Problem Set 6

Due on October 17, 2023, 10am

To successfully complete this problem set, please follow these steps:

1. Download the contents of the assignment Dropbox link in a folder (directory) on your computer designated for this problem set only. Follow the directory structure, e.g., putting the `data` folder containing the datasets as a subdirectory of the project directory.

2. Insert your answers in the yellow boxes using Microsoft Word, and prepare a single `.R` script for what you produce. Save the word document as a`.PDF`.

3. Please submit the PDF to the designated `PS-XX: pdf` link and your R Script to the `PS-XX: R` link.

(1) Your name:

Nikhilla Bhuvana Sundar

(2) Group members, if any:

(3) Compliance with the Academic Code on problem set[1] (sign with an X below)

X

[1] You may use the same code from classmates, Ed Discussion Board, instructors, and generative AI. However, you must hand in your own unique written work and code in all cases. Any wholesale copy/paste of another’s work is plagiarism. In other words, you can work with your classmate(s), sitting side-by-side and going through the problem set question-by-question, or use generative AI to provide potential code for you, but you must each type your own answers and your own code.

Autograder will have the following packages installed:

```
library(tidyverse)
library(haven)
library(modelsummary)
library(panelView)
library(fixest)
library(broom)
library(glue)
library(flextable)
library(scales)
```

This problem set continues with the following article:

> Ken Scheve and David Stasavage (2012), “Democracy, War, and Wealth: Lessons from Two Centuries of Inheritance Taxation”. *American Political Science Review*.

We will discuss this article on October 12th. Please enter the pre-class exercise before class.

To review the core DID / TWFE implementation, you might also want to look at the screencasts below.

- Part 1, panel structure: https://vimeo.com/872554679
- Part 2, fixed effects: https://vimeo.com/872555241

Problem 1

The base data for this paper is an annual panel dataset in long form. This is renamed to `data/ScheveStasavage_annual.dta` from the previous problem set and includes the variables:

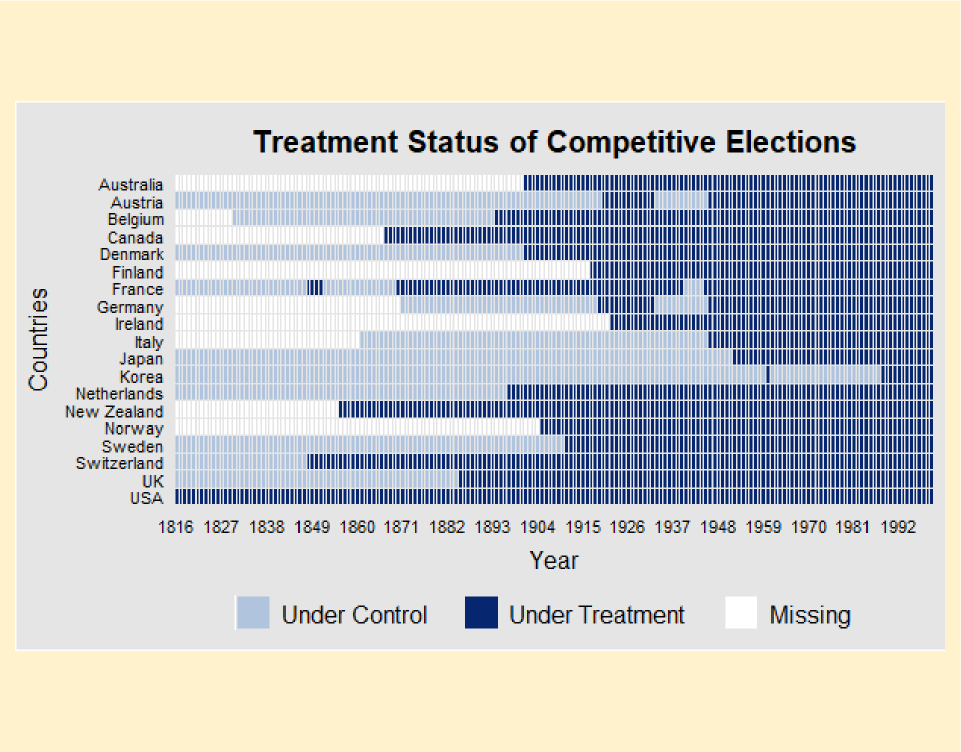
Variable name	**Description**
`year`	Year of measurement
`country`, `ccode`	country, and associated code
`topitax`	Top marginal tax rate for a single direct descendant receiving cash inheritance, in percent units (so that a value of 50 indicates a 50% marginal tax rate). Named `topitaxrate2` in the original dataset
`war2p`	War Mobilization: engaged in interstate war, 2% of population serving in the military (named `himobpopyear2p` in the original dataset)
`unisuffrage`	Universal Male Suffrage: all adult males eligible to vote in national elections

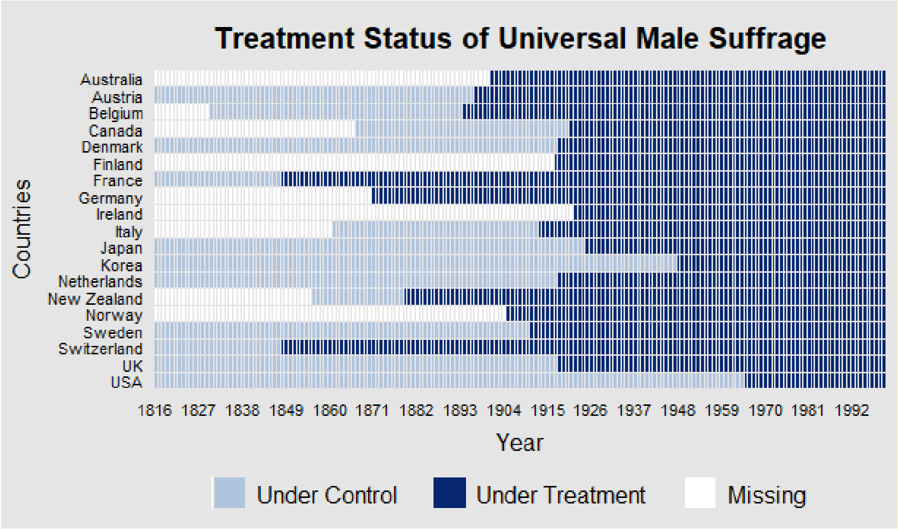
Variable name	Description
`democracy`	Competitive elections: whether legislative elections are free multiparty elections, the executive is elected by popular vote, and at least majority of the male population is eligible to vote.
`leftexec`	Left executive: whether the head of government is from the Socialist or Social Democratic Party
`rgdppc`	Real Gross GDP per capita

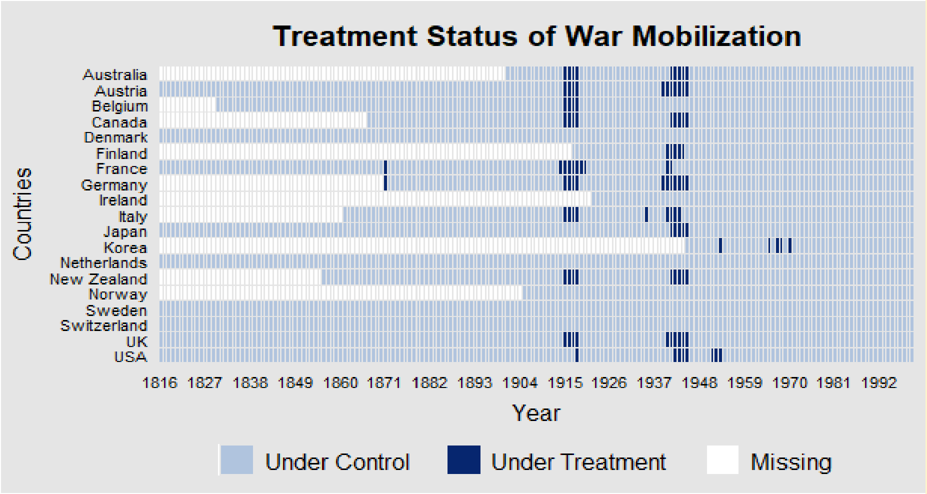
1.1

When working with complex panel data, it is revealing to visualize the structure of treatment assignment before getting buried in regression specifications. In the previous problem set, you visualized the outcome variable (the inheritance tax rate). Here, let's visualize the treatment status with the annual dataset.

The function `panelview` in the `panelView` package provides some shortcuts to make a wide visualization with long form. Using this function, **make three separate treatment status panelview plots**, as shown in the tutorials https://yiqingxu.org/packages/panelview/articles/tutorial.html. The three treatment variables in the paper are war mobilization, universal male suffrage, and competitive elections. Use the annual data for this graphic, but make sure the x-axis labels are legible.







1.2

In words, explain the parallel trends assumption — in the context of this study — that is necessary for the simple two way fixed effects estimator to be valid (i.e., provide an unbiased estimate of the causal effect). Limit your response to a sentence or two.

The counterfactual trend of inheritance tax rates in UK absent war mobilization is the same as those of the Netherlands

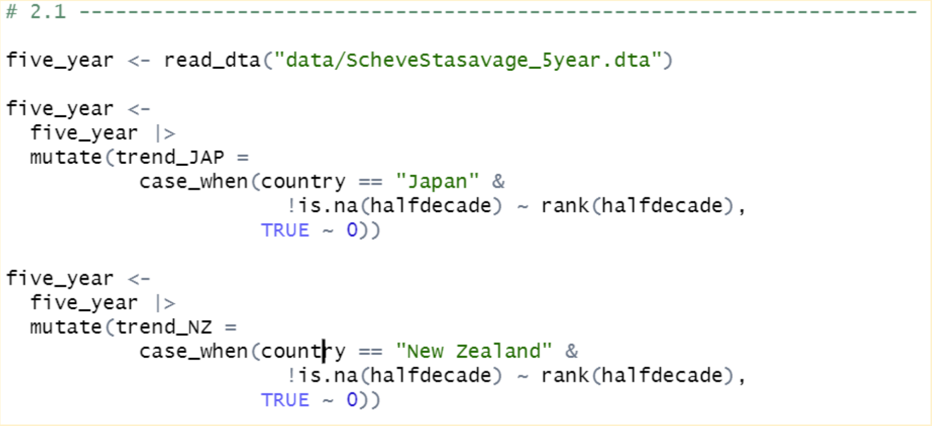
Problem 2

In this problem, you will use the 5 year version of the same dataset that the authors use in their main specifications. This dataset, `data/ScheveStasavage_5year.dta`, is in long form where each row is a [country] × [5 year interval] combination, where intervals start in 1816. Therefore, there are 5 times fewer rows than the annual data. The variables are summarized as follows.

Variable name	Description
`country`, `ccode`	country, and associated code
`year`	A character for the range of years included, e.g. the range `"1816-1820"`, or `"1996-2000"`.
`yearfirst`	A numeric variable for the first year in the year range (i.e., 1816 in the above example)
`yearlast`	A numeric variable for the last year in the year range (i.e., 1820 in the above example)
`topitax`	The value of `topitax` in the first year of the year range
`war2p`, `unisuffrage`, `leftexec`, `rgdppc`, `democracy`	The mean of the same variables within the year range. For example, `war2p` is the mean of the binary variable `war2p` from the annual dataset across the 5-year range.
`trend_US`, `trend_UK`, etc..	A variable for each country's country-specific linear time trend, of the format `trend_{country}`. The trends for Japan and New Zealand are not included.

2.1

One of the regression specifications you will replicate in 2.2 controls for *country-specific (linear) time trends*. We have included the variables for time trend controls in all but two countries: Japan and New Zealand. In this problem, create these variables to the dataset in a way that is suitable for 2.2, and provide the snippet or screenshot of your code in the yellow box.



```
# 2.1 ---------------------------------------------------------------------

five_year <- read_dta("data/ScheveStasavage_5year.dta")

five_year <-
  five_year |>
  mutate(trend_JAP =
           case_when(country == "Japan" &
                       !is.na(halfdecade) ~ rank(halfdecade),
                     TRUE ~ 0))

five_year <-
  five_year |>
  mutate(trend_NZ =
           case_when(country == "New Zealand" &
                       !is.na(halfdecade) ~ rank(halfdecade),
                     TRUE ~ 0))
```

2.2

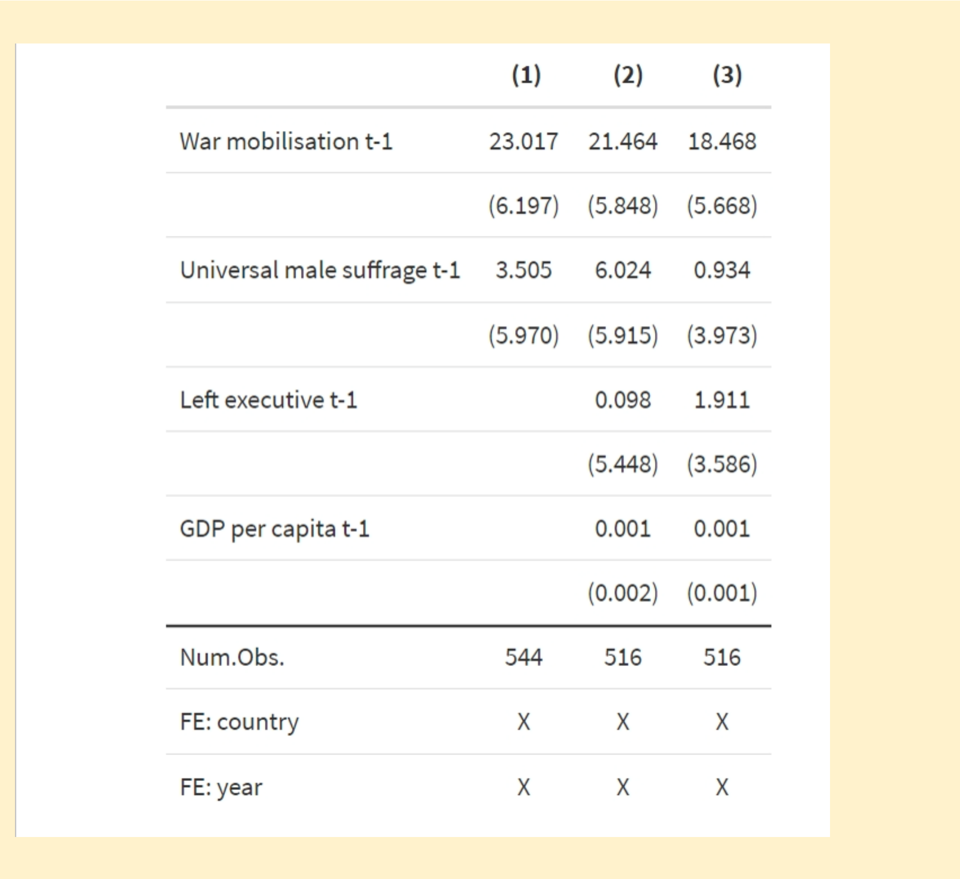
In one clear table, **replicate the coefficient estimates in the paper's Table 2, columns (1) and (2), and (3).** If done correctly, you should get the exact same coefficient estimates (your standard error estimates will differ somewhat from the paper). Note that you will need to "lag" the treatment variables, as the paper suggests.

Formatting notes:

- All variables displayed in the table should be renamed to be human-readable, as in the article. The `coef_rename` argument in `modelsummary` can rename variables for display.

- Column (3) in the regression table has controls for country-specific time trends. We ask that you do *not* manually type out each time trend covariate, and instead use more efficient functions that use regex (regular expression) and concatenation to use all countries. The hints have more detail. We strongly recommend using the `glue` function since it will be useful in many future cases.

- When putting column (3) in a table, notice that we do not report the coefficient on each time trend. The `coef_omit` argument in `modelsummary` can be used to omit groups of coefficients using regex.



	(1)	(2)	(3)
War mobilisation t-1	23.017	21.464	18.468
	(6.197)	(5.848)	(5.668)
Universal male suffrage t-1	3.505	6.024	0.934
	(5.970)	(5.915)	(3.973)
Left executive t-1		0.098	1.911
		(5.448)	(3.586)
GDP per capita t-1		0.001	0.001
		(0.002)	(0.001)
Num.Obs.	544	516	516
FE: country	X	X	X
FE: year	X	X	X

Problem 3

Standard two-way fixed effects only estimate the effect of the treatment in the one-term period of treatment. One might be interested in the lagged effect instead, e.g. the estimate of a war mobilization in 1945 on the tax rate at 1950, because the treatment may take time to shape the outcome.

A standard way to estimate a lagged effect is to create a treatment variable that is lagged so that it indicates past treatment. The main specification of your regression in problem 2.1 already uses a variable lagged by one term, but you could lag it two or three terms. For example, in the dataset, Austria experienced high war mobilization starting from 1915 until 1918. Then, lagged variables of war mobilization will look like the following in the data:

country	year	war	war, lagged by 1	war, lagged by 2
Austria	1914	0	0	0
Austria	1915	1	0	0
Austria	1916	1	1	0
Austria	1917	1	1	1

The opposite of lags, called *leads*, serves a different purpose. It serves as a placebo test, checking if the estimated effect of *future treatment* on an outcome is null. A treatment variable that is “lead-ed” has the following structure:

country	year	war	war, lead by 1	war, lead by 2
Austria	1913	0	0	0
Austria	1913	0	0	1
Austria	1914	0	1	1
Austria	1915	1	1	1
Austria	1915	1	1	1

The placebo test on leads is important because it serves as a partial test of the parallel trends assumption in the pre-period.

All of these lags and leads terms can be included in a single regression, and the coefficients on them can be read off as effects of these past and future treatments. In other words, one could run the regression `Y ~ lead2 + lead1 + lag1 + lag2` where `lag1` is the treatment lagged by 1, and the coefficient estimate on `lag1` is the estimated treatment effect of the lagged treatment.

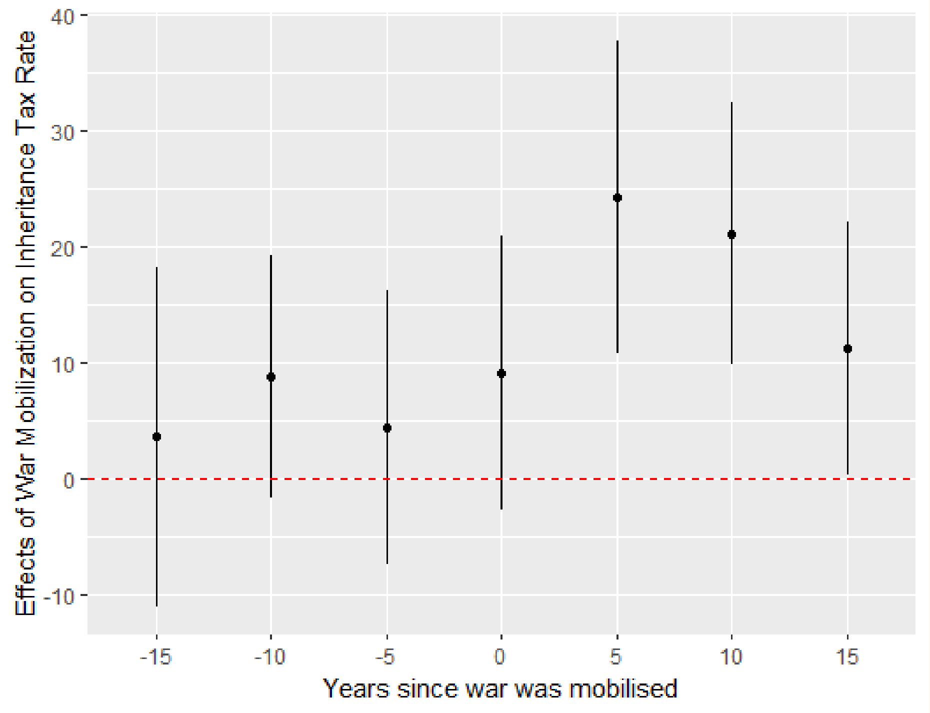
In writing, we can represent this as

$$Y_{it} = \eta_i + \theta_t + \sum_{\tau=0}^{2} \beta_{-\tau} W_{i,t-\tau} + \sum_{\tau=1}^{2} \beta_{+\tau} W_{i,t+\tau} + \boldsymbol{\gamma}\mathbf{X}_{it-1} + \varepsilon_{it},$$

following the equation *Mostly Harmless Econometrics* by Angrist and Pischke, where $W_{i,t-\tau}$ indicates treatment lagged by τ and $W_{i,t+\tau}$ indicates treatment lead-ed by τ.

A popular way to summarize the parallel trends and lagged effects is the *event study* plot (or *lags and leads* plot), which shows the lead effects on the left and the lagged effects on the right. As an example, see this figure from Rendleman and Yoder on the effect of the Earned Income Tax Credit on incumbent politician's approval in the contemporary U.S. (see end of this pset).

Using the Scheve and Stasavage 5 year dataset, **make an event study plot of the effects of war mobilization with up to 3-term leads and 3-term lags**, on top of specification (2) of the previous regression. The plot must be well-labelled, show a horizontal line at 0 indicating a null effect, and show the point estimates. Showing the 95 percent confidence intervals are optional (but recommended if you are familiar with standard errors or wait to 10/16 lecture; see hints). Then, **briefly summarize what we should take away** from the figure.



The effect of the leads or future treatment crosses over 0 in the left side of the plot and this indicates that parallel trends assumption in the pre-treatment period has not been violated and serves as a partial test of the parallel trends assumption.

Additionally, by looking at the patterns of the lags, we can see that the effect of war mobilization on inheritance tax was highest at 5 years after war mobilisation. The effect continued till 10 years after mobilisation, however with some reduction.

Appendix

1.1

- See the tutorial's examples and helppage for how to use and customize the graph. A particular option you may want to use is the `axis.lab.gap` argument to declutter the axis labels. A value of `c(10, 0)`, for example, only shows every 10th year in the graph.

2.1

- Lecture covered the main idea behind a time trend covariate. One difference here is that the data comes in 5 year intervals. However, the scaling of the values themselves do not matter for the estimation of the treatment effects as long as the values are linear in time. To replicate the paper's results exactly, you can see how the other trends are constructed and treat the years `1816-1820` gets a 1, a `1821-1825` gets a 2, and so forth.

2.2

- Use the `lag` function in dplyr to create a lagged version of the variable. `lag(var, 1)` looks for the value of the variable `var` that is 1 row previous to the row in question. Therefore, it is important to ensure that your rows **are ordered in sequential of year, and the `lag` function only looks for rows within the same country**. Use `group_by()` to ensure that all subsequent functions obey the grouping, and use `arrange()` to ensure that rows are ordered in sequential order by year.

- `arrange()` is a dplyr verb that will reorder rows by a variable. Its default is to reorder in increasing order, so that 1800 comes first, then 1805, then 1810, etc.. Knowing this default is important to make sure you implement leads and lags correctly, because `lead()` and `lag()` only work off the order of the rows in the data.

- To remove irrelevant summary statistics, recall we use the argument `gof_map` in `modelsummary`. `gof_map = c("nobs", "FE: country", "FE: year")` will limit the display to sample size and indicators for the presence of Fixed Effects, which is what we want. You can also get the same result by specifying regular expressions of what *not* to show by `gof_omit`.

- There is one time-trend for each country, and we control for them all in specification (3). Writing out 19 variables in a formula is rather tedious and prone to typos. A shorter and more robust way to code this would be to identify all the variables that start with `trend_` and paste them together. One way to do this would be to use the following functions in succession:

  - `colnames` gets the column names of a tibble and outputs a vector

  - `stringr::str_subset` takes a character vector and only returns the values that match a user-specified character given in the `pattern` argument. For example, `colnames(data) |> str_subset(pattern = "gdp")` would extract a vector of the variable names in `data` that include the characters `"gdp"`.

  - `stringr::str_c` or `paste` takes a character vector and combines them together. When the `collapse` argument is specified, it concatenates the values in a single string. For example, try: `paste(c("a", "b", "c"), collapse = " + ")`.

  - **`glue::glue`** (recommended more than the base-R `paste`) is a convenient function to combine different string variables together. See https://glue.tidyverse.org/. For example if you have an object `Xs` that is `"X1 + X2 + X3"` and an object `Trends` that is `"trend_US + trend_UK + trend_IT”`, then `glue("Y ~ {Xs} + {Trends}")` will give `"Y ~ X1 + X2 + X3 + trend_US + trend_UK + trend_IT"`. R will treat the contents of the curly braces as objects and expand them out.

  - Note that the fixed effects are part of the formula, and also must be included in the string.

  - Coerce the text into a formula class by `as.formula()`. Such a formula specification can be directly used as a formula for `lm` or `feols`.

- These steps feel like overkill at first, but as you deal with more complex data in your own projects, these small steps become important to keep your code concise.

- To get the standard errors to exactly match the paper, you can add `ssc = ssc(fixef.K = "full")` in the feols regression. feols automatically clusters standard errors, but the exact specifications of clustering varies by software, e.g. Stata vs. R. Setting `fixef.K = "full"` will match the Stata version (see https://cran.r-project.org/web/packages/fixest/vignettes/standard_errors.html).

3

- The leads function is `lead()` and used the same way as the lag.
- The function `tidy()` will become very useful as you translate the regression output into a object that can be used by `ggplot()`. You will then want to recode or relevel the terms of the regression so that they are ordered from far leads to far lags. You can use `recode` to turn them into numbers like the Rendleman and Yoder example, or relevel them as factors.
- If you feel comfortable with standard errors, you can put 95% confidence intervals on your coefficients as well. The `tidy()` function provides a column for standard errors (`std.error`), so the 95% CI is roughly +/- 2 standard errors away from the coefficient estimate. The geom to use for errorbars is `geom_errorbar()` with the aesthetics `ymin` and `ymax` to set the bars. See examples in the ggplot help page.
- For discussion on lags and leads, I recommend *Mostly Harmless Econometrics* by Angrist and Pischke, chapter 5 (especially equation 5.2.6).
- For more advanced treatment of difference in difference methods, see for example “How to Make Causal Inferences with Time-Series Cross-Sectional Data under Selection on Observables” by Blackwell and Glynn (2018), and “Matching Methods for Causal Inference with Time-Series Cross-Sectional Data”, by Imai, Kim, and Wang (2021).

Rendelman and Yoder Example:

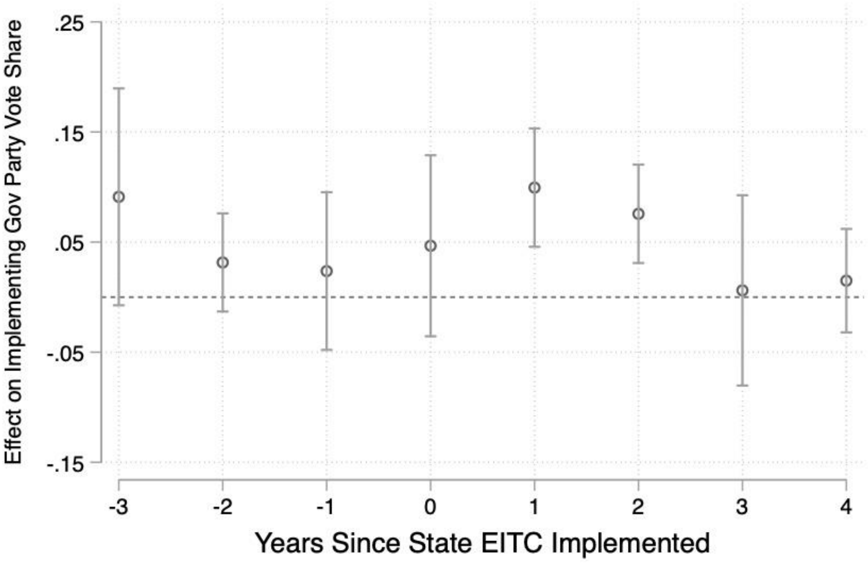


Figure 4 – Dynamic Effect of EITC on Implementing Governor Party's Vote Share The plot models the dynamic effect of the EITC program on the implementing Governor party's vote share. Year = 0 is the year the state adopted the EITC program. Vertical lines indicate 95% confidence intervals using robust standard errors clustered by state. The plot shows that EITC programs might lead to an increase in the implementing party's vote share in the short term, but the effect dissipates by three years after the program's introduction.