Problem set 5: Do police reduce car thefts?

Answer key - PMAP 8521, Spring 2021

March 22, 2021

Contents

In 2004, Rafael Di Tella and Ernesto Schargrodsky published a study that analyzed the effect of increased police presence on crime. You looked at this study previously in your threats to validity assignment. To measure this effect, Di Tella and Schargrodsky leveraged a quasi-experiment. Following a synagogue bombing in Buenos Aires, Argentina on July 18, 1994, extra municipal police were assigned to protect synagogues around the city. The increase of police patrols on some city blocks, but not others, means that there is arguably a treatment group and control group for increased police presence, which Di Tella and Schargrodsky used to measure the effect of extra police on car thefts.

The dataset I've provided (`MonthlyPanel.dta`) is a Stata data file nearly 10,000 observations. It comes directly from Di Tella and Schargrodsky's data appendix available at their study's *AER* webpage. This is non-experimental data that includes counts of car thefts for every city block in Buenos Aires from April to December 1994. There are 12 variables:

- `observ` (we'll rename to `block`): The ID number of the block
- `barrio`: The barrio (neighborhood) for the block
- `calle`: The street for the block
- `altura`: The street number
- `institu1` (we'll rename to `same_block`): Indicator variable marking if there's a Jewish institution on the block (1 if yes, 0 if no)
- `institu3`: Indicator variable marking if there's a Jewish institution within one block (1 if yes, 0 if no)
- `distanci` (we'll rename to `distance`): Distance to the nearest Jewish institution, measured in blocks
- `edpub`: Indicator variable marking if there's an educational building or embassy on the block (1 if yes, 0 if no)
- `estserv`: Indicator variable marking if there's a gas station on the block (1 if yes, 0 if no)
- `banco`: Indicator variable marking if there's a bank on the block (1 if yes, 0 if no)
- `totrob` (we'll rename to `car_theft`): Total number of car robberies
- `mes` (we'll rename to `month`): Month

```r
library(tidyverse)      # For ggplot, %>%, mutate, filter, group_by, and friends
library(haven)          # For loading data from Stata
library(broom)          # For showing models as data frames
library(fixest)         # For fast, nice, fixed effects regression
library(modelsummary)   # For side-by-side regression tables

# This turns off this message that appears whenever you use summarize():
# `summarise()` ungrouping output (override with `.groups` argument)
options(dplyr.summarise.inform = FALSE)

# Load terror data
terror <- read_stata("data/MonthlyPanel.dta") %>%
  # The attack happened on July 18. The authors omitted data from July 19-31, so
  # all July observations are from before the attack. Make a new indicator
  # variable `after` to mark if the row is from before or after the attack
  mutate(after = mes > 7) %>%
  # There are some weird months in the data like 73. Filter out anything > 12
  filter(mes <= 12) %>%
  # Rename some columns to be more readable
  rename(same_block = institu1,
         distance = distanci,
         car_theft = totrob,
         month = mes,
         block = observ) %>%
  # Create indicator variables for the distance of each block to a synagogue
  mutate(one_block_away = ifelse(distance == 1, 1, 0),
         two_blocks_away = ifelse(distance == 2, 1, 0),
         more_than_two_away = ifelse(distance > 2, 1, 0)) %>%
  # Make these factors/categories
  mutate(block = as.factor(block),
         month = as.factor(month),
         same_block_factor = as.factor(same_block))
```

1. Research design

Imagine you went out and collected data on the presence of police in each city, and the amount of crime in each city, and found a positive relationship. Does this mean police *cause* crime? Explain.

No! This would not reflect any causal relationship—this is a situation where you can legitimately say that correlation is not causation. Police might be sent to areas with higher crime, or police might cause additional crime (reverse causality). The main issue here is selection—the presence of police is not randomly assigned across the city.

Di Tella and Ernesto Schargrodsky explore this question with a difference-in-difference design. They collected data on both the presence of police and car robberies in Buenos Aires city blocks both before and after the attack. Their interest is in seeing whether the extra police reduced the amount of car theft. **How is this data suitable for a diff-in-diff design? What would we be comparing here? Be specific about the pre/post treatment/control groups.**

A diff-in-diff design works here because we have data before and after an event (so we have pre and post time periods) *and* we have blocks that can arguably be considered treatment (blocks with synagogues with additional police) and control (blocks without synagogues with no additional police and that were unaffected by the attack). We would compare the average number of crimes in treatment and control blocks both before and after the attack.

Why does it help the researchers that the police were dispatched to certain blocks *because of terrorist attacks?*

Because we're using observational data, there are unobserved factors in the data that help assign police to certain areas of the city, and we can't measure or know what those unobserved confounders are. Because police are sent out in response to a single attack, we know exactly why extra police were dispatched: to protect synagogues. There's no other unobserved confounding reason for the extra deployment, which helps us identify the arrow between police presence and crime.

2. Trends

One of the most crucial assumptions for difference-in-differences designs is the idea that the trends in the treatment and control groups need to be parallel prior to the intervention or program. **Why?**

We must assume parallel trends with diff-in-diff designs because we have to have a credible counterfactual. If the trends for both treatment and control groups are similar, we can make a strong argument for what *would have happened* in the treatment group had they not undergone the treatment, since they would have just followed the same trend as the control group.

Create a plot that shows the average number of car thefts per month for blocks with synagogues and blocks without (Hints: it'll be easiest if you make a smaller dataset using `group_by()` and `summarize()` and then plot that smaller dataset with `ggplot()` Make sure you group by `month` and `same_block_factor`. Add `group = same_block_factor` as an aesthetic so the line goes across the categorical months on the x-axis). Add a vertical line (`geom_vline(xintercept = "7")`) in the month where the terror attack happened.

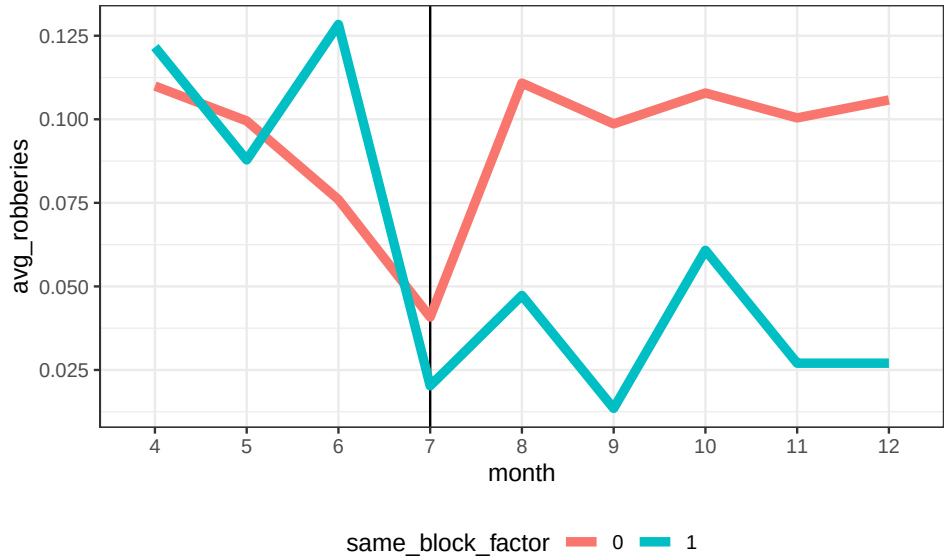
What would you say about the parallel trends assumption here? Does it hold up? Maybe? Maybe not?

These trends are kind of parallel, I guess. In April, both types of blocks had high levels of car thefts, and in May they both decreased at roughly the same rate. In June, for whatever reason the blocks with synagogues saw a huge spike in car thefts and there was no corresponding spike in the control group, but both groups were back on track and parallel in July. After the attack in July, car thefts went back up to high levels in the control group, but remained low in the treatment group, likely because of the increased police presence.

Does the assumption hold up? Sure. It's not perfect, and there's a weird June spike, but the study was good enough to be published in the *American Economic Review* (the top journal for economics research), so, sure.

```
terror_trends <- terror %>%
  group_by(month, same_block_factor) %>%
  summarize(avg_robberies = mean(car_theft))

ggplot(terror_trends, aes(x = month, y = avg_robberies,
                          color = same_block_factor, group = same_block_factor)) +
  geom_vline(xintercept = "7") +
  geom_line(size = 2) +
  theme_bw() +
  theme(legend.position = "bottom")
```



3. Difference-in-differences by hand-ish

Calculate the average number of car thefts in the treatment and control groups before and after the attack. (Hint: group by `same_block` and `after` and find the average of `car_theft`.)

```
# Calculate average of car_theft across same_block and after
terror_diff_diff_means <- terror %>%
  group_by(same_block, after) %>%
  summarize(avg_thefts = mean(car_theft))

cell_A <- terror_diff_diff_means %>%
  filter(same_block == FALSE, after == FALSE) %>%
  pull(avg_thefts)

cell_B <- terror_diff_diff_means %>%
  filter(same_block == FALSE, after == TRUE) %>%
  pull(avg_thefts)

cell_C <- terror_diff_diff_means %>%
  filter(same_block == TRUE, after == FALSE) %>%
  pull(avg_thefts)

cell_D <- terror_diff_diff_means %>%
  filter(same_block == TRUE, after == TRUE) %>%
  pull(avg_thefts)

diff_in_diff <- (cell_D - cell_C) - (cell_B - cell_A)
```

Calculate the difference-in-difference estimate given these numbers.

	Before attack	After attack	Difference
Block without synagogue	0.082	0.105	0.023
Block with synagogue	0.09	0.035	-0.054
Difference	0.008	-0.07	-0.078

Answer these questions (you don't have to write your answers in list form—a paragraph is fine:

- **How did car thefts change from before-to-after in blocks *without* synagogues?**: Thefts increased slightly in the control blocks (by 0.023 on average)
- **How did car thefts change from before-to-after in blocks *with* synagogues?**: But thefts *decreased* even more in the treatment/synagogue clocks (by -0.054 on average)
- **What's the difference-in-differences?**: The difference-in-differences here is -0.078
- **What does that mean? Interpret the finding.**: This means that thefts decreased *because of* the increased police presence in the treatment blocks.

4. Difference-in-differences with regular OLS

Run a regression model to find the diff-in-diff estimate of the effect of the increased police presence (`after`) on car thefts (`car_theft`) (hint: remember that you'll be using an interaction term).

```
model_simple <- lm(car_theft ~ same_block*after, data = terror)

tidy(model_simple)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.082	0.004	19.557	0.000
same_block	0.008	0.020	0.392	0.695
afterTRUE	0.023	0.006	4.135	0.000
same_block:afterTRUE	-0.078	0.027	-2.847	0.004

```
glance(model_simple)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.004	0.003	0.242	9.33	0	3	13.1	-16.2	18.7	460	7880	7884

How does this value compare with what you found in part 3 earlier? What is the advantage of doing this instead of making a table?

Calculating the diff-in-diff estimate with regression instead of a table is (1) easier, (2) allows for additional control variables, and (3) provides other statistics like standard errors and confidence intervals.

5. Difference-in-differences with fixed effects OLS

The diff-in-diff coefficient you found in part 4 is accurate, but the standard errors and R^2 are wrong (run `glance()` on your model object to see how tiny the R^2 is)! This is because of a host of mathy reasons, but also because of the DAG. The effect of increased police presence is confounded by both month and block, but all we've really adjusted for binary before/after (for month) and binary synagogue/no synagogue (for block). By reducing these confounders to just binary variables, we lose a lot of the variation across months and blocks.

To fix this, run a diff-in-diff model that includes two additional control variables: `block + month`.

> ***Warning***: this will be *incredibly* slow! There are 876 blocks and `nrow(distinct(terror, month))` months, and R is finding estimates for each block and month, and the math to do that is complex. Every time you knit this document, R will rerun the model, which takes 5-10 seconds, and the delay when knitting can be annoying. If you want to speed this up across knitting sessions, add the option `cache=TRUE` to the chunk options for this chunk. R will store the results in a temporary file and won't re-run the model if the data hasn't changed.

Don't use `tidy` to view the results. You'll get a table with almost 900 rows and it'll take up pages and pages of your knitted document. If you really want to see the results, filter out the block and month rows (like this:).

```
model_fe_slow <- lm(car_theft ~ same_block*after + block + month, data = terror)

# Show results, but hide all the block and month coefficients
tidy(model_fe_slow) %>%
  filter(!str_starts(term, "block"),
         !str_starts(term, "month"))
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.016	0.077	0.204	0.839
same_block	0.043	0.109	0.394	0.694
afterTRUE	-0.005	0.011	-0.427	0.669
same_block:afterTRUE	-0.078	0.026	-2.992	0.003

```
glance(model_fe_slow)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.198	0.097	0.23	1.96	0	884	871	31	6209	370	6999	7884

That slowness is miserable. You can get around that by using a different function for OLS that has built-in support for fixed effects (or indicator variables). The `feols()` (fixed-effects OLS) function from the **fixest** package lets you include indicator variables in regression in a more sophisticated way. The math is lighting fast, and the coefficients for each block and year are hidden by default (though you can still see them if you really want).

The syntax for `feols()` is the same as `lm()`, but with a slight change to accommodate the fixed effects. Use the `|` character to specify a section of the formula that contains the fixed effects:

```
model_name <- feols(car_theft ~ same_block*after | block + month,
                    data = terror)
```

One more cool thing that `feols()` can do that normal `lm()` can't is provide robust standard errors. There is systematic variation within blocks and across time, and we can mathematically account for that variation

in the standard errors of the regression. (If you've ever used Stata you do this with `reg y x, robust`). If you ever want to use robust and/or clustered standard errors with regular OLS regression in R, check out the `lm_robust()` function in the **estimatr** package. With `feols()`, you can add an argument to `tidy()` to get the robust standard errors.

```
# Stata's default robust SE algorithm is called "Huber-White standard errors",
# and we can get those same numbers here. Look at the documentation for
# summary.fixest() for more robustness and clustering options
tidy(model_name, se = "white")
```

Phew. Now that you know about `feols()` and robust standard errors, build a model that finds the diff-in-diff effect that includes fixed effects for block and month. Show the results with `tidy()` using Huber-White standard errors.

```
model_fe_a <- feols(car_theft ~ same_block*after | block + month,
                    data = terror)

tidy(model_fe_a, se = "white")
```

term	estimate	std.error	statistic	p.value
same_block:afterTRUE	-0.078	0.022	-3.46	0.001

In the original study, the authors also considered the effect of two other treatment variables. Maybe the extra police presence in blocks with synagogues reduced car thefts not just for those blocks, but areas 1 block away or 2 blocks away.

Run two more models. In the first, keep the `same_block*after` interaction term and add another diff-in-diff interaction for `one_block_away*after`. In the second, keep the same block and one block interaction terms and add one more diff-in-diff interaction for `two_blocks_away*after`

```
model_fe_b <- feols(car_theft ~ same_block*after +
                      one_block_away*after | block + month,
                    data = terror)
tidy(model_fe_b)
```

term	estimate	std.error	statistic	p.value
same_block:afterTRUE	-0.080	0.024	-3.39	0.001
afterTRUE:one_block_away	-0.013	0.015	-0.91	0.363

```
model_fe_c <- feols(car_theft ~ same_block*after + one_block_away*after +
                      two_blocks_away*after | block + month,
                    data = terror)
tidy(model_fe_c)
```

term	estimate	std.error	statistic	p.value
same_block:afterTRUE	-0.081	0.024	-3.375	0.001
afterTRUE:one_block_away	-0.014	0.015	-0.927	0.354
afterTRUE:two_blocks_away	-0.002	0.012	-0.176	0.860

Recreate columns A, B, and C from Table 3 from the original article with `modelsummary()`. You'll need to show the results from your three `feols()` models (with one interaction term, with two interactions, and

with three interactions). You can tell the table to show robust standard errors like the authors did in their original study by including the `se = "white"` argument, and you can control how many digits are used with the `fmt` (format) argument (the original article used 5 decimal points, so you can too). You can add significance stars by including `stars = TRUE`.

This is pretty much identical to the published Table 3! Magical!

```
# By default, modelsummary() shows a ton of goodness-of-fit (GOF) statistics
# like R2 at the bottom of the table. Too many. You can control what goes there
# by using the gof_map argument. This argument takes a data frame of GOF
# statistics as its input. It's best to make a copy of the default GOF mapping
# dataset and then modify the omit column to determine what things get omitted.
# Here, I make a dataset called gof that is based on the default set of GOF
# stats (gof_map). Everything is omitted except N (nobs or number of obs) and R2
gof <- gof_map %>%
  # Make the omit column omit everything!
  mutate(omit = TRUE) %>%
  # ...except these statistics
  mutate(omit = !(raw %in% c("nobs", "r.squared")))

modelsummary(list("(A)" = model_fe_a, "(B)" = model_fe_b, "(C)" = model_fe_c),
             se = "white", fmt = "%.5f", stars = TRUE, gof_map = gof)
```

	(A)	(B)	(C)
same_block × afterTRUE	-0.07753***	-0.08007***	-0.08080***
	(0.02244)	(0.02257)	(0.02294)
afterTRUE × one_block_away		-0.01326	-0.01399
		(0.01386)	(0.01447)
afterTRUE × two_blocks_away			-0.00218
			(0.01232)
Num.Obs.	7884	7884	7884
R2	0.198	0.198	0.198

* p < 0.1, ** p < 0.05, *** p < 0.01

Answer these questions: (again, you don't have to keep this in list form when you answer):

- **Does having extra police reduce thefts on the same block? Is the effect significant?**: Having extra police seems to significantly reduce car thefts on the same block across all three models.
- **Does having extra police reduce thefts one block away? Is the effect significant?** Extra police does not seem to significantly reduce car thefts on blocks one block away.
- **Does having extra police reduce thefts two blocks away Is the effect significant?** And extra police also doesn't significantly reduce car thefts two blocks away.

6. Translate results to something more interpretable

According to the third model, having additional police on a block caused a reduction of 0.081 car thefts per month on average. What the heck does that even mean though? This whole outcome variable is weird anyway—it's the average number of thefts per block per month, and most block-months have 0 thefts. Having a number like 0.081 doesn't quite represent the proportion of crime or anything logically interpretable or anything. It's a little hard to talk about.

To fix this, we can talk about percent changes instead. Recall from past classes (like microeconomics or GRE prep questions) that you can calculate the percent change (or growth) between two numbers with this formula:

$$\text{percent change} = \frac{\text{new} - \text{old}}{\text{old}}$$

You can remember this as **NOO**, for **n**ew minus **o**ld divided by **o**ld. With treatment and outcome groups, you can find the percent change because of a program or policy by using treatment as "new" and outcome as "old".

Imagine if after some program, the treatment group had an outcome of 3 while the control group had an outcome of 6. The percent change in outcome because of the causal effect of the program is $\frac{3-6}{6}$, or -0.5:

```
(3 - 6) / 6
```

```
## [1] -0.5
```

This means that this fake program *caused* a 50% reduction in the outcome.

Find the percent change in car thefts because of the increase police presence after the July terror attack *using the results from Model C*. To do this, you need two numbers: (1) the average number of thefts in control blocks after the attack, and (2) the average number of thefts in treatment blocks after the attack. Because you're using Model C, your control group includes blocks that don't have synagogues within two blocks.

Use `group_by()` and `summarize()` to calculate the average number of thefts after the attack in control blocks (Hint: this will be just like the diff-in-diff by hand table you made in section 3, but instead of grouping by `same_block`, group by `more_than_two_away`).

```
terror_treatment_control <- terror %>%
  group_by(more_than_two_away, after) %>%
  summarize(avg_thefts = mean(car_theft))
terror_treatment_control
```

more_than_two_away	after	avg_thefts
0	FALSE	0.082
0	TRUE	0.095
1	FALSE	0.081
1	TRUE	0.108

Subtract the diff-in-diff effect for "same_block × after" from Model C from the average in the control group to find the average number of car thefts in treatment blocks. (Note: It'll be really tempting to just look at the table for the average for treatment + after, but this won't be right! You need to use control + diff-in-diff, since that's the counterfactual.)

Finally, calculate the percent change in car thefts after the terror attack across treatment and control blocks (hint: the answer is in the third full paragraph on p. 123 of the original article).

```r
# Extract specific numbers from these tables. You can also just write down the
# numbers by hand, but that's not 100% reproducible
control_after <- terror_treatment_control %>%
  filter(more_than_two_away == 1, after == TRUE) %>%
  pull(avg_thefts)
control_after
```

```
## [1] 0.108
```

```r
dd_effect <- tidy(model_fe_c) %>%
  filter(term == "same_block:afterTRUE") %>%
  pull(estimate)
dd_effect
```

```
## [1] -0.0808
```

```r
# (new - old) / old, or ((control + dd) - control) / control
pct_change <- ((control_after + dd_effect) - control_after) / control_after
pct_change
```

```
## [1] -0.749
```

The extra police presence thus reduced car thefts by 75%.