

# EPPE Term Paper\*

Ricardo Duque Gabriel

**Version 1.0**

February 15, 2019

## **Abstract**

This is a term project for the Effective Programming Practices for Economists course at University of Bonn. In this project, I try to improve the Python `synth` package created by Guilherme Nazareth. I add flexibility to the package by allowing a manual imputation of an initial guess for the weights used to compute the synthetic control group allowing for a more customized usage. Then, I provide two minimum working examples using a sample of 32 OECD countries. The first one studies the impact of the Japanese financial crisis which started in 1997. The second one follows Born et al. (2018) and tries to measure the impact of the Brexit vote in the United Kingdom economy. Finally, it is worth noticing that this project uses the template from von Gaudecker (2019) and can be easily replicated.

**JEL classification:** E65, F42, G01, P16

**Keywords:** Synthetic Control Method, Japan, Financial Crisis, United Kingdom, Brexit

---

\*Ricardo Duque Gabriel, University of Bonn. Email: `ricardo[dot]gabriel[at]uni-bonn[dot]de`.

# 1 Introduction to the Synthetic Control Method

This term paper is related to the literature strand focused on the use of the synthetic control method (SCM) for quantitative inference in small-sample comparative studies which has been considered to be the most important innovation in the policy evaluation literature in the last 15 years by Athey & Imbens (2017). Abadie & Gardeazabal (2003) were the first ones to use this method by assessing the impact of terrorism on the economic growth of the Basque Country in Spain. A simple comparison of the economic performance of the Basque country and the rest of Spain during the terrorism years would not reflect the impact of terrorism. Thus, the authors created a doppelganger (named synthetic Basque country) as the weighted average of other regions of Spain which could resemble most characteristics of the Basque Country but did not experience the terrorism.

Other relevant papers using and developing the SCM are the ones from Abadie et al. (2010) who use the same approach to evaluate the effect of a large-scale tobacco control program that California implemented in 1988 by using other states from the U.S. to construct a synthetic California. Abadie et al. (2015) who quantify the impact of the 1990 German reunification on West Germany and Born et al. (2018) who compute the economic costs of the Brexit vote. These papers use a donor pool composed of countries from all over the world to construct the doppelgangers.

The dynamic effect of a policy intervention at a given time in a country can be expressed by:

$$\tau_{it} = Y_{it}^I - Y_{it}^C$$

where  $Y_{it}^I$  represents the outcome of country  $i$  in period  $t$  and  $Y_{it}^C$  is the outcome country  $i$  would have had in period  $t$  absent from the policy intervention and it can not be observed.

To evaluate whether such policy intervention had any impact on the economy of a country one must have a counterfactual, i.e., what this country's economic activity would have been if it had not have such intervention. This comparison unit is of crucial importance for a correct

policy evaluation as represented by the equation above. Ideally, the counterfactual should be identical to the treated unit in all dimensions except for the treatment assignment. When the treated unit is a country, this “ideal” control unit rarely exists in observed data as countries differ widely across demographic, economic, legislative, among other characteristics Born et al. (2018). Thus, we can not directly use countries which were not treated as a counterfactual because they do not truly represent what would have happened to the treated unit if it had not been assigned into treatment.

Let  $d$  be the number of available control countries (constituting our “donor pool”), suppose that we have  $d + 1$  countries and, without loss of generality, consider that only the last one  $i = d + 1$  is exposed to the intervention of interest. Abadie et al. (2010) show that an unbiased estimate for the counterfactual outcome can be obtained by assigning a weight  $w$  to each control country, i.e.  $\sum_{j=1}^d w_j^* Y_{jt} = \hat{Y}_{it}^C$  for  $t \geq T_0$ . Therefore, an unbiased estimate of the previous equation’s  $\tau_{it}$  can be computed by:

$$\hat{\tau}_{it} = Y_{it}^I - \sum_{i=1}^d w_i^* Y_{it}$$

The weights assigned to the countries in the donor pool are chosen to minimize the preintervention differences between the treated unit and the doppelganger in terms of the predictors of the outcome variable. Formally, let  $X_1$  denote the matrix of observations for the predictor variables in the treated unit and let  $X_0$  denote a matrix with observations for the donor countries’ predictor variables. Finally, let  $W$  denote the  $(d \times 1)$  vector of nonnegative weights  $w_j$ ,  $j = 1, \dots, d$  which sum to one. Then, the doppelganger is defined by  $W^*$  which minimizes the following mean squared error:

$$(X_1 - X_0 W)' V (X_1 - X_0 W)$$

subject to  $w_j \geq 0$  for  $j = 1, \dots, d$  and  $\sum_{j=1}^d w_j = 1$ . In this expression,  $V$  is a  $(d \times d)$  diagonal matrix. As Abadie & Gardeazabal (2003), one can choose  $V$  such that the outcome variable path for the treated country during the pre-treatment period is best reproduced by

the resulting doppelganger.  $V^*$  minimizes:

$$(Z_1 - Z_0W)'V(Z_1 - Z_0W)$$

in which  $Z_0$  is the matrix of outcomes for control units,  $Z_1$  is the matrix of outcomes for the treated unit. So,  $W^*(V)$  is the optimal combination of weights given the matrix  $V$ .

On the one hand, this approach has several advantages. Firstly, it is a transparent method with the relative contribution of each control unit to the counterfactual of interest being explicit by the used weights to construct the synthetic control group. The similarities between the unit affected by the event and the synthetic control are also clearly observed in terms of preintervention outcomes. Secondly, SCM is safeguarded against extrapolation as the weights are restricted to be positive and sum to one (Abadie et al. (2010)). This avoids the estimation of “extreme counterfactuals” that fall far outside the convex hull of the data. Yet, interpolation bias might arise if the potential control units have not been selected appropriately and present characteristics that are far from the treated unit.

On the other hand, SCM requires some assumptions to be made (Abadie et al. (2010)). The first one is that the treatment is exclusively assigned to the treated country and thus, that there are no spillover effects. Secondly, since the weights are constructed according to the preintervention characteristics, one has to assume that there are no differentiated shocks in the post treatment period in the countries which constitute the donor pool. Third, the intervention has no effect on the outcome before the implementation period. Otherwise, one would have to redefine the date of treatment to an earlier period.

## 2 Synth Package

This term paper has the goal to improve an already existent Python package and apply it to two interesting case studies. I start by using the already existent synth package created by Guilherme Nazareth de Souza.

In his implementation, Synth uses a nested optimization with SciPy quadratic programming minimization functions `optimize.minimize` and `optimize.fmin_slsqp`. To achieve this, he calls the function `get_v_1`, which finds the variable `v` that minimizes the function `get_v_0`, which in turn calls the function `get_w` and returns the residual sum of squares associated with given values of `v` and `w`. This last function, `get_w`, finds the variable `w` that minimizes the rss associated with `v` and any values of `w`.

---

```
def w_rss(w, v, x0, x1):
    k = len(x1)
    importance = np.zeros((k,k))
    np.fill_diagonal(importance, v)
    predictions = np.dot(x0,w)
    errors = x1 - predictions
    weighted_errors = np.dot(importance, errors)
    weighted_rss = sum(weighted_errors**2)[0]
    return weighted_rss

def w_constraint(w, v, x0, x1):
    return np.sum(w) - 1

def v_rss(w, z0, z1):
    predictions = np.dot(z0,w)
    errors = z1 - predictions
    rss = sum(errors ** 2)[0]
    return rss

def get_w(w, v, x0, x1):
    weights = fmin_slsqp(w_rss, w, f_eqcons=w_constraint,
                        bounds=[(0.0, 1.0)]*len(w),
                        args=(v, x0, x1), disp=False, full_output=True)[0]
    return weights
```

```

def get_v_0(v, w, x0, x1, z0, z1):
    weights = get_w(w,v,x0,x1)
    rss = v_rss(weights, z0, z1)
    return rss

def get_v_1(v, w, x0, x1, z0, z1):
    result = fmin_slsqp(get_v_0, v, args=(w, x0, x1, z0, z1),
                        bounds=[(0.0, 1.0)]*len(v))
    return result

def get_estimate(x0, x1, z0, z1, y0, w):
    (k,j) = x0.shape
    v = [1.0]*k
    predictors = get_v_1(v, w, x0, x1, z0, z1)
    controls = np.array(get_w(w, predictors, x0, x1)).transpose()
    estimates = np.dot(y0,controls)
    return estimates, predictors, controls

```

---

Upon finding the optimal vector  $W^*$  (and, by definition,  $V$  as well), synth multiplies the  $Z_0$  by  $W$  to find an estimate (or a counterfactual) of the  $Z_1$  vector. Synth can plot both the estimate  $Z_1$  and the actual  $Z_1$ , so the researcher can compare how well the model represents the actual data:

---

```

def synth_tables(foo,
                predictors,
                treated_unit,
                control_units,
                index_variable,
                measured_variable,
                Weights,
                time_variable,

```

```

        predict_time,
        optimize_time,
        plot_time,
        function="mean"):

control_units.sort()
predictors.sort()
plot_time.sort()

X0, X1, Y0, Y1, Z0, Z1 = dataprep(foo,
        predictors,
        treated_unit,
        control_units,
        index_variable,
        measured_variable,
        Weights,
        time_variable,
        predict_time,
        optimize_time,
        plot_time,
        function="mean")

(est, predict, ctrl) = get_estimate(X0, X1, Z0, Z1, Y0, Weights)
predict = [ round(elem,6) for elem in predict ]
ctrl = [ round(elem,6) for elem in ctrl ]
estimated_predictors = np.dot(X0,ctrl).transpose()
predictors_table = pd.DataFrame({'Synthetic':estimated_predictors,
                                'Actual': X1.transpose()[0]}, index=predictors)
estimated_outcomes = np.dot(Y0,ctrl)
outcomes_table = pd.DataFrame({'Synthetic':estimated_outcomes,
                                'Actual':Y1.transpose()[0]}, index=plot_time)

```

```

predictors_weights = pd.DataFrame({'Weight':predict}, index=predictors)
controls_weights = pd.DataFrame({'Weight':ctrls}, index=control_units)

print ("Predictors Table")
print ("——")
print (predictors_table)
print (" ")
print ("Outcomes Table")
print ("——")
print (outcomes_table)
print (" ")
print ("Predictors' Weights")
print ("——")
print (predictors_weights)
print (" ")
print ("Controls' Weights")
print ("——")
print (controls_weights)

estimates = estimated_outcomes
actual_values = Y1.transpose()[0]
plt.plot(range(len(estimates)),estimates, 'r—',
         label="Synthetic Control")
plt.plot(range(len(estimates)),actual_values, 'b—',
         label="Actual Data")
plt.axvline(x=predict_time[-1])
plt.title("Synthetic Control Model")
plt.ylabel(measured_variable)
plt.xlabel(time_variable)
plt.legend(loc='upper left')
plt.savefig('Synthetic_Control_Method.png')

```



```
plt.show()
```

```
return
```

---

I add some flexibility to the package by allowing a manual imputation of an initial guess for the weights used to compute the synthetic control group (henceforth, doppelganger). The common practice is to start with equal weights for all countries nevertheless, one might have a good idea of what the weights are, especially if one is doing a replication work just as I will do later on. In order to have a correct implementation of this new feature I also add some tests to the package:

---

```
### Weights
```

```
if not type(Weights) in [np.ndarray]:  
    raise NameError("Weights should be a numpy.ndarray")  
if not Weights.dtype in [int, float]:  
    raise NameError("Weights contain non-numerical data")  
if not len(Weights)==len(control_units):  
    raise NameError("Weights should have same length than control_units")
```

---

### 3 Case Studies

#### 3.1 The Japanese 1997 financial crisis

Table 3.1 displays the weights of each control country in the synthetic Japan. The weights reported in it indicate that the Japanese real per capita GDP is best reproduced by a combination of Australia, Switzerland, and The Netherlands. All other countries in the donor pool are assigned zero W-weights.

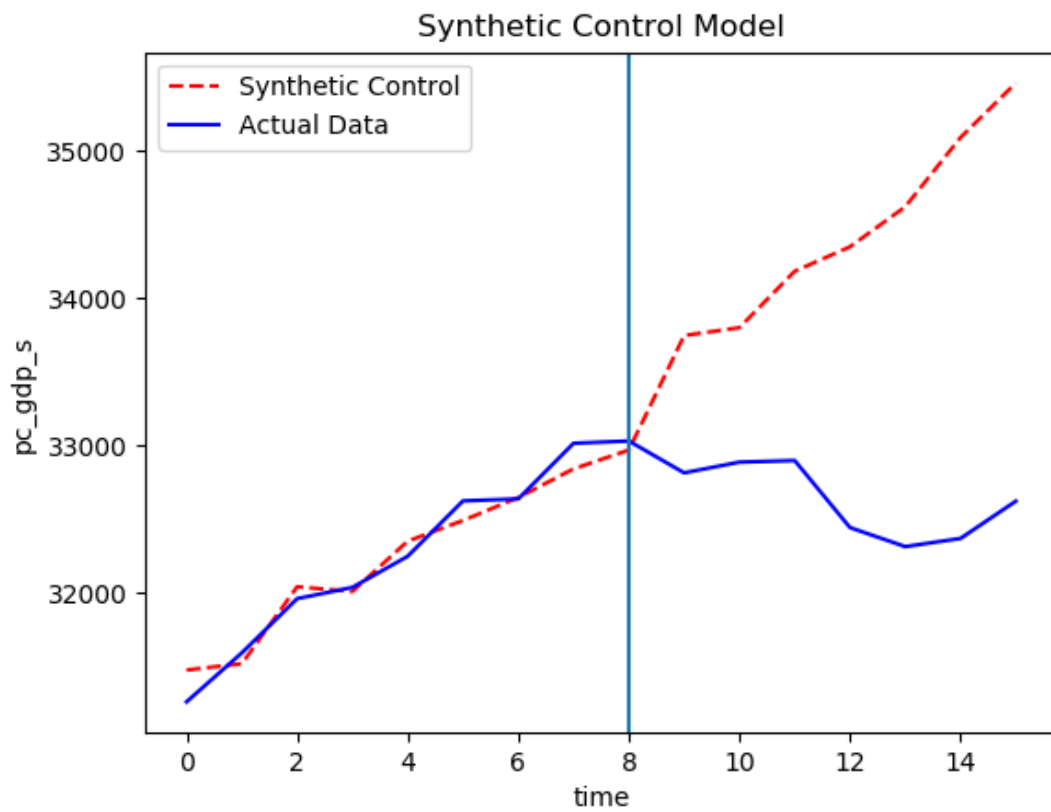
---

Country	Weight	Country	Weight	Country	Weight	Country	Weight
AUS	0.825	DNK	0.0	ISL	0.0	NLD	0.094
AUT	0.0	ESP	0.0	ISR	0.0	NOR	0.0
BEL	0.0	EST	0.0	ITA	0.0	NZL	0.0
CAN	0.0	FIN	0.0	KOR	0.0	POL	0.0
CHE	0.081	FRA	0.0	LTU	0.0	PRT	0.0
CHL	0.0	GBR	0.0	LUX	0.0	SVK	0.0
CZE	0.0	HUN	0.0	LVA	0.0	SWE	0.0
DEU	0.0	IRL	0.0	MEX	0.0		

---

Figure 1 displays real per capita GDP for Japan and its doppelganger during the period between 1995q1 to 1998q4. From this figure one can conclude that there is some evidence pointing out to a loss of about 2000 euros per capita.

Figure 1: Synthetic Control Method for the Japanese real per capita GDP



### 3.2 The output effect of the Brexit vote

In this subsection, I follow closely the work by Born et al. (2018) to measure the immediate output effect of the Brexit vote. In order to do that I contrast the output performance in the UK and in the doppelganger economy in the quarters following the referendum shock. First, I start by computing the doppelganger for the UK using the synth package.

Table 3.2 displays the weights of each control country in the synthetic UK. The weights reported in it indicate that UK's real per capita GDP is best reproduced by a combination of Canada, Hungary, Italy, Japan, Korea, Norway, Portugal and USA. All other countries in the donor pool are assigned zero W-weights.

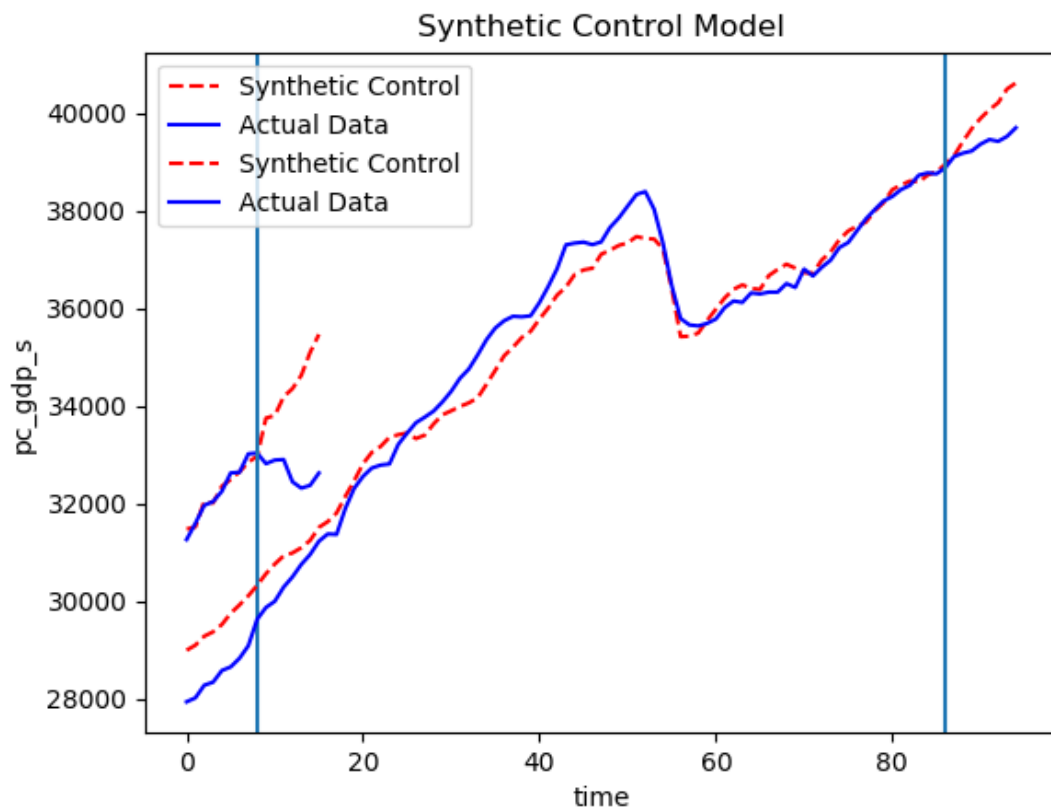
---

Country	Weight	Country	Weight	Country	Weight	Country	Weight
AUS	0.0	DNK	0.0	ISR	0.0	NLD	0.0
AUT	0.0	ESP	0.0	ITA	0.04	NOR	0.04
BEL	0.0	EST	0.0	JPN	0.24	NZL	0.0
CAN	0.16	FIN	0.0	KOR	0.03	POL	0.0
CHE	0.0	FRA	0.0	LTU	0.0	PRT	0.06
CHL	0.0	HUN	0.21	LUX	0.0	SVK	0.0
CZE	0.0	IRL	0.0	LVA	0.0	SWE	0.0
DEU	0.0	ISL	0.0	MEX	0.0	USA	0.22

---

Figure 2 displays real per capita GDP for the UK and its doppelganger during the period between 1995q1 to 2018q3. From this figure one can conclude that there is some evidence pointing out to a loss of about 1000 euros per capita.

Figure 2: Synthetic Control Method for the British real per capita GDP as in Born2018



## 4 Conclusion

In this term paper, I improved an already existent python package about the synthetic control method. Then, using a sample of 32 OECD countries I perform two case studies. The first one studies the impact of the Japanese financial crisis which started in 1997. Here, I found that the financial crisis had a very negative impact on the Japanese economy - the real per capita GDP was 2000 lower than if the country had never gone through such episode.

The second one follows Born et al. (2018) and tries to measure the impact of the Brexit vote in the United Kingdom economy. Here, I also found a negative effect - the real per capita GDP was 1000 euros lower than if the Brexit vote had never occurred.

Finally, I adjusted this project such that it uses the template from von Gaudecker (2019) which can be easily replicated.

## 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., Diamond, A., & Hainmueller, J. (2015). Comparative politics and the synthetic control method. *American Journal of Political Science*, 59(2), 495–510.
- Abadie, A., & Gardeazabal, J. (2003). The economic costs of conflict: A case study of the basque country. *American economic review*, 93(1), 113–132.
- Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, 31(2), 3–32.
- Born, B., Muller, G., Schularick, M., & Sedláček, P. (2018). *The costs of economic nationalism: evidence from the brexit experiment* (Tech. Rep.). CEPR Discussion Paper.
- von Gaudecker, H.-M. (2019). *Templates for reproducible research projects in economics*. (<https://doi.org/10.5281/zenodo.2533241>)