

Reproduction Code for “Randomization Tests in Observational Studies with Staggered Adoption of Treatment”

August 25, 2021

Contact

For any questions please email `panos.toulis@chicagobooth.edu`.

File structure

The file directory structure is as follows:

- `README.pdf`. This file. Contains instructions on how to use the code.
- `/Section 5- Simulation study`. Used to reproduce all Tables in Section 4 of the paper. Specifically, it contains the following scripts:
 - * `Section_5.R`. Main script file to reproduce Tables 1-4. See below.
 - * `functions.R`. Helper functions.
 - * `checks.R`. Unit tests and other checks.
- `/Section 6- Abadie-replicate`.
 - * `synth_fisher.R`. Main script.
 - * `synth_fisher_DATA.R`. Script to generate datasets to use in main script.
 - * `smoking_with_dems.rdata`. Dataset that contains state information, including information about legislature.
 - * `RTtobacco.RData`. Smoking data.
 - * `SynthFisher_Results.rda`. File with pre-computed results — can be reproduced by `full_DATA_analysis()` function in main script.
 - * `tax_hike_adoption_data.txt`. Text file with some notes on defining treatment adoption from various states.
 - * `tax_burden_2014.pdf`. Contains comprehensive information about tobacco tax legislation, which we used to define treatment adoption times.
- * `code_original.zip`. The original code that we used in our paper. This package contains parallelization code used in the Mercury cluster at Booth, but is not portable. Please use the above scripts for quick replication on a typical personal computer.

Reproducing Tables 1-4

To reproduce the Tables in Section 5, use the script `Section_5.R` and the function `Reproduce_Table()`. The function has the flag `make_it_quick` which can be used to quickly, but approximately, reproduce a table, without running the full simulation. The argument `nreps` denotes how many randomization tests to execute in order to calculate the rejection rates. Higher values lead to higher precision but also more computation time. When `make_it_quick = TRUE` then the argument `nreps` is upper-bounded at 500 to speed up computation.

For instance, to quickly reproduce Table 1 of the paper, run:

```
> source("Section_5.R")
> tab = Reproduce_Table(1, nreps=500, make_it_quick=TRUE)
[1] ">> Generating table 1 with 500 randomization reps. Quick Mode? TRUE"
[1] ">.. Running simulation setting 1 / 3 ...<"
[1] ">>> gamma = 0 (model param.), n = 25 (#units) , tau = 0 (treatment effect)"
[1] "rtest(): Iter = 303/1000 -- Misspecified Model? FALSE"
... (output omitted) ...
...
> tab
```

	g	n	tau	unif	phat	oracle
[1,]	0.0	25	0	5.2	5.1	5.5
[2,]	0.5	25	0	14.6	4.3	4.3
[3,]	2.0	25	0	19.2	4.5	5.6

As in the paper (Section 5.1, Table 1) we see that the feasible (`phat`) and the infeasible test (`oracle`) produce valid tests at the 5% level across all values of γ , whereas the `uniform` test does not. To increase precision and get even closer to the nominal level, we can adjust the `nreps` to a larger value (e.g., 10,000) and set `make_it_quick = FALSE`. In the paper, we run 100,000 replications using parallel computation in a computing cluster.

To reproduce Table 3 on the misspecified model of Section 5.2, run

```
> tab = Reproduce_Table(3, 500, T)
[1] ">> Generating table 3 with 500 randomization reps. Quick Mode? TRUE"
[1] ">.. Running simulation setting 1 / 8 ...<"
[1] ">>> g = 0 (model param.), n = 25 (#units) ,k1, k2 = 0 0"
[1] ">>> Using misspecified adoption model to sample treatment times."
[1] ... (output omitted) ...
...
> tab
```

	g	k1	k2	unif	phat	oracle
[1,]	0	0	0	4.0	3.8	3.4
[2,]	5	0	0	6.0	5.2	6.0
[3,]	0	2	0	4.8	4.6	6.2
[4,]	5	2	0	14.6	5.2	6.2
[5,]	0	0	2	5.6	3.8	5.0
[6,]	5	0	2	13.2	7.6	5.4
[7,]	0	2	2	6.0	5.6	6.6
[8,]	5	2	2	13.6	13.2	5.0

We see that these results roughly agree with Table 3 of the paper. Table 4 can be reproduced accordingly.

Reproducing Section 6

Section 6 revisits the application of Abadie et al (2010) on tobacco legislation. The main functionality is in `synth_fisher.R`:

```
> source("synth_fisher.R")
[1] "Found 512 total specifications..."
[1] "Different states"
      state   when
2   Arkansas 06/2003
...
[1] "##### DONE WITH DATA LOAD   #####"
```

To reproduce Figure 1 of the paper, simply run:

```
> Make_Fig1()
[1] "[INFO] -- Legislation times defined."
```

To reproduce Table 5 with the estimated propensity scores, look at the code in function `paper_analysis()` of the main script. Alternative, we can run:

```
> out = synth_fisher(adoption_Data_2, vars=c("lnincome", "retprice"), verbose=T)
...
Call:
coxph(formula = f, data = adoption)

      n= 39, number of events= 38

              coef exp(coef)    se(coef)      z    Pr(>|z|)
lnincome 0.41196937 1.50978818 1.17452415 0.35075    0.72577
retprice 0.04718159 1.04831235 0.01037266 4.54865 5.3991e-06 ***
....
> out
```

	State	PS	State	PS	State	PS
[1,]	"Nevada"	"0.1037"	"Arkansas"	"0.0250"	"Virginia"	"0.0086"
[2,]	"Connecticut"	"0.1018"	"Pennsylvania"	"0.0225"	"Alabama"	"0.0081"
[3,]	"Rhode Island"	"0.0681"	"Louisiana"	"0.0201"	"West Virginia"	"0.0066"
[4,]	"North Dakota"	"0.0617"	"Ohio"	"0.0200"	"Oklahoma"	"0.0037"
[5,]	"Maine"	"0.0605"	"Delaware"	"0.0186"	"South Carolina"	"0.0033"
[6,]	"Illinois"	"0.0580"	"Minnesota"	"0.0171"	"South Dakota"	"0.0027"
[7,]	"Wisconsin"	"0.0517"	"Tennessee"	"0.0163"	"Vermont"	"0.0026"
[8,]	"Texas"	"0.0491"	"Montana"	"0.0151"	"Utah"	"0.0023"
[9,]	"Nebraska"	"0.0460"	"Idaho"	"0.0139"	"Iowa"	"0.0018"
[10,]	"California"	"0.0440"	"Indiana"	"0.0134"	"North Carolina"	"0.0016"
[11,]	"New Hampshire"	"0.0360"	"Kansas"	"0.0124"	"Missouri"	"0.0014"
[12,]	"Wyoming"	"0.0291"	"Georgia"	"0.0124"	"Kentucky"	"0.0013"
[13,]	"New Mexico"	"0.0279"	"Colorado"	"0.0111"	"Mississippi"	"0.0006"

To reproduce the sensitivity analysis (Figure 2), run:

```
> paper_appendix()
```

This is using the pre-computed results in `SynthFisher_Results.rda`. To re-compute those results, run `full_DATA_analysis()`. This is time-consuming, however, but can be parallelized.