Estimating the causal effect

Synthetic control method

„arguably the most important innovation in the policy evaluation literature in the last 15 years”

Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. Journal of Economic perspectives, 31(2), 3-32.

In this part

- Introducing the synthetic control method
- How to quantify uncertainty
- What choices do we need to make and how do these impact our causal effect estimates?
- Performing the synthetic control method with tidysynth package

Synthetic control

Basic idea

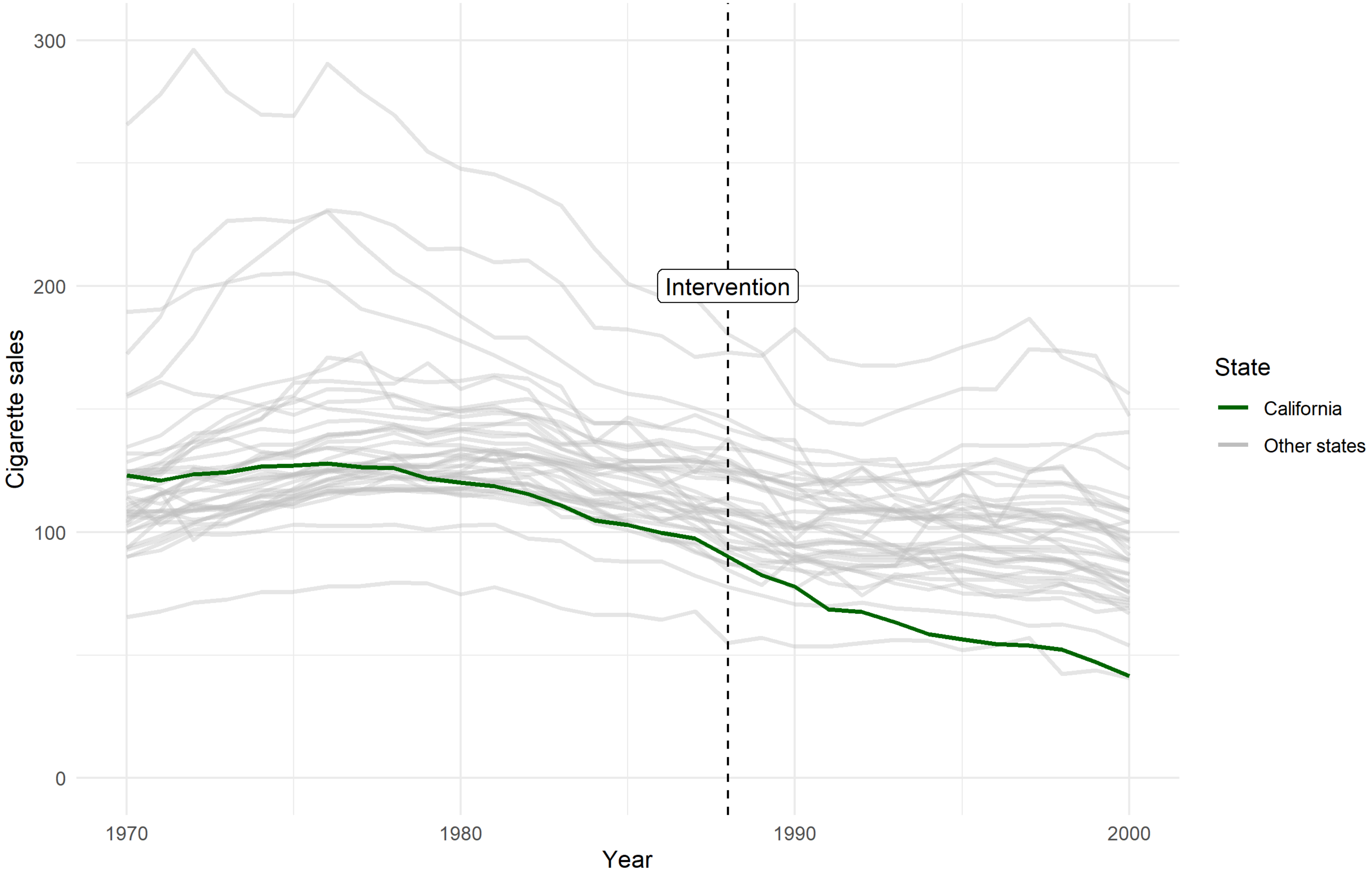
With diff-in-diff we used a control unit to attempt a correction for unmeasured time-varying confounders (e.g., macroeconomic situation in U.S.A.)

- You need a good control unit!
- How much is Utah like California?

We can instead use a weighted average of a **donor pool** of control units to create a **synthetic control** unit

- Choose the weights such that control is like California

Panel data for proposition 99



Synthetic control

Introduced in 2000s

- *Abadie, A., & Gardeazabal, J. (2003). The economic costs of conflict: A case study of the Basque Country. American Economic Review, 93(1), 113-132.*
- *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.*

An R package with JSS paper in 2011

- *Abadie, A., Diamond, A., & Hainmueller, J. (2011). Synth: An R package for synthetic control methods in comparative case studies. Journal of Statistical Software, 42(13).*

A great overview paper with recent learnings in 2021

- *Abadie, A. (2021). Using synthetic controls: Feasibility, data requirements, and methodological aspects. Journal of Economic Literature, 59(2), 391-425.*

Synthetic control

Causal **estimand** is the effect of the intervention at time t:

$$CE_t = Y_t^1 - Y_t^0$$

where $t > T_0$ (i.e., the post-intervention time period)

Synthetic control

$$CE_t = Y_t^1 - Y_t^0$$

- Again, Y_t^1 is observed
  the post-intervention time series for the treated unit
- But Y_t^0 is an unobserved counterfactual
  what would have happened had the treated unit been untreated?

Synthetic control

$$CE_t = Y_t^1 - Y_t^0$$

The problem of estimating the effect of a policy intervention is equivalent to the problem of estimating Y_t^0

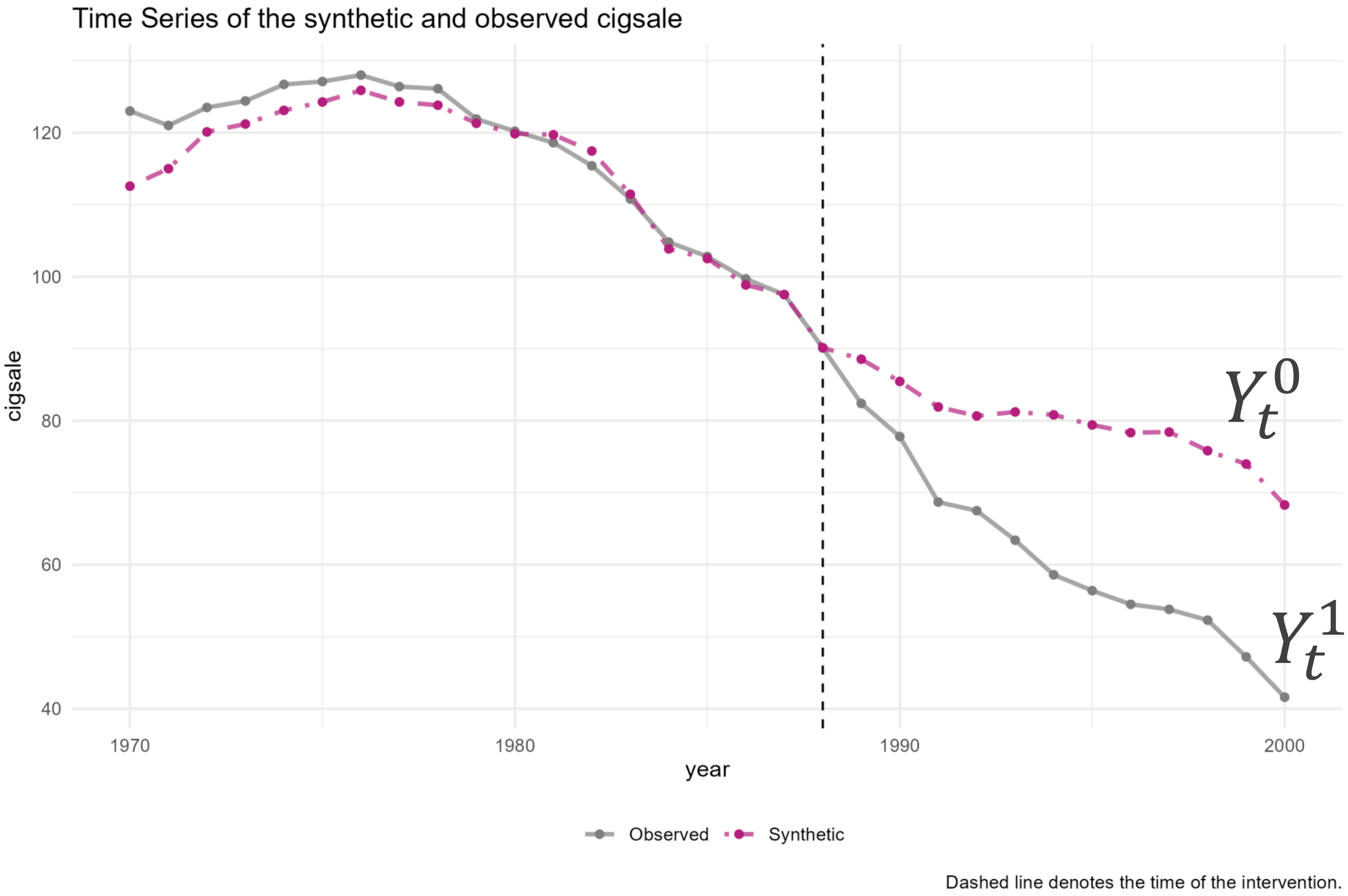
Abadie, A. (2021). Using synthetic controls: Feasibility, data requirements, and methodological aspects. Journal of Economic Literature, 59(2), 391-425.

Synthetic control

We can estimate the counterfactual as follows:

$$Y_t^0 = \sum_{j=1}^{J} w_j C_{jt}$$

- C_{jt} is the time-series for donor pool unit j at time t
  e.g., cigarette sales in Utah in 1989-2000
- w_j is a weight for this state
  e.g., a single value like 0.334



Time Series of the synthetic and observed cigsale

Dashed line denotes the time of the intervention.

Three questions

- How to choose the weights?
- Which units can go in the donor pool?
- How to make sure that the synthetic control is interpretable?

Estimating weights

Estimating weights

- Choose weights such that the synthetic control **looks like** the treated unit
- Use only pre-intervention data for this
- Weights should be positive and sum to one
  Interpolation constraint / convex hull

Estimating weights

What does it mean to looks like California? This is a choice by the researcher!

- Pre-intervention target variables
    - Cigarette sales
- Pre-intervention covariates
    - Population composition
    - Average income of population
    - Price of cigarettes
    - Beer consumption

Estimating weights

- Simultaneous estimation of two weights
  - Unit weights w_j
    How important is each donor pool unit j?
  - Variable weights v_h
    How important is each variable p?
- Choose w to minimize v-weighed multivariate Euclidean distance between treated and synthetic control pre-intervention

$$\widehat{w}_j = \min_{w_j} \left\| v \cdot (X_T - w^T X_D) \right\|$$

- Like nearest neighbours matching!

Estimating weights

How to choose v_h?

Simple

Use inverse of variance of each variable h

Like scaling the variables and then using unweighted Euclidean distance matching

Complex

Choose v such that root mean squared prediction error (RMSPE) on pre-intervention target variable is minimized

Increased importance of good prediction. We will get back to this later

Choosing donor pool

No interference / spillover

The donor pool unit does not receive any intervention effect

Example spillover effects

- Californians living near the border may buy their cigarettes in states across the border
- Other states may pass laws similar to on California

Measurement

Measure control variables and target variable in the donor pool unit **before and after** the intervention

- Ideally, large pre-intervention time window
  Otherwise, risk overfitting pre-intervention; bad prediction for counterfactual
- Be able to measure target variable after intervention
  counterfactual is weighted average of this

Convex hull condition

Distribution of control and target variables in donor pool should cover treated unit

- It should be possible to interpolate the target unit values pre-intervention using the donor pool units
- If donor pool units all have much higher cigarette sales, it is impossible to represent cigarette sales in California using positive weights which sum to 1

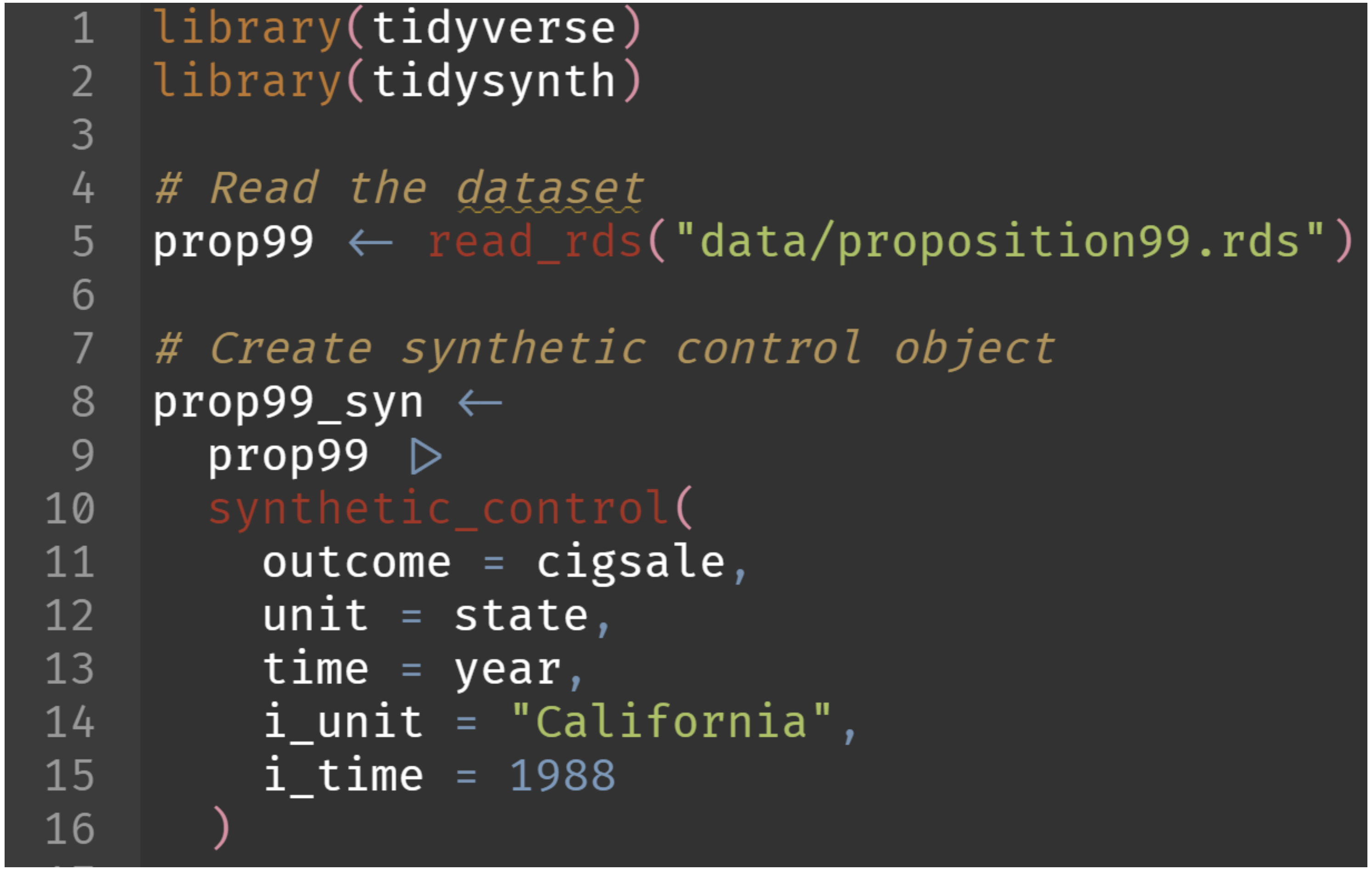
Interpretability

Interpretability

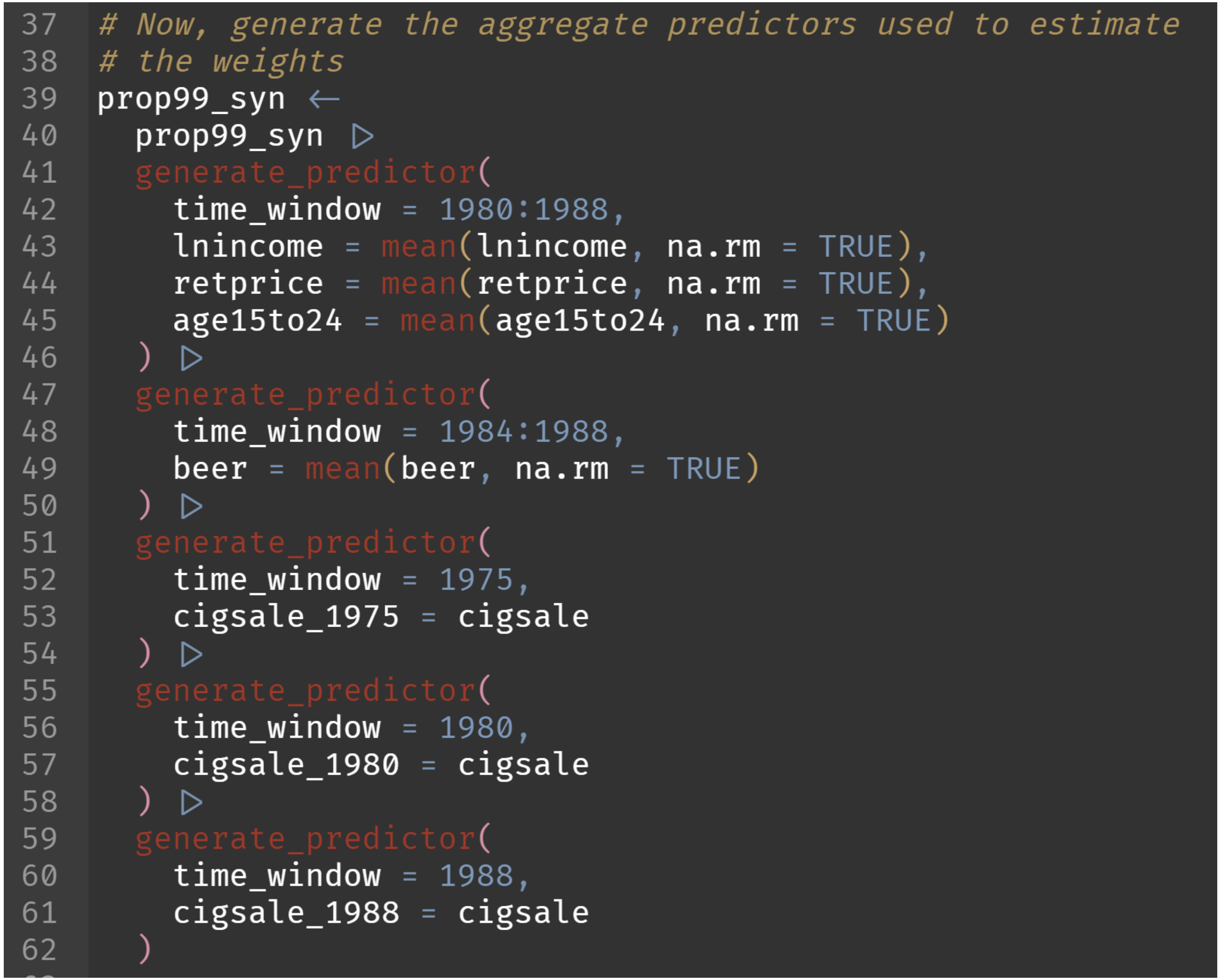
- If donor pool is large, synthetic control can be combination of many units
- Hard to interpret what the synthetic control unit is!
- Therefore: sparse estimation of weights
- Additional penalty such that most weights are 0
- The units belonging to nonzero weights can be manually inspected

Synthetic control using tidysynth

Synthetic control in practice

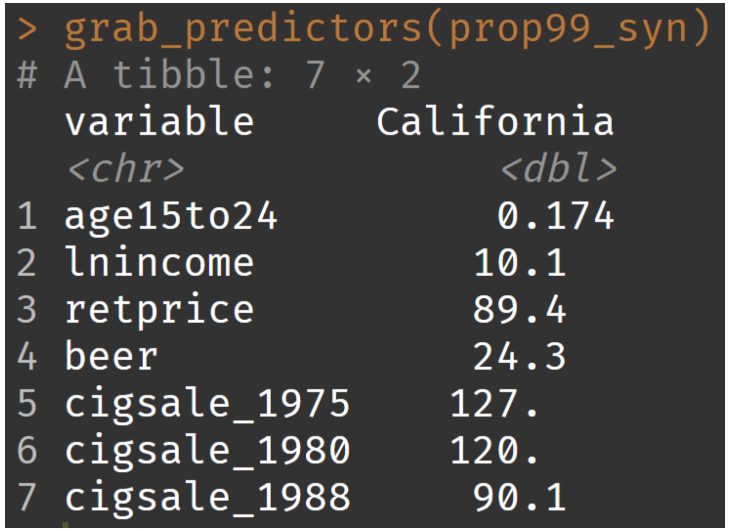


```
library(tidyverse)
library(tidysynth)

# Read the dataset
prop99 ← read_rds("data/proposition99.rds")

# Create synthetic control object
prop99_syn ←
  prop99 ▷
  synthetic_control(
    outcome = cigsale,
    unit = state,
    time = year,
    i_unit = "California",
    i_time = 1988
  )
```

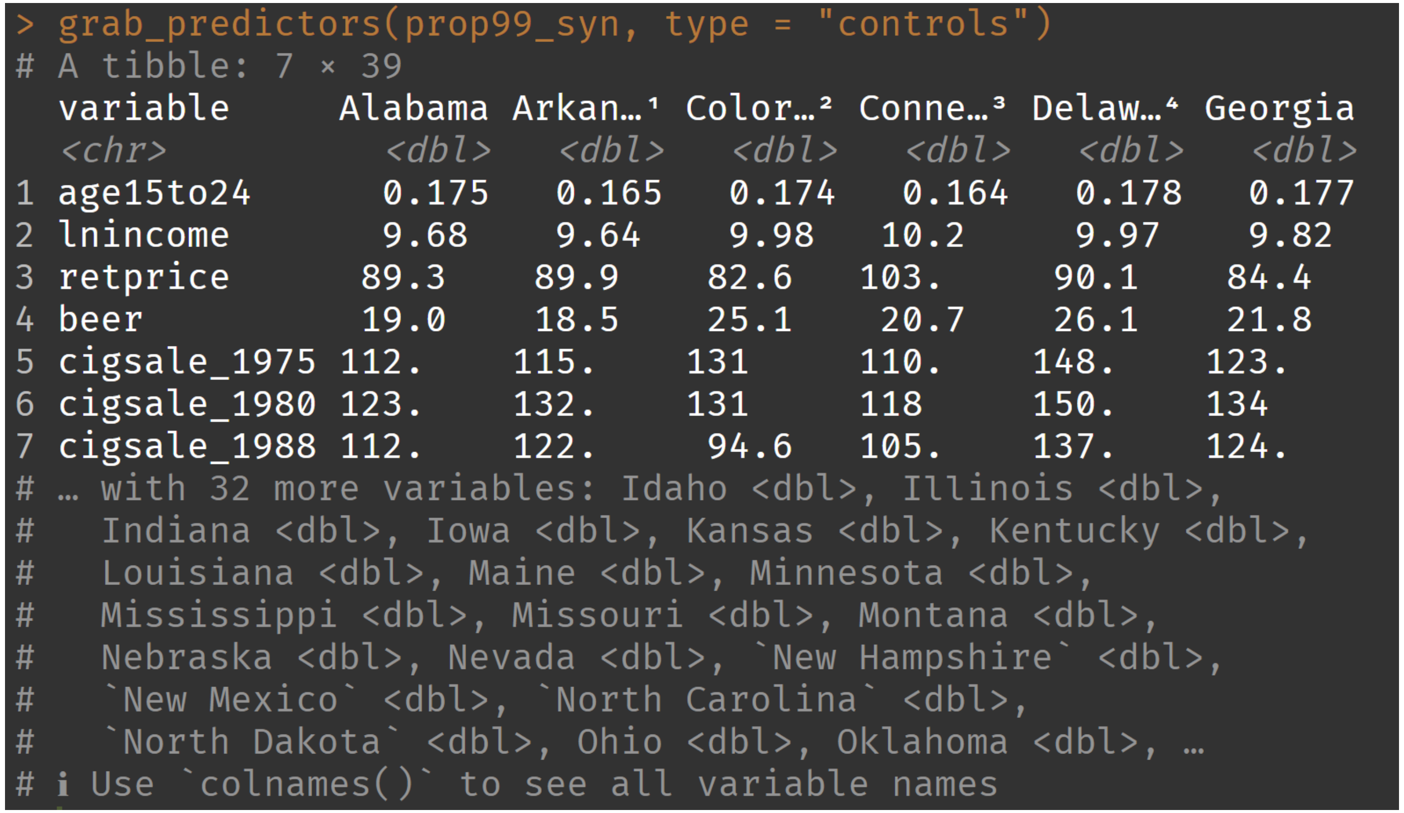


```r
# Now, generate the aggregate predictors used to estimate
# the weights
prop99_syn <-
  prop99_syn |>
  generate_predictor(
    time_window = 1980:1988,
    lnincome = mean(lnincome, na.rm = TRUE),
    retprice = mean(retprice, na.rm = TRUE),
    age15to24 = mean(age15to24, na.rm = TRUE)
  ) |>
  generate_predictor(
    time_window = 1984:1988,
    beer = mean(beer, na.rm = TRUE)
  ) |>
  generate_predictor(
    time_window = 1975,
    cigsale_1975 = cigsale
  ) |>
  generate_predictor(
    time_window = 1980,
    cigsale_1980 = cigsale
  ) |>
  generate_predictor(
    time_window = 1988,
    cigsale_1988 = cigsale
  )
```

Inspecting predictors

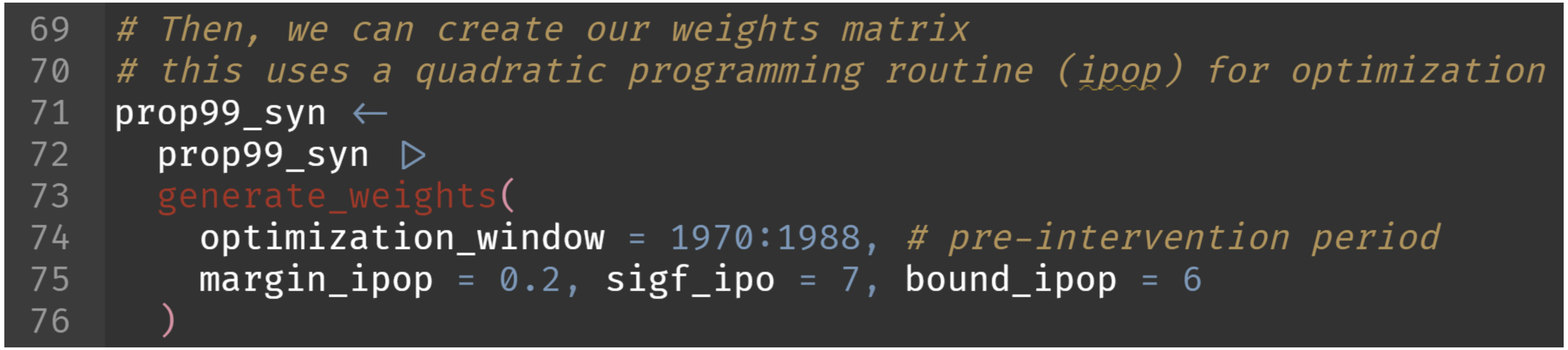


```
> grab_predictors(prop99_syn)
# A tibble: 7 × 2
  variable     California
  <chr>             <dbl>
1 age15to24         0.174
2 lnincome         10.1
3 retprice         89.4
4 beer             24.3
5 cigsale_1975    127.
6 cigsale_1980    120.
7 cigsale_1988     90.1
```



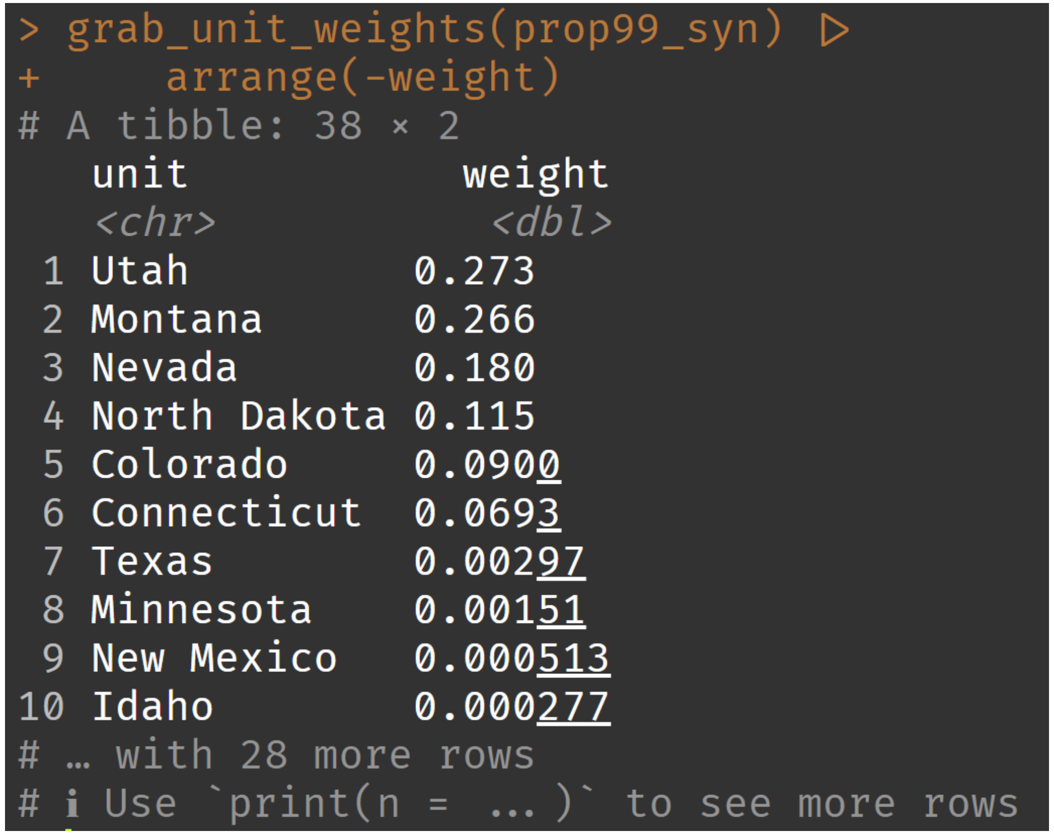
```
> grab_predictors(prop99_syn, type = "controls")
# A tibble: 7 × 39
  variable     Alabama Arkan…¹ Color…² Conne…³ Delaw…⁴ Georgia
  <chr>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 age15to24      0.175   0.165   0.174   0.164   0.178   0.177
2 lnincome       9.68    9.64    9.98   10.2     9.97    9.82
3 retprice      89.3    89.9    82.6   103.     90.1    84.4
4 beer          19.0    18.5    25.1    20.7    26.1    21.8
5 cigsale_1975 112.    115.    131     110.    148.    123.
6 cigsale_1980 123.    132.    131     118     150.    134
7 cigsale_1988 112.    122.     94.6   105.    137.    124.
# … with 32 more variables: Idaho <dbl>, Illinois <dbl>,
#   Indiana <dbl>, Iowa <dbl>, Kansas <dbl>, Kentucky <dbl>,
#   Louisiana <dbl>, Maine <dbl>, Minnesota <dbl>,
#   Mississippi <dbl>, Missouri <dbl>, Montana <dbl>,
#   Nebraska <dbl>, Nevada <dbl>, `New Hampshire` <dbl>,
#   `New Mexico` <dbl>, `North Carolina` <dbl>,
#   `North Dakota` <dbl>, Ohio <dbl>, Oklahoma <dbl>, …
# ℹ Use `colnames()` to see all variable names
```

Estimating weights (magic!)

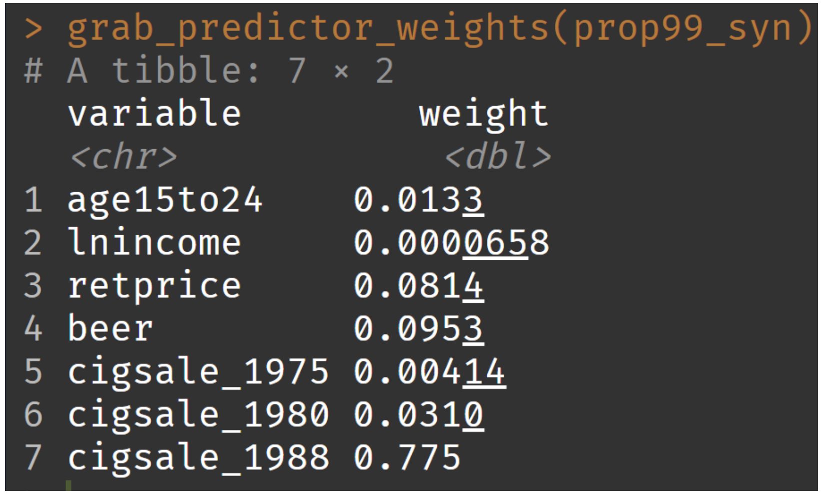


```
# Then, we can create our weights matrix
# this uses a quadratic programming routine (ipop) for optimization
prop99_syn ←
  prop99_syn ▷
  generate_weights(
    optimization_window = 1970:1988, # pre-intervention period
    margin_ipop = 0.2, sigf_ipo = 7, bound_ipop = 6
  )
```

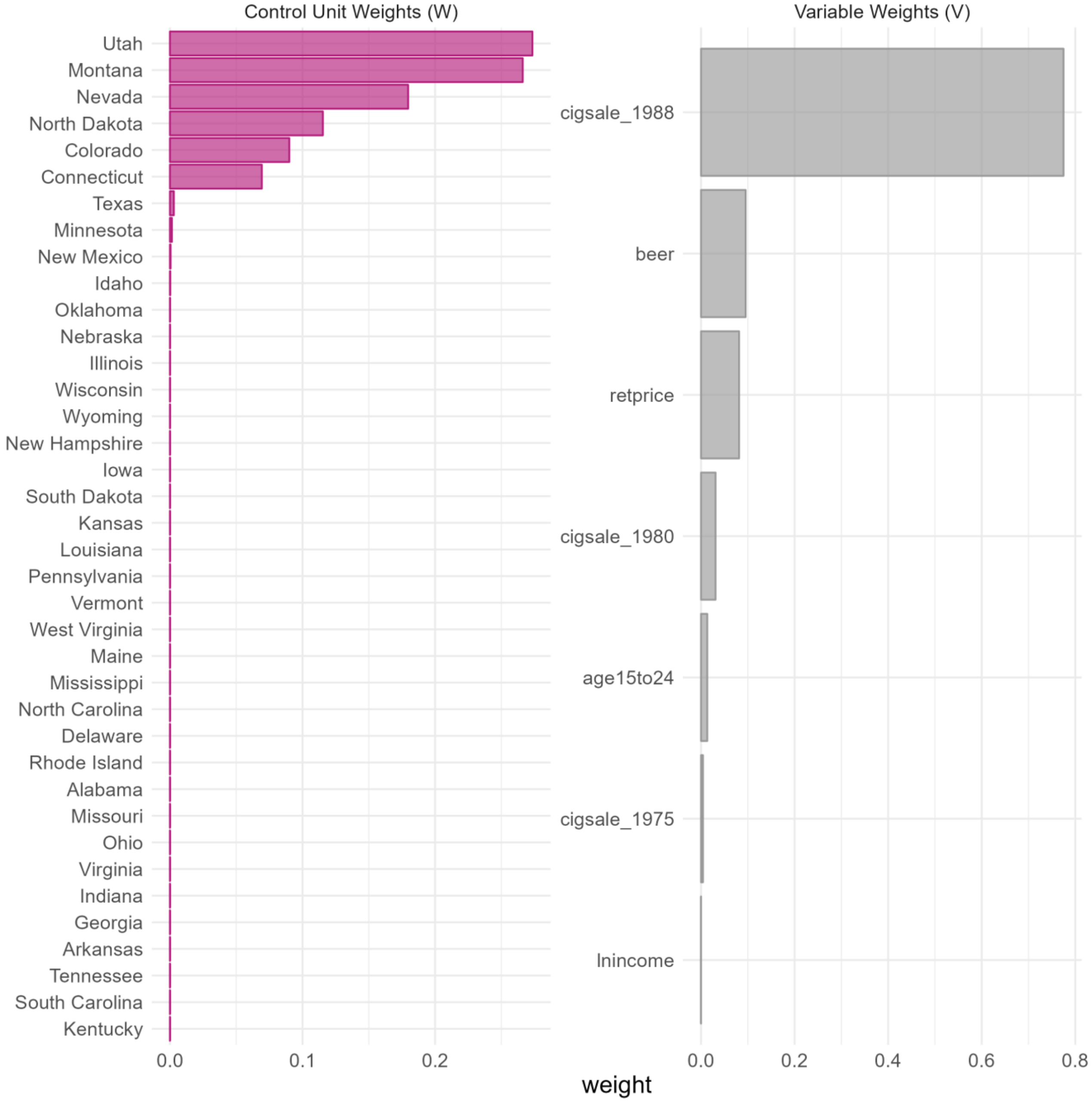
Inspecting weights



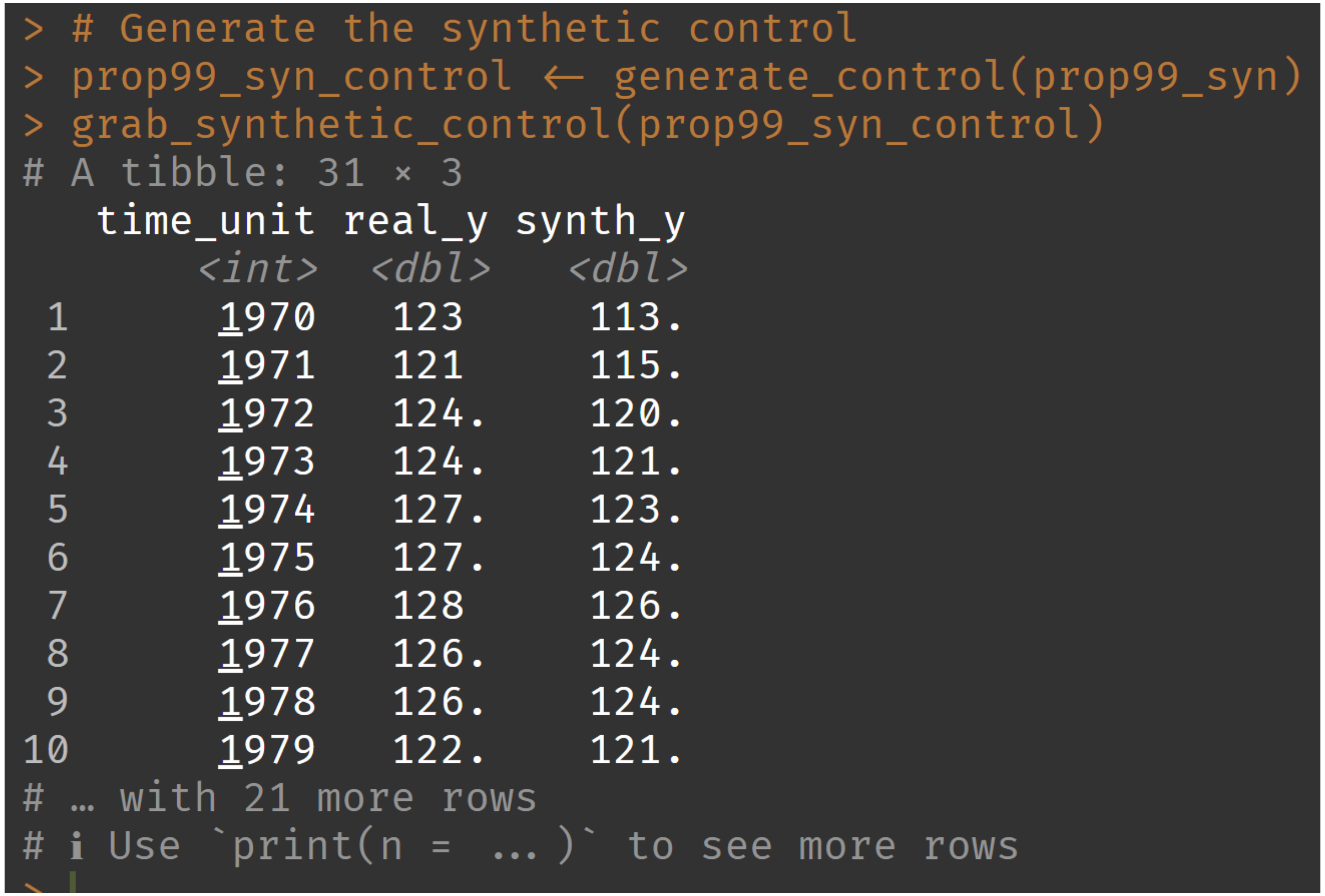
```
> grab_unit_weights(prop99_syn) ▷
+     arrange(-weight)
# A tibble: 38 × 2
   unit           weight
   <chr>           <dbl>
 1 Utah         0.273   
 2 Montana      0.266   
 3 Nevada       0.180   
 4 North Dakota 0.115   
 5 Colorado     0.0900  
 6 Connecticut  0.0693  
 7 Texas        0.00297 
 8 Minnesota    0.00151 
 9 New Mexico   0.000513
10 Idaho        0.000277
# … with 28 more rows
# ℹ Use `print(n = ...)` to see more rows
```



```
> grab_predictor_weights(prop99_syn)
# A tibble: 7 × 2
  variable        weight
  <chr>            <dbl>
1 age15to24    0.0133  
2 lnincome     0.0000658
3 retprice     0.0814  
4 beer         0.0953  
5 cigsale_1975 0.00414 
6 cigsale_1980 0.0310  
7 cigsale_1988 0.775   
```

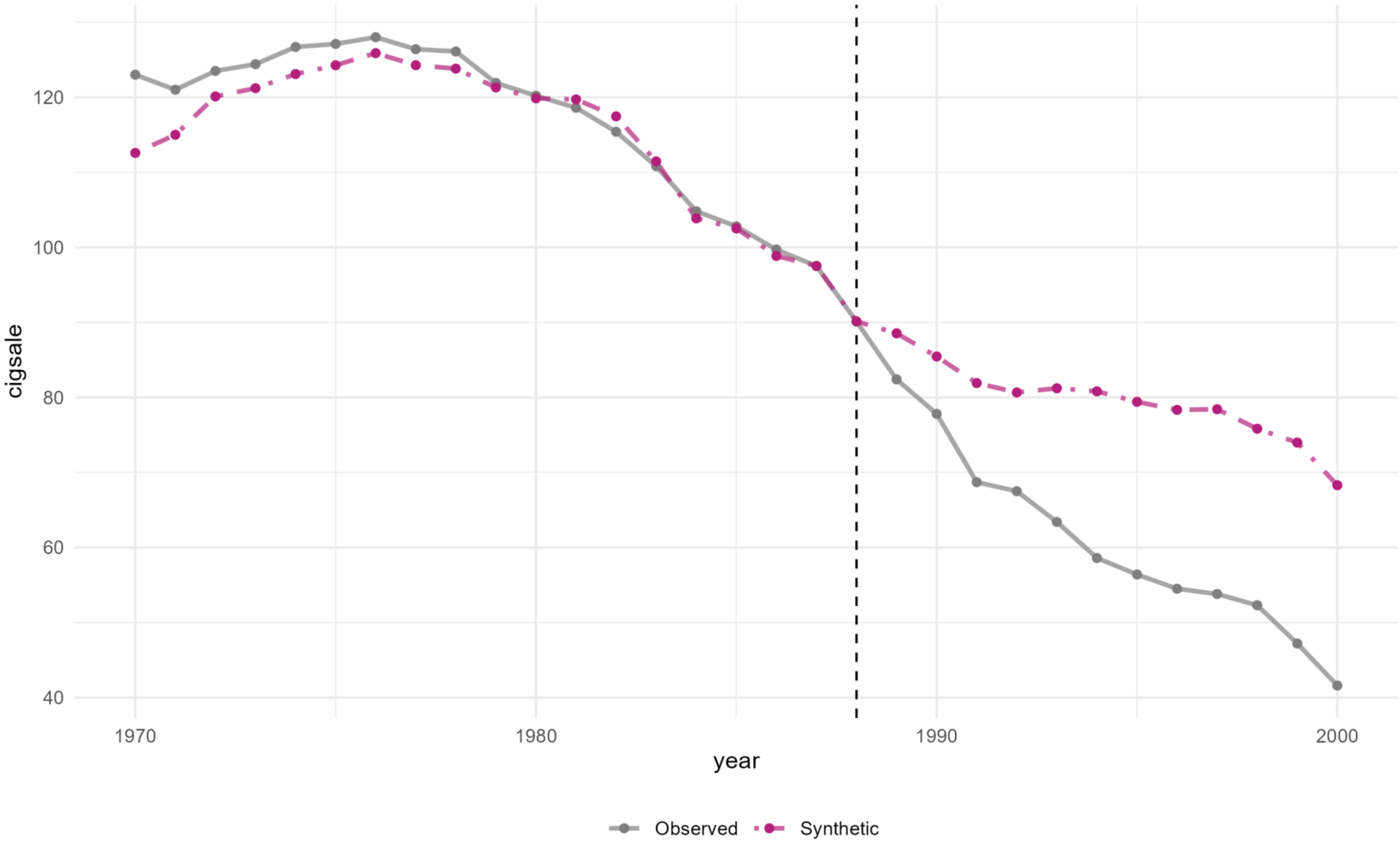

Control Unit Weights (W)
Variable Weights (V)
Utah
Montana
Nevada
North Dakota
Colorado
Connecticut
Texas
Minnesota
New Mexico
Idaho
Oklahoma
Nebraska
Illinois
Wisconsin
Wyoming
New Hampshire
Iowa
South Dakota
Kansas
Louisiana
Pennsylvania
Vermont
West Virginia
Maine
Mississippi
North Carolina
Delaware
Rhode Island
Alabama
Missouri
Ohio
Virginia
Indiana
Georgia
Arkansas
Tennessee
South Carolina
Kentucky
0.0
0.1
0.2
cigsale_1988
beer
retprice
cigsale_1980
age15to24
cigsale_1975
lnincome
0.0
0.2
0.4
0.6
0.8
weight

Creating synthetic control



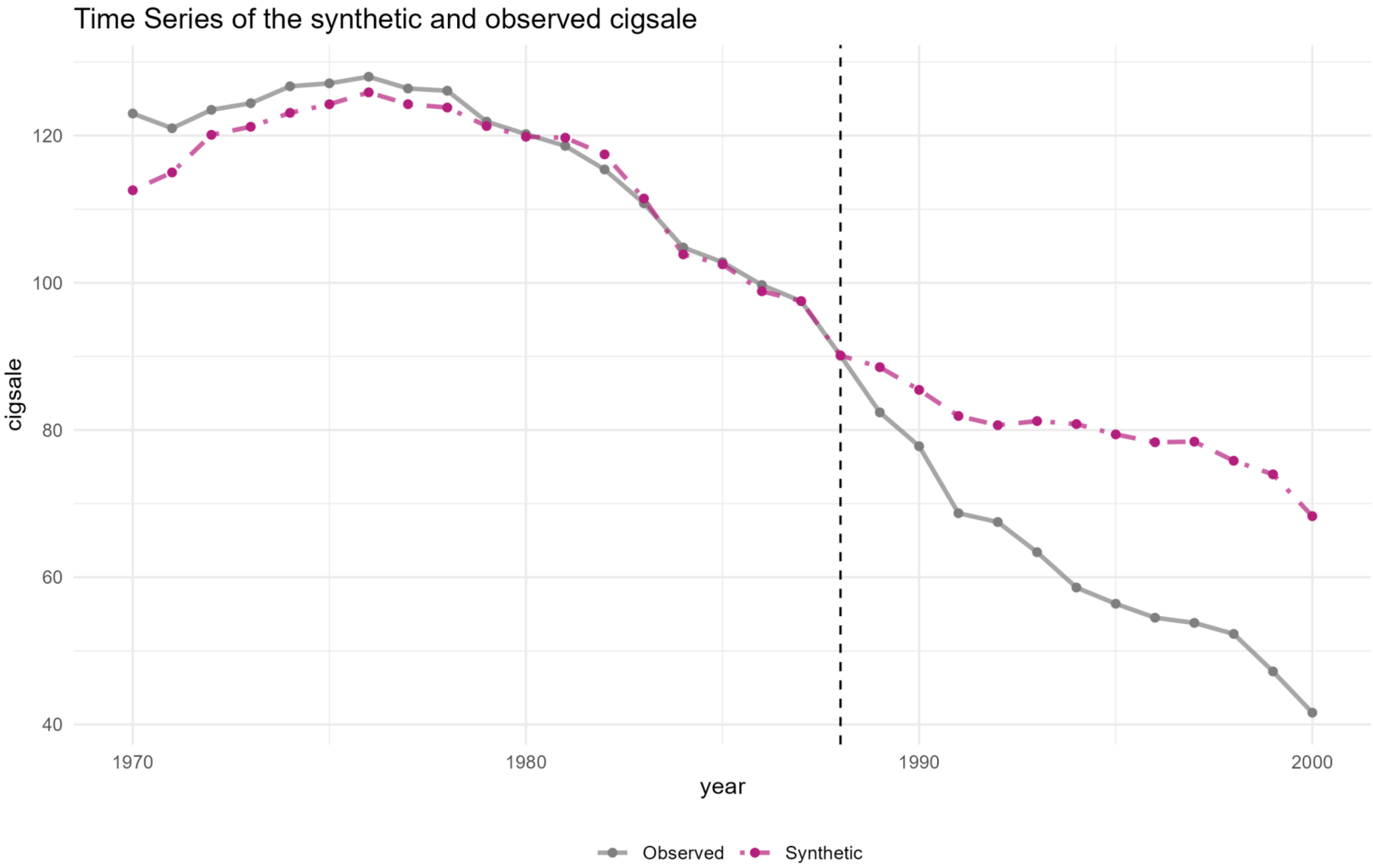
```
> # Generate the synthetic control
> prop99_syn_control ← generate_control(prop99_syn)
> grab_synthetic_control(prop99_syn_control)
# A tibble: 31 × 3
   time_unit real_y synth_y
       <int>  <dbl>   <dbl>
 1      1970   123     113.
 2      1971   121     115.
 3      1972   124.    120.
 4      1973   124.    121.
 5      1974   127.    123.
 6      1975   127.    124.
 7      1976   128     126.
 8      1977   126.    124.
 9      1978   126.    124.
10      1979   122.    121.
# … with 21 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Time Series of the synthetic and observed cigsale



Dashed line denotes the time of the intervention.

Inference

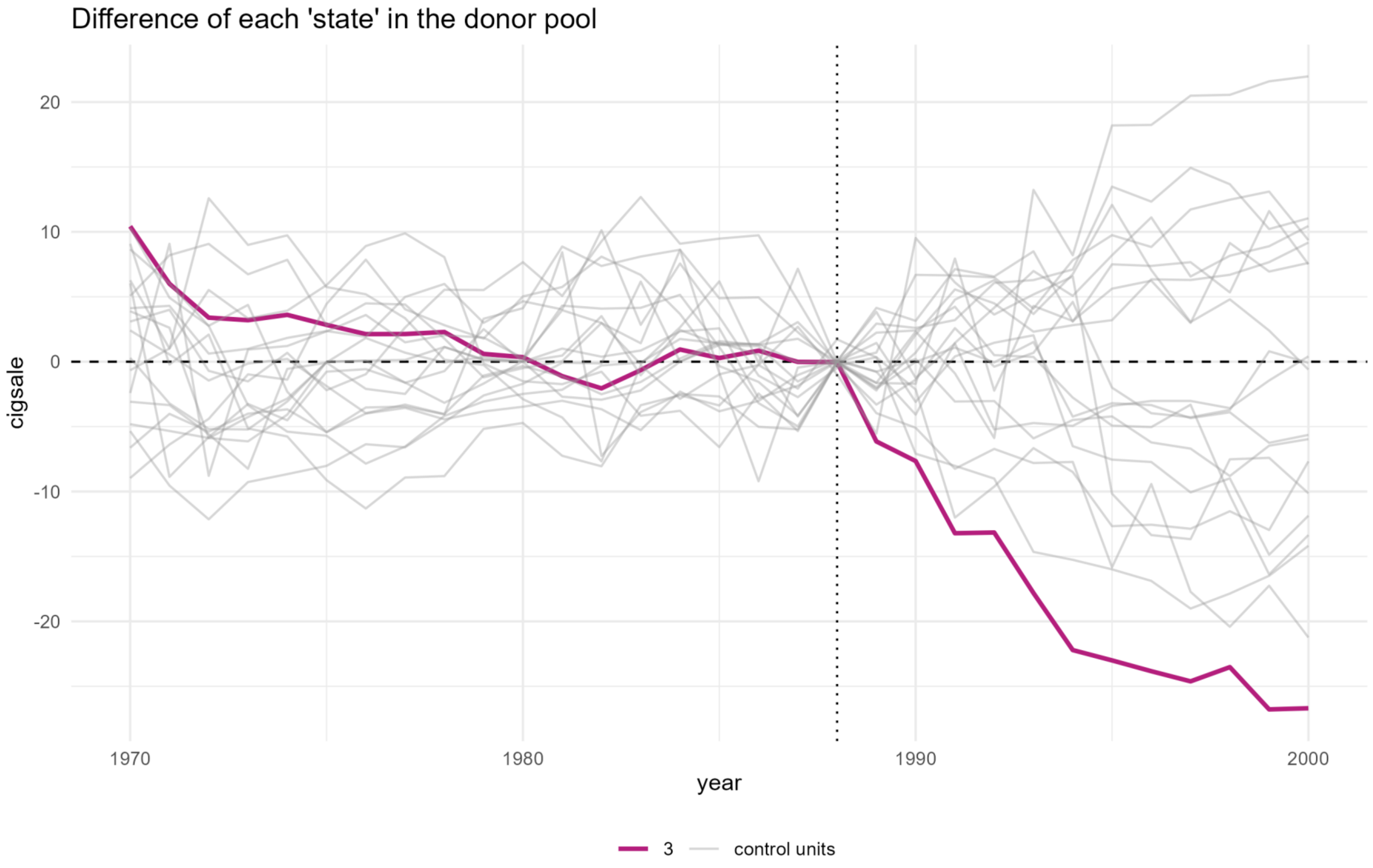


Time Series of the synthetic and observed cigsale

Dashed line denotes the time of the intervention.

How to quantify uncertainty?

- Most common method: **permutation test**
- Apply synthetic control method many times, once for each unit in the donor pool
- These units have no intervention effect
- Create reference/null distribution of Y_t^0
- Compare target unit's counterfactual to reference distribution
- Obtain a permutation p-value



Pruned all placebo cases with a pre-period RMSPE exceeding two times the treated unit's pre-period RMSPE.

Choices, choices …

There are many choices

- Which units in the donor pool?
- Which control variables?
- What should my weights optimize?
- How many nonzero unit weights should I get?
- What settings do I give to the nonlinear optimizer?

“researcher degrees of freedom”

There are many choices

- These choices influence your causal estimate $\widehat{CE}_t$
- Make good choices ☺
- Think of your causal estimate as “conditional” on the “model” (choices)
- Investigate the impact of different choices through robustness checks / sensitivity analysis

Leave-one-unit-out validation

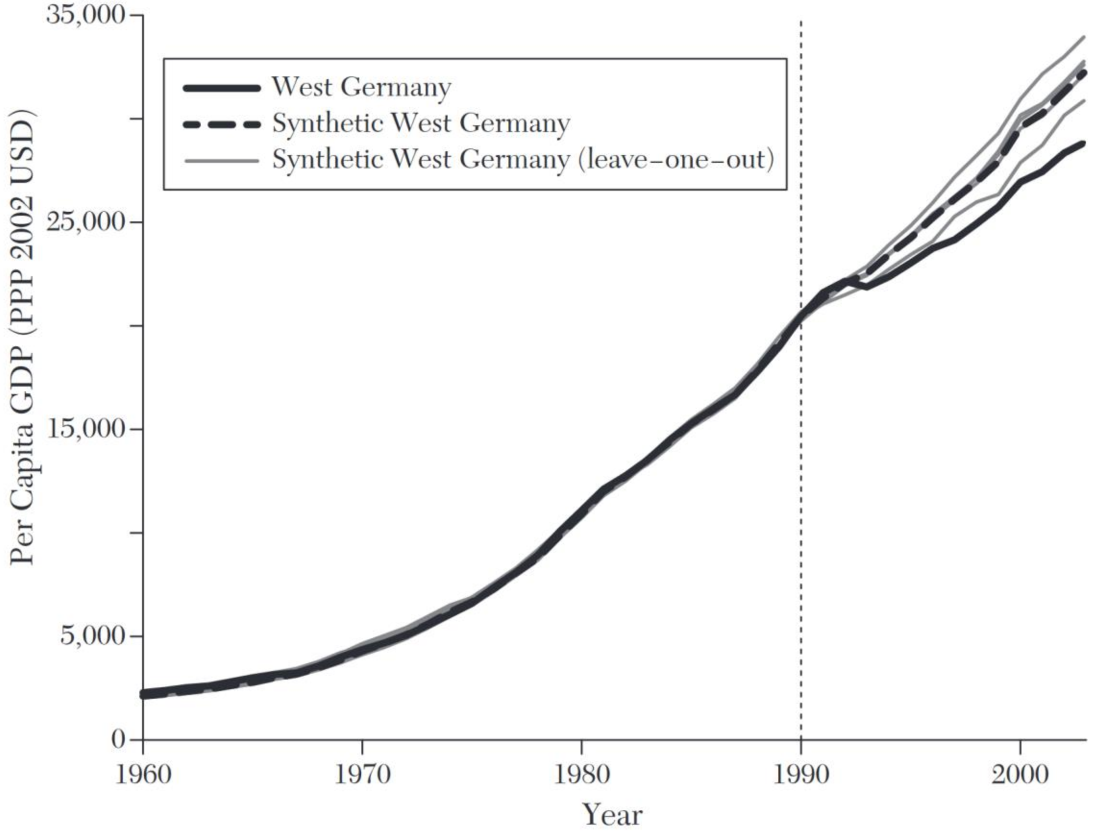


Figure 4. Leave-one-out Estimates of the Effect of the 1990 German Reunification

More of this in the practical

Practical: tidysynth, inference, robustness

Work in your groups!
Take a break from 14:30 to 14:45

Break