

Difference-in-Differences

Alex Alekseev

```
library(tidyverse)
library(fixest)
library(modelsummary)
options(modelsummary_factory_default = 'kableExtra')

library(lmtest)
library(sandwich)
```

Intro

In this section, we will be working with the data from [Kessler and Roth \(2014\)](#), which studies the rate at which people sign up to be organ donors. In the United States, people are not signed up to be organ donors by default. In most states, you are assumed to not be an organ donor. When you sign up for a driver's license, you can choose to opt in to the organ donation program. Outside of the opt-in and opt-out varieties of organ donation, there's also "active choice." Under active choice, when you sign up for a driver's license, you are asked to choose whether or not to be a donor. You can choose yes or no, but now the "no" option is actively checking the "no" box rather than skipping the question entirely as you can with opt-in approaches. Some policymakers have been advocating for active choice, with a goal of increasing donation rates, and active choice is the way things work in many states. In July 2011, the state of California switched from opt-in to active choice. Kessler and Roth compare California against the twenty-five states that either have opt-in or a verbally given question with no fixed response.

Data

Let's load the `organ_donations.csv` dataset.

```
df <- read_csv("path/to/organ_donations.csv")
```

Our dependent variable is `Rate`. We observe multiple states over multiple quarters. The treated state is California, and the post-treatment period starts with "Q32011" (period 4). We will create the indicator variables for the treated state and post-treatment period to simplify further calculations.

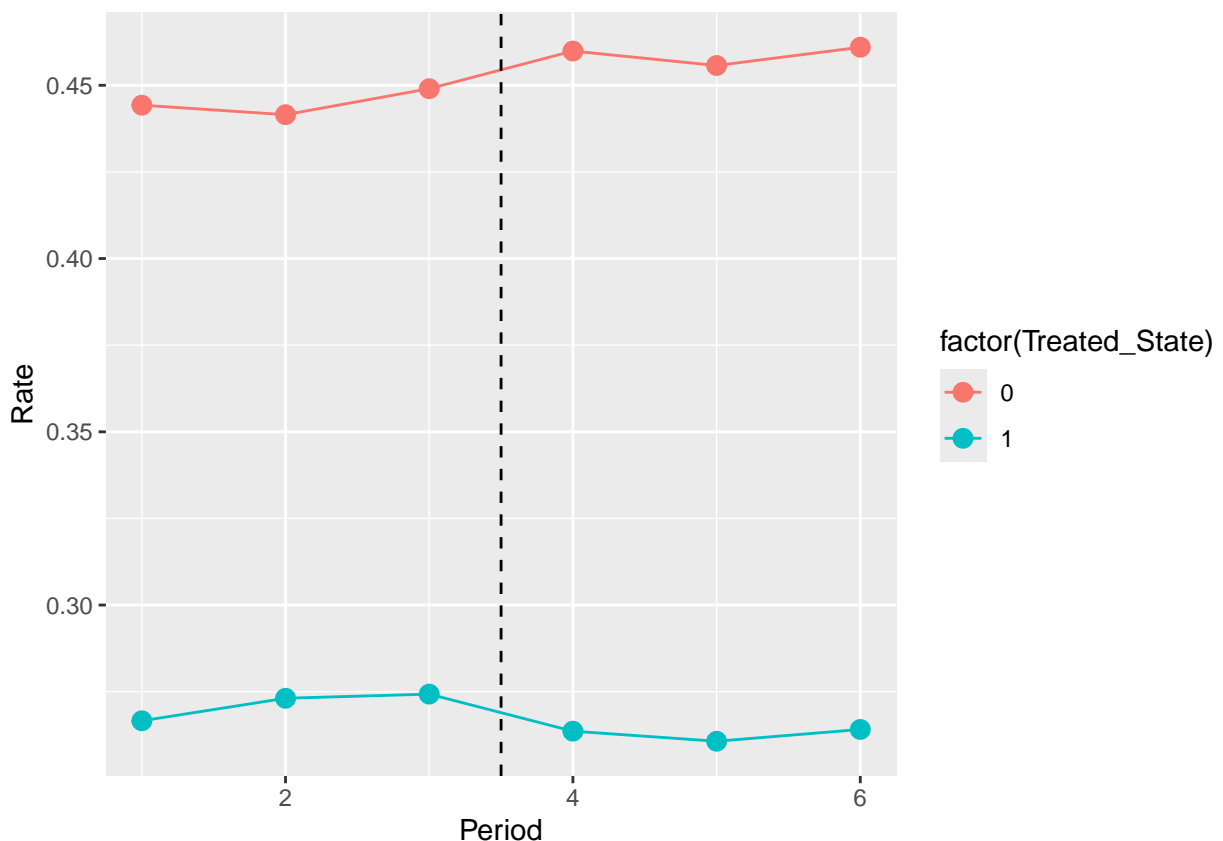
```
df <-
  df |>
  mutate(Treated_State = 1*(State == "California")) |>
```

```
mutate(Post_Treatment = 1*(Period >= 4))
```

Graph of Raw Data

First, let's examine visually how the donations rates evolve over time in the treated state and all other states. We will compute the average donation rates by period for California and all other states combined. The dashed line separates the pre- and post-treatment periods.

```
df |>
  group_by(Period, Treated_State) |>
  summarize(Rate = mean(Rate)) |>
  ggplot(
    aes(Period, Rate, color = factor(Treated_State), group = Treated_State)
  ) +
  geom_point(size = 3) +
  geom_line() +
  geom_vline(xintercept = 3.5, linetype = "dashed")
```



The graph shows that California experienced a drop in its donation rate after the policy was enacted. Other states experienced an increase in their rates. In the absence of treatment, if California followed the trend of other states, its donation rate should have also increased, not dropped. This suggests that the treatment had a negative effect on the donation rate.

Diff-in-Diff Table

Let's aggregate the data even further and make a two-by-two DD table with two periods (pre- and post-treatment) and two groups (California and everyone else). We have to use pivoting to get a two-by-two table.

```
dd_table <-  
  df |>  
  group_by(Post_Treatment, Treated_State) |>  
  summarize(Rate = mean(Rate)) |>  
  pivot_wider(  
    id_cols = Treated_State  
    , names_from = Post_Treatment  
    , names_prefix = "Period_"  
    , values_from = Rate  
  )
```

dd_table

	Treated_State	Period_0	Period_1
	0	0.4449308	0.4588564
	1	0.2713333	0.2628000

This is a DD table that we worked with during the lecture. Using this table, we can compute the DD effect by hand. The code below computes the differences along each margin and the number in the bottom right is the DD effect. The row-wise operations are not as straightforward as column-wise operations, which is why we are using pivoting.

```
dd_table |>  
  mutate(diff_time = Period_1 - Period_0) |>  
  pivot_longer(cols = -Treated_State, names_to = "Time") |>  
  pivot_wider(  
    id_cols = Time  
    , names_from = Treated_State  
    , names_prefix = "Treated_State_"  
  ) |>  
  mutate(diff_group = Treated_State_1 - Treated_State_0) |>  
  pivot_longer(cols = -Time, names_to = "Group") |>  
  pivot_wider(  
    id_cols = Group  
    , names_from = Time  
  )
```

Group	Period_0	Period_1	diff_time
Treated_State_0	0.4449308	0.4588564	0.0139256
Treated_State_1	0.2713333	0.2628000	-0.0085333
diff_group	-0.1735974	-0.1960564	-0.0224590

```
dd_table |>
  mutate(diff_time = Period_1 - Period_0) |>
  mutate(dd = diff(diff_time))
```

	Treated_State	Period_0	Period_1	diff_time	dd
	0	0.4449308	0.4588564	0.0139256	-0.022459
	1	0.2713333	0.2628000	-0.0085333	-0.022459

We can also simply compute the DD effect itself.

```
dd_tab2 <-
  dd_table |>
  mutate(diff_time = Period_1 - Period_0) |>
  mutate(dd = diff(diff_time))
```

```
dd_eff <- dd_tab2$dd[1]
dd_eff
```

```
## [1] -0.02245897
```

As expected, the effect is negative. Specifically, the treatment decreased California's donation rate by about 2.25% from the pre-treatment to post-treatment period.

Counterfactual

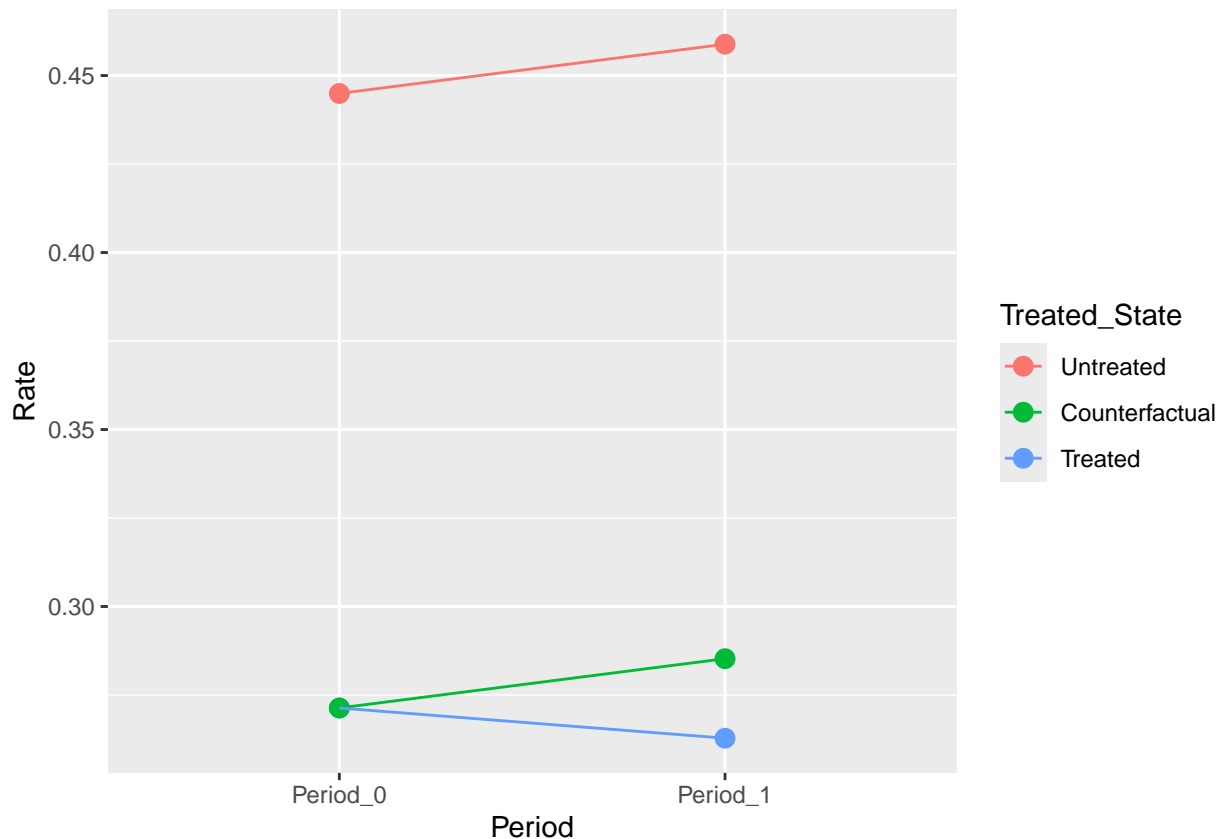
Working with the aggregated two-by-two table, we can also create a graph showing the counterfactual prediction and the DD effect. Notice again that working with rows is a bit tricky. We first create another row for the counterfactual California and then plot the data. As expected, the counterfactual line is parallel to the untreated group by construction.

```
dd_table |>
  add_row(
    Treated_State = 0.5
    , Period_0 = dd_table$Period_0[dd_table$Treated_State == 1]
    , Period_1 =
      dd_table$Period_1[dd_table$Treated_State == 0] -
      dd_table$Period_0[dd_table$Treated_State == 0] +
      dd_table$Period_0[dd_table$Treated_State == 1]
  ) |>
  mutate(
    Treated_State =
      factor(
        Treated_State
        , labels = c("Untreated", "Counterfactual", "Treated")
      )
  ) |>
  pivot_longer(
    cols = c(Period_0, Period_1)
```

```

, values_to = "Rate"
, names_to = "Period"
) |>
ggplot(aes(Period, Rate, color = Treated_State, group = Treated_State)) +
geom_point(size = 3) +
geom_line()

```



Regression

There are several options for running a DD regression. The simplest would be to split all the dates into pre- and post-treatment and split all the groups into untreated and treated, just like we did with the two-by-two table. This amounts to running a simple regression with two binary variables and an interaction term.

$$Y = \alpha + \lambda \mathbb{I}(i = T) + \tau \mathbb{I}(t = 1) + \delta \mathbb{I}(i = T) \mathbb{I}(t = 1) + \epsilon,$$

We can use the `*` shortcut, which means that we want the main effects plus the interaction. It is a standard practice to cluster the standard errors by group when running a DD regression, hence we will be modifying the default standard errors using the `coefest` function from the `lmtest` package.

```

reg <- lm(Rate ~ Treated_State*Post_Treatment, data = df)
coefest(reg, vcov = vcovCL(reg, type = "HC1", cluster = ~ State))

```

```

##
## t test of coefficients:

```

```
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.4449308  0.0313963 14.1715 < 2.2e-16 ***
## Treated_State     -0.1735974  0.0313963 -5.5292 1.303e-07 ***
## Post_Treatment      0.0139256  0.0060727  2.2931 0.023159 *
## Treated_State:Post_Treatment -0.0224590  0.0060727 -3.6983 0.000299 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The coefficient on `Treated_State:Post_Treatment` is our DD estimate. It is exactly the same number as we got by hand. Notice that the coefficient estimates allow us to recreate the two-by-two table. For example, the constant term corresponds to the untreated group in the pre-treatment period. If we add the coefficient on `Treated_State` to that constant term, we will get the treated group in the pre-treatment period, and so on.

Another option would be to manually include all the groups and time periods as factor variables (fixed effects) plus the interaction term. The model to be estimated is

$$Y = \alpha_g + \alpha_t + \delta D + \epsilon.$$

This, however, does not change the DD estimate. We do get a lot of fixed effects estimates, which we do not really care about. I am omitting the output of the command.

```
reg3 <-
  lm(
    Rate ~ factor(State) + factor(Period) + Treated_State:Post_Treatment
    , data = df
  )

coeftest(reg3, vcov = vcovCL(reg3, type = "HC1", cluster = ~ State))
```

A better way to include fixed effects is to use a specialized function. We will be using the `feols` function from the `fixest` package. This function automatically clusters the standard errors by group.

```
reg2 <-
  feols(
    Rate ~ Treated_State:Post_Treatment | State + Period
    , data = df
  )
summary(reg2)
```

```
## OLS estimation, Dep. Var.: Rate
## Observations: 162
## Fixed-effects: State: 27, Period: 6
## Standard-errors: Clustered (State)
##              Estimate Std. Error t value Pr(>|t|)
## Treated_State:Post_Treatment -0.022459  0.006131 -3.66304 0.0011185 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.021982      Adj. R2: 0.974196
##              Within R2: 0.009221
```

	OLS	OLS 2	FE
Treated_State:Post_Treatment	−0.022 (0.006)	−0.022 (0.007)	−0.022 (0.006)
Num.Obs.	162	162	162
R2	0.054	0.979	0.979
R2 Within			0.009
R2 Within Adj.			0.002
Std.Errors	by: State	by: State	by: State
FE: Period			X
FE: State			X

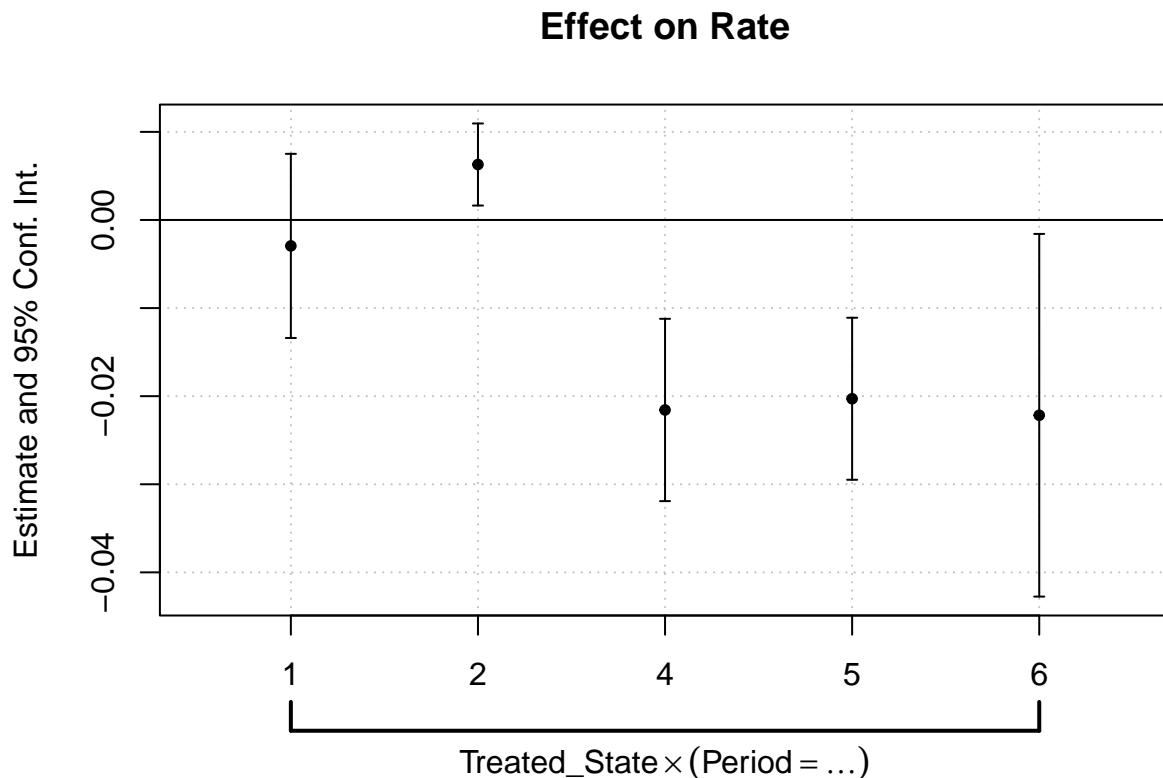
Putting the results from all three regression together, we see that the estimate of the effect is the same, although the standard errors may vary depending on the specification.

```
msummary(
  list("OLS" = reg, "OLS 2" = reg3, "FE" = reg2)
  , vcov = list(~ State, ~ State, NULL)
  , coef_map = "Treated_State:Post_Treatment"
  , gof_omit = c("AIC|BIC|RMSE|R2 Adj.|Log.Lik.")
)
```

The `feols` function allows for an easy way to compute and visualize the dynamic effects. Instead of interacting the treated group with a single post-treatment indicator, we can interact it with all the dates individually. All of the effects are computed relative to the last pre-treatment date (period 3).

```
reg_dyn <- feols(
  Rate ~ i(Period, Treated_State, ref = 3) | State + Period
  , data = df
)

coefplot(reg_dyn)
```



Starting from period 4 (first post-treatment date), the effect on rate remains negative (relative to Period 3) and does not seem to change much over time.

Robustness Checks

We can run a few placebo tests in the pre-treatment period and see if picking a fake treatment date produces a significant effect (it should not). In other words, we pretend that, say, Period 2, was the start of the post-treatment period and compute the effect. Remember to subset your dataset to the pre-treatment period only.

```
reg_placebo1 <-
  feols(
    Rate ~ Treated_State*(Period >= 2) | State + Period
    , data = filter(df, Period <= 3)
  )

reg_placebo2 <- feols(
  Rate ~ Treated_State*(Period >= 3) | State + Period
  , data = filter(df, Period <= 3)
)

msummary(
  list("Placebo 1" = reg_placebo1, "Placebo 2" = reg_placebo2)
  , gof_omit = c("AIC|BIC|RMSE|R2 Adj.|Log.Lik.")
)
```

None of the fake periods results in a significant effect, which gives us more confidence in our findings.

	Placebo 1	Placebo 2
Treated_State \times Period ≥ 2	0.006 (0.005)	
Treated_State \times Period ≥ 3		-0.002 (0.003)
Num.Obs.	81	81
R2	0.994	0.994
R2 Within	0.002	0.000
R2 Within Adj.	-0.018	-0.019
Std.Errors	by: State	by: State
FE: State	X	X
FE: Period	X	X

We can also do a test for prior trends. The test shows whether there are any systematic differences in the pre-treatment time trends between the treated and untreated groups. We will estimate a simple model with a linear time trend and an interaction of the time trend with the `Treated_State` indicator (which allows the time trend to be different in the treated group), while allowing for group fixed effects (but not time fixed effects).

$$Y = \alpha_g + \beta_1 t + \beta_2 t \times \text{Group} + \epsilon$$

As we see in the table below, the time trends do not differ significantly between the groups.

```
reg_trends <- feols(
  Rate ~ Period + Period:Treated_State | State
  , data = filter(df, Period <= 3)
)
summary(reg_trends)
```

```
## OLS estimation, Dep. Var.: Rate
## Observations: 81
## Fixed-effects: State: 27
## Standard-errors: Clustered (State)
##
##              Estimate Std. Error  t value Pr(>|t|)
## Period          0.002379   0.002493  0.954120  0.34881
## Period:Treated_State 0.001471   0.002493  0.590058  0.56025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.01259      Adj. R2: 0.990071
##
##              Within R2: 0.024607
```