ARE 213 PS 2b

S. Sung, H. Husain, T. Woolley, A. Watt

2021-11-23

Contents

Problem 1

We first estimate an event study specification.

Part (a)

First determine the minimum and maximum event time values that you can estimate in this data set. Code up a separate event time indicator for each possible value of event time in the data set. Estimate an event study regression using all the event time indicators. What happens?

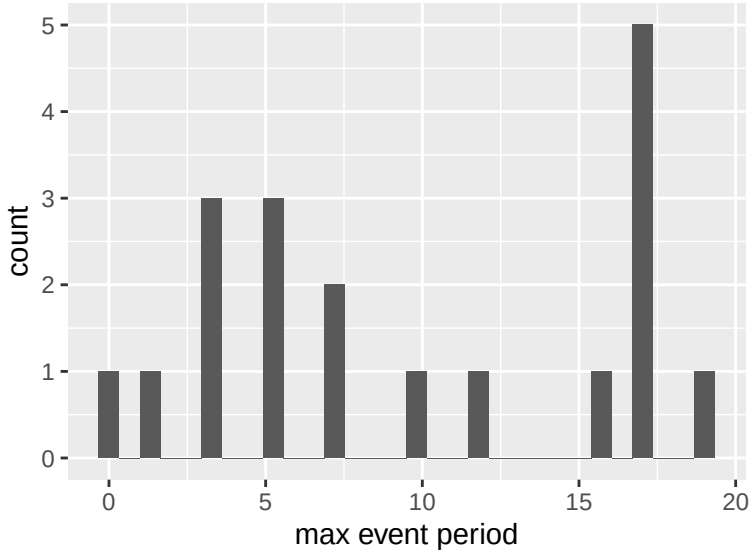
Table 1 lists the minimum and maximum event times that exist in the data for states that enacted a primary seat belt law in our study period (1981-2003).

Table 1: Maximum & Minimum Event Time Values

Max j	Min j
19	-22

Notice from Figure 1 that we have a wide range of maximum event times across our panel of states. One state even has a maximum event time of 0 – meaning they only added primary seat belt laws in the last year of our panel (2003). This means we will have an unbalanced panel if we run a regression on all possible event time dummies. In fact, we will still have an unbalance panel if we create max and min event time bins to aggregate early and late periods (an indicator for a event times greater than 5 and an indicator for all event times less than -5), we will still have an unbalanced panel.

Figure 1: Treated States, maximum event time histogram



To estimate the event study treatment effects, corresponding regression equation is:

$$Y_{st} = \alpha + \sum_{j=min\_t}^{max\_t} \tau_j D_{jst} + \gamma_s + \delta_t + \varepsilon_{st} + u_{st}$$

Note that we are estimating the regression with state and year fixed effects. In practice, we would want to omit a specific event time indicator so all our treatment effects are measured with respect to that event

time. If we keep all our indicators, then R will implicitly choose which to event time indicator to omit for us because the event time dummies, along with state and year fixed effects, are colinear.

We can see column (1) of Table 2 that the indicator for event time +19 was omitted for the regression. However, for interpretability, we'd rather have the treatment effects relative to a period closer to the year of initial treatment.

Part (b)

Estimate another event study regression using all the event time indicators save one that you choose to omit. Generate a plot of the event study coefficients.

We have chosen to omit the event time -1 from the regression so the other event time inidicator coefficients can be interpreted as relative to the year immediately before the passage of the primary seat belt law. Column (2) of Table 2 shows that the -1 event time period was omitted, and we we see that all of the treatment effects occuring before event time -1 are not significantly different from zero, whereas all the event time coefficient estimates for after event time -1 are negative and all are signficantly less than zero starting with event period 5.

Table 2: Event Study Regressions

	Log(Fatality per Population)	
	Event Study a	Event Study b
	(1)	(2)
‘-22_ET‘	0.1385 (0.1578)	−0.1545 (0.1137)
‘-21_ET‘	0.4172*** (0.1372)	0.1242 (0.0828)
‘-20_ET‘	0.3697*** (0.1371)	0.0767 (0.0826)
‘-19_ET‘	0.3091** (0.1231)	0.0162 (0.0568)
‘-18_ET‘	0.2851** (0.1231)	−0.0079 (0.0567)
‘-17_ET‘	0.3460*** (0.1191)	0.0531 (0.0475)
‘-16_ET‘	0.3543*** (0.1191)	0.0613 (0.0477)
‘-15_ET‘	0.3526*** (0.1176)	0.0597 (0.0436)
‘-14_ET‘	0.3113*** (0.1175)	0.0184 (0.0440)
‘-13_ET‘	0.3412*** (0.1175)	0.0482 (0.0435)
‘-12_ET‘	0.3161*** (0.1169)	0.0231 (0.0425)
‘-11_ET‘	0.3235*** (0.1168)	0.0305 (0.0423)
‘-10_ET‘	0.3105*** (0.1163)	0.0176 (0.0411)
‘-9_ET‘	0.2946** (0.1162)	0.0017 (0.0412)
‘-8_ET‘	0.3096*** (0.1162)	0.0166 (0.0408)
‘-7_ET‘	0.3313*** (0.1161)	0.0383 (0.0411)
‘-6_ET‘	0.3385*** (0.1156)	0.0455 (0.0397)
‘-5_ET‘	0.3155*** (0.1144)	0.0225 (0.0368)
‘-4_ET‘	0.3268*** (0.1143)	0.0338 (0.0369)
‘-3_ET‘	0.3225*** (0.1138)	0.0295 (0.0359)
‘-2_ET‘	0.2981*** (0.1137)	0.0051 (0.0363)
‘-1_ET‘	0.2929** (0.1137)	
‘0_ET‘	0.2623** (0.1135)	−0.0306 (0.0363)
‘1_ET‘	0.2484** (0.1136)	−0.0445 (0.0365)
‘2_ET‘	0.2464** (0.1139)	−0.0466 (0.0375)
‘3_ET‘	0.2466** (0.1134)	−0.0463 (0.0375)
‘4_ET‘	0.2278** (0.1144)	−0.0651 (0.0397)
‘5_ET‘	0.2127* (0.1138)	−0.0803** (0.0401)
‘6_ET‘	0.1965* (0.1154)	−0.0965** (0.0432)
‘7_ET‘	0.2173* (0.1149)	−0.0756* (0.0438)
‘8_ET‘	0.1612 (0.1165)	−0.1317*** (0.0471)
‘9_ET‘	0.1790 (0.1165)	−0.1140** (0.0471)
‘10_ET‘	0.1772 (0.1162)	−0.1157** (0.0474)
‘11_ET‘	0.1594 (0.1174)	−0.1336*** (0.0494)
‘12_ET‘	0.1443 (0.1170)	−0.1487*** (0.0500)
‘13_ET‘	0.1522 (0.1185)	−0.1408*** (0.0526)
‘14_ET‘	0.1574 (0.1185)	−0.1355** (0.0533)
‘15_ET‘	0.1504 (0.1184)	−0.1425*** (0.0531)
‘16_ET‘	0.0973 (0.1181)	−0.1956*** (0.0533)
‘17_ET‘	0.0842 (0.1183)	−0.2087*** (0.0576)
‘18_ET‘	0.0144 (0.1520)	−0.2785** (0.1135)
‘19_ET‘		−0.2929** (0.1137)
Constant	−1.4815*** (0.1158)	−1.1885*** (0.0387)
Chose dummy to omit	No	Yes
Observations	1,104	1,104
R^2	0.9111	0.9111
Adjusted R^2	0.9013	0.9013
Note:		*p<0.1; **p<0.05; ***p<0.01

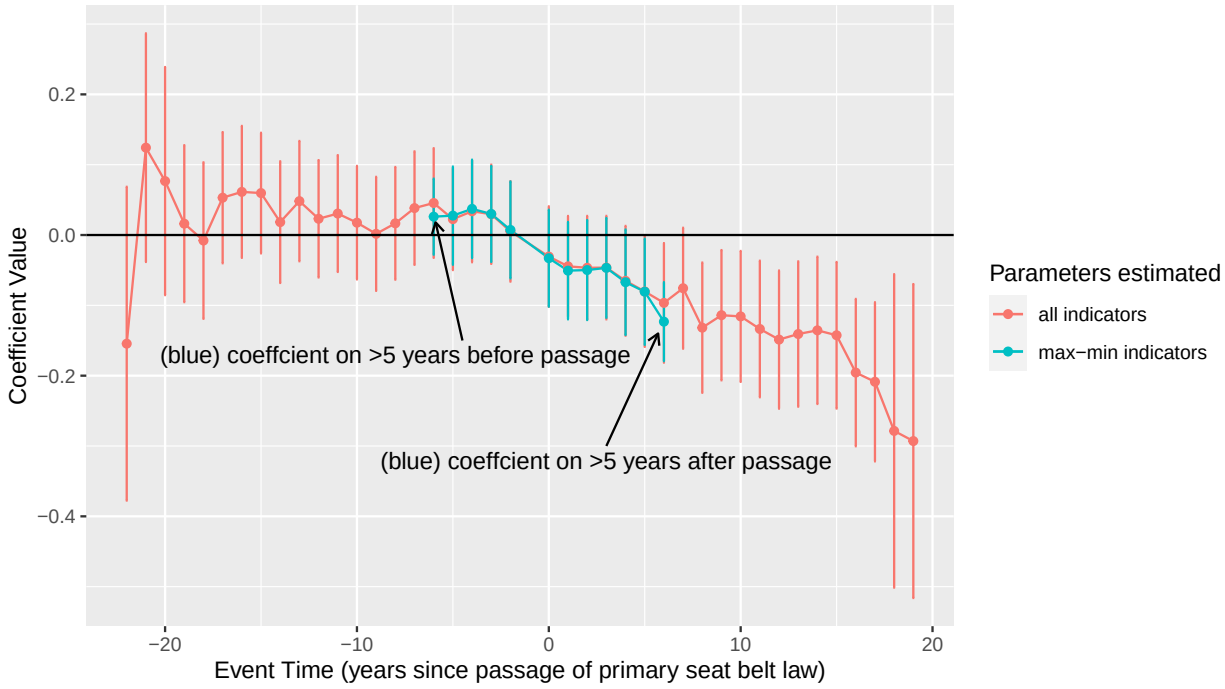
Part (c)

Create minimum and maximum event time indicators that correspond to bins of event time < -5 and event time > 5 respectively. Appropriately specify and estimate an event study regression using these min and max event time indicators. Generate a plot of the event study coefficients. Explain which specification you prefer, this one or the one in part (b).

Table 3: Event Study Regression with Threshold Indicators

	Log(Fatality per Population)
	Event Study c
below_ET	0.0261 (0.0275)
'-5_ET'	0.0276 (0.0355)
'-4_ET'	0.0372 (0.0356)
'-3_ET'	0.0300 (0.0347)
'-2_ET'	0.0073 (0.0350)
'0_ET'	−0.0329 (0.0350)
'1_ET'	−0.0506 (0.0352)
'2_ET'	−0.0498 (0.0362)
'3_ET'	−0.0467 (0.0361)
'4_ET'	−0.0671* (0.0381)
'5_ET'	−0.0806** (0.0384)
above_ET	−0.1232*** (0.0285)
Constant	−1.1828*** (0.0373)
Chose dummy to omit	Yes
Agg. Threshold Indicators	Yes
Observations	1,127
R^2	0.9090
Adjusted R^2	0.9019
Note:	*p<0.1; **p<0.05; ***p<0.01
	below_ and above_ variables are aggregate indicators for all event times that are below -5 and above 5, respectively.

Figure 2: (red) Plot of all possible event-time coefficients and 95% confidence intervals from Table 2; (blue) plot of event-time coefficients and 95% confidence intervals for event-times from years before the passage of the primary seat belt law to 5 years after and coefficents for aggregate event-time indicators of more than 5 years before and more than 5 years after the law passed, plotted at x-values of -6 and 6, respectively (from Table 3)



Part (d)

What happens to your estimates from part (b) if you exclude the "pure control" states from your sample? What about if you exclude the pure controls in part (c)?

Table 4: Event Study Regressions with and without Pure Control States

	Log(Fatality per Population)			
	All indicators	All indicators	Min-max indicators	Min-max indicators
	(1)	(2)	(3)	(4)
'-22_ET'	−0.1545 (0.1137)	−0.4460*** (0.1608)		
'-21_ET'	0.1242 (0.0828)	−0.1641 (0.1383)		
'-20_ET'	0.0767 (0.0826)	−0.1925 (0.1333)		
'-19_ET'	0.0162 (0.0568)	−0.2338** (0.1165)		
'-18_ET'	−0.0079 (0.0567)	−0.2606** (0.1117)		
'-17_ET'	0.0531 (0.0475)	−0.1681 (0.1023)		
'-16_ET'	0.0613 (0.0477)	−0.1639* (0.0985)		
'-15_ET'	0.0597 (0.0436)	−0.1401 (0.0905)		
'-14_ET'	0.0184 (0.0440)	−0.1808** (0.0872)		
'-13_ET'	0.0482 (0.0435)	−0.1329 (0.0811)		
'-12_ET'	0.0231 (0.0425)	−0.1496* (0.0768)		
'-11_ET'	0.0305 (0.0423)	−0.1295* (0.0720)		
'-10_ET'	0.0176 (0.0411)	−0.1295* (0.0667)		
'-9_ET'	0.0017 (0.0412)	−0.1296** (0.0635)		
'-8_ET'	0.0166 (0.0408)	−0.1028* (0.0577)		
'-7_ET'	0.0383 (0.0411)	−0.0643 (0.0553)		
'-6_ET'	0.0455 (0.0397)	−0.0442 (0.0495)		
below_ET			0.0261 (0.0275)	−0.0165 (0.0311)
'-5_ET'	0.0225 (0.0368)	−0.0422 (0.0447)	0.0276 (0.0355)	0.0114 (0.0342)
'-4_ET'	0.0338 (0.0369)	−0.0222 (0.0420)	0.0372 (0.0356)	0.0180 (0.0343)
'-3_ET'	0.0295 (0.0359)	−0.0014 (0.0375)	0.0300 (0.0347)	0.0222 (0.0326)
'-2_ET'	0.0051 (0.0363)	−0.0183 (0.0372)	0.0073 (0.0350)	−0.0015 (0.0336)
'0_ET'	−0.0306 (0.0363)	−0.0184 (0.0353)	−0.0329 (0.0350)	−0.0296 (0.0336)
'1_ET'	−0.0445 (0.0365)	−0.0219 (0.0346)	−0.0506 (0.0352)	−0.0487 (0.0331)
'2_ET'	−0.0466 (0.0375)	−0.0061 (0.0378)	−0.0498 (0.0362)	−0.0416 (0.0348)
'3_ET'	−0.0463 (0.0375)	0.0002 (0.0385)	−0.0467 (0.0361)	−0.0406 (0.0347)
'4_ET'	−0.0651 (0.0397)	0.0018 (0.0436)	−0.0671* (0.0381)	−0.0555 (0.0367)
'5_ET'	−0.0803** (0.0401)	−0.0022 (0.0456)	−0.0806** (0.0384)	−0.0638* (0.0376)
'6_ET'	−0.0965** (0.0432)	0.0028 (0.0520)		
'7_ET'	−0.0756* (0.0438)	0.0349 (0.0549)		
'8_ET'	−0.1317*** (0.0471)	−0.0104 (0.0618)		
'9_ET'	−0.1140** (0.0471)	0.0231 (0.0658)		
'10_ET'	−0.1157** (0.0474)	0.0219 (0.0689)		
'11_ET'	−0.1336*** (0.0494)	0.0361 (0.0753)		
'12_ET'	−0.1487*** (0.0500)	0.0449 (0.0787)		
'13_ET'	−0.1408*** (0.0526)	0.0874 (0.0858)		
'14_ET'	−0.1355** (0.0533)	0.0965 (0.0904)		
'15_ET'	−0.1425*** (0.0531)	0.0924 (0.0944)		
'16_ET'	−0.1956*** (0.0533)	0.0714 (0.0984)		
'17_ET'	−0.2087*** (0.0576)	0.0512 (0.1007)		
'18_ET'	−0.2785** (0.1135)	0.0239 (0.1414)		
'19_ET'	−0.2929** (0.1137)			
above_ET			−0.1232*** (0.0285)	−0.0820** (0.0326)
Constant	−1.1885*** (0.0387)	−0.9387*** (0.1113)	−1.1828*** (0.0373)	−1.1331*** (0.0463)
Agg. Threshold Indicators	No	No	Yes	Yes
Include Pure Controls	Yes	No	Yes	No
Observations	1,104	414	1,127	437
R^2	0.9111	0.9273	0.9090	0.9224
Adjusted R^2	0.9013	0.9102	0.9019	0.9119
Note:				*p<0.1; **p<0.05; ***p<0.01

For the all-event-times-indicators regressions from part (b), we can see in Table 4 that for event-time indicators after the primary seat belt law is passed (event times ≥ 0), the coefficients change from being all negative and mostly significant to being all insignificant at the 0.1 level and mixed signs. Interestingly, the coefficients

on the event-time indicators for times before the passage of the law become significantly negative – indicating that the passage of the law might have caused an increase in traffic fatalities in states in the treatment group (i.e., the effect of treatment on the treated might be the opposite sign we expect). However, we also should note that the coefficient on event time 19 (19 years after the passage of the law) is dropped by the regression because of colinearity – without pure control states, the combination of state and year fixed effects with indicators for all but one event times becomes colinear.

Because we have omitted the coefficient on event time 19 and event time 0, we cannot readily interpret the coefficients in column (2) of Table 4.

If we focus on columns (3) and (4) from Table 4, we can see removing the pure control states (column 4) reduces the significance and magnitude of all our coefficients, but the last two coefficients for event time 5 ('5_ET') and the aggregate of all event times greater than 5 (above_ET) are still significant and negative. One way to interpret this is that the treatment effect on the treated is smaller than that of the average treatment effect. We also could hypothesize that the states that passed the primary seat belt laws during our study period were doing something other than passing the primary law and something different from the pure control states to reduce traffic fatalities, so our comparison within the treatment states (column 4) estimates a smaller effect than when we compare our treated states to the pure control states.

Part (e)

Overall, does the event study regression make you more confident or less confident that seat belt laws reduce fatalities (relative to the fixed effects results that you estimated on the last problem set)? Briefly explain.

Part (f*)

Building off the event study regression from part (c), estimate the interaction weighted event study estimator from Sun and Abraham (2020). As a reminder, the interacted event study regression takes the standard event time indicators (without any binning) and interacts each one with a cohort indicator (a cohort refers to a group of states that share the same date on which they were first treated). You then form the estimate for event time coefficient τ_j by averaging the estimates of the cohort-specific τ_j using the weights described in Sun and Abraham (2020).

Figure 3: Plot of Sun-and-Abraham(2020)-style cohort-by-event-time coefficients.

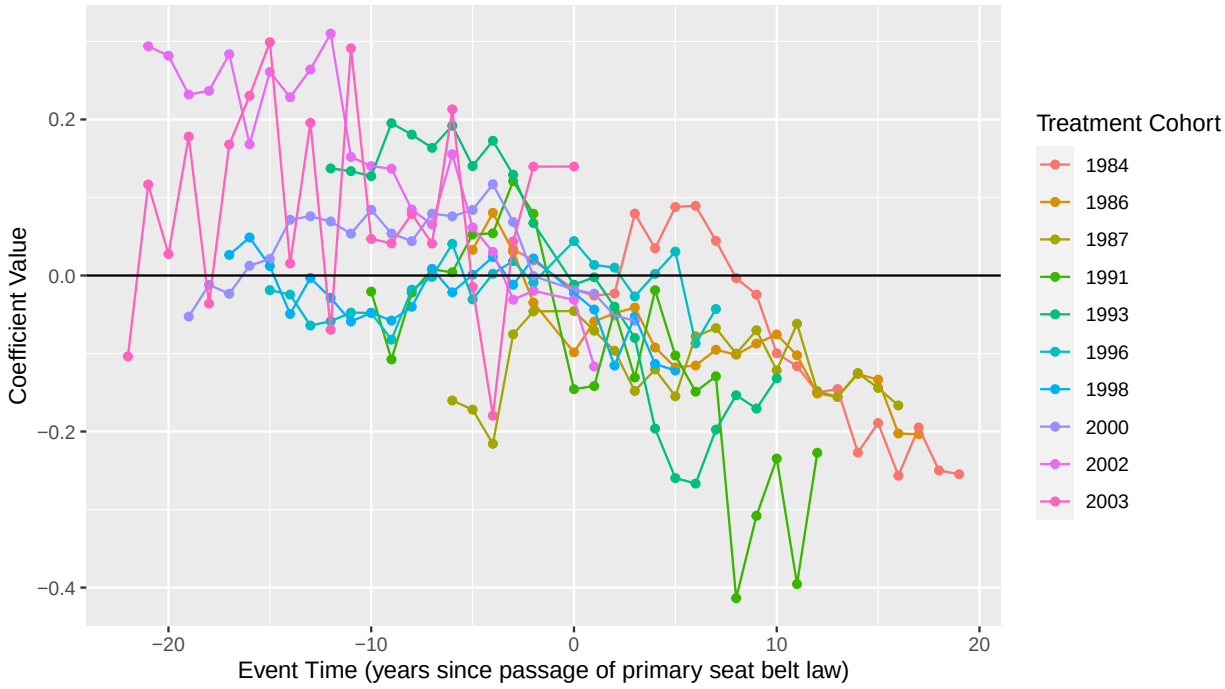
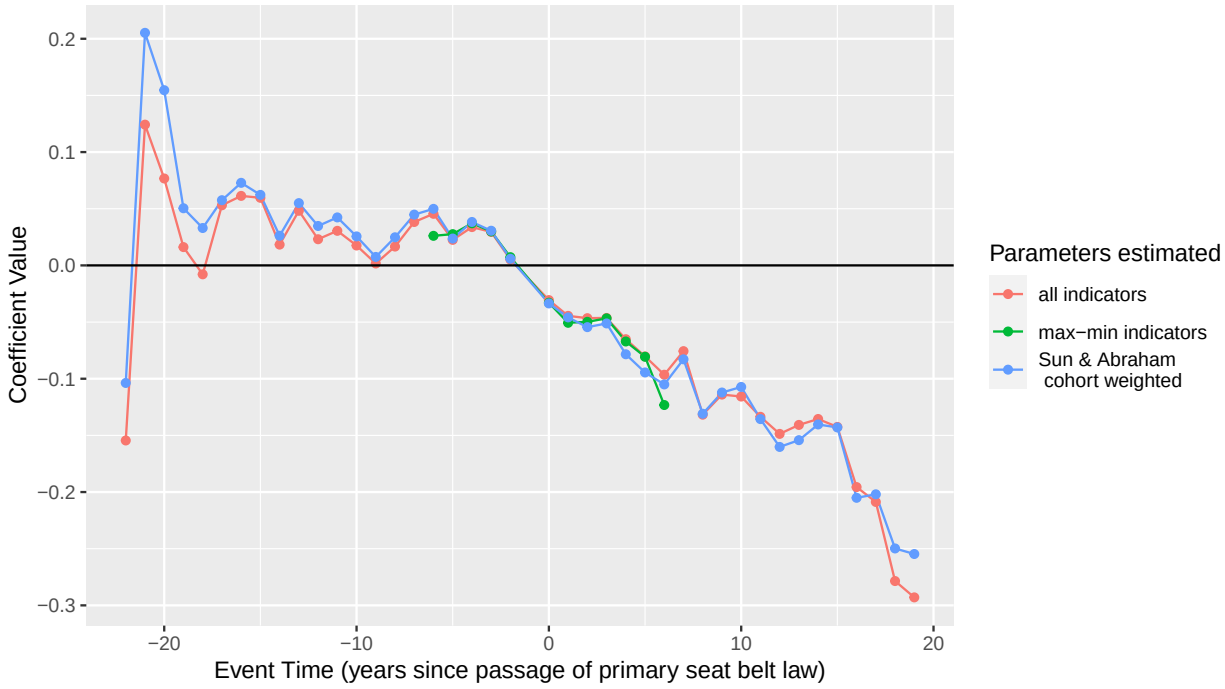


Figure 4: Plot of cohort-weighted avereage event-time coefficents – weighted using the number of states in each cohort



Problem 2

We now apply the synthetic control methods from Abadie et al (2010).

Part (a)

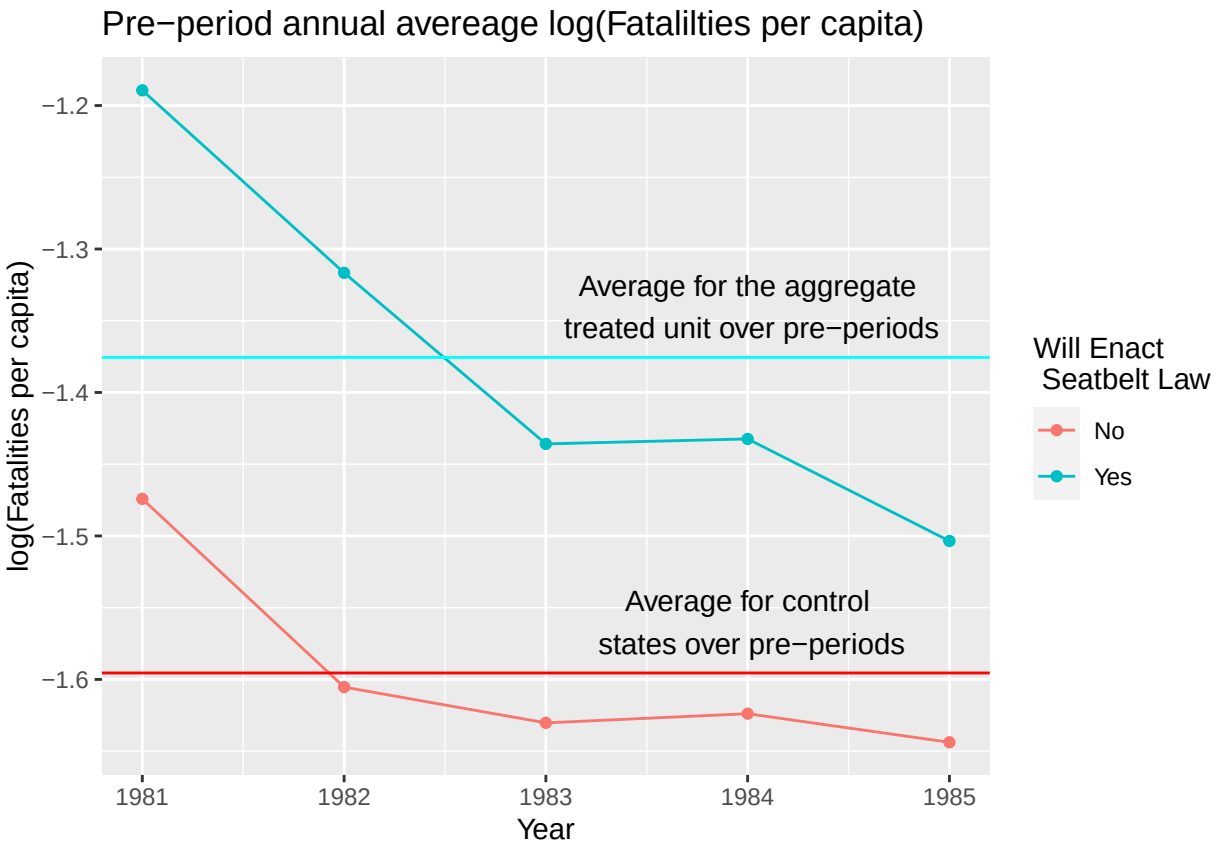
We created an aggregate "treatment" state (state number 99 or "TU") which combines the (population weighted) data from the first 4 states to have a primary seatbelt law (CT, IA, NM, TX). Please use this state as the "treatment" state in the synthetic control analysis.

—— **a.i**

Compare the average pre-period log traffic fatalities per capita of the TU site to that of the average of all the "control" states. Next, graph the pre-period log traffic fatalities by year for the pre-period for both the TU and the average of the control group. Interpret.

Table 5: Average Pre-period log(traffic fatalities per capita)

Aggregate Treatment State	Aggregate Control State
-1.375546	-1.595503


Pre−period annual avereage log(Fatalilties per capita)
log(Fatalities per capita)
−1.2
−1.3
−1.4
−1.5
−1.6
Average for the aggregate
treated unit over pre−periods
Average for control
states over pre−periods
1981
1982
1983
1984
1985
Year
Will Enact
Seatbelt Law
No
Yes

—— **a.ii**

Compare the dependent variable between the TU site and each control state for the year before the treatment. Which control state best matches the TU? Now compare this state's covariates with the TU covariates. Do they appear similar? What might this imply for in terms of using this state as the counterfactual state?

Table 6: Dependent variable in 1985 for treated (99) and control states

State ID	log(fatatilities per captia)	Absolute difference between this state and treated
99	-1.503526	0.0000000
47	-1.512943	0.0094169
44	-1.528011	0.0244846
2	-1.470088	0.0334382
40	-1.454592	0.0489344
14	-1.608368	0.1048414
8	-1.389040	0.1144866
23	-1.363415	0.1401115
11	-1.360526	0.1430004
15	-1.646608	0.1430818
28	-1.668061	0.1645347
22	-1.680986	0.1774596
39	-1.681263	0.1777369
42	-1.690492	0.1869652
24	-1.304958	0.1985685
31	-1.304588	0.1989382
5	-1.714055	0.2105290
19	-1.730826	0.2272997
3	-1.271162	0.2323642
38	-1.244085	0.2594411
43	-1.766389	0.2628626
48	-1.190119	0.3134070
46	-1.853388	0.3498619
33	-1.878806	0.3752799
36	-1.894083	0.3905567
27	-1.900066	0.3965393
21	-1.927277	0.4237503
12	-2.005066	0.5015393
26	-2.017831	0.5143042
17	-2.072787	0.5692603
37	-2.184871	0.6813450

We can see from Table 6 that state # 47 is the control state that is closest to the aggregate treatment state in the dependent variable.

Table 7: Dependent variable in 1985 for treated (99) and control states

	Control State 47	Aggregate Treatment State
state	47.0000	99.0000
college	0.1189	0.2339
beer	1.1100	1.5643
population	1906.8310	12009.3484
unemploy	13.0000	6.9451
totalvmt	12664.0000	104389.7266
precip	3.6542	2.4307
snow32	0.8333	0.1511
rural_speed	55.0000	55.0000
urban_speed	55.0000	55.0000
fat_pc	0.2203	0.2223

From Table 7, we can see that there are identical speed limits between state 47 and the aggregate treatment state. But most of the other covariates are very far apart in the distributions – total vehicle miles traveled is different by an order of magnitude and are in different sides of the distribution; both snow and precipitation are on different sides of their distributions; the unemployment rates, population levels, and college rates are very far apart as well. This makes it hard to believe that state #47 would be a good control for our aggregate treatment state.

We also generated percentiles for each of the covariates by state and year. Using the average pre-treatment percentiles of covariates, the two states do not look similar at all. For instance,

- college: WV is 1p and TU is 49p
- beer: WV is 15p and TU is 92p
- unemployment: WV is 99p and TU is 68p
- ln_tvmt: WV is 20p and TU is 87p
- ln_precip: WV is 64p and TU is 33p
- snow: WV is 61p and TU is 33p.

They only match percentiles for log rural speed and log urban speed (19 and 39).

Part (b)

Apply the synthetic control method using the available covariates and pre-treatment outcomes to construct a synthetic control group.

—— **b.i**

Discuss the synthetic control method including its benefits and potential drawbacks.

The synthetic controls approach circumvents the issue of not having plausible control group by constructing one by weighting untreated units selected by weighting covariates based on their predictability of the pre-treament outcome. Even though this method does not calculate a standard error for the treatment effect in the conventional sense, it does allow us to test the believability of the treatment result via a placebo test (as long as the number of potential control units is sufficiently large). Synthetic controls become virtually impossible to use without sufficiently many potential controls because the placebo test is vital to the interpretability of the treatment effect. Essentially, synthetic controls is only as good as the number of potential control units.

—— **b.ii**

Use the software package provided by Abadie et al to apply the synthetic control method. (You are free to use either Stata, Matlab, or R but answers will be provided in Stata and R only). Please be sure to state precisely what the command is doing and how you determined your preferred specification.

Can't use rural or urban speed limits as predictors because they do not change across the control units.

The Synth package calibrated the weights in a weighed average of the control states to fit the covariates of the aggregate treated unit. The weights were adjusted to minimize the sum of squared differences between the covariates of the control units and the covariates of the aggreagte treated unit, in the years before the treated unit was treated (1981 - `r max(preperiod_years)`).

The weights that minimize the sum of these distances are in Table 8.

Table 8: Synthetic Control Weights

Weight	State
0.005	AR
0.008	AZ
0.017	CO
0.267	FL
0.006	ID
0.011	IL
0.006	KS
0.006	KY
0.009	MA
0.005	ME
0.008	MN
0.007	MO
0.012	MS
0.008	MT
0.005	ND
0.006	NE
0.137	NH
0.005	NV
0.008	OH
0.007	PA
0.008	RI
0.006	SC
0.004	SD
0.006	TN
0.005	UT
0.077	VA
0.000	VT
0.021	WI
0.005	WV
0.324	WY

There are sums of squared differences for each covariate – instead of just taking a simple sum or average of these and picking control unit weights to minimize that sum, the Synth package creates weights for the covariates to create a weighted average of the sum of squared differences to minimize. The covariates were first used to predict the pre-treatment outcome for the treated unit, and covariates that have more power in predicting the outcome receive large weights, and are more important in generating the synthetic control weights. These weights are in Table 9 and are used in a weighted sum of the squared differences to find the optimal control unit weights in constructing the synthetic control. We can that the weight is fairly even across covariates except that precipitation has absorbed the weight that snow would get – probably because they are highly correlated.

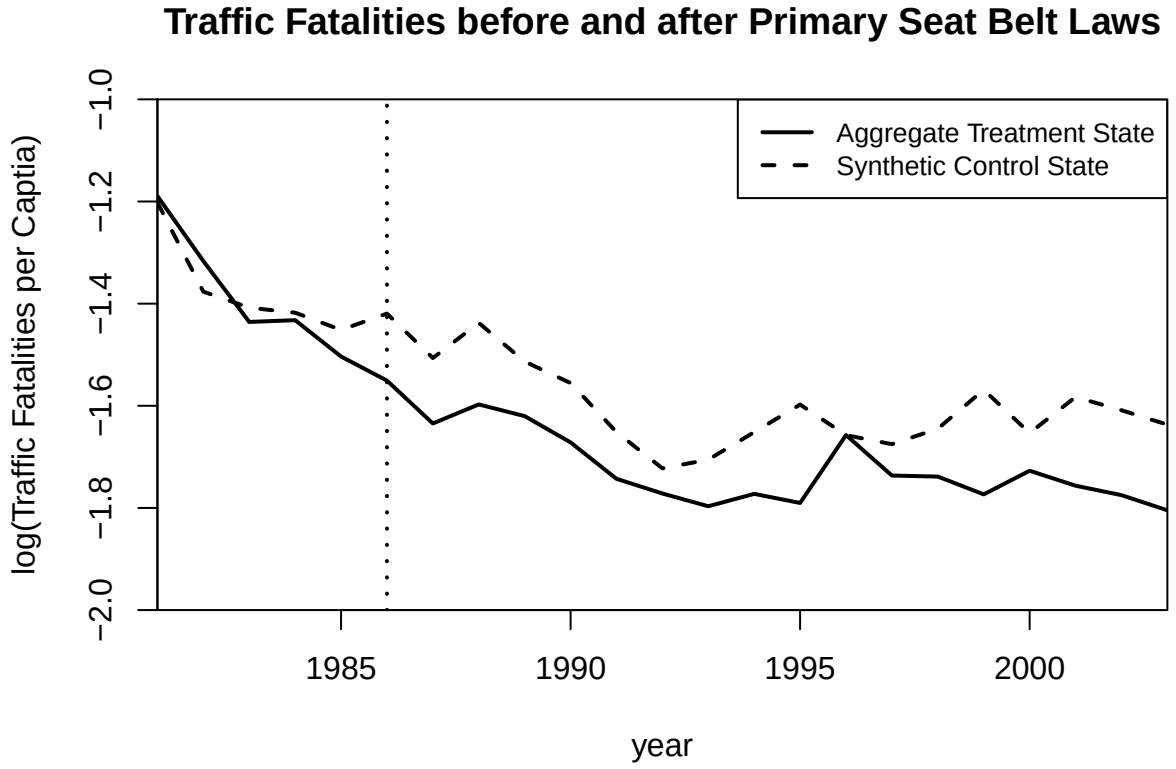
Table 9: Covariate Weights used sum of SSE's

	Covariate Weights
college	0.17
beer	0.164
unemploy	0.157
ln_tvmt_pc	0.194
ln_precip	0.315
snow32	0.001

The weighted synthetic control results in covariate balance Table 10 between the synthetic control and aggregate treated unit. We can see that the balance is very close in the pre-treatment periods, except under snow, but that was weighted very low in the synthetic control construction procedure because of its strong correlation with precipitation.

Table 10: Covariate Balance in Pre-treatment Periods

	Treated	Synthetic
college	0.224	0.224
beer	1.620	1.619
unemploy	6.753	6.748
ln_tvmt_pc	2.116	2.116
ln_precip	0.892	0.891
snow32	0.178	0.357

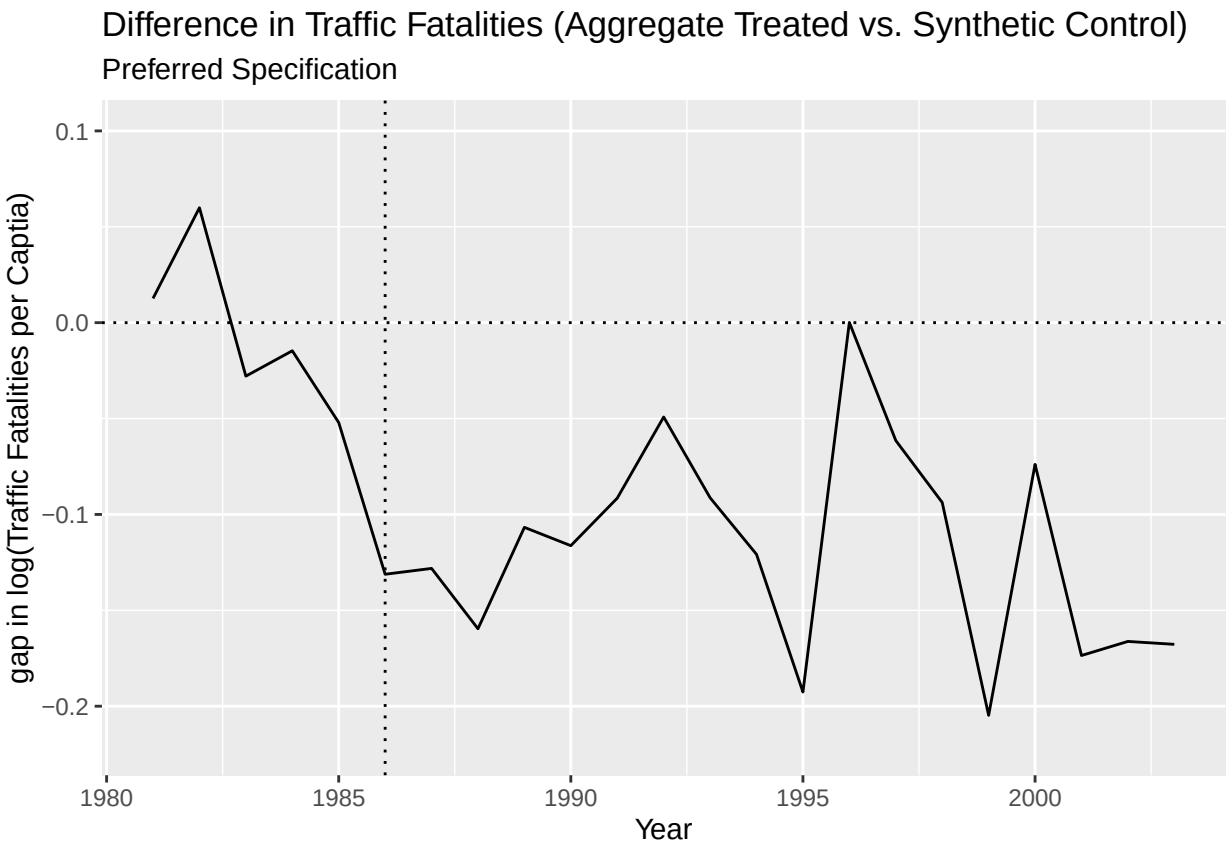


Part (c)

Graphical interpretation and treatment significance.

—— c.i

Generate graphs plotting the gap between the TU and the synthetic control group under both your preferred specification and a few other specifications you tried.



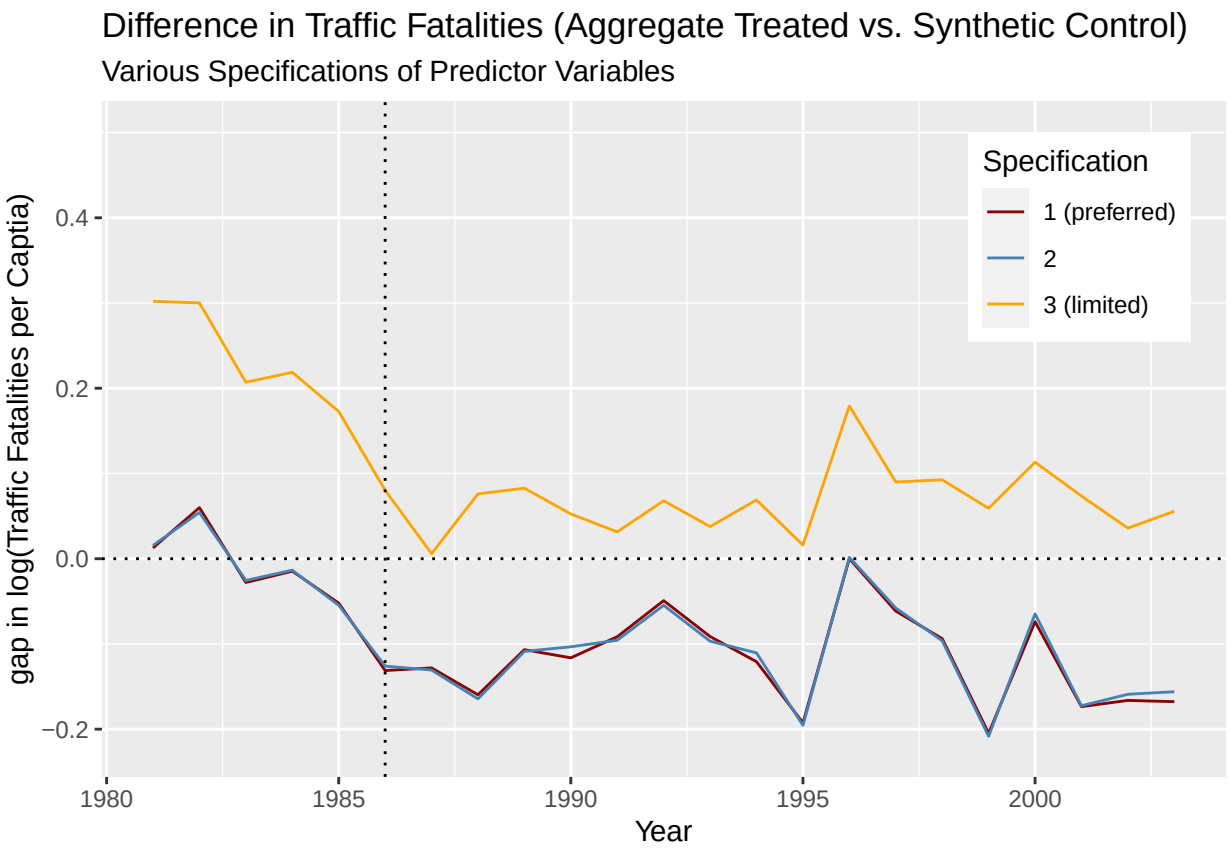


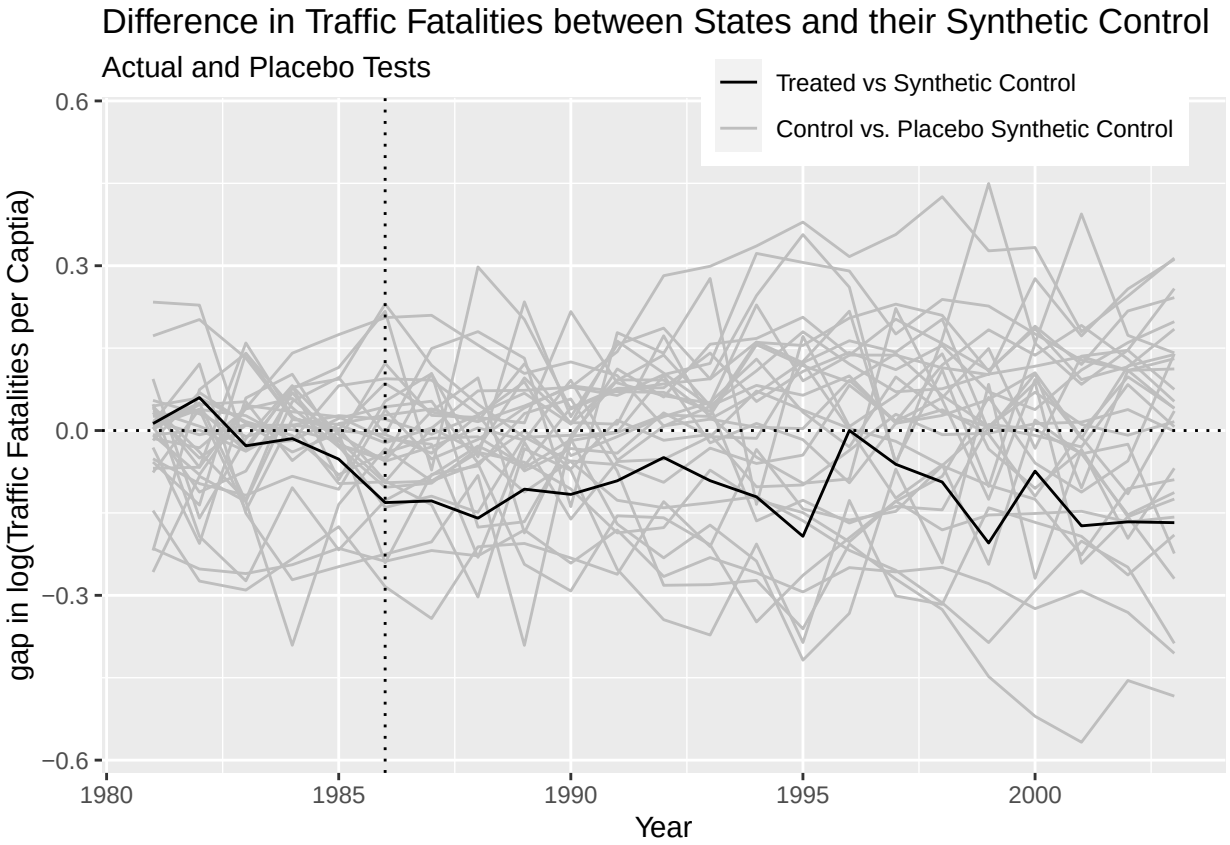
Table 11: Predictors included in each specification

predictor	Specification 1	Specification 2	Specification 3
college	x	x	x
beer	x	x	
unemploy	x	x	
ln_tvmt_pc	x	x	
ln_precip	x	x	x
snow32	x		

c.ii

Compare the graph plotting the gap between the TU and the synthetic control group under your preferred specification with the graphs plotting the gap between each control state and its "placebo" treatment. Do you conclude that the treatment was significant? Why or why not?

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



Based on this graph alone (before seeing the placebo test), we might conclude that there was indeed a treatment effect of the policy. However, without more context (provided in 2ciii), we can't say.

—— **c.iii**

Create a graph of the post-treatment/pre-treatment prediction ratios of the Mean Squared Prediction Errors (MSPE) for the actual and "placebo" treatment gaps in (ii). [See Abadie et al. for an example]. Do you conclude that the treatment was significant? Why or why not?

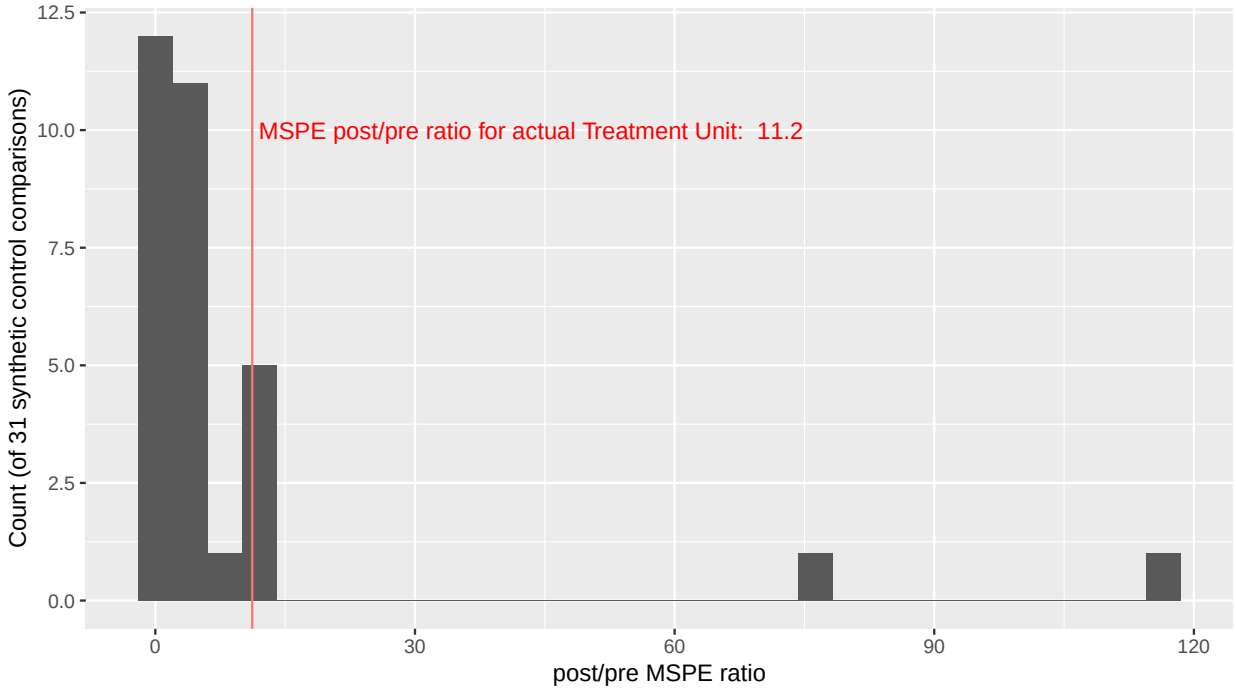
From the distribution of MSPEs, we see that our treatment cannot be ruled out as mere chance. It is not even more than one standard deviation from the mean MSPE of placebo tests.

Since we have 30 control states, this gives us a total of 31 synthetic control comparisons. We can calculate the MSPE of the synthetic control compared to the Treatment Unit separately for the pre-treatment years and post-treatement years. Taking the post-pre ratio of MSPE gives us a sense of how divergent the Treatment Unit is from its synthetic control. We can do this for all 31 synthetic control experiments (30 placebos using the 30 controls, and the actual Treatment Unit).

In 5, we can see that the post/pre MSPE ratio falls somewhat on the right side of the distribution. We can calculate a p-value as the probability that the result we got was not just from random chance. This is the number of MSPE ratios that are equal to or greater than the Treatment Unit's MSPE ratio, divided by the total number of ratios. The p-value for our actual synthetic control estimate is 0.194, so we cannot reject the null hypothesis that the decrease in fatalities is merely from random variation in the data.

$$\text{p-value} = \frac{\#\text{ of MSPE ratios at or above the TU}}{\text{total }\#\text{ of MSPE ratios}} = \frac{6}{31} = 0.194$$

Figure 5: Histogram of the ratios of Mean Squared Prediction Errors of the synthetic controls in the post-treatment compared to the pre-treatment years.



Part (d)

How do your synthetic control results compare to your fixed effects results from Question (3) in the last problem set? Interpret any differences.

Appendix A: R Code

```
rm(list=ls())
knitr::opts_chunk$set(echo = F)

# stargazer table type (html, latex, or text)
# Change to latex when outputting to PDF, html when outputting to html
table_type = "latex"
Cache = TRUE

# Packages
# install.packages("Synth")
library(haven)
library(stargazer)
library(ggplot2)
library(tinytex)
library(kableExtra)
library(fastDummies)
library(Synth)
library(progress)
library(tidyverse)

# Load data from PS2a with previous log variables
data = read_dta('traffic_safety2.dta') %>%
    mutate(fat_pc = fatalities/population,
           ln_fat_pc = log(fat_pc),
           ln_tvmt_pc = log(totalvmt/population),
           ln_precip = log(precip),
           ln_rspeed = log(rural_speed),
           ln_uspeed = log(urban_speed)) %>%
    data.frame(.)

# Create list of event dates for states that passed primary laws in our study
event_dates = data %>%
    group_by(state) %>%
    mutate(event = primary - lag(primary), # event=1 ==> first year primary=1
           event_year = year) %>%
    filter(event == 1) %>%
    select(state, event_year)

# Add year of primary event and event time (t) to dataframe
data = data %>%
    left_join(event_dates, by='state') %>%
    mutate(j = ifelse(is.na(event_year), 99, year - event_year))
    # t = 99 ==> control state (doesn't pass primary during study period)

# Table of max and min event times
# Shouldn't these be our event study thresholds?
df_temp = data %>%
    filter(j < 99) %>%
    group_by(state) %>%
    summarize(min_j = min(j), max_j = max(j))
```

```r
max_j_inclusive = min(df_temp$max_j, na.rm = T)
min_j_inclusive = max(df_temp$min_j, na.rm = T)
max_j = max(filter(data, j<99)$j, na.rm = T)
min_j = min(filter(data, j<99)$j, na.rm = T)


data.frame(max_j = max_j, min_j = min_j) %>%
  kbl(caption = "Maximum \\& Minimum Event Time Values",
      col.names = c('Max j', 'Min j'),
      align = 'cc') %>%
  kable_styling(latex_options = "HOLD_position")

df_temp %>%
    arrange(max_j) %>%
    filter(!is.na(max_j), max_j < 99) %>%
    select(max_j) %>%
    ggplot(aes(x=max_j), data=.) +
    geom_histogram() +
    xlab("max event period")

# Function for adding dummies to a dataframe for all uniuqe values between given numbers
create_dummies = function(df, colname, min_value, max_value) {
    # Create dummies for each value of colname between min_value and max_value
    df1 = df
    for (val in min_value:max_value) {
        df1 = mutate(df1, "{colname}_{val}" := ifelse(eval(as.symbol(colname)) == val, 1, 0))
    }
    return(df1)
}

# Create order of dummies for dataframe (then used for regression table)
name_order1 = paste('j', min_j_inclusive:max_j_inclusive, sep='_')
# Create Dummies that make a balanced panel
# (only dummies for event times j that are shared across all states)
df_few_dummies = create_dummies(data,
                                colname = 'j',
                                min_value = min_j_inclusive,
                                max_value = max_j_inclusive) %>%
    relocate(all_of(name_order1)) %>%
    # Change "j_..." to "..._ET" because LaTeX doesn't like j_-3 type variable names
    rename_with(~ paste0(str_replace(., 'j_', ''), '_ET'), contains("j_"))

# Create order of dummies for dataframe (then used for regression table)
name_order2 = paste('j', min_j:max_j, sep='_')
# Create Dummies for all possible event times
# (results in unbalanced panel over event times j)
df_all_dummies = dummy_cols(data, select_columns = 'j') %>%
    select(-j_99) %>%
    filter(state != 99)  %>%
    relocate(all_of(name_order2)) %>%
    rename_with(~ paste0(str_replace(., 'j_', ''), '_ET'), contains("j_"))

# Regression: dummy-trap event study
```

```r
reg_1a = df_all_dummies %>%
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('_ET')) %>%
    lm(ln_fat_pc ~ ., data = .)

# Regression: event study, omit t=-1
reg_1b = df_all_dummies %>%
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('_ET'), -`-1_ET`) %>%
    lm(ln_fat_pc ~ ., data = .)

# Event Study Table
stargazer(reg_1a, reg_1b,
          title = "Event Study Regressions\\label{tab:event-study-dummy-trap}",
          dep.var.caption = "Log(Fatality per Population)",
          dep.var.labels.include = FALSE,
          column.labels = c("Event Study a", "Event Study b"),
          omit = c("state", "year"),
          add.lines=list(c('Chose dummy to omit', 'No', 'Yes')),
          font.size = "footnotesize", column.sep.width = "1pt", no.space = TRUE,
          omit.stat=c("f", "ser"),
          single.row = TRUE,
          digits = 4, type = table_type, header = FALSE)


# Function for adding dummies to a dataframe for all uniuqe values between given numbers
create_dummies_threshold = function(df, colname, min_value, max_value, suffix = NULL) {
    # Create indicator variables for each value of colname between min_value and max_value
    # then create indicator variables for all values of colname below min_value
    # and another for above max_value
    if (is.null(suffix)) {suffix = colname}
    df1 = df
    # add aggregate indicator for all values below min_value
    df1 = mutate(df1, "below_{suffix}" := ifelse(eval(as.symbol(colname)) < min_value, 1, 0))
    # add all indicators in between min and max_value
    for (val in min_value:max_value) {
        df1 = mutate(df1, "{val}_{suffix}" := ifelse(eval(as.symbol(colname)) == val, 1, 0))
    }
    # add aggregate indicator for all values above max_value
    df1 = mutate(df1, "above_{suffix}" := ifelse(eval(as.symbol(colname)) > max_value, 1, 0))
    return(df1)
}

# Create Dummies that make a balanced panel
# (only dummies for event times j that are shared across all states)
df_threshold_dummies = create_dummies_threshold(data,
                                                colname = 'j',
                                                min_value = -5,
                                                max_value = 5,
                                                suffix = 'ET')

# Regression: event study, threshold dummies
reg_1c = df_threshold_dummies %>%
```

```
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('_ET'), -`-1_ET`) %>%
    lm(ln_fat_pc ~ ., data = .)

# Event study table
stargazer(reg_1c,
          title = "Event Study Regression with Threshold Indicators\\label{tab:event-study-thresholds}"
          dep.var.caption = "Log(Fatality per Population)",
          dep.var.labels.include = FALSE,
          column.labels = c("Event Study c"),
          omit = c("state", "year"),
          add.lines=list(c('Chose dummy to omit', 'Yes'), c('Agg. Threshold Indicators', 'Yes')),
          font.size = "footnotesize", column.sep.width = "1pt", no.space = TRUE,
          omit.stat=c("f", "ser"),
          single.row = TRUE,
          notes = c("below\\_ and above\\_ variables are aggregate indicators for",
                    "all event times that are below -5 and above 5, respectively."),
          digits = 4, type = table_type, header = FALSE)

# Text used in below figure (inserted using R markdown chunk arguments)
plot_text1 = "(red) Plot of all possible event-time coefficients and 95\\% confidence intervals from Ta

# Function to create new varname from old
new_varname = function(oldname, min_value, max_value) {
    name = str_replace_all(oldname, "`","")
    name = str_replace(name, "_ET", "")
    name = str_replace(name, "below", paste("<", min_value))
    name = str_replace(name, "above", paste(">", max_value))
    return(name)
}

# Function to extract value from varname string
xvalue = function(oldname, min_value = NULL, max_value = NULL) {
    name = str_replace_all(oldname, "`","")
    name = str_replace(name, "_ET", "")
    if (!is.null(min_value)) {
        name = str_replace(name, "below", as.character(min_value - 1))
        name = str_replace(name, "above", as.character(max_value + 1))
    }
    return(as.numeric(name))
}

# Function to return a dataframe of regression results
regression_dataframe = function(reg, reg_type, min_value = NULL, max_value = NULL, suffix = 'ET') {
    df = data.frame(coef = names(reg$coefficients),
                    value = reg$coefficients,
                    lower = confint(reg)[,1],
                    upper = confint(reg)[,2],
                    reg_type = reg_type) %>%
        filter(grepl(suffix, coef)) %>%
        mutate(x_tick = new_varname(coef, min_value, max_value),
               event_time = xvalue(coef, min_value, max_value)) %>%
        return()
```

```
}

df1 = rbind(
    regression_dataframe(reg_1b, "all indicators"),
    regression_dataframe(reg_1c, "max-min indicators", min_value = -5, max_value = 5)
)

# Plot the event-time regression coefficients
df1 %>%
    ggplot(aes(x=event_time, y=value, color = reg_type)) +
    geom_errorbar(aes(ymin=lower, ymax=upper), width=.1) +
    geom_line() +
    geom_point() +
    geom_hline(yintercept = 0) +
    labs(color = "Parameters estimated") +
    xlab("Event Time (years since passage of primary seat belt law)") +
    ylab("Coefficient Value") +
    annotate(geom = "segment", x = 3, y = -0.3, xend = 5.7, yend = -0.14,
             arrow = arrow(length = unit(2, "mm"))) +
    annotate(geom = "text", x = 3, y = -0.32,
             label = "(blue) coeffcient on >5 years after passage",
             hjust = "center") +
    annotate(geom = "segment", x = -4.5, y = -0.15, xend = -5.9, yend = 0.018,
             arrow = arrow(length = unit(2, "mm"))) +
    annotate(geom = "text", x = -8, y = -0.17,
             label = "(blue) coeffcient on >5 years before passage",
             hjust = "center")

# Regression with all possible event-time indicators, removing pure control states
reg_1b2 = df_all_dummies %>%
    group_by(state) %>%
    filter(mean(primary) > 0) %>%
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('_ET'), -`-1_ET`) %>%
    lm(ln_fat_pc ~ ., data = .)

# Regression with max, min aggregated event-time indicators, removing pure control states
reg_1c2 = df_threshold_dummies %>%
    group_by(state) %>%
    filter(mean(primary) > 0) %>%
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('_ET'), -`-1_ET`) %>%
    lm(ln_fat_pc ~ ., data = .)

# Event study table dropping pure controls
stargazer(reg_1b, reg_1b2, reg_1c, reg_1c2,
          title = "Event Study Regressions with and without Pure Control States\\label{tab:event-study-r
          dep.var.caption = "Log(Fatality per Population)",
          dep.var.labels.include = FALSE,
          column.labels = c("All indicators",
                            "All indicators",
                            "Min-max indicators",
                            "Min-max indicators"),
```

```
          omit = c("state", "year"),
          add.lines=list(c('Agg. Threshold Indicators', 'No', 'No', 'Yes', 'Yes'),
                         c('Include Pure Controls', 'Yes', 'No', 'Yes', 'No')),
          font.size = "footnotesize", column.sep.width = "1pt", no.space = TRUE,
          omit.stat=c("f", "ser"),
          single.row = TRUE,
          digits = 4, type = table_type, header = FALSE)


# Text used fro below figure caption
plot_text2 = "Plot of Sun-and-Abraham(2020)-style cohort-by-event-time coefficients."

# Function for adding dummies to a dataframe for all interactions between two columns
create_interacted_indicators = function(df, col1, col2) {
    # Create dummies for each combination of non-constant interactions of col1 and col2
    df1 = df %>% mutate("{col1}" := ifelse(eval(as.symbol(col1)) == 99, NA, eval(as.symbol(col1))),
                        "{col2}" := ifelse(eval(as.symbol(col2)) == 99, NA, eval(as.symbol(col2))))
    df1 = df
    df1[, col1] = replace(df1[, col1], df1[, col1]==99, NA)
    df1[, col2] = replace(df1[, col2], df1[, col2]==99, NA)
    for (val1 in df1[, col1] %>% unique() %>% sort()) {
        for (val2 in df1[, col2] %>% unique() %>% sort()) {
            inter = as.integer(df1[, col1] == val1) * as.integer(df1[, col2] == val2)
            inter = replace_na(inter, 0)
            # Only keep the interaction if there is variation in the data
            if (var(inter) > 0) {
                name1 = ifelse(val1<0, paste0('n', abs(val1)), val1)
                name2 = ifelse(val2<0, paste0('n', abs(val2)), val2)
                name = paste('inter', name1, name2, sep='_')
                df1[, name] = inter
            }
        }
    }
    return(df1)
}

# Regression with cohort interactions
reg_1f = data %>%
    filter(state != 99) %>%
    create_interacted_indicators(., 'j', 'event_year') %>%
    mutate(state=factor(state), year=factor(year)) %>%
    select(ln_fat_pc, state, year, contains('inter_'), -contains('_n1_')) %>%
    lm(ln_fat_pc ~ ., data = .)

# Regression results dataframe
reg_1f_df = data.frame(summary(reg_1f)$coefficients) %>%
    mutate(var = row.names(.)) %>%
    # select(var, everything()) %>%
    rename(est = Estimate, SE = Std..Error, t = t.value, p = Pr...t..) %>%
    filter(grepl('inter_', var)) %>%
    separate(var, c('x', 'event_time', 'event_year')) %>%
    select(-x) %>%
    mutate(event_year = as.integer(event_year)) %>%
```

```
    mutate(event_time = as.integer(str_replace(event_time, 'n', '-')))

# Plot all event_year cohort treatment effects
reg_1f_df %>%
    ggplot(aes(x=event_time, y=est, color=factor(event_year))) +
    geom_point() + geom_line() +
    geom_hline(yintercept = 0) +
    labs(color='Treatment Cohort') +
    xlab("Event Time (years since passage of primary seat belt law)") +
    ylab("Coefficient Value")

# Text for below figure caption
plot_text3 = "Plot of cohort-weighted avereage event-time coefficents -- weighted using the number of s

# Calculate treatment effect weights for each combination of event time and event year
for (i in (1:nrow(reg_1f_df))) {
    t_ = reg_1f_df[i, 'event_time']
    y_ = reg_1f_df[i, 'event_year']
    # Find number of states in cohort event_year
    numerator = data %>%
        filter(j == t_, event_year == y_) %>%
        nrow()
    denominator = data %>%
        filter(j == t_) %>%
        nrow()
    reg_1f_df[i, 'event_time_weight'] = numerator / denominator
}

# Calculate weighted average of treatment effects for a given event time
reg_1f_df %>%
    group_by(event_time) %>%
    summarize(value = weighted.mean(est, event_time_weight)) %>%
    mutate(reg_type = 'Sun & Abraham \n cohort weighted') %>%
    rbind(., select(df1, event_time, value, reg_type)) %>%
    arrange(reg_type) %>%
    # Plot the treatment effects over event time
    ggplot(aes(x=event_time, y=value, color = reg_type)) +
    geom_line() +
    geom_point() +
    geom_hline(yintercept = 0) +
    labs(color = "Parameters estimated") +
    xlab("Event Time (years since passage of primary seat belt law)") +
    ylab("Coefficient Value")


###################################
#  PROBLEM 2
###################################
# Calculate average pre-period log traffic fatalities per capita
preperiod_years = data %>%
    filter(state == 99, primary == 0) %>%
    select(year) %>%
    unlist(.) %>%
```

```
    as.numeric(.)

data2 = data %>%
    group_by(state) %>%
    mutate(control = ifelse(mean(primary) > 0, 0, 1),
           treated = ifelse(state == 99, 'Yes', 'No')) %>%
    filter(control == 1 | state == 99)

means = data2 %>%
    filter(year %in% preperiod_years) %>%
    group_by(control) %>%
    summarize(avg = mean(ln_fat_pc))

# Create table to show difference in pre-treatment outcome means
means %>%
    select(avg) %>%  # control = 0 on top ==> TU on top
    t() %>%  # control = 0 on left ==> TU on left
    kbl(caption = "Average Pre-period log(traffic fatalities per capita)\\label{tab:avg-fat-TU}",
        col.names = c('Aggregate Treatment State', 'Aggregate Control State'),
        row.names = F,
        align = 'cc') %>%
    kable_styling(latex_options = "HOLD_position")

# Plot log fat per cap for pre-treatment years, for treatment and control
data2 %>%
    filter(year %in% preperiod_years) %>%
    group_by(year, treated) %>%
    summarize(year_avg = mean(ln_fat_pc)) %>%
    ggplot(aes(x=year, y=year_avg, color=factor(treated))) +
    geom_line() +
    geom_point() +
    geom_hline(yintercept = filter(means, control==0)$avg, color='cyan') +
    geom_hline(yintercept = filter(means, control==1)$avg, color='red') +
    labs(color = "Will Enact\n Seatbelt Law") +
    xlab("Year") +
    ylab("log(Fatalities per capita)") +
    ggtitle("Pre-period annual avereage log(Fatalilties per capita)") +
    annotate(geom = "text", x = 1984, y = -1.56,
             label = "Average for control\n states over pre-periods",
             hjust = "center") +
    annotate(geom = "text", x = 1984, y = -1.34,
             label = "Average for the aggregate\n treated unit over pre-periods",
             hjust = "center")


# Compare log(fat per cap) for last pre-treatment year, for treatment and each control
treat_val = (data2 %>%
    filter(year == tail(preperiod_years, n=1),
           state == 99))$ln_fat_pc

# Table of state differences in pre-period outcomes compared to TU
data2 %>%
    filter(year == tail(preperiod_years, n=1)) %>%
```

```
    filter(control == 1 | state == 99) %>%
    mutate(diff = abs(treat_val - ln_fat_pc)) %>%
    arrange(diff) %>%
    select(ln_fat_pc, diff) %>%
    # head() %>%
    kbl(caption = paste("Dependent variable in",
                        tail(preperiod_years, n=1),
                        "for treated (99) and control states\\label{tab:1985-depvar}"),
        col.names = c('State ID', 'log(fatatilities per captia)',
                      'Absolute difference between this state and treated'),
        row.names = F,
        align = 'cc') %>%
    kable_styling(latex_options = "HOLD_position")

# Compare covariates for last pre-treatment year, for treatment and control 47
data2 %>%
    filter(year == tail(preperiod_years, n=1)) %>%
    filter(state %in% c(47, 99)) %>%
    select(college, beer, population, unemploy, totalvmt, precip,
           snow32, rural_speed, urban_speed, fat_pc) %>%
    round(4) %>%
    t() %>%
    kbl(caption = paste("Dependent variable in",
                        tail(preperiod_years, n=1),
                        "for treated (99) and control states\\label{tab:1985-covar}"),
        col.names = c('Control State 47', 'Aggregate Treatment State'),
        align = 'cc') %>%
    kable_styling(latex_options = "HOLD_position")


# Function to create inputs for Synth dataprep and synth functions
apply_synth = function(data, predictors_, treated_unit=99) {
    # Create list of all years in sample
    all_years = data %>%
        arrange(year) %>%
        select(year) %>%
        unique() %>%
        unlist(.) %>%
        as.numeric(.)

    # Create list of pre-treatment years
    preperiod_years = data %>%
        filter(state == 99, primary == 0) %>%
        select(year) %>%
        unlist(.) %>%
        as.numeric(.)

    # Create list of control states
    control_states = data %>%
        group_by(state) %>%
        mutate(control = ifelse(mean(primary) > 0, 0, 1)) %>%
        filter(year == tail(preperiod_years, n=1)) %>%
        # need to make sure we don't select the treated unit (for the placebo tests)
```

```
        filter(control == 1, state != treated_unit) %>%
        select(state) %>%
        unlist(.) %>%
        as.numeric(.)

    # Create lookup table of state names to join onto data
    state_ids = data.frame(attr(data$state, 'label'))[,1]
    state_names = rownames(data.frame(attr(data$state, 'label')))
    state_names_df = data.frame(state = state_ids,
                                state_name = state_names,
                                stringsAsFactors=FALSE)

    # Prepare data for Synth
    dataprep.out = data %>%
        mutate(state = as.numeric(unlist(state))) %>%
        left_join(state_names_df, by='state') %>%  # add state_name
        data.frame(.) %>%
        dataprep(foo = .,
                 predictors = predictors_,
                 time.predictors.prior = preperiod_years,
                 dependent = "ln_fat_pc",
                 unit.variable = "state",
                 unit.names.variable = "state_name",
                 time.variable = "year",
                 treatment.identifier = treated_unit,
                 controls.identifier = control_states,
                 time.optimize.ssr = preperiod_years,
                 time.plot = all_years
        )

    # Run Synth to create weighted control
    synth.out <- invisible(synth(data.prep.obj = dataprep.out, method = "BFGS"))

    return(list(
        dataprep.out = dataprep.out,
        synth.out = synth.out
    ))
}


# Run Synth on preferred specification
predictors1 = c("college" , "beer" , "unemploy" , "ln_tvmt_pc" , "ln_precip", "snow32")

synth1 = apply_synth(data, predictors1)

synth.tables <- synth.tab(dataprep.res = synth1$dataprep.out,
                          synth.res = synth1$synth.out)

# Create table of Synth weights
synth.tables$tab.w %>%
    select(w.weights, unit.names) %>%
  kbl(caption = "Synthetic Control Weights\\label{tab:synth-weights}",
      col.names = c('Weight', 'State'),
```

```
      align = 'cc',
      row.names = F) %>%
  kable_styling(latex_options = "HOLD_position")

# Create table of Synth variable weights
synth.tables$tab.v %>%
  kbl(caption = "Covariate Weights used sum of SSE's\\label{tab:synth-cov-weights}",
      col.names = c('Covariate Weights'),
      align = 'cc') %>%
  kable_styling(latex_options = "HOLD_position")

# Create variable balance table
synth.tables$tab.pred %>%
    data.frame() %>%
    select(Treated, Synthetic) %>%
  kbl(caption = "Covariate Balance in Pre-treatment Periods\\label{tab:cov-balance}",
      col.names = c('Treated', 'Synthetic'),
      align = 'ccc') %>%
  kable_styling(latex_options = "HOLD_position")

# Plot the TU and Synth control
path.plot(synth.res = synth1$synth.out,
          dataprep.res = synth1$dataprep.out,
          Ylab = "log(Traffic Fatalities per Captia)",
          Xlab = "year",
          Ylim = c(-2,-1),
          Legend = c("Aggregate Treatment State","Synthetic Control State"),
          Main = "Traffic Fatalities before and after Primary Seat Belt Laws",
          tr.intake = tail(preperiod_years, n=1)+1
          )

# Run Synth on two more specifications
predictors2 = c("college" , "beer" , "unemploy" , "ln_tvmt_pc" , "ln_precip")
synth2 = apply_synth(data, predictors2)
predictors3 = c("college", "ln_precip")
synth3 = apply_synth(data, predictors3)

# Plot the gaps of three specifications using Synth
gaps.plot(synth.res = synth3$synth.out,
          dataprep.res = synth3$dataprep.out,
          Ylab = "gap in log(Traffic Fatalities per Captia)",
          Xlab = "year",
          Main = "Difference in Traffic Fatalities (Aggregate Treated vs. Synthetic Control)",
          tr.intake = tail(preperiod_years, n=1)+1
          )

# Retrieve gap between TU and synth control
get_gaps = function(synth_out){
    # Return vector of years and gaps between treated and synthetic control
    # negative gap ==> treated is less than control
    gaps = synth_out$dataprep.out$Y1plot -
        (synth_out$dataprep.out$Y0plot %*% synth_out$synth.out$solution.w)
    return(list(
```

```
        year = as.numeric(rownames(gaps)),
        gaps = as.numeric(gaps)
    ))
}

gaps1 = get_gaps(synth1)
gaps2 = get_gaps(synth2)
gaps3 = get_gaps(synth3)

gaps_df = rbind(
    data.frame(Year = gaps1$year, gaps = gaps1$gaps, specification = '1 (preferred)'),
    data.frame(Year = gaps2$year, gaps = gaps2$gaps, specification = '2'),
    data.frame(Year = gaps3$year, gaps = gaps3$gaps, specification = '3 (limited)')
)

# Plot the preferred specification in ggplot
gaps_df %>%
    filter(specification == '1 (preferred)') %>%
    ggplot(aes(x=Year, y=gaps)) +
    geom_line() +
    ylim(c(-0.22,0.1)) +
    ylab("gap in log(Traffic Fatalities per Captia)") +
    ggtitle('Difference in Traffic Fatalities (Aggregate Treated vs. Synthetic Control)',
            subtitle='Preferred Specification') +
    geom_hline(yintercept = 0, linetype = "dotted") +
    geom_vline(xintercept = tail(preperiod_years, n=1)+1, linetype = "dotted")

# Plot the gaps of three specifications using ggplot
gaps_df %>%
    ggplot(aes(x=as.integer(Year), y=gaps)) +
  geom_line(aes(color = factor(specification))) +
  scale_color_manual(values = c("darkred", "steelblue", "orange")) +
    ylim(c(-0.22,0.5)) +
    xlab('Year') +
    ylab("gap in log(Traffic Fatalities per Captia)") +
    ggtitle('Difference in Traffic Fatalities (Aggregate Treated vs. Synthetic Control)',
            subtitle='Various Specifications of Predictor Variables') +
    labs(color = 'Specification') +
    geom_hline(yintercept = 0, linetype = "dotted") +
    geom_vline(xintercept = tail(preperiod_years, n=1)+1, linetype = "dotted") +
  theme(legend.position = c(0.87, 0.8))


# Create table showing variables included in different specifications
predictors = unique(c(predictors1, predictors2, predictors3))
spec_df = data.frame(predictor=predictors)
for (i in 1:3) {
    # spec_df[, paste('Specification', i)] =
    spec_df = spec_df %>%
        mutate("Specification {i}" :=
                   ifelse(predictor %in% eval(as.symbol(paste0('predictors', i))), 'x', ''))
}
```

```
spec_df %>%
    kbl(caption = paste("Predictors included in each specification\\label{tab:specifications}"),
        # col.names = c('State ID', 'log(fatatilities per captia)',
        #               'Absolute difference between this state and treated'),
        row.names = F,
        align = 'cc') %>%
    kable_styling(latex_options = "HOLD_position")


# Create list of control states
control_states = data %>%
    group_by(state) %>%
    mutate(control = ifelse(mean(primary) > 0, 0, 1)) %>%
    filter(year == tail(preperiod_years, n=1)) %>%
    filter(control == 1) %>%
    select(state) %>%
    unlist(.) %>%
    as.numeric(.)

# Create list of all years in sample
all_years = data %>%
    arrange(year) %>%
    select(year) %>%
    unique() %>%
    unlist(.) %>%
    as.numeric(.)

# Dataframe to store gaps results
gaps_df2 = data.frame(Year = gaps1$year,
                      gaps = gaps1$gaps,
                      unit = 99,
                      color = 'black')

# Create a progress bar ... this might take a while
progress = progress_bar$new(
    format = " State :id [:bar] :percent eta: :eta",
    total = length(control_states),
    clear = FALSE, width= 60)

# Create placebo synthetic controls for all control states
for (state_id in control_states){
    progress$tick(tokens = list(id = state_id))
    # Run Synth for control state state_id, get placebo gaps
    gaps_temp = invisible(get_gaps(apply_synth(data,
                                       predictors1,
                                       treated_unit = state_id)))
    # Add gaps to dataframe
    temp_df = data.frame(Year = gaps_temp$year,
                         gaps = gaps_temp$gaps,
                         unit = state_id,
                         color = 'grey')
    gaps_df2 = rbind(gaps_df2, temp_df)
}
```

```
# Plot all the gaps (actual treatment estimate and control placebos)
gaps_df2 %>%
    ggplot(aes(x=as.integer(Year), y=gaps, group=unit)) +
    geom_line(aes(color = color)) +
    scale_color_identity() + scale_linetype_identity() +
    ylim(c(min(gaps_df2$gaps), max(gaps_df2$gaps)+0.1)) +
    xlab('Year') +
    ylab("gap in log(Traffic Fatalities per Captia)") +
    ggtitle('Difference in Traffic Fatalities between States and their Synthetic Control',
            subtitle='Actual and Placebo Tests') +
    geom_hline(yintercept = 0, linetype = "dotted") +
    geom_vline(xintercept = tail(preperiod_years, n=1)+1, linetype = "dotted") +
    theme(legend.position = c(0.75, 1), legend.title=element_blank()) +
    scale_color_manual(labels = c("Treated vs Synthetic Control",
                                  "Control vs. Placebo Synthetic Control"),
                       values = c("black", "grey"))


# Text used in below figure caption
ratio_text = "Histogram of the ratios of Mean Squared Prediction Errors of the synthetic controls in the

# Calculate Mean Squared Prediction Error ratios (post vs pre period, errors = gaps)
MSPE_ratios = gaps_df2 %>%
    mutate(post = ifelse(Year > max(preperiod_years), 'post', 'pre')) %>%
    group_by(unit, post) %>%
    summarize(MSPE = mean(gaps^2)) %>%
    pivot_wider(names_from = post, values_from = MSPE) %>%
    mutate(MSPE_post_pre = post / pre) %>%
    data.frame() %>%
    mutate(rank = dense_rank(desc(MSPE_post_pre)))

# Treated unit post/pre ratio
TU_ratio = MSPE_ratios %>%
    filter(unit == 99) %>%
    select(contains('MSPE')) %>% unlist %>% unname

# Treated unit's ranking in MSPE ratio
TU_rank = MSPE_ratios %>%
    filter(unit == 99) %>%
    select(contains('rank')) %>% unlist %>% unname

total_ratios = nrow(MSPE_ratios)

# Histogram of MSPE Ratios
MSPE_ratios %>%
    ggplot(aes(x = MSPE_post_pre)) +
    geom_histogram() +
    ylab(paste('Count (of', length(control_states) + 1, 'synthetic control comparisons)')) +
    xlab('post/pre MSPE ratio') +
    geom_vline(aes(xintercept=TU_ratio, color='red')) +
    annotate(geom = "text", x = TU_ratio, y = 10,
             label = paste(" MSPE post/pre ratio for actual Treatment Unit: ", round(TU_ratio,1)),
             hjust = "left", color = 'red') +
```

```
    theme(legend.position = "none")
```