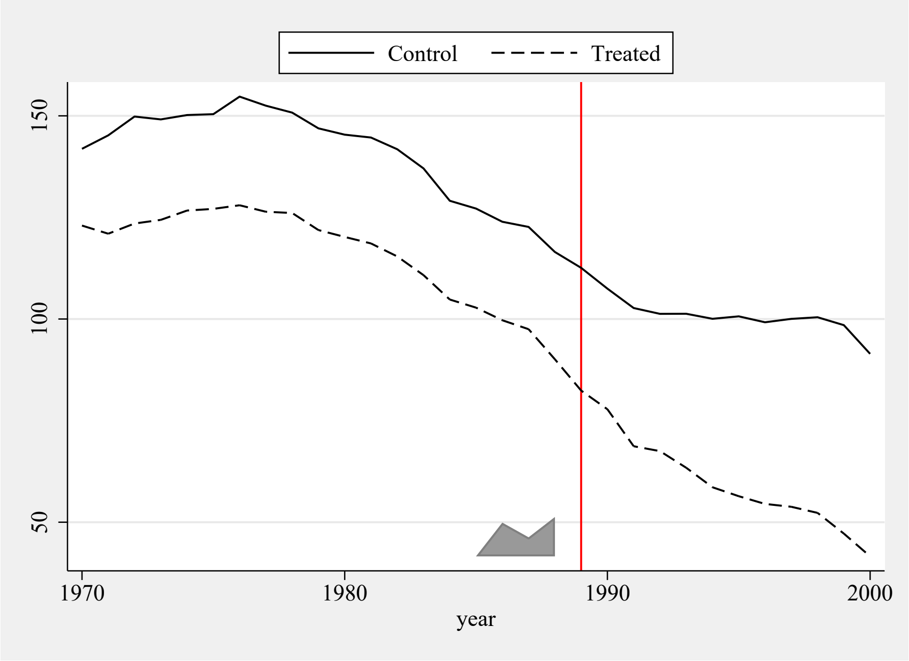


Figure 3: Outcome Trends and Time-Specific Weights

4.2 A Staggered Adoption Design

We present an example of a staggered adoption design, based on data and the context studied in Bhalotra et al. (2023). In this case, the impact of parliamentary gender quotas which reserve seats for women in parliament are estimated, first on rates of women in parliament, and secondly on rates of maternal mortality. This is conducted on a country by year panel, where for each of 1990-2015, 115 countries are observed, 9 of which implement a parliamentary gender quota.[5] For each of these countries, data on the rates of women in parliament and the maternal mortality ratio are collected, as well as a number of covariates.

This example presents a staggered adoption configuration, given that in the period under study, quota adoption occurred in seven different yearly periods between 2000 and 2013. `sdid` handles a staggered adoption configuration seamlessly without any particular changes to the syntax. In the code below, we implement the synthetic difference-in-differences estimator using the bootstrap procedure to calculate standard errors. The output by default reports the weighted ATT which is defined in (8) above. However, as laid out in (8), this is based on each adoption-period specific synthetic difference-in-differences estimate. These adoption-period specific estimates

[5] This is a subset of the full sample studied in Bhalotra et al. (2023). Here we only work with countries for which observations of women in parliament and maternal mortality exist for the full time period, without missing observations.

are returned in the matrix `e(tau)`, which is tabulated below the standard command output.

```
. webuse quota_example.dta, clear

. sdid womparl country year quota, vce(bootstrap) seed(1213)
Bootstrap replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................     50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
     womparl |     ATT     Std. Err.     t      P>|t|    [95% Conf. Interval]
-------------+---------------------------------------------------------------
       quota |   8.03410    3.74040     2.15    0.032     0.70305    15.36516
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.

. matlist e(tau)

             |       Tau       Time
-------------+----------------------
          r1 |  8.388868       2000
          r2 |  6.967746       2002
          r3 |  13.95226       2003
          r4 | -3.450543       2005
          r5 |  2.749036       2010
          r6 |  21.76272       2012
          r7 | -.8203235       2013
```

All other elements are identical to those documented in the case of a single adoption period, however generalised to multiple adoptions. For example, if requesting graphical output, a single treatment versus synthetic control trend graph and corresponding unit-level weight graph is provided for each adoption date. Similarly, `ereturn`ed matrices such as `e(lambda)`, `e(omega)` and `e(series)` provide columns for each particular adoption period.

Adding Covariates As laid out in Section 2.2, covariates can be handled in synthetic difference-in-differences in a number of ways. Below we document the inclusion of a single covariate (the natural logarithm of GDP per capita). As `sdid` is based on a balanced panel of observations, we must first ensure that there are no missing observations for all covariates, in this case dropping a small number of (control) countries for which this measure is not available. We then include covariates via the `covariates()` option. In the first case, this is conducted exactly following the procedure discussed by Arkhangelsky et al. (2021), in which parameters on covariates are estimated within the optimization routines in Mata. This is analogous to indicating `covariates(, optimized)`. Estimates in this particular case suggest that the inclusion of this control does little to dampen effects. After estimation, the coefficients on the covariates can be inspected as part of `e(beta)`, where an adoption-specific value for each covariate is provided, given that the underlying SDID estimate is calculated for each adoption period.

```
. drop if lngdp==.
. sdid womparl country year quota, vce(bootstrap) seed(1213) covariates(lngdp)
Bootstrap replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................     50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
     womparl |     ATT     Std. Err.      t      P>|t|     [95% Conf. Interval]
-------------+---------------------------------------------------------------
       quota |   8.05150    3.09252      2.60    0.009      1.99027    14.11272
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

The inclusion of covariates in the previous implementation adds considerably to the computational time as it increases the complexity of the underlying optimization routine, and this is conducted in each adoption period and each bootstrap replicate. An alternative manner to capture covariates described in section 2.2 above is that of Kranz (2022), where the impact of covariates are projected out using a baseline regression of the outcome on covariates and fixed effects only in units where the treatment status is equal to zero. This is implemented as below, with `covariates(, projected)`.

```
. sdid womparl country year quota, vce(bootstrap) seed(1213) covariates(lngdp, projected)
Bootstrap replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................     50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
     womparl |     ATT     Std. Err.      t      P>|t|     [95% Conf. Interval]
-------------+---------------------------------------------------------------
       quota |   8.05927    3.11913      2.58    0.010      1.94589    14.17264
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

Here, results are slightly different, though quantitatively comparable to those when using alternative procedures for conditioning out covariates. In this case, if examining the `e(beta)` matrix, only a single coefficient will be provided, as the regression used to estimate the coefficient vector is always based on the same sample. This additionally offers a non-trivial speed up in the execution of the code. For example, on a particular personal computer with Stata SE 15.1 and relatively standard specifications, using the `optimized` method above requires 324 seconds of computational time while using `projected` requires 61 seconds (compared with 58 seconds where covariates are not included in `sdid`).

Post-Estimation Commands While `sdid` provides a standard tabular and graphical output as displayed previously, the command can used to provide output in alternative formats. For example, the `sdid` command interacts seamlessly with routines such as `estout` (Jann 2004) for the exportation of results tables. To see this, the below block of code estimates three specific versions of the model discussed above, storing each model using an `eststo:` prefix, before finally exporting estimated ATTs and standard errors to a LaTeX file, which can be included in tabular form as displayed in Table 1. Similar such procedures could be conducted with routines such as `outreg` or `outreg2`, and tabular output could be further enriched using additional options within `esttab` if desired.

```
. webuse set www.damianclarke.net/stata/
. webuse quota_example.dta, clear
. lab var quota "Parliamentary Gender Quota"

. eststo sdid_1: sdid womparl country year quota, vce(bootstrap) seed(2022)

. drop if lngdp==.
. eststo sdid_2: sdid womparl country year quota, vce(bootstrap) seed(2022)
>     covariates(lngdp, optimized)

. eststo sdid_3: sdid womparl country year quota, vce(bootstrap) seed(2022)
>     covariates(lngdp, projected)

. esttab sdid_1 sdid_2 sdid_3 using "example1.tex",
>     nonotes nomtitles stats(N, labels("Observations") fmt(%9.0fc))
>     addnotes("* p$<$0.10, ** p$<$0.05, *** p$<$0.01")
>     starlevel ("*" 0.10 "**" 0.05 "***" 0.01) lab
>     b(%-9.3f) se(%-9.3f) style(tex) replace
```

	(1)	(2)	(3)
Parliamentary Gender Quota	8.034**	8.051***	8.059***
	(3.940)	(3.047)	(3.099)
Observations	3,094	2,990	2,990

* p<0.10, ** p<0.05, *** p<0.01

Table 1: Tabular Output Following `sdid`

4.3 Inference Options

In this section we provide examples of the implementation of alternative inference options, as laid out in algorithms 2-4. For this illustration we will keep only treated units which adopt gender quotas in 2002 and 2003, as otherwise adoption periods will exist in which only a single unit is treated, and jackknife procedures will not be feasible.

```
. webuse quota_example.dta, clear
. drop if country=="Algeria"   | country=="Kenya" | country=="Samoa" |
>         country=="Swaziland" | country=="Tanzania"
```

In the following three code blocks we document bootstrap, placebo and jackknife inference procedures. The difference in implementation in each case is very minor, simply indicating either `bootstrap`, `placebo` or `jaccknife` in the `vce()` option. For example, in the case of bootstrap inference, where block bootstraps over the variable `country` are performed, the syntax is as follows:

```
. sdid womparl country year quota, vce(bootstrap) seed(1213)
Bootstrap replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
     womparl |     ATT     Std. Err.     t      P>|t|     [95% Conf. Interval]
-------------+---------------------------------------------------------------
       quota |  10.33066    4.72911     2.18    0.029     1.06178    19.59954
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

By default, only 50 bootstrap replicates are performed, though in practice, a substantially higher number should be used, and this can be indicated in the **`reps(#)`** option. In the case of placebo, the syntax and output are virtually identical. The suitability of each method depends on the underlying structure of the panel, and in this particular case, given the relatively small number of treated units, it may be the case that placebo procedures are preferred.

```
. sdid womparl country year quota, vce(placebo) seed(1213)
Placebo replications (50). This may take some time.
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
     womparl |     ATT     Std. Err.     t      P>|t|     [95% Conf. Interval]
-------------+---------------------------------------------------------------
       quota |  10.33066   2.34040     4.41    0.000     5.74356    14.91776
-----------------------------------------------------------------------------
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

Finally, in the interests of completeness, the jackknife procedure, which is by far the fastest of the three to execute[6], is provided below. Note that unlike the case with placebo or bootstrap inference, it is not necessary (or relevant) to set a seed, nor indicate the number of replications, as the jackknife procedure implies conducting a leave-one-out procedure over each unit. In this particular case, jackknife inference appears to be more conservative than bootstrap procedures, in line with what may be

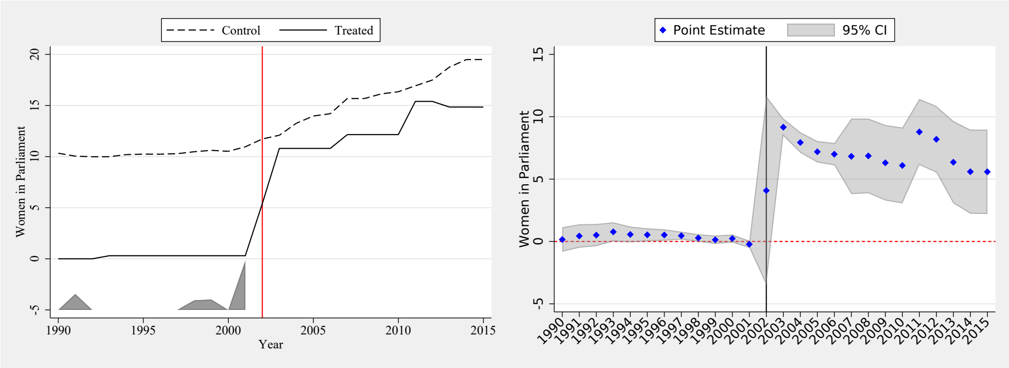
[6] As an example, with 50 replicates for bootstrap and placebo, and on a standard personal computer running Stata SE, 15.1, the execution time for bootstrap was 18.2 seconds, for placebo permutations was 10.09 seconds, and for jackknife was 0.7 seconds. This time scales approximately linearly with the number of replicates in the case of bootstrap and placebo. With 500 replicates the time was 178.1 and 101.6 for bootstrap and placebo procedures respectively.

expected based on Arkhangelsky et al. (2021)'s demonstration that jackknife inference is in general conservative.

```
. sdid womparl country year quota, vce(jackknife)

Synthetic Difference-in-Differences Estimator

─────────────┬──────────────────────────────────────────────────────────────
     womparl │      ATT     Std. Err.       t      P>|t|    [95% Conf. Interval]
─────────────┼──────────────────────────────────────────────────────────────
       quota │  10.33066    6.00560      1.72    0.085    -1.44009    22.10141
─────────────┴──────────────────────────────────────────────────────────────
95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.
```

4.4 Event Study Style Output

While `sdid` offers a simple implementation to conduct standard synthetic difference-in-difference procedures and provide output, with some work results can also be visualized in alternative ways. For example, consider the standard 'panel event-study' style setting (see *e.g.* Freyaldenhoven et al. (2019); Schmidheiny and Siegloch (2019); Clarke and Tapia-Schythe (2021)), where one wishes to visualize how the dynamics of some treatment effect evolve over time, as well as how differences between treated and control units evolve prior to the adoption of treatment. Such graphs are frequently used to efficiently provide information on both the credibility of parallel pre-trends in an observational setting, as well as the emergence of any impact owing to treatment once treatment is switched on.



(a) Outcome Trends (b) Event Study

Figure 4: Outcome trends and event study style estimate of the impact of quotas on % women in parliament

What such an analysis seeks to document is the differential evolution of treated and (synthetic) control units, abstracting away from any baseline difference between the groups. As an example, refer to Figure 4(a), which is based on the adoption of gender quotas laid out in section 4.2, and in particular quota adoption year 2002. This is standard output from `sdid`, presenting trends in rates of women in parliament in countries which adopted quotas in 2002 (solid line), and synthetic control countries which did not adopt quotas (dashed line). We will refer to the values plotted in these trend lines as $\bar{Y}_t^{Tr}$ for treated units in year t, and $\bar{Y}_t^{Co}$ for synthetic control units in year t. While this standard output allows us to visualize trends in the two groups in a simple way, it is not immediately clear how the *differences* in these outcomes evolve over time compared to baseline differences, nor the confidence intervals on any such changes over time.

For this to resemble the logic of an event study analysis, we wish to consider, for each period t, whether differences between treated units and synthetic controls have changed when compared to baseline differences. Namely, for each period t, we wish to calculate:

$$\left(\bar{Y}_t^{Tr} - \bar{Y}_t^{Co}\right) - \left(\bar{Y}_{baseline}^{Tr} - \bar{Y}_{baseline}^{Co}\right), \tag{9}$$

along with the confidence interval for this quantity. Here $\bar{Y}_{baseline}^{Tr}$ and $\bar{Y}_{baseline}^{Co}$ refer to baseline (pre-treatment) means for treated and synthetic control units respectively. In standard panel event studies, some arbitrary baseline period is chosen off of which to estimate pre-treatment differences. This is often one year prior to treatment. In the case of SDID where pre-treatment weights are optimally chosen as $\widehat{\lambda}_t^{sdid}$ (refer to section 2), this suggests an alternative quantity for $\bar{Y}_{baseline}^{Tr}$ and $\bar{Y}_{baseline}^{Co}$, namely:

$$\bar{Y}_{baseline}^{Tr} = \sum_{t=1}^{T_{pre}} \widehat{\lambda}_t^{sdid} \bar{Y}_t^{Tr} \qquad \bar{Y}_{baseline}^{Co} = \sum_{t=1}^{T_{pre}} \widehat{\lambda}_t^{sdid} \bar{Y}_t^{Co}. \tag{10}$$

In words these baseline outcomes are simply pre-treatment aggregates, where weights are determined by optimal pre-treatment weights (indicated by the shaded gray area in Figure 4(a)). The event study then plots the quantities defined in (9), for each time t.

An example of such an event study style plot is presented in Figure 4(b). Here, blue points present the quantity indicated in (9) for each year. In this case, t ranges from 1990 to 2015. While all these points are based off a simple implementation of `sdid` comparing outcomes between treated and control units following (9), confidence intervals documented in gray shaded areas of Figure 4(b) can be generated following the resampling or permutation procedures discussed earlier in this paper. Specifically, in the case of re-sampling, a block bootstrap can be conducted, and in each iteration the quantity in (9) can be re-calculated for each t. The confidence interval associated with each of these quantities can then be calculated based on its variance across many (block)-bootstrap resamples.

Figure 4(b), and graphs following this principle more generally, can be generated following the use of `sdid`. However, by default `sdid` simply provides output on trends among the treated and synthetic control units (as displayed in Figure 4(a)). In the code below, we lay out how one can move from these trends to the event study in panel (b). As this procedure requires conducting the inference portion of the plot manually (unlike most other procedures involving sdid where inference is conducted automatically as part of the command) the code is somewhat more involved. For this reason, we discuss the code below in a number of blocks, terminating with the generation of the plot displayed in Figure 4(b).

In a first code block, we will open the parliamentary gender quota data which we used in section 4.2, and keep the particular adoption period considered here (countries which adopt quotas in 2002), as well as un-treated units:

```
. webuse set www.damianclarke.net/stata/
. webuse quota_example.dta, clear

. egen m=min(year) if quota==1, by(country)  //indicator for the year of adoption
. egen mm=mean(m), by(country)
. keep if mm==2002 | mm==.                   //keep only one time of adoption
. drop if lngdp==.                           //keep if covariates observed
```

We can then implement the standard SDID procedure, additionally exporting the trend graphs which is displayed in Figure 4(a). This is done in the first line below, after which a number of vectors are stored. These vectors allow us to calculate the quantity $(\bar{Y}^{Tr}_{baseline} - \bar{Y}^{Co}_{baseline})$ indicated in (9), which is generated from $\widehat{\lambda}^{sdid}$, from the returned matrix e(lambda), and pre-treatment values for $\bar{Y}^{Tr}_t$ and $\bar{Y}^{Co}_t$, from the returned matrix e(series). This baseline quantity is referred to as meanpre_o below. Finally, the quantity of interest in (9) for each time period t is generated as the variable d, which is plotted below as the blue points on the event study in Figure 4(b).

```
. qui sdid womparl country year quota, vce(noinference) graph g2_opt(ylab(-5(5)20)
>     ytitle("Women in Parliament") scheme(sj)) graph_export(groups, .pdf)
>     covariates(lngdp, projected)

. matrix lambda = e(lambda)[1..12,1]    //save lambda weight
. matrix yco = e(series)[1..12,2]       //control baseline
. matrix ytr = e(series)[1..12,3]       //treated baseline
. matrix aux = lambda´*(ytr - yco)      //calculate the pre-treatment mean
. scalar meanpre_o = aux[1,1]

. matrix difference = e(difference)[1..26,1..2]  // Store Ytr-Yco
. svmat difference
. ren (difference1 difference2) (time d)
. replace d = d - meanpre_o                      // Calculate vector in (8)
```

Perhaps the most complicated portion of code is that which implements the bootstrap procedure. This is provided below, where for ease of replication we consider a relatively small number of bootstrap resamples, which are set as the local B = 100. In each bootstrap resample, we first ensure that both treatment and control units are present (using the locals r1 and r2), and then re-estimate the sdid procedure with the new bootstrap sample generated using Stata's bsample command. This is precisely the same block bootstrap procedure laid out by Arkhangelsky et al. (2021), and which sdid conducts internally, however here we are interested in collecting, for each bootstrap resample, the same quantity estimated above with the main sample as d, which captures the estimate defined in (9) for each t. To do so, we simply follow an identical procedure as that conducted above, however now save the resulting resampled values of the quantities from (9) as a series of matrices d‘b’ for later processing to generate confidence intervals in the event study.

```
. local b = 1
. local B = 100
. while `b´<=`B´ {
.     preserve
.     bsample, cluster(country) idcluster(c2)
.     qui count if quota == 0
.     local r1 = r(N)
.     qui count if quota != 0
.     local r2 = r(N)
.     if (`r1´!=0 & `r2´!=0) {
.         qui sdid womparl c2 year quota, vce(noinference) graph covariates(lngdp, projected)

.         matrix lambda_b = e(lambda)[1..12,1]     //save lambda weight
.         matrix yco_b = e(series)[1..12,2]        //control baseline
.         matrix ytr_b = e(series)[1..12,3]        //treated baseline
.         matrix aux_b = lambda_b´*(ytr_b - yco_b) //calculate the pre-treatment mean
.         matrix meanpre_b = J(26,1,aux_b[1,1])

.         matrix d`b´ = e(difference)[1..26,2] - meanpre_b

.         local ++b
.     }
.     restore
. }
```

The final step is to calculate the standard deviation of each estimate from (9) based on the bootstrap resamples, and then to generate confidence intervals for each parameter based on the estimates generated above (d), as well as their standard errors. This is conducted in the first lines of the code below. For each of the $B = 100$ resamples conducted above, we import the vector of resampled estimates from (9), and then using rowsd() calculate the standard deviation of the estimates across each time period t. This is the bootstrap standard error, which is used below to calculate the upper and lower bounds of 95% confidence intervals as [LCI;UCI]. Finally, based on these generated elements (d, as blue points on the event study, and LCI, UCI as the end points of confidence intervals) we generate the output for Figure 4(b) in the final lines of code.

```
. preserve
. keep time d
. keep if time!=.

. forval b=1/`B´ {
.     svmat d`b´   // import each bootstrap replicate of difference between trends
. }

. egen rsd = rowsd(d11 - d`B´1)      //calculate standard deviation of this difference
. gen LCI = d + invnormal(0.025)*rsd //lower bounds on bootstrap CIs
. gen UCI = d + invnormal(0.975)*rsd //upper bounds on bootstrap CIs

. *generate plot
. tw rarea UCI LCI time, color(gray%40) || scatter d time, color(blue) m(d)
>     xtitle("") ytitle("Women in Parliament") xlab(1990(1)2015, angle(45))
>     legend(order(2 "Point Estimate" 1 "95% CI") pos(12) col(2))
>     xline(2002, lc(black) lp(solid)) yline(0, lc(red) lp(shortdash))
>     scheme(sj)
. graph export "event_sdid.pdf", replace
. restore
```

As noted above, the outcome of this graph is provided in Figure 4(b), where we observe that, as expected, the synthetic difference-in-difference algorithm has resulted in quite closely matched trends between the synthetic control and treatment group in the pre-treatment period, given that all pre-treatment estimates lie close to zero. The observed impact of quotas on women in parliament occurs from the treatment year onward, where these differences are observed to be large and statistically significant.

This process of estimating an event study style plot is conducted here for a specific adoption year. In the case of a block adoption design where there is only one adoption period, this will be the only resulting event study to consider. However in a staggered adoption design, a single event study could be generated for each adoption period. Potentially, such event studies could be combined, but some way would be required to deal with unbalanced lags and leads, and additionally some weighting function would be required to group treatment lags and leads where multiple such lags and leads are available. One such procedure has been proposed in Sun and Abraham (2021), and could be a way forward here.

4.5 Other Parameters

In general, estimation and inference for synthetic difference-in-differences is concerned with the ATT, or adoption-specific treatment effects $\widehat{\tau}_a^{sdid}$ from (8) in the case of staggered adoption designs. Nevertheless, in considering the weighted regression model laid out in (1), it is clear that there are a range of other parameters involved in the estimation of the ATT including a constant term μ, unit fixed effects α_i, and time fixed effects β_t.

In practice, once unit and time-specific weights are calculated there are a number of ways which one can proceed to estimate the ATT. In the implementation of `sdid`, given that the main estimand of interest is the ATT, a direct approach is taken where $\widehat{\tau}^{sdid}$ (or $\widehat{\tau}_a^{sdid}$ in the case of staggered adoption designs) can be estimated directly in a single weighted difference-in-differences step. This procedure is laid out fully in Appendix A3, and the `sdid` code works in this way given that it is substantially faster in terms of total computation time than implementing the weighted regression described in (1). Nevertheless, if one wishes to estimate the *other* parameters associated with the objective function (1), this parameter estimation can be conducted as a post-estimation step in a relatively straightforward way.

To see this, note that (1) is simply a weighted regression where weights for each unit are defined as $w = \widehat{\omega}_i^{sdid} \cdot \widehat{\lambda}_t^{sdid}$. Given that `sdid` returns $\widehat{\omega}_i^{sdid}$ and $\widehat{\lambda}_t^{sdid}$ as `e(omega)` and `e(lambda)`, and these can additionally be requested to be stored directly in a data via the `returnweights` option, it is illustrative to see that the weighted regression returns the parameters described in (1). We illustrate this below, taking advantage of the `returnweights` option in `sdid`.

To see how this works, we return to the case of proposition 99, first estimating synthetic difference-in-differences. By default when `returnweights` is indicated, λ weights are returned as `lambda----` (where `----` refers to the treatment period; 1989 below), and ω weights are returned as `omega----`. If multiple treatment periods exist, a set of weights are returned for each period. In the code block below, we generate unit-specific weights implicit in `sdid` as `weight`, before finally estimating the weighted regression. Output from this regression is presented below, where we observe the equivalence with the treatment effect estimated by `sdid`. *However* we additionally observe estimated state-specific fixed effects and year-specific fixed effects, as well as the constant indicated in (1). It is additionally of note that fixed effects *only* exist for those units which receive non-zero weights in the `sdid` algorithm. When considering state-specific fixed effects, a single fixed effect is omitted by default (in this case, Arkansas), while all other fixed effects apart from those receiving zero weights (indicated by an “x” symbol in Figure 2d) are presented in regression output. Similarly in the case of year fixed-effects, estimates exist for pre-treatment years 1987 and 1988, while 1986 is omitted as the baseline reference category, in line with non-zero ω weights for these periods only.

```
. webuse set www.damianclarke.net/stata/
(prefix now "http://www.damianclarke.net/stata")

. webuse prop99_example.dta, clear

. sdid packspercapita state year treated, vce(noinference) returnweights

Synthetic Difference-in-Differences Estimator

-----------------------------------------------------------------------------
packsperca~a |     ATT     Std. Err.     t      P>|t|    [95% Conf. Interval]
-------------+---------------------------------------------------------------
     treated | -15.60383          .        .        .            .           .
-----------------------------------------------------------------------------

95% CIs and p-values are based on Large-Sample approximations.
Refer to Arkhangelsky et al., (2020) for theoretical derivations.

. gen weight = lambda*omega

. encode state, gen(st)

. reg packspercapita treated i.st i.year [aw=weight]
(sum of wgt is 4.000000031635864)

      Source |       SS           df       MS      Number of obs   =       435
-------------+----------------------------------   F(43, 391)      =    261.27
       Model |  403548.457        43  9384.84784   Prob > F        =    0.0000
    Residual |  14044.9445       391  35.9205742   R-squared       =    0.9664
-------------+----------------------------------   Adj R-squared   =    0.9627
       Total |  417593.402       434  962.196778   Root MSE        =    5.9934

---------------------------------------------------------------------------------
 packspercapita |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
----------------+----------------------------------------------------------------
        treated |  -15.60383   1.149442   -13.58   0.000    -17.86369   -13.34397
                |
             st |
    California |  -32.06146   6.962705    -4.60   0.000    -45.75048   -18.37244
      Colorado |  -24.36931   7.131295    -3.42   0.001    -38.38979   -10.34883
   Connecticut |  -23.70395   7.077667    -3.35   0.001      -37.619   -9.788907
      Delaware |   16.48742   7.094419     2.32   0.021     2.539443     30.4354
       Georgia |  -1.780844   12.32708    -0.14   0.885     -26.0165    22.45481
         Idaho |  -31.77529   7.296028    -4.36   0.000    -46.11965   -17.43094
      Illinois |  -18.41872   7.146835    -2.58   0.010    -32.46975   -4.367685
       Indiana |   15.53584   8.017371     1.94   0.053    -.2267099    31.29839
          Iowa |  -19.89129   7.372302    -2.70   0.007     -34.3856   -5.396975
        Kansas |  -20.07296   7.458412    -2.69   0.007    -34.73657   -5.409355
         Maine |  -4.256553   7.337436    -0.58   0.562    -18.68232    10.16921
     Minnesota |  -25.18321   7.222564    -3.49   0.001    -39.38313   -10.98329
      Missouri |   6.716917   8.315669     0.81   0.420    -9.632101    23.06594
       Montana |  -28.77168   7.186296    -4.00   0.000     -42.9003   -14.64307
      Nebraska |   -24.6252   7.171822    -3.43   0.001    -38.72536   -10.52505
        Nevada |   8.607337   7.022139     1.23   0.221    -5.198538    22.41321
 New Hampshire |   56.12127     7.0396     7.97   0.000     42.28106    69.96147
    New Mexico |  -43.26666   7.214899    -6.00   0.000    -57.45151   -29.08181
North Carolina |   20.11322   7.281355     2.76   0.006     5.797714    34.42873
          Ohio |  -.6654243   7.296113    -0.09   0.927    -15.00994     13.6791
  Pennsylvania |  -14.43475   7.664293    -1.88   0.060    -29.50313    .6336355
  Rhode Island |  -2.872521   14.37156    -0.20   0.842    -31.12772    25.38268
  South Dakota |  -25.59506   9.401702    -2.72   0.007    -44.07927   -7.110842
         Texas |  -26.52882   8.054488    -3.29   0.001    -42.36435    -10.6933
          Utah |  -61.21125   7.208445    -8.49   0.000    -75.38341   -47.03909
 West Virginia |  -8.568432   7.273484    -1.18   0.239    -22.86846    5.731599
     Wisconsin |  -19.62645   7.244853    -2.71   0.007    -33.87019   -5.382704
       Wyoming |  -5.602418   12.92684    -0.43   0.665    -31.01722    19.81239
               |
           year |
          1987 |  -1.735867   1.118305    -1.55   0.121    -3.934509    .4627757
          1988 |  -8.517289   .9150735    -9.31   0.000    -10.31637   -6.718209
          1989 |  -6.517374   1.662163    -3.92   0.000    -9.785269    -3.24948
          1990 |  -11.37696   1.662163    -6.84   0.000    -14.64485   -8.109061
          1991 |  -18.31309   1.662163   -11.02   0.000    -21.58099    -15.0452
          1992 |   -19.6303   1.662163   -11.81   0.000     -22.8982   -16.36241
          1993 |  -21.66713   1.662163   -13.04   0.000    -24.93502   -18.39923
          1994 |  -24.68675   1.662163   -14.85   0.000    -27.95465   -21.41886
          1995 |  -25.48698   1.662163   -15.33   0.000    -28.75487   -22.21908
          1996 |  -27.16479   1.662163   -16.34   0.000    -30.43269    -23.8969
          1997 |  -27.09811   1.662163   -16.30   0.000      -30.366   -23.83021
          1998 |  -27.64908   1.662163   -16.63   0.000    -30.91697   -24.38118
          1999 |   -31.1674   1.662163   -18.75   0.000    -34.43529    -27.8995
          2000 |  -37.49742   1.662163   -22.56   0.000    -40.76532   -34.22953
               |
          _cons |   131.2032   6.959487    18.85   0.000     117.5205    144.8859
----------------------------------------------------------------------------------

. drop lambda omega weight
```

It is important to note that this regression *does not* give valid inference for the estimated ATT, nor for any fixed effects or constants as it treats weights as known, rather than estimated. More generally, the variance derivations in Arkhangelsky et al. (2021) are only valid for the ATT, so any inference considerations on fixed effects are not theoretically founded at present. Nevertheless, if we do wish to consider valid inference in a regression-based framework following `sdid`, we can follow similar procedures to those laid out in bootstrap, jackknife or placebo procedures described previously. To consider a simple illustration, below we implement a permutation-based procedure which is virtually identical to `sdid`'s internal permutation routine, where in each iteration an ATT estimate is produced for a randomly permuted (untreated) state assuming treatment occurs in 1989. We once again estimate weighted regression using the weights returned from `sdid`, in each iteration storing the estimated ATT in the variable `Taus`. Finally, at the end of this code block we inspect estimated `Taus`, which provides an estimate of the standard error (the standard deviation of permutations), and similarly would allow for the calculation of confidence intervals based on empirical quantiles of this variable. We observe that the standard error is in line with that reported previously when estimating `sdid` directly.

```
. //Taus will be used to store permutation estimates
. gen Taus = .
(1,209 missing values generated)

. local B=50

. set seed 1213

. forvalues b=1/`B´ {
  2.     preserve
  3.     qui {
  4.         //Drop treatment state
.        drop if state=="California"
  5.         //Permute treatment to one untreated state using highest value of random draw
.        gen draw=rnormal()
  6.         bys state: replace draw=draw[1]
  7.         sum draw
  8.         gen treat = draw == r(max) & year>=1989
  9.
.        //Estimate sdid based on permuted treatment and return weights
.        sdid packspercapita state year treat, vce(noinference) returnweights
 10.         gen weight = lambda*omega
 11.
.        //Estimate regression based version which also returns FEs
.        reg packspercapita treat i.st i.year [aw=weight]
 12.     }
 13.     restore
 14.     qui replace Taus = _b[treat] in `b´
 15. }

. sum Taus

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        Taus |         50    1.559062    9.312628  -31.75704   14.86168
```

This procedure is documented as it is illustrative to note that other parameters from (1) can be simply estimated by weighted regression. Given the lack of valid standard errors directly from regression, this procedure should be implemented with care, though as we document in the permutation procedure above, one *can* implement valid inference directly via regression if desired, though this is less direct than implementing procedures directly in `sdid`. What's more, one could follow similar such procedures to examine the variability of *other* parameters such as returned weights, fixed effects, or the model constant, though at present, theoretical results are not available to validate the use of such procedures for valid inference on parameters apart from $\widehat{\tau}^{sdid}$ or $\widehat{\tau}^{sdid}_a$.

5 Conclusions

Recent advances in panel data methods have resulted in a rich set of tools which can be used to conduct policy and other analyses. Such methods, combined with a growing emphasis on formal identification and broadening availability of large datasets and computing power available to researchers (Currie et al. 2020) suggests we will see an increasing adoption of these tools in applied research in the social sciences. Two particularly common approaches to analysis in a panel data or repeated cross sectional settings are difference-in-differences, and synthetic control methods. While both are sharply increasing in frequency in applied research (Currie et al. 2020), they are often used in different cases: DID is commonly employed with many treated units and potentially fewer time period, while SC is often implemented with fewer treatment units, many controls, and longer pre-treatment periods. What's more, both settings have particular assumptions which may not be met in practice: DID requires parallel trends between treated and control units, while SC requires a convex hull assumption which is not met if treatment outcomes are more extreme than control outcomes.

In this paper we have laid out the details behind Arkhangelsky et al. (2021)'s SDID method, which allows for the relaxation of assumptions required in both DID and SC settings, while also maintaining desirable properties if these assumptions are met. We have discussed its implementation in Stata using the `sdid` command, which provides a computationally efficient way to conduct estimation, inference, and graphing. We have discussed the theory behind this command, as well as laid out extensions into a staggered adoption setting. We provide two empirical examples to demonstrate the usage of the command. While the main novelty of the `sdid` command introduced here is the implementation of SDID methods, it can similarly be used to implement DID and SC methods. The efficiency of estimation routines implemented for SDID are leveraged when estimating SC and DID methods, which implies that `sdid` may provide faster implementations than other SC libraries currently available in Stata in a range of contexts.

Nevertheless, despite the ease of implementation of `sdid`, there are a number of requirements which should be considered prior to usage. From a **practical standpoint**, given the nature of the SDID algorithm, a number of conditions must be met for this to be applied to data. Firstly, and most importantly, a balanced panel of data is required which provides outcomes and treatment measures for each unit in all periods under study. Should missing values in such outcomes be present in the panel, these either must be eliminated from the estimation sample, or data should be sought to fill in gaps (a similar requirement exists for control variables if controls are to be included in estimation). Secondly, no units can be considered if they were exposed to treatment from the first period in which data is observed. If this occurs, there is no pre-treatment period on which to generate synthetic control cohorts. If always-treated units are present in the data, these either need to be eliminated, or earlier data sought. Third, pure control units are required. At least some units must never be treated in order to act as donor units. If all units are treated at some point in the panel, no donor units exist, and synthetic controls cannot be generated. Finally, in the case of inference, a number of additional requirements must be met. In the case of bootstrap or jackknife procedures, the number of treated units should be larger than 1 (and ideally considerably larger than this). Should only 1 treated unit be present, placebo inference should be conducted. Additionally, in the case of placebo inference, this can only be conducted if the number of control units exceeds the number of treated units. If implementing DID or SC methods using `sdid`, these procedures rely on similar computational routines, and so inherit the same requirements mentioned above (though also inherit the same computational efficiencies built into SDID implementations).

From a **theoretical standpoint**, there are additional considerations which are important to keep in mind for the implementation of SDID procedures. In particular, it is important to note that normality results underlying the generation of variance estimates and confidence intervals hold asymptotically. Specifically, these results require that the entire panel size grows without limit, as does the number of control units, and the number pre-treatment periods. However, these results do not require for the number of treatment units to grow without limit, in particular, requiring that *either* the number of post-treatment periods *or* the number of treated units grows without limit (see Arkhangelsky et al. (2021, p. 4016). Reassuringly, simulation results from Arkhangelsky et al. (2021) suggest the the SDID estimator has good properties even with as few as 1 treatment unit, and with panels consisting of 50 states and 40 time periods. This raises the general question of what types of data properties are required for SDID estimation. A key element of estimation, as in the case of synthetic control estimation (Ben-Michael et al. 2021; Abadie et al. 2010), is the adequate modeling of pre-treatment trends. This requires a sufficient number of periods off of which to model the generation of synthetic cohorts, but also the avoidance of periods in which structural breaks occur in treatment outcomes (see, for example, practical discussion related to synthetic controls in Abadie (2021)). While there is not yet a method to determine the precise number of required pre-treatment periods for SDID modelling, simulation results in Arkhangelsky et al. (2021) show good behaviour in settings with 30 pre-treatment periods. In practice, considerably fewer pre-treatments are likely to be required, and indeed, the canonical example of synthetic control methods based on California's proposition 99 is based of 19 pre-treatment periods. What's more, SDID in general requires fewer pre-treatment periods than SC implementations given that SDID's optimal weighting of years *and* states in generating estimates has a double robustness property, while SC does not have this property (Arkhangelsky et al. 2021). Future work could further clarify data requirements in terms of minimum required pre-periods in SDID modelling.

In general, this is a fast-evolving field (Roth et al. 2022; Arkhangelsky and Imbens 2023) and there is substantial work of interest beyond the synthetic DID methods described in this paper. Among other things, synthetic control has been shown to gain from regularization (Ben-Michael et al. 2021; Victor Chernozhukov and Zhu 2021; Abadie and L'Hour 2021, among others), debiasing methods (Ferman and Pinto 2021) and extend to distributional considerations (Gunsilius 2023). Overall though, should a balanced panel of data be available, the SDID method, and the `sdid` command described here, offers a flexible, easy to implement and robust option for the analysis of impacts of policies or treatments in certain groups at certain times. These methods provide clear graphical results to describe outcomes, and an explicit description of how counterfactual outcomes are inferred. These methods are likely well suited to a large body of empirical work in social sciences, where treatment assignment is not random, and offer benefits over both difference-in-differences and synthetic control methods.

Acknowledgments

We are grateful to Asjad Naqvi for comments relating to this code, and many users of the `sdid` ado for sending feedback and suggestions related to certain features implemented here. We thank the editor Stephen P. Jenkins and an anonymous referee for many useful comments.

6 References

Abadie, A. 2021. Using Synthetic Controls: Feasibility, Data Requirements, and Methodological Aspects. *Journal of Economic Literature* 59(2): 391–425. https://www.aeaweb.org/articles?id=10.1257/jel.20191450.

Abadie, A., A. Diamond, and J. Hainmueller. 2010. Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association* 105(490): 493–505.

———. 2011. SYNTH: Stata module to implement Synthetic Control Methods for Comparative Case Studies. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s457334.html.

———. 2015. Comparative Politics and the Synthetic Control Method. *American Journal of Political Science* 59(2): 495–510. https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12116.

Abadie, A., and J. Gardeazabal. 2003. The Economic Costs of Conflict: A Case Study of the Basque Country. *American Economic Review* 93(1): 113–132.

Abadie, A., and J. L'Hour. 2021. A Penalized Synthetic Control Estimator for Disaggregated Data. *Journal of the American Statistical Association* 116(536): 1817–1834.

Arkhangelsky, D., S. Athey, D. A. Hirshberg, G. W. Imbens, and S. Wager. 2021. Synthetic Difference-in-Differences. *American Economic Review* 111(12): 4088–4118. https://www.aeaweb.org/articles?id=10.1257/aer.20190159.

Arkhangelsky, D., and G. Imbens. 2023. Causal Models for Longitudinal and Panel Data: A Survey. Working Paper 31942, National Bureau of Economic Research. http://www.nber.org/papers/w31942.

Athey, S., and G. W. Imbens. 2022. Design-based analysis in Difference-In-Differences settings with staggered adoption. *Journal of Econometrics* 226(1): 62–79. Annals Issue in Honor of Gary Chamberlain. https://www.sciencedirect.com/science/article/pii/S0304407621000488.

Ben-Michael, E., A. Feller, and J. Rothstein. 2021. The Augmented Synthetic Control Method. *Journal of the American Statistical Association* 116(536): 1789–1803.

Bhalotra, S., D. Clarke, J. F. Gomes, and A. Venkataramani. 2023. Maternal Mortality and Women's Political Power. *Journal of the European Economic Association* 21(5): 2172–2208. https://doi.org/10.1093/jeea/jvad012.

Bhuller, M., T. Havnes, E. Leuven, and M. Mogstad. 2013. Broadband Internet: An Information Superhighway to Sex Crime? *Review of Economic Studies* 80(4): 1237–1266. https://EconPapers.repec.org/RePEc:oup:restud:v:80:y:2013:i:4:p:1237-1266.

Bilinski, A., and L. A. Hatfield. 2018. Nothing to see here? Non-inferiority approaches to parallel trends and other model assumptions. *arXiv e-prints* arXiv:1805.03273.

Bleiberg, J. 2021. STACKEDEV: Stata module to implement stacked event study estimator. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459027.html.

Borusyak, K. 2021. DID_IMPUTATION: Stata module to perform treatment effect estimation and pre-trend testing in event studies. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458957.html.

de Chaisemartin, C., X. D'Haultfoeuille, and Y. Guyonvarch. 2019. DID_MULTIPLEGT: Stata module to estimate sharp Difference-in-Difference designs with multiple groups and periods. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458643.html.

de Chaisemartin, C., and X. D'Haultfœuille. 2022. Two-way fixed effects and differences-in-differences with heterogeneous treatment effects: a survey. *The Econometrics Journal* Utac017. https://doi.org/10.1093/ectj/utac017.

Clarke, D., and K. Tapia-Schythe. 2021. Implementing the panel event study. *The Stata Journal* 21(4): 853–884. https://doi.org/10.1177/1536867X211063144.

Conley, T. G., and C. R. Taber. 2011. Inference with "Difference in Differences" with a Small Number of Policy Changes. *The Review of Economics and Statistics* 93(1): 113–125.

Currie, J., H. Kleven, and E. Zwiers. 2020. Technology and Big Data Are Changing Economics: Mining Text to Track Methods. *AEA Papers and Proceedings* 110: 42–48. https://www.aeaweb.org/articles?id=10.1257/pandp.20201058.

Doudchenko, N., and G. W. Imbens. 2016. Balancing, Regression, Difference-In-Differences and Synthetic Control Methods: A Synthesis. https://arxiv.org/abs/1610.07748.

Ferman, B., and C. Pinto. 2021. Synthetic controls with imperfect pretreatment fit. *Quantitative Economics* 12(4): 1197–1221. https://onlinelibrary.wiley.com/doi/abs/10.3982/QE1596.

Frank, M., and P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Research Logistics Quarterly* 3(1-2): 95–110. https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109.

Freyaldenhoven, S., C. Hansen, J. E. P. Perez, and J. Shapiro. 2021. XTEVENT: Stata module to estimate and visualize linear panel event-study models. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458987.html.

Freyaldenhoven, S., C. Hansen, and J. M. Shapiro. 2019. Pre-event Trends in the Panel Event-Study Design. *American Economic Review* 109(9): 3307–38. http://www.aeaweb.org/articles?id=10.1257/aer.20180609.

Galiani, S., and B. Quistorff. 2017. The synth runner package: Utilities to automate synthetic control estimation using synth. *The Stata Journal* 17(4): 834–849.

Goodman-Bacon, A. 2021. The Long-Run Effects of Childhood Insurance Coverage: Medicaid Implementation, Adult Health, and Labor Market Outcomes. *American Economic Review* 111(8): 2550–93. https://www.aeaweb.org/articles?id=10.1257/aer.20171671.

Greathouse, J. 2022. SCUL: Stata module to implement regularized synthetic control (using LASSO) estimators for single and multiple-treated unit settings. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459107.html.

Gunsilius, F. F. 2023. Distributional Synthetic Controls. *Econometrica* 91(3): 1105–1117. https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA18260.

Hirshberg, D. A. Undated. synthdid: Synthetic Difference in Differences Estimation. https://synth-inference.github.io/synthdid/.

Holland, P. W. 1986. Statistics and Causal Inference. *Journal of the American Statistical Association* 81(396): 945–960.

Jann, B. 2004. ESTOUT: Stata module to make regression tables. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s439301.html.

Kranz, S. 2022. Synthetic Difference-in-Differences with Time-Varying Covariates. Technical report. Available online at: https://github.com/skranz/xsynthdid/blob/main/paper/synthdid_with_covariates.pdf.

Lawphongpanich, S. 2009. *Encyclopedia of Optimization*, chap. Frank–Wolfe Algorithm, 1094–1097. Boston, MA: Springer US.

Manski, C. F., and J. V. Pepper. 2018. How Do Right-to-Carry Laws Affect Crime Rates? Coping with Ambiguity Using Bounded-Variation Assumptions. *The Review of Economics and Statistics* 100(2): 232–244. https://doi.org/10.1162/REST_a_00689.

Orzechowski, W., and R. C. Walker. 2005. The Tax Burden on Tobacco. Historical Compilation Volume 40, Arlington, VA.

Pailañir, D., and D. Clarke. 2022. SDID: Stata module to perform synthetic difference-in-differences estimation, inference, and visualization. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459058.html.

Rambachan, A., and J. Roth. 2023. A More Credible Approach to Parallel Trends. *The Review of Economic Studies* 90(5): 2555–2591.

Rios-Avila, F. 2022. JWDID: Stata module to estimate Difference-in-Difference models using Mundlak approach. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459114.html.

Rios-Avila, F., P. H. Sant'Anna, and B. Callaway. 2021. CSDID: Stata module for the estimation of Difference-in-Difference models with multiple time periods. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458976.html.

Roth, J., P. H. C. Sant'Anna, A. Bilinski, and J. Poe. 2022. What's Trending in Difference-in-Differences? A Synthesis of the Recent Econometrics Literature. https://arxiv.org/abs/2201.01194.

Rubin, D. B. 2005. Causal Inference Using Potential Outcomes. *Journal of the American Statistical Association* 100(469): 322–331.

Schmidheiny, K., and S. Siegloch. 2019. On Event Study Designs and Distributed-Lag Models: Equivalence, Generalization and Practical Implications. IZA Discussion Papers 12079, Institute of Labor Economics (IZA).

Sun, L. 2021. EVENTSTUDYINTERACT: Stata module to implement the interaction weighted estimator for an event study. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458978.html.

Sun, L., and S. Abraham. 2021. Estimating dynamic treatment effects in event studies with heterogeneous treatment effects. *Journal of Econometrics* 225(2): 175–199. https://EconPapers.repec.org/RePEc:eee:econom:v:225:y:2021:i:2:p:175-199.

Victor Chernozhukov, K. W., and Y. Zhu. 2021. An Exact and Robust Conformal Inference Method for Counterfactual and Synthetic Controls. *Journal of the American Statistical Association* 116(536): 1849–1864.

Wiltshire, J. C. 2022. ALLSYNTH: Stata module to automate estimation of (i) bias-corrected synthetic control gaps ("treatment effects"). Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459076.html.

Yan, G., and Q. Chen. 2023. synth2: Synthetic control method with placebo tests, robustness test, and visualization. *The Stata Journal* 23(3): 597–624.

About the authors

Susan Athey is the Economics of Technology Professor at Stanford Graduate School of Business.

Damian Clarke is an Associate Professor at The Department of Economics of The Universidad de Chile, a Research Fellow at IZA and an Associate at the Millennium Institute for Market Imperfections and Public Policy and CAGE, Warwick.

Guido Imbens is the Applied Econometrics Professor and Professor of Economics at Stanford Graduate School of Business.

Daniel Pailañir is a Senior Analyst at Ministry of Economics, Development and Tourism of Chile.