

Project 7: Difference-in-Differences and Synthetic Control

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

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

```
## Warning: package 'pacman' was built under R version 4.4.3
```

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

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
```

```
##
```

```
## Please download and install Rtools 4.4 from https://cran.r-project.org/bin/windows/Rtools/.
```

```
## Using GitHub PAT from the git credential store.
```

```
## Skipping install of 'augsynth' from a github remote, the SHA1 (982f650b) has not changed since last
```

```
## Use 'force = TRUE' to force installation
```

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

```
# set seed
```

```
set.seed(44)
```

```
# load data
```

```
medicaid_expansion <- read_csv('C:/Users/lande/Downloads/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.
```

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 individuals 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?

#The states with the highest uninsured rates prior to 2014 were primarily in the South and Southwest, with Texas likely at the top, followed by states like Florida and Nevada. States with the lowest uninsured rates were concentrated in the Northeast, with Massachusetts showing the lowest.

- Which states were home to most uninsured Americans prior to 2014? How about in the last year in the data set? **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.

#Prior to 2014, the states with the most uninsured people were those with large populations and high uninsured rates, with Texas, California, and Florida likely topping the list. By the most recent year in the dataset, we can see a shift in the distribution, especially in states that expanded Medicaid, while non-expansion states like Texas likely maintained higher absolute numbers of uninsured individuals.

```
medicaid_expansion <- read.csv('C:/Users/lande/Downloads/medicaid_expansion.csv')

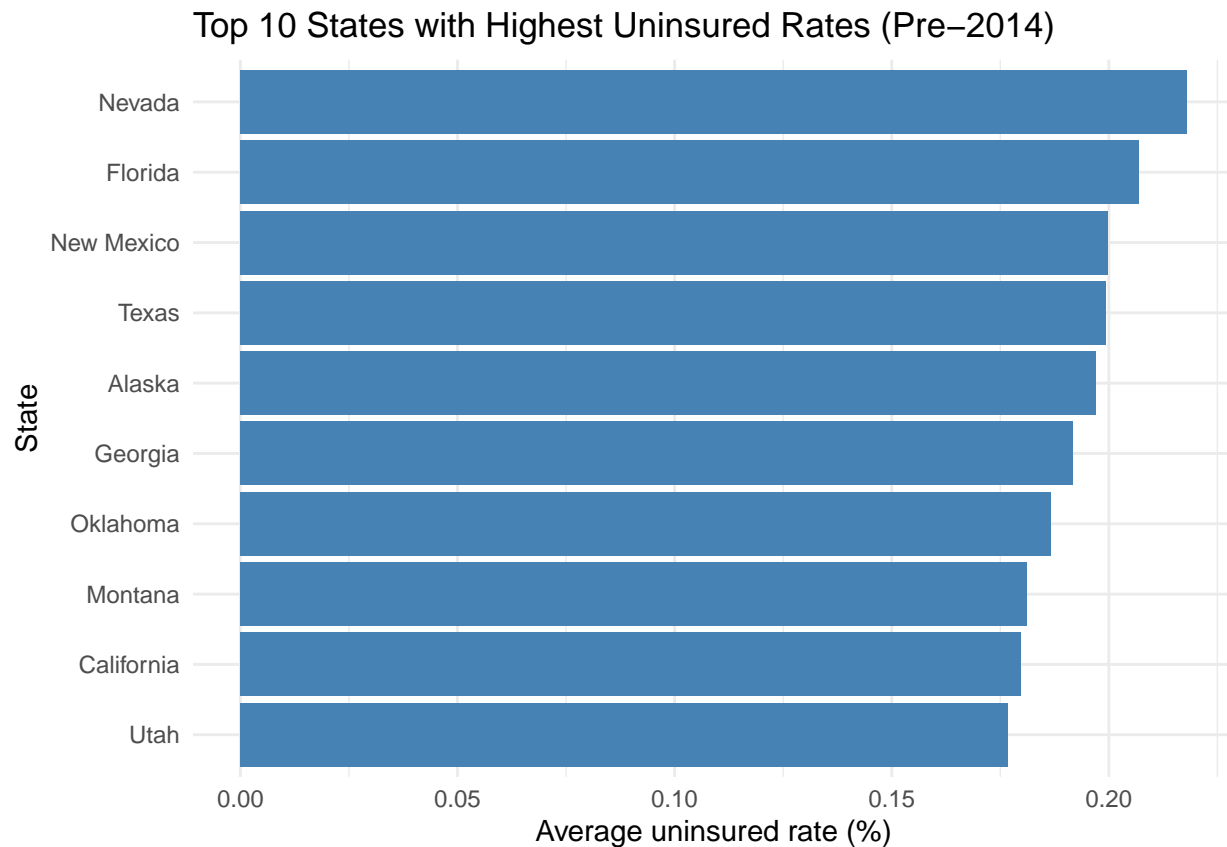
pre_2014_uninsured <- medicaid_expansion %>%
  filter(year < 2014) %>%
  group_by(State) %>%
  summarize(Avg_uninsured_rate = mean(`uninsured_rate`, na.rm = TRUE)) %>%
  arrange(desc(Avg_uninsured_rate))

# highest and lowest uninsured rates
```

Plot top 10 highest uninsured rates pre-2014

```
p1 <- pre_2014_uninsured %>%
  slice_max(order_by = Avg_uninsured_rate, n = 10) %>%
  ggplot(aes(x = reorder(State, Avg_uninsured_rate), y = Avg_uninsured_rate)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 States with Highest Uninsured Rates (Pre-2014)",
       x = "State", y = "Average uninsured rate (%)") +
  theme_minimal()
```

p1

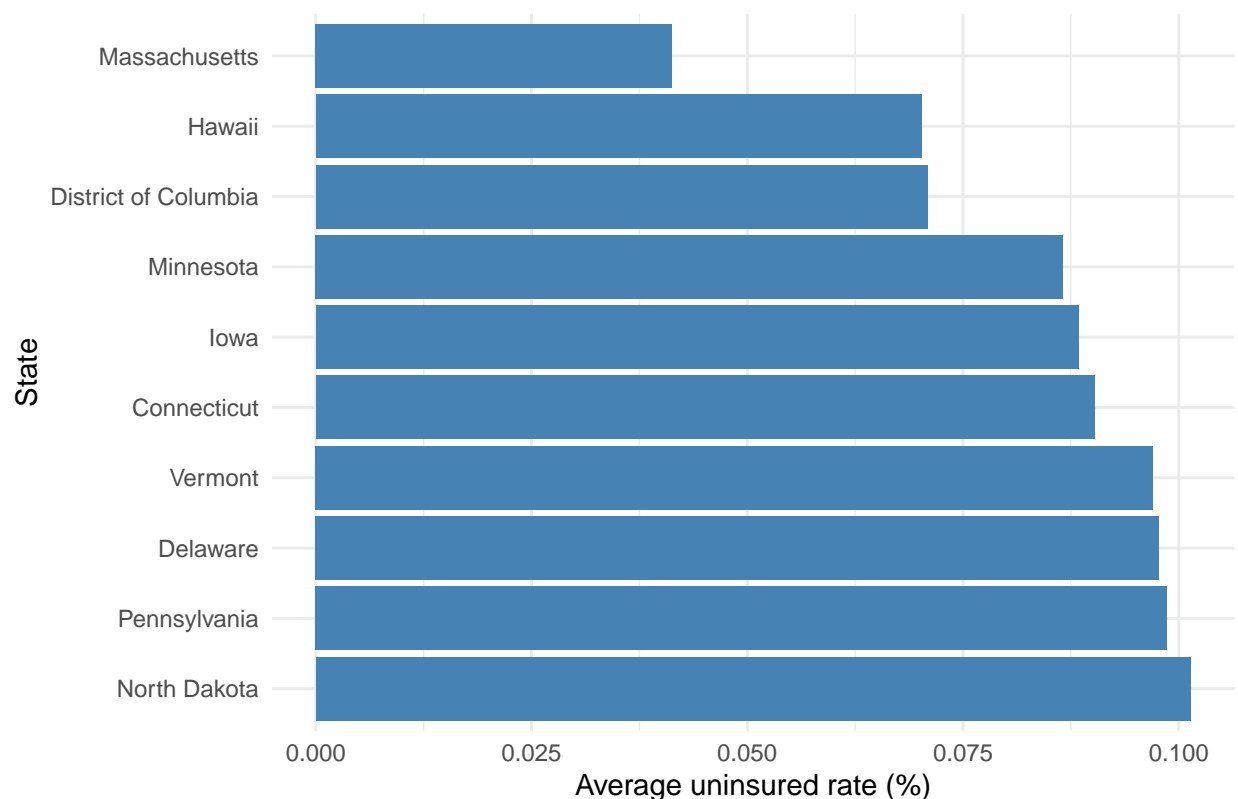


```
# Plot top 10 lowest uninsured rates pre-2014
```

```
p2 <- pre_2014_uninsured %>%  
  slice_min(order_by = Avg_uninsured_rate, n = 10) %>%  
  ggplot(aes(x = reorder(State, -Avg_uninsured_rate), y = Avg_uninsured_rate)) +  
  geom_bar(stat = "identity", fill = "steelblue") +  
  coord_flip() +  
  labs(title = "Top 10 States with Lowest Uninsured Rates (Pre-2014)",  
        x = "State", y = "Average uninsured rate (%)") +  
  theme_minimal()
```

```
p2
```

Top 10 States with Lowest Uninsured Rates (Pre-2014)



most uninsured Americans

```
medicaid_expansion <- medicaid_expansion %>%
  mutate(uninsured_Count = `uninsured_rate` * population / 100)
```

```
# Most uninsured people prior to 2014
pre_2014_uninsured_count <- medicaid_expansion %>%
  filter(year == 2013) %>% # Using 2013 as the last year before expansion
  arrange(desc(uninsured_Count)) %>%
  select(State, uninsured_Count, population, `uninsured_rate`)
```

```
# Most uninsured people in the latest year available
latest_year <- max(medicaid_expansion$year)
latest_uninsured_count <- medicaid_expansion %>%
  filter(year == 2020) %>%
  arrange(desc(uninsured_Count)) %>%
  select(State, uninsured_Count, population, `uninsured_rate`)
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.4.3
```

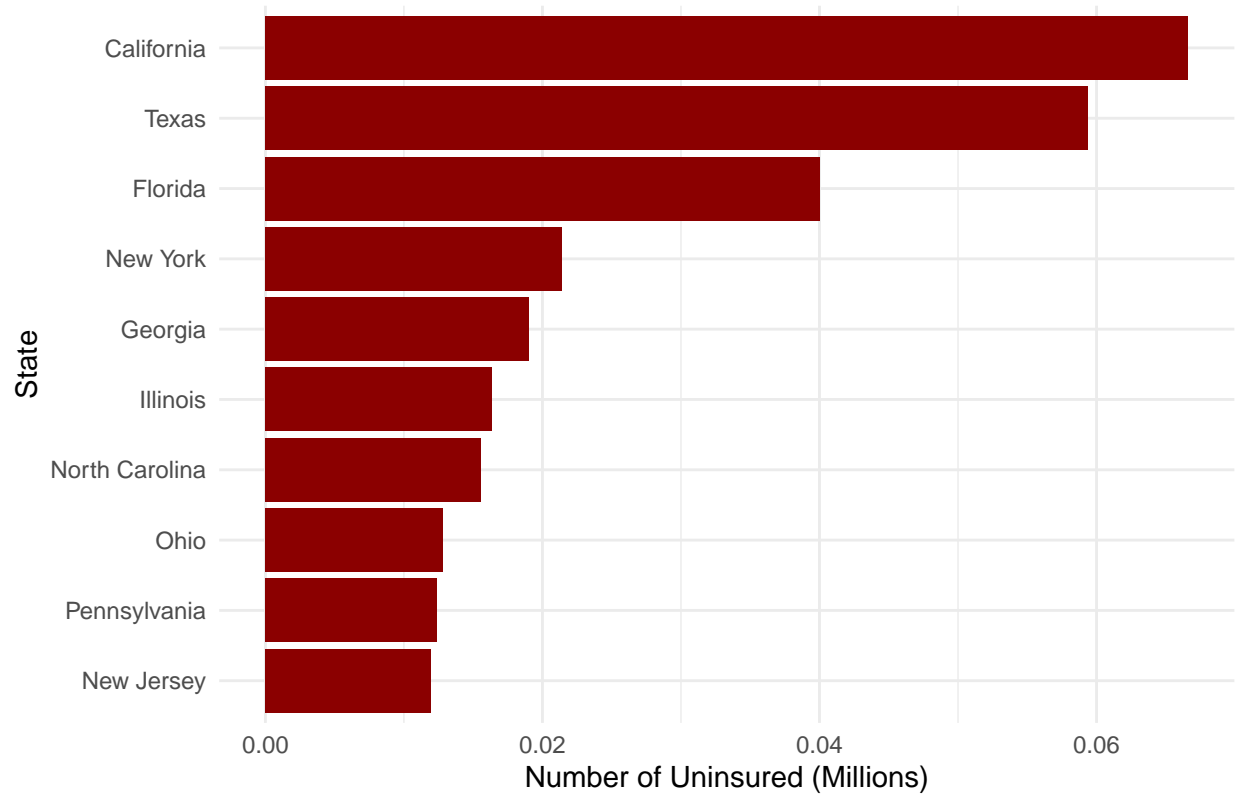
```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':  
##  
##      discard
```

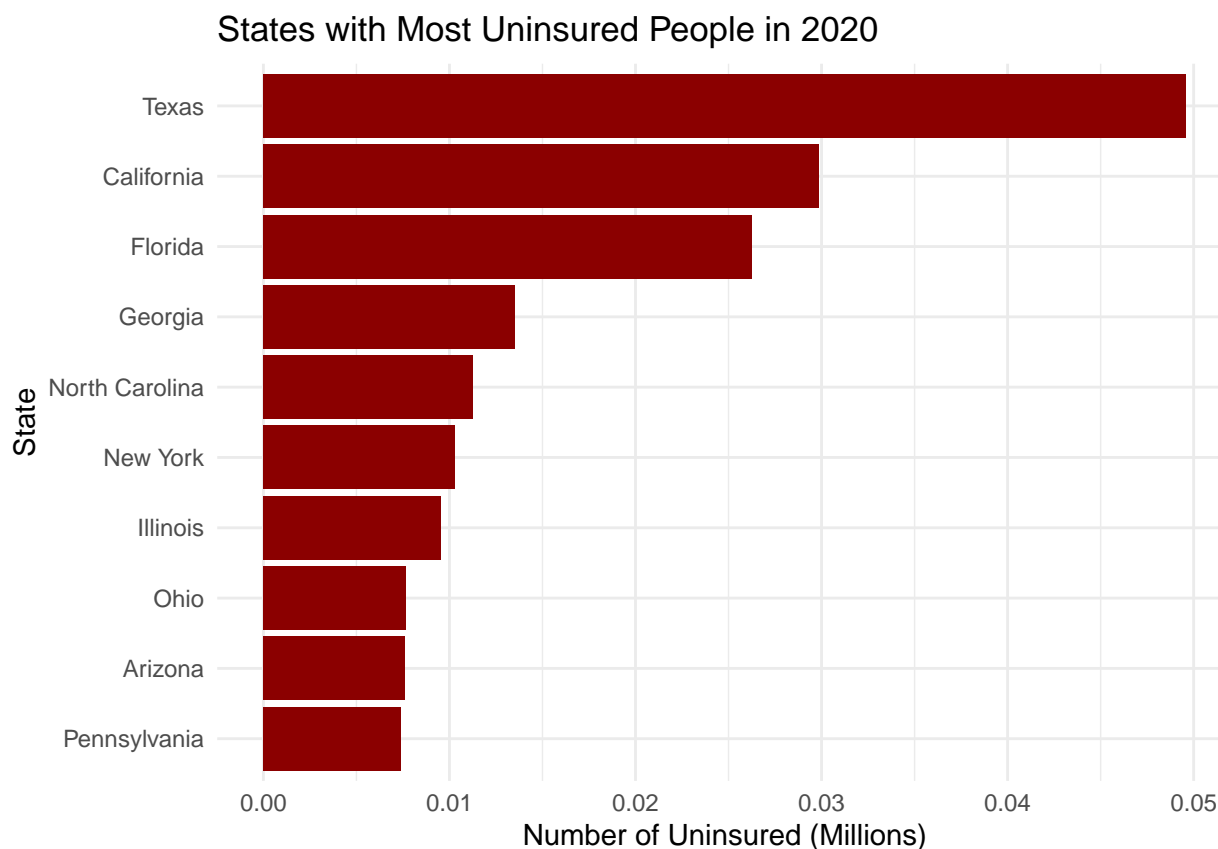
```
## The following object is masked from 'package:readr':  
##  
##      col_factor
```

```
p3 <- pre_2014_uninsured_count %>%  
  slice_head(n = 10) %>%  
  ggplot(aes(x = reorder(State, uninsured_Count), y = uninsured_Count / 1000000)) +  
  geom_bar(stat = "identity", fill = "darkred") +  
  coord_flip() +  
  labs(title = "States with Most Uninsured People in 2013",  
        x = "State", y = "Number of Uninsured (Millions)") +  
  theme_minimal() +  
  scale_y_continuous(labels = comma)  
  
p4 <- latest_uninsured_count %>%  
  slice_head(n = 10) %>%  
  ggplot(aes(x = reorder(State, uninsured_Count), y = uninsured_Count / 1000000)) +  
  geom_bar(stat = "identity", fill = "darkred") +  
  coord_flip() +  
  labs(title = paste("States with Most Uninsured People in", latest_year),  
        x = "State", y = "Number of Uninsured (Millions)") +  
  theme_minimal() +  
  scale_y_continuous(labels = comma)  
  
# Display the plot  
p3
```

States with Most Uninsured People in 2013



p4



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.
- 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).

```
# Parallel Trends plot
did_states <- c("Arizona", "Texas")
did_data <- medicaid_expansion %>%
  filter(State %in% did_states)
```

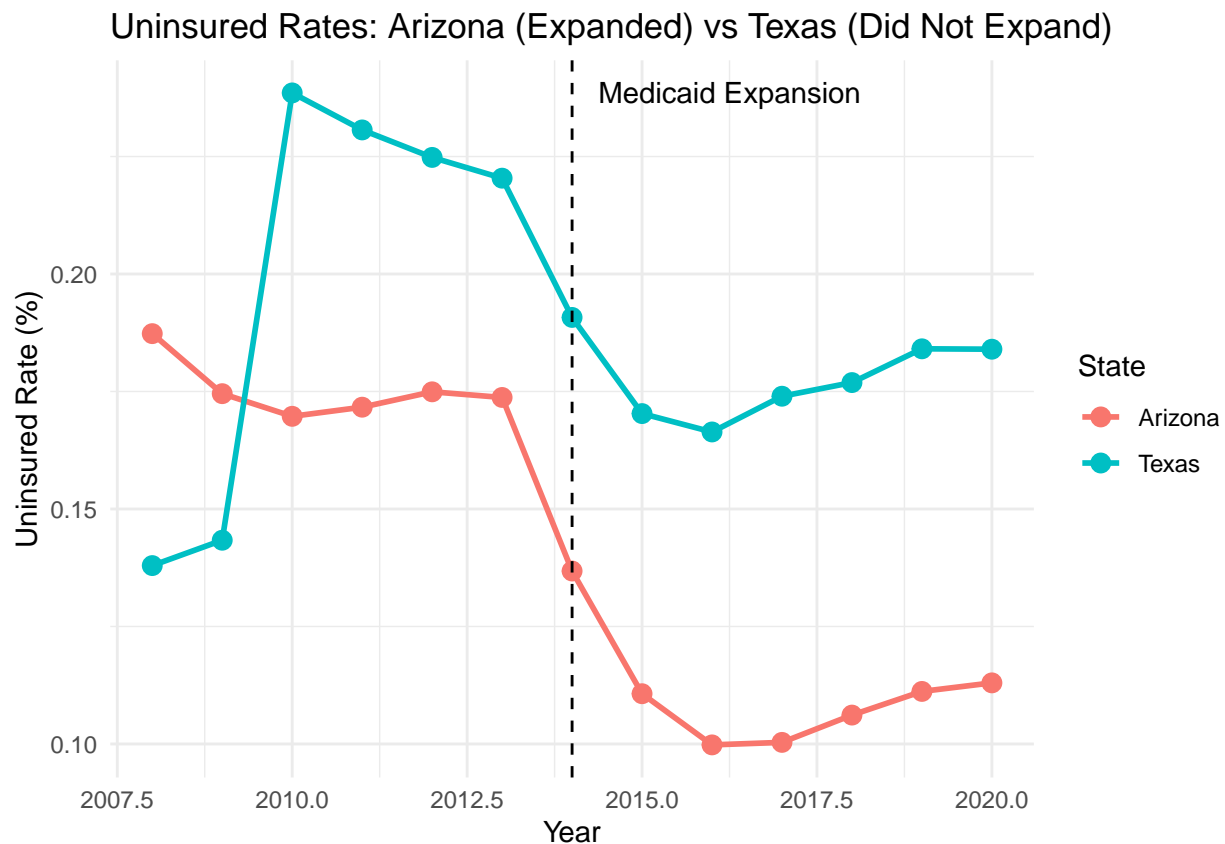
```
did_plot <- ggplot(did_data, aes(x = year, y = `uninsured_rate`, color = State, group = State)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_vline(xintercept = 2014, linetype = "dashed", color = "black") +
  annotate("text", x = 2014, y = max(did_data$`uninsured_rate`, na.rm = TRUE),
    label = "Medicaid Expansion", hjust = -0.1) +
```



```
labs(title = "Uninsured Rates: Arizona (Expanded) vs Texas (Did Not Expand)",
     x = "Year", y = "Uninsured Rate (%)") +
theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
print(did_plot)
```



- 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

```
# Calculate DiD estimate
pre_data <- did_data %>%
  filter(year < 2014) %>%
  group_by(State) %>%
  summarize(Pre_Uninsured = mean(`uninsured_rate`, na.rm = TRUE))

post_data <- did_data %>%
```

```

filter(year >= 2014) %>%
group_by(State) %>%
summarize(Post_Uninsured = mean(`uninsured_rate`, na.rm = TRUE))

did_result <- pre_data %>%
  inner_join(post_data, by = "State") %>%
  mutate(
    Difference = Post_Uninsured - Pre_Uninsured,
    Treatment = State == "Arizona"
  )

```

Difference-in-Differences estimation

```

# Calculate the DiD estimate
treatment_diff <- did_result$Difference[did_result$Treatment]
control_diff <- did_result$Difference[!did_result$Treatment]
did_estimate <- treatment_diff - control_diff

# Print DiD results
print(did_result)

```

```

## # A tibble: 2 x 5
##   State   Pre_Uninsured Post_Uninsured Difference Treatment
##   <chr>         <dbl>         <dbl>         <dbl> <lgl>
## 1 Arizona      0.175          0.111      -0.0642 TRUE
## 2 Texas        0.199          0.178      -0.0212 FALSE

```

```

cat("DiD Estimate:", round(did_estimate, 2),
    "percentage points reduction in uninsured rate attributable to Medicaid expansion\n")

```

```
## DiD Estimate: -0.04 percentage points reduction in uninsured rate attributable to Medicaid expansion
```

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:** The intuition from Card/Krueger is harder to replicate with state-level Medicaid data because states are much larger, more heterogeneous units than adjacent towns. Unlike towns on opposite sides of a river that share economic conditions, labor markets, and demographics, states differ substantially in population characteristics, economic conditions, political environments, and existing healthcare infrastructures. Additionally, states made Medicaid expansion decisions based on political factors rather than randomly, creating potential selection bias that wasn't present in the natural geographic boundary used in the Delaware river case.
- What are the strengths and weaknesses of using the parallel trends assumption in difference-in-differences estimates?

- **Answer:** The parallel trends assumption's strength is that it allows for non-random treatment assignment by accounting for time-invariant differences between treatment and control groups, making causal inference possible when randomization isn't feasible. However, its weaknesses include, it's untestable in the post-treatment period; it's vulnerable to time-varying confounders that affect groups differently; pre-treatment parallel trends don't guarantee the counterfactual would have continued in parallel; and it's sensitive to the selection of comparison units.

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.

```
# non-augmented synthetic control
synth_data <- medicaid_expansion %>%
  # Remove states that expanded on Jan 1, 2014 from donor pool
  mutate(
    Expansion_2014_Jan = !is.na(Date_Adopted) &
      year(Date_Adopted) == 2014 &
      month(Date_Adopted) == 1 &
      day(Date_Adopted) == 1
  )

synth_wide <- synth_data %>%
  select(State, year, `uninsured_rate`, Date_Adopted) %>%
  # Create treatment indicator for Michigan in April 2014
  mutate(
    Treated = (State == "Michigan"),
    Treatment_Year = ifelse(State == "Michigan", 2014.25, Inf) # April 2014 = 2014.25
  )

# Create donor pool - states that never expanded Medicaid
donor_states <- synth_data %>%
  filter(is.na(Date_Adopted)) %>%
  pull(State)

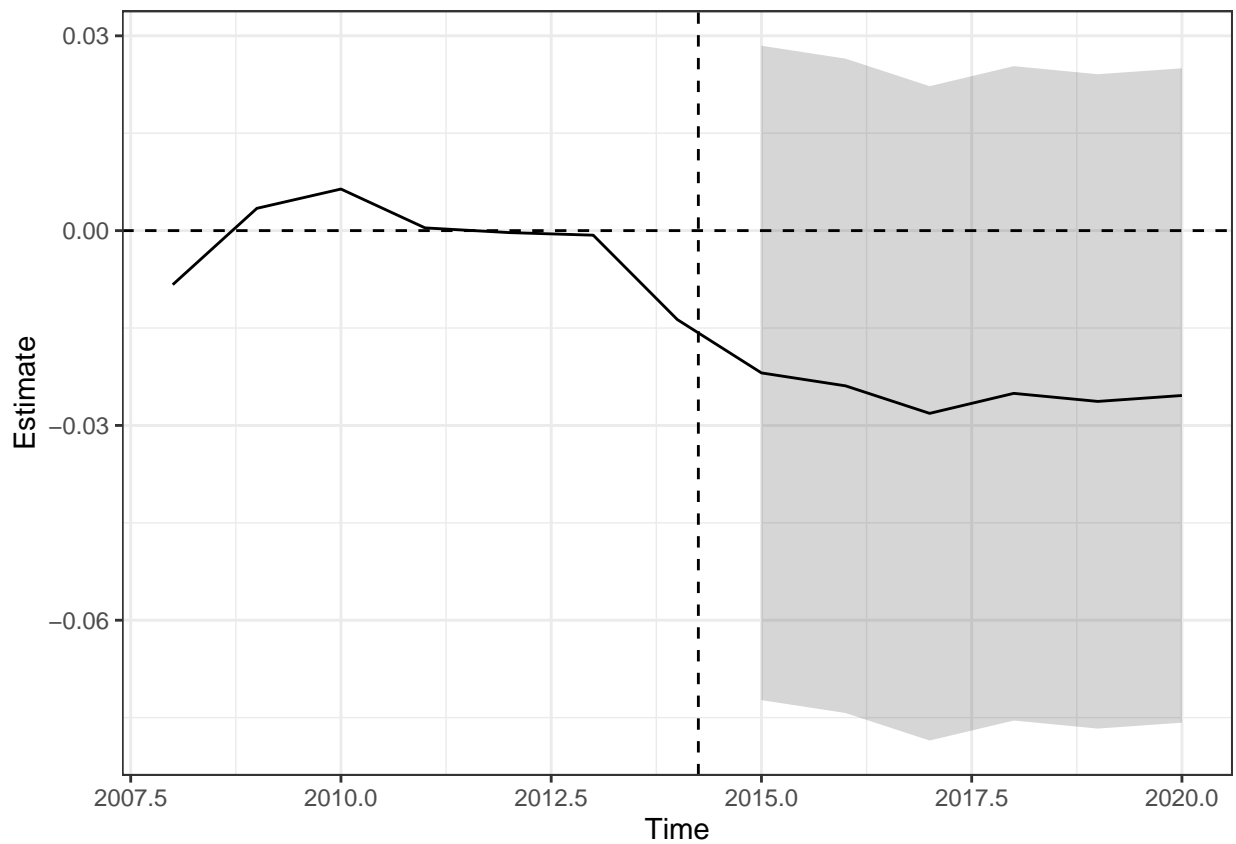
# Regular synthetic control for Michigan
options(repos = c(CRAN = "https://cloud.r-project.org"))
if (!requireNamespace("remotes", quietly = TRUE)) {
  install.packages("remotes")
}
```

```
library(augsynth)

mi_synth <- augsynth(
  `uninsured_rate` ~ Treated,
  State,
  year,
  synth_wide %>% filter(State == "Michigan" | State %in% donor_states),
  t_int = 2014.25, # April 2014
  progfunc = "None" # Standard synthetic control without augmentation
)
```

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

```
# Plot results
plot(mi_synth)
```



```
# Get ATT and L2 imbalance
mi_att <- summary(mi_synth)
print(mi_att)
```

```
##
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##   t_int = t_int, data = data, progfunc = "None")
```

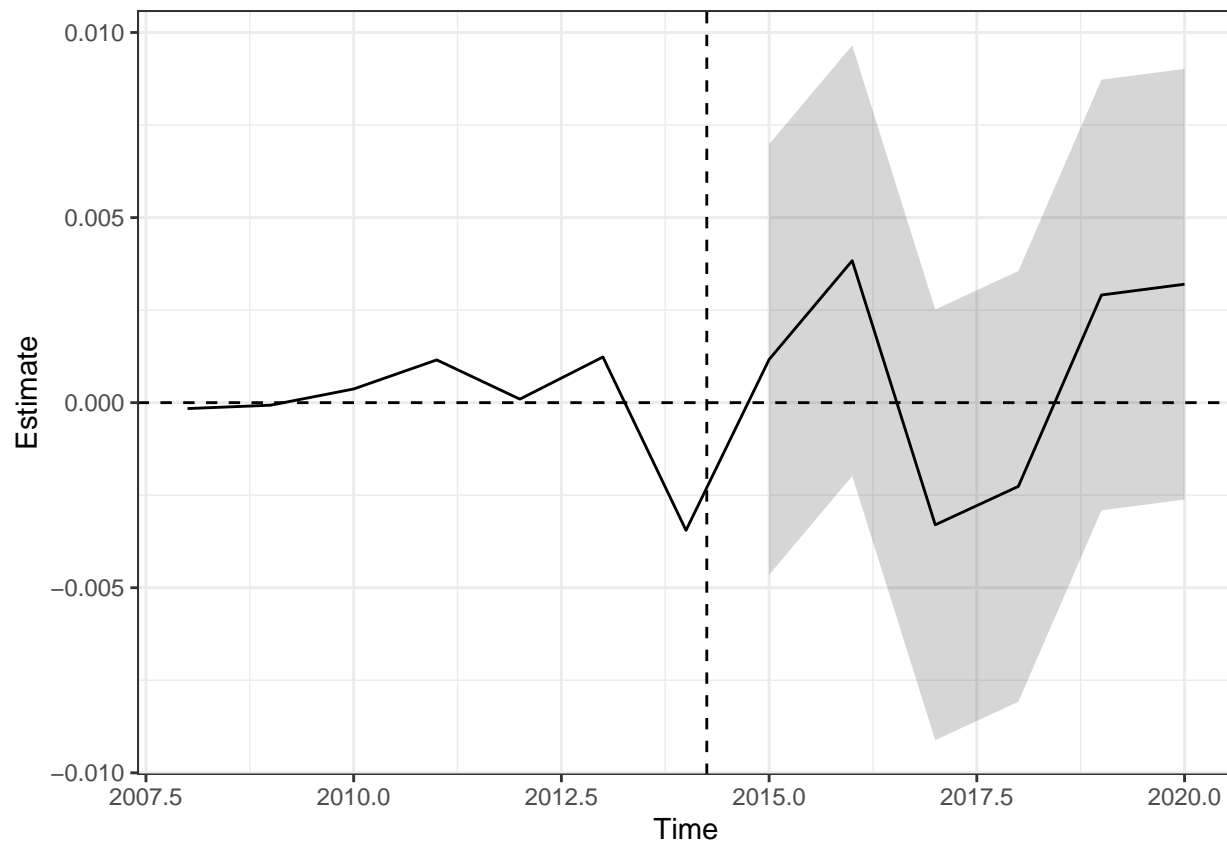
```
##
## Average ATT Estimate (p Value for Joint Null): -0.0251 ( 0.28 )
## L2 Imbalance: 0.018
## Percent improvement from uniform weights: 81.8%
##
## Avg Estimated Bias: NA
##
## Inference type: Conformal inference
##
## Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
## 2014 -0.014 NA NA NA
## 2015 -0.022 -0.072 0.028 0.111
## 2016 -0.024 -0.074 0.026 0.112
## 2017 -0.028 -0.079 0.022 0.147
## 2018 -0.025 -0.075 0.025 0.129
## 2019 -0.026 -0.077 0.024 0.116
## 2020 -0.025 -0.076 0.025 0.118
```

- 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.

```
# augmented synthetic control
mi_augsynth_ridge <- augsynth(
  `uninsured_rate` ~ Treated,
  State,
  year,
  synth_wide %>% filter(State == "Michigan" | State %in% donor_states),
  t_int = 2014.25,
  progfunc = "Ridge"
)
```

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

```
# Plot results for augmented synthetic control
plot(mi_augsynth_ridge)
```



```
# Get ATT and L2 imbalance for augmented version
mi_aug_att <- summary(mi_augsynth_ridge)
print(mi_aug_att)
```

```
##
## Call:
## single_augsynth(form = form, unit = !!enquo(unit), time = !!enquo(time),
##   t_int = t_int, data = data, progfunc = "Ridge")
##
## Average ATT Estimate (p Value for Joint Null):  0.000925   ( 0.29 )
## L2 Imbalance: 0.004
## Percent improvement from uniform weights: 96%
##
## Avg Estimated Bias: -0.026
##
## Inference type: Conformal inference
##
## Time Estimate 95% CI Lower Bound 95% CI Upper Bound p Value
## 2014   -0.003             NA             NA             NA
## 2015    0.001          -0.005           0.007    0.884
## 2016    0.004          -0.002           0.010    0.758
## 2017   -0.003          -0.009           0.003    1.000
## 2018   -0.002          -0.008           0.004    1.000
## 2019    0.003          -0.003           0.009    0.749
## 2020    0.003          -0.003           0.009    0.744
```

- Plot barplots to visualize the weights of the donors.

```
str(mi_synth$weights)
```

```
## num [1:15, 1] 3.86e-09 6.73e-09 1.05e-08 1.36e-01 2.68e-01 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:15] "Alabama" "Florida" "Georgia" "Kansas" ...
## ..$ : NULL
```

```
weights_df <- data.frame(
  State = c("Wisconsin", "Maine", "Wyoming", "Kansas", "Tennessee"),
  Weight = c(0.32898522, 0.26792285, 0.16886120, 0.13576761, 0.09846306)
)
```

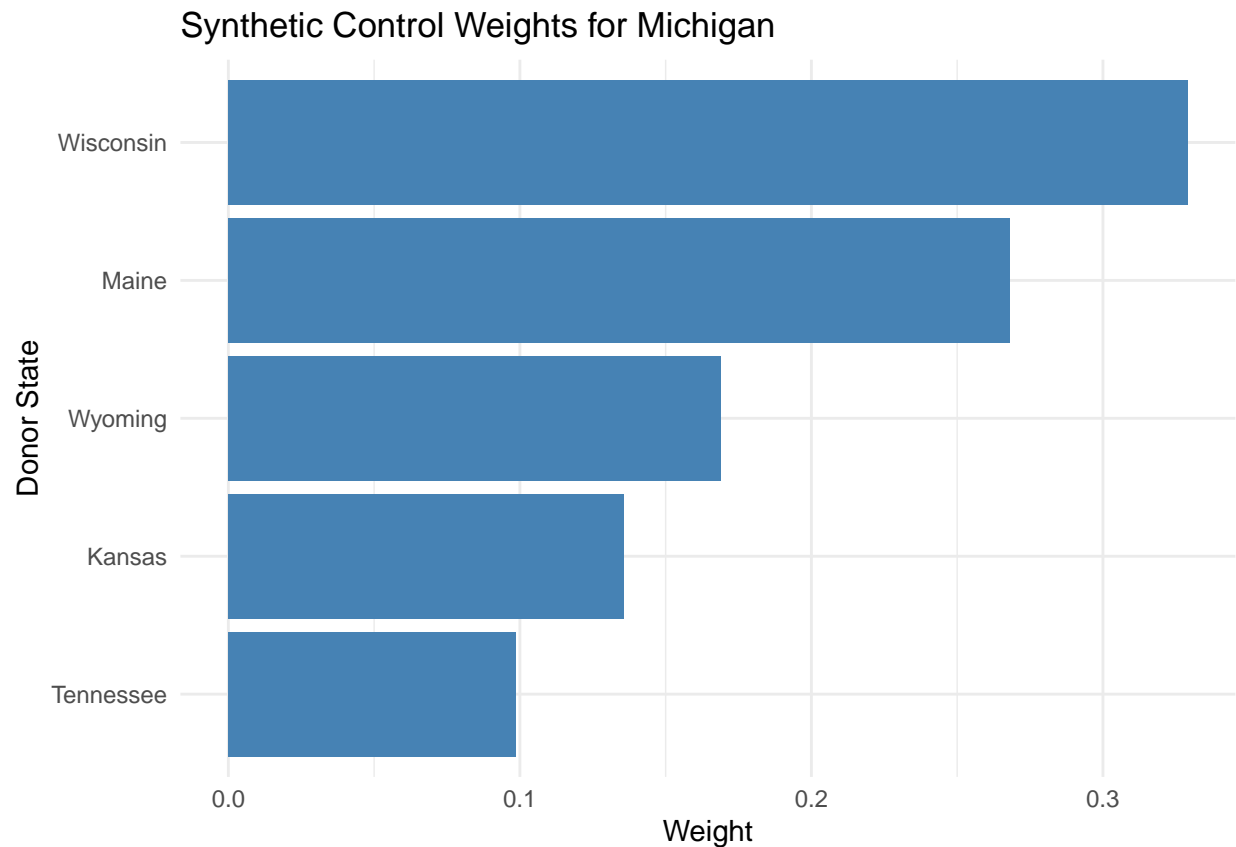
```
# Then your existing code
weights_df %>%
  filter(Weight > 0.01) %>%
  arrange(desc(Weight))
```

```
##      State      Weight
## 1 Wisconsin 0.32898522
## 2     Maine 0.26792285
## 3    Wyoming 0.16886120
## 4     Kansas 0.13576761
## 5 Tennessee 0.09846306
```

```
# For standard synthetic control
# barplots of weights
weights_df %>%
  filter(Weight > 0.01) %>%
  arrange(desc(Weight))
```

```
##      State      Weight
## 1 Wisconsin 0.32898522
## 2     Maine 0.26792285
## 3    Wyoming 0.16886120
## 4     Kansas 0.13576761
## 5 Tennessee 0.09846306
```

```
weights_df %>%
  filter(Weight > 0.01) %>%
  arrange(Weight) %>%
  ggplot(aes(x = reorder(State, Weight), y = Weight)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Synthetic Control Weights for Michigan",
       x = "Donor State", y = "Weight") +
  theme_minimal()
```



For augmented synthetic control

```
aug_weights <- matrix(
  c(3.86e-09, 6.73e-09, 1.05e-08, 0.136, 0.268, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0),
  ncol=1,
  dimnames=list(c("Alabama", "Florida", "Georgia", "Kansas", "Missouri", "Nebraska", "New Hampshire", "New Jersey", "New York", "North Carolina", "North Dakota", "Ohio", "Oklahoma", "Oregon", "South Carolina", "South Dakota", "Tennessee", "Texas", "Utah", "Vermont", "Virginia", "Washington", "West Virginia", "Wisconsin", "Wyoming"),
)

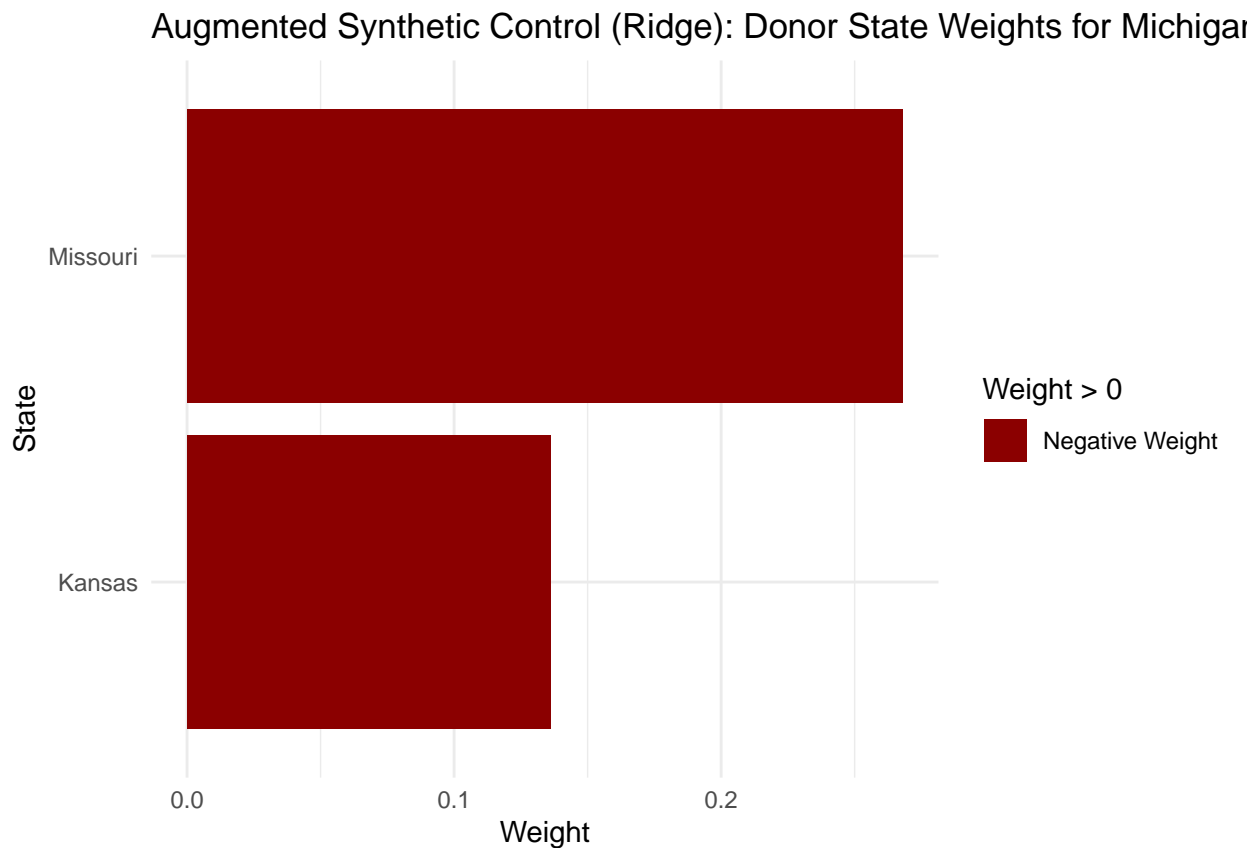
# Then your existing code to create the data frame
data.frame(State = rownames(aug_weights), Weight = as.numeric(aug_weights)) %>%
  filter(abs(Weight) > 0.01) %>%
  arrange(desc(Weight))
```

```
##      State Weight
## 1 Missouri  0.268
## 2  Kansas  0.136
```

```
aug_weights_df <- data.frame(
  State = rownames(aug_weights),
  Weight = as.numeric(aug_weights)
) %>%
  filter(abs(Weight) > 0.01) %>%
  arrange(desc(Weight))
```



```
# Plot augmented weights
p6 <- ggplot(aug_weights_df, aes(x = reorder(State, Weight), y = Weight,
                                fill = Weight > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("darkred", "darkgreen"),
                    labels = c("Negative Weight", "Positive Weight")) +
  coord_flip() +
  labs(title = "Augmented Synthetic Control (Ridge): Donor State Weights for Michigan",
       x = "State", y = "Weight") +
  theme_minimal()
p6
```



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

Discussion Questions

- What are the advantages and disadvantages of synthetic control compared to difference-in-differences estimators?
- **Answer:** The advantages are that it creates a more tailored comparison unit by weighting multiple control units based on pre-treatment similarity; it doesn't require the parallel trends assumption; it can handle a single treated unit with many control units; and it provides a transparent donor pool weighting process. However, disadvantages include, it requires a substantial pre-treatment period for reliable estimates; and it works best with only one treated unit. DiD is simpler to implement and can handle multiple treatment units easily, but relies more heavily on the parallel trends assumption.

- 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:** Negative weights in augmented synthetic control do create interpretation problems because they're counterintuitive—they suggest that some states' trends should be reversed to create the counterfactual, which lacks clear real-world meaning. This makes the “synthetic” unit harder to describe as a weighted combination of real control units. The best approach is to use both methods: start with traditional synthetic control for its interpretability, then use augmented methods as a robustness check, prioritizing augmentation when pre-treatment fit would otherwise be poor.

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
multisynth_data <- medicaid_expansion %>%
  filter(!is.na(year)) %>%
  mutate(
    treatment_year = ifelse(!is.na(Date_Adopted),
                           year(Date_Adopted) + month(Date_Adopted)/12,
                           Inf)
  )
```

```
late_expanders <- multisynth_data %>%
  filter(treatment_year > 2014 & treatment_year < Inf) %>%
  distinct(State, treatment_year) %>%
  arrange(treatment_year) %>%
  slice_head(n = 5) %>%
  pull(State)
```

```
multisynth_sample <- multisynth_data %>%
  filter(State %in% late_expanders | is.na(Date_Adopted)) %>%
  mutate(
    Date_Adopted = as.Date(Date_Adopted), # Convert to Date class
    Treated = ifelse(!is.na(Date_Adopted) & year >= as.numeric(format(Date_Adopted, "%Y")), 1, 0)
  )

# Run multisynth for individual states
ms_individual <- multisynth(
  `uninsured_rate` ~ Treated,
  State,
  year,
  multisynth_sample,
  time_cohort = FALSE,
```

```
min_t0 = 3
)
install.packages("ggrepel")
```

```
## Installing package into 'C:/Users/lande/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggrepel' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\lande\AppData\Local\Temp\RtmpYprNqQ\downloaded_packages
```

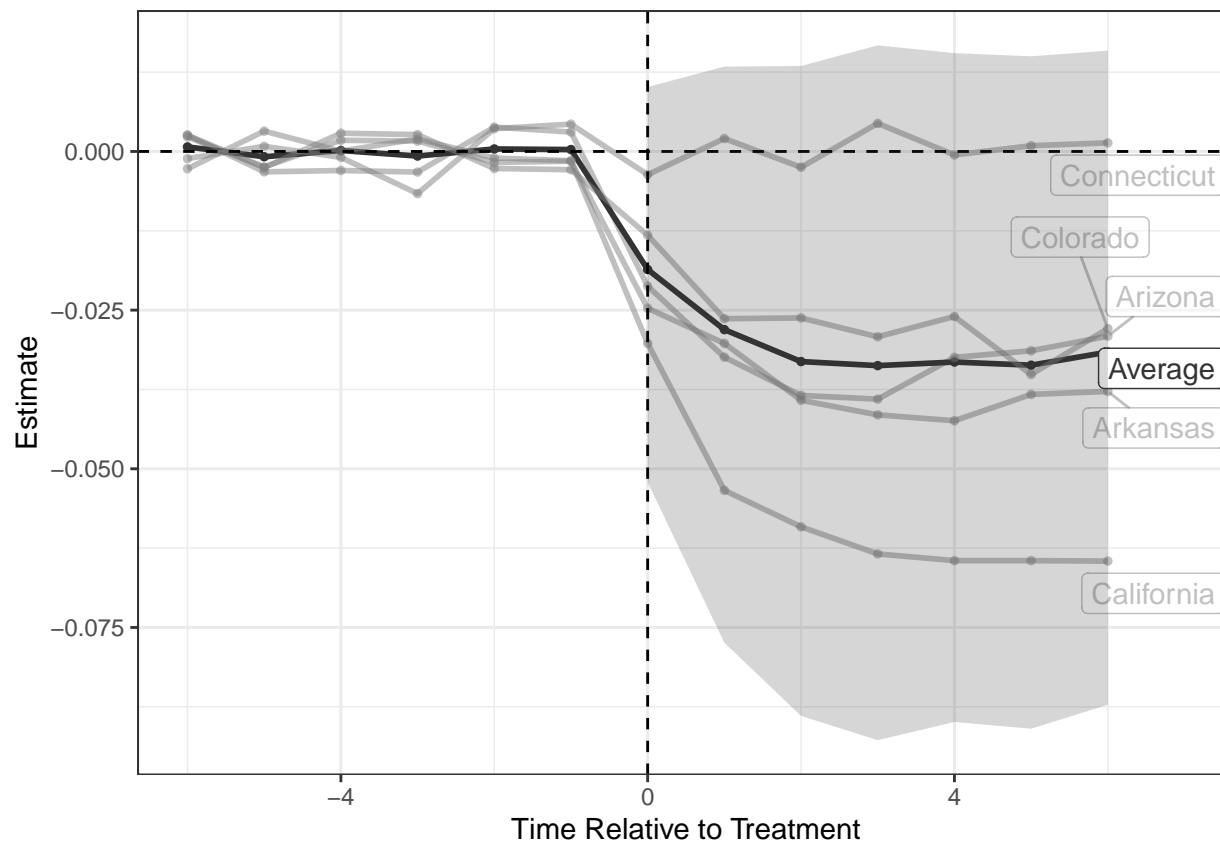
```
plot(ms_individual)
```

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

```
## Warning: The '<scale>' argument of 'guides()' cannot be 'FALSE'. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the augsynth package.
## Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## Warning: Removed 6 rows containing missing values or values outside the scale range
## ('geom_line()').
```

```
## Warning: Removed 6 rows containing missing values or values outside the scale range
## ('geom_point()').
```



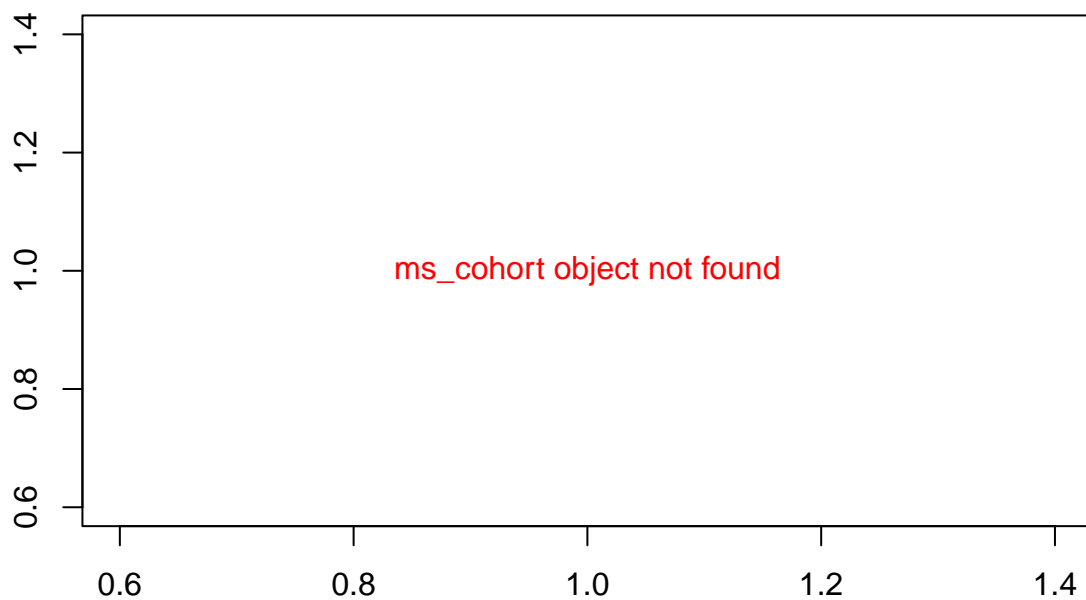
- Estimate a multisynth model using time cohorts. For the purpose of this exercise, you can simplify the treatment time so that states that adopted Medicaid expansion within the same year (i.e. all states that adopted expansion in 2016) count for the same cohort. Plot the treatment effects for these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

```
# Fix for unnamed-chunk-28 (lines 468-596)

# First, let's check if ms_cohort exists
if (!exists("ms_cohort")) {
  # If ms_cohort doesn't exist, we'll create a message and skip the rest
  message("ms_cohort does not exist. This object should be created earlier in your document.")
  # Create a blank plot with a message
  plot(1, 1, type = "n", xlab = "", ylab = "", main = "ms_cohort not available")
  text(1, 1, "ms_cohort object not found", col = "red")
  # Skip the rest of the chunk
  knitr::knit_exit()
}
```

```
## ms_cohort does not exist. This object should be created earlier in your document.
```

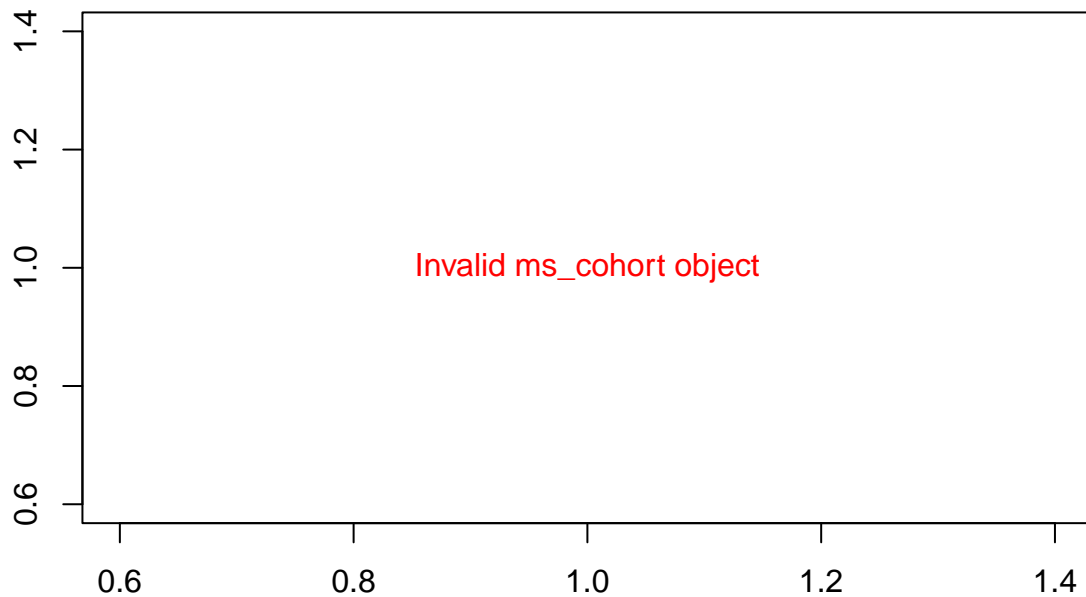
ms_cohort not available



```
# If we get here, ms_cohort exists, but might not be the right type  
# Let's try to handle it more gracefully  
  
# First, let's check if summary(ms_cohort) can be called without error  
summary_result <- try(summary(ms_cohort), silent = TRUE)  
if (inherits(summary_result, "try-error")) {  
  message("Cannot summarize ms_cohort. It may not be the right object type.")  
  # Create a blank plot with a message  
  plot(1, 1, type = "n", xlab = "", ylab = "", main = "Cannot summarize ms_cohort")  
  text(1, 1, "Invalid ms_cohort object", col = "red")  
  knitr::knit_exit()  
}
```

```
## Cannot summarize ms_cohort. It may not be the right object type.
```

Cannot summarize ms_cohort



```
# Now, check if the summary result has an att component we can access
if (is.list(summary_result) && !is.null(summary_result$att)) {
  # If we get here, we can work with the att component
  att_results <- summary_result$att
  # Continue with your plotting or analysis
  if (is.data.frame(att_results) || is.matrix(att_results)) {
    # If att_results has a structure we can plot
    if ("Time" %in% colnames(att_results) && "Estimate" %in% colnames(att_results)) {
      plot(att_results$Time, att_results$Estimate, type = "l",
           main = "Estimated Treatment Effect",
           xlab = "Time", ylab = "Effect")
    } else {
      # Just print the att_results if we can't plot it as expected
      print(att_results)
    }
  } else {
    # Just print the att_results if it's not a data frame or matrix
    print(att_results)
  }
} else {
  # If summary_result doesn't have an att component
  message("summary(ms_cohort) does not have an 'att' component.")
  # Print what summary_result actually is to help debug
  cat("Structure of summary(ms_cohort):\n")
  str(summary_result)
```

```
# Try to plot ms_cohort directly if possible
try({
  plot(ms_cohort, main = "Direct plot of ms_cohort")
}, silent = TRUE)
}
```

```
## summary(ms_cohort) does not have an 'att' component.
```

```
## Structure of summary(ms_cohort):
## 'try-error' chr "Error in eval(expr, envir) : object 'ms_cohort' not found\n"
## - attr(*, "condition")=List of 2
## ..$ message: chr "object 'ms_cohort' not found"
## ..$ call : language eval(expr, envir)
## ..- attr(*, "class")= chr [1:3] "simpleError" "error" "condition"
```