Synthetic Control Method

Prof. Tzu-Ting Yang
楊子霆

Institute of Economics, Academia Sinica
中央研究院經濟研究所

May 27, 2026

Main Idea

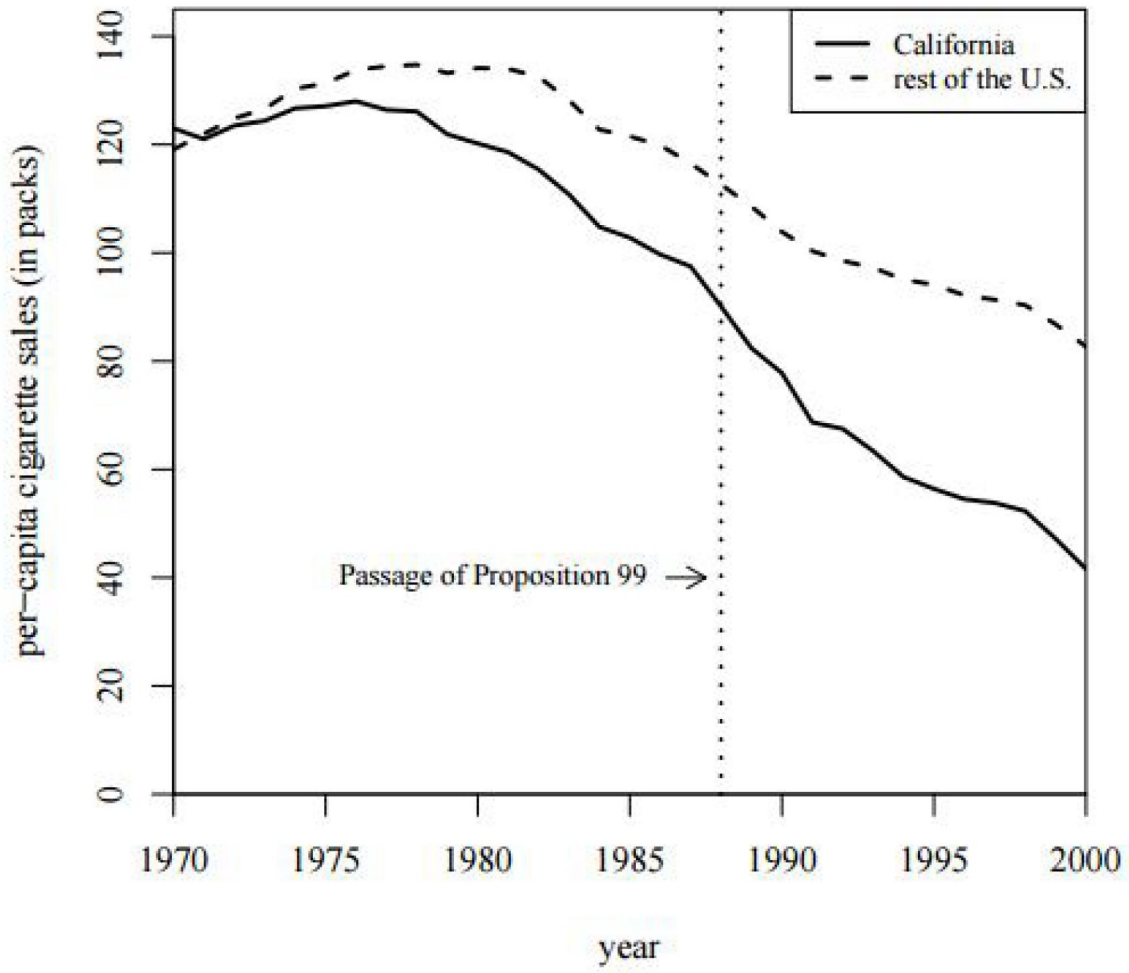
Synthetic Control Method

Main Idea

- Synthetic Control (SC) is a method to evaluate the causal effect of treatment.
  - Use (long) panel data to build the **weighted average of non-treated units**
    - The **weighted average of non-treated units** is the **synthetic unit**
    - Synthetic unit can best reproduce characteristics of the **treated unit** over time in pre-treatment period
  - Causal effect of treatment can be quantified by a simple difference in the post-treatment period:
    - **treated unit vs synthetic unit**

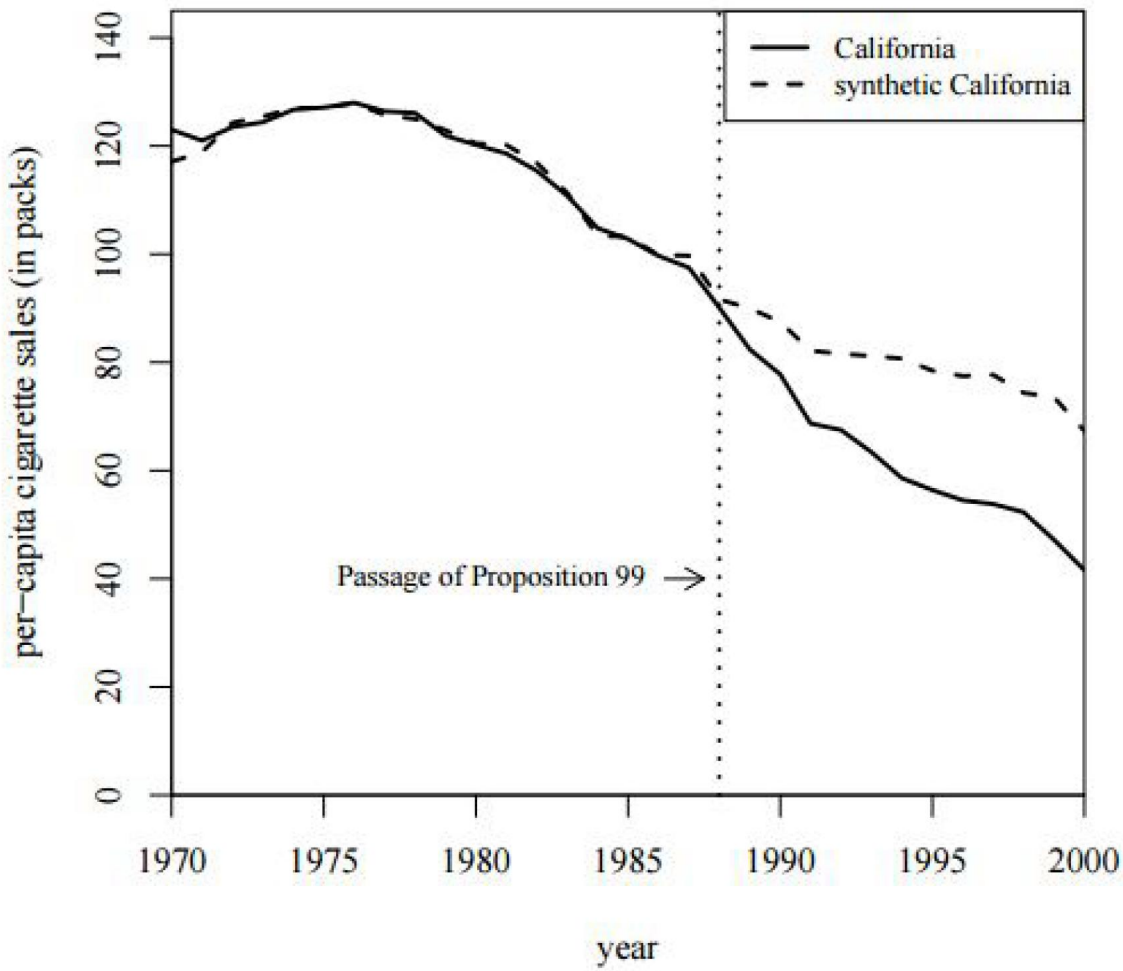
Synthetic Control Method

Graphical Representation



Synthetic Control Method

Graphical Representation



SC v.s. DID

- Treatment Units:
  - DiD: Naturally accommodates multiple treated units
  - SCM: Originally designed for a single treated unit
- Control Group Construction:
  - DID: Uses all untreated units with equal weights and require parallel trends assumption
  - SCM: Constructs a synthetic control through data-driven weighted average

SC v.s. DID

- Data Requirements:
  - DID: Works with limited pre-treatment data
  - SCM: Requires extended pre-treatment periods to construct reliable counterfactual
- Inference Approach:
  - DID: Typically relies on asymptotic statistical inference
  - SCM: Uses placebo tests and permutation-based inference

Identification

Basic Setup: Single Treated Model

Example

Alberto Abadie, Alexis Diamond, and Jens Hainmueller (2010), "**Synthetic control methods for comparative case studies: Estimating the effect of California's Tobacco Control Program**", Journal of the American Statistical Association

- This is one of the first papers using SC method
  - Use this example to go through the key concepts of SC method
  - Apply SC method to study the effects of Proposition 99, a large-scale tobacco control program that California implemented in 1988
  - A single treated unit with multiple non-treated units

Basic Setup: Single Treated Model

Example

- In 1988, California passed comprehensive tobacco control legislation:
  - Increased cigarette taxes by \$0.25 per pack
  - Funded anti-smoking media campaigns
  - Spurred clean-air ordinances
- They want to estimate the causal effect of the policy on cigarette consumption in California

Basic Setup: Single Treated Model

- Suppose we observe $J+1$ units over $t = 1, ..., T$ periods
- A "treatment" occurs at period $T_0 + 1$
  - Unit 1 being treated
  - Units $\{2, ..., J+1\}$ being unaffected
  - Pre-treatment period: $1, \ldots, T_0$
  - Post-treatment period: $T_0 + 1, \ldots, T$
- We aim to identify the causal effect of the treatment on unit 1
  - Individual treatment effect for unit 1

Potential Outcomes Framework

- Treatment
  - $D_{it} = 1$: the units that are treated from periods $T_0 + 1$ until T
  - $D_{it} = 0$: the units that are always untreated

Potential Outcomes Framework

- Potential Outcomes
  - Y_{it}^1: the potential outcome we *would* observe for unit i at time t if unit i receives the treatment
    - Note that the treated unit would receive treatment from periods $T_0 + 1$ until T
  - Y_{it}^0: the potential outcome we *would* observe for unit i at time t if unit i does not receive the treatment
- Note that **unit in synthetic control method is usually aggregate level**: country, state, county, or region

Potential Outcomes Framework

- Observed Outcomes
  - Y_{it} is the observed outcome for unit i at time t
    - Observed outcomes before period $T_0 + 1$:

$$Y_{it} = \mathrm{Y}_{it}^{0}$$

    - Observed outcomes after period $T_0 + 1$:

$$Y_{it} = \mathrm{Y}_{it}^{0}(1 - D_{it}) + \mathrm{Y}_{it}^{1} D_{it}$$

Potential Outcomes Framework

- Since only unit 1 is treated, we aim to estimate the causal effect of treatment over time $(T_0 + 1, ..., T)$ for the treated unit 1

$$\alpha_{1t} = (\alpha_{1T_0+1}, ..., \alpha_{1T})$$

where for $t > T_0$:

$$\alpha_{1t} = \underbrace{Y^1_{1t}}_{\text{observed}} - \underbrace{Y^0_{1t}}_{\text{counterfactual}}$$

- We need to construct the **unobserved counterfactual** for unit 1

Estimation

SC Estimation

$$\alpha_{1t} = \underbrace{Y^1_{1t}}_{\text{observed}} - \underbrace{Y^0_{1t}}_{\text{counterfactual}}$$

- SC method suggests treatment effect can be estimated by the simple difference:

$$\hat{\alpha}_{1t} = Y_{1t} - \sum_{i=2}^{J+1} w_i^* Y_{it}$$

SC Estimation

- Choose weights $W = (w_2^*, ..., w_{J+1}^*) \in [0,1]$ to minimize difference in pre-treatment characteristics X between treated and weighted average of non-treated units

$$W^* = \arg\min_W \sum_{k=1}^{K} v_k \left(X_{1k} - \sum_{i=2}^{J+1} w_i X_{ik} \right)^2$$

$$\text{subject to } \sum_{i=2}^{J+1} w_i = 1, \; w_i \geq 0 \; \forall \, i \in \{2, \ldots, J+1\}$$

  - X_{1k} is the k-th predictor variable for the treated unit
  - X_{ik} is the k-th predictor variable for non-treated unit i
  - v_k represents relative importance of each predictor variable (chosen by cross-validation)

SC Estimation

Choosing v_k: Intuition

- v_k controls how much weight predictor k receives when measuring "closeness" between the treated unit and the synthetic unit
  - A **larger** $v_k \Rightarrow$ predictor k must be matched more precisely
  - A **smaller** $v_k \Rightarrow$ predictor k matters less in constructing the synthetic unit
- **Key question:** Which predictors are most informative for reproducing the pre-treatment outcome trajectory?
  - We want v_k to be **large** for predictors that help fit pre-treatment Y_{1t} well

SC Estimation

Choosing v_k: Cross-Validation

- Split the pre-treatment period into two sub-periods:
  - **Training period:** $1, \ldots, T_1$ ("in-sample")
  - **Validation period:** $T_1+1, \ldots, T_0$ ("out-of-sample")
- **Nested (bilevel) optimization:**
  - **Inner problem** (given V): find $W^*(V)$ that minimizes predictor mismatch *in the training period*
  - **Outer problem**: choose V^* to minimize out-of-sample prediction error *in the validation period*:

$$V^* = \arg\min_V \sum_{t=T_1+1}^{T_0} \left(Y_{1t} - \sum_{i=2}^{J+1} w_i^*(V)\, Y_{it} \right)^2$$

- **Intuition:** Pick v_k's so that the synthetic unit fitting predictors well *also* predicts the outcome well in the portion of the pre-treatment period it has not yet "seen"

SC Estimation

- Predictor variables X typically include:
  - Pre-treatment outcomes $Y_1, ..., Y_{T_0}$ (most predictive)
  - Other observed covariates Z
- Different predictor variables X and choice of weights v_k result in distinct synthetic units

Assumptions of SC

- Assume potential outcome if unit 1 would not receive treatment Y_{1t}^0 can be expressed as follows:

$$Y_{1t}^0 = \delta_t + \theta_t Z_1 + \lambda_t \mu_1 + \varepsilon_{1t}$$

  - δ_t are common time effects (e.g. year fixed effects)
  - Z_1 are the observed, pre-treatment covariates
  - μ_1 are **permanent** unobserved variables
  - ε_{1t} are unobserved transitory shocks at the unit level with zero mean

- This assumption allows time-varying responses of multiple unobserved factors ($\lambda_t \mu_1$)
- However, it implicitly assumes **the unobservable factor loadings μ_i are fixed over time**
  - No structural breaks

SC Estimation

- Ideally, one would want to select W^* such that

$$\sum_{j=2}^{J+1} w_j^* Z_j = Z_1$$

$$\sum_{j=2}^{J+1} w_j^* \mu_j = \mu_1$$

- Thus, causal effect of treatment $\hat{\alpha}_{1t}$ is unbiased

SC Estimation

- Problem: μ_i **is unobserved**

SC Estimation

- Solution: Choose $W = (w_2^*, ..., w_{J+1}^*)$ satisfying

$$\sum_{i=2}^{J+1} w_i^* Z_i = Z_1,$$

$$\sum_{i=2}^{J+1} w_i^* Y_{i1} = Y_{11},$$

$$\sum_{i=2}^{J+1} w_i^* Y_{i2} = Y_{12}, ...,$$

$$\sum_{i=2}^{J+1} w_i^* Y_{iT_0} = Y_{1T_0}$$

SC Estimation

- **SC Theorem**: Suppose there exists W^* such that the SC matches the treated unit in the pre-treatment period ($\forall t \in \{1, ..., T_0\}$):

$$\sum_{i=2}^{J+1} w_i^* Y_{it} = Y_{1t}$$

$$\sum_{i=2}^{J+1} w_i^* Z_i = Z_1,$$

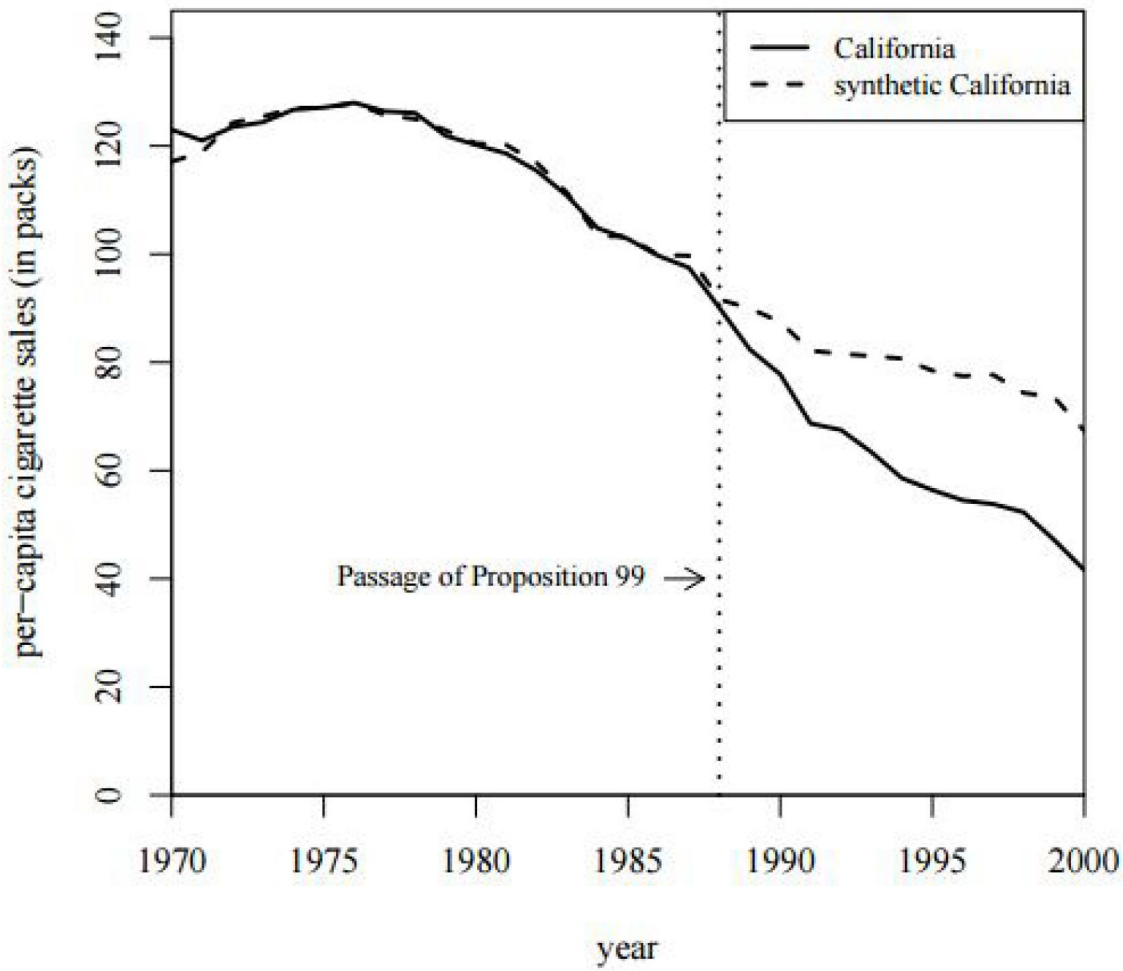
and $\sum_{t=1}^{T_0} \lambda_t' \lambda_t$ is non-singular. Then for all $t > T_0$ we have

$$\mathbb{E}\left[Y_{1t}^0 - \sum_{i=2}^{J+1} w_i^* Y_{it} \right] \to 0$$

as $T_0 \to \infty$

SC Estimation

Graphical Representation



SC Estimation

Intuition

- An approximately unbiased estimator of causal effect of treatment α_{1t} is then given by:

$$\hat{\alpha}_{1t} = Y_{1t} - \sum_{i=2}^{J+1} w_i^* Y_{it}$$

  - The difference between the observed outcome and the outcome of synthetic unit.

- Beauty of SC is that even though the μ_1 **are unobservable**, fitting $\{Y_{1t}, Z_1\}$ is sufficient to match the process
- **Intuition:** a synthetic cohort can fit $(Z_1, Y_{11}, ...Y_{1T_0})$ for a large T_0 only if it fits (Z_1, μ_1)

Example: Abadie et al. (2010)

Results: Unit Weights for SC

Table 2. State weights in the synthetic California

State	Weight	State	Weight
Alabama	0	Montana	0.199
Alaska	–	Nebraska	0
Arizona	–	Nevada	0.234
Arkansas	0	New Hampshire	0
Colorado	0.164	New Jersey	–
Connecticut	0.069	New Mexico	0
Delaware	0	New York	–
District of Columbia	–	North Carolina	0
Florida	–	North Dakota	0
Georgia	0	Ohio	0
Hawaii	–	Oklahoma	0
Idaho	0	Oregon	–
Illinois	0	Pennsylvania	0
Indiana	0	Rhode Island	0
Iowa	0	South Carolina	0
Kansas	0	South Dakota	0
Kentucky	0	Tennessee	0
Louisiana	0	Texas	0
Maine	0	Utah	0.334
Maryland	–	Vermont	0
Massachusetts	–	Virginia	0
Michigan	–	Washington	–
Minnesota	0	West Virginia	0
Mississippi	0	Wisconsin	0
Missouri	0	Wyoming	0

Example: Abadie et al. (2010)

Comparison of Synthetic Fit and Simple Average

	California		Average of
Variables	Real	Synthetic	38 control states
Ln (GDP per capita)	10.08	9.86	9.86
Percent aged 15-24	17.40	17.40	17.29
Retail price	89.42	89.41	87.27
Beer consumption per capita	24.28	24.20	23.75
Cigarette sales per capita 1988	90.10	91.62	114.20
Cigarette sales per capita 1980	120.20	120.43	136.58
Cigarette sales per capita 1975	127.10	126.99	132.81

Statistical Inference

Statistical Inference of SC

- Large-sample asymptotic inference is not possible with SC
- Abadie et al. (2010) suggest the use of **permutation methods** for inference
  - Does not rely on large-sample asymptotics
- **Main Idea:**
  - How often would we obtain results of this magnitude if we had chosen a state at random for the study instead of California ?
  - Run placebo studies by applying the SC method to states that did NOT receive treatment
    - If the placebo studies create gaps of magnitude similar to the one estimated for California
    - Then our interpretation is that our analysis does NOT provide significant evidence of causal effect

Permutation Test: Basic Steps

- **Step 1:** Estimate treatment effect for treated unit (e.g., California)

$$\hat{\alpha}_{1t} = Y_{1t} - \sum_{j=2}^{J+1} w_j^* Y_{jt}, \quad t > T_0$$

- **Step 2:** Conduct placebo studies
  - Apply SC to each control unit in donor pool as if it were treated
  - Calculate placebo effects $\{\hat{\alpha}_{2t}, ..., \hat{\alpha}_{J+1,t}\}$
- **Key Question:** How unusual is the estimated effect for California compared to effects we estimate for units that did NOT receive treatment?

Permutation Test: Statistical Inference

- **Step 3:** Calculate empirical p-value
  - Plot all estimated effect (treated and placebos) for visual inspection
  - Calculate empirical p-value at each post-treatment time t:

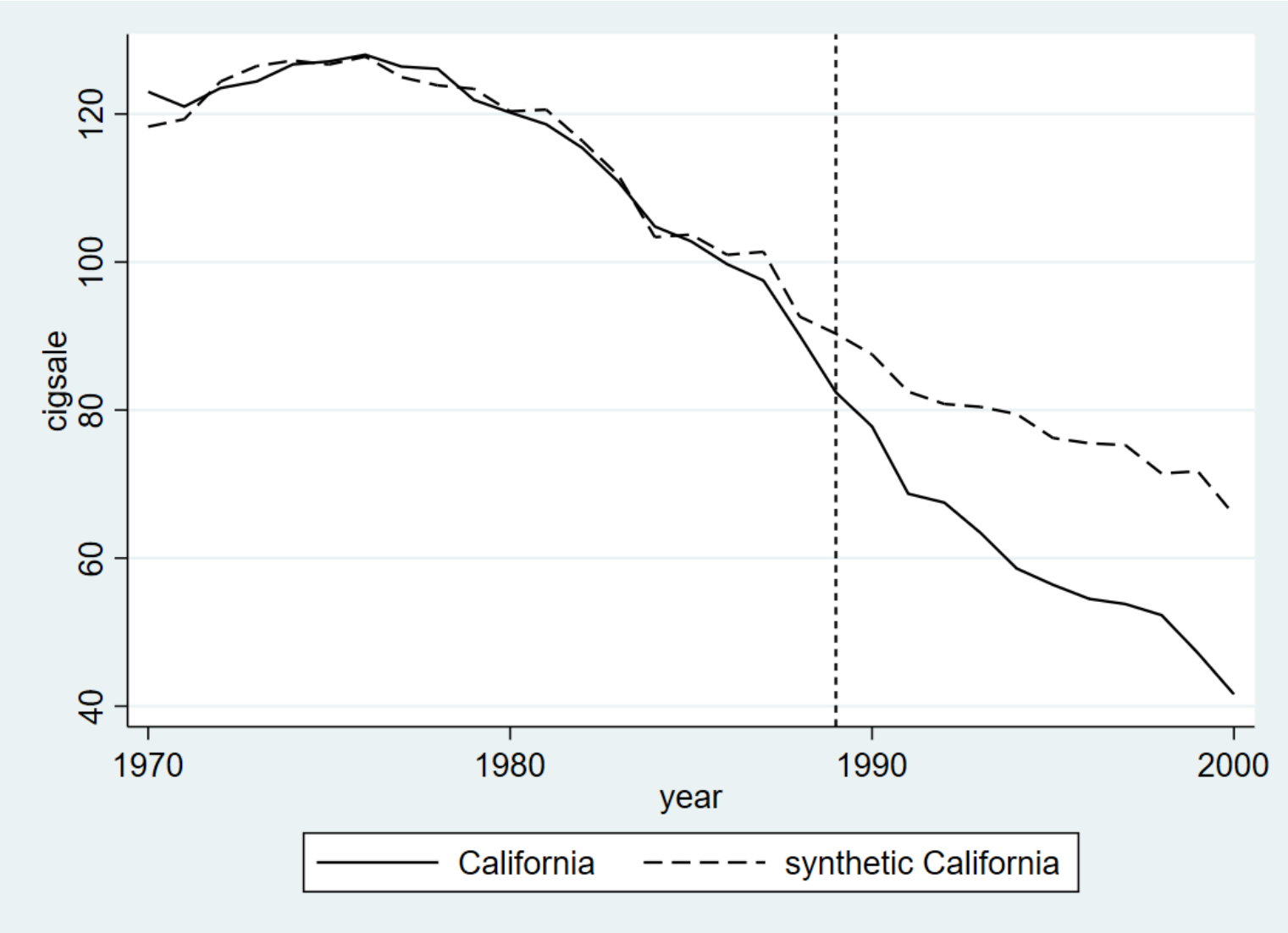
$$p_{1t} = \frac{\sum_{i=2}^{J+1} \mathbf{1}\{\hat{\alpha}_{it} \geq \hat{\alpha}_{1t}\}}{J}$$

    where $\mathbf{1}\{\cdot\}$ is the indicator function

- **Interpretation:**
  - Small p_{1t}: treatment effect is unusually large compared to placebo effects
  - Large p_{1t}: cannot distinguish treatment effect from random chance

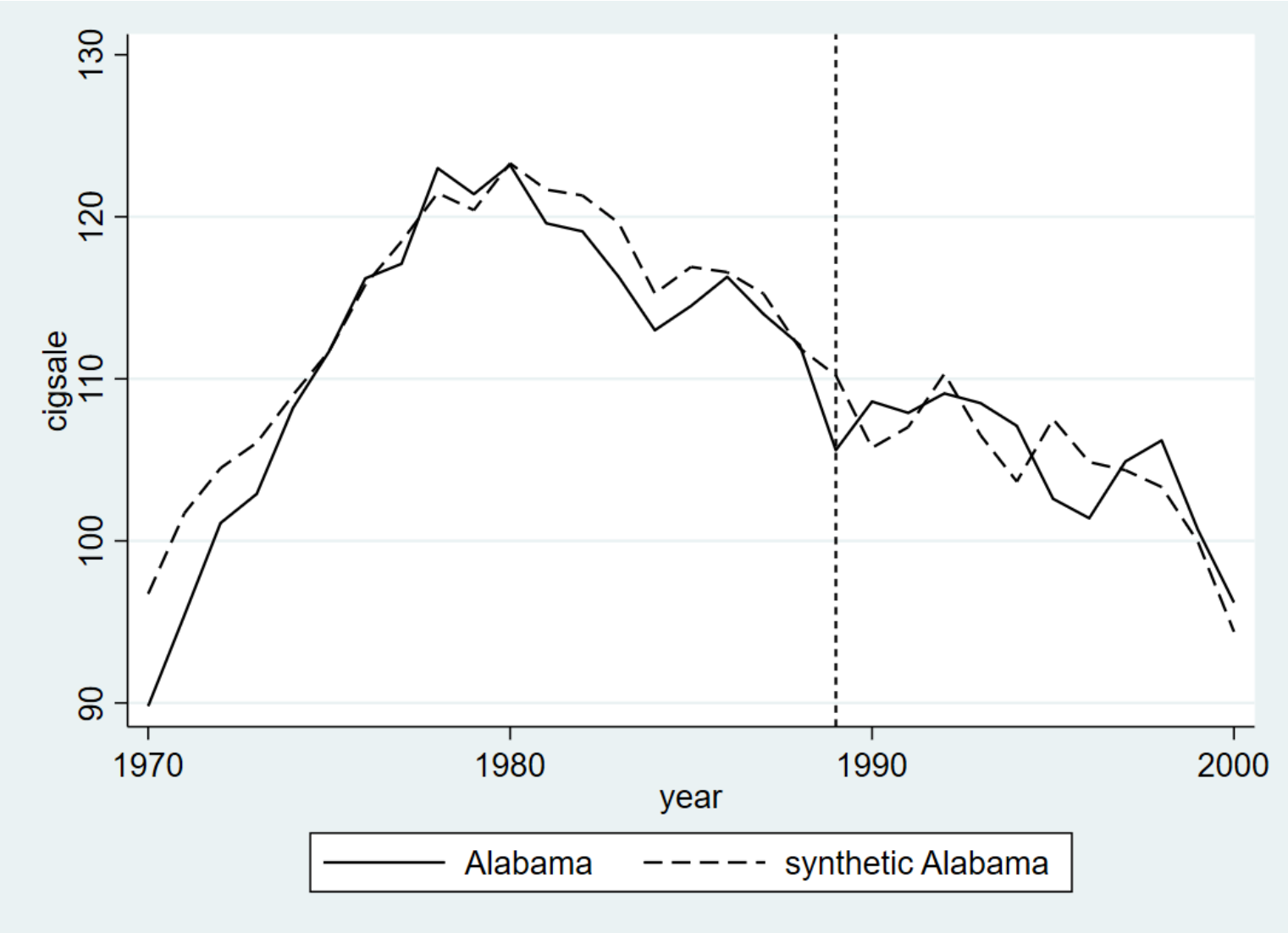
SC Analysis for Treated State

California



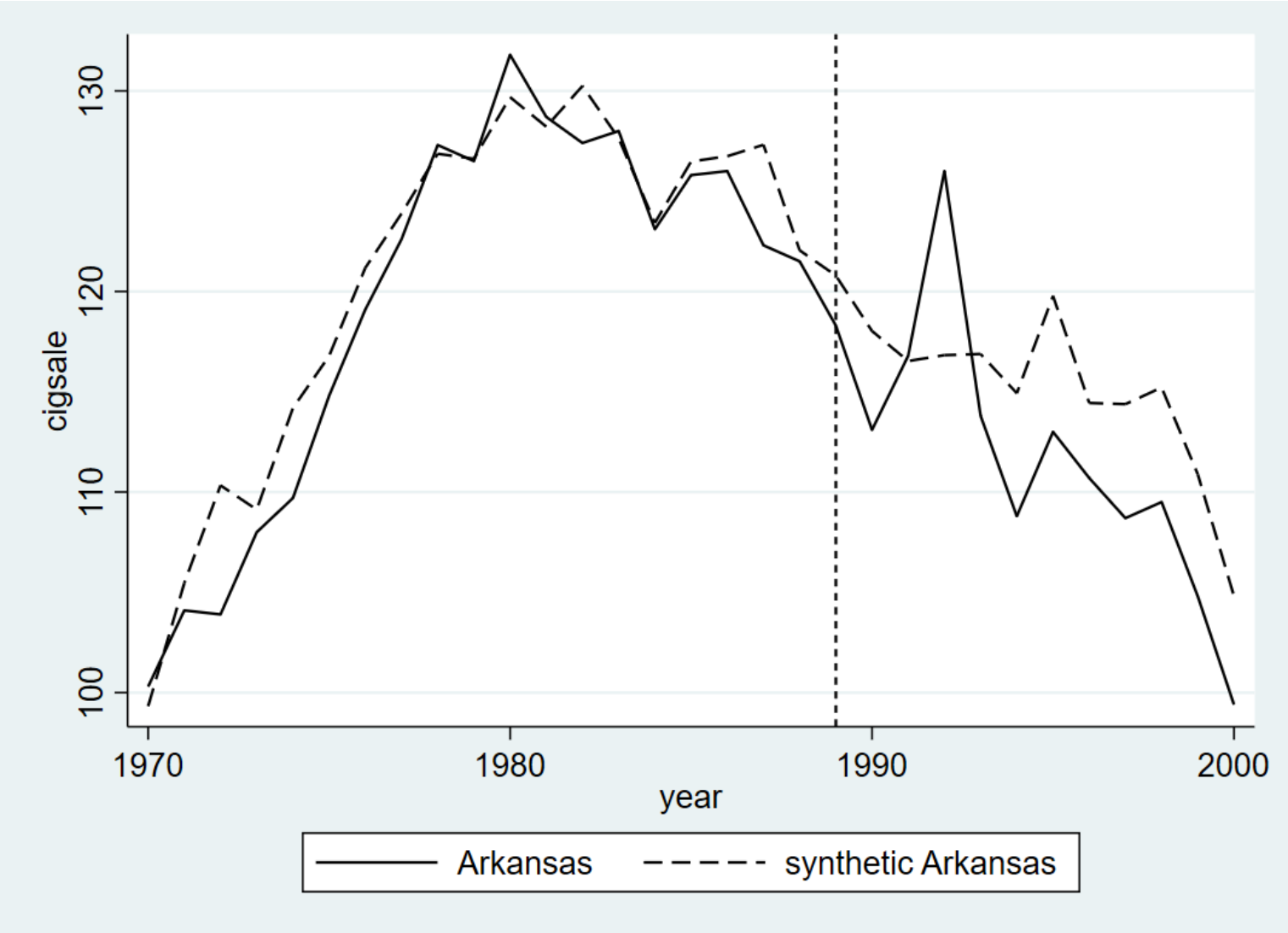
SC Analysis for Non-treated States

Alabama



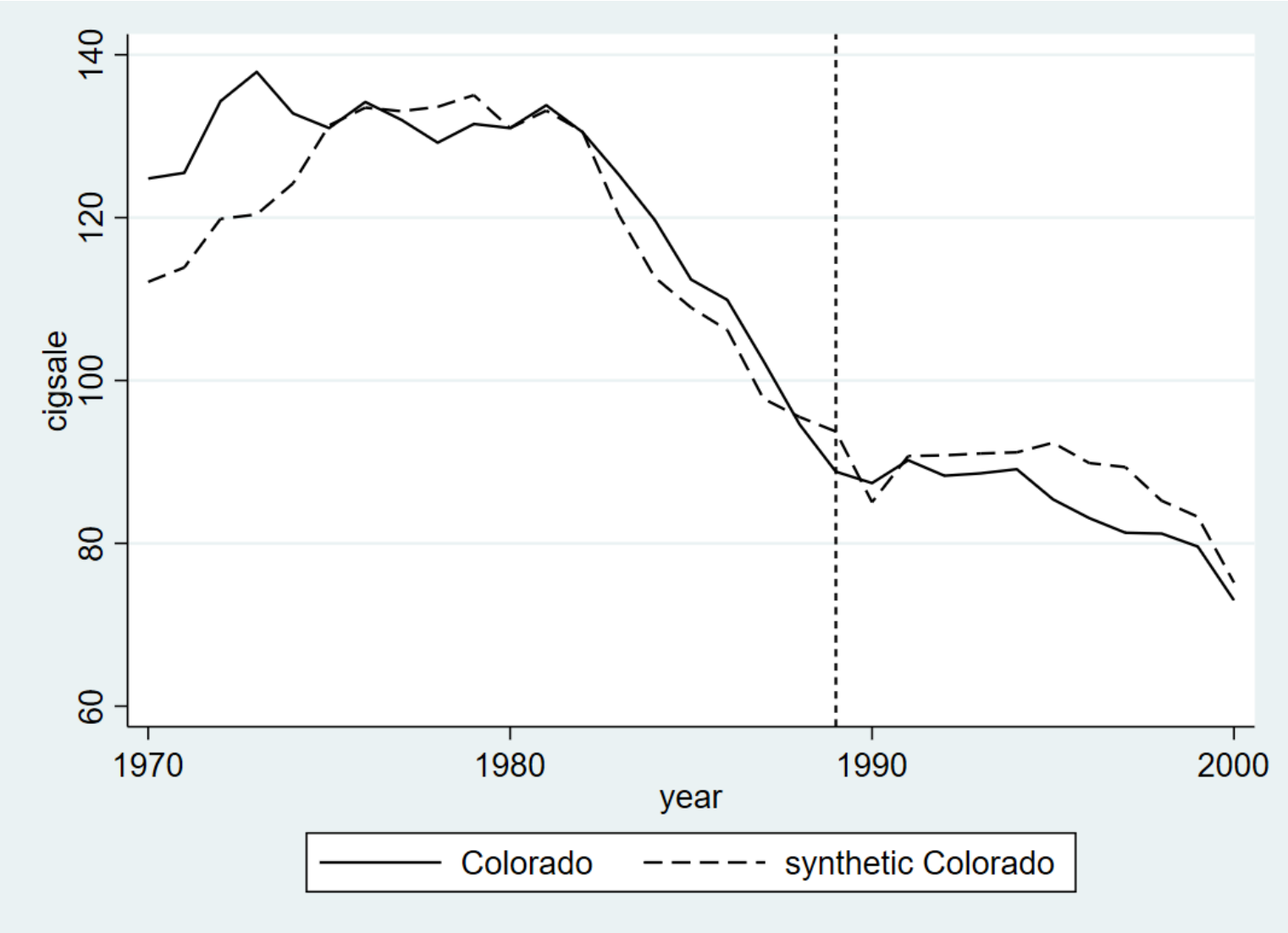
SC Analysis for Non-treated States

Arkansas



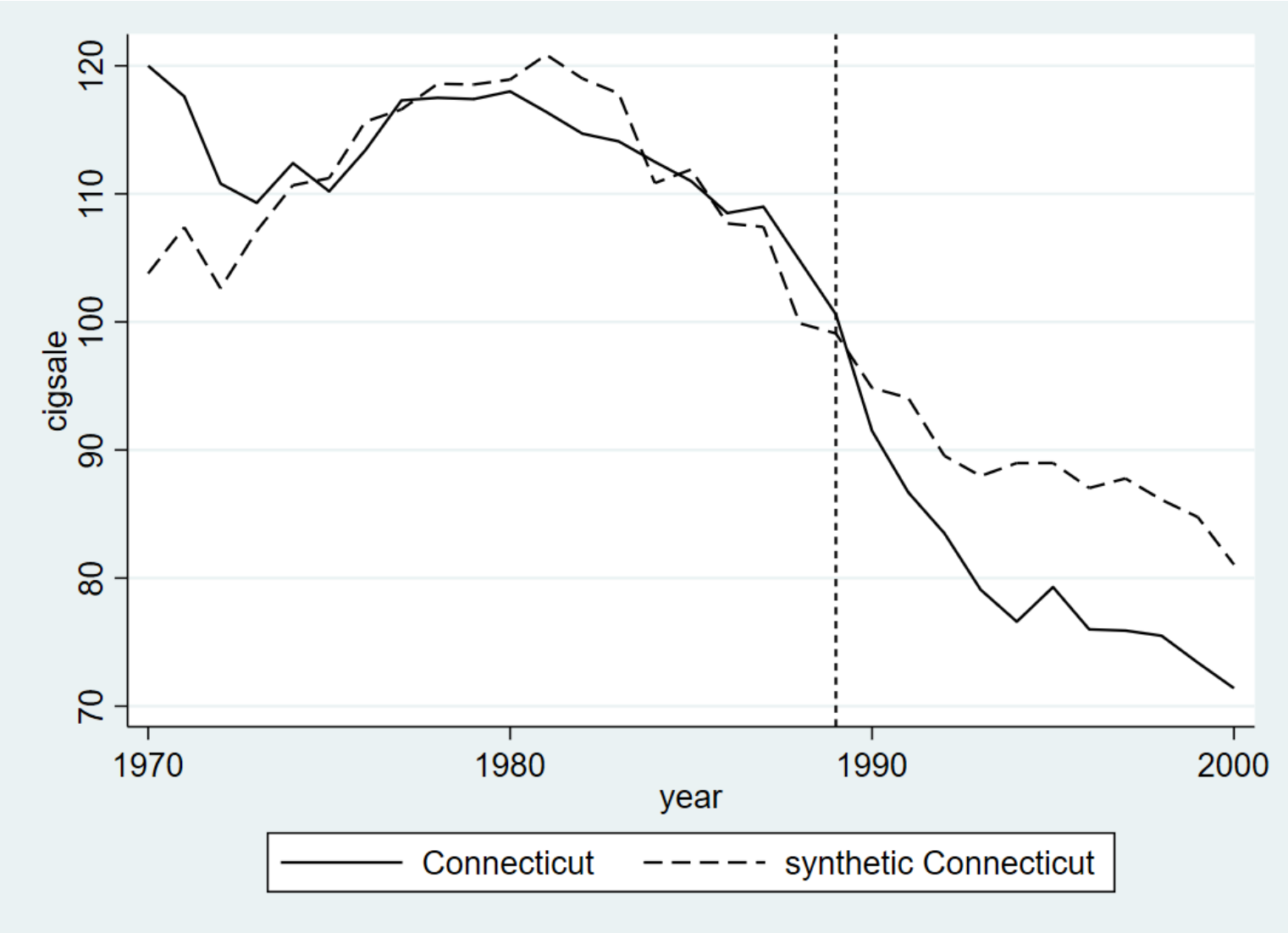
SC Analysis for Non-treated States

Colorado



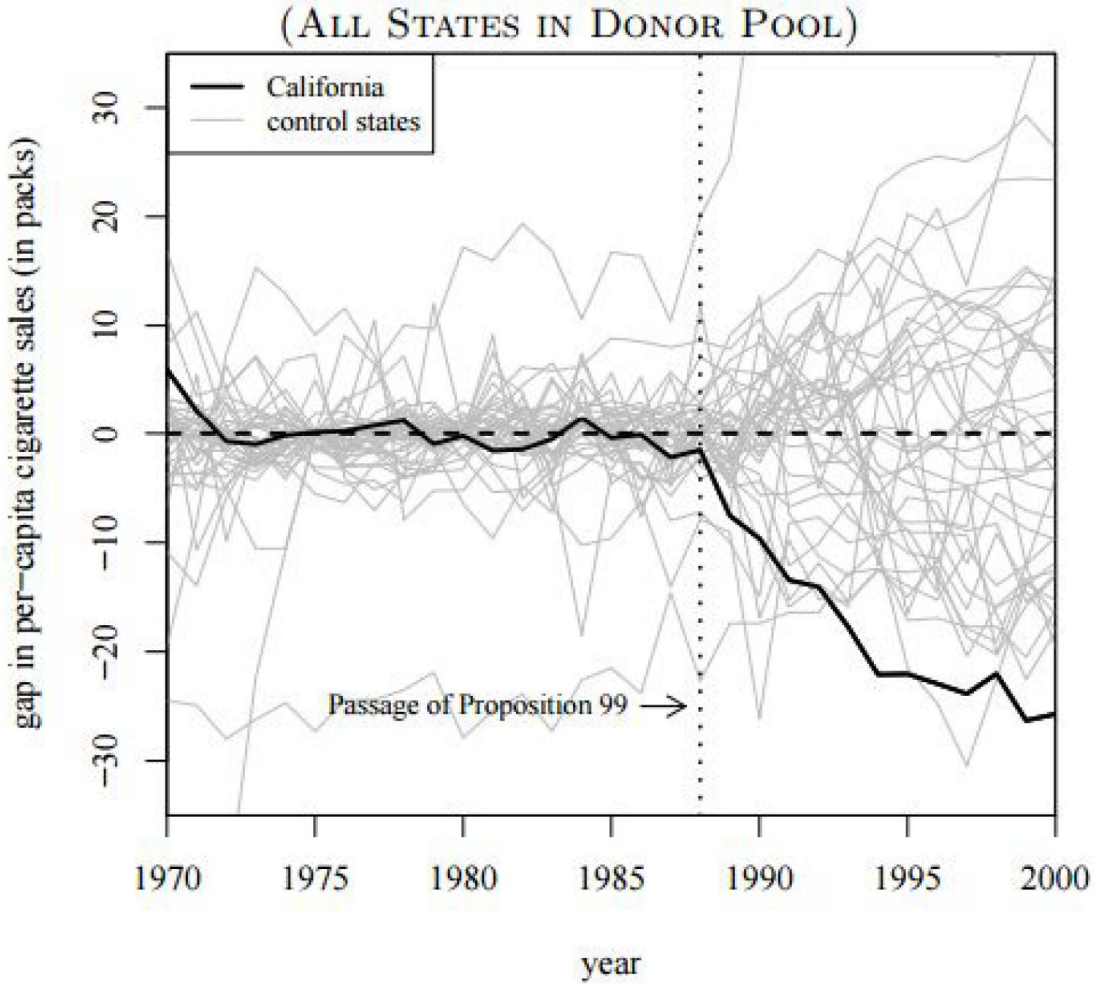
SC Analysis for Non-treated States

Connecticut



SC Estimates for California vs. Non-treated States

Use All Placebo Estimates



Statistical Inference of SC

- The placebo effects may be quite large if those units were NOT matched well in the pre-treatment period
- For example, when the SC fit is bad, we may get erroneous inferences
  - Look at bottom gray line in the above figure
- This would cause too large p-values

Statistical Inference of SC

- To control for this, one may want to **adjust $\hat{\alpha}_{it}$ for the quality of the pre-treatment matches**
- **Method 1:**
  - Removing observations with too big Root Mean Squared Predictive Error (RMSPE) during the pre-treatment period

$$RMSPE_i^{pre} = \sqrt{\frac{\sum_{t=1}^{T_0}(Y_{it} - Y_{it}^{SC})^2}{T_0}}$$

SC Estimates for CA vs. Placebo Treatments

Use High-Quality Placebo Estimates

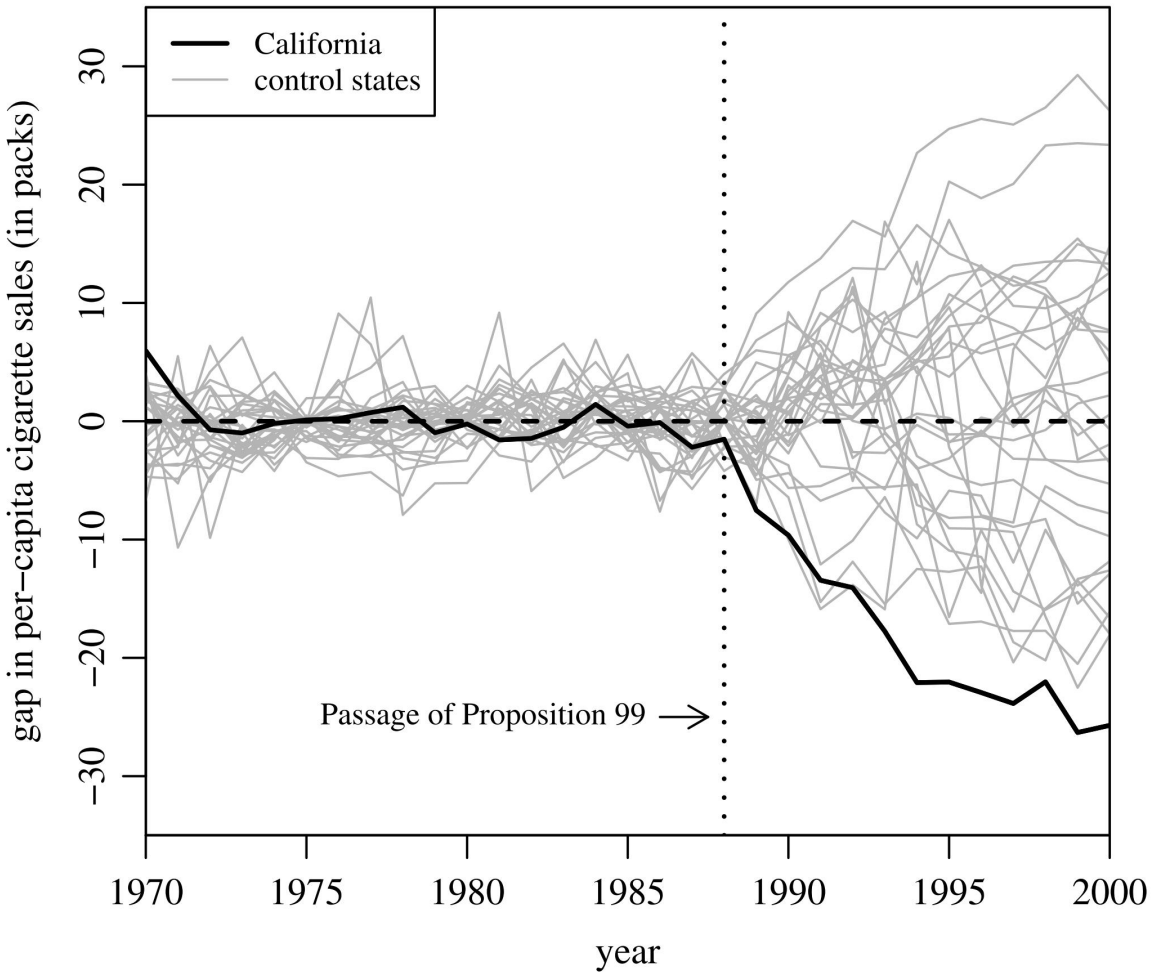


Figure 6. Per-capita cigarette sales gaps in California and placebo gaps in 29 control states (discards states with pre-Proposition 99 MSPE five times higher than California's).

Statistical Inference of SC

Standardized p-value

- **Method 2:**
  - We can also calculate p-value by taking pre-treatment matching quality into account:

$$p_{1t}^{std} = \frac{\sum_{i=2}^{J+1} \mathbf{1}\{\frac{\hat{\alpha}_{it}}{RMSPE_i^{pre}} \geq \frac{\hat{\alpha}_{1t}}{RMSPE_1^{pre}}\}}{J}$$

$$RMSPE_i^{pre} = \sqrt{\frac{\sum_{t=1}^{T_0}(Y_{it} - Y_{it}^{SC})^2}{T_0}}$$

- **Method 3:**
  - We can evaluate the significance by comparing the ratio of post/pre-treatment RMSPE:

$$p_1^{ratio} = \frac{\sum_{i=2}^{J+1} \mathbf{1}\{\frac{RMSPE_i^{post}}{RMSPE_i^{pre}} \geq \frac{RMSPE_1^{post}}{RMSPE_1^{pre}}\}}{J}$$

$$RMSPE_i^{post} = \sqrt{\frac{\sum_{t=T_0+1}^{T}(Y_{it} - Y_{it}^{SC})^2}{T - T_0}}$$

  - This method evaluates whether the treatment effect is large relative to the pre-treatment fit
  - A large ratio indicates that the post-treatment deviation is substantial compared to pre-treatment matching quality

STATA Example

STATA Example: Kim (2022)

The Effect of Paid Leave on Fertility

Brian Kim (2022), **"The Effect of Paid Leave on Fertility: Evidence from New York"**, Master Thesis

- The study examines the effect of paid parental leave on fertility rates in New York using synthetic control analysis.
- Synthetic control method is used to estimate the counterfactual fertility rates in the absence of the policy.

Overview

Program and Data

- See SCM.do
- Use fert_rate_synth.csv
- Install the following ado files:
  - synth2.ado

- `synth2`:
  - Can be used for **single treated unit**
  - Supports **placebo tests** and allows for the exclusion of control units with poor pre-treatment fit.
  - Provides options for **graphing and post-processing**, including the ability to save and export results as graphs.
  - Offers robust options for conducting **statistical inference**, including the ability to calculate p-values.

Overview

Data

- This project examines the effect of paid-parental leave on fertility
  - New York implements paid-parental leave in 2018
- Use synthetic control method to evaluate its impact
- The dataset contains state-level fertility rate and other demographic variables from 2007-2019
  - **fert:** Fertility rate, which is the outcome variable
  - **year:** This is the year and is the time variable
  - **state:** This is an id number for each state and provides the individual identifier in this panel data context

Install package

```
ssc install synth2
```

```
net install gr_postproduce, from("https://raw.
    githubusercontent.com/DiegoCiccia/StataPostProduce/main"
    ) replace
```

- `synth2` is used for synthetic control analysis, while `gr_postproduce` helps with post-processing and customizing graphs.

Step 0: Data Preparation

- Assign labels to states based on their IPUMS codes. This allows us to translate numeric codes into meaningful state names in our analysis:

```
label define state_labels 1 "Alabama" 2 "Alaska" ...
label values ipums state_labels
```

- The `label define` command creates a mapping between the state codes and their names, while `label values` applies this label to the `ipums` variable.
- This step ensures that state names are displayed in the results instead of numeric codes, making interpretation easier.

Step 1: Setting Up Panel Data

- Declare the data as panel data:

```
1 xtset ipums year
```

- The **xtset** command declares `ipums` as the panel variable (state) and ‘year‘ as the time variable.

Step 2: SC Estimation

- Syntax:

```
synth2 depvar predictorvars, trunit(#) trperiod(#)
```

- **depvar**: the outcome variable (Y)
- **predictorvars**: the list of predictor variables (Z)
- By default, all predictor variables are **averaged over the entire pre-treatment period**

Step 2: SC Estimation

- Run the synthetic control method using ‘synth2‘:

```
synth2 fert avginc black educattain white asian_pi hisp
    ///
fert(2017) fert(2016) fert(2015) fert(2014) fert(2013)
    fert(2012) ///
fert(2011) fert(2010) fert(2009) fert(2008) fert(2007),
    ///
trunit(36) trperiod(2018) xperiod(2007(1)2017)
```

- **fert(2017)**: the value of the variable **fert** in 2017 is entered as a predictor
- **xperiod(2007(1)2017)**: periods over 2007-2017 the covariates specified in predictorvars are averaged

Results: Unit Weights for SC

- Top 3 states that constitute the synthetic New York
  - New Hampshire: 23.1%
  - Kansas: 23.1%
  - Massachusetts: 15.8%

Step 3: Statistical Inference

- Conduct placebo tests using the 'synth2' command:

```
synth2 fert fert(2017) fert(2016) fert(2015) fert(2014)
    ///
fert(2013) fert(2012) fert(2011) fert(2010) fert(2009)
    ///
fert(2008) fert(2007), trunit(36) trperiod(2018) ///
xperiod(2007(1)2017) placebo(unit cutoff(2))
```

- The **placebo(unit cutoff(2))** option excludes states where pre-treatment RMSPE is more than twice that of the treated unit (New York).

Graphing Results

- Generate a data frame for storing results and save graphs:

```
synth2 fert fert(2017) fert(2016) fert(2015) fert(2014)
    ///
fert(2013) fert(2012) fert(2011) fert(2010) fert(2009)
    ///
fert(2008) fert(2007), trunit(36) trperiod(2018) ///
xperiod(2007(1)2017) frame(ny_fert) savegraph(ny_fert,
    replace)
```

Post-Processing Graphs

- Enhance graph appearance:

```
graph use "$pic\ny_fert_eff.gph"

gr_postproduce ny_fert_eff, ///
title("The Effect of Paid Leave on Fertility") ///
ytitle("Effects") xtitle("Year")
graph export "$pic\ny_fert_eff.png"
```

- Post-process and export the graph with **gr_postproduce**

Working with Frames

- Switch to the results frame:

```
frame change ny_fert
```

- Export results to CSV:

```
export delimited "$rawdata\ny_fert.csv",replace
```

- Clear all frames and return to default:

```
frames reset
```

R Example

Overview

Program and Data

- See SCM.R
- Use fert_rate_synth.csv
- Install the following package:
  - tidysynth

Step 0: Install package

- Install required packages:

```
install.packages("tidysynth")


library(tidyverse)
library(tidysynth)
library(haven)
```

- **tidysynth**: Main package for synthetic control
- **tidyverse**: Data manipulation and visualization
- **haven**: Reading various data formats

Step 0: Data Preparation

- Read and clean the data:

```
fert_data <- fert_data_raw %>%
# Drop specific states and years
filter(!(ipums %in% c(6, 34, 44, 53))) %>%
filter(year < 2020) %>%
mutate(
avginc = Personal.income/Population
) %>%
select(-statecode)
```

- Key steps:
  - Drop CA, NJ, RI, WA (states with similar policies)
  - Remove data after 2020
  - Calculate average income
  - Remove unnecessary columns

Step 2: SC Estimation

- Initialize synthetic control:

```
synth_fert <- fert_data %>%
synthetic_control(
outcome = fert,
unit = state,
time = year,
i_unit = "New York",
i_time = 2018,
generate_placebos = TRUE
)
```

- Parameters:
  - `outcome`: Fertility rate
  - `unit`: State identifier
  - `i_unit`: Treated unit (New York)
  - `i_time`: Treatment year (2018)

Step 2: SC Estimation

Add Predictors

- Add demographic predictors:

```
generate_predictor(
time_window = 2007:2017,
avg_income = mean(avginc, na.rm = TRUE),
black_pop = mean(black, na.rm = TRUE),
educ_attain = mean(educattain, na.rm = TRUE),
white_pop = mean(white, na.rm = TRUE),
asian_pop = mean(asian_pi, na.rm = TRUE),
hisp_pop = mean(hisp, na.rm = TRUE)
)
```

- Takes average of demographics over 2007-2017
- Handles missing values with `na.rm = TRUE`

Step 2: SC Estimation

Add Predictors

- Add lagged fertility rates:

```
generate_predictor(
time_window = 2017,
fert_2017 = fert
) %>%
generate_predictor(
time_window = 2016,
fert_2016 = fert
)
# ... (similar for other years)
```

- Adds fertility rates for each year from 2007-2017
- Each year added separately for flexibility

Step 2: SC Estimation

Weight Generation

- Generate synthetic control weights:

```
# Generate weights using default optimization parameters
generate_weights(optimization_window = 2007:2017) %>%
generate_control()
```

- Parameters:
  - `optimization_window`: Time period for weight calculation

Graphing Results

- Generate and display plots:

```
plot_trends <- synth_fert %>%
plot_trends() +
labs(
title = "Fertility Trends: NY vs Synthetic Control",
y = "Fertility Rate",
x = "Year"
)

print(plot_trends)
ggsave("synth_fert.png", plot = plot_trends)
```

- Three types of plots:
  - Trends plot: Actual vs synthetic
  - Differences plot: Gap analysis
  - Placebos plot: Statistical inference

Graphing Results

- Generate and display difference plot:

```
plot_differences <- synth_fert %>%
plot_differences() +
labs(
title = "Gap in Fertility Rates",
y = "Difference in Fertility Rate",
x = "Year"
)

print(plot_differences)
ggsave("synth_fert_differences.png",
plot = plot_differences)
```

- Gap Analysis Features:
  - Shows treatment effect over time
  - Zero line indicates no effect
  - Negative values show fertility decrease
  - Helps identify pre-treatment fit quality

Graphing Results

- Generate and display placebo plot:

```
plot_placebos <- synth_fert %>%
plot_placebos() +
labs(
title = "Placebo Tests",
y = "Gap in Fertility Rate",
x = "Year"
)

print(plot_placebos)
ggsave("synth_fert_placebos.png",
plot = plot_placebos)
```

- Placebo Test Features:
  - Shows treatment effects for all states
  - Treated unit highlighted
  - Provides visual inference
  - Helps assess significance of results

Results Export

- Save results and tables:

```
summary_tables <- synth_fert %>%
grab_synthetic_control()
predictor_balance <- synth_fert %>%
grab_balance_table()


write_csv(summary_tables,
file.path(workdata, "ny_fert_results.csv"))
write_csv(predictor_balance,
file.path(workdata, "ny_fert_balance.csv"))
```

- Exports:
  - Synthetic control results
  - Predictor balance tables

Recent Development: Regression-Based SCM

Regression-Based Synthetic Control

Motivation

- Standard SC constrains the weights:

$$w_i \geq 0 \quad \text{and} \quad \sum_{i=2}^{J+1} w_i = 1$$

- These constraints ensure **no extrapolation**: the synthetic unit always lies **within** the convex hull of the donor pool
- **Problem:** if the treated unit lies **outside** the convex hull, perfect pre-treatment fit is impossible
  - The synthetic unit cannot fully match the treated unit's trajectory
  - Residual pre-treatment mismatch may bias post-treatment estimates

Regression-Based Synthetic Control

Idea: Relax the Constraints

- **Key idea:** relax the non-negativity and summation constraints to allow **extrapolation**
- Concretely, estimate counterfactual outcomes using a regression:

$$\hat{Y}_{1t}^{0} = \hat{\alpha} + \sum_{j=2}^{J+1} \hat{w}_j \, Y_{jt}, \quad t > T_0$$

- Two relaxations compared to standard SC:
  - Add an **intercept** α
  - Allow weights to be **negative** and **not sum to one**
- But with J potentially large, we need **regularization** — enter LASSO

Regression-Based Synthetic Control

Why an Intercept?

- Suppose the treated unit's outcome level is **systematically higher** than any donor, even before treatment
- Standard SC cannot handle this: the convex combination of donors can never reach the treated unit's level
- The intercept α absorbs this **constant level difference**:

$$Y_{1t} = \underbrace{\alpha}_{\text{level shift}} + \sum_{j=2}^{J+1} w_j \, Y_{jt} + \varepsilon_t$$

- Think of it like adding a constant in a regression — it lets the donor outcomes *track the trend* of Y_{1t} without needing to match its *level* exactly

Regression-Based Synthetic Control

Baseline: OLS with Intercept

- The simplest approach: just run an **OLS regression** of Y_{1t} on donor outcomes, adding an intercept

$$\min_{\alpha, \{w_j\}} \sum_{t=1}^{T_0} \left(Y_{1t} - \alpha - \sum_{j=2}^{J+1} w_j Y_{jt} \right)^2$$

- This is exactly OLS — the "regressors" are the **pre-treatment outcomes of donor units**
- Two relaxations compared to standard SC:
  - **Intercept** α: absorbs a permanent level difference between NY and donors
  - **Unconstrained weights**: w_j can be negative and need not sum to one $\Rightarrow$ extrapolation allowed
- **Doudchenko & Imbens (2016)**: this is the baseline version of regression-based SC

Regression-Based Synthetic Control

Extension: When J is Large

- OLS works well when J is small relative to T_0
- **Problem:** when J is large (many donors), OLS overfits the pre-treatment period — poor out-of-sample performance
- **Solution:** add a **regularization penalty**, just as in standard regression:
  - **LASSO** (ℓ_1 penalty $\lambda \sum_j |w_j|$): selects a sparse set of donors
  - **Ridge** (ℓ_2 penalty $\lambda \sum_j w_j^2$): shrinks all weights, stable under collinearity
  - **Elastic net**: mixes both — sparse *and* stable
- In our empirical example, J is small $\Rightarrow$ we use the **OLS version** as our baseline

Regression-Based Synthetic Control

A LASSO Version

- Treat the weight-finding step as a **LASSO regression**:

$$\min_{\alpha,\{w_j\}} \underbrace{\sum_{t=1}^{T_0}\left(Y_{1t}-\alpha-\sum_{j=2}^{J+1} w_j Y_{jt}\right)^2}_{\text{pre-treatment fit}} + \underbrace{\lambda\sum_{j=2}^{J+1}|w_j|}_{\text{LASSO penalty}}$$

- Exactly like standard LASSO, but the "regressors" are the **pre-treatment outcomes of donor units**
- The LASSO penalty $\lambda\sum|w_j|$ does two things you already know:
  - **Shrinks** weights toward zero $\Rightarrow$ prevents overfitting to pre-treatment period
  - **Selects** a sparse set of donors (many $\hat{w}_j = 0$) $\Rightarrow$ interpretable synthetic control

Regression-Based Synthetic Control

Doudchenko & Imbens (2016): Elastic Net

- **Problem with pure LASSO:** when donor outcomes are highly correlated, LASSO picks one arbitrarily and drops the rest — unstable
- **Solution:** mix LASSO with **ridge** $\Rightarrow$ the **elastic net** penalty:

$$\lambda_1 \left(\frac{1-\lambda_2}{2} \sum_j w_j^2 + \lambda_2 \sum_j |w_j| \right)$$

- The mixing parameter $\lambda_2 \in [0,1]$ interpolates between:
  - $\lambda_2 = 1$: pure **LASSO** (sparse, may be unstable under collinearity)
  - $\lambda_2 = 0$: pure **ridge** (stable, but all donors kept)
  - $0 < \lambda_2 < 1$: **elastic net** — sparse *and* stable
- Both λ_1 and λ_2 are chosen by cross-validation over the pre-treatment period

Regression-Based SCM

Inference: Permutation Test

- The **same permutation (placebo) test** from standard SCM applies directly
- Procedure:
  1. Apply regression-based SC to **each donor unit** as if it were treated
  2. Compute two statistics per unit:
     - **RMSPE ratio**: $\frac{\sqrt{\text{post-MSPE}}}{\sqrt{\text{pre-MSPE}}}$ — adjusts for units that are hard to fit in-sample
     - **Mean signed post-gap**: $\frac{1}{T_1}\sum_{t>T_0}(Y_{it} - \hat{Y}_{it}^{(0)})$ — tests direction and magnitude
  3. **P-value** = fraction of placebo statistics $\geq$ the treated unit's statistic
- The test is **agnostic to weight constraints** — valid for standard SC, LASSO-based SC, or any variant

R Example: Regression-Based SCM

Package: scpi

Regression-Based SCM in R

- scpi implements regression-based SCM in pure R
- Three main functions:
  - scdata() — structures the panel into pre/post matrices
  - scest() — finds optimal weights (and intercept $\hat{\alpha}$)
  - scpi() — prediction intervals (not used here)

```
install.packages("scpi")
library(scpi); library(tidyverse); library(purrr)
```

- **Why not tidysynth?** Only supports constrained weights ($w_j \geq 0$, $\sum w_j = 1$). scpi allows unconstrained weights, an intercept $\hat{\alpha}$, and regularization (OLS / ridge / LASSO)

Step 1: scdata()

Data Preparation

```
df_sc <- scdata(
  df           = fert_sc,       # long-format panel data
  id.var       = "state",       # unit identifier
  time.var     = "year",        # time variable
  outcome.var  = "fert",        # outcome of interest
  unit.tr      = "New York",    # treated unit
  unit.co      = donors,        # donor pool (character vector
      )
  period.pre   = 2007:2017,     # pre-treatment window
  period.post  = 2018:2019,     # post-treatment window
  constant     = TRUE,          # add intercept alpha-hat
  cointegrated = FALSE)
```

- `constant = TRUE`: adds intercept $\hat{\alpha}$ — absorbs a constant level difference between NY and the donor pool
- Returns a structured object passed directly to `scest()`

Step 2: `scest()`

Weight Estimation

```
res_est <- scest(
  data      = df_sc,
  w.constr = list(name = "ols"),  # unconstrained OLS weights
  solver    = "ECOS")
```

- `w.constr` mirrors the estimators we discussed:
  - `"simplex"`: standard SCM — $w_j \geq 0$, $\sum_j w_j = 1$
  - `"ols"`: OLS, no regularization — **our baseline** (small J)
  - `"lasso"`: unconstrained + ℓ_1 penalty — use when J is large
  - `"ridge"`: unconstrained + ℓ_2 penalty — stable under collinearity
- `"ols"` + `constant = TRUE` $\Rightarrow$ baseline Doudchenko & Imbens (2016)
- λ for `"lasso"`/`"ridge"` is chosen by cross-validation over the pre-treatment period

Permutation Test

Why Write It Ourselves?

- `scpi()` provides **prediction intervals**; there is **no built-in permutation test**
- We implement **placebo-in-space** manually:
  1. Apply regression-based SC to **every unit** (NY + all donors), each time treating that unit as the "treated" unit
  2. Compute each unit's gap series: $\text{gap}_{it} = Y_{it} - \hat{Y}_{it}^{(0)}$
  3. Compute two statistics per unit (RMSPE ratio and mean signed post-gap)
  4. P-value = fraction of placebo statistics $\geq$ NY's statistic
- `run_sc_one()` applies steps 1–3 for a single unit

Permutation Function

run_sc_one() (R)

```r
run_sc_one <- function(unit_id, df_long) {
  tryCatch({
    don_i <- setdiff(unique(df_long$state), unit_id)
    df_i <- scdata(df=df_long, id.var="state",
      time.var="year", outcome.var="fert",
      unit.tr=unit_id, unit.co=don_i,
      period.pre=period_pre, period.post=period_post,
      constant=TRUE, cointegrated=FALSE)
    res_i <- scest(data=df_i,
      w.constr=list(name="lasso"), solver="ECOS")
    g <- c(res_i$data$Y.pre[,1], res_i$data$Y.post[,1]) -
        c(res_i$est.results$Y.pre.fit[,1],
          res_i$est.results$Y.post.fit[,1])
    t0 <- length(period_pre)
    list(unit=unit_id, gap=g,
         mspe_pre =mean(g[1:t0]^2),
         mspe_post=mean(g[(t0+1):length(g)]^2),
         mean_gap =mean(g[(t0+1):length(g)]))
  }, error=function(e) NULL)
}
```

- Returns `NULL` if the LASSO solver fails for a given unit (filtered out before computing p-values)

Two P-values

What Do They Measure?

- **P1 — RMSPE ratio** $= \dfrac{\sqrt{\text{post-MSPE}}}{\sqrt{\text{pre-MSPE}}}$
  - How much larger is the post-treatment misfit relative to pre-treatment fit?
  - Adjusts for units that inherently fit worse in-sample
- **P2 — Mean signed post-gap** $= \dfrac{1}{T_1}\sum_{t>T_0} \text{gap}_{it}$
  - Tests whether the direction and magnitude of the post-treatment gap is unusual
  - One-tailed: direction auto-detected from NY's gap sign
- **Pre-MSPE filter**: exclude placebos with pre-MSPE $>$ 5$\times$ NY's
  - Poor-fitting placebos inflate the reference distribution
  - Filtering focuses inference on "well-matched" placebos

P-value Computation

R

```r
perm_ok <- Filter(Negate(is.null),
  lapply(unique(fert_sc$state), run_sc_one, df_long=fert_sc)
      )

stats_tbl <- map_dfr(perm_ok, function(x)
  tibble(unit=x$unit, mspe_pre=x$mspe_pre,
         ratio   =sqrt(x$mspe_post)/sqrt(x$mspe_pre),
         mean_gap=x$mean_gap))

# P1: RMSPE ratio
ny_ratio <- stats_tbl$ratio[stats_tbl$unit == "New York"]
p_ratio  <- mean(stats_tbl$ratio >= ny_ratio)

# P2: mean signed gap (one-tailed, direction from NY's sign)
ny_gap <- stats_tbl$mean_gap[stats_tbl$unit == "New York"]
p_gap  <- if (ny_gap < 0) mean(stats_tbl$mean_gap <= ny_gap)
     else
                          mean(stats_tbl$mean_gap >= ny_gap)

# Pre-MSPE filter (5x cutoff)
ny_mspe   <- stats_tbl$mspe_pre[stats_tbl$unit == "New York"
    ]
stats_f   <- stats_tbl %>% filter(mspe_pre <= 5 * ny_mspe)
p_ratio_f <- mean(stats_f$ratio >= ny_ratio)
```

Gap Trajectory Plot

R

```r
gap_df <- map_dfr(perm_ok, function(x)
  tibble(unit =x$unit,
         year =c(period_pre, period_post)[seq_along(x$gap)],
         gap  =x$gap,
         is_ny=(x$unit == "New York")))

ggplot(gap_df, aes(x=year, y=gap, group=unit)) +
  geom_line(data=subset(gap_df, !is_ny),
            color="grey70", alpha=0.6, linewidth=0.4) +
  geom_line(data=subset(gap_df,  is_ny),
            color="firebrick", linewidth=1.2) +
  geom_vline(xintercept=2017.5, linetype="dashed") +
  geom_hline(yintercept=0, color="grey30") +
  labs(title="Permutation Test: Regression-Based SCM",
       x="Year", y=expression(gap == Y[it] - hat(Y)[it]^{"
           (0)"})) +
  theme_bw()
```

- Pre-treatment gaps near zero = good fit; NY's post-treatment gap (red) should stand out from placebos (grey)

Practical Issues

Practical Issues with SC

Computation Time

- The STATA command `synth` implements an optimization procedure to select the optimal W^* for each treated i
- The optimization scheme suffers from a curse of dimensionality
  - Lots of non-treated units increase the number of potential W's
- One way to reduce computation time is to only use non-treated units that are similar to treated units
  - It also improve match quality

Practical Issues with SC

Length of Pre-treatment Period

- For the SC estimator to estimate the parameter of interest, the "**pre-treatment period must be large relative to the scale of the transitory shocks**"
  - This is ambiguous, and it is unclear how to test necessary conditions of whether this holds in the data

Suggested Readings

- Chapter 10, Causal Inference: The Mixtape
- Abadie, A. (2019). Using synthetic controls: Feasibility, data requirements, and methodological aspects. Journal of Economic Literature.