Project 7: Difference-in-Differences and Synthetic Control

```r
# Install and load packages
#if (!require("pacman")) install.packages("pacman")
#devtools::install_github("susanathey/MCPanel")
#install.packages("C:/Users/MACKGS~1/AppData/Local/Temp/RtmpiszvSs/filef4a06b5d5e78/MCPanel_0.0.tar.gz"

#devtools::install_github("ebenmichael/augsynth")
#install.packages("tinytex")
#tinytex::install_tinytex()

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


library(dplyr)
library(augsynth)
library(ggplot2)
library(tidyr)
#library(MCPanel) # this worked once but I haven't been able to get it to work again. It also didn't ch
library(lubridate)
library(knitr)


# set seed
set.seed(44)

# load data
medicaid_expansion <- read_csv('./data/medicaid_expansion.csv')
```

```
## Rows: 663 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): State
## dbl  (3): year, uninsured_rate, population
## date (1): Date_Adopted
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#medicaid_expansion <- read_csv("C:/git/Computational-Social-Science-Projects/Project 7/data/medicaid_e
```

```r
# # Set CRAN mirror first
# options(repos = c(CRAN = "https://cloud.r-project.org"))
#
# # Then install packages
# if (!requireNamespace("tinytex", quietly = TRUE)) {
#   install.packages("tinytex")
# }
#
# if (!requireNamespace("devtools", quietly = TRUE)) {
#   install.packages("devtools")
# }
#
# if (!requireNamespace("multisynth", quietly = TRUE)) {
#   devtools::install_github("ebenmichael/multisynth")
# }
#
# # Load required libraries
# library(multisynth)
```

Introduction

For this project, you will explore the question of whether the Affordable Care Act increased health insurance coverage (or conversely, decreased the number of people who are uninsured). The ACA was passed in March 2010, but several of its provisions were phased in over a few years. The ACA instituted the "individual mandate" which required that all Americans must carry health insurance, or else suffer a tax penalty. There are four mechanisms for how the ACA aims to reduce the uninsured population:

- Require companies with more than 50 employees to provide health insurance.
- Build state-run healthcare markets ("exchanges") for individuals to purchase health insurance.
- Provide subsidies to middle income individuals and families who do not qualify for employer based coverage.
- Expand Medicaid to require that states grant eligibility to all citizens and legal residents earning up to 138% of the federal poverty line. The federal government would initially pay 100% of the costs of this expansion, and over a period of 5 years the burden would shift so the federal government would pay 90% and the states would pay 10%.

In 2012, the Supreme Court heard the landmark case NFIB v. Sebelius, which principally challenged the constitutionality of the law under the theory that Congress could not institute an individual mandate. The Supreme Court ultimately upheld the individual mandate under Congress's taxation power, but struck down the requirement that states must expand Medicaid as impermissible subordination of the states to the federal government. Subsequently, several states refused to expand Medicaid when the program began on January 1, 2014. This refusal created the "Medicaid coverage gap" where there are indivudals who earn too much to qualify for Medicaid under the old standards, but too little to qualify for the ACA subsidies targeted at middle-income individuals.

States that refused to expand Medicaid principally cited the cost as the primary factor. Critics pointed out however, that the decision not to expand primarily broke down along partisan lines. In the years since the initial expansion, several states have opted into the program, either because of a change in the governing party, or because voters directly approved expansion via a ballot initiative.

You will explore the question of whether Medicaid expansion reduced the uninsured population in the U.S. in the 7 years since it went into effect. To address this question, you will use difference-in-differences estimation, and synthetic control.

Data

The dataset you will work with has been assembled from a few different sources about Medicaid. The key variables are:

- **State**: Full name of state
- **Medicaid Expansion Adoption**: Date that the state adopted the Medicaid expansion, if it did so.
- **Year**: Year of observation.
- **Uninsured rate**: State uninsured rate in that year.

Exploratory Data Analysis

Create plots and provide 1-2 sentence analyses to answer the following questions:

- Which states had the highest uninsured rates prior to 2014? The lowest?

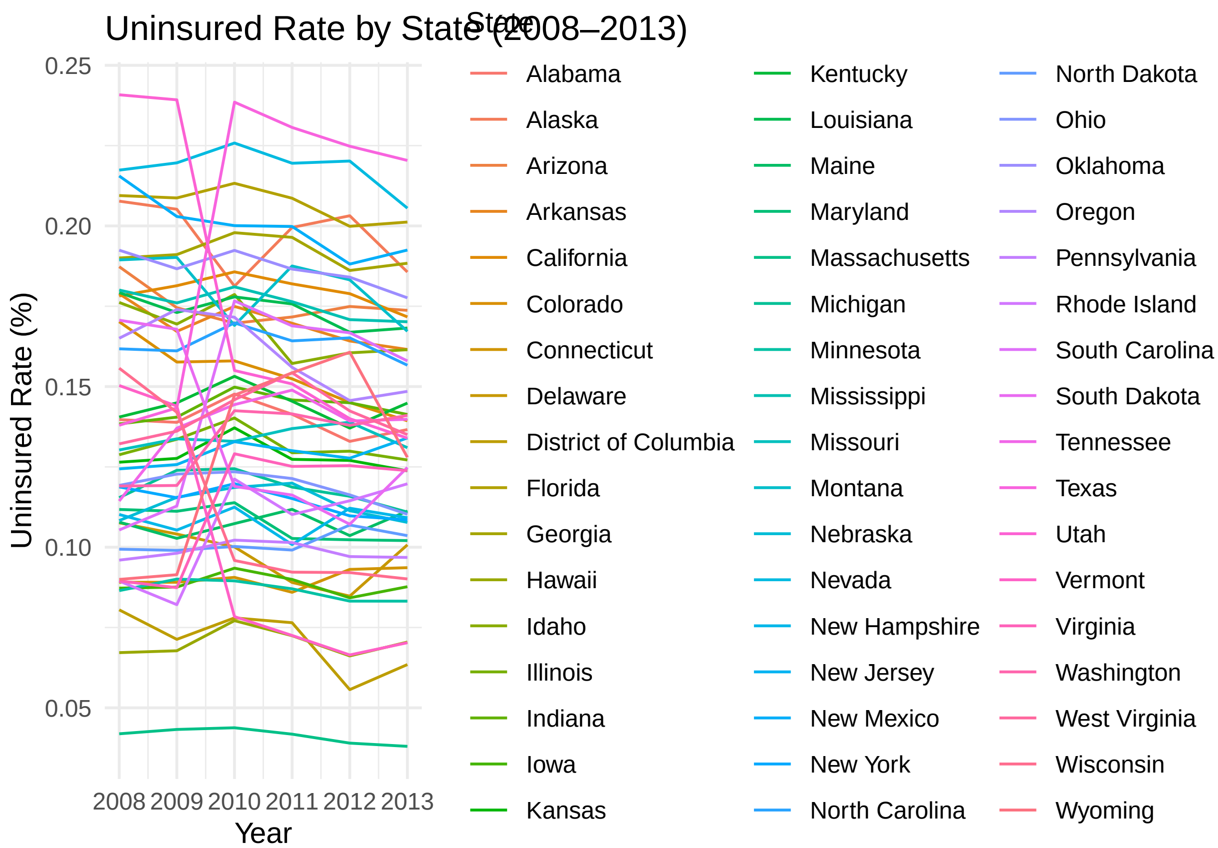
Texas had the highest single uninsured rate, and Nevada had the highest average uninsured rate across that time. Lowest were Massachusetts, Hawaii, and Vermont, which is not surprising.

- Which states were home to most uninsured Americans prior to 2014? Looks like California and Texas, which makes sense because of population. Florida is third,and New York is fourth. California had the highest average total over the pre-2014 time period.

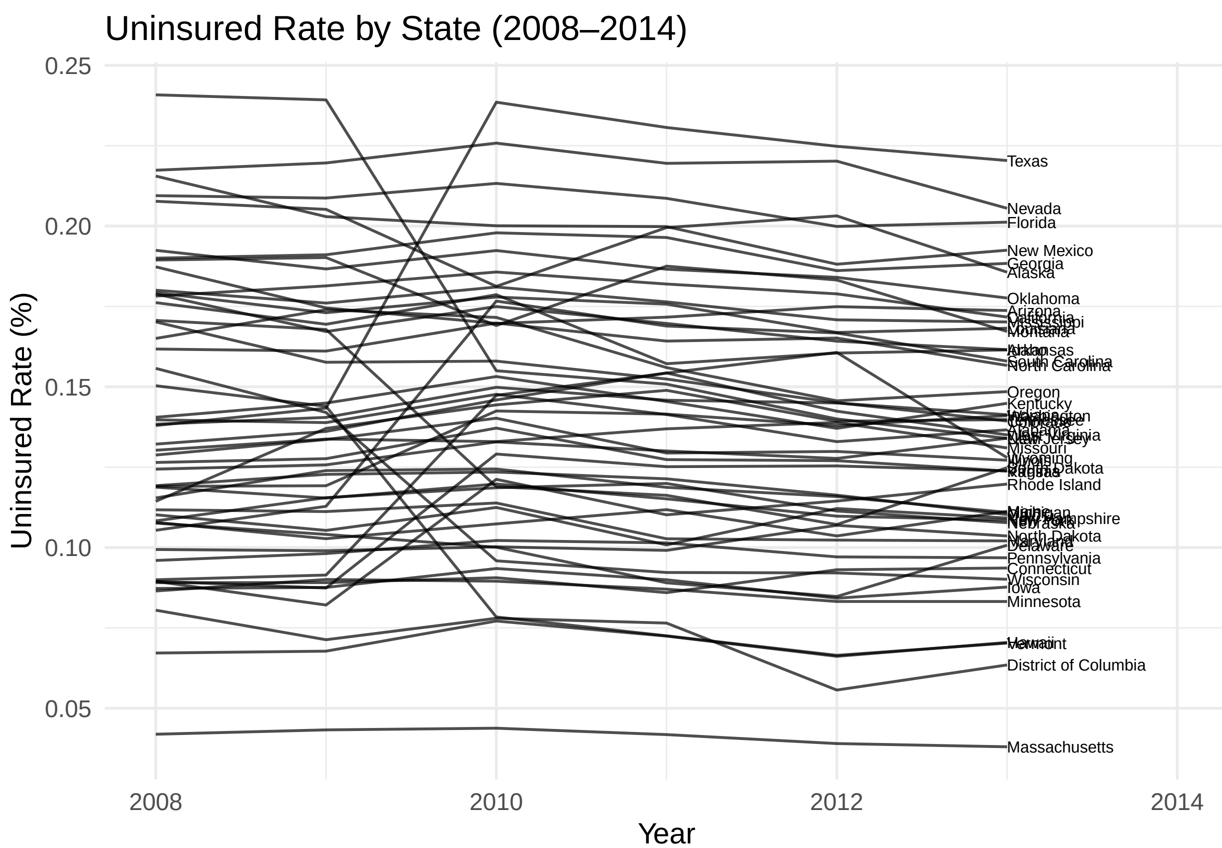
- How about in the last year in the data set? In the last year, 2020, Texas has surpassed California (but CA is still 2nd). Florida and Georgia are still up there but New York has dropped down to fifth or sixth.

Note: 2010 state population is provided as a variable to answer this question. In an actual study you would likely use population estimates over time, but to simplify you can assume these numbers stay about the same.

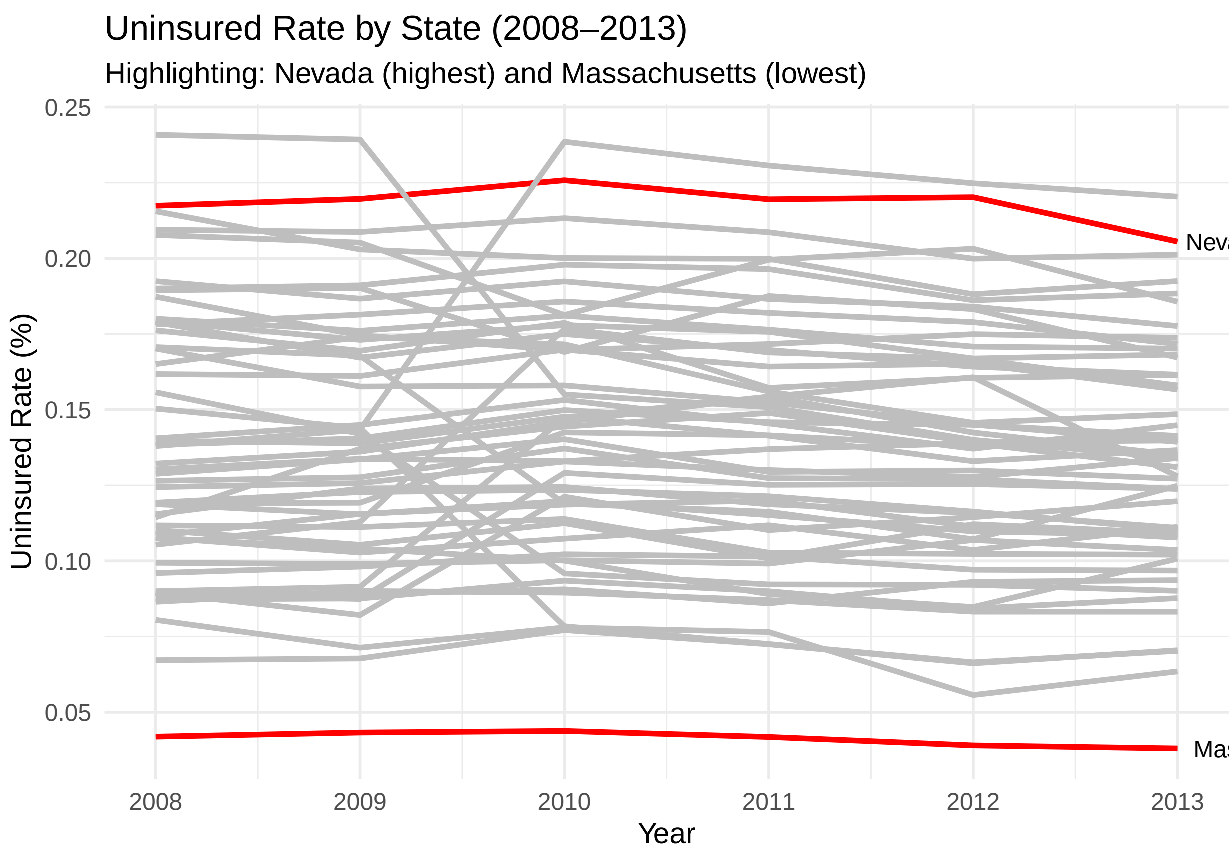
```
# highest and lowest uninsured rates

medicaid_expansion_to_2014 <- medicaid_expansion %>%
  filter(year >= 2008, year < 2014)

##nope
ggplot(medicaid_expansion_to_2014, aes(x = year, y = uninsured_rate, group = State, color = State)) +
  geom_line() +
  labs(title = "Uninsured Rate by State (2008-2013)",
       x = "Year",
       y = "Uninsured Rate (%)") +
  theme_minimal()
```



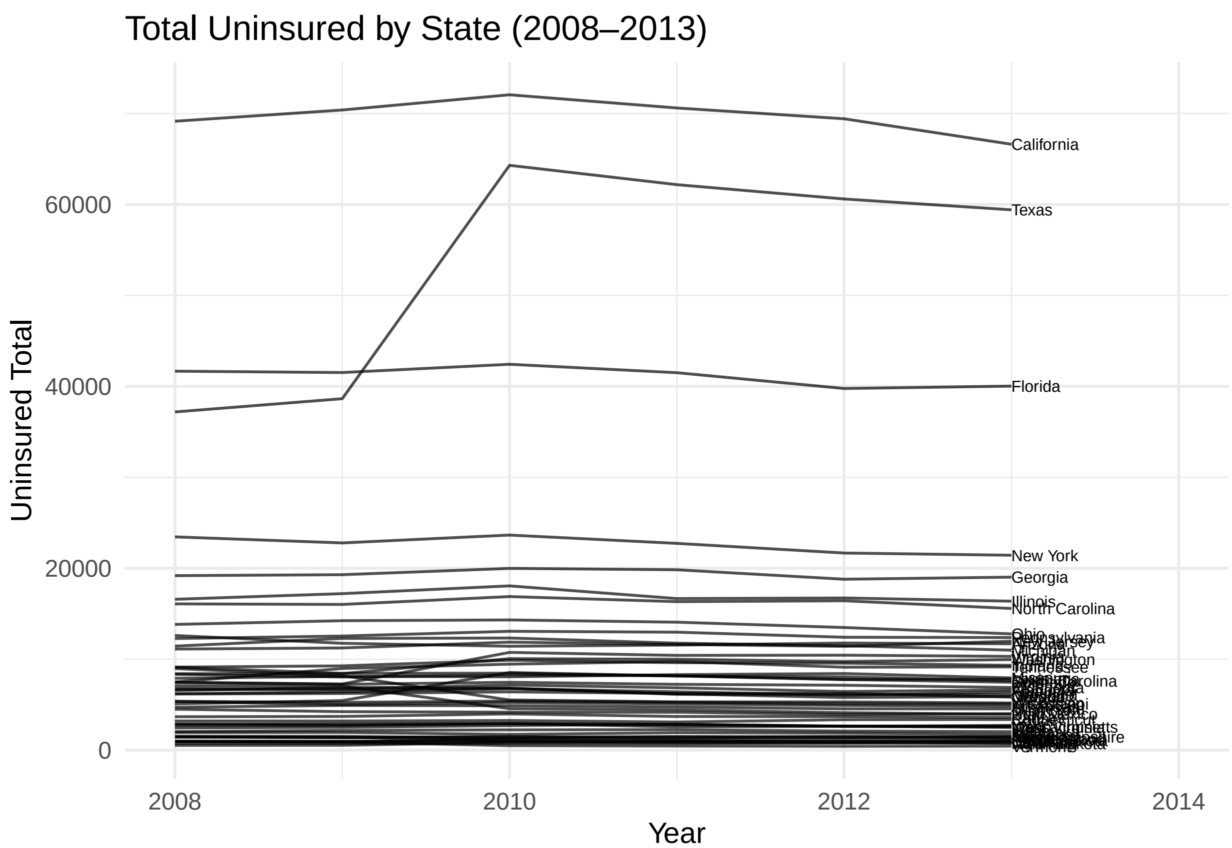

```r
#messy but effective
ggplot(medicaid_expansion_to_2014, aes(x = year, y = uninsured_rate, group = State)) +
  geom_line(alpha = 0.7) +  # Lines, slightly transparent
  geom_text(data = medicaid_expansion_to_2014 %>% filter(year == 2013),
            aes(label = State),
            hjust = 0, vjust = 0.5, size = 2) +  # Labels at the end of lines
  labs(title = "Uninsured Rate by State (2008-2014)",
       x = "Year",
       y = "Uninsured Rate (%)") +
  theme_minimal() +
  theme(legend.position = "none") +
  xlim(2008, 2014)
```



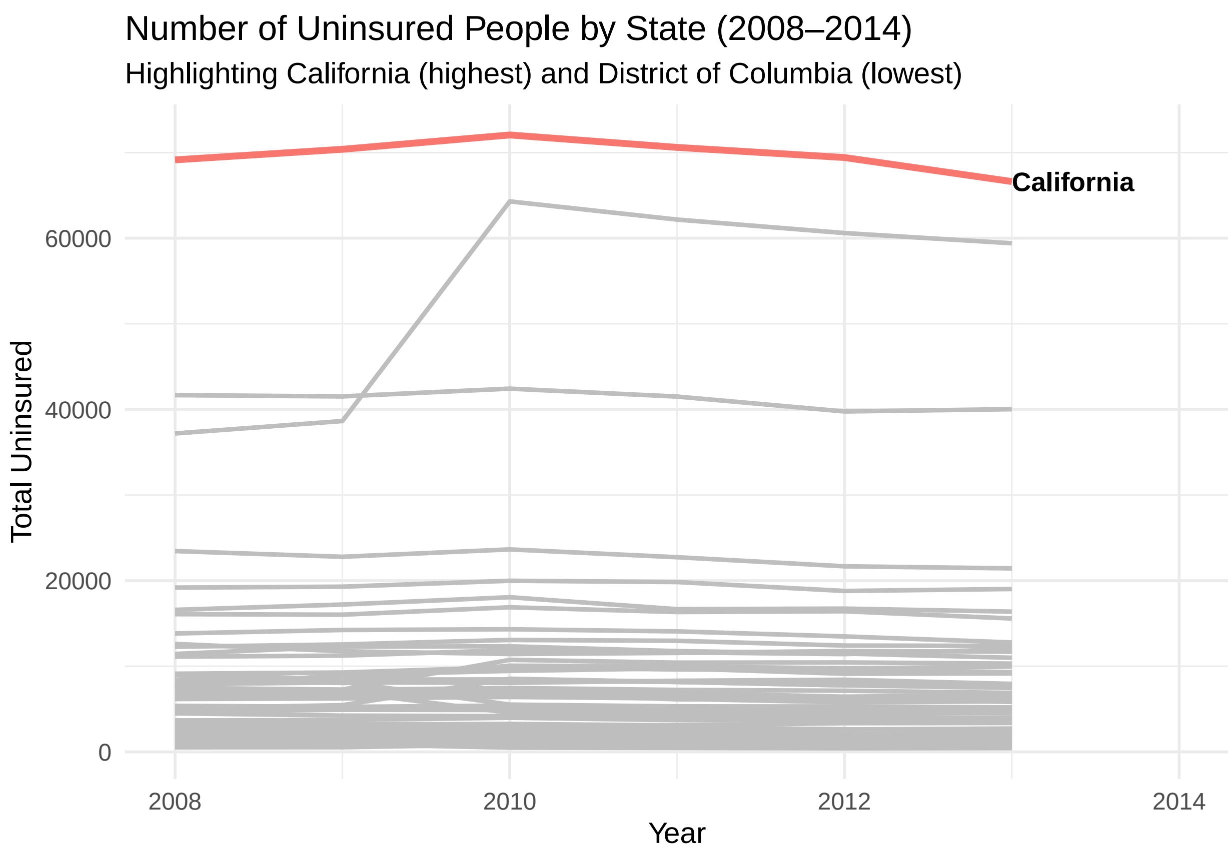

```r
##trying something else -- highest by average across 2008-2014
state_summary <- medicaid_expansion_to_2014 %>%
  group_by(State) %>%
  summarize(avg_uninsured_rate = mean(uninsured_rate, na.rm = TRUE)) %>%
  arrange(desc(avg_uninsured_rate))

highest_state <- state_summary$State[1]
lowest_state <- state_summary$State[nrow(state_summary)]

#same viz, just highlighting those ones
ggplot(medicaid_expansion_to_2014, aes(x = year, y = uninsured_rate, group = State)) +
  geom_line(aes(color = State %in% c(highest_state, lowest_state)), linewidth = 1) +
  scale_color_manual(values = c("gray", "red")) + # Highlight selected states in red
  geom_text(data = medicaid_expansion_to_2014 %>%
              filter(State %in% c(highest_state, lowest_state), year == 2013),
            aes(label = State),
            hjust = -0.1, size = 3) +
  labs(title = "Uninsured Rate by State (2008-2013)",
       subtitle = paste("Highlighting:", highest_state, "(highest) and", lowest_state, "(lowest)"),
       x = "Year",
       y = "Uninsured Rate (%)",
       color = "Highlighted State") +
  theme_minimal() +
  theme(legend.position = "none") # Remove legend if you want it cleaner
```



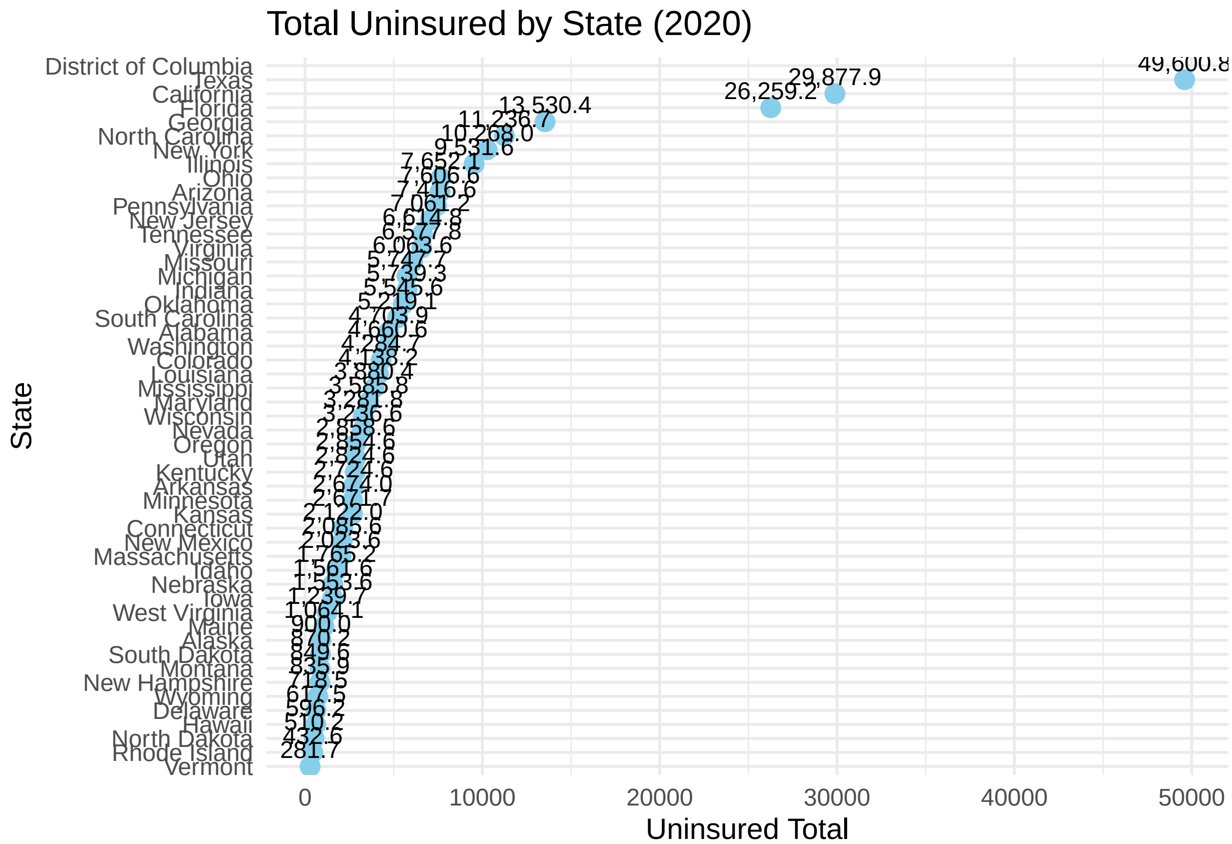

```r
# most uninsured Americans prior to 2014
medicaid_expansion_uninsured <- medicaid_expansion %>%
  mutate(uninsured_total = (uninsured_rate / 100) * population) %>%
  filter(year >= 2008, year < 2014)

ggplot(medicaid_expansion_uninsured, aes(x = year, y = uninsured_total, group = State)) +
  geom_line(alpha = 0.7) +  # Lines, slightly transparent
  geom_text(data = medicaid_expansion_uninsured %>% filter(year == 2013),
            aes(label = State),
            hjust = 0, vjust = 0.5, size = 2) +  # Labels at the end of lines
  labs(title = "Total Uninsured by State (2008-2013)",
       x = "Year",
       y = "Uninsured Total") +
  theme_minimal() +
  theme(legend.position = "none") +
  xlim(2008, 2014)
```




```r
#Average
state_summary_uninsured <- medicaid_expansion_uninsured %>%
  group_by(State) %>%
  summarize(avg_uninsured_total = mean(uninsured_total, na.rm = TRUE)) %>%
  arrange(desc(avg_uninsured_total))

highest_state_uninsured <- state_summary_uninsured$State[1]
lowest_state_uninsured <- state_summary_uninsured$State[nrow(state_summary_uninsured)]

ggplot(medicaid_expansion_uninsured, aes(x = year, y = uninsured_total, group = State)) +
  geom_line(color = "gray", size = 0.8) +  # All lines gray
  geom_line(data = medicaid_expansion_uninsured %>% filter(State %in% c(highest_state_uninsured, lowest_
            aes(color = State), size = 1.2) +  # Highlight important states
  geom_text(data = medicaid_expansion_uninsured %>%
              filter(State %in% c(highest_state_uninsured, lowest_state_uninsured), year == 2013),
            aes(label = State),
            hjust = 0, vjust = 0.5, size = 3.5, fontface = "bold") +
  labs(title = "Number of Uninsured People by State (2008-2014)",
       subtitle = paste("Highlighting", highest_state_uninsured, "(highest) and", lowest_state_uninsure
       x = "Year",
       y = "Total Uninsured",
       color = "State") +
  theme_minimal() +
  theme(legend.position = "none") +
  xlim(2008, 2014)
```




```r
# most uninsured Americans in 2020. DC is empty, it isn't really the most

medicaid_expansion_uninsured_2020 <- medicaid_expansion %>%
  mutate(uninsured_total = (uninsured_rate / 100) * population) %>%
  filter(year == 2020)

ggplot(medicaid_expansion_uninsured_2020, aes(x = reorder(State, uninsured_total), y = uninsured_total))
  geom_point(size = 3, color = "skyblue") +  # Dot plot
  geom_text(aes(label = scales::comma(uninsured_total)), vjust = -0.5, size = 3) +  # Labels on dots
  labs(title = "Total Uninsured by State (2020)",
       x = "State",
       y = "Uninsured Total") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_flip()  # Flip to make it horizontal for better readability
```



Difference-in-Differences Estimation

Estimate Model

Do the following:

- Choose a state that adopted the Medicaid expansion on January 1, 2014 and a state that did not. **Hint**: Do not pick Massachusetts as it passed a universal healthcare law in 2006, and also avoid picking a state that adopted the Medicaid expansion between 2014 and 2015.

```r
#set of states that expanded on the date of interest
expanded_on_dof <- medicaid_expansion %>%
  filter(Date_Adopted == as.Date("2014-01-01")) %>%
  distinct(State)

#Somehow took forever just to figure out how to see them all at once
# print(expanded_on_dof)
# print(expanded_on_dof, n = 30)
# view(expanded_on_dof)
print(expanded_on_dof$State)
```

```
##  [1] "Arizona"              "Arkansas"             "California"
##  [4] "Colorado"             "Connecticut"          "Delaware"
```

```
##  [7] "District of Columbia" "Hawaii"               "Illinois"
## [10] "Iowa"                 "Kentucky"             "Maryland"
## [13] "Massachusetts"        "Minnesota"            "Nevada"
## [16] "New Jersey"           "New Mexico"           "New York"
## [19] "North Dakota"         "Ohio"                 "Oregon"
## [22] "Rhode Island"         "Vermont"              "Washington"
## [25] "West Virginia"
```

```r
# set of states that didn't expand
not_expanded <- medicaid_expansion %>%
  filter(is.na(Date_Adopted)) %>%
  distinct(State)

print(not_expanded$State)
```

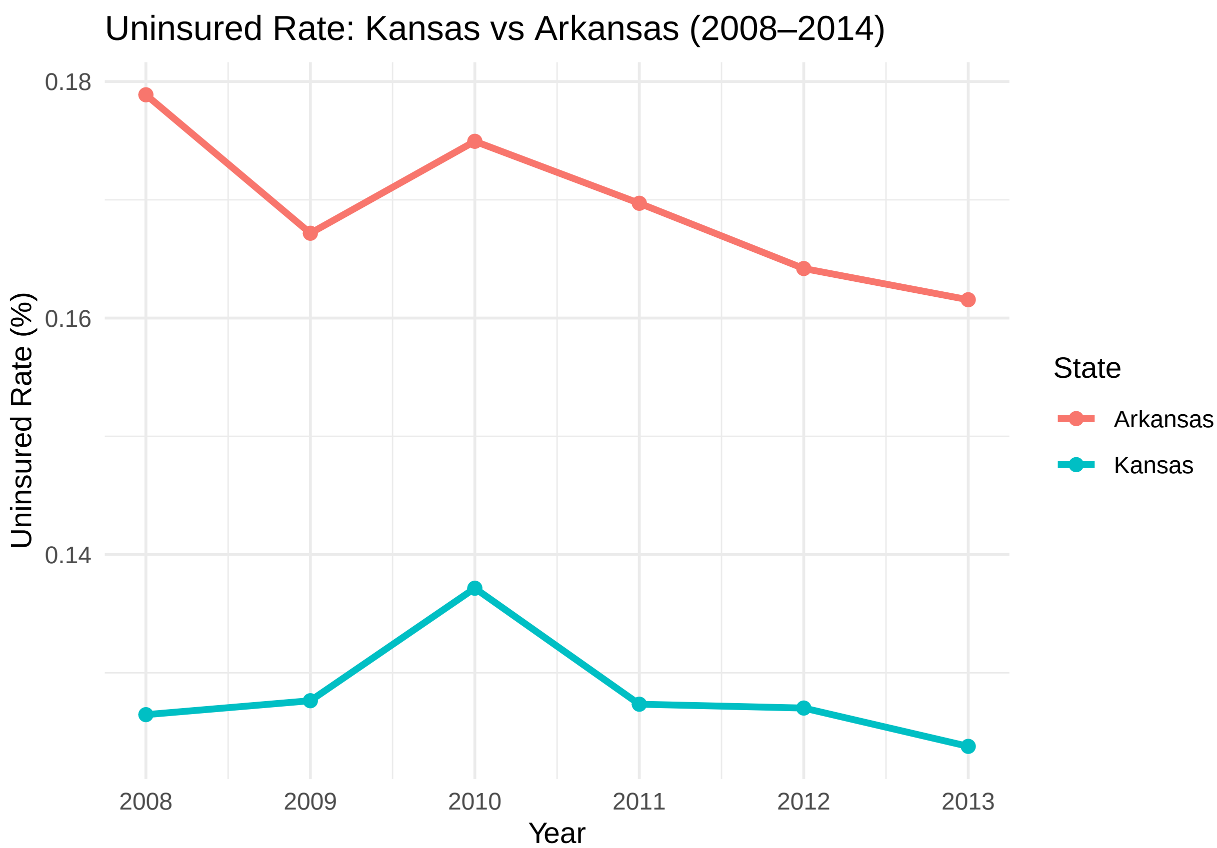
```
##  [1] "Alabama"        "Florida"        "Georgia"        "Kansas"
##  [5] "Maine"          "Mississippi"    "Missouri"       "North Carolina"
##  [9] "Oklahoma"       "South Carolina" "South Dakota"   "Tennessee"
## [13] "Texas"          "Wisconsin"      "Wyoming"
```

I'm going to start off using Kansas (didn't expand) and Arkansas (expanded on 1/1/2014), see how they look.

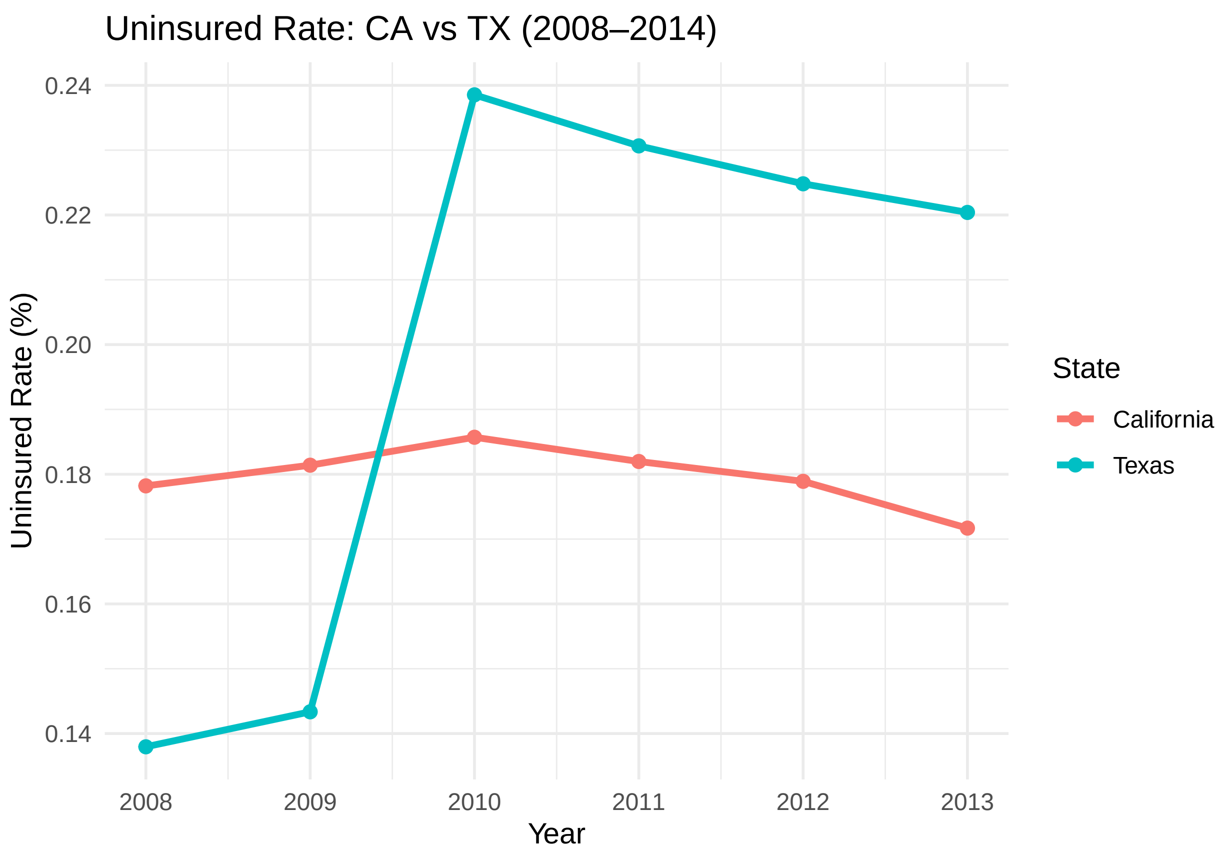
- Assess the parallel trends assumption for your choices using a plot. If you are not satisfied that the assumption has been met, pick another state and try again (but detail the states you tried).

Kansas and Arkansas are looking pretty good, especially between 2012 and 2013. Going to try a few more combos for fun

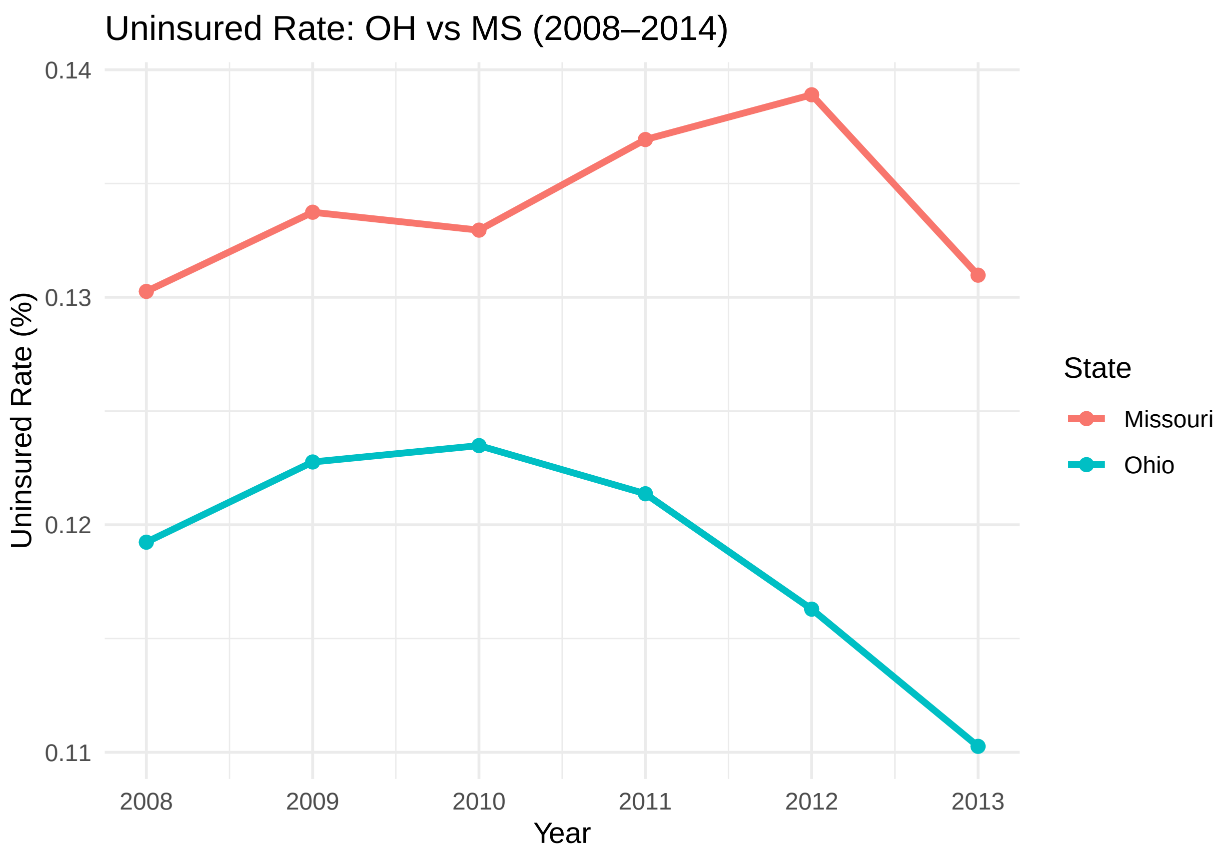
```r
# Parallel Trends plot: Arkansas and Kansas -- pretty good!
# Just Kansas and Arkansas
states_to_compare <- medicaid_expansion %>%
  filter(State %in% c("Kansas", "Arkansas"),
         year >= 2008, year < 2014)  # Pre-treatment years

ggplot(states_to_compare, aes(x = year, y = uninsured_rate, color = State)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Uninsured Rate: Kansas vs Arkansas (2008-2014)",
       x = "Year",
       y = "Uninsured Rate (%)",
       color = "State") +
  theme_minimal()
```




```r
# Parallel Trends plot: California and Texas: definitely not, terrible choices
states_to_compare <- medicaid_expansion %>%
  filter(State %in% c("Texas", "California"),
         year >= 2008, year < 2014)  # Pre-treatment years

ggplot(states_to_compare, aes(x = year, y = uninsured_rate, color = State)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Uninsured Rate: CA vs TX (2008-2014)",
       x = "Year",
       y = "Uninsured Rate (%)",
       color = "State") +
  theme_minimal()
```




```r
# Parallel Trends plot: Ohio and Missouri: also fairly poor trend match
states_to_compare <- medicaid_expansion %>%
  filter(State %in% c("Missouri", "Ohio"),
         year >= 2008, year < 2014)  # Pre-treatment years

ggplot(states_to_compare, aes(x = year, y = uninsured_rate, color = State)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Uninsured Rate: OH vs MS (2008-2014)",
       x = "Year",
       y = "Uninsured Rate (%)",
       color = "State") +
  theme_minimal()
```



- Estimates a difference-in-differences estimate of the effect of the Medicaid expansion on the uninsured share of the population. You may follow the lab example where we estimate the differences in one pre-treatment and one post-treatment period, or take an average of the pre-treatment and post-treatment outcomes

```
# Difference-in-Differences estimation

# DiD for: Kansas-Arkansas. trying the point in time/subtraction thing
# ----------------------------------------------------------------
# create a dataset for Kansas and Arkansas
states_of_int <-
  medicaid_expansion %>%
  filter(State %in% c("Kansas","Arkansas")) %>%
  filter(between(year, 2012, 2015)) #I don't think I need this actually? Also -- using "between", is th
  glimpse(states_of_int)
```

```
## Rows: 8
## Columns: 5
## $ State          <chr> "Arkansas", "Kansas", "Arkansas", "Kansas", "Arkansas",~
## $ Date_Adopted   <date> 2014-01-01, NA, 2014-01-01, NA, 2014-01-01, NA, 2014-0~
## $ year           <dbl> 2012, 2012, 2013, 2013, 2014, 2014, 2015, 2015
## $ uninsured_rate <dbl> 0.16419, 0.12702, 0.16155, 0.12378, 0.11779, 0.10649, 0~
## $ population     <dbl> 2994079, 2904021, 2994079, 2904021, 2994079, 2904021, 2~
```

```r
# pre-treatment difference
# ----------
pre_diff <-
  states_of_int %>%
  # filter out only the quarter we want
  filter(year == 2012) %>%
  # subset to select only vars we want
  select(State,
         uninsured_rate) %>%
  # make the data wide
  pivot_wider(names_from = State,
              values_from = uninsured_rate) %>%
  # subtract to make calculation
  summarise(Kansas - Arkansas)
  glimpse(pre_diff)
```

```
## Rows: 1
## Columns: 1
## $ `Kansas - Arkansas` <dbl> -0.03717
```

```r
# post-treatment difference
# ----------
post_diff <-
  states_of_int %>%
  # filter out only the quarter we want
  filter(year == 2015) %>%
  # subset to select only vars we want
  select(State,
         uninsured_rate) %>%
  # make the data wide
  pivot_wider(names_from = State,
              values_from = uninsured_rate) %>%
  # subtract to make calculation
  summarise(Kansas - Arkansas)
  glimpse(post_diff)
```

```
## Rows: 1
## Columns: 1
## $ `Kansas - Arkansas` <dbl> -0.00147
```

```r
# diff-in-diffs
# ----------
diff_in_diffs <- post_diff - pre_diff
diff_in_diffs
```

```
##   Kansas - Arkansas
## 1            0.0357
```

Looks like if this is true, the expansion resulted in a 3.6% decrease in the uninsured population between 2012 and 2015.

1. Discussion Questions

- Card/Krueger's original piece utilized the fact that towns on either side of the Delaware river are likely to be quite similar to one another in terms of demographics, economics, etc. Why is that intuition harder to replicate with this data?

- **Answer**: Card and Krueger compared places in NJ and PA, on either side of the Delaware River; the small populations and close geographies meant that demographics, economies, and culture, etc, were very similar. This helps with the parallel trends assumption, because you can more confidently say that any observed differences were due to the policy change only (in this case minimum wage), and not other factors. We, on the other hand, are comparing entire states – larger, necessarily more diverse in terms of demographics, healthcare quality and availability, the economy, etc. Picking culturally similar states in the middle of the country helps but lots of things other than the expansion could actually be affecting the uninsured rates.

- What are the strengths and weaknesses of using the parallel trends assumption in difference-in-differences estimates?

- **Answer**: Strengths:

1. You can (attempt to) estimate causal effects without controlling for every possible confounder, since supposedly DiD controls for changes over time and automatically adjusts for differences between groups.
2. DiD is easy to implement and also to explain – simple and intuitive.

Weakness: I don't think there are actually any solid tests for the parallel trends assumption? At least in class we've only just assessed it visually. And that might feel like you're on kind of shaky ground because if you're off, and the groups in question were actually not on the same trajectories before the treatment, your conclusions are pretty much useless.

Synthetic Control

Estimate Synthetic Control

Although several states did not expand Medicaid on January 1, 2014, many did later on. In some cases, a Democratic governor was elected and pushed for a state budget that included the Medicaid expansion, whereas in others voters approved expansion via a ballot initiative. The 2018 election was a watershed moment where several Republican-leaning states elected Democratic governors and approved Medicaid expansion. In cases with a ballot initiative, the state legislature and governor still must implement the results via legislation. For instance, Idaho voters approved a Medicaid expansion in the 2018 election, but it was not implemented in the state budget until late 2019, with enrollment beginning in 2020.

Do the following:

- Choose a state that adopted the Medicaid expansion after January 1, 2014. Construct a non-augmented synthetic control and plot the results (both pre-treatment fit and post-treatment differences). Also report the average ATT and L2 imbalance.

```r
#looking at my options
late_expanders <- medicaid_expansion %>%
  filter(Date_Adopted > as.Date("2015-01-01")) %>%
  distinct(State, Date_Adopted)

print(late_expanders)
```

```
## # A tibble: 8 x 2
##   State     Date_Adopted
##   <chr>     <date>
## 1 Alaska    2015-09-01
## 2 Idaho     2020-01-01
## 3 Indiana   2015-02-01
## 4 Louisiana 2016-07-01
## 5 Montana   2016-01-01
## 6 Nebraska  2020-10-01
## 7 Utah      2020-01-01
## 8 Virginia  2019-01-01
```

```r
#want to pick a state with an average population, Louisiana is a good bet
medicaid_expansion_late <- medicaid_expansion %>%
  filter(State %in% late_expanders$State) %>%
  filter(year == 2016)

print(medicaid_expansion_late)
```

```
## # A tibble: 8 x 5
##   State     Date_Adopted  year uninsured_rate population
##   <chr>     <date>       <dbl>          <dbl>      <dbl>
## 1 Alaska    2015-09-01    2016         0.145      737732
## 2 Idaho     2020-01-01    2016         0.103     1634464
## 3 Indiana   2015-02-01    2016         0.0807    6596855
## 4 Louisiana 2016-07-01    2016         0.102     4649676
## 5 Montana   2016-01-01    2016         0.0836    1023579
## 6 Nebraska  2020-10-01    2016         0.0888    1881503
## 7 Utah      2020-01-01    2016         0.0851    2942902
## 8 Virginia  2019-01-01    2016         0.0884    8326289
```

-Construct a non-augmented synthetic control and plot the results (both pre-treatment fit and post-treatment differences). Also report the average ATT and L2 imbalance.

```r
# non-augmented synthetic control using Louisiana



#define a treatment var. Uisng 2017 as first true post-treatment year since the adoption was mid-2016
medicaid_expansion <- medicaid_expansion %>%
  mutate(treatment = if_else(State == "Louisiana" & year >= 2017, 1, 0))

# Run non-augmented synthetic control for Louisiana
syn_louisiana <- augsynth(
  uninsured_rate ~ treatment,   # outcome ~ treatment indicator
  unit = State,                 # unit of analysis
  time = year,                  # time variable
  t_int = 2017,
  data = medicaid_expansion,   # your dataset
  unit_name = "Louisiana",     # treated unit
  progfunc = "None",           # no augmentation (use "ridge" for augmented)
  scm = TRUE                   # use synthetic control method
)
```

```
## One outcome and one treatment time found. Running single_augsynth.
```

```r
# # View summary of results

# this is not working not sure why
# summary(syn_louisiana)
#
#
# att <- summary(syn_louisiana)$att.avg
# imbalance <- summary(syn_louisiana)$imbalance
#
# print(att)
# print(imbalance)

syn_summary <- summary(syn_louisiana)
glimpse(syn_summary)
```

```
## List of 9
##  $ att                :'data.frame': 13 obs. of  5 variables:
##   ..$ Time       : num [1:13] 2008 2009 2010 2011 2012 ...
##   ..$ Estimate   : num [1:13] 1.25e-03 -1.54e-03 3.35e-05 -2.21e-04 -1.23e-03 ...
##   ..$ lower_bound: num [1:13] NA NA NA NA NA ...
##   ..$ upper_bound: num [1:13] NA NA NA NA NA ...
##   ..$ p_val      : num [1:13] NA NA NA NA NA NA NA NA NA 0.118 ...
##  $ average_att        :'data.frame': 1 obs. of  6 variables:
##   ..$ Value      : chr "Average Post-Treatment Effect"
##   ..$ Estimate   : num -0.0257
##   ..$ Std.Error  : logi NA
##   ..$ p_val      : num 0.312
##   ..$ lower_bound: num NA
##   ..$ upper_bound: num NA
##  $ alpha              : num 0.05
##  $ t_int              : num 2017
##  $ call               : language single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(ti
##  $ l2_imbalance       : num 0.00515
##  $ scaled_l2_imbalance: num 0.0483
##  $ bias_est           : logi NA
##  $ inf_type           : chr "conformal"
##  - attr(*, "class")= chr "summary.augsynth"
```

```r
att <- syn_summary$average_att$Estimate
pval <- syn_summary$average_att$p_val
imbalance <- syn_summary$l2_imbalance
improvement <- 1 - (syn_summary$scaled_l2_imbalance / syn_summary$l2_imbalance)

print(att)
```

```
## [1] -0.02567633
```

```r
print(pval)
```

```
## [1] 0.312
```

```r
print(imbalance)
```

```
## [1] 0.005154465
```

```r
print(improvement)
```

```
## [1] -8.370567
```

ATT of -0.02567633 shows that Louisiana's uninsured rate dropped by 2.57 more percentage points on average than the synthetic control post-treatment.

the p-value of 0.284 isn't great, a p value of under 0.05 is a normal level of significance so this is pretty large.

Imbalance: 0.0052, low imbalance, I think this just means that the pre-treatment fit is good.

Improvement: The internet says to test for this too. It's -8.37%. I don't totally understand this to be honest, but I think negative is not good, because it means I could have just averaged all the other states instead of doing all this synthetic control modeling and it would have been better.

```r
str(syn_louisiana) # what even is the data now
```

```
## List of 17
##  $ weights            : num [1:50, 1] -1.26e-09 3.61e-02 1.26e-09 -4.31e-10 1.85e-09 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##   .. ..$ : NULL
##  $ l2_imbalance       : num 0.00515
##  $ scaled_l2_imbalance: num 0.0483
##  $ mhat               : num [1:51, 1:13] 0 0 0 0 0 0 0 0 0 0 ...
##  $ lambda             : NULL
##  $ ridge_mhat         : num [1:51, 1:4] 0 0 0 0 0 0 0 0 0 0 ...
##  $ synw               : num [1:50] -1.26e-09 3.61e-02 1.26e-09 -4.31e-10 1.85e-09 ...
##  $ lambdas            : NULL
##  $ lambda_errors      : NULL
##  $ lambda_errors_se   : NULL
##  $ data               :List of 5
##   ..$ X         : num [1:51, 1:9] 0.14 0.208 0.187 0.179 0.178 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   ..$ trt       : num [1:51] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ y         : num [1:51, 1:4] 0.0954 0.139 0.1003 0.0801 0.0723 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:4] "2017" "2018" "2019" "2020"
##   ..$ synth_data:List of 6
##   .. ..$ Z0    : num [1:9, 1:50] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ Z1    : num [1:9, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
```

```
##   .. .. .. ..$ : NULL
##   .. ..$ Y0plot: num [1:13, 1:50] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:13] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ Y1plot: num [1:13, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:13] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ X0    : num [1:9, 1:50] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ X1    : num [1:9, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   ..$ time      : num [1:13] 2008 2009 2010 2011 2012 ...
##  $ progfunc           : chr "none"
##  $ scm                : logi TRUE
##  $ fixedeff           : logi FALSE
##  $ extra_args         :List of 1
##   ..$ unit_name: chr "Louisiana"
##  $ call               : language single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(t
##  $ t_int              : num 2017
##  - attr(*, "class")= chr "augsynth"
```

```r
head(syn_louisiana)
```

```
## $weights
##                               [,1]
## Alabama              -1.261392e-09
## Alaska                3.606503e-02
## Arizona               1.258457e-09
## Arkansas             -4.309632e-10
## California            1.851176e-09
## Colorado             -2.158358e-09
## Connecticut           2.746357e-09
## Delaware             -1.171908e-09
## District of Columbia -1.521726e-09
## Florida              -5.454430e-10
## Georgia              -7.469676e-10
## Hawaii                1.868655e-10
## Idaho                -1.241576e-09
## Illinois              1.286835e-09
## Indiana               1.317462e-01
## Iowa                  7.155797e-10
## Kansas                3.701347e-10
## Kentucky              1.021510e-09
## Maine                 9.976434e-02
## Maryland              3.712496e-10
## Massachusetts         2.641933e-10
## Michigan              1.854103e-09
## Minnesota             7.373793e-10
```

```
## Mississippi          -1.322128e-09
## Missouri              2.413770e-09
## Montana               4.101717e-09
## Nebraska             -1.674727e-09
## Nevada                3.303568e-10
## New Hampshire         2.419574e-09
## New Jersey            1.498277e-09
## New Mexico            2.969921e-01
## New York              2.335643e-10
## North Carolina        4.763013e-10
## North Dakota          2.681588e-09
## Ohio                  1.368876e-09
## Oklahoma             -1.106363e-09
## Oregon               -6.236481e-10
## Pennsylvania          8.975819e-10
## Rhode Island          2.590328e-09
## South Carolina        1.236680e-09
## South Dakota          8.836577e-02
## Tennessee             2.459158e-09
## Texas                 2.055323e-01
## Utah                  1.415343e-01
## Vermont              -5.320474e-10
## Virginia              7.283955e-10
## Washington            1.887524e-09
## West Virginia         1.525876e-09
## Wisconsin            -4.269493e-10
## Wyoming               1.781470e-09
##
## $l2_imbalance
## [1] 0.005154465
##
## $scaled_l2_imbalance
## [1] 0.04830026
##
## $mhat
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [3,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0     0
```

```
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [22,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [23,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [24,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [25,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [26,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [27,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [28,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [29,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [30,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [31,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [32,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [33,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [34,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [35,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [36,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [37,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [38,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [39,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [40,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [41,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [42,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [43,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [44,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [45,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [46,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [47,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [48,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [49,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [50,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [51,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##
## $lambda
## NULL
##
## $ridge_mhat
##       [,1] [,2] [,3] [,4]
##  [1,]    0    0    0    0
##  [2,]    0    0    0    0
##  [3,]    0    0    0    0
##  [4,]    0    0    0    0
##  [5,]    0    0    0    0
##  [6,]    0    0    0    0
##  [7,]    0    0    0    0
##  [8,]    0    0    0    0
##  [9,]    0    0    0    0
## [10,]    0    0    0    0
## [11,]    0    0    0    0
## [12,]    0    0    0    0
## [13,]    0    0    0    0
## [14,]    0    0    0    0
## [15,]    0    0    0    0
```

```
## [16,]    0    0    0    0
## [17,]    0    0    0    0
## [18,]    0    0    0    0
## [19,]    0    0    0    0
## [20,]    0    0    0    0
## [21,]    0    0    0    0
## [22,]    0    0    0    0
## [23,]    0    0    0    0
## [24,]    0    0    0    0
## [25,]    0    0    0    0
## [26,]    0    0    0    0
## [27,]    0    0    0    0
## [28,]    0    0    0    0
## [29,]    0    0    0    0
## [30,]    0    0    0    0
## [31,]    0    0    0    0
## [32,]    0    0    0    0
## [33,]    0    0    0    0
## [34,]    0    0    0    0
## [35,]    0    0    0    0
## [36,]    0    0    0    0
## [37,]    0    0    0    0
## [38,]    0    0    0    0
## [39,]    0    0    0    0
## [40,]    0    0    0    0
## [41,]    0    0    0    0
## [42,]    0    0    0    0
## [43,]    0    0    0    0
## [44,]    0    0    0    0
## [45,]    0    0    0    0
## [46,]    0    0    0    0
## [47,]    0    0    0    0
## [48,]    0    0    0    0
## [49,]    0    0    0    0
## [50,]    0    0    0    0
## [51,]    0    0    0    0
```

```r
# Extract time and outcome values
#time_vals <- as.integer(colnames(syn_louisiana$data$X))  # Tried to get the years from column names
#needed to hard code it bc it was too hard to extract the years
time_vals <- 2008:2020
#time_vals <- syn_louisiana$data$synth_data$Y0plot[,1]  # Assuming time is the same for all units?
actual_vals <- syn_louisiana$data$synth_data$Y1plot[,1]  # treated LA outcome
synthetic_vals <- syn_louisiana$data$synth_data$Y0plot[,1]  # synthetic control outcome


# Check the extracted values
head(time_vals)
```

```
## [1] 2008 2009 2010 2011 2012 2013
```

```r
head(actual_vals)
```

```
##      2008      2009      2010      2011      2012      2013
```

```
## 0.179265 0.173049 0.177919 0.175711 0.166900 0.168220
```

```r
print(synthetic_vals)
```

```
##      2008      2009      2010      2011      2012      2013      2014      2015
## 0.1397160 0.1388650 0.1476530 0.1413780 0.1329400 0.1366100 0.1198200 0.1015400
##      2016      2017      2018      2019      2020
## 0.0920043 0.0954480 0.1017140 0.0965850 0.0970000
```

```r
# Create a data frame with columns for Year, Actual, and Synthetic
outcome_df <- data.frame(
  Year = time_vals,
  Actual = actual_vals,
  Synthetic = synthetic_vals
)


#is this working
head(outcome_df)
```

```
##      Year   Actual Synthetic
## 2008 2008 0.179265  0.139716
## 2009 2009 0.173049  0.138865
## 2010 2010 0.177919  0.147653
## 2011 2011 0.175711  0.141378
## 2012 2012 0.166900  0.132940
## 2013 2013 0.168220  0.136610
```

```r
summary(outcome_df)
```

```
##       Year          Actual           Synthetic
##  Min.   :2008   Min.   :0.07937   Min.   :0.0920
##  1st Qu.:2011   1st Qu.:0.08900   1st Qu.:0.0970
##  Median :2014   Median :0.14522   Median :0.1198
##  Mean   :2014   Mean   :0.13455   Mean   :0.1186
##  3rd Qu.:2017   3rd Qu.:0.17305   3rd Qu.:0.1389
##  Max.   :2020   Max.   :0.17927   Max.   :0.1477
```

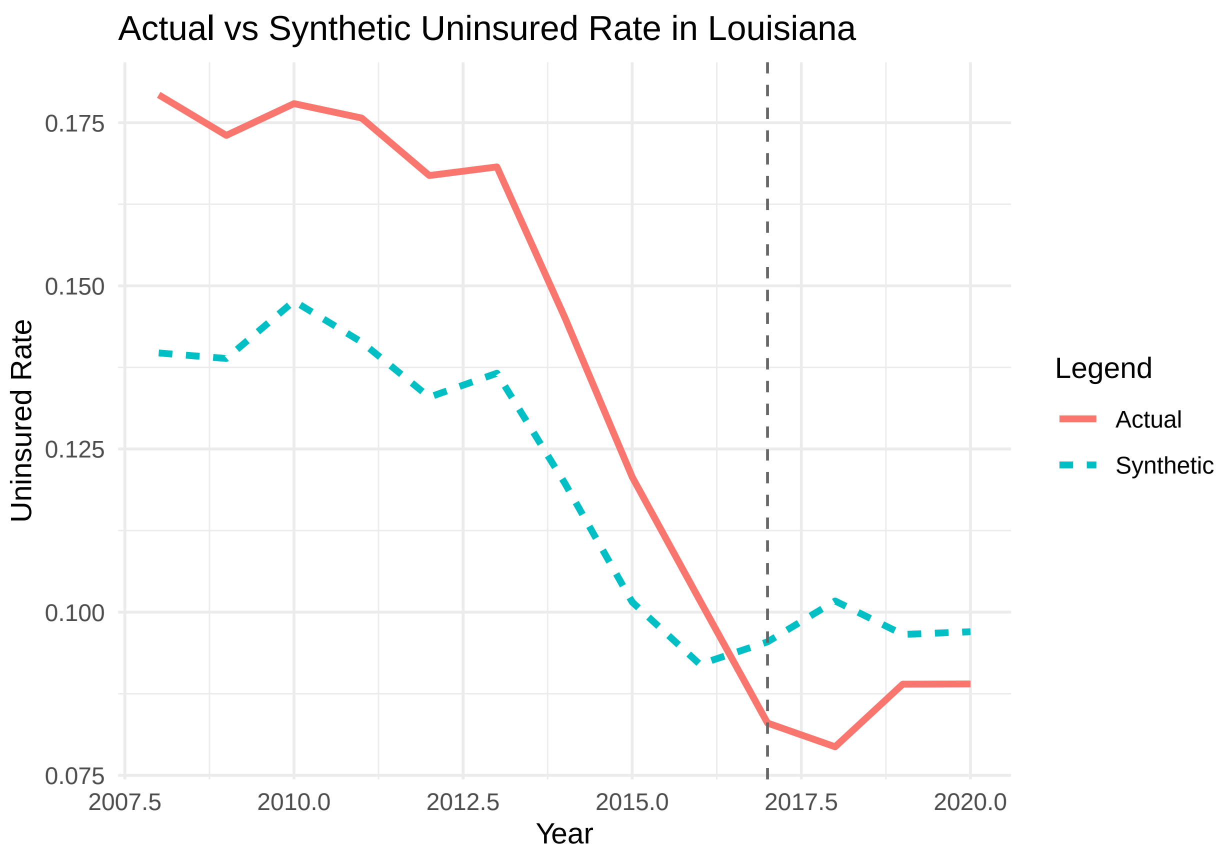
```r
#plots

#pre-treatment plot


# Convert to long format for ggplot
# Make sure the reshaping process is working correctly
outcome_long <- pivot_longer(outcome_df,
                             cols = c("Actual", "Synthetic"),
                             names_to = "Type",
                             values_to = "UninsuredRate")

# Check the reshaped data frame
head(outcome_long)
```

```
## # A tibble: 6 x 3
##    Year Type      UninsuredRate
##   <int> <chr>             <dbl>
## 1  2008 Actual            0.179
## 2  2008 Synthetic         0.140
## 3  2009 Actual            0.173
## 4  2009 Synthetic         0.139
## 5  2010 Actual            0.178
## 6  2010 Synthetic         0.148
```

```r
# Check
head(outcome_long)
```

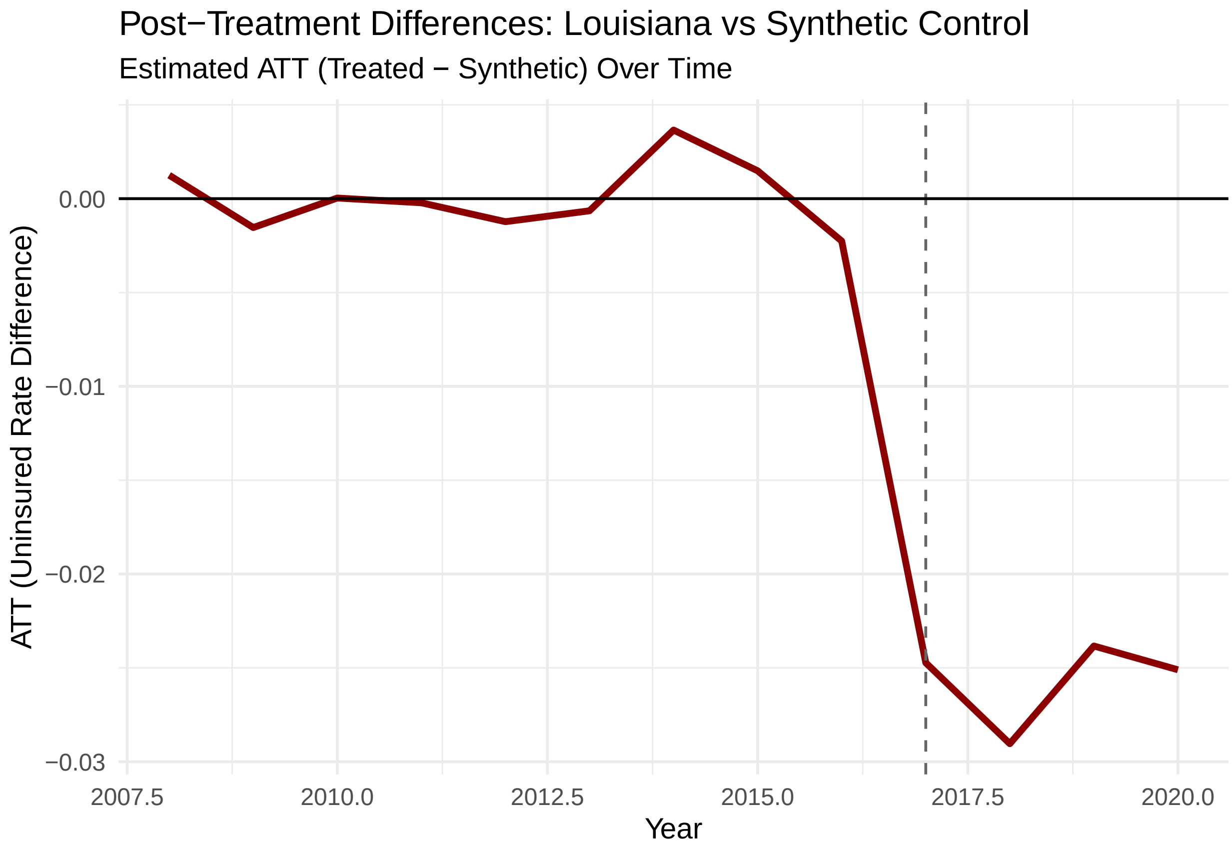
```
## # A tibble: 6 x 3
##    Year Type      UninsuredRate
##   <int> <chr>             <dbl>
## 1  2008 Actual            0.179
## 2  2008 Synthetic         0.140
## 3  2009 Actual            0.173
## 4  2009 Synthetic         0.139
## 5  2010 Actual            0.178
## 6  2010 Synthetic         0.148
```

```r
# Create the plot
#pre-treatment

ggplot(outcome_long, aes(x = Year, y = UninsuredRate, color = Type, linetype = Type)) +
  geom_line(size = 1.2) +
  geom_vline(xintercept = syn_louisiana$t_int, linetype = "dashed", color = "gray40") +  # Treatment ti
  labs(
    title = "Actual vs Synthetic Uninsured Rate in Louisiana",
    x = "Year",
    y = "Uninsured Rate",
    color = "Legend",
    linetype = "Legend"
  ) +
  theme_minimal()
```



Ok so this plot should be showing a line for actual treated unit outcome (red) (e.g., LA's uninsured rate) and one for the synthetic control estimate (blue, dotted). They are pretty close right before treatment and then after treatment they diverge, more specifically, the "actual" line dips, implying that maybe the treatment works to reduce uninsured rates.

```r
#post-treatment
library(ggplot2)

# Extract ATT time-series
att_df <- summary(syn_louisiana)$att

ggplot(att_df, aes(x = Time, y = Estimate)) +
  geom_line(color = "darkred", size = 1.2) +
  geom_vline(xintercept = summary(syn_louisiana)$t_int, linetype = "dashed", color = "gray40") +
  geom_hline(yintercept = 0, linetype = "solid", color = "black") +
  labs(
    title = "Post-Treatment Differences: Louisiana vs Synthetic Control",
    subtitle = "Estimated ATT (Treated - Synthetic) Over Time",
    x = "Year",
    y = "ATT (Uninsured Rate Difference)"
  ) +
  theme_minimal()
```



This plot is supposed to show the difference between the actual and synthetic outcomes. Makes sense that the difference should be about zero until the point of treatment, which we've classified at 2017.

- Re-run the same analysis but this time use an augmentation (default choices are Ridge, Matrix Completion, and GSynth). Create the same plot and report the average ATT and L2 imbalance.

```
summary(medicaid_expansion$uninsured_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02495 0.07702 0.10475 0.10978 0.13888 0.24082
```

```
summary(medicaid_expansion$treatment)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.006033 0.000000 1.000000
```

```
summary(medicaid_expansion$population)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##   584153  1850326  4531566  6364343  7061530 38802500       13
```

```
medicaid_expansion <- medicaid_expansion %>%
  filter(!is.na(uninsured_rate), !is.na(treatment), !is.na(population))
```

```r
# augmented synthetic control

#define a treatment var. Uisng 2017 as first true post-treatment year since the adoption was mid-2016
medicaid_expansion <- medicaid_expansion %>%
  mutate(treatment = if_else(State == "Louisiana" & year >= 2017, 1, 0))

# Run augmented synthetic control for Louisiana
# syn_louisiana <- augsynth(
#   uninsured_rate ~ treatment,    # outcome ~ treatment indicator
#   unit = State,                  # unit of analysis
#   time = year,                   # time variable
#   t_int = 2017,
#   data = medicaid_expansion,    # your dataset
#   unit_name = "Louisiana",      # treated unit
#   progfunc = "MCP",            # using matrix completion bc the internet says ridge won't do anything
#   scm = TRUE                    # use synthetic control method
# )


#Ok I tried running the ridge a bunch of ways and it's still identical to non-augmented. going to try "
syn_louisiana <- augsynth(
  form = uninsured_rate ~ treatment | population,  # outcome ~ treatment + covariate
  unit = State,                                    # unit identifier
  time = year,                                     # time variable
  t_int = 2017,                                    # first treated period
  data = medicaid_expansion,                       # data
  unit_name = "Louisiana",                         # treated unit
  progfunc = "ridge",                              # ridge regression for augmentation
  scm = TRUE                                       # use synthetic control method
)
```

```
## One outcome and one treatment time found. Running single_augsynth.
```

```r
# # View summary of results

# this is not working not sure why
# summary(syn_louisiana)
#
#
# att <- summary(syn_louisiana)$att.avg
# imbalance <- summary(syn_louisiana)$imbalance
#
# print(att)
# print(imbalance)

syn_summary <- summary(syn_louisiana)
glimpse(syn_summary)
```

```
## List of 11
##  $ att                          :'data.frame':   13 obs. of  5 variables:
##   ..$ Time       : num [1:13] 2008 2009 2010 2011 2012 ...
##   ..$ Estimate   : num [1:13] 0.001056 -0.001529 0.000881 0.000014 -0.001865 ...
##   ..$ lower_bound: num [1:13] NA NA NA NA NA ...
```

```
##    ..$ upper_bound: num [1:13] NA NA NA NA NA ...
##    ..$ p_val      : num [1:13] NA NA NA NA NA NA NA NA NA 0.103 ...
##  $ average_att                  :'data.frame':   1 obs. of  6 variables:
##    ..$ Value      : chr "Average Post-Treatment Effect"
##    ..$ Estimate   : num -0.0239
##    ..$ Std.Error  : logi NA
##    ..$ p_val      : num 0.307
##    ..$ lower_bound: num NA
##    ..$ upper_bound: num NA
##  $ alpha                        : num 0.05
##  $ t_int                        : num 2017
##  $ call                         : language single_augsynth(form = form, unit = !!enquo(unit), time =
##  $ l2_imbalance                 : num 0.00561
##  $ scaled_l2_imbalance          : num 0.0544
##  $ covariate_l2_imbalance       : num 0.000333
##  $ scaled_covariate_l2_imbalance: num 0.0357
##  $ bias_est                     : num [1:4, 1] -372 -537 -547 -481
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "2017" "2018" "2019" "2020"
##   .. ..$ : NULL
##  $ inf_type                     : chr "conformal"
##  - attr(*, "class")= chr "summary.augsynth"
```

```r
att <- syn_summary$average_att$Estimate
pval <- syn_summary$average_att$p_val
imbalance <- syn_summary$l2_imbalance
improvement <- 1 - (syn_summary$scaled_l2_imbalance / syn_summary$l2_imbalance)

print(att)
```

```
## [1] -0.02386252
```

```r
print(pval)
```

```
## [1] 0.307
```

```r
print(imbalance)
```

```
## [1] 0.005607674
```

```r
print(improvement)
```

```
## [1] -8.700375
```

```r
str(syn_louisiana) # ridge data
```

```
## List of 19
##  $ weights                      : num [1:49, 1] 1.14e-01 1.22e-01 8.58e-06 -4.40e-06 -1.68e-04 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:49] "Alabama" "Alaska" "Arizona" "Arkansas" ...
```

```
##   .. ..$ : NULL
##  $ l2_imbalance               : num 0.00561
##  $ scaled_l2_imbalance        : num 0.0544
##  $ mhat                       : num [1:50, 1:13] 0 0 0 0 0 0 0 0 0 0 ...
##  $ lambda                     : num 0.564
##  $ ridge_mhat                 : num [1:50, 1:4] -8447 -30854 1810 -18558 176588 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "2017" "2018" "2019" "2020"
##  $ synw                       : num [1:49] 1.14e-01 1.22e-01 -2.10e-09 -6.34e-10 -5.13e-09 ...
##  $ lambdas                    : num [1:21] 0.5641 0.2246 0.0894 0.0356 0.0142 ...
##  $ lambda_errors              : num [1:21] 3.39e-05 3.39e-05 3.39e-05 3.40e-05 3.42e-05 ...
##  $ lambda_errors_se           : num [1:21] 9.85e-06 9.88e-06 9.94e-06 1.01e-05 1.02e-05 ...
##  $ covariate_l2_imbalance     : num 0.000333
##  $ scaled_covariate_l2_imbalance: num 0.0357
##  $ data                       :List of 6
##   ..$ X         : num [1:50, 1:9] 0.14 0.208 0.187 0.179 0.178 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   ..$ trt       : num [1:50] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ y         : num [1:50, 1:4] 0.0954 0.139 0.1003 0.0801 0.0723 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:4] "2017" "2018" "2019" "2020"
##   ..$ Z         : num [1:50, 1] 4849377 737732 6731484 2994079 38802500 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr "population"
##   ..$ synth_data:List of 6
##   .. ..$ Z0    : num [1:9, 1:49] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ Z1    : num [1:9, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ Y0plot: num [1:13, 1:49] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:13] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ Y1plot: num [1:13, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:13] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ X0    : num [1:9, 1:49] 0.14 0.139 0.148 0.141 0.133 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
##   .. ..$ X1    : num [1:9, 1] 0.179 0.173 0.178 0.176 0.167 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "2008" "2009" "2010" "2011" ...
##   .. .. .. ..$ : NULL
```

```
##   ..$ time      : num [1:13] 2008 2009 2010 2011 2012 ...
##  $ progfunc                    : chr "ridge"
##  $ scm                         : logi TRUE
##  $ fixedeff                    : logi FALSE
##  $ extra_args                  :List of 2
##   ..$ unit_name: chr "Louisiana"
##   ..$ lambda   : num 0.564
##  $ call                        : language single_augsynth(form = form, unit = !!enquo(unit), time =
##  $ t_int                       : num 2017
##  - attr(*, "class")= chr "augsynth"
```

```r
head(syn_louisiana)
```

```
## $weights
##                         [,1]
## Alabama         1.138666e-01
## Alaska          1.224505e-01
## Arizona         8.577500e-06
## Arkansas       -4.399032e-06
## California     -1.682583e-04
## Colorado       -2.855364e-05
## Connecticut    -1.181415e-05
## Delaware       -9.993479e-06
## Florida         5.923672e-06
## Georgia         4.779263e-05
## Hawaii         -2.645079e-05
## Idaho           2.048227e-02
## Illinois       -5.051311e-05
## Indiana         7.726559e-05
## Iowa           -5.208281e-05
## Kansas          5.295293e-05
## Kentucky       -1.213246e-04
## Maine           9.124955e-05
## Maryland       -2.167091e-05
## Massachusetts  -1.142316e-05
## Michigan       -6.404104e-05
## Minnesota      -6.805126e-05
## Mississippi     6.837538e-05
## Missouri        2.602074e-05
## Montana         5.970718e-05
## Nebraska       -4.210162e-05
## Nevada         -6.900935e-05
## New Hampshire   4.988209e-05
## New Jersey      3.906728e-05
## New Mexico      3.181449e-01
## New York       -2.829671e-05
## North Carolina  1.274303e-05
## North Dakota   -3.727399e-05
## Ohio           -5.543564e-05
## Oklahoma        4.237935e-05
## Oregon         -1.027181e-04
## Pennsylvania    1.775762e-05
## Rhode Island   -6.010410e-05
## South Carolina  1.274309e-01
```

```
## South Dakota    1.568010e-01
## Tennessee       1.596000e-05
## Texas           9.226681e-02
## Utah            4.919943e-02
## Vermont        -9.139877e-05
## Virginia        6.679980e-05
## Washington     -1.014398e-04
## West Virginia  -1.167453e-04
## Wisconsin      -1.475153e-05
## Wyoming         3.310635e-05
##
## $l2_imbalance
## [1] 0.005607674
##
## $scaled_l2_imbalance
## [1] 0.05439654
##
## $mhat
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [3,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [18,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [19,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [20,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [21,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [22,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [23,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [24,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [25,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [26,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [27,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [28,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [29,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [30,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [31,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [32,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [33,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [34,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [35,]    0    0    0    0    0    0    0    0    0     0     0     0     0
```

```
## [36,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [37,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [38,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [39,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [40,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [41,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [42,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [43,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [44,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [45,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [46,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [47,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [48,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [49,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [50,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##
## $lambda
## [1] 0.5640954
##
## $ridge_mhat
##              2017        2018        2019        2020
##  [1,]  -8446.8183 -12200.055 -12419.810 -10919.592
##  [2,] -30854.0481 -44563.661 -45366.370 -39886.457
##  [3,]   1810.1182   2614.416   2661.508   2340.020
##  [4,] -18557.6384 -26803.492 -27286.293 -23990.317
##  [5,] 176587.6750 255052.151 259646.309 228283.026
##  [6,]  -5686.6570  -8213.452  -8361.397  -7351.404
##  [7,] -15273.6808 -22060.339 -22457.703 -19744.987
##  [8,] -29775.6859 -43006.126 -43780.779 -38492.398
##  [9,]  73538.1202 106213.835 108127.025  95066.112
## [10,]  20153.0628  29107.810  29632.118  26052.788
## [11,] -27138.3269 -39196.886 -39902.924 -35082.960
## [12,] -25967.1361 -37505.306 -38180.875 -33568.917
## [13,]  35320.8588  51015.233  51934.152  45660.904
## [14,]   1076.4207   1554.713   1582.718   1391.538
## [15,] -17941.5911 -25913.700 -26380.474 -23193.917
## [16,] -19048.4376 -27512.367 -28007.937 -24624.793
## [17,] -10822.4578 -15631.278 -15912.838 -13990.689
## [18,]  -9535.1194 -13771.934 -14020.003 -12326.492
## [19,] -27625.9075 -39901.124 -40619.848 -35713.282
## [20,]  -2304.8506  -3328.977  -3388.940  -2979.584
## [21,]   1885.9568   2723.972   2773.038   2438.072
## [22,]  19131.4005  27632.198  28129.927  24732.046
## [23,]  -5134.5300  -7415.985  -7549.566  -6637.638
## [24,] -18607.5134 -26875.531 -27359.630 -24054.795
## [25,]  -1829.7235  -2642.738  -2690.341  -2365.368
## [26,] -29296.2763 -42313.709 -43075.890 -37872.651
## [27,] -24620.8629 -35560.826 -36201.370 -31828.523
## [28,] -19402.2177 -28023.356 -28528.131 -25082.148
## [29,] -27643.7624 -39926.911 -40646.100 -35736.364
## [30,]  13835.9182  19983.733  20343.692  17886.330
## [31,] -23508.7236 -33954.533 -34566.143 -30390.816
## [32,]  72736.6010 105056.185 106948.523  94029.960
## [33,]  19317.1824  27900.522  28403.083  24972.209
```

```
## [34,] -30844.5435 -44549.919 -45352.379 -39874.161
## [35,]  28310.2619  40889.569  41626.098  36597.983
## [36,] -13740.2415 -19845.551 -20203.022 -17762.649
## [37,] -13237.8602 -19119.936 -19464.336 -17113.193
## [38,]  34812.0056  50280.282  51185.962  45003.088
## [39,] -29124.1229 -42065.050 -42822.752 -37650.093
## [40,]  -8538.8862 -12333.035 -12555.185 -11038.614
## [41,] -30224.9416 -43655.008 -44441.349 -39073.175
## [42,]    817.5429   1180.807   1202.076   1056.875
## [43,] 112032.9904 161813.407 164728.091 144830.203
## [44,] -18836.5373 -27206.316 -27696.373 -24350.862
## [45,] -31462.9356 -45443.081 -46261.630 -40673.582
## [46,]  10501.3183  15167.447  15440.652  13575.539
## [47,]   3608.7562   5212.262   5306.149   4665.207
## [48,] -24790.7651 -35806.224 -36451.188 -32048.164
## [49,]  -3497.4817  -5051.538  -5142.529  -4521.352
## [50,] -31691.0284 -45772.534 -46597.017 -40968.455
```

```r
# Extract time and outcome values
#time_vals <- as.integer(colnames(syn_louisiana$data$X))  # Tried to get the years from column names
#needed to hard code it bc it was too hard to extract the years
time_vals <- 2008:2020
#time_vals <- syn_louisiana$data$synth_data$Y0plot[,1]  # Assuming time is the same for all units?
actual_vals <- syn_louisiana$data$synth_data$Y1plot[,1]  # treated LA outcome
synthetic_vals <- syn_louisiana$data$synth_data$Y0plot[,1]  # synthetic control outcome


# Check the extracted values
head(time_vals)
```

```
## [1] 2008 2009 2010 2011 2012 2013
```

```r
head(actual_vals)
```

```
##     2008     2009     2010     2011     2012     2013
## 0.179265 0.173049 0.177919 0.175711 0.166900 0.168220
```

```r
print(synthetic_vals)
```

```
##      2008      2009      2010      2011      2012      2013      2014      2015
## 0.1397160 0.1388650 0.1476530 0.1413780 0.1329400 0.1366100 0.1198200 0.1015400
##      2016      2017      2018      2019      2020
## 0.0920043 0.0954480 0.1017140 0.0965850 0.0970000
```

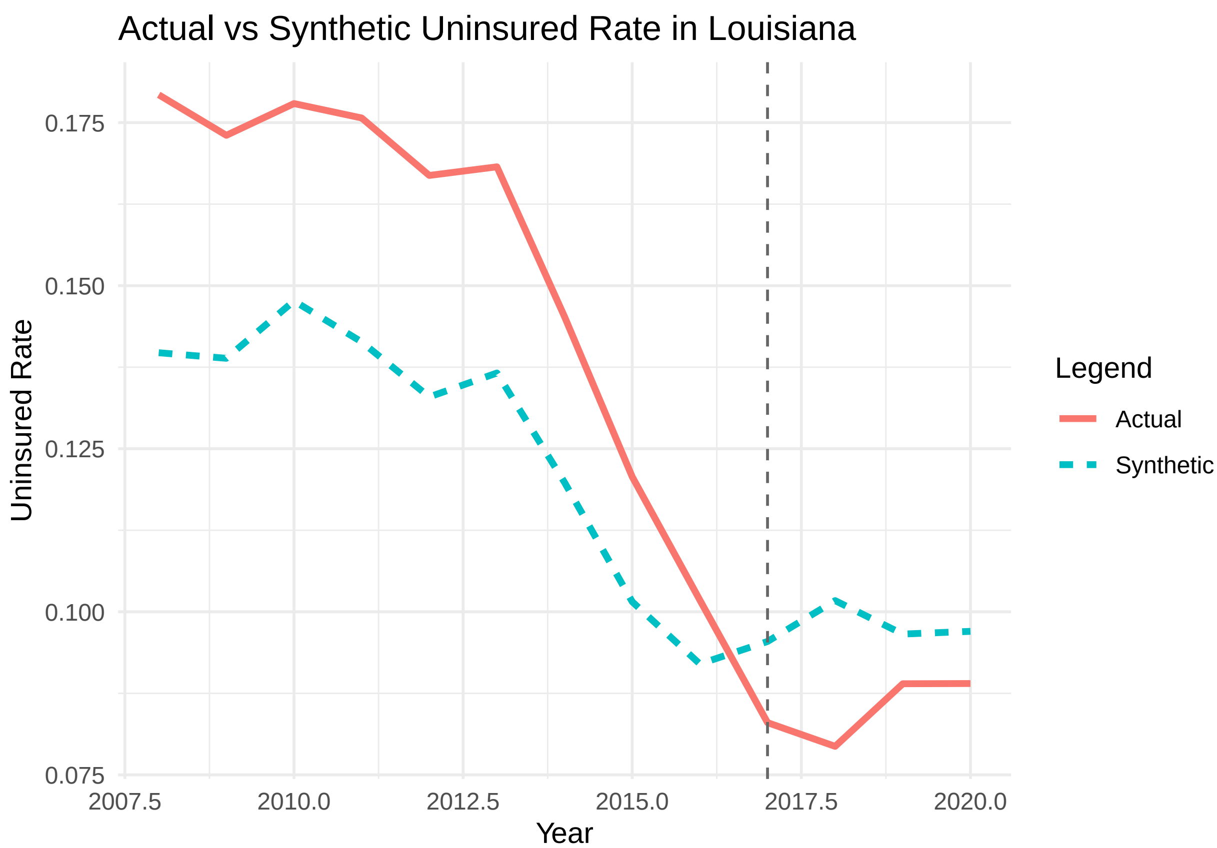
```r
# Create a data frame with columns for Year, Actual, and Synthetic
outcome_df <- data.frame(
  Year = time_vals,
  Actual = actual_vals,
  Synthetic = synthetic_vals
)
```

```r
#is this working
head(outcome_df)
```

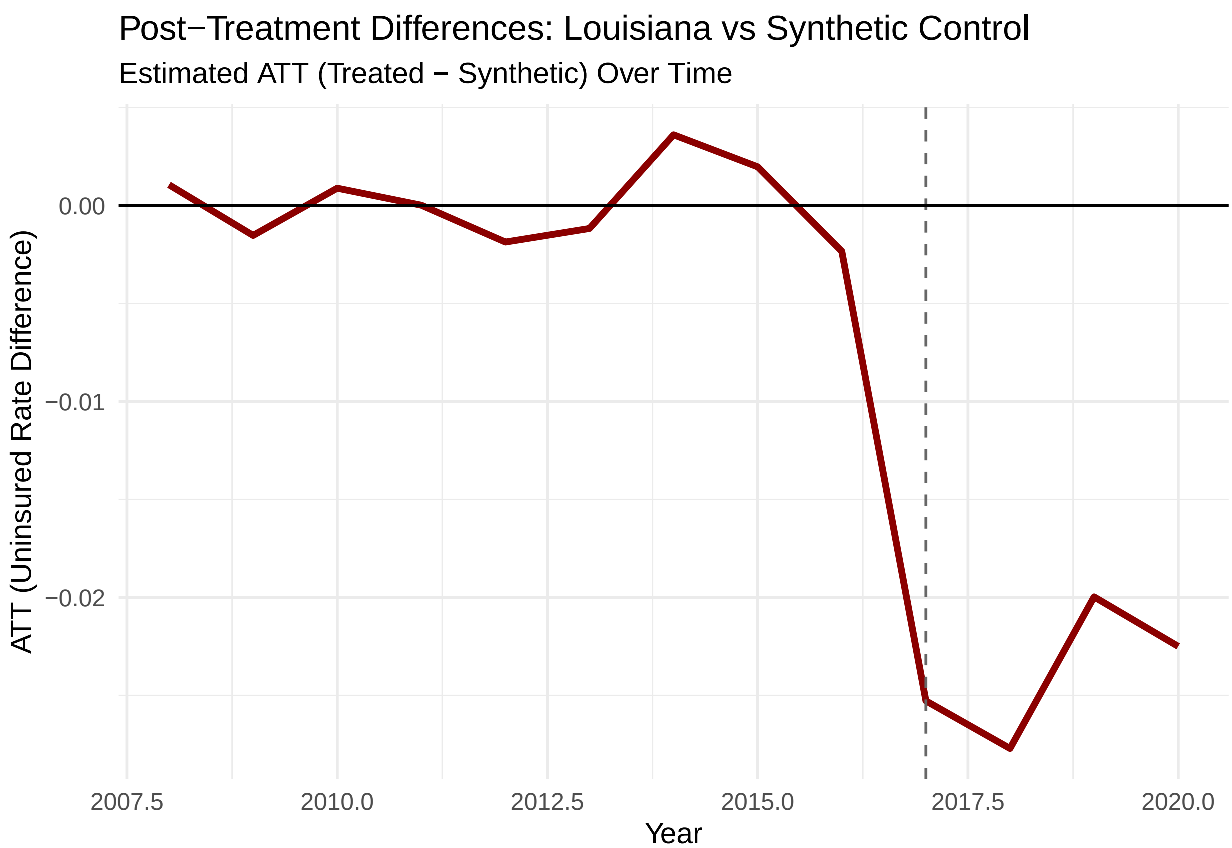
```
##      Year   Actual Synthetic
## 2008 2008 0.179265  0.139716
## 2009 2009 0.173049  0.138865
## 2010 2010 0.177919  0.147653
## 2011 2011 0.175711  0.141378
## 2012 2012 0.166900  0.132940
## 2013 2013 0.168220  0.136610
```

```r
# Convert to long format for ggplot
# Make sure the reshaping process is working correctly
outcome_long_ridge <- pivot_longer(outcome_df,
                             cols = c("Actual", "Synthetic"),
                             names_to = "Type",
                             values_to = "UninsuredRate")

# Check the reshaped data frame
view(outcome_long_ridge)
view(outcome_long)
```

```r
# Create the plot
#pre-treatment

ggplot(outcome_long, aes(x = Year, y = UninsuredRate, color = Type, linetype = Type)) +
  geom_line(size = 1.2) +
  geom_vline(xintercept = syn_louisiana$t_int, linetype = "dashed", color = "gray40") +  # Treatment ti
  labs(
    title = "Actual vs Synthetic Uninsured Rate in Louisiana",
    x = "Year",
    y = "Uninsured Rate",
    color = "Legend",
    linetype = "Legend"
  ) +
  theme_minimal()
```




```r
# Extract ATT time-series
att_df <- summary(syn_louisiana)$att

ggplot(att_df, aes(x = Time, y = Estimate)) +
  geom_line(color = "darkred", size = 1.2) +
  geom_vline(xintercept = summary(syn_louisiana)$t_int, linetype = "dashed", color = "gray40") +
  geom_hline(yintercept = 0, linetype = "solid", color = "black") +
  labs(
    title = "Post-Treatment Differences: Louisiana vs Synthetic Control",
    subtitle = "Estimated ATT (Treated - Synthetic) Over Time",
    x = "Year",
    y = "ATT (Uninsured Rate Difference)"
  ) +
  theme_minimal()
```



- Plot barplots to visualize the weights of the donors.

```r
# barplots of weights
donor_weights <- coef(syn_louisiana)$weights

weights_df <- tibble(
  State = names(donor_weights),
  Weight = as.numeric(donor_weights)
)

summary(syn_louisiana)
```

```
##
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##     t_int = t_int, data = data, progfunc = "ridge", scm = TRUE,
##     unit_name = "Louisiana")
##
## Average ATT Estimate (p Value for Joint Null):  -0.0239   ( 0.29 )
## L2 Imbalance: 0.006
## Percent improvement from uniform weights: 94.6%
##
## Covariate L2 Imbalance: 0.000
## Percent improvement from uniform weights: 96.4%
##
```

```
## Avg Estimated Bias: -484.097
##
## Inference type: Conformal inference
##
##  Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
##  2017   -0.025             -0.073              0.023   0.096
##  2018   -0.028             -0.076              0.020   0.108
##  2019   -0.020             -0.068              0.028   0.188
##  2020   -0.022             -0.071              0.026   0.097
```

```
print(names(donor_weights))
```

```
## NULL
```

```
print(coef(syn_louisiana)$weights)
```

```
## NULL
```

```
print(syn_louisiana$weights)
```

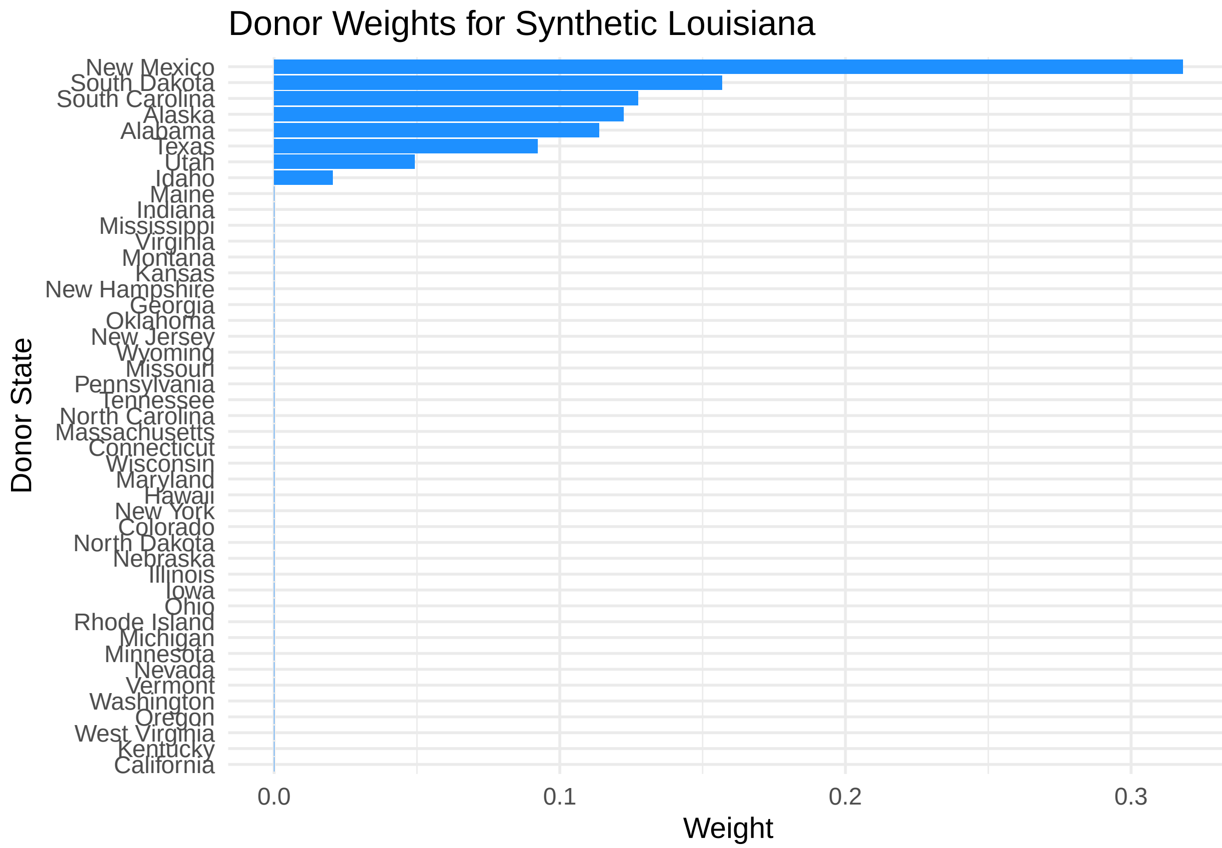
```
##                         [,1]
## Alabama         1.138666e-01
## Alaska          1.224505e-01
## Arizona         8.577500e-06
## Arkansas       -4.399032e-06
## California     -1.682583e-04
## Colorado       -2.855364e-05
## Connecticut    -1.181415e-05
## Delaware       -9.993479e-06
## Florida         5.923672e-06
## Georgia         4.779263e-05
## Hawaii         -2.645079e-05
## Idaho           2.048227e-02
## Illinois       -5.051311e-05
## Indiana         7.726559e-05
## Iowa           -5.208281e-05
## Kansas          5.295293e-05
## Kentucky       -1.213246e-04
## Maine           9.124955e-05
## Maryland       -2.167091e-05
## Massachusetts  -1.142316e-05
## Michigan       -6.404104e-05
## Minnesota      -6.805126e-05
## Mississippi     6.837538e-05
## Missouri        2.602074e-05
## Montana         5.970718e-05
## Nebraska       -4.210162e-05
## Nevada         -6.900935e-05
## New Hampshire   4.988209e-05
## New Jersey      3.906728e-05
## New Mexico      3.181449e-01
## New York       -2.829671e-05
```

```
## North Carolina  1.274303e-05
## North Dakota   -3.727399e-05
## Ohio           -5.543564e-05
## Oklahoma        4.237935e-05
## Oregon         -1.027181e-04
## Pennsylvania    1.775762e-05
## Rhode Island   -6.010410e-05
## South Carolina  1.274309e-01
## South Dakota    1.568010e-01
## Tennessee       1.596000e-05
## Texas           9.226681e-02
## Utah            4.919943e-02
## Vermont        -9.139877e-05
## Virginia        6.679980e-05
## Washington     -1.014398e-04
## West Virginia  -1.167453e-04
## Wisconsin      -1.475153e-05
## Wyoming         3.310635e-05
```

```r
 weights_df <- as.data.frame(syn_louisiana$weights) %>%
  rownames_to_column(var = "State") %>%
  rename(Weight = V1) %>%
  filter(abs(Weight) > 1e-5)
 glimpse(weights_df)
```

```
## Rows: 45
## Columns: 2
## $ State  <chr> "Alabama", "Alaska", "California", "Colorado", "Connecticut", "~
## $ Weight <dbl> 1.138666e-01, 1.224505e-01, -1.682583e-04, -2.855364e-05, -1.18~
```

```r
ggplot(weights_df, aes(x = reorder(State, Weight), y = Weight)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  coord_flip() +
  labs(
    title = "Donor Weights for Synthetic Louisiana",
    x = "Donor State",
    y = "Weight"
  ) +
  theme_minimal()
```



HINT: Is there any preprocessing you need to do before you allow the program to automatically find weights for donor states?

I needed to check for NAs. Should I have standardized? I don't know. I didn't end up using any covariates, so I guess that didn't really matter. I had to create a treatment variable. Not sure what else I was supposed to do.

2. Discussion Questions

- What are the advantages and disadvantages of synthetic control compared to difference-in-differences estimators?
- **Answer** They are kind of similar, but DiD is easier to explain and less computationally expensive than synthetic control (SC). DiD depends so heavily on the parallel trends assumption, so SC is perhaps more robust in certain cases. SC and DiD are both pretty easy to convey in plots, but maybe SC is a little bit easier to understand visually? In general DiD is simpler, faster, and easier but also requires strong assumptions.
- One of the benefits of synthetic control is that the weights are bounded between [0,1] and the weights must sum to 1. Augmentation might relax this assumption by allowing for negative weights. Does this create an interpretation problem, and how should we balance this consideration against the improvements augmentation offers in terms of imbalance in the pre-treatment period?
- **Answer**: Allowing negative weights definitely causes a loss of clear interpretation. The balance has got to be case-specific, I think – it's going to always be a tradeoff between better fit (and therefore better causal estimates) versus a less intuitive interpretation. So it depends what's more important for the study at hand – easily explainable results, or minimized bias.

Staggered Adoption Synthetic Control

Estimate Multisynth

Do the following:

- Estimate a multisynth model that treats each state individually. Choose a fraction of states that you can fit on a plot and examine their treatment effects.

```
# multisynth model states

#starting fresh with the data 663
medicaid_expansion <- read_csv('./data/medicaid_expansion.csv')
```

```
## Rows: 663 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): State
## dbl  (3): year, uninsured_rate, population
## date (1): Date_Adopted
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(medicaid_expansion)
```

```
## Rows: 663
## Columns: 5
## $ State          <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "California~
## $ Date_Adopted   <date> NA, 2015-09-01, 2014-01-01, 2014-01-01, 2014-01-01, 20~
## $ year           <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2~
## $ uninsured_rate <dbl> 0.139716, 0.207716, 0.187312, 0.178883, 0.178212, 0.170~
## $ population     <dbl> 4849377, 737732, 6731484, 2994079, 38802500, 5355856, 3~
```

```
#new treatment var
medicaid_expansion$expansion_year <- year(medicaid_expansion$Date_Adopted)
medicaid_expansion <- medicaid_expansion %>%
  mutate(treatment = if_else(year >= expansion_year, 1, 0))
medicaid_expansion$treatment <- ifelse(is.na(medicaid_expansion$treatment), 0, medicaid_expansion$treatm
glimpse(medicaid_expansion)
```

```
## Rows: 663
## Columns: 7
## $ State          <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "California~
## $ Date_Adopted   <date> NA, 2015-09-01, 2014-01-01, 2014-01-01, 2014-01-01, 20~
## $ year           <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2~
## $ uninsured_rate <dbl> 0.139716, 0.207716, 0.187312, 0.178883, 0.178212, 0.170~
## $ population     <dbl> 4849377, 737732, 6731484, 2994079, 38802500, 5355856, 3~
## $ expansion_year <dbl> NA, 2015, 2014, 2014, 2014, 2014, 2014, 2014, 2014, NA,~
## $ treatment      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```r
medicaid_expansion <- medicaid_expansion %>%
  mutate(treatment = replace_na(treatment, 0))

#debugging. this is not working
medicaid_expansion <- medicaid_expansion %>%
  filter(!is.na(State), !is.na(treatment))
glimpse(medicaid_expansion)
```

```
## Rows: 663
## Columns: 7
## $ State          <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "California~
## $ Date_Adopted   <date> NA, 2015-09-01, 2014-01-01, 2014-01-01, 2014-01-01, 20~
## $ year           <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2~
## $ uninsured_rate <dbl> 0.139716, 0.207716, 0.187312, 0.178883, 0.178212, 0.170~
## $ population     <dbl> 4849377, 737732, 6731484, 2994079, 38802500, 5355856, 3~
## $ expansion_year <dbl> NA, 2015, 2014, 2014, 2014, 2014, 2014, 2014, 2014, NA,~
## $ treatment      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```r
#is it NAs? var formats??
medicaid_expansion$State <- as.factor(medicaid_expansion$State)
medicaid_expansion$year <- as.numeric(medicaid_expansion$year)
medicaid_expansion$expansion_year <- as.numeric(medicaid_expansion$expansion_year)
sum(is.na(medicaid_expansion$State))
```

```
## [1] 0
```

```r
sum(is.na(medicaid_expansion$treatment))
```

```
## [1] 0
```

```r
table(medicaid_expansion$treatment)
```

```
##
##   0   1
## 441 222
```

```r
medicaid_expansion %>%
  filter(is.na(State)) %>%
  nrow()
```

```
## [1] 0
```

```r
medicaid_expansion %>%
  group_by(State) %>%
  summarise(ever_treated = any(treatment == 1)) %>%
  count(ever_treated)
```

```
## # A tibble: 2 x 2
##   ever_treated     n
##   <lgl>        <int>
## 1 FALSE           15
## 2 TRUE            36
```

```r
sum(is.na(medicaid_expansion$uninsured_rate))
```

```
## [1] 0
```

```r
str(medicaid_expansion)
```

```
## tibble [663 x 7] (S3: tbl_df/tbl/data.frame)
##  $ State         : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Date_Adopted  : Date[1:663], format: NA "2015-09-01" ...
##  $ year          : num [1:663] 2008 2008 2008 2008 2008 ...
##  $ uninsured_rate: num [1:663] 0.14 0.208 0.187 0.179 0.178 ...
##  $ population    : num [1:663] 4849377 737732 6731484 2994079 38802500 ...
##  $ expansion_year: num [1:663] NA 2015 2014 2014 2014 ...
##  $ treatment     : num [1:663] 0 0 0 0 0 0 0 0 0 0 ...
```

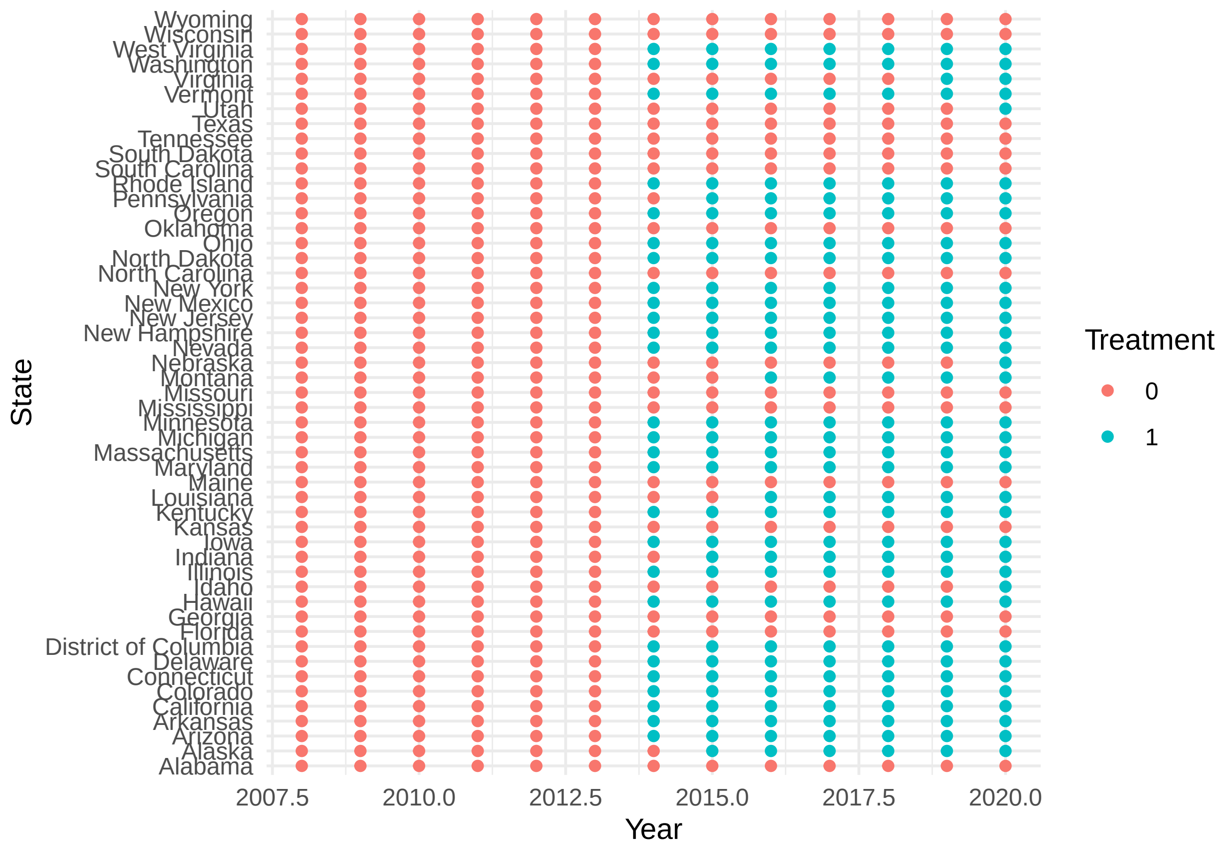
```r
table(medicaid_expansion$treatment, useNA = "always")
```

```
##
##    0    1 <NA>
##  441  222    0
```

```r
#the data appears to make sense....
medicaid_expansion %>%
  group_by(State) %>%
  summarise(min_year = min(year), max_year = max(year)) %>%
  arrange(min_year)
```

```
## # A tibble: 51 x 3
##    State                min_year max_year
##    <fct>                   <dbl>    <dbl>
##  1 Alabama                  2008     2020
##  2 Alaska                   2008     2020
##  3 Arizona                  2008     2020
##  4 Arkansas                 2008     2020
##  5 California               2008     2020
##  6 Colorado                 2008     2020
##  7 Connecticut              2008     2020
##  8 Delaware                 2008     2020
##  9 District of Columbia     2008     2020
## 10 Florida                  2008     2020
## # i 41 more rows
```

```r
#the data appears to make sense...
ggplot(medicaid_expansion, aes(x = year, y = reorder(State, year), color = factor(treatment))) +
  geom_point() +
  labs(color = "Treatment", y = "State", x = "Year") +
  theme_minimal()
```




```r
##what is happening with trt_time
# medicaid_expansion %>%
#   group_by(State) %>%
#   summarise(
#     trt_time = ifelse(any(treatment == 1),
#                       min(year[treatment == 1]),
#                       Inf)  # Inf for control states
#   ) %>%
#   filter(is.na(trt_time))

#All the control states have NA for trt_time, I don't know why

# table(medicaid_expansion$treatment, useNA = "always")
# length(unique(medicaid_expansion$State))


##why is this still not working? Might have to drop some states
good_states <- medicaid_expansion %>%
  group_by(State) %>%
  summarise(has_treatment_info = any(treatment == 1) | all(treatment == 0)) %>%
  filter(has_treatment_info) %>%
  pull(State)
medicaid_expansion <- medicaid_expansion %>%
  filter(State %in% good_states)

table(medicaid_expansion$treatment, useNA = "always")
```

```
##
##    0    1 <NA>
##  441  222    0
```

```
length(unique(medicaid_expansion$State))
```

```
## [1] 51
```

```
##trt_time is the problem for sure, my control states are NA and not Inf, going to do this manually
# trt_times <- medicaid_expansion %>%
#   group_by(State) %>%
#   summarise(
#     trt_time = ifelse(any(treatment == 1),
#                       min(year[treatment == 1]),
#                       Inf)
#   )

# medicaid_expansion <- medicaid_expansion %>%
#   left_join(trt_times, by = "State")

table(medicaid_expansion$trt_time, useNA = "always")
```

```
##
## <NA>
##    0
```

```
# medicaid_expansion <- medicaid_expansion %>%
#   mutate(trt_time = replace_na(trt_time, Inf))

summary(medicaid_expansion$trt_time)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
table(medicaid_expansion$State)
```

```
##
##              Alabama               Alaska              Arizona
##                   13                   13                   13
##             Arkansas           California             Colorado
##                   13                   13                   13
##          Connecticut             Delaware District of Columbia
##                   13                   13                   13
##              Florida              Georgia               Hawaii
##                   13                   13                   13
##                Idaho             Illinois              Indiana
##                   13                   13                   13
##                 Iowa               Kansas             Kentucky
##                   13                   13                   13
##            Louisiana                Maine             Maryland
##                   13                   13                   13
```

```
##        Massachusetts              Michigan             Minnesota
##                   13                    13                    13
##          Mississippi              Missouri               Montana
##                   13                    13                    13
##             Nebraska                Nevada         New Hampshire
##                   13                    13                    13
##           New Jersey            New Mexico              New York
##                   13                    13                    13
##       North Carolina          North Dakota                  Ohio
##                   13                    13                    13
##             Oklahoma                Oregon          Pennsylvania
##                   13                    13                    13
##         Rhode Island        South Carolina          South Dakota
##                   13                    13                    13
##            Tennessee                 Texas                  Utah
##                   13                    13                    13
##              Vermont              Virginia            Washington
##                   13                    13                    13
##        West Virginia             Wisconsin               Wyoming
##                   13                    13                    13
```

```
#is expansion_year the problem?
medicaid_expansion %>% select(State, year, expansion_year) %>% head(10)
```

```
## # A tibble: 10 x 3
##    State                 year expansion_year
##    <fct>                <dbl>          <dbl>
##  1 Alabama               2008             NA
##  2 Alaska                2008           2015
##  3 Arizona               2008           2014
##  4 Arkansas              2008           2014
##  5 California            2008           2014
##  6 Colorado              2008           2014
##  7 Connecticut           2008           2014
##  8 Delaware              2008           2014
##  9 District of Columbia  2008           2014
## 10 Florida               2008             NA
```

```
expansion_info <- medicaid_expansion %>%
  group_by(State) %>%
  summarize(expansion_year = unique(na.omit(expansion_year))[1], .groups = "drop")

medicaid_expansion <- medicaid_expansion %>%
  left_join(expansion_info, by = "State")

glimpse(medicaid_expansion)
```

```
## Rows: 663
## Columns: 8
## $ State              <fct> Alabama, Alaska, Arizona, Arkansas, California, Color~
## $ Date_Adopted       <date> NA, 2015-09-01, 2014-01-01, 2014-01-01, 2014-01-01, ~
## $ year               <dbl> 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008, 2008,~
## $ uninsured_rate     <dbl> 0.139716, 0.207716, 0.187312, 0.178883, 0.178212, 0.1~
```

```
## $ population       <dbl> 4849377, 737732, 6731484, 2994079, 38802500, 5355856,~
## $ expansion_year.x <dbl> NA, 2015, 2014, 2014, 2014, 2014, 2014, 2014, 2014, N~
## $ treatment        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ expansion_year.y <dbl> NA, 2015, 2014, 2014, 2014, 2014, 2014, 2014, 2014, N~
```

```
#class(trt_time)
str(medicaid_expansion)
```

```
## tibble [663 x 8] (S3: tbl_df/tbl/data.frame)
##  $ State           : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Date_Adopted    : Date[1:663], format: NA "2015-09-01" ...
##  $ year            : num [1:663] 2008 2008 2008 2008 2008 ...
##  $ uninsured_rate  : num [1:663] 0.14 0.208 0.187 0.179 0.178 ...
##  $ population      : num [1:663] 4849377 737732 6731484 2994079 38802500 ...
##  $ expansion_year.x: num [1:663] NA 2015 2014 2014 2014 ...
##  $ treatment       : num [1:663] 0 0 0 0 0 0 0 0 0 0 ...
##  $ expansion_year.y: num [1:663] NA 2015 2014 2014 2014 ...
```

```
# Trying a smaller subset

#
# subset_data %>%
#   group_by(State) %>%
#   summarise(
#     min_year = min(year),
#     max_year = max(year),
#     first_treatment_year = min(year[treatment == 1], na.rm = TRUE),
#     trt_time = unique(trt_time)  # See the trt_time values
#   )
#
# subset_data <- medicaid_expansion %>% filter(State == "Louisiana")  # Replace with one state


##I've tried these models every way I can think of and I'm getting the same error.
#
# multi_synth_model <- multisynth(
#   outcome = "uninsured_rate",
#   form = as.formula("uninsured_rate ~ 1"),
#   unit = "State",
#   time = "year",
#   treatment = "treatment",
#   t_int = 2017,
#   data = subset_data,
#   scm = TRUE,
#   progfunc = "none"
# )
#
# multi_syn_model <- multisynth(uninsured_rate ~ treatment,
#                          State,                        # unit
#                          year,                         # time
#                          subset_data,                   # data
#                          trt_time = "expansion_year"
#                          #n_leads = 10 #for staggered adoption
#                          )
```

```r
#more attempts
#
# multi_synth_model <- multisynth(
#   outcome = "uninsured_rate",
#   form = as.formula("uninsured_rate ~ 1"),
#   unit = "State",
#   time = "year",
#   treatment = "treatment",
#   t_int = 2017,
#   data = medicaid_expansion,
#   scm = TRUE,
#   progfunc = "none"
# )
#
# multi_syn_model <- multisynth(
#   uninsured_rate ~ treatment,              # Outcome variable and treatment formula
#   unit = "State",                           # Unit of analysis
#   time = "year",                            # Time variable
#   treatment = "treatment",                  # Treatment indicator (just to be explicit)
#   data = medicaid_expansion,                # Cleaned dataset
#   n_leads = 7                               # For staggered adoption
# )
#
#
# multi_syn_model <- multisynth(
#   form = uninsured_rate ~ treatment,  # Outcome and treatment formula
#   unit = State,                      # Unit column
#   time = year,                       # Time column
#   data = medicaid_expansion,           # Dataset
#   n_leads = 7                          # Number of post-treatment periods
# )
```

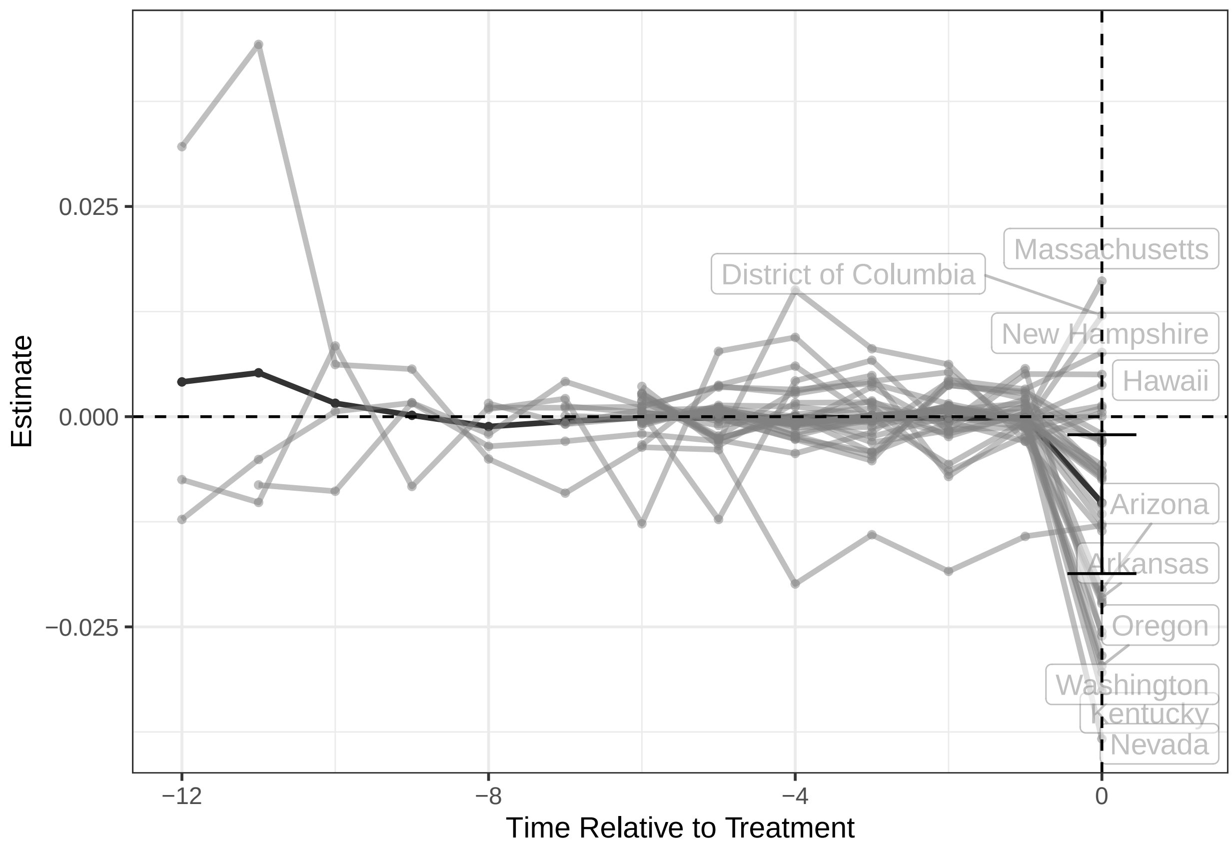
```r
##Ok I was using quotes and I shouldn't have been. Oy

# medicaid_expansion <- medicaid_expansion %>%
#   mutate(treatment = ifelse(!is.na(expansion_year) & year >= expansion_year, 1, 0))
# view(medicaid_expansion)

# Now run multisynth
ms_result <- multisynth(
  uninsured_rate ~ treatment, # Outcome ~ treatment
  unit = State,               # Unit identifier
  time = year,                # Time variable
  data = medicaid_expansion   # Dataset
)

# summary of results
summary(ms_result)
```

```
##
## Call:
## multisynth(form = uninsured_rate ~ treatment, unit = State, time = year,
##     data = medicaid_expansion)
```

```
##
## Average ATT Estimate (Std. Error): -0.010  (0.004)
##
## Global L2 Imbalance: 0.000
## Scaled Global L2 Imbalance: 0.014
## Percent improvement from uniform global weights: 98.6
##
## Individual L2 Imbalance: 0.004
## Scaled Individual L2 Imbalance: 0.090
## Percent improvement from uniform individual weights: 91
##
##  Time Since Treatment   Level    Estimate  Std.Error lower_bound  upper_bound
##                     0 Average -0.01024363 0.00411673 -0.01813015 -0.002367885
```

```
# Plot results
plot(ms_result)
```

```
## Joining with `by = join_by(Level)`
```



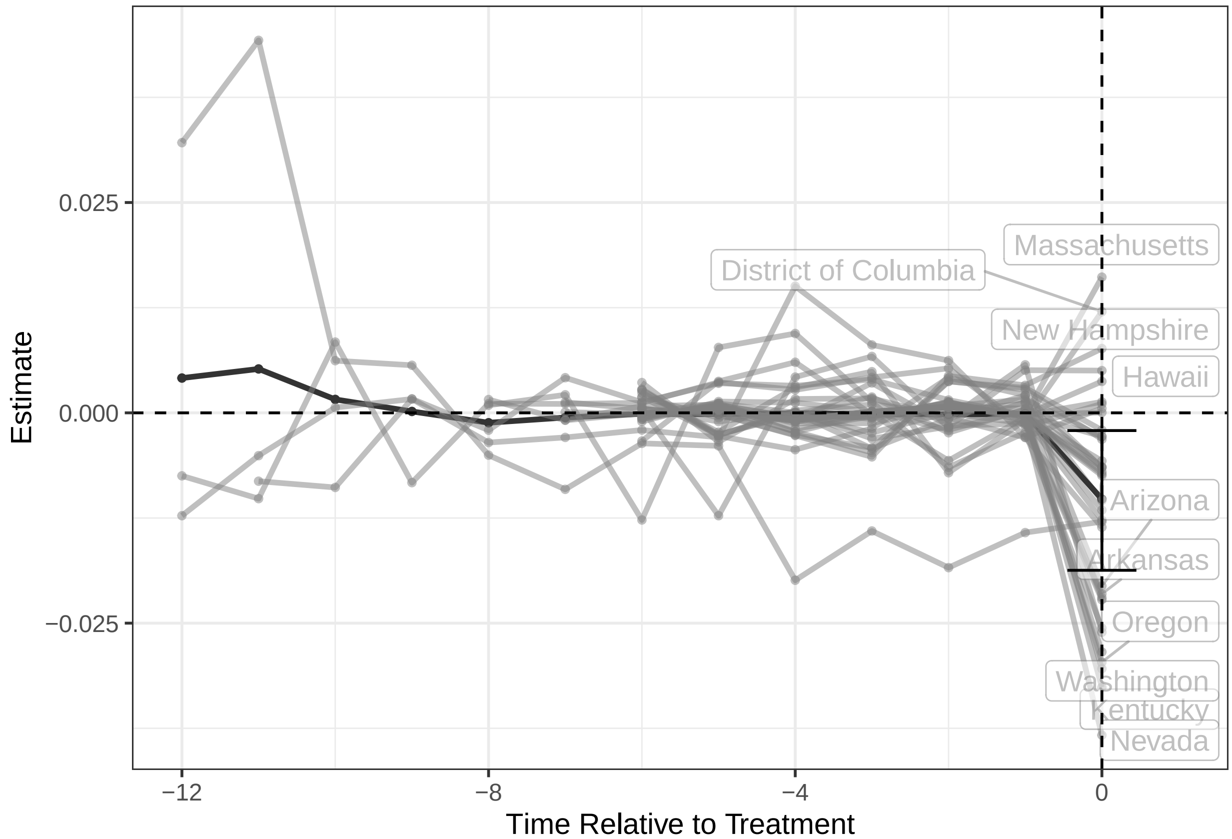
- Estimate a multisynth model using time cohorts. For the purpose of this exercise, you can simplify the treatment time so that states that adopted Medicaid expansion within the same year (i.e. all states that adopted epxansion in 2016) count for the same cohort. Plot the treatment effects for these time cohorts.

```r
# multisynth model time cohorts

# Run multisynth model
multi_syn_model <- multisynth(uninsured_rate ~ treatment,
                        State,                          # unit
                        year,                           # time
                        medicaid_expansion,              # data
                        n_leads = 7,                      #for staggered adoption
                        time_cohort = TRUE
                        )

# summary of results
summary(ms_result)
```

```
##
## Call:
## multisynth(form = uninsured_rate ~ treatment, unit = State, time = year,
##     data = medicaid_expansion)
##
## Average ATT Estimate (Std. Error): -0.010  (0.004)
##
## Global L2 Imbalance: 0.000
## Scaled Global L2 Imbalance: 0.014
## Percent improvement from uniform global weights: 98.6
##
## Individual L2 Imbalance: 0.004
## Scaled Individual L2 Imbalance: 0.090
## Percent improvement from uniform individual weights: 91
##
##  Time Since Treatment   Level    Estimate   Std.Error lower_bound upper_bound
##                     0 Average -0.01024363 0.004251093 -0.01809752 -0.00204918
```

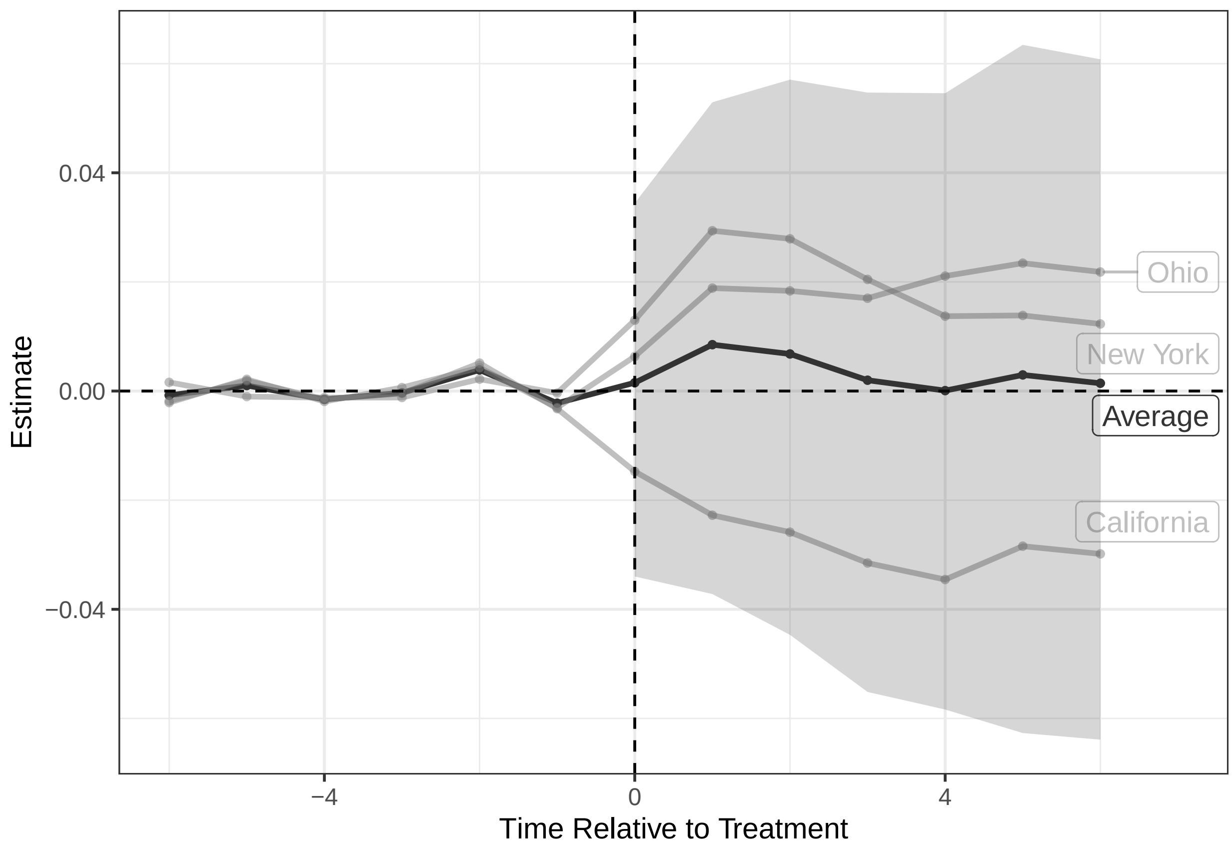
```r
# Plot results
plot(ms_result)
```

```
## Joining with `by = join_by(Level)`
```




```r
##Repeating with just a subset, as requested
states_of_interest <- c("California", "New York", "Texas", "Florida", "Ohio")

subset_data <- medicaid_expansion %>%
  filter(State %in% states_of_interest)

# Run multisynth on subset
subset_ms <- multisynth(
  uninsured_rate ~ treatment,
  unit = State,
  time = year,
  data = subset_data
)

# Plot subset results
plot(subset_ms)
```

```
## Joining with ‘by = join_by(Level)‘
```



3. Discussion Questions

- One feature of Medicaid is that it is jointly administered by the federal government and the states, and states have some flexibility in how they implement Medicaid. For example, during the Trump administration, several states applied for waivers where they could add work requirements to the eligibility standards (i.e. an individual needed to work for 80 hours/month to qualify for Medicaid). Given these differences, do you see evidence for the idea that different states had different treatment effect sizes?
- **Answer**: So if they are given 1115/1915b waivers, states can adapt different conditions to MediCaid – the work requirements are adopted in some states, in others additional services are provided (like tenancy support in CA). In this context, that would mean that the expansions are being implemented heterogenously, and the way they are being implemented (rather than the expansion itself) might be impacting the takeup rate. So yes, it makes sense that different states would have different treatment effect sizes.
- Do you see evidence for the idea that early adopters of Medicaid expansion enjoyed a larger decrease in the uninsured population?
- **Answer**: There's some evidence for that. The straight DiD between Kansas and Arkansas showed a 3.6% decrease for the early adopter. After that, I'm honestly not sure I did the analysis correctly – I was struggling just to get things to run and I don't know why the non-augmented and augmented analyses looked the same, or why only a portion of my selected states showed up in the plot.

4. General Discussion Questions

- Why are DiD and synthetic control estimates well suited to studies of aggregated units like cities, states, countries, etc?
- **Answer**: DiD and SC are good for aggregated units like states because they work well with small numbers of treated units, observational settings (where policies aren't randomly assigned), and instances in which longitudinal data is available. These kinds of units also have abrupt policy changes (like the Medicaid expansion!) which works well for DiD and SC. And these methods use untreated units as controls, so they can take construct counter-factuals from the kind of data typically available at state/county/etc levels.
- What role does selection into treatment play in DiD/synthetic control versus regression discontinuity? When would we want to use either method?
- **Answer**: In DiD and SC, treatment isn't random — we just assume that treated and control groups would've kept following similar trends if there was no treatment. If that assumption is false, the results can be biased. In regression discontinuity (RD), treatment depends on hitting a cutoff, so around that cutoff, it's almost like random assignment. We should use DiD or SC when there are big treatment units (like states) and good pre-treatment trend data, and RD when there's some kind of more or less clear rule that dictates who gets treated.