Diff in Diffs and Synthetic Control

```r
# Install packages
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```r
devtools::install_github("ebenmichael/augsynth")
```

```
## Skipping install of 'augsynth' from a github remote, the SHA1 (c68b2e2b) has not changed since last i
##   Use `force = TRUE` to force installation
```

```r
pacman::p_load(# Tidyverse packages including dplyr and ggplot2
               tidyverse,
               ggthemes,
               augsynth)


set.seed(1)
```

Introduction

In this lab we will explore difference-in-differences estimates and a newer extension, synthetic control. The basic idea behind both of these methods is simple - assuming two units are similar in a pre-treatment period and one undergoes treatment while the other stays in control, we can estimate a causal effect by taking three differences. First we take the difference between the two in the pre-treatment period, then take another difference in the post-treatment period. Then we take a difference between these two differences (hence the name difference in differences). Let's see how this works in practice!

Basic DiD

We'll use the kansas dataset that comes from the `augsynth` library. Our goal here is to estimate the effect of the 2012 Kansas tax cuts on state GDP. Let's take a look at our dataset:

```r
data(kansas)
summary(kansas)
```

```
##       fips            year           qtr            state
##  Min.   : 1.00   Min.   :1990   Min.   :1.000   Length:5250
##  1st Qu.:17.00   1st Qu.:1996   1st Qu.:1.000   Class :character
##  Median :29.50   Median :2003   Median :2.000   Mode  :character
##  Mean   :29.32   Mean   :2003   Mean   :2.486
```

```
##  3rd Qu.:42.00   3rd Qu.:2009   3rd Qu.:3.000
##  Max.   :56.00   Max.   :2016   Max.   :4.000
##
##       gdp           revenuepop    rev_state_total  rev_local_total
##  Min.   :  11509   Min.   : 1335   Min.   :  1668   Min.   :   550
##  1st Qu.:  55151   1st Qu.: 3057   1st Qu.:  7026   1st Qu.:  3268
##  Median : 130650   Median : 3628   Median : 13868   Median : 10041
##  Mean   : 228237   Mean   : 3851   Mean   : 20813   Mean   : 17197
##  3rd Qu.: 276303   3rd Qu.: 4365   3rd Qu.: 24405   3rd Qu.: 18774
##  Max.   :2568986   Max.   :14609   Max.   :182530   Max.   :143137
##                    NA's   :2250    NA's   :2850     NA's   :2850
##   popestimate       qtrly_estabs_count month1_emplvl      month2_emplvl
##  Min.   :  453690   Min.   :  15133    Min.   :  178737   Min.   :  178587
##  1st Qu.: 1652585   1st Qu.:  48170    1st Qu.:  657056   1st Qu.:  663786
##  Median : 3997978   Median : 108822    Median : 1675988   Median : 1684341
##  Mean   : 5767107   Mean   : 161021    Mean   : 2482331   Mean   : 2494933
##  3rd Qu.: 6611215   3rd Qu.: 188730    3rd Qu.: 2990530   3rd Qu.: 2993158
##  Max.   :39250017   Max.   :1448488    Max.   :16600851   Max.   :16633834
##
##  month3_emplvl      total_qtrly_wages   taxable_qtrly_wages avg_wkly_wage
##  Min.   :  181521   Min.   :8.811e+08   Min.   :0.000e+00   Min.   : 301.0
##  1st Qu.:  667492   1st Qu.:5.403e+09   1st Qu.:0.000e+00   1st Qu.: 515.2
##  Median : 1699044   Median :1.362e+10   Median :1.096e+09   Median : 658.0
##  Mean   : 2510204   Mean   :2.402e+10   Mean   :3.776e+09   Mean   : 674.8
##  3rd Qu.: 3016494   3rd Qu.:2.973e+10   3rd Qu.:4.177e+09   3rd Qu.: 804.0
##  Max.   :16606038   Max.   :2.753e+11   Max.   :7.689e+10   Max.   :1792.0
##
##     year_qtr       treated            gdpcapita         lngdp
##  Min.   :1990   Min.   :0.000000   Min.   :15029   Min.   : 9.351
##  1st Qu.:1996   1st Qu.:0.000000   1st Qu.:27989   1st Qu.:10.918
##  Median :2003   Median :0.000000   Median :36449   Median :11.780
##  Mean   :2003   Mean   :0.003048   Mean   :37808   Mean   :11.754
##  3rd Qu.:2010   3rd Qu.:0.000000   3rd Qu.:45531   3rd Qu.:12.529
##  Max.   :2016   Max.   :1.000000   Max.   :84382   Max.   :14.759
##
##   lngdpcapita     revstatecapita  revlocalcapita   emplvl1capita
##  Min.   : 9.618   Min.   : 2021   Min.   : 883.6   Min.   :0.3249
##  1st Qu.:10.240   1st Qu.: 2903   1st Qu.:2012.4   1st Qu.:0.4113
##  Median :10.504   Median : 3380   Median :2428.3   Median :0.4356
##  Mean   :10.486   Mean   : 3742   Mean   :2480.2   Mean   :0.4368
##  3rd Qu.:10.726   3rd Qu.: 4048   3rd Qu.:2819.4   3rd Qu.:0.4621
##  Max.   :11.343   Max.   :20353   Max.   :7160.9   Max.   :1.0524
##                   NA's   :2850    NA's   :2850
##  emplvl2capita    emplvl3capita     emplvlcapita    totalwagescapita
##  Min.   :0.3251   Min.   :0.3289   Min.   :0.3269   Min.   : 1493
##  1st Qu.:0.4138   1st Qu.:0.4163   1st Qu.:0.4138   1st Qu.: 2941
##  Median :0.4378   Median :0.4406   Median :0.4378   Median : 3787
##  Mean   :0.4390   Mean   :0.4420   Mean   :0.4393   Mean   : 3869
##  3rd Qu.:0.4644   3rd Qu.:0.4676   3rd Qu.:0.4644   3rd Qu.: 4608
##  Max.   :1.0507   Max.   :1.0513   Max.   :1.0515   Max.   :10275
##
##  taxwagescapita   avgwklywagecapita  estabscapita         abb
##  Min.   :   0.0   Min.   : 301.0    Min.   :0.01992   Length:5250
##  1st Qu.:   0.0   1st Qu.: 515.2    1st Qu.:0.02553   Class :character
```

```
##  Median : 355.7   Median : 658.0    Median :0.02845   Mode  :character
##  Mean   : 728.8   Mean   : 674.8    Mean   :0.02928
##  3rd Qu.:1224.4   3rd Qu.: 804.0    3rd Qu.:0.03211
##  Max.   :5254.4   Max.   :1792.0    Max.   :0.07071
##
```

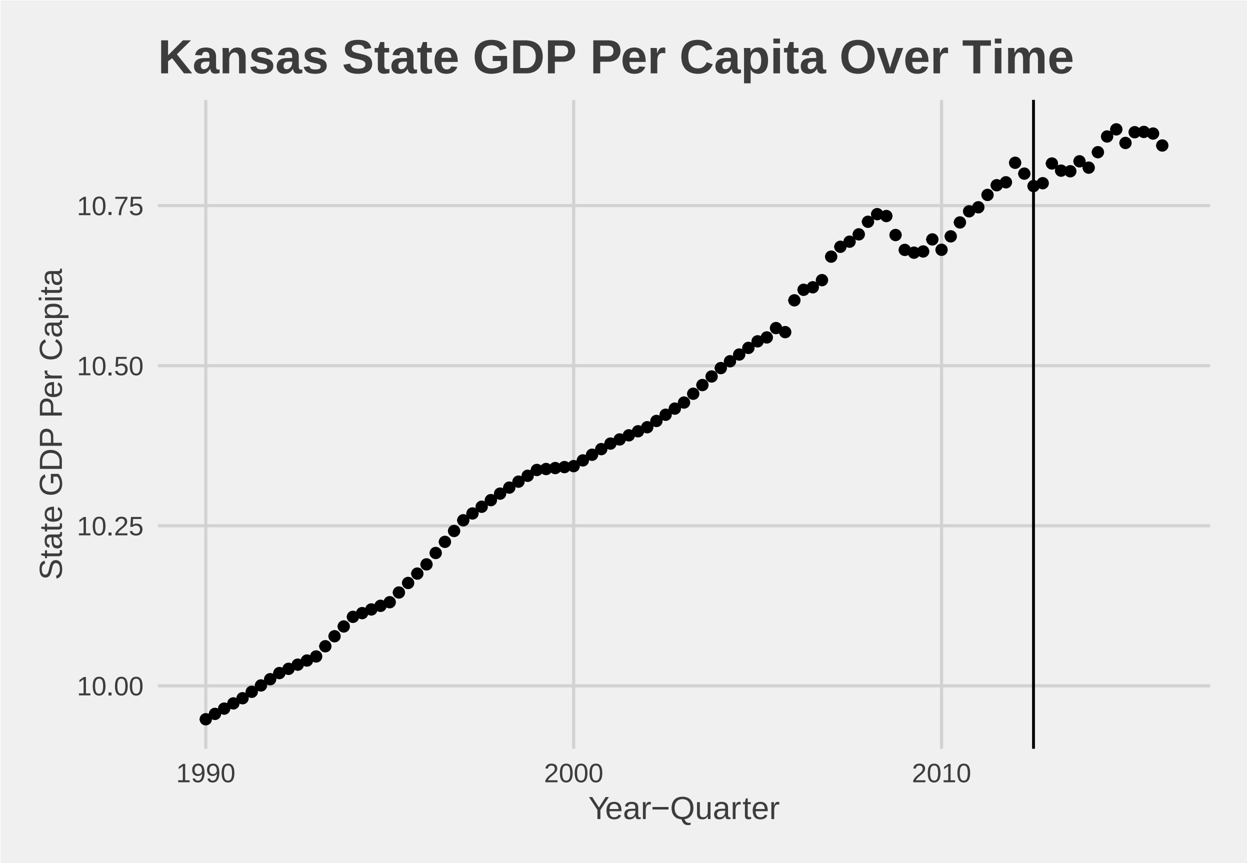
We have a lot of information here! We have quarterly state GDP from 1990 to 2016 for each U.S. state, as well as some other covariates. Let's begin by adding a treatment indicator to Kansas in Q2 2012 and onward.

```
kansas <- kansas %>%
  select(year, qtr, year_qtr, state, treated, gdp, lngdpcapita, fips) %>%
  mutate(treatment = ifelse(state == "Kansas" & year_qtr >= 2012.50,
                            1,
                            0))
head(kansas)
```

```
## # A tibble: 6 x 9
##    year   qtr year_qtr state   treated    gdp lngdpcapita  fips treatment
##   <dbl> <dbl>    <dbl> <chr>     <dbl>  <dbl>       <dbl> <dbl>     <dbl>
## 1  1990     1    1990  Alabama       0 71610         9.78     1         0
## 2  1990     2    1990. Alabama       0 72718.        9.79     1         0
## 3  1990     3    1990. Alabama       0 73826.        9.80     1         0
## 4  1990     4    1991. Alabama       0 74935.        9.82     1         0
## 5  1991     1    1991  Alabama       0 76043         9.83     1         0
## 6  1991     2    1991. Alabama       0 77347.        9.84     1         0
```

One approach might be to compare Kansas to itself pre- and post-treatment. If we plot state GDP over time we get something like this:

```
kansas %>%
  filter(state == 'Kansas') %>%
  ggplot() +
  geom_point(aes(x = year_qtr, y = lngdpcapita)) +
  geom_vline(xintercept = 2012.5) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  ggtitle('Kansas State GDP Per Capita Over Time') +
  xlab('Year-Quarter') +
  ylab('State GDP Per Capita')
```

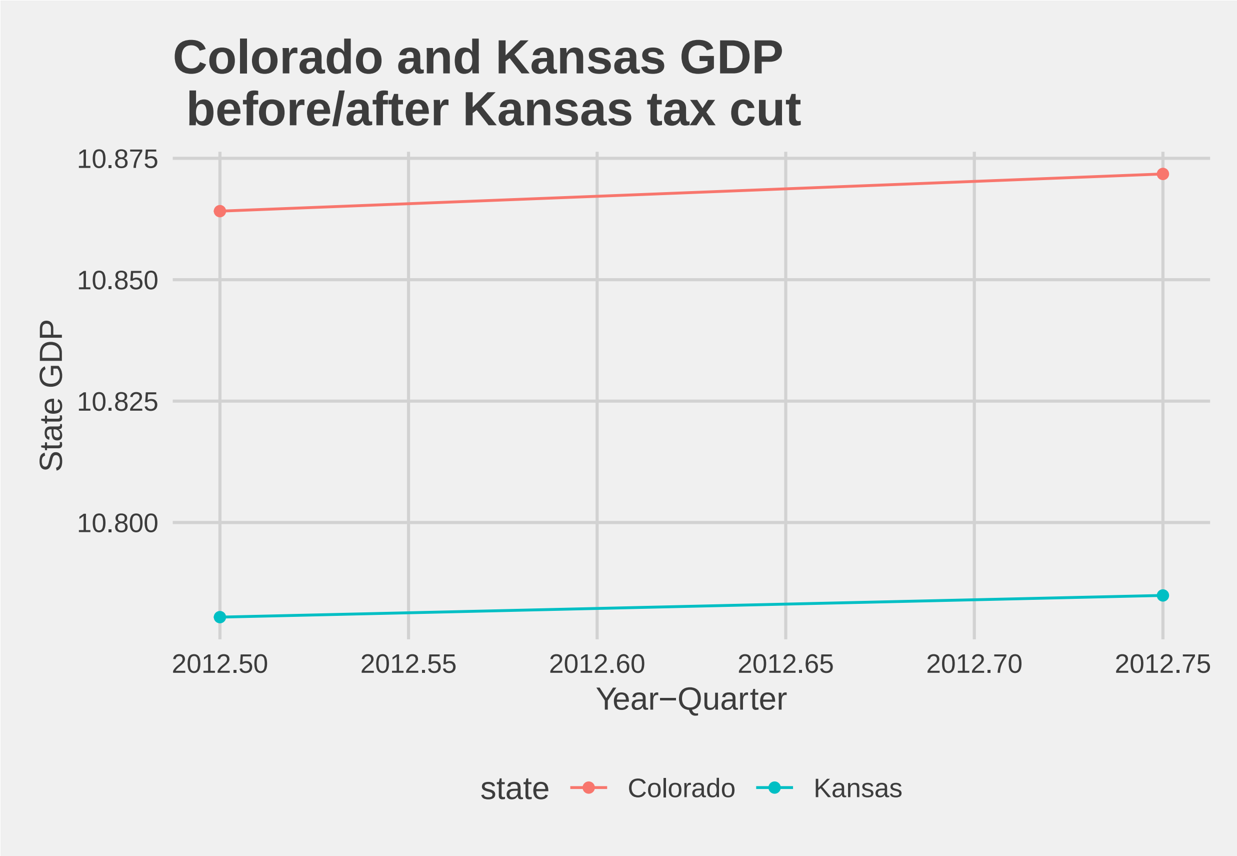


Question: Looks like GDP went up after the tax cut! What is the problem with this inference?

Solution: It looks like GDP went up after the tax cut, but we have no way of telling whether it went up because of the tax cut or went up because it would have otherwise. In short, we need to compare the treated Kansas to a counterfactual for if taxes weren't cut.

Ideally, we would like to compare treated Kansas to control Kansas. Because of the fundamental problem of causal inference, we will never oberserve both of these conditions though. The core idea behind DiD is that we could instead use the fact that our treated unit was similar to a control unit, and then measure the differences between them. Perhaps we could choose neighboring Colorado:

```
kansas %>%
  filter(state %in% c("Kansas","Colorado")) %>%
  filter(year_qtr >= 2012.5 & year_qtr<= 2012.75) %>%
  ggplot() +
  geom_point(aes(x = year_qtr,
                 y = lngdpcapita,
                 color = state)) +
  geom_line(aes(x = year_qtr,
                y = lngdpcapita,
                color = state)) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  ggtitle('Colorado and Kansas GDP \n before/after Kansas tax cut') +
  xlab('Year-Quarter') +
  ylab('State GDP')
```



This is basically what Card-Krueger (1994) did measuring unemployment rates among New Jersey and Pennsylvania fast food restaurants.

Challenge: Try writing a simple DiD estimate using dplyr/tidyr (use subtraction instead of a regression):

```
# kansas-colorado
kc <- kansas %>%
  filter(state %in% c("Kansas","Colorado")) %>%
  filter(year_qtr >= 2012.5 & year_qtr<= 2012.75)

# pre-treatment difference

pre_diff <- kc %>%
  filter(year_qtr == 2012.5) %>%
  select(state,
         lngdpcapita) %>%
  spread(state,
         lngdpcapita) %>%
  summarise(Colorado - Kansas)

# post-treatment difference

post_diff <- kc %>%
  filter(year_qtr == 2012.75) %>%
  select(state,
         lngdpcapita) %>%
  spread(state,
```

```
        lngdpcapita) %>%
  summarise(Colorado - Kansas)

# diff-in-diffs

diff_in_diffs <- post_diff - pre_diff
diff_in_diffs
```

```
##   Colorado - Kansas
## 1       0.003193447
```

Looks like our treatment effect is about .003 (in logged thousands dollars per capita). Again this is the basic idea behind Card-Krueger.

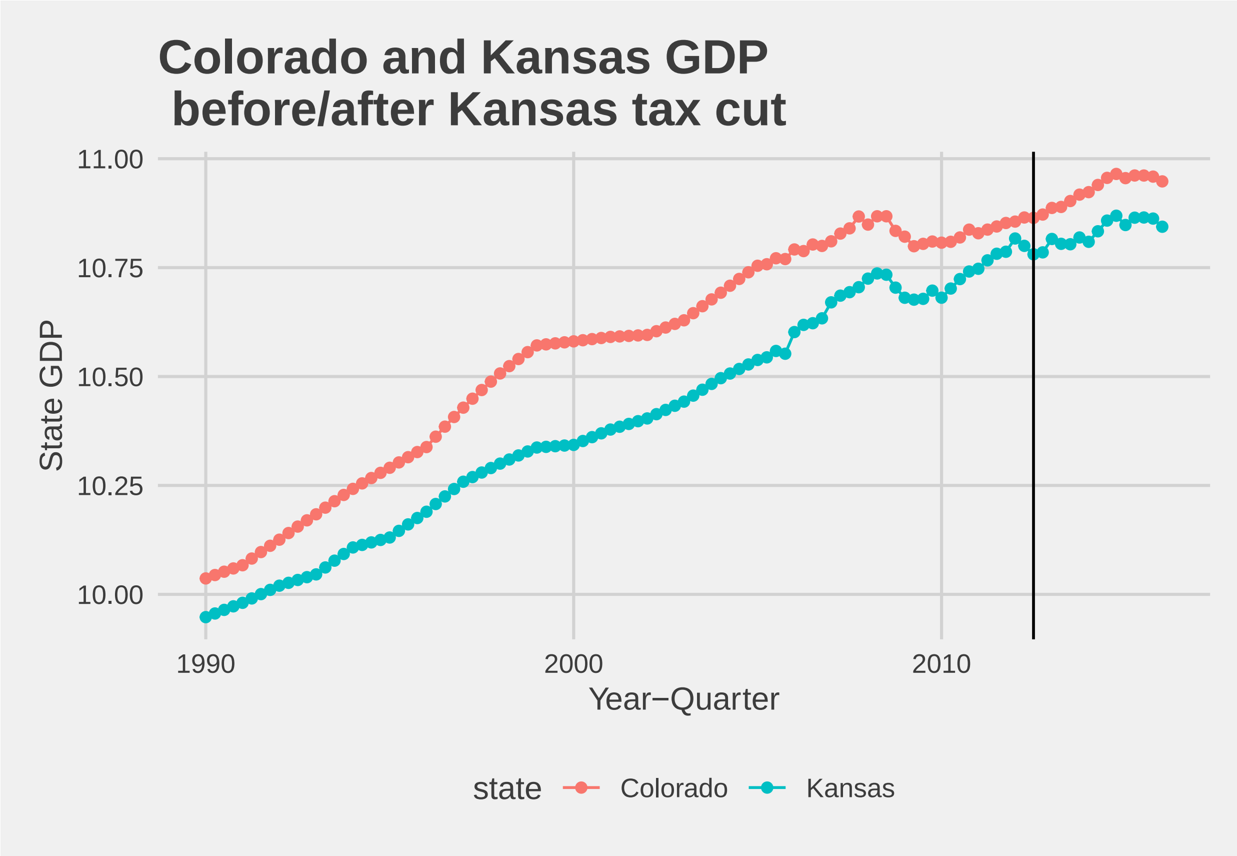
Question: Why might there still be a problem with this estimate?

Answer: We just assumed that Colorado was similar to Kansas because they are neighbors - we don't really have evidence for this idea.

Parallel Trends Assumptions

One of the core assumptions for difference-in-differences estimation is the "parallel trends" or "constant trends" assumption. Essentially, this assumption requires that the difference between our treatment and control units are constant in the pre-treatment period. Let's see how Kansas and Colorado do on this assumption:

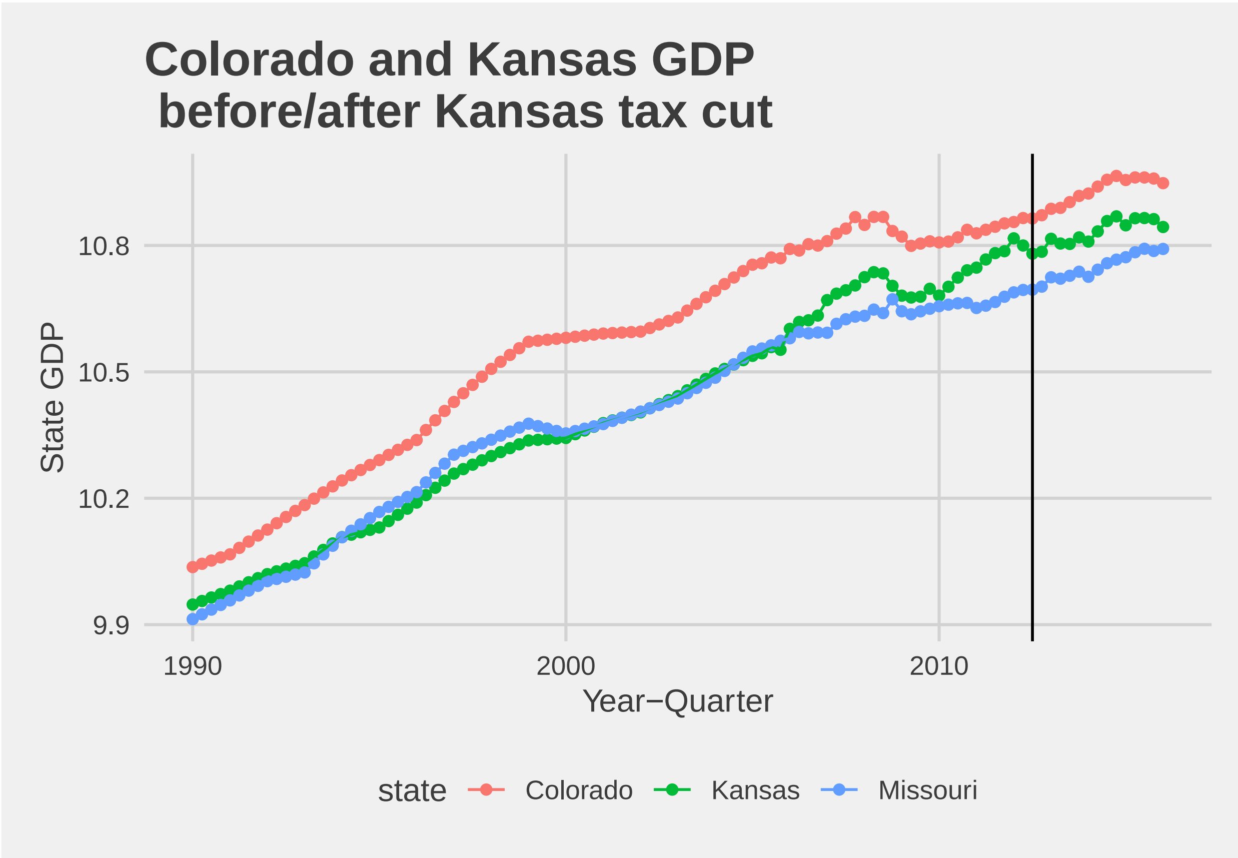
```
kansas %>%
  filter(state %in% c("Kansas","Colorado")) %>%
  ggplot() +
  geom_point(aes(x = year_qtr,
                 y = lngdpcapita,
                 color = state)) +
  geom_line(aes(x = year_qtr,
                y = lngdpcapita,
                color = state)) +
  geom_vline(aes(xintercept = 2012.5)) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  ggtitle('Colorado and Kansas GDP \n before/after Kansas tax cut') +
  xlab('Year-Quarter') +
  ylab('State GDP')
```



The two lines somewhat move together, but the gap does grow and shrink at various points over time. The most concerning part here is that the gap quickly shrinks right before treatment. What do we do if we do not trust the parallel trends assumption? Perhaps we pick a different state.

Challenge: Choose another state that you think would be good to try out, and plot it alongside Kansas and Colorado.

```
kansas %>%
  filter(state %in% c("Kansas","Colorado", "Missouri")) %>%
  ggplot() +
  geom_point(aes(x = year_qtr,
                 y = lngdpcapita,
                 color = state)) +
  geom_line(aes(x = year_qtr,
                y = lngdpcapita,
                color = state)) +
  geom_vline(aes(xintercept = 2012.5)) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  ggtitle('Colorado and Kansas GDP \n before/after Kansas tax cut') +
  xlab('Year-Quarter') +
  ylab('State GDP')
```



Question: Would Colorado or your choice? be the more plausible control unit in this case? Why?

Solution: There is a good argument for both of them (Missouri in this case). However, the gap between Colorado and Kansas closes quickly before the treatment period, and similarly it grows between between Kansas and Missouri at the same point.

Selecting comparative units this way can be hard to justify theoretically, and sometimes we do not have a good candidate. What can we do then? This is where synthetic control comes in.

Synthetic Control

Synthetic control is motivated by the problem of choosing comparison units for comparative case studies. It aims to create a "synthetic" version of the treatment unit by combining and weighting covariates from other units ("donors"). In this case, we would construct a synthetic Kansas by creating a weighted average of the other 49 U.S. states. Ideally, the synthetic unit would match the treatment unit in the pre-treatment periods.

For constructing a synthetic control, we are going to use the `augsynth` library. The basic syntax for this library is:

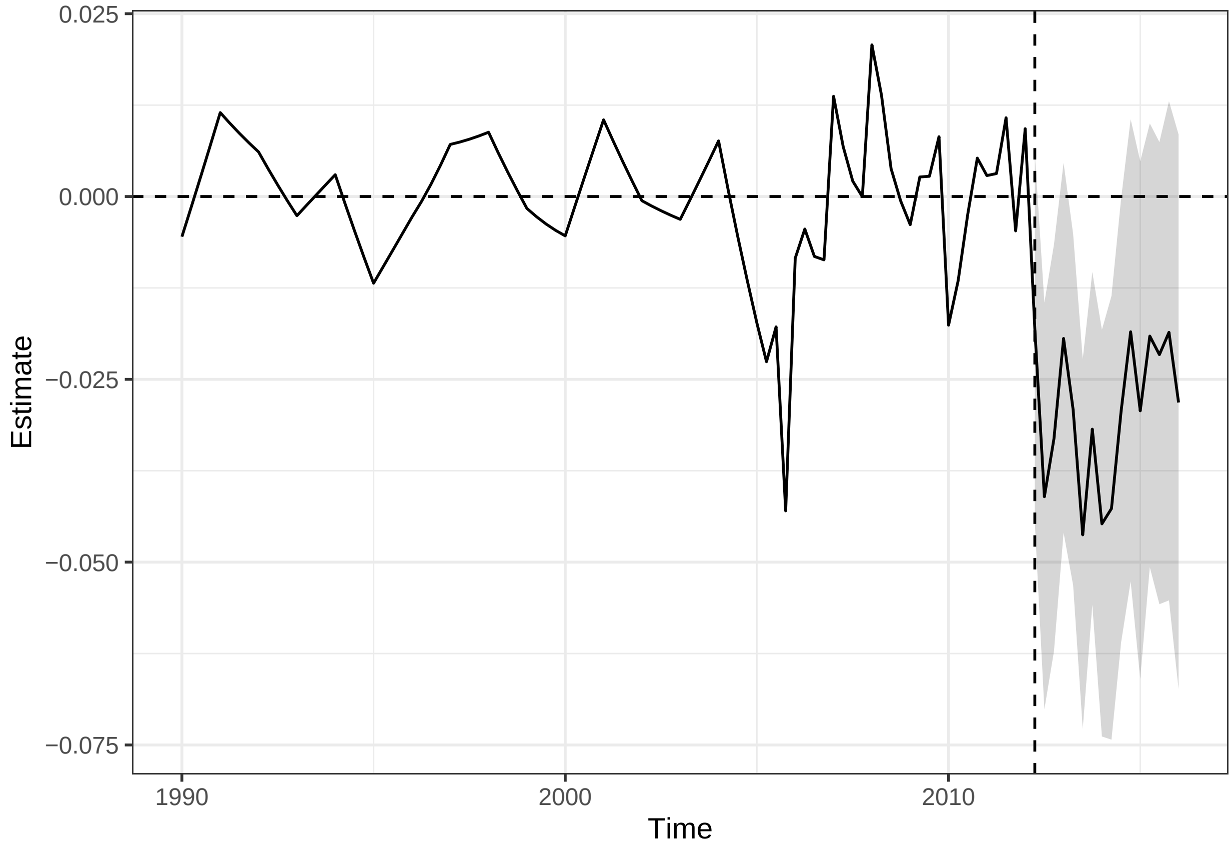
`augsynth(outcome ~ trt, unit, time, t_int, data)`

```
syn <- augsynth(lngdpcapita ~ treated, state, year_qtr, kansas,
                progfunc = "None", scm = T)

summary(syn)
```

```
##
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##     t_int = t_int, data = data, progfunc = "None", scm = ..2)
##
## Average ATT Estimate (p Value for Joint Null):  -0.029   ( 0.332 )
## L2 Imbalance: 0.083
## Percent improvement from uniform weights: 79.5%
##
## Avg Estimated Bias: NA
##
## Inference type: Conformal inference
##
##     Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
##  2012.25   -0.018             -0.045              0.006   0.111
##  2012.50   -0.041             -0.070             -0.015   0.022
##  2012.75   -0.033             -0.062             -0.007   0.044
##  2013.00   -0.019             -0.046              0.005   0.111
##  2013.25   -0.029             -0.053             -0.005   0.044
##  2013.50   -0.046             -0.073             -0.022   0.022
##  2013.75   -0.032             -0.056             -0.010   0.022
##  2014.00   -0.045             -0.074             -0.018   0.022
##  2014.25   -0.043             -0.074             -0.014   0.022
##  2014.50   -0.029             -0.061              0.000   0.044
##  2014.75   -0.018             -0.053              0.011   0.144
##  2015.00   -0.029             -0.066              0.005   0.078
##  2015.25   -0.019             -0.051              0.010   0.122
##  2015.50   -0.022             -0.056              0.007   0.111
##  2015.75   -0.019             -0.055              0.013   0.189
##  2016.00   -0.028             -0.067              0.008   0.100
```

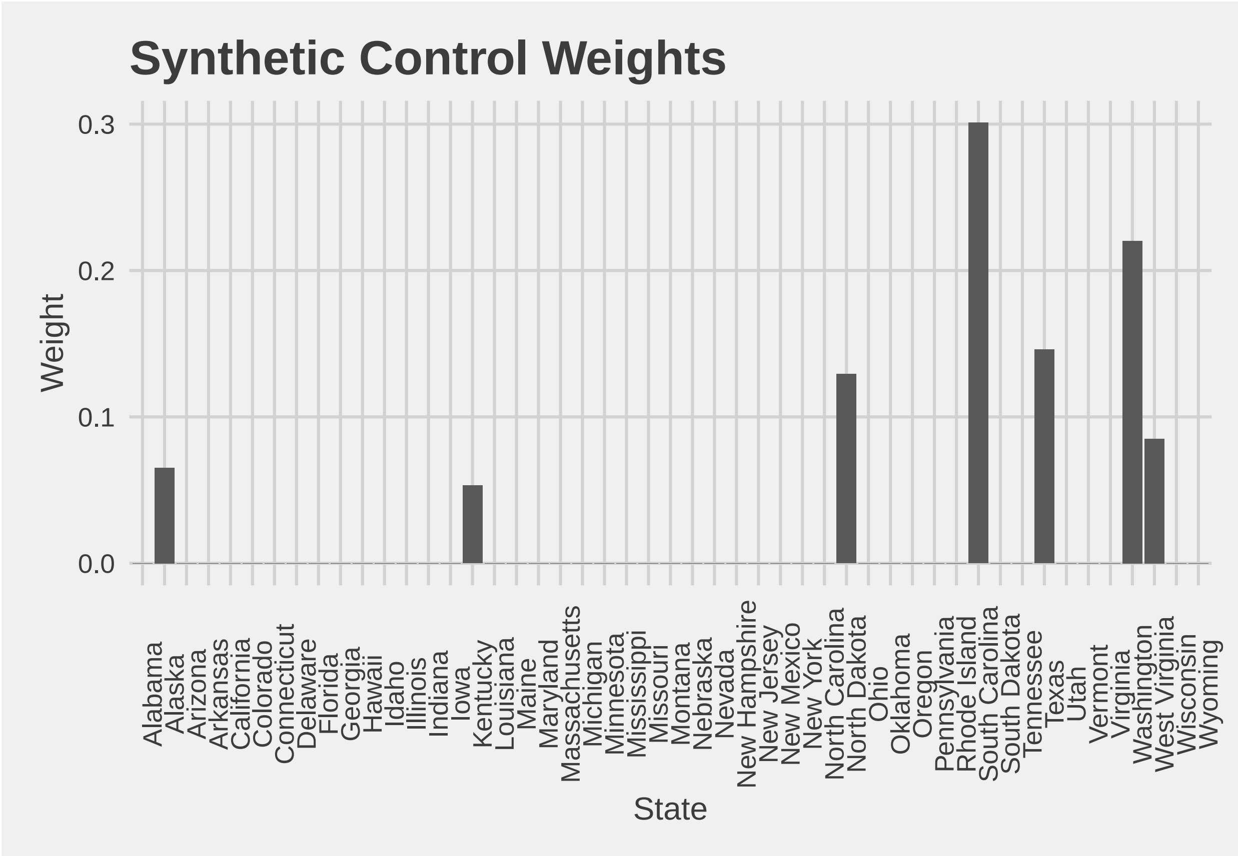
We can use the built in plot function to see how Kansas did relative to synthetic Kansas:

```
plot(syn)
```



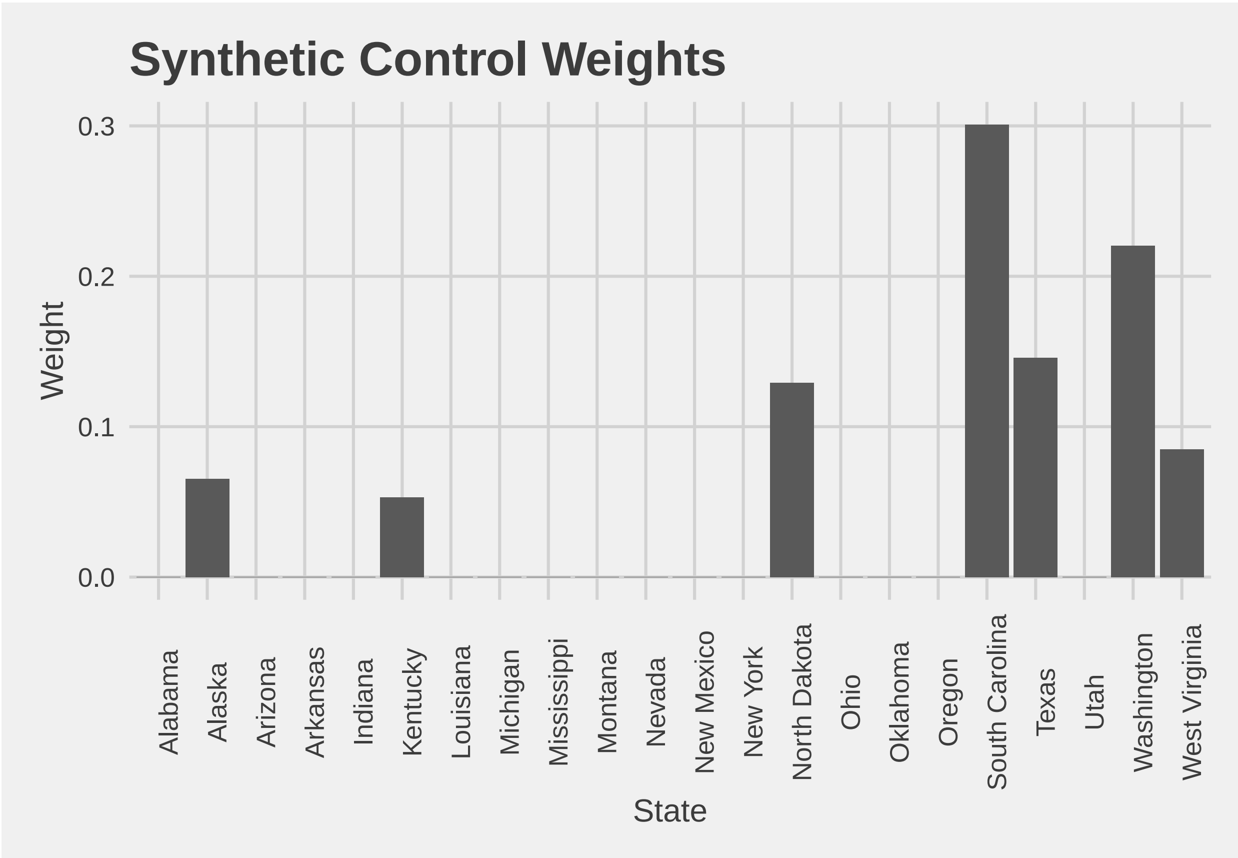
We can see which donors contributed the most to the synthetic Kansas:

```
# Convert weights to dataframe
data.frame(syn$weights) %>%
  # change index to a column
  tibble::rownames_to_column('State') %>%
  ggplot() +
  # stat = identity to take the literal value instead of a count for geom_bar()
  geom_bar(aes(x = State,
               y = syn.weights),
           stat = 'identity') +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(),
        axis.text.x = element_text(angle = 90)) +
  ggtitle('Synthetic Control Weights') +
  xlab('State') +
  ylab('Weight')
```



Surprisingly, only a few units ended up contributing! Let's take a closer look at the ones that did:

```
data.frame(syn$weights) %>%
  tibble::rownames_to_column('State') %>%
  filter(syn.weights > 0) %>%
  ggplot() +
  geom_bar(aes(x = State,
               y = syn.weights),
           stat = 'identity') +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(),
        axis.text.x = element_text(angle = 90)) +
  ggtitle('Synthetic Control Weights') +
  xlab('State') +
  ylab('Weight')
```

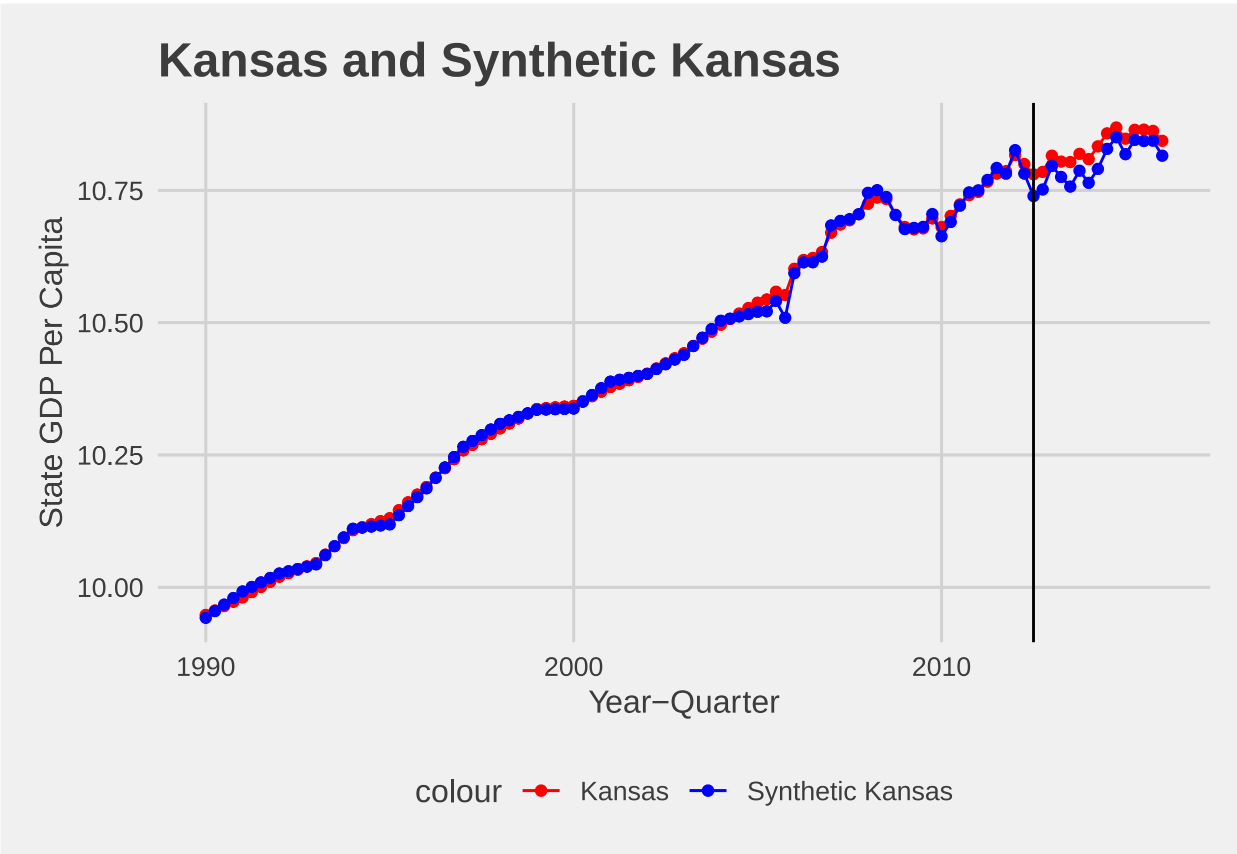


Synthetic Control Augmentation

The main advantage of the `asynth` package is that it allows for "augmented synthetic control". One of the main problems with synthetic control is that if the pre-treatment balance between treatment and control outcomes is poor, the estimate is not valid. Specifically, they advocate for using L2 imbalance, which he first encountered as the penalty that ridge regression uses. L2 uses "squared magnitude" of the coefficient to penalize a particular feature.

```
# Aniket's method for getting the underlying data
syn_sum <- summary(syn)

kansas_synkansas <- kansas %>%
  filter(state == "Kansas") %>%
  bind_cols(difference = syn_sum$att$Estimate) %>%
  mutate(synthetic_kansas = lngdpcapita + difference)

# Plot

kansas_synkansas %>%
  ggplot() +
  geom_point(aes(x = year_qtr,
                 y = lngdpcapita,
                 color = 'Kansas')) +
  geom_line(aes(x = year_qtr,
```

```
                y = lngdpcapita,
                color = 'Kansas')) +
  geom_point(aes(x = year_qtr,
                 y = synthetic_kansas,
                 color = 'Synthetic Kansas')) +
  geom_line(aes(x = year_qtr,
                y = synthetic_kansas,
                color = 'Synthetic Kansas')) +
  scale_color_manual(values = c('Kansas' = 'red', 'Synthetic Kansas' = 'blue')) +
  geom_vline(aes(xintercept = 2012.5)) +
  theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  ggtitle('Kansas and Synthetic Kansas') +
  xlab('Year-Quarter') +
  ylab('State GDP Per Capita')
```



Question: How does pre-treatment matching between Kansas and Synthetic Kansas look here?

Answer: Pretty good! We may not need to augment this synthetic control, though let's try anyway.

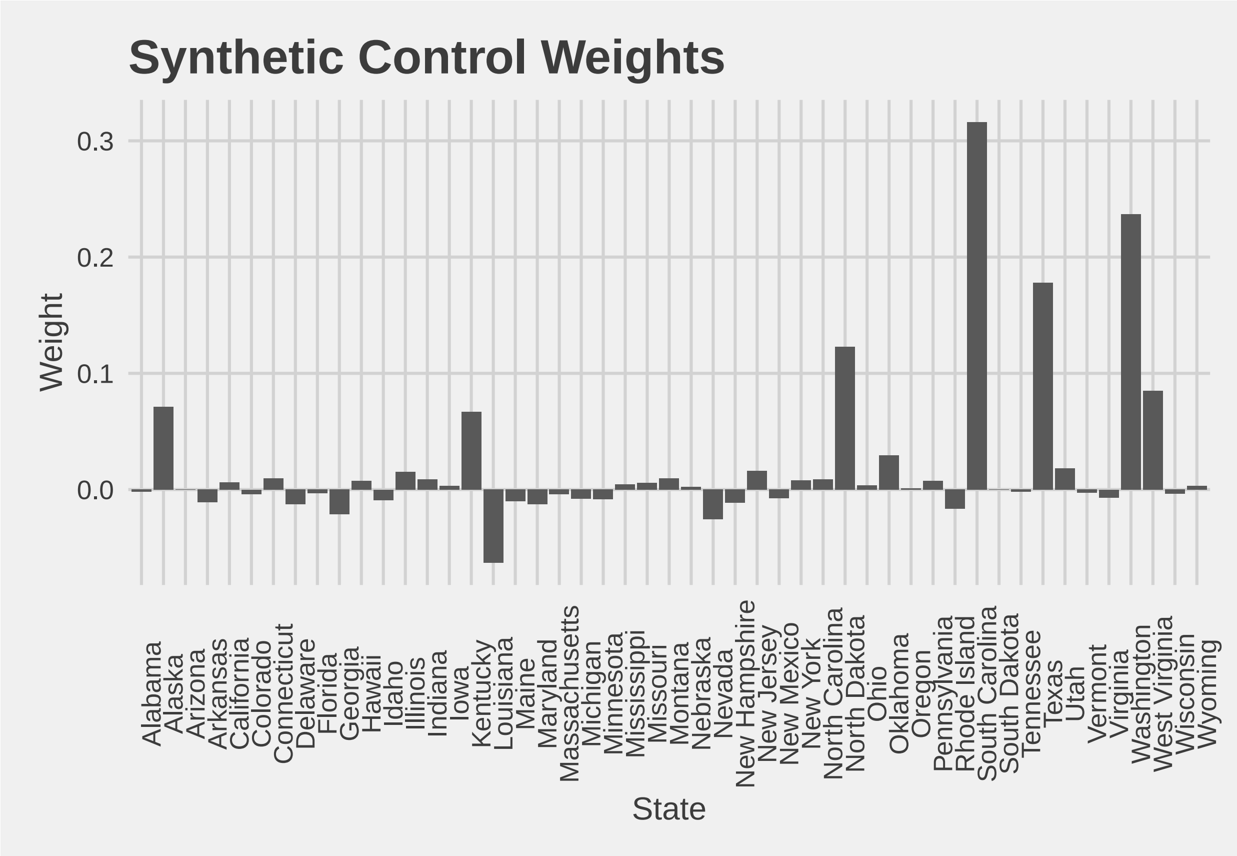
```
ridge_syn <- augsynth(lngdpcapita ~ treated, state, year_qtr, kansas,
                      progfunc = "ridge", scm = T)

summary(ridge_syn)
```

```
##
```

```
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##     t_int = t_int, data = data, progfunc = "ridge", scm = ..2)
##
## Average ATT Estimate (p Value for Joint Null):  -0.040   ( 0.081 )
## L2 Imbalance: 0.062
## Percent improvement from uniform weights: 84.7%
##
## Avg Estimated Bias: 0.011
##
## Inference type: Conformal inference
##
##     Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
##  2012.25   -0.022             -0.044              0.003   0.056
##  2012.50   -0.047             -0.076             -0.018   0.022
##  2012.75   -0.043             -0.071             -0.010   0.022
##  2013.00   -0.030             -0.055             -0.004   0.033
##  2013.25   -0.041             -0.067             -0.012   0.022
##  2013.50   -0.059             -0.088             -0.030   0.022
##  2013.75   -0.045             -0.073             -0.019   0.022
##  2014.00   -0.058             -0.090             -0.026   0.022
##  2014.25   -0.055             -0.091             -0.020   0.022
##  2014.50   -0.041             -0.080             -0.006   0.033
##  2014.75   -0.029             -0.068              0.006   0.056
##  2015.00   -0.040             -0.082              0.000   0.056
##  2015.25   -0.030             -0.066              0.002   0.056
##  2015.50   -0.033             -0.072              0.003   0.056
##  2015.75   -0.029             -0.071              0.010   0.056
##  2016.00   -0.038             -0.087              0.004   0.056
```

Let's look at the weights:

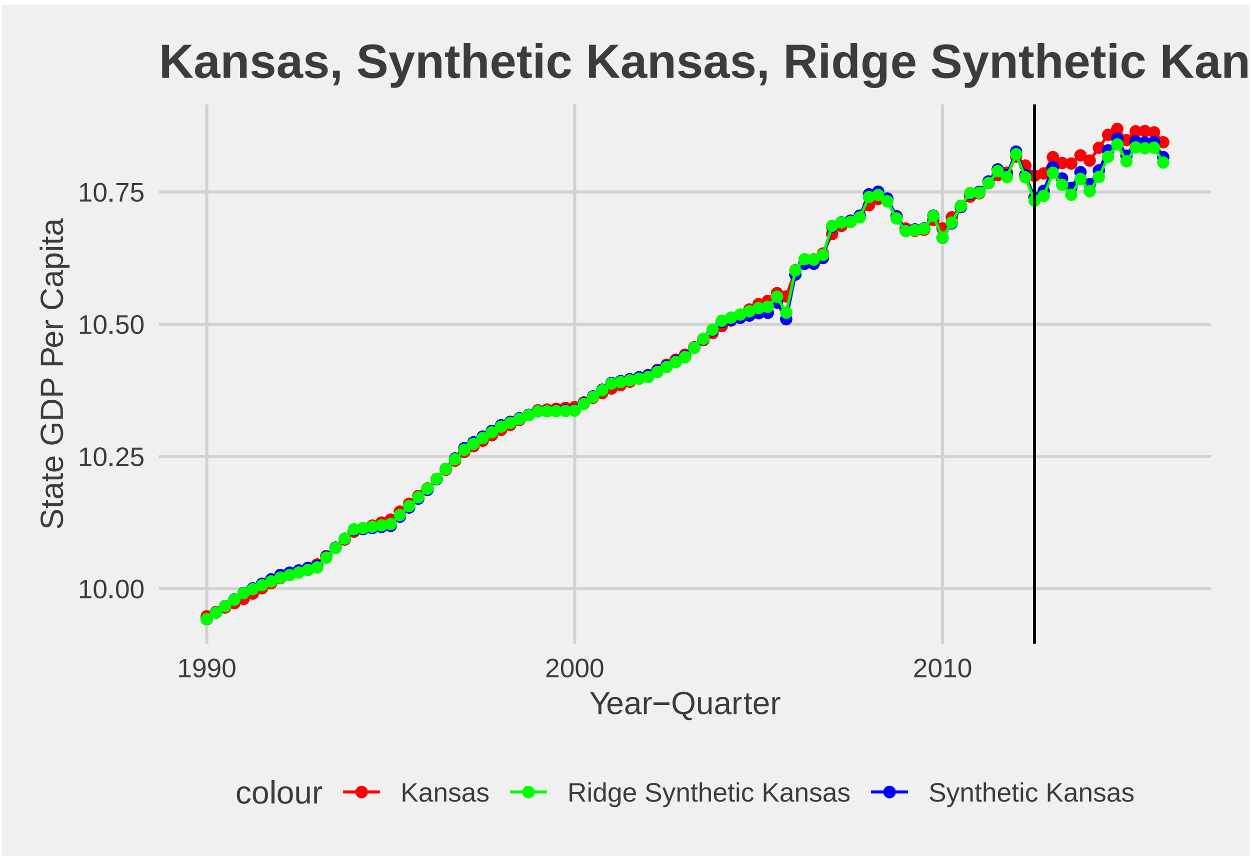
```
data.frame(ridge_syn$weights) %>%
  tibble::rownames_to_column('State') %>%
  ggplot() +
  geom_bar(aes(x = State, y = ridge_syn.weights),
           stat = 'identity') +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(),
        axis.text.x = element_text(angle = 90)) +
  ggtitle('Synthetic Control Weights') +
  xlab('State') +
  ylab('Weight')
```



Notice how with the ridge augmentation, some weights are allowed to be negative now. Now let's go ahead and plot the ridge augmented synthetic Kansas alongside Kansas and synthetic Kansas:

```
ridge_sum <- summary(ridge_syn)

kansas_synkansas_ridgesynkansas <- kansas_synkansas %>%
  bind_cols(ridge_difference = ridge_sum$att$Estimate) %>%
  mutate(ridge_synthetic_kansas = lngdpcapita + ridge_difference)

kansas_synkansas_ridgesynkansas %>%
  ggplot() +
  geom_point(aes(x = year_qtr,
                 y = lngdpcapita,
                 color = 'Kansas')) +
  geom_line(aes(x = year_qtr,
                y = lngdpcapita,
                color = 'Kansas')) +
  geom_point(aes(x = year_qtr,
                 y = synthetic_kansas,
                 color = 'Synthetic Kansas')) +
  geom_line(aes(x = year_qtr,
                y = synthetic_kansas,
                color = 'Synthetic Kansas')) +
  geom_point(aes(x = year_qtr,
                 y = ridge_synthetic_kansas,
                 color = 'Ridge Synthetic Kansas')) +
```

```
geom_line(aes(x = year_qtr,
              y = ridge_synthetic_kansas,
              color = 'Ridge Synthetic Kansas')) +
scale_color_manual(values = c('Kansas' = 'red',
                              'Synthetic Kansas' = 'blue',
                              'Ridge Synthetic Kansas' = 'green')) +
geom_vline(aes(xintercept = 2012.5)) +
theme_fivethirtyeight() +
theme(axis.title = element_text()) +
ggtitle('Kansas, Synthetic Kansas, Ridge Synthetic Kansas') +
xlab('Year-Quarter') +
ylab('State GDP Per Capita')
```



These all seem pretty good! Like we thought, augmentation did not necessarily improve the matches in this particular dataset. We can check the two L2 imbalances and see that we have reduced the overall imbalance a bit with our ridge model:

```
print(syn$l2_imbalance)
```

```
## [1] 0.08255471
```

```
print(ridge_syn$l2_imbalance)
```

```
## [1] 0.06151525
```

Finally, we can add covariates to our model if we would like:

```
data(kansas)

covsyn <- augsynth(lngdpcapita ~ treated | lngdpcapita + log(revstatecapita) +
                                           log(revlocalcapita) + log(avgwklywagecapita) +
                                           estabscapita + emplvlcapita,
                   fips, year_qtr, kansas,
                   progfunc = "ridge", scm = T)

summary(covsyn)
```

```
##
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##     t_int = t_int, data = data, progfunc = "ridge", scm = ..2)
##
## Average ATT Estimate (p Value for Joint Null):  -0.061   ( 0.136 )
## L2 Imbalance: 0.054
## Percent improvement from uniform weights: 86.6%
##
## Covariate L2 Imbalance: 0.005
## Percent improvement from uniform weights: 97.7%
##
## Avg Estimated Bias: 0.027
##
## Inference type: Conformal inference
##
##     Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
##  2012.25   -0.021             -0.044              0.002   0.067
##  2012.50   -0.047             -0.076             -0.014   0.033
##  2012.75   -0.050             -0.083             -0.007   0.033
##  2013.00   -0.045             -0.074             -0.012   0.033
##  2013.25   -0.055             -0.088             -0.022   0.022
##  2013.50   -0.071             -0.105             -0.033   0.022
##  2013.75   -0.058             -0.091             -0.025   0.022
##  2014.00   -0.081             -0.119             -0.037   0.022
##  2014.25   -0.078             -0.121             -0.034   0.022
##  2014.50   -0.065             -0.114             -0.021   0.033
##  2014.75   -0.057             -0.110             -0.008   0.044
##  2015.00   -0.075             -0.124             -0.022   0.033
##  2015.25   -0.063             -0.106             -0.014   0.033
##  2015.50   -0.067             -0.106             -0.019   0.022
##  2015.75   -0.063             -0.101             -0.009   0.022
##  2016.00   -0.078             -0.122             -0.019   0.022
```

Staggered Adoption