The synth_runner Package: Utilities to Automate Synthetic Control Estimation Using synth*

Sebastian Galiani
University of Maryland

Brian Quistorff
University of Maryland

December 4, 2015

Abstract

Synthetic Controls Methodology (Abadie and Gardeazabal, 2003; Abadie et al., 2010) allows for a data-driven approach to small-sample comparative studies. `synth_runner` automates the process of running multiple synthetic controls estimations using `synth`. It conducts placebo estimates in-space (estimations for the same treatment period but on all the control units). Inference (p-values) is provided by comparing the estimated main effect to the distribution of placebo effects. It allows several units to receive treatment, possible at different time periods. Automatically generating the outcome predictors and diagnostics by splitting the pre-treatment into training and validation portions is allowed. Additionally, it provides diagnostics to assess fit and generates visualizations of results.

Keywords: Synthetic Controls Methodology,

1 Introduction

Synthetic Controls Methodology (SCM) (Abadie and Gardeazabal, 2003; Abadie et al., 2010) is a data-drive approach to small-sample comparative case-studies to

*Contact: Galiani@econ.umd.edu, bquistorff@gmail.com.

We thank the Inter-American Development Bank for financial support.

estimate treatment effects. Similar to a difference-in-difference design, synthetic controls exploits the difference in treated and untreated units across the event of interest. However, in contrast to a difference-in-differences design, synthetic controls does not give all untreated units the same weight in the comparison. Instead, it generates a weighted average of the untreated units that closely matches the treated unit over the pre-treatment period. Outcomes for this synthetic control are then projected into the post-treatment period using the weights identified from the pre-treatment comparison. This projection is used as the counterfactual for the treated unit. Inference is conducting using placebo tests.

Along with their paper, Abadie et al. (2010) released the `synth` Stata command for single estimations. The `synth_runner` package builds on top of that command to help conduct multiple estimations, inference, diagnostics, and generate visualizations of results.

2 Synthetic Controls Methodology

Abadie et al. (2010) posit the following data-generating process. Let D be an indicator for treatment and the observed outcome variable Y_{jt} (for unit j and time t) is the sum of a time-varying treatment effect $\alpha_{jt}D_{jt}$ and the no-treatment counterfactual Y_{jt}^N, which is specified using a factor model

$$\begin{aligned} Y_{jt} &= \alpha_{jt}D_{jt} + Y_{jt}^N \\ &= \alpha_{jt}D_{jt} + (\delta_t + \theta_t \mathbf{Z}_j + \lambda_t \mu_j + \varepsilon_{jt}) \end{aligned} \tag{1}$$

where δ_t is an unknown time factor, $\mathbf{Z}_j$ is a $(r \times 1)$ vector of observed covariates unaffected by treatment, θ_t is a $(1 \times r)$ vector of unknown parameters, λ_t is a $(1 \times F)$ vector of unknown factors, μ_j is a $(F \times 1)$ vector of unknown factor loadings, and the error ε_{jt} is independent across units and time with zero mean. Letting the first unit be the treated unit, the treatment effect is estimated by approximating the unknown Y_{1t}^N with a weighted average of untreated units

$$\hat{\alpha}_{1t} = Y_{1t} - \sum_{j \geq 2} w_j Y_{jt}$$

Equation 1 simplifies to the traditional fixed effect equation if $\lambda_t \mu_j = \phi_j$. The fixed effect model allows for unobserved heterogeneity that is only time-invariant. The factor model employed by synthetic controls generalizes this to allow for the existence of non-parallel trends between treated and untreated units after controlling for observables.

2.1 Estimation

To begin with, let there be a single unit that receives treatment. Let T_0 be the number of pre-treatment periods of the T total periods. Index units $\{1, ..., J+1\}$ such that the first unit is the treated unit and the others are "donors". Let $\mathbf{Y}_j$ be $(T \times 1)$ the vector of outcomes for unit j and $\mathbf{Y}_0$ be the $(T \times J)$ matrix of outcomes for all donors. Let $\mathbf{W}$ be a $(J \times 1)$ observation-weight matrix $(w_2, w_3, ..., w_{J+1})'$ where $\sum_{j=2}^{J+1} w_j = 1$ and $w_j \geq 0 \ \forall j \in \{2, ..., J+1\}$. A weighted average of donors over the outcome is constructed as $\mathbf{Y}_0\mathbf{W}$. Partition the outcome into pre-treatment and post-treatment vectors $\mathbf{Y}_j = (\overleftarrow{\mathbf{Y}}_j \backslash \overrightarrow{\mathbf{Y}}_j)$. Let $\mathbf{X}$ represent a set of k pre-treatment characteristics ("predictors"). This includes $\mathbf{Z}$ (the observed covariates above) and M linear combinations of $\overleftarrow{\mathbf{Y}}$ so that $k = r + M$. Analogously, let $\mathbf{X}_0$ be the $(k \times J)$ matrix of donor predictors. Let $\mathbf{V}$ be a $(k \times k)$ variable-weight matrix indicating the relative significance of the predictor variables.

Given $\mathbf{Y}$ and $\mathbf{X}$, estimation of synthetic controls consists of finding the optimal weighting matrices $\mathbf{W}$ and $\mathbf{V}$. The inferential procedure is valid for any $\mathbf{V}$ but Abadie et al. (2010) suggest that $\mathbf{V}$ be picked to minimize the prediction error of the pre-treatment outcome between the treated unit the synthetic control. Define distance measures $\|\mathbf{A}\|_{\mathbf{B}} = \sqrt{\mathbf{A}'\mathbf{B}\mathbf{A}}$ and $\|\mathbf{A}\| = \sqrt{\mathbf{A}'\text{cols}(\mathbf{A})^{-1}\mathbf{A}}$. $\left\|\overleftarrow{\mathbf{Y}}_1 - \overleftarrow{\mathbf{Y}}_0\mathbf{W}\right\|$ is then the pre-treatment root mean squared prediction error (RMSPE) with a given weighted average of the control units. Let $\overleftarrow{s}_1$ be the pre-treatment RMSPE and $\overrightarrow{s}_1$ be the post-treatment RMSPE. $\mathbf{W}$ is picked to minimize the RMSPE of the predictor variables, $\|\mathbf{X}_1 - \mathbf{X}_0\mathbf{W}\|_{\mathbf{V}}$. In this way, the treated unit and its synthetic control look similar along dimensions that matter for predicting pre-treatment outcomes.

If weights can be found such that the synthetic control matches the treated unit in the pre-treatment period:

$$\left\|\overleftarrow{\mathbf{Y}}_1 - \overleftarrow{\mathbf{Y}}_0\mathbf{W}\right\| = 0 = \|\mathbf{Z}_1 - \mathbf{Z}_0\mathbf{W}\| \tag{2}$$

and $\sum_{t=1}^{T_0} \lambda_t' \lambda_t$ is non-singular, then $\hat{\alpha}_1$ will have a bias that goes to zero as the number of pre-intervention periods grows large relative to the scale of the ε_{jt}.

2.2 Inference

After estimating the effect, significance is determined by running placebo tests. Estimate the same model on each untreated unit (as if it were treated) to get a distribution of effects. These are called in-place placebos. If the distribution of placebo effects yields many effects as large as the main estimate, then it is likely that it was observed by chance. This non-parametric, exact test has the advantage of not imposing any distribution on the errors.

For inference, one may want to consider other measures of effect aside from the $\hat{\alpha}_{1t}$ estimates for post-treatment periods. To gauge the joint effect across all periods Abadie et al. (2010) suggest using post-treatment RMSPE $\vec{s}_1$. Additionally, one may want to adjust these estimates for the quality of the pre-treatment match. The quality of the pre-treatment match in each placebo may vary significantly. If the match is poor, then the post-treatment prediction error is also likely to be large. One can adjust the measures of effect by dividing them by the pre-treatment match quality ($\overleftarrow{s}$)[1] to get "pseudo t-statistics". Let $\hat{\beta}$ be a measure of effect. For inference then, there are four types: $\hat{\beta} \in \{\hat{\alpha}_t, \vec{s}, {}^{\hat{\alpha}_t}/_{\overleftarrow{s}}, {}^{\vec{s}}/_{\overleftarrow{s}}\}$

Then $\hat{\beta}_1$ is compared against the distribution of placebo effects conducted with treatment at the same time as unit 1, $\hat{\beta}_1^{PL} = \{\hat{\beta}_1^{PL(j)}\}_j$. The two-sided p-values are then

$$\begin{aligned} p\text{-value} &= \Pr\left(|\hat{\beta}_1^{PL}| \geq |\hat{\beta}_1|\right) \\ &= \frac{\sum_j 1(|\hat{\beta}_1^{PL(j)}| \geq |\hat{\beta}_1|)}{J} \end{aligned}$$

and the one-sided p-values being (for positive estimated effects)

$$p\text{-value} = \Pr\left(\hat{\beta}_1^{PL} \geq \hat{\beta}_1\right)$$

Even if treatment is not truly random (conditional on covariates and unknown

[1] Abadie et al. (2010) note an alternative way to adjust estimates by match quality by removing from comparison donors that have a match quality above a fixed multiple of the treated unit. This does not generalize as simply to the case of multiple treated units and is therefore not discussed here.

factors) the p-value has the straightforward interpretation of being the proportion of control units that have an estimated effect at least as large as that of the treated unit.

2.3 Multiple Events

The extension by Cavallo et al. (2013) allows for more than one unit to experience treatment and at possibly different times. Index treatment units $g \in \{1...G\}$. Let J be the set of donors, those units that never undergo treatment. For a particular treatment g, one can estimate the effect $\hat{\beta}_g$ and the set of placebos $\{\beta_g^{PL(j)}\}_{j\in J}$ where each untreated unit is thought of as entering treatment at the same time as unit g. If two treated units have the same treatment period, then their placebo sets will the be the same. The overall treatment measure is the average of the individual treatment effects $\bar{\beta} = G^{-1}\sum_{g=1}^{G}\hat{\beta}_g$.

Averaging over the treatments smooths out noise in the estimate. The same should be done to the set of placebos against which the treatment estimate is compared for inference. Let i index a selection where a single placebo effect is chosen from each treatment placebo set. Then $\bar{\beta}^{PL(i)}$ represents one such placebo average. There are $N_{\overline{PL}} = J^G$ such possible averages. Inferences is now

$$\begin{aligned} p\text{-value} &= \Pr\left(|\bar{\beta}^{PL}| \geq |\bar{\beta}|\right) \\ &= \frac{\sum_{i=1}^{N_{\overline{PL}}} 1(\bar{\beta}^{PL(i)} \geq \bar{\beta})}{N_{\overline{PL}}} \end{aligned}$$

Confidence intervals can be constructed by inverting the p-values for post-treatment $\{\hat{\alpha}_t\}$. These only have the proper interpretation, however, if treatment is truly random. They do not retain a meaningful interpretation when treatment is not random.

2.4 Diagnostics

Cavallo et al. (2013) perform two basic checks to see if the synthetic control serves as a valid counterfactual. The first is to check if a weighted average of donors is able to approximate the treated unit in the pre-treatment. This should be satisfied if the treated unit lies within the convex hull of the control units. One can visually compare the difference in pre-treament outcomes between a unit and its synthetic control. Additionally one could look at the distribution of pre-treatment RMSPE's and see what proportion of control units have values at least as high as that of the treated unit. Cavallo et al. (2013) discard several events from study as they can not be matched appropriately.

Secondly, one can exclude some pre-treatment outcomes from the list of predictors and see if the synthetic control matches well the treated unit in these periods.[2] As this is still pre-treatment, the synthetic control should match well. The initial section of the pre-treatment period is often designated the "training" period with the later part being the "validation" period. Cavallo et al. (2013) set aside the first half of the pre-treatment period as the training period.

3 The `synth_runner` Package

The `synth_runner` package contains several tools to help conduct Synthetic Controls estimation. It requires the `synth` package which can be obtained from the SSC archive. The main program is `synth_runner`, which is outlined here. Additionally, there are simple graphing utilities (`effect_graphs`, `pval_graphs`, `single_treatment_graphs`) that can be modified easily to show basic graphs. These are explained in the following code examples.

3.1 Syntax

`synth_runner` *depvar predictorvars* `, [ trunit(#) trperiod(#) d(`*varname*`)` `trends pre_limit_mult(`*real*`) training_propr(`*real*`) keep(`*file*`) replace ci pvals1s` *synthsettings* `]`

Post-estimation graphing commands are shown in the examples below.

3.2 Settings

Required Settings:

- `depvar` the outcome variable.

[2]Note also that unless some pre-treatment outcome variables are dropped from the set of predictors, all other covariate predictors are rendered redundant. The optimization of V will put no weight on those additional predictors in terms of predicting pre-treatment outcomes.

- `predictorvars` the list of predictor variables. See `help synth` help for more details.

For specifying the unit and time period of treatment, there are two methods. Exactly one of these is required.

- `trunit(#)` and `trperiod(#)`. This syntax (used by `synth`) can be used when there is a single unit entering treatment.

- `d(`*varname*`)`. The `d` variable should be a binary variable which is 1 for treated units in treated periods, and 0 everywhere else. This allows for multiple units to undergo treatment, possibly at different times.

Options:

- `trends` will force `synth` to match on the trends in the outcome variable. It does this by scaling each unit's outcome variable so that it is 1 in the last pre-treatment period.

- `pre_limit_mult`($real \geq 1$) will not include placebo effects in the pool for inference if the match quality of that control (pre-treatment RMSPE) is greater than *pre_limit_mult* times the match quality of the treated unit.

- `training_propr`($0 \leq real \leq 1$) instructs `synth_runner` to automatically generate the outcome predictors. The default (0) is to not generate any (the user then includes the desired ones in `predictorvars`). If set to a number greater than 0, then that initial proportion of the pre-treatment period is used as a training period with the rest being the validation period. Outcome predictors for every time in the training period will be added to the `synth` commands. Diagnostics of the fit for the validation period will be outputted. If the value is between 0 and 1, there will be at least one training period and at least one validation period. If it is set to 1, then all the pre-treatment period outcome variables will be used as predictors. This will make other covariate predictors redundant.

- `ci` outputs confidence intervals from randomization inference for raw effect estimates. These should only be used if the treatment is randomly assigned (conditional on covariates and interactive fixed-effects). If treatment is not randomly assigned then these confidence intervals do not have a straightforward interpretation (in contrast to p-values which do).

- `pvals1s` outputs one-sided p-values in addition to the two-sided p-values.
- `keep(`*filename*`)` saves a dataset with the results. This is only allowed if there is a single period in which unit(s) enter treatment. It is easy to merge this in the initial dataset. If `keep(`*filename*`)` is specified, it will hold the following variables:
  - *`panelvar`* contains the respective panel unit (from the `tsset` panel unit variable *panelvar*}.
  - *`timevar`* contains the respective time period (from the `tsset` panel time variable *timevar*).
  - `lead` contains the respective time period relative to the treatment period.
  - `effect` contains the difference between the unit's outcome and its synthetic control for that time period.
  - `pre_rmspe` contains the pre-treatment match quality in terms of Root Mean Squared Predictive Error. It is constant for a unit.
  - `post_rmspe` contains a measure of the post-treatment effect (jointly over all post-treatment time periods) in terms of Root Mean Squared Predictive Error. It is constant for a unit.
  - *`depvar`*`_scaled` (if the match was done on trends) is the unit's outcome variable normalized so that its last pre-treatment period outcome is 1.
  - `effect_scaled` (if the match was done on trends) is the difference between the unit's scaled outcome and its scaled synthetic control for that time period.
- `replace` replaces the dataset specified in `keep(`*filename*`)` if it already exists.
- *synthsettings* pass-through options sent to `synth`. See `help synth` for more information.

3.3 Saved Results

By default, `synth_runner` returns the following scalars and matrices:

- `e(treat_control)` - A matrix with the average treatment outcome (centered around treatment) and the average of the outcome of those unit's synthetic controls for the pre- and post-treatment periods.
- `e(b)` - A vector with the per-period effects (unit's actual outcome minus the outcome of its synthetic control) for post-treatment periods.
- `e(pvals)` - A vector of the proportions of placebo effects that are at least as large as the main effect for each post-treatment period.
- `e(pvals_t)` - A vector of the proportions of placebo pseudo t-statistics (unit's effect divided by its pre-treatment RMSPE) that are at least as large as the main pseudo t-statistic for each post-treatment period.
- `e(pval_joint_post)` - The proportion of placebos that have a post-treatment RMSPE at least as large as the average for the treated units.
- `e(pval_joint_post_t)`- The proportion of placebos that have a ratio of post-treatment RMSPE over pre-treatment RMSPE at least as large as the average ratio for the treated units.
- `e(avg_pre_rmspe_p)` - The proportion of placebos that have a pre-treatment RMSPE at least as large as the average of the treated units. A measure of fit. Bad if significant.
- `e(avg_val_rmspe_p)` - When specifying `training_propr`, this is the proportion of placebos that have a RMSPE for the validation period at least as large as the average of the treated units. A measure of fit. Bad if significant.

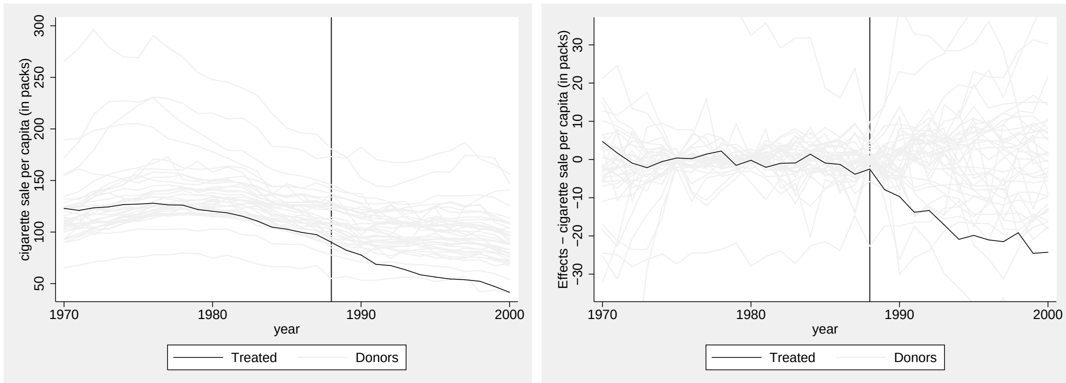
3.4 Example Usage

First load the Example Data from `synth`: This panel dataset contains information for 39 US States for the years 1970-2000 (see Abadie et al. (2010) for details).

```
. sysuse smoking
. tsset state year
```

Figure 1: Graphs from `single_treatment_graphs`

(a) Outcome trends (b) Effects



Example 1

Reconstruct the initial `synth` example:

```
. tempfile keepfile
. synth_runner cigsale beer(1984(1)1988) lnincome
   (1972(1)1988) retprice age15to24 cigsale(1988)
   cigsale(1980) cigsale(1975), trunit(3) trperiod(1989)
    keep(‘keepfile’)
. merge 1:1 state year using ‘keepfile’, nogenerate
. gen cigsale_synth = cigsale-effect
```

In this example, `synth_runner` conducts all the estimations and inference. Since there was only a single treatment period we can save the output and merge it back into the dataset. Then we can create the various graphs. Graphs produced by the graphing utilities are showing in Figures 1, 2, and 3.

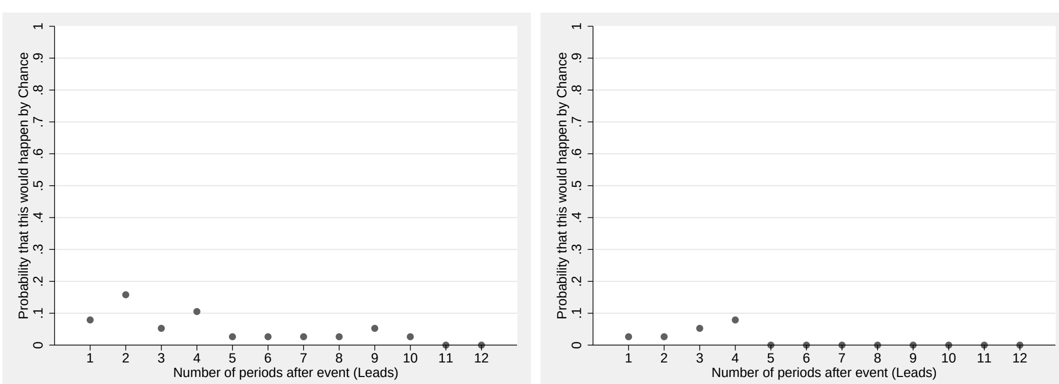
```
. single_treatment_graphs, depvar(cigsale) trunit(3)
   trperiod(1989) effects_ylabels(-30(10)30)
   effects_ymax(35) effects_ymin(-35)
. effect_graphs , depvar(cigsale) depvar_synth(
   cigsale_synth) trunit(3) trperiod(1989) effect_var(
   effect)
. pval_graphs
```

Figure 2: Graphs from `effect_graphs`

(a) Effect (b) Treatment vs Control

Figure 3: Graphs from `pval_graphs`

(a) p-values (b) p-values (psuedo t-stats)



Example 2

Same treatment, but a bit more complicated setup:

```
. gen byte D = (state==3 & year>=1989)
. tempfile keepfile2
. synth_runner cigsale beer(1984(1)1988) lnincome
   (1972(1)1988) retprice age15to24, trunit(3) trperiod
   (1989) trends training_propr(`=13/18') pre_limit_mult
   (10) keep(`keepfile2')
. merge 1:1 state year using `keepfile2', nogenerate
. gen cigsale_scaled_synth = cigsale_scaled -
   effect_scaled
. single_treatment_graphs, depvar(cigsale_scaled)
   effect_var(effect_scaled) trunit(3) trperiod(1989)
. effect_graphs , depvar(cigsale_scaled) depvar_synth(
   cigsale_scaled_synth) effect_var(effect_scaled)
   trunit(3) trperiod(1989)
. pval_graphs
```

Again there is a single treatment period, so output can be saved and merged back into the dataset. In this setting we (a) specify the treated units/periods with a binary variable, (b) generate the outcome predictors automatically using the initial 13 periods of the pre-treatment era (the rest is the "validation" period), (c) we match on trends, and (d) we limit during inference control units whose pre-treatment match quality more than 10 times worse than the match quality of the corresponding treatment units. The graphing commands are equivalent. The ones showing the range of effects and raw data are shown in Figure 4.

Example 3

Multiple treatments at different time periods:

```
. gen byte D = (state==3 & year>=1989) | (state==7 &
   year>=1988)
. synth_runner cigsale beer(1984(1)1987) lnincome
   (1972(1)1987) retprice age15to24, d(D) trends
   training_propr(`=13/18')
```

Figure 4: Graphs from `single_treatment_graphs`

(a) Outcome trends (b) Effects

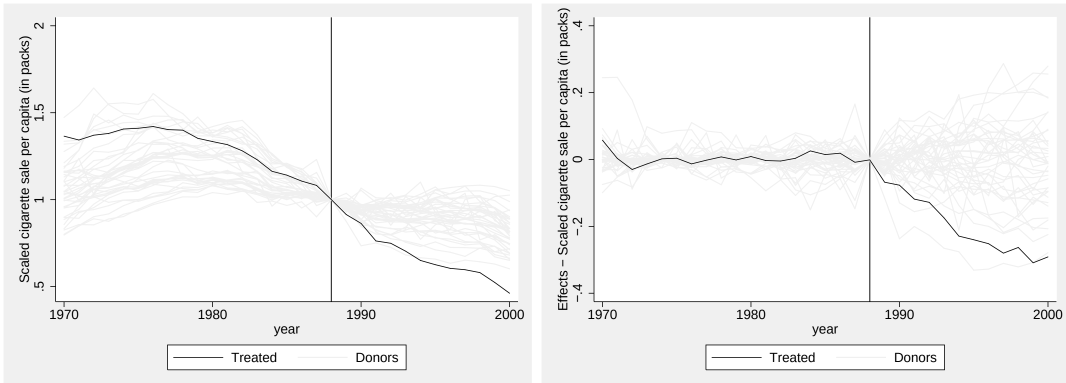
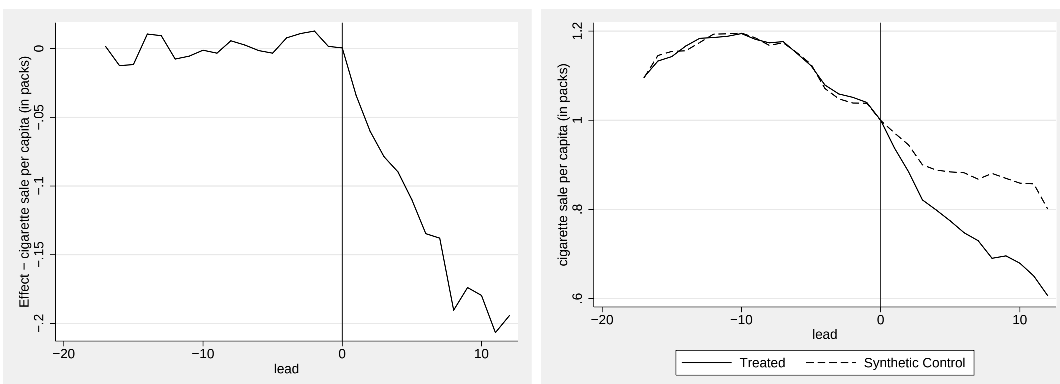


Figure 5: Graphs from `effect_graphs`

(a) Effect (b) Treatment vs Control




```
. effect_graphs , multi depvar(cigsale)
. pval_graphs
```

We extend Example 2 by considering a control state now to be treated (Georgia in addition to California). No treatment actually happened in Georgia in 1987. Now that we have several treatment periods we can not merge in a simple file. Some of the graphs (of `single_treatment_graphs`) can no longer be made. The option *multi* is now passed to `effect_graphs` and those are shown in Figure 5.

4 Discussion

Synthetic Controls Methodology (SCM) (Abadie and Gardeazabal, 2003; Abadie et al., 2010) is allowing many researchers to quantitatively estimate effects in small sample settings in a manner grounded by theory. This article provides an overview of the theory of SCM and the `synth_runner` package, which builds on the `synth` package of Abadie et al. (2010). `synth_runner` provides tools to help with the common tasks of estimating a synthetic control model. It automates the process of conducting in-place placebos and calculating inference on the various possible measures. Following Cavallo et al. (2013) it (a) extends the initial estimation strategy to allow for multiple units that receive treatment (at potentially different times), (b) allows for matching on trends in the outcome variable rather than on the level, and (c) automates the process of splitting pre-treatment periods into "training" and "validation" sections. It provides graphs of diagnostics, inference, and estimate effects.

References

Alberto Abadie and Javier Gardeazabal. The economic costs of conflict: A case study of the Basque country. *American Economic Review*, 93(1):113–132, Mar 2003. doi:10.1257/000282803321455188.

Alberto Abadie, Alexis Diamond, and Jens Hainmueller. 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, Jun 2010. doi:10.1198/jasa.2009.ap08746.

Eduardo Cavallo, Sebastian Galiani, Ilan Noy, and Juan Pantano. Catastrophic natural disasters and economic growth. *Review of Economics and Statistics*, 95 (5):1549–1561, Dec 2013. doi:10.1162/rest_a_00413.