Replication Exercise

Tristan Misko

11/10/2021

Section 1: Overview of Paper

Exercise 1

In general, we would expect the causal effect of police presence on car thefts to be negative via a deterence or arrest effect. The regression

$$\text{car\_thefts}_d = \alpha + \beta \text{police\_officers}_d + \sum_{k=1}^{K} \gamma_k X_{k,d} + \epsilon_d$$

likely suffers significantly from omitted variables bias. In particular, it is likely that districts which have experienced high levels of car theft in the past are likely to have higher levels of police presence in response and are also likely to suffer from more car thefts during the response period. This positive correlation would bias the downward the magnitude of estimated causal effect $\hat{\beta}$, making it less negative than the true causal effect, and could even induce a perverse positive sign.

Question 2

The paper's research design uses plausibly exogenous shocks to police presence heterogeneously dispersed across districts to obtain causal estimates for the effect of policing on car thefts. The protection mandate for Jewish and Muslim institutions was implemented at the national level, so the increase in police presence in the vicinities of these institutions is unrelated to potential local police responses, eliminating the main source of confounding. This allows us to overcome endogeneity to obtain causal estimates.

Question 3

The parallel trends assumption is crucial to the analysis performed. All difference-in-differences designs require that the groups behave similarly pre-treatment so that we can attribute the difference observed post-treatment to the treatment itself and not to differing underlying trends. If the groups are following different trends before the treatment, we cannot untangle how much of of the observed difference is due to the trend and how much is due to the treatment.

Section 2: Setup

Dataset was downloaded and manipulated inside of *R*.

```
# read in data
mp2 <- read.csv("MonthlyPanel2.csv", header = T)
```

Variable descriptions:

1. observ (*type: integer*): Records the unit (block) of observation.
2. barrio (*type: string*): Records the *barrio* (district or borough) of the observation.
3. calle (*type: string*): Records the street name of the observation.
4. altura (*type: alphanumeric*): Records the address number of the block.
5. institu1 (*type: dummy*): Records the presence of a Jewish cultural institution on the block.
6. institu3 (*type: dummy*): Records the presence of a Jewish or Muslim cultural institution on the block.
7. distanci (*type: integer*): Records the distance to the nearest Jewish or Muslim cultural institution (CI) in blocks.
8. edpub (*type: dummy*): Records the presence of a public building on the block.
9. estserv (*type: dummy*): Records the presence of a gas station on the block.
10. banco (*type: dummy*): Records the presence of a bank on the block.
11. totrob (*type: float*): Records the number of motor vehicle thefts on a given block in a given month, coding thefts which occur on corners as 0.25 of a theft.
12. mes (*type: integer*): Records the month of the observation, with the coding of 72 for the first half of July and 73 for the second half.
13. totrob2 (*type: float*): Records the number of motor vehicle thefts on a given block in a given month, coding thefts which occur on corners as 0.25 of a theft.

Section 3: Data Creation

Exercise 1

We create a categorical variable by writing the `cat` function with the desired logical structure and applying it to the `distanci` variable and recording the result in the new column `category`.

```
ctg <- Vectorize(function(distanci){
    if (distanci == 0){
        return(1)
    } else if (distanci == 1){
        return(2)
    } else if (distanci == 2){
        return(3)
    } else {
        return(4)
    }
}, "distanci")
mp2$category <- ctg(mp2$distanci)
unique(mp2$mes)
```

```
##  [1]  4  5  6  7  8  9 10 11 12 72 73
```

Exercise 2

We use the `filter` function from the `dplyr` package to remove all observations for which `mes` takes value 72.

```
mp2 <- filter(mp2, mes != 72)
```

Exercise 3

We generate the variable `month` which takes the value of `mes` if `mes` is less than or equal to seven and `mes + 1` otherwise.

```
mes.shift <- Vectorize(function(mes){
    if (mes <= 7){
        return(mes)
    } else {
        return(mes + 1)
    }
}, "mes")
mp2$month <- mes.shift(mp2$mes)
unique(mp2$month)
```

```
##  [1]  4  5  6  7  9 10 11 12 13 74
```

Exercise 4

We map values of `month` which are equal to 74 to the value 8.

```
month.shift <- Vectorize(function(month){
    if (month == 74){
        return(8)
    } else {
        return(month)
    }
}, "month")
mp2$month <- month.shift(mp2$month)

#check that the remapping has worked correctly
sort(unique(mp2$month))
```

```
##  [1]  4  5  6  7  8  9 10 11 12 13
```

Exercise 5

Since *R* does not use the same labelling conventions as STATA, we opt to store the labels as a column in the table containing the desired string values.

```
label.month <- Vectorize(function(month){
    label.as <- c("April",
                  "May",
                  "June",
                  "July (1-17)",
                  "July (18-31)",
                  "August",
                  "September",
                  "October",
                  "November",
                  "December")
    return(label.as[month - 3])
}, "month")
mp2$month.label <- label.month(mp2$month)
```

Exercise 6

We store the cleaned data in a new file called `DataClean.csv`.

```
write.csv(mp2, "DataClean.csv")
```

Section 4: Descriptives

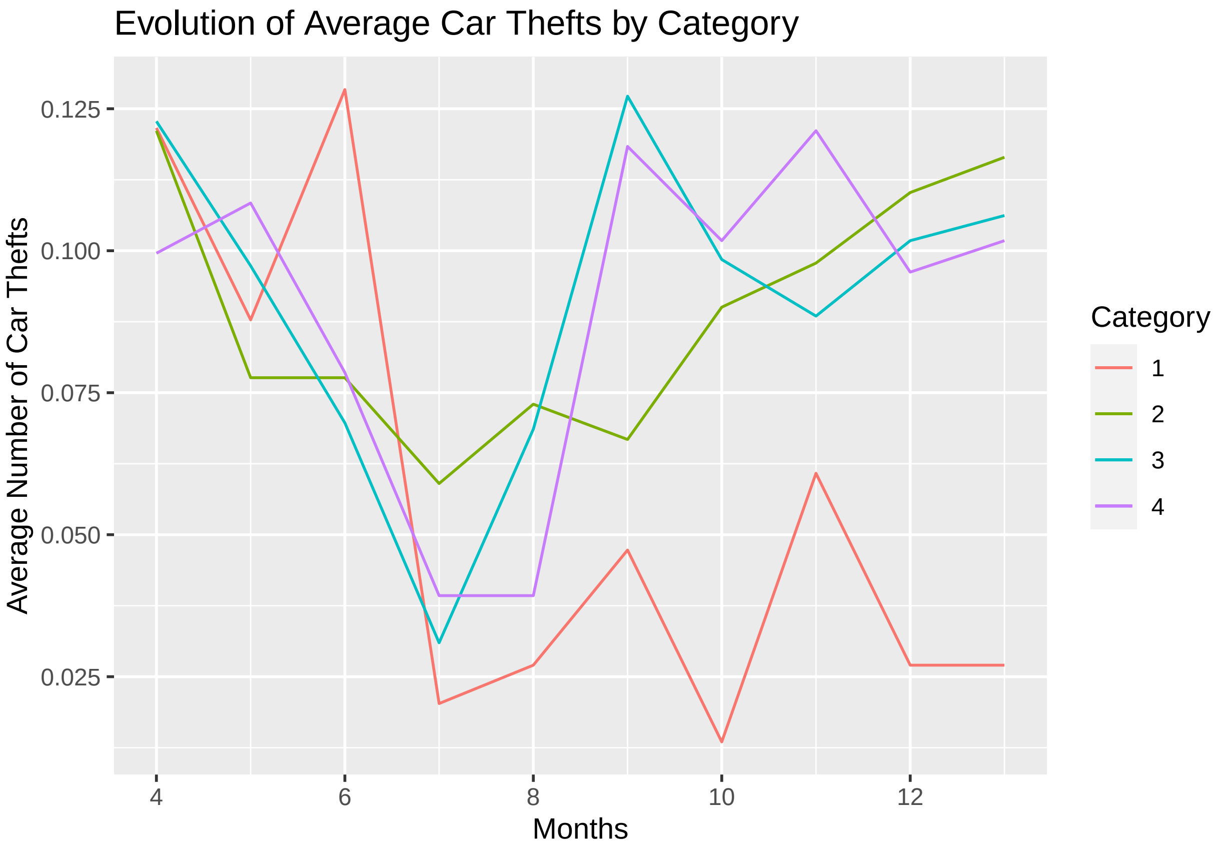
Exercise 1

We group the data by month and category using `group_by` from the `dplyr` package and we summarize the means using `summarize` from the dplyr package.
We feed the resulting table into `ggplot` to produce a line graph showing the evolution of the means by category.

```
#group the data by month and category
grouped <- mp2 %>% group_by(month, category) %>% summarize(mean(totrob2))
```

```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argument.
```

```
#generate plot from group structure:
ggplot(, aes(x = month, y = `mean(totrob2)`)) +
    geom_line(data = filter(grouped, category == 1), aes(color = factor(category))) +
    geom_line(data = filter(grouped, category == 2), aes(color = factor(category))) +
    geom_line(data = filter(grouped, category == 3), aes(color = factor(category))) +
    geom_line(data = filter(grouped, category == 4), aes(color = factor(category))) +
    labs(title = "Evolution of Average Car Thefts by Category",
         color = "Category") +
    ylab("Average Number of Car Thefts") + xlab("Months")
```



Exercise 2

We construct a function to produce the proper latex string for the table from the input data frame, then we paste the result into the document to obtain the outputed table.

```
dat <- read.csv("DataClean.csv")
mean.and.sd <- dat %>% group_by(month.label, category) %>%
    summarize(m = mean(month), mean = round(mean(totrob2),5), sd = round(sd(totrob2), 5))
```

```
## `summarise()` has grouped output by 'month.label'. You can override using the `.groups` argument.
```

```
cat4 <- select(filter(mean.and.sd, category == 4),c(1,3,4,5))
cat1 <- select(filter(mean.and.sd, category == 1),c(1,4,5))
cat2 <- select(filter(mean.and.sd, category == 2),c(1,4,5))
cat3 <- select(filter(mean.and.sd, category == 3),c(1,4,5))
table2 <- select(arrange(select(data.frame(cat4, cat1, cat2, cat3), c(-5, -8, -11)), m), -2)

generate_table <- function(table2){
  table_string <-
    "\\begin{table}[!htbp] \\centering \n
    \\caption{Evolution of Mean Car Thefts by Category} \n
    \\label{} \n
    \\begin{tabular}{lcccc} \n
    \\\\[-1.8ex]\\hline \n
    \\hline \\\\[-1.8ex] \n
    \\begin{minipage}[t]{0.10\\columnwidth}\\vspace{1.2cm}Month\\end{minipage} &
    \\begin{minipage}[t]{0.15\\columnwidth}\\centering More than two blocks from
    nearest Jewish institution \\\\(A)\\end{minipage} &
    \\begin{minipage}[t]{0.15\\columnwidth}\\centering Jewish institution on the
    block \\\\ (B) \\end{minipage} &
    \\begin{minipage}[t]{0.15\\columnwidth}\\centering One block from nearest
    Jewish institution \\\\ (C)\\end{minipage} &
    \\begin{minipage}[t]{0.15\\columnwidth}\\centering Two blocks from nearest
    Jewish institution \\\\ (D)\\end{minipage} \\\\ \\hline \\\\ \n"
    for (row in 1:length(table2$month.label)){
        means <- paste(c(paste(as.character(table2[row,c(1,2,4,6,8)]),
                         collapse = " & "), "\\\\", "\n"), collapse = " ")
        sds <- paste(c(paste(c(" ", paste0("(", as.character(table2[row,c(3,5,7,9)]), ")")),
                             collapse = " & "), "\\\\", "\n"), collapse = "")
        table_string <- paste0(table_string, paste0(means, sds))
    }
    table_string <- paste0(table_string,"\\multicolumn{5}{c}{\\begin{minipage}[t]{0.8\\columnwidth}\\vs
    cat(table_string)
}

generate_table(table2)
```

Table 1: Evolution of Mean Car Thefts by Category

Month	More than two blocks from nearest Jewish institution (A)	Jewish institution on the block (B)	One block from nearest Jewish institution (C)	Two blocks from nearest Jewish institution (D)
April	0.09956	0.12162	0.12112	0.12279
	(0.24814)	(0.36143)	(0.28791)	(0.29743)
May	0.10841	0.08784	0.07764	0.09735
	(0.23574)	(0.20595)	(0.18165)	(0.25913)
June	0.07854	0.12838	0.07764	0.06969
	(0.19606)	(0.28639)	(0.21512)	(0.18667)
July (1-17)	0.03927	0.02027	0.05901	0.03097
	(0.14506)	(0.06918)	(0.21013)	(0.14194)
July (18-31)	0.03927	0.02703	0.07298	0.06858
	(0.14601)	(0.0787)	(0.21766)	(0.23863)
August	0.11836	0.0473	0.06677	0.12721
	(0.28707)	(0.17518)	(0.21965)	(0.30481)
September	0.10177	0.01351	0.09006	0.09845
	(0.25659)	(0.05731)	(0.27607)	(0.2483)
October	0.12113	0.06081	0.09783	0.0885
	(0.26712)	(0.21575)	(0.26097)	(0.23669)
November	0.09624	0.02703	0.11025	0.10177
	(0.2404)	(0.0787)	(0.28892)	(0.21766)
December	0.10177	0.02703	0.11646	0.10619
	(0.26821)	(0.0787)	(0.27815)	(0.2256)

Notes: The first four columns present the mean and standard deviation (in parentheses) of the number of car thefts for each type of block per month. The average number of car thefts for July can be obtained by summing the subperiods.

Exercise 3

The monthly means are comparable in all of the months preceding the attack except for June, in which there is a large spike in average car thefts on the blocks which contain Jewish cultural institutions which does not appear in any of the other categories. This is a bit of an issue for the parallel trends assumption and we must proceed with caution. The anomaly abates by early July in the pre-treatment period, so we can argue that we get back to the trend by the time the treatment occurs. If we consider the bias that the June spike would introduce, we see that a high pre-treatment level should imply a high post-treatment level, so if anything the reduction in vehicle thefts on blocks containing a Jewish cultural institution after the treatment is more compelling for the spike. Hence, the observed deviation from parallel trends increases our Type II error rate (we are more likely to fail to reject a false null hypothesis that there is no difference in differences) but will decrease our Type I error rate. The estimated coefficient is likely slightly lower in magnitude than it would be if there were no June spike.

Section 5: Difference-in-Differences

Exercise 1

We filter out the observations with `month == 8` and produce the `post` variable. We are then free to run the regressions using the `factor` command to create dummies for `month` and `observ`, thereby implementing time and space fixed effects. We use the `coeftest` command to obtain heteroskedasticity robust standard errors, and we report the results in the table below.

```
# drop observations with with month
mp2 <- mp2 %>% filter(month != 8)
#create post treatment variable
mp2$post <- as.numeric(mp2$month > 8)
#regression (A)
mp2$same_block_police <- as.numeric(mp2$category == 1)*mp2$post
reg.a <- lm(data = mp2,
            totrob2 ~ same_block_police +
                factor(month) + factor(observ))
coeftest(reg.a, vcov = vcovHC(reg.a, "HC1"))[1:2,]
```

```
##                      Estimate Std. Error   t value    Pr(>|t|)
## (Intercept)        0.01568456 0.01123335  1.396249 0.162683734
## same_block_police -0.07752956 0.02243617 -3.455561 0.000552408
```

```
#regression (B)
mp2$one_block_police <- as.numeric(mp2$category == 2)*mp2$post
reg.b <- lm(data = mp2,
            totrob2 ~ same_block_police + one_block_police +
                factor(month) + factor(observ))
coeftest(reg.b, vcov = vcovHC(reg.b, "HC1"))[1:3,]
```

```
##                      Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)        0.01427095 0.01146761  1.2444567 0.2133732168
## same_block_police -0.08007405 0.02256866 -3.5480201 0.0003906822
## one_block_police  -0.01325979 0.01386399 -0.9564196 0.3388933270
```

```
#regression (C)
mp2$two_blocks_police <- as.numeric(mp2$category == 3)*mp2$post
reg.c <- lm(data = mp2,
            totrob2 ~ same_block_police + one_block_police + two_blocks_police +
                factor(month) + factor(observ))
coeftest(reg.c, vcov = vcovHC(reg.c, "HC1"))[1:4,]
```

```
##                       Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)        0.013866369 0.01167206  1.1879970 0.2348749232
## same_block_police -0.080802290 0.02294494 -3.5215732 0.0004317281
## one_block_police  -0.013988038 0.01446689 -0.9668999 0.3336274871
## two_blocks_police -0.002184735 0.01231580 -0.1773928 0.8592050271
```

Table 2: The Effect of Police Presence on Car Theft

	Difference-in-difference		
	totrob2		
	(1)	(2)	(3)
same_block_police	−0.07752***	−-0.08007***	−0.08080***
	(0.022)	(0.022)	(0.022)
one_block_police		−0.01325	−0.01398
		(0.013)	(0.014)
two_blocks_police			−0.00218
			(0.011)
Observations	7,884	7,884	7,884
R^2	0.1983	0.1984	0.1984
Note:		*p<0.1; **p<0.05; ***p<0.01	

Exercise 2

The interpretation of the coefficent in column (A) of the table is that the presence of police on a particular block decreases the number of car thefts on that block by 0.0775 on average. Considering the average number of car thefts on each block per month is 0.0929414, this represents a very large effect size.

Exercise 3

From Table 4 we learn that the time that the implementation dummy activates is crucial for the significance of the results. The table shows the estimates of the effect size if the treatment dummy turns on before the actual treatment is given for three sample dates: April 30, May 31, and June 30. If these estimates were significant, then we would have some cause to doubt that the treatment is driving the difference outcomes. The non-significant results of these tests make us more confident that the difference-in-differences design is working as intended and picking up the causal effect of the treatment.