Synth: An R Package for Synthetic Control Methods in Comparative Case Studies

Journal of Statistical Software

June 2011, Volume 42, Issue 13. *http://www.jstatsoft.org/*

Synth: An R Package for Synthetic Control Methods in Comparative Case Studies

Alberto Abadie
Harvard University

Alexis Diamond
Harvard University

Jens Hainmueller
Massachusetts Institute of Technology

Abstract

The R package **Synth** implements synthetic control methods for comparative case studies designed to estimate the causal effects of policy interventions and other events of interest (Abadie and Gardeazabal 2003; Abadie, Diamond, and Hainmueller 2010). These techniques are particularly well-suited to investigate events occurring at an aggregate level (i.e., countries, cities, regions, etc.) and affecting a relatively small number of units. Benefits and features of the **Synth** package are illustrated using data from Abadie and Gardeazabal (2003), which examined the economic impact of the terrorist conflict in the Basque Country.

Keywords: synthetic control methods, differences in differences estimation, program evaluation, comparative case studies, causal inference.

1. Introduction

Much of social science is concerned with causal questions about the effects of historical events and policy interventions on aggregate units, such as cities, regions, and countries. A classic method of answering such questions is the comparative case study, in which investigators compare outcomes for the unit(s) affected by an event or intervention (the treated group) to outcomes for one or more unaffected units (the control group). The rationale behind this method is to use the control group's outcome to approximate the outcome that would have been observed for the treated group in the absence of treatment. Traditional comparative case study methods leave the choice of control units to the analyst, prompting questions about the arbitrariness of selection and the degree to which control units can credibly proxy for treated units' counterfactual outcomes. Synthetic control methods, introduced by Abadie and Gardeazabal (2003) and Abadie *et al.* (2010), address these shortcomings by proposing

a data-driven control-group selection procedure, a framework for assessing the suitability of the chosen control group, and a means of producing quantitative inference.

Abadie and Gardeazabal (2003) and Abadie *et al.* (2010) define a synthetic control unit as a weighted average of available control units that approximates the most relevant characteristics of the treated unit prior to the treatment. Synthetic control methods make explicit the relative contribution of each available control unit and the degree of similarity prior to treatment between a treated unit and its synthetic counterpart. An attractive feature of the synthetic control method is that it guards against extrapolation outside the convex hull of the data because weights from all control units can be chosen to be positive and sum to one. Abadie *et al.* (2010) motivate the synthetic control method with a model that generalizes the difference-in-differences (fixed-effects) model commonly applied in the empirical social science literature by allowing the effect of unobserved confounding characteristics to vary over time.

The aim of this paper is to present the **Synth** package which implements synthetic control methods in R (R Development Core Team 2011) and is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=Synth`.[1] The central function in the package is `synth()`, which constructs the synthetic control unit by solving an optimization problem to identify a set of weights that are assigned to potential control units. Another important function is `dataprep()` which allows the user to easily organize the data in a format needed to run `synth()`. Other functions such as `synth.tables()`, `path.plot()`, and `gaps.plot()` produce tables and figures that summarize and illustrate the results. Our data example is from Abadie and Gardeazabal (2003), which introduced synthetic control methods to investigate the effects of the terrorist conflict in the Basque Country on the Basque economy using other Spanish regions as potential control units.

The rest of the paper is organized as follows. Section 2 briefly reviews the synthetic control method, restating the key elements of Abadie and Gardeazabal (2003) and Abadie *et al.* (2010). In section 3 we demonstrate the use of the main functions and methods of **Synth** with an example. Section 4 concludes by describing future extensions to the **Synth** package.

2. Synthetic Control Methods

Synthetic control methods involve the construction of synthetic control units as convex combinations of multiple control units. The weights that define the synthetic control unit are chosen such that the synthetic control unit best approximates the relevant characteristics of the treated unit during the pretreatment period. The post-intervention outcomes for the synthetic control unit are then used to estimate the outcomes that would have been observed for the treated unit in the absence of the intervention.

Abadie *et al.* (2010) provide a formal discussion of the theoretical properties of the synthetic control method. In particular, they derive the synthetic control estimator using an econometric model that generalizes the usual difference-in-differences (fixed-effects) model commonly applied in the empirical literature. Here we focus on how the **Synth** package can be used to implement the synthetic control method, and thus provide only a very brief review of the material in Abadie and Gardeazabal (2003) and Abadie *et al.* (2010).

[1] Software implementations of the synthetic control method for other packages such as MATLAB (The MathWorks, Inc. 2007) and Stata (StataCorp. 2007) are also available on the corresponding author's website at `http://www.mit.edu/~jhainm/software.htm`.

Suppose that we observe units $j = 1, ..., J+1$ for time periods $t = 1, \ldots, T$. Without loss of generality, we assume that only the first unit is exposed to the intervention so we have J remaining control units that can contribute to the synthetic control.[2] The set of control units is termed the donor pool. In the context of comparative case studies units are usually aggregate entities such as schools, regions, or countries, and the interventions or treatments are events such as economic shocks, the passages of laws, etc. The intervention occurs at time period $T_0 + 1$ so that $1, 2, ..., T_0$ are the pre-intervention periods and $T_0 + 1, T_0 + 2, ..., T$ are the post-intervention periods.[3]

We define two potential outcomes: Y_{it}^N refers to the outcome that would be observed for unit i at time t if unit i is not exposed to the intervention, and Y_{it}^I refers to the outcome that would be observed if unit i is exposed to the intervention. Our goal is to estimate the effect of the intervention on the outcome for the treated unit in the post-intervention period. This effect is formally defined as the difference between the two potential outcomes $\alpha_{1t} = Y_{1t}^I - Y_{1t}^N$ for periods $T_0 + 1, T_0 + 2, \ldots, T$. Notice that Y_{it}^N is unobserved for the treated unit in the post-intervention period. The goal of the synthetic control method is to construct a synthetic control group that yields a reasonable estimate for this missing potential outcome.

Ideally, we would like to construct a synthetic control that resembles the treated unit in all relevant pre-intervention characteristics. Formalizing this idea we define U_i as a $(r \times 1)$ vector of observed covariates for each unit. These variables will commonly consist of a set of predictors of the outcome variable.[4] Moreover, we define a $(T_0 \times 1)$ vector $K = (k_1, \ldots, k_{T_0})'$ that denotes some linear combination of pre-intervention outcomes: $\bar{Y}_i^K = \sum_{s=1}^{T_0} k_s Y_{is}$. Linear combinations of pre-intervention outcomes can be used to control for unobserved common factors whose effects vary over time.[5] The user can choose to include as many as M (linearly independent) combinations of pre-intervention outcomes (with $M \leq T_0$) to control for such unobserved common factors.[6]

To construct our synthetic control unit we define a $(J \times 1)$ vector of weights $W = (w_2, \ldots, w_{J+1})'$ such that $w_j \geq 0$ for $j = 2, \ldots, J+1$ and $w_2 + \cdots + w_{J+1} = 1$. Each W then represents one particular weighted average of control units and therefore one potential synthetic control unit. Abadie and Gardeazabal (2003) and Abadie *et al.* (2010) propose to chose the weights W^* such that the resulting synthetic control unit best approximates the unit exposed to the intervention with respect to the outcome predictors U_i and M linear combinations of pre-intervention outcomes $\bar{Y}_i^{K_1}, \ldots, \bar{Y}_i^{K_M}$. Formally, we select $W^* = w_2^* + \cdots + w_{J+1}^*$ such that $\sum_{j=2}^{J+1} w_j^* \bar{Y}_j^{K_1} = \bar{Y}_1^{K_1} \cdots \sum_{j=2}^{J+1} w_j^* \bar{Y}_j^{K_M} = \bar{Y}_1^{K_M}$ and $\sum_{j=2}^{J+1} w_j^* U_j = U_1$ hold (or hold approximately). Then

$$\widehat{\alpha}_{1t} = Y_{1t} - \sum_{j=2}^{J+1} w_j^* Y_{jt}$$

yields an estimator of α_{1t} in periods $T_0+1, T_0+2, \ldots, T$. A formal discussion of the properties

[2] In cases where there are multiple units exposed to the intervention the user can first aggregate the data from the regions exposed to the intervention.

[3] For notational convenience and without loss of generality we assume that the treated unit is uninterruptedly exposed to the intervention in the post-treatment period.

[4] The set of covariates is usually restricted to variables that are measured before the intervention occurs, but the user could include post-intervention characteristics as long as they are unaffected by the intervention.

[5] See Abadie *et al.* (2010) for details.

[6] In the example of section III below, we use only one of such linear combinations of pre-intervention outcomes: the average of the outcome variable for ten pre-intervention periods.

of the synthetic control estimator is provided in Abadie *et al.* (2010).

In empirical applications it is often the case that there exists no sets of weights such that $\sum_{j=2}^{J+1} w_j^* \bar{Y}_j^{K_1} = \bar{Y}_1^{K_1} \cdots \sum_{j=2}^{J+1} w_j^* \bar{Y}_j^{K_M} = \bar{Y}_1^{K_M}$ and $\sum_{j=2}^{J+1} w_j^* U_j = U_1$ hold exactly, because the characteristics of the treated unit $(U_1, \bar{Y}_1^K, \ldots, \bar{Y}_1^{K_M})$ are outside of the convex hull of the characteristics of the control units $\{(U_2, \bar{Y}_2^{K_1}, \ldots, \bar{Y}_2^{K_M}), \ldots, (U_{J+1}, \bar{Y}_{J+1}^{K_1}, \ldots, \bar{Y}_{J+1}^{K_M})\}$. In these cases the weights are chosen so that the identity conditions hold approximately. Due to the transparency of the method, the user can easily check how similar a particular synthetic control unit is to the treated unit.

To implement the synthetic control estimator numerically, we need to define a distance between the synthetic controls unit and the treated unit. To do that, we combine the characteristics of the exposed unit in the $(k \times 1)$ matrix $X_1 = (U_1', \bar{Y}_1^{K_1}, \ldots, \bar{Y}_1^{K_M})'$ and the values of the same characteristics of the control units in the $(k \times J)$ matrix X_0 with the j-th row $(U_j', \bar{Y}_j^{K_1}, \ldots, \bar{Y}_j^{K_M})'$. Notice that $k = r + M$. To create the most similar synthetic control unit, the `synth()` function chooses the vector W^* to minimize a distance, $\|X_1 - X_0 W\|$, between X_1 and $X_0 W$, subject to the weight constraints. In particular, in the `synth()` function we solve for a W^* that minimizes

$$\|X_1 - X_0 W\|_V = \sqrt{(X_1 - X_0 W)' V (X_1 - X_0 W)} \tag{1}$$

where V is defined as some $(k \times k)$ symmetric and positive semidefinite matrix. The V matrix is introduced to allow different weights to the variables in X_0 and X_1 depending on their predictive power on the outcome. An optimal choice of V assigns weights that minimize the mean square error of the synthetic control estimator, that is the expectation of $(Y_1 - Y_0 W^*)'(Y_1 - Y_0 W^*)$.

The `synth()` function allows for flexibility in the choice of V. Sometimes the researcher has a good subjective assessment of the predictive power of the variables in X_1 and X_0. In this case the user could supply his own weights via the `custom.V` option. These weighs populate the diagonal of the V matrix (with the non-diagonal elements equal to zero) and `synth()` simply minimizes equation (1) conditional on the user supplied V.

Both Abadie and Gardeazabal (2003) and Abadie *et al.* (2010) propose a data-driven procedure to choose V, which is implemented by default in `synth()` (if no `custom.V` is specified). In this procedure a V^* is chosen among all positive definite and diagonal matrices such that the mean squared prediction error (MSPE) of the outcome variable is minimized over some set of pre-intervention periods.[7] In other words, let Z_1 be the $(T_P \times 1)$ vector with the values of the outcome variable for the treated unit for some set of the pre-intervention periods and Z_0 be the $(T_P \times J)$ analogous matrix for the control units, where T_P $(1 \leq T_P \leq T_0)$ is the number of pre-intervention periods over which the MSPE is minimized.[8] Then V^* is chosen to minimize

$$\arg\min_{V \in \mathcal{V}} (Z_1 - Z_0 W^*(V))'(Z_1 - Z_0 W^*(V)) \tag{2}$$

[7] Other potential choices are also described in Abadie *et al.* (2010).

[8] Notice that the rows of Z_1 and Z_0 form subsets of the rows of Y_1 and Y_0 respectively, where Y_1 refers to the $(T \times 1)$ vector of values for the outcome variable for the treated unit and Y_0 refers to the $(T \times J)$ analogous matrix for the control units. A natural choice is to set T_P equal to T_0 and thus chose V^* to minimize the MSPE over the entire pre-intervention period, but often it will be sufficient to choose $T_P < T_0$ to achieve a low MSPE over the entire pre-treatment period.

where $\mathcal{V}$ is the set of all positive definite and diagonal matrices and the weights for the synthetic control are given by W^*. `synth()` solves a nested optimization problem that minimizes equation (2), for $W^*(V)$ given by equation (1).

Abadie *et al.* (2010) describe how synthetic control methods facilitate inferential techniques akin to permutation tests that are well-suited to comparative case studies in which the number of units in the comparison group and the number of periods in the sample are relatively small. They propose inferential techniques for the synthetic control method that proceed by conducting so-called placebo studies. The basic principle is to iteratively apply the synthetic control method by randomly reassigning the intervention in time (i.e., pre-intervention dates) or across units (i.e., to control units where the intervention did not occur) to produce a set of placebo effects. Subsequently, we can compare the set of placebo effects to the effect that was estimated for the time and unit where the intervention actually occurred. This comparison is informative about the rarity of the magnitude of the treatment effect that was observed for the exposed unit. We can assess whether the effect estimated by the synthetic control method for the actual intervention is large relative to the effect estimated for a unit or date chosen at random. By construction, this exercise produces exact inference regardless of the number of available comparison units, time periods, and whether the data are individual or aggregate. However, as described in more detail in Abadie *et al.* (2010), the quality of some of the inferential exercises increases with the number of available comparison units.[9] The underlying idea of the placebo tests is thus akin to permutation inference (see, for example, Lehmann (1997)), where a test statistic is iteratively computed under random permutations of the assignment vector that determines whether a unit is in the treatment or the control group.

In section III we illustrate the placebo test proposed in Abadie *et al.* (2010) applying the synthetic control method to units that were not exposed to the treatment. Examples of placebo studies using the longitudinal dimension of the data are found in Appendix B of Abadie *et al.* (2010) and Bertrand, Duflo, and Mullainathan (2004).

3. Implementing Synth

We demonstrate the synthetic control method using data from Abadie and Gardeazabal (2003), which studied the economic effects of conflict, using the terrorist conflict in the Basque Country as a case study. Abadie and Gardeazabal (2003) used a combination of other Spanish regions to construct a synthetic Basque Country resembling many relevant economic characteristics of the Basque Country before the onset of political terrorism in the 1970s. The `basque` data contains information from 1955–1997 on 17 Spanish regions (excluding the small autonomous towns of Ceuta and Melilla on the coast of Africa), including per-capita GDP (the outcome variable), as well as population density, sectoral production, investment, and human capital (the predictor variables). Missing data are denoted by `NA`.

```
R> library("Synth")
R> data("basque")
```

[9] Notice that the attribute 'exact' refers to the fact that we can compute the exact probability of estimating an effect as large as the one we estimate for the treated unit if we reassign the treatment at random across the control units.

This dataset is organized in standard (long) panel-data format, with variables extending across the columns and the rows sorted first by region and then by time-period.[10] A name (character-string) and number is provided for each region.[11] At least one of these two types of unit-identifiers is required for **Synth** to implement the analysis below.

```
R> basque[85:89, 1:4]
```

```
   regionno regionname year   gdpcap
85        2  Andalucia 1996 5.995930
86        2  Andalucia 1997 6.300986
87        3     Aragon 1955 2.288775
88        3     Aragon 1956 2.445159
89        3     Aragon 1957 2.603399
```

In Abadie and Gardeazabal (2003) the 13 predictor variables, for each region, were:

- 1964–1969 averages for gross total investment/GDP (`invest`).
- 1964–1969 averages for the share of the working-age population that was illiterate (`school.illit`), the share with up to primary school education (`school.prim`), the with some high school (`school.med`), the share with high school (`school.high`), and the share with more than high school (`school.post.high`).[12]
- 1961–1969 averages for six industrial-sector shares as a percentage of total production (these variables are named with a `sec.` prefix).
- 1960–1969 averages for real GDP per-capita (`gdpcap`) measured in thousands of 1986 USD.
- 1969 population density measured in persons per square kilometer (`popdens`).

3.1. Using `dataprep()`

The first step is to reorganize the panel dataset into an appropriate format that is suitable for the main estimator function `synth()`. At a minimum, `synth()` requires as inputs the four data matrices X_1, X_0, Z_1, and Z_0 that are needed to construct a synthetic control unit. In our example, these four data matrices are as follows: X_1 is the (13×1) vector of Basque region predictors and X_0 is the (13×16) matrix of values of the same variables for the 16 control regions.[13] Z_1 is a (10×1) vector and Z_0 is a (10×16) matrix which contain the values for the outcome variable for the Basque country and the control units for the 10 pre-intervention periods over which we want to minimize the MSPE.

[10] The panel dataset does not have to be sorted in this standard form. If the time-periods are out of order and/or units are interspersed down the rows, `dataprep()` works correctly just the same.

[11] The first unit in this dataset refers to the data aggregated for the whole country of Spain.

[12] Notice that in the `basque` data these highest educational attainment variables are provided as the total number of people in each category (in thousands). They are transformed to percentage shares below.

[13] Note that all but one of these predictors is an average value over some range of the pre-treatment period and the precise date-range varies across predictor variables.

While the user can choose to provide preprocessed data matrices and load them into `synth()`, our package provides a convenience function called `dataprep()` that the user can run first to properly organize the data. We strongly recommend using `dataprep()`, because it allows to conveniently extract and package all the necessary inputs for `synth()` in a single list object that can be passed to `synth()` without further arguments. The list returned by `dataprep()` is also used by other convenience functions such as `synth.tables()`, `path.plot()`, and `gaps.plot()` to produce tables and figures that summarize and illustrate the results. `dataprep()` also implements a number of checks that will alert the user to missing data and inconsistencies in the extracted objects.

The code example below briefly illustrates the use of `dataprep()`. More examples and details on data extraction are available in the `dataprep()` help file. To obtain X_1 and X_0 the user is required to define the predictor variables as well as the operator (e.g., `mean`) and time-period (e.g., `1964:1969`) applied to these variables. The user must also specify the dependent variable (e.g., `gdpcap`), the variable(s) identifying unit names (e.g., `regionname`) and/or numbers (e.g., `regionno`), the variable identifying time-periods (e.g., `year`), the treated unit (e.g., region number `17` which is the Basque country), the control units (e.g., regions number `c(2:16,18)`), the time-period over which to optimize (e.g., the pre-treatment period `1960:1969`),[14] and the time-period over which outcome data should be plotted (usually before and after treatment, e.g., `1955:1997`).[15]

```
R> dataprep.out <- dataprep(
+    foo = basque,
+    predictors = c("school.illit", "school.prim", "school.med",
+      "school.high", "school.post.high", "invest"),
+    predictors.op = "mean",
+    time.predictors.prior = 1964:1969,
+    special.predictors = list(
+      list("gdpcap",                1960:1969 ,         "mean"),
+      list("sec.agriculture",       seq(1961, 1969, 2), "mean"),
+      list("sec.energy",            seq(1961, 1969, 2), "mean"),
+      list("sec.industry",          seq(1961, 1969, 2), "mean"),
+      list("sec.construction",      seq(1961, 1969, 2), "mean"),
+      list("sec.services.venta",    seq(1961, 1969, 2), "mean"),
+      list("sec.services.nonventa", seq(1961, 1969, 2), "mean"),
+      list("popdens",               1969,               "mean")),
+    dependent = "gdpcap",
+    unit.variable = "regionno",
+    unit.names.variable = "regionname",
+    time.variable = "year",
+    treatment.identifier = 17,
+    controls.identifier = c(2:16, 18),
+    time.optimize.ssr = 1960:1969,
+    time.plot = 1955:1997)
```

[14] This refers to Z_1 and Z_0 accordingly.

[15] This argument usually refers to Y_1 and Y_0, the matrices that contain the outcome data for both the pre- and post-intervention period for the treated unit and the control units. These matrices may be used to plot and summarize results.

Notice that some of the predictor information is given by `predictors`, `predictors.op`, and `time.predictors.prior`, and the rest of the information for the other predictors is specified in the `special.predictors` list. This functionality was designed to allow for easy handling of several predictors with the same operator over the same pre-treatment period (in this case, the school and investment variables) as well as additional custom (or "special") predictors with heterogeneous operators and time-periods. For example, the variables for the sector production shares (with the `sec` prefix) is only available on a biennial basis (1961,1963,...,1969) extracted via `seq(1961,1969,2)`. Averaging over the available years is easily accommodated using the `special.predictors` list. More details and examples on the use of the `special.predictors` argument are provided in the `dataprep()` help file.

`dataprep()` returns a list object `dataprep.out` that contains several elements, among them `dataprep.out$X0` and `dataprep.out$X1`, denoting X_0 and X_1 respectively. Both of these objects are easily interpreted, as variable labels have been retained. For example, here is how X_1 has been stored:

```
R> dataprep.out$X1
```

```
                                          17
school.illit                          39.888465
school.prim                         1031.742299
school.med                            90.358668
school.high                           25.727525
school.post.high                      13.479720
invest                                24.647383
special.gdpcap.1960.1969               5.285468
special.sec.agriculture.1961.1969      6.844000
special.sec.energy.1961.1969           4.106000
special.sec.industry.1961.1969        45.082000
special.sec.construction.1961.1969     6.150000
special.sec.services.venta.1961.1969  33.754000
special.sec.services.nonventa.1961.1969  4.072000
special.popdens.1969                246.889999
```

Notice how `dataprep.out` appends the associated date-range to the names of the special variables' labels. As another example, the list object `dataprep.out` also contains `dataprep.out$Z0` and `dataprep.out$Z1`, denoting Z_0 and Z_1 respectively. In our case, Z_1 (the Basque GDP per-capita for the pre-intervention period) has been stored as:

```
R> dataprep.out$Z1
```

```
           17
1960 4.285918
1961 4.574336
1962 4.898957
1963 5.197015
1964 5.338903
1965 5.465153
```

```
1966 5.545916
1967 5.614896
1968 5.852185
1969 6.081405
```

It may at times be useful to manipulate and modify X_0 and X_1 without going back to the original dataset. To demonstrate, we work with the five different education variables (`school.illit`, `school.prim`, `school.med`, `school.high`, `school.post.high`) representing the numbers, in thousands, of individuals with various levels of schooling. Abadie and Gardeazabal (2003) consolidate the two highest variables (`school.high` and `school.post.high`) to represent all those with more than high school education and use the percentage share for each predictor instead of the total number of individuals. The following code illustrates how to consolidate these variables in both X_0 and X_1 and transform the necessary values into percentage shares:

```
R> dataprep.out$X1["school.high",] <- dataprep.out$X1["school.high",] +
+    dataprep.out$X1["school.post.high",]
R> dataprep.out$X1 <- as.matrix(dataprep.out$X1[
+    -which(rownames(dataprep.out$X1) == "school.post.high"),])
R> dataprep.out$X0["school.high",] <- dataprep.out$X0["school.high",] +
+    dataprep.out$X0["school.post.high",]
R> dataprep.out$X0 <- dataprep.out$X0[
+    -which(rownames(dataprep.out$X0) == "school.post.high"),]
R> lowest <- which(rownames(dataprep.out$X0) == "school.illit")
R> highest <- which(rownames(dataprep.out$X0) == "school.high")
R> dataprep.out$X1[lowest:highest,] <-
+    (100 * dataprep.out$X1[lowest:highest,]) /
+    sum(dataprep.out$X1[lowest:highest,])
R> dataprep.out$X0[lowest:highest,] <-
+   100 * scale(dataprep.out$X0[lowest:highest,], center = FALSE,
+   scale = colSums(dataprep.out$X0[lowest:highest,]))
```

3.2. Running `synth()`

The `synth()` command searches for the W^* vector of weights that identifies the synthetic control for the Basque region by solving the nested optimization problem described in equations (1) and (2) above.

For any V `synth()` finds a $W^*(V)$ by minimizing equation (1) using a constrained quadratic optimization function from R's **kernlab** package (Karatzoglou, Smola, Hornik, and Zeileis 2004). `synth()` solves for the diagonal matrix V^* that minimizes equation (2) and thus the MSPE for the pre-intervention period. To solve the optimization given by equation (2) above, we run `optim()` (R's general-purpose optimization function).[16]

[16]Depending on the exact setup of the data there exist situations in which the objective function may contain local minima, so that (as is routinely the case in these types of problems) there is no analytical guarantee that the derivative-based algorithms routinely used by `optim()` (i.e., `Nelder-Mead` and `BFGS`) will converge to the global minimum. Notice that `synth()` offers various safeguards against this potential problem. First, by

As shown below, the `synth()` command knows how to extract its required arguments (`Z1`, `Z0`, `X1`, `X0`) from the `data.prep` list output.[17] No additional arguments are necessary, though one may pass arguments to `optim()`, `ipop()`, or `genoud()` if desired. For example, below we set `optim()` to use the BFGS quasi-Newton algorithm. After `synth()` finishes, the values of V^*'s diagonal and W^* are shown, as are the corresponding values of equation (2) and (1):

```
R> synth.out <- synth(data.prep.obj = dataprep.out, method = "BFGS")

X1, X0, Z1, Z0 all come directly from dataprep object.

****************
 searching for synthetic control unit
****************

LOSS (V): 0.08864642

LOSS (W): 0.2501998
```

The `LOSS (V)` output corresponds to the loss associated with equation (2). `LOSS (W)` is the loss associated with equation (1).

3.3. Obtaining Results: Tables, Figures, and Estimates

There are various ways to obtain and summarize results. `synth()` returns a list object that allows the user to easily access the output from the optimization. For example, the $(J \times 1)$ matrix of W^* weights is stored in `synth.out$solution.w`. The results from `synth()` can easily be combined with the output from `dataprep()` to compute other quantities of interest. For example, the annual discrepancies in the GPD trend between the Basque region and its synthetic counterpart may be calculated by typing

```
R> gaps <- dataprep.out$Y1plot - (dataprep.out$Y0plot %*% synth.out$solution.w)
R> gaps[1:3, 1]

      1955       1956       1957
0.15029816 0.09174669 0.03723351
```

default `synth()` always runs the optimization twice using two sets of starting values (equal V weights and a specialized regression based method to pick V) and returns the run that obtains lower loss. Second, the user may choose to rely on one of non-derivative based algorithms offered by `optim()` (e.g., `SANN`). Finally, `synth()` offers an additional argument called `genoud()`. If `genoud()` is set to `TRUE` then `synth()` will embark on a two-step optimization procedure. A first optimization is conducted using the `genoud()` optimizer from the **rgenoud** package that combines evolutionary algorithm methods with a derivative-based (quasi-Newton) method to solve difficult optimization problems (see Mebane, Jr. and Sekhon 2011 for details). Solutions from `genoud()` are then passed to `optim()` in the second step. This option will require more computing time, but could be used in difficult cases if one were concerned about local minima.

[17] As indicated above, instead of first running `dataprep()` the user could also choose to provide preprocessed data matrices and load them into `synth()` via the `X0`, `X1`, `Z0`, and `Z1` options. In this case, no `data.prep.obj` should be specified. The `synth()` help file contains such a stand-alone example. However, in most cases it will be more convenient for the user to first run `dataprep()` to properly organize the data.

Thus the user has the flexibility to create customized summary tables and figures. To facilitate the presentation of the results and to assess the quality of the synthetic control unit, the **Synth** package also contains three convenience functions. Tables are produced by using the `synth.tab()` function:

```
R> synth.tables <- synth.tab(dataprep.res = dataprep.out,
+    synth.res = synth.out)
```

This function produces four different types of tables:

```
R> names(synth.tables)

[1] "tab.pred" "tab.v"    "tab.w"    "tab.loss"
```

The first object (`synth.tables$tab.pred`) is a table comparing pre-treatment predictor values for the treated unit, the synthetic control unit, and all the units in the sample.

```
R> synth.tables$tab.pred[1:5, ]

             Treated Synthetic Sample Mean
school.illit   3.321     7.645      10.983
school.prim   85.893    82.286      80.911
school.med     7.522     6.964       5.427
school.high    3.264     3.105       2.679
invest        24.647    21.583      21.424
```

We see that the synthetic Basque country is fairly similar to the real Basque country. The sample means of the predictor variables over the 16 control regions are provided as a comparison. The second object is a table showing the V^*-weight corresponding to each predictor (not shown here). The third object shows the W^*-weight for each potential control unit.

```
R> synth.tables$tab.w[8:14, ]

   w-weights            unit.names unit.numbers
9      0.000    Castilla-La Mancha            9
10     0.851               Cataluna           10
11     0.000  Comunidad Valenciana           11
12     0.000           Extremadura           12
13     0.000                Galicia           13
14     0.149 Madrid (Comunidad De)           14
15     0.000    Murcia (Region de)           15
```

We see that Cataluna contributes 85 percent and Madrid contributes 15 percent to the synthetic Basque country; a zero weight is assigned to all the other control regions. The `path.plot()` function produces Figure 1, showing the trajectories of the treated unit and the synthetic control unit. To make a convincing case for a large treatment effect, we would like to see the two trajectories of the outcome variable for the treated unit and its synthetic

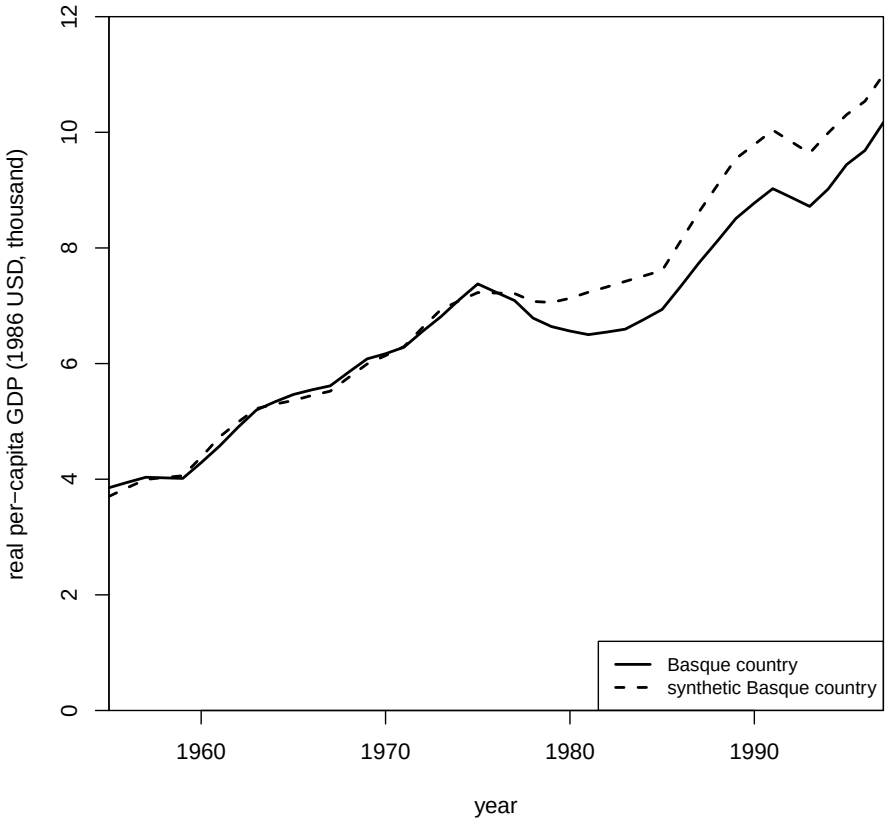

Figure 1: Trends in Per-Capita GDP: Basque country vs. synthetic Basque country.

control unit to be quite similar prior to the intervention and to diverge sharply when the intervention occurs. This is what we see for the Basque country example where the terrorism claimed its first victim in 1968, but large scale terrorist activity only began in the mid-70s.[18] Notice that the `path.plot()` function allows the user to pass many arguments to the `plot()` command to customize items like axes labels and titles.

```
R> path.plot(synth.res = synth.out, dataprep.res = dataprep.out,
+    Ylab = "real per-capita GDP (1986 USD, thousand)", Xlab = "year",
+    Ylim = c(0, 12), Legend = c("Basque country",
+    "synthetic Basque country"), Legend.position = "bottomright")
```

Instead of tracking the two trajectories over time, the `gaps.plot()` function produces Figure 2, showing how the difference between treated and synthetic control outcomes change over time. Here we see that the GDP trajectory for the Basque country is very similar to that of the synthetic Basque country for almost the entire pre-terrorism period. Once large scale terrorist activity arises in the mid 70s, however, the GDP trajectory in the Basque country

[18] See Abadie and Gardeazabal (2003, Table 1) for details.

Figure 2: Per-capita GDP Gap between Basque country and synthetic Basque country.

grows at a much lower rate than in the synthetic Basque country suggesting a large negative effect of terrorism on Basque GDP.

```
R> gaps.plot(synth.res = synth.out, dataprep.res = dataprep.out,
+     Ylab = "gap in real per-capita GDP (1986 USD, thousand)", Xlab = "year",
+     Ylim = c(-1.5, 1.5), Main = NA)
```

3.4. Placebo Tests

One benefit of synthetic control methods is that they lend themselves to placebo tests. These tests involve applying the synthetic control method after reassigning the intervention in the data to units and periods where the intervention did not occur.

Abadie and Gardeazabal (2003) introduced the placebo test for the synthetic control method by demonstrating that when the synthetic control method is applied to Catalonia, a region similar to the Basque country in terms of the variables is X_1 and X_0, and `path.plot()` is run, there is no identifiable treatment effect. As shown in Figure 3, the outcome trajectories for Catalonia and its synthetic version are very similar. To produce Figure 3,

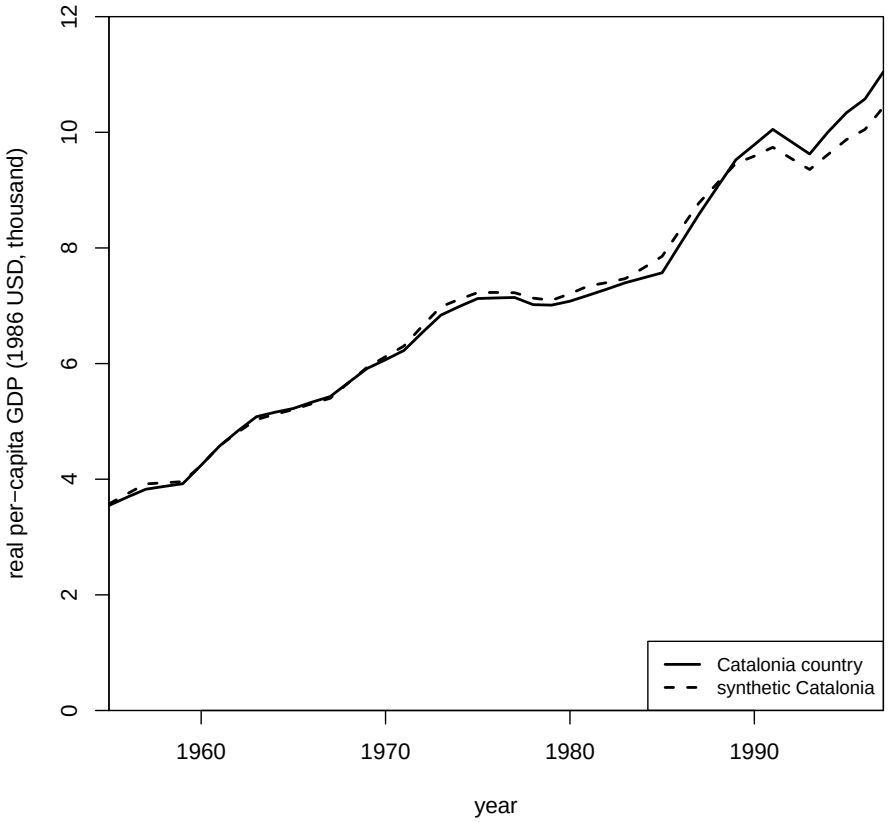

Figure 3: Placebo Study: Trends in Per-Capita GDP: Catalonia vs. synthetic Catalonia.

one must re-run `dataprep()`, this time setting the `treatment.identifier` to `10` and the `controls.identifier` to `c(2:9,11:16,18)`, since Catalonia is region number 10 in the dataset.[19]

This test is only one of several different types of placebo tests that users can run with this package. For example, one can run placebos-in-time, by using `dataprep()` to assign a treatment before the true treatment occurred and checking to ensure that the trajectories of the synthetic control and the treated unit follow the same path beyond that arbitrary point in time. One may also wish to run placebos with outcome variables that should be unaffected by the treatment.

As described in Section II above, one can perform exact inferential techniques akin to permutation tests by applying the synthetic control method to every control unit in the sample and collecting information on the gaps associated with each iteration. Then, as in Abadie *et al.* (2010), the user can plot these gaps and visually determine whether the line associated with the true synthetic control unit (e.g., the synthetic Basque region) conspicuously differentiates itself from the rest with small gaps prior to treatment and large gaps afterward. The approach

[19]The Basque country is excluded from the pools of controls.

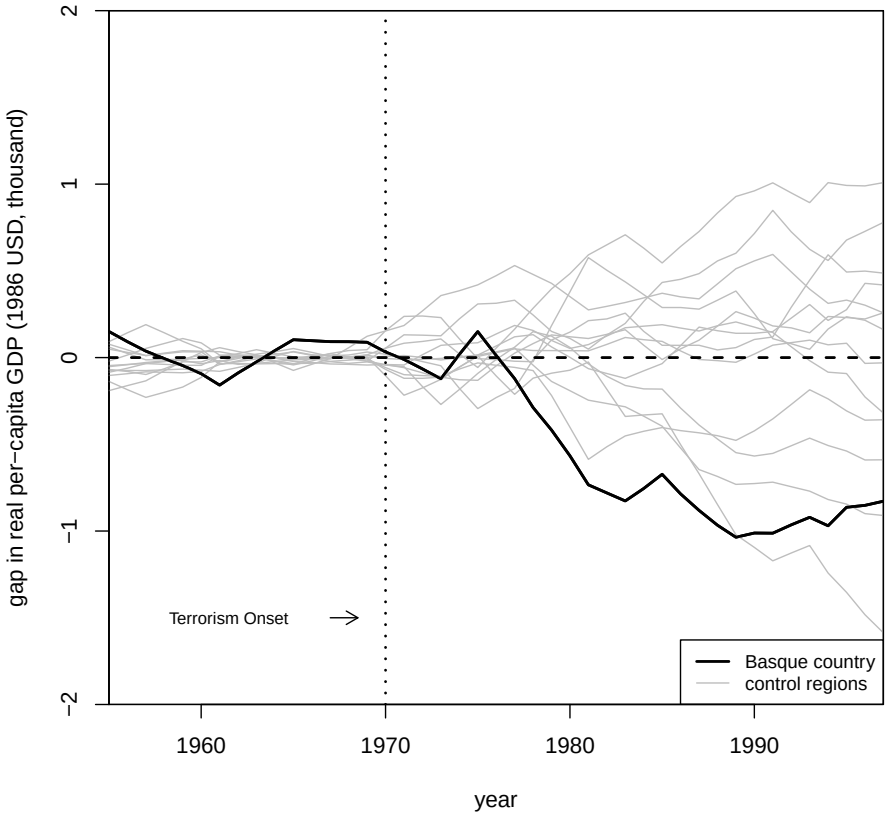

Figure 4: Permutation Test: Per-Capita GDP Gaps in the Basque country and 12 Control Regions (Discards Regions with Pre-Terrorism MSPEs More than Five Times Higher than the MSPE for the Basque country).

is easily implemented by running a `for` loop to implement placebo tests across all control units in the sample and collecting information on the gaps. Figure 4 shows the results obtained when this inferential technique is applied to our data example. As recommended in Abadie *et al.* (2010) we exclude regions with a poor fit for the pre-treatment period (i.e., regions with a MSPE that is five time higher than for the Basque country). Placebo studies for these regions do not provide information to measure the relative rarity of the post-treatment gap obtained for the Basque country which was well-fitted prior to treatment.[20] The resulting figure demonstrates that when we reassign exposure to terrorism to other regions there is a very low probability of obtaining a gap as large as the one obtained for the Basque region.

[20]Notice that the poor-fitting regions are the ones that are very unusual in their pre-treatment characteristics (for example the Balearic Islands or Madrid) so no combination of other regions in the sample can reproduce the pre-treatment trends for these regions. As explained in Abadie *et al.* (2010), one alternative to excluding regions based on MPSE is to compute the distribution of the ratio of post- to pre-treatment MPSE.

4. Conclusion

We have described synthetic control methods and the way they may be implemented to estimate causal effects and perform exact inferential techniques using the **Synth** package for R. Several extensions to **Synth** are currently under active development. We are testing a regression-based method for populating the entire V matrix (not just the diagonal) and a version of **Synth** that selects the W weights that best fit multiple outcome variables simultaneously. We are also working on a version that accommodates multiple treatments phased-in over time.

Acknowledgments

We would like to thank Luke Keele, Micah Altman, Simon Jackman, the anonymous reviewers, and the editors John Fox and Achim Zeileis for helpful comments. Funding for this research was generously provided by NSF grant SES-0350645 (Abadie).

References

Abadie A, Diamond A, Hainmueller J (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.

Abadie A, Gardeazabal J (2003). "The Economic Costs of Conflict: A Case Study of the Basque Country." *American Economic Review*, **93**(1), 112–132.

Bertrand M, Duflo E, Mullainathan S (2004). "How Much Should We Trust Differences-in-Differences Estimates?" *Quarterly Journal of Economics*, **119**(1), 249–275.

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004). "**kernlab** – An S4 Package for Kernel Methods in R." *Journal of Statistical Software*, **11**(9), 1–20. URL http://www.jstatsoft.org/v11/i09/.

Lehmann EL (1997). *Testing Statistical Hypotheses*. 2nd edition. University of California Press, Berkeley.

Mebane, Jr WR, Sekhon JS (2011). "Genetic Optimization Using Derivatives: The **rgenoud** Package for R." *Journal of Statistical Software*, **42**(11), 1–26. URL http://www.jstatsoft.org/v42/i11/.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

StataCorp (2007). *Stata Statistical Software: Release 10*. StataCorp LP, College Station, TX. URL http://www.stata.com/.

The MathWorks, Inc (2007). *MATLAB – The Language of Technical Computing, Version 7.5*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Affiliation:

Jens Hainmueller
Department of Political Science
Massachusetts Institute of Technology
Cambridge, MA 02139, United States of America
E-mail: jhainm@mit.edu
URL: http://http://www.mit.edu/~jhainm/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 42, Issue 13 *Submitted:* 2007-02-01
June 2011 *Accepted:* 2007-12-17