Florida's Family Tranistion Program - A Difference-in-Differences Approach

Daniel Chen

June 2021

Project Overview

The goal of this project is to analyze the effect of Florida's Family Transition Program on outcomes such as employment and welfare receipt through a difference-in-differences approach. The treatment variable will not be the original assignment variable, `e`, but instead, I will derive a variable indicating whether or not an individual *believed* that their benefits were time limited.

Understanding the Data

Before we continue with the diff-in-diff analysis, we may wonder if there is a difference in employment rates between the treatment and control group prior to the period of random assignment? Let's start by checking out the data.

```r
# Load libraries
library(tidyverse)
library(foreign)
library(broom)
library(plm)
library(lmtest)
library(kableExtra)

# Load in data
admin <- read.dta(paste("/Users/danielchen/Desktop/UChicago/Year Two/Autumn 2020/",
                        "Program Evaluation/Problem Sets/Problem Set 4/ftp_ar.dta",
                        sep = ""))
survey <- read.dta(paste("/Users/danielchen/Desktop/UChicago/Year Two/Autumn 2020/",
                         "Program Evaluation/Problem Sets/Problem Set 4/ftp_srv.dta",
                         sep = ""))

# View snippet of data
admin %>%
  select(1:11) %>%
  head(10) %>%
  kable("latex")
```

sampleid	e	cflag	longtdec	b_aidst	gender	ethnic	marital	afdctime	afdchild	higrade
1	0	1	1	2	2	5	5	2	3	12
10	1	NA	5	1	2	5	1	5	2	14
100	0	NA	2	2	2	1	1	2	2	12
1000	0	NA	1	2	2	1	5	2	3	12
1001	1	NA	5	1	2	1	1	4	1	12
1002	0	NA	2	1	2	1	3	7	1	10
1003	0	NA	2	1	2	5	3	1	3	11
1004	0	1	1	1	2	1	4	5	1	12
1005	0	NA	10	2	2	1	3	6	1	10
1006	0	NA	7	1	2	5	5	6	3	10

```r
survey %>%
  select(1:10) %>%
  head(10) %>%
  kable("latex")
```

sampleid	fmwavtyp	fmsamtyp	fma1	fma1dk	fma1a1	fma1a2	fma1a21	fma1a31	fma1a22
1	2	C	2	NA	NA	NA	NA	NA	NA
100	2	C	2	NA	NA	NA	NA	NA	NA
1000	2	N	2	NA	NA	NA	NA	NA	NA
1004	1	C	2	NA	NA	NA	NA	NA	NA
1007	2	C	2	NA	NA	NA	NA	NA	NA
1009	1	C	2	NA	NA	NA	NA	NA	NA
101	1	N	1	NA	2	96	NA	NA	NA
1010	1	C	2	NA	NA	NA	NA	NA	NA
1013	1	C	2	NA	NA	NA	NA	NA	NA
1018	1	C	2	NA	NA	NA	NA	NA	NA

We can see above that the data is currently wide. We'll need to merge the admin and survey data, and then reshape it from wide to long in order to analyze it as a time series. We'll also need to create a new treatment dummy variable because we're interested in the effect of believing in the time limit on employment outcomes - not assignment to treatment or control itself.

Reshaping the Data

Because there's a number of variables we're interested in, I'll write a loop to return the names of the variables we're interested in instead of typing them out individually. All the employment variables start with either "empq" or "emppq". Note that variables that start with the prefix "emppq" denote time periods prior to random assignment and variables that start with "empq" denote periods after random assignment.

```r
# Create function that uses a loop to create a vector containing the names of
# the employment variables we're interested in
make_variable <- function(prefix, start, end) {

  vars <- c()
  for (number in start:end) {
    variable <- str_c(prefix, as.character(number))
    vars <- c(vars, variable)
  }

  return(vars)
}

# Create variable names and store to object
pre_vars <- make_variable(prefix = "emppq", start = 1, end = 10)
post_vars <- make_variable(prefix = "empq", start = 1, end = 20)

# View our variables
pre_vars
```

```
##  [1] "emppq1"  "emppq2"  "emppq3"  "emppq4"  "emppq5"  "emppq6"  "emppq7"
##  [8] "emppq8"  "emppq9"  "emppq10"
```

```
post_vars
```

```
##  [1] "empq1"  "empq2"  "empq3"  "empq4"  "empq5"  "empq6"  "empq7"  "empq8"
##  [9] "empq9"  "empq10" "empq11" "empq12" "empq13" "empq14" "empq15" "empq16"
## [17] "empq17" "empq18" "empq19" "empq20"
```

Now that we have vectors containing the names of our variables of interest as strings, we can pass them through tidyverse syntax and reshape the data as needed.

```
# Merge admin data, only keep valid responses for variable fmi2, derive a
# new variable named TLyes denoting whether an individual believed in the time
# limit or not, keep only sampleid, TLyes, and all the employment variables,
# reshape data from wide to long, and add dummy indicating whether employment
# var falls into pre-random assignment or post-random assignment
ftp <- merge(admin, survey, by = "sampleid") %>%
  rename_at(vars(ends_with(".x")), ~str_replace(., "\\..$", "")) %>%
  select(-ends_with(".y")) %>%
  as_tibble() %>%
  filter(fmi2 == 1 | fmi2 == 2) %>%
  mutate(TLyes = case_when(fmi2 == 1 ~ 1,
                           fmi2 == 2 ~ 0)) %>%
  select(sampleid, TLyes, all_of(c(pre_vars, post_vars))) %>%
  pivot_longer(starts_with("e"), names_to = "quarter", values_to = "employed") %>%
  mutate(post_treat = case_when(str_sub(quarter, 1, 5) == "emppq" ~ 0,
                                TRUE ~ 1))

# Preview new dataframe
ftp %>%
  head(5) %>%
  kable("latex")
```

sampleid	TLyes	quarter	employed	post_treat
1007	1	emppq1	0	0
1007	1	emppq2	0	0
1007	1	emppq3	0	0
1007	1	emppq4	1	0
1007	1	emppq5	1	0

Determining the Period of Analysis

Even though our data is in the wide format, we need to check if there are any missing values. If any quarters contain NA values, we cannot use a difference-in-differences approach. If any quarters contain missing observations, we'll drop them from the analysis. For ease, we'll return to the wide format admin data.

```r
# Calculate means
admin %>%
  select(starts_with(c("emppq", "empq"))) %>%
  map(~sum(is.na(.))) %>%
  bind_cols() %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column(var = "variable") %>%
  rename(`NA count` = V1) %>% # Using spaces in variable names for presentation
  kable("pipe")
```

variable	NA count
emppq10	379
emppq9	0
emppq8	0
emppq7	0
emppq6	0
emppq5	0
emppq4	0
emppq3	0
emppq2	0
emppq1	0
empq1	0
empq2	0
empq3	0
empq4	0
empq5	0
empq6	0
empq7	0
empq8	0
empq9	0
empq10	0
empq11	0
empq12	0
empq13	0
empq14	0
empq15	0
empq16	0
empq17	0
empq18	1
empq19	5
empq20	646

We'll need to drop `emppq10`, `empq18`, `empq19`, and `empq20` because they contain missing observations.

Do Employment Rates Differ Based on Belief in the Time Limit?

We may be curious to see if employment rates differed based on whether or not an individual believed in the time limit or not.

```r
ftp %>%
  group_by(TLyes, post_treat) %>%
  summarise(`Employment Rate` = mean(employed, na.rm = TRUE), .groups = "drop") %>%
  ungroup() %>%
  kable("pipe")
```

TLyes	post_treat	Employment Rate
0	0	0.1917808
0	1	0.4299208
1	0	0.2115616
1	1	0.4922737

In comparing individuals who believed in the time limit versus those who did not believe in the time limit in the pre-random assignment period, there's about a two percentage point difference between the groups. The employment rate among those who did not believe in the time limit in the pre-period (`TLyes` = 0 and `post_treat` = 0) was roughly 19%. Conversely, the employment rate among those who did believe in the time limit in the pre-period (`TLyes` = 1 and `post_treat` = 0) was about 21%. In terms of magnitude, the percentages appear fairly similar. We can run a regression to see if the difference is statistically significant.

```r
ftp %>%
  filter(post_treat == 0) %>%
  filter(is.na(employed) == FALSE) %>%
  lm(employed ~ TLyes, data = .) %>%
  tidy() %>%
  kable("pipe")
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.1917808	0.0066756	28.728695	0.0000000
TLyes	0.0197807	0.0083058	2.381557	0.0172577

From the table above, we can see that the t-statistic on the variable `TLyes` is larger than two suggesting a statistically significant difference in employment rates between those who believed in the time limit and those who did not.

Checking the for Pre-Treatment Parallel Trends

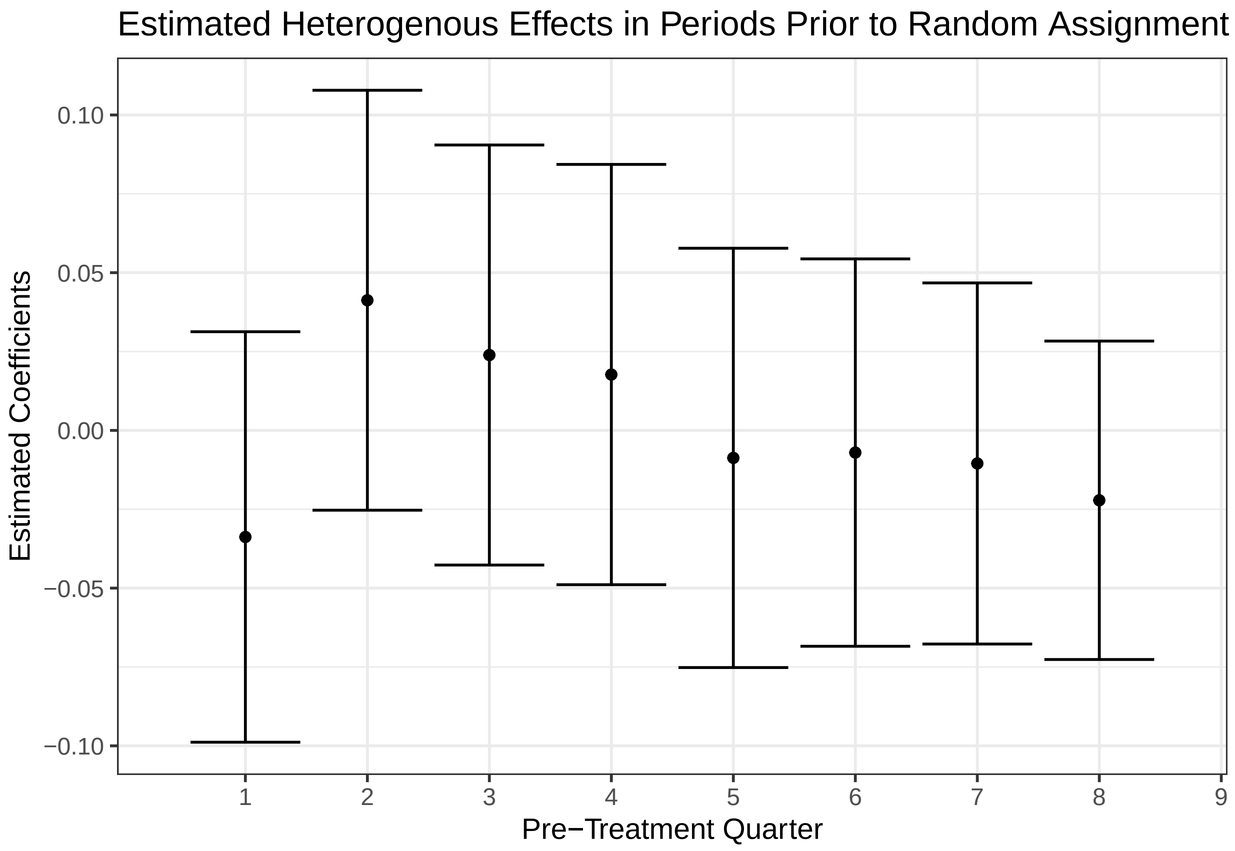
Before continuing with the DiD approach, we need to check for pre-treatment parallel trends. By testing to see if the treatment and control groups have parallel trends in the pre-treatment period, we can then attribute any differences in post period outcomes to the treatment itself as the treatment is the only change introduced after random assignment.

```r
# Store vector containing coefficients of interest
pre_treat_estimates <- c("quarteremppq2:TLyes",
                         "quarteremppq3:TLyes",
                         "quarteremppq4:TLyes",
                         "quarteremppq5:TLyes",
                         "quarteremppq6:TLyes",
                         "quarteremppq7:TLyes",
                         "quarteremppq8:TLyes",
                         "quarteremppq9:TLyes")

# Run DiD using plm() function and store output to object
period_effects <- plm(
  employed ~ sampleid + quarter + TLyes*quarter,
  data = ftp %>% filter(!quarter %in% c("empq18", "empq19", "empq20", "emppq10")),
  index = c("sampleid", "quarter"),
  model = "within"
)

# Cluster standard errors by sampleid and return a dataframe containing the pre-
# treatment estimates
pre_treat_results <- coeftest(
  period_effects,
  vcovHC(period_effects, type = "HC1", cluster = "group")
) %>%
  tidy(conf.int = TRUE) %>%
  filter(term %in% pre_treat_estimates)

# View results
pre_treat_results %>%
  kable("pipe")
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
quarteremppq2:TLyes	-0.0221523	0.0257500	-0.8602835	0.3896408	-0.0726237	0.0283191
quarteremppq3:TLyes	-0.0104817	0.0292036	-0.3589185	0.7196590	-0.0677224	0.0467590
quarteremppq4:TLyes	-0.0070303	0.0313289	-0.2244033	0.8224453	-0.0684368	0.0543762
quarteremppq5:TLyes	-0.0087169	0.0339105	-0.2570569	0.7971369	-0.0751835	0.0577496
quarteremppq6:TLyes	0.0177054	0.0339898	0.5209027	0.6024390	-0.0489165	0.0843273
quarteremppq7:TLyes	0.0238965	0.0339697	0.7034659	0.4817718	-0.0426859	0.0904789
quarteremppq8:TLyes	0.0412604	0.0339616	1.2149147	0.2244098	-0.0253062	0.1078271
quarteremppq9:TLyes	-0.0337858	0.0331969	-1.0177400	0.3088111	-0.0988537	0.0312820

When running the regression above, our coefficient of interest is the interaction between `TLyes` and each quarter in the pre-treatment period. By specifying each quarter individually, we can estimate the effect of being in the treatment group prior to the treatment period. In other words, each estimate above tells us the effect on employment for those in the treatment group relative to those in the control group. We can also plot the estimates and their corresponding 95% confidence intervals to visualize the data.

```r
pre_treat_results %>%
  mutate(
    period = case_when(
      term == "quarteremppq2:TLyes" ~ 8,
      term == "quarteremppq3:TLyes" ~ 7,
      term == "quarteremppq4:TLyes" ~ 6,
      term == "quarteremppq5:TLyes" ~ 5,
      term == "quarteremppq6:TLyes" ~ 4,
      term == "quarteremppq7:TLyes" ~ 3,
      term == "quarteremppq8:TLyes" ~ 2,
      term == "quarteremppq9:TLyes" ~ 1
    )
  ) %>%
  mutate(period = as.numeric(period)) %>%
  ggplot(aes(x = period, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  theme_bw() +
  scale_x_discrete(breaks = c(1:9), labels = c(1:9), limits = c(1:9)) +
  labs(
    x     = "Pre-Treatment Quarter",
    y     = "Estimated Coefficients",
    title = "Estimated Heterogenous Effects in Periods Prior to Random Assignment"
  )
```



From the plot above, each point represents the estimate and the whiskers represent the 95% confidence intervals. As we move from left to right on the on the x-axis, we're moving closer towards the period of random assignment. As we can see, all of the 95% confidence intervals encompass zero, so none of the estimates are statistically significant. In other words, the parallel trends assumption is met. There is no statistical difference in employment rates between the treatment and control groups in the pre-treatment period. This should be the case because an intervention has not yet took place.

Note on parallel trends assumption: Testing for pre-treatment parallel trends is neither necessary nor sufficient because it is an approximate test that doesn't actually identify the important component of parallel trends. We really need to observe the untreated outcome in the treatment group in the post-period. Through pre-treatment parallel trends, we *hope* that in the absence of the intervention, the potential untreated outcomes would have trended in a parallel manner in the periods before and after the intervention took place. The key word here is hope, as there is no guarantee that untreated outcomes would be parallel absent the intervention in the post-period.

Difference-in-Differences Analysis

Homogenous Treatment Effects in the Post-Period

For part one of the diff-in-diff, we'll assume homogenous treatment effects in the post-treatment period. However, effects may vary by period throughout the post-treatment period. We'll take a look at the heterogenous effects in part two.

We can now continue with the diff-in-diff approach. On the right side we want the individual dummies, the quarter dummies, and the interaction between belief in the time limit (`TLyes`) and period of analysis (`post_treat`). We need this interaction because we're interested in the effect after the period of random assignment when an individual believes in the time limit. In the pre-treatment period, the `post_treat` variable is equal to zero, canceling out any effects.

```
ftp %>%
  filter(!quarter %in% c("empq18", "empq19", "empq20", "emppq10")) %>%
  lm(employed ~ sampleid + quarter + TLyes*post_treat, data = .) %>%
  tidy() %>%
  filter(str_sub(term, 1, 6) == "TLyes:") %>%
  kable("pipe")
```

term	estimate	std.error	statistic	p.value
TLyes:post_treat	0.0500453	0.0106414	4.702879	2.6e-06

From our estimates above, we may be worried about standard errors. Autocorrelation tends to be a problem for panel data as observations are not independent. We can use the `plm()` function from the `plm` library to cluster standard errors at the individual level to correct for this time dependence.

```
# Run DiD using plm() function and store output to object
clustered_plm <- plm(
  employed ~ sampleid + quarter + TLyes*post_treat,
  data = ftp %>% filter(!quarter %in% c("empq18", "empq19", "empq20", "emppq10")),
  index = c("sampleid", "quarter"),
  model = "within"
)

# Cluster standard errors at the individual level
coeftest(
  clustered_plm,
  vcovHC(clustered_plm, type = "HC1", cluster = "group")
```

```r
) %>%
  tidy() %>%
  filter(term == "TLyes:post_treat") %>%
  kable("pipe")
```

term	estimate	std.error	statistic	p.value
TLyes:post_treat	0.0500453	0.0199897	2.503553	0.0123015

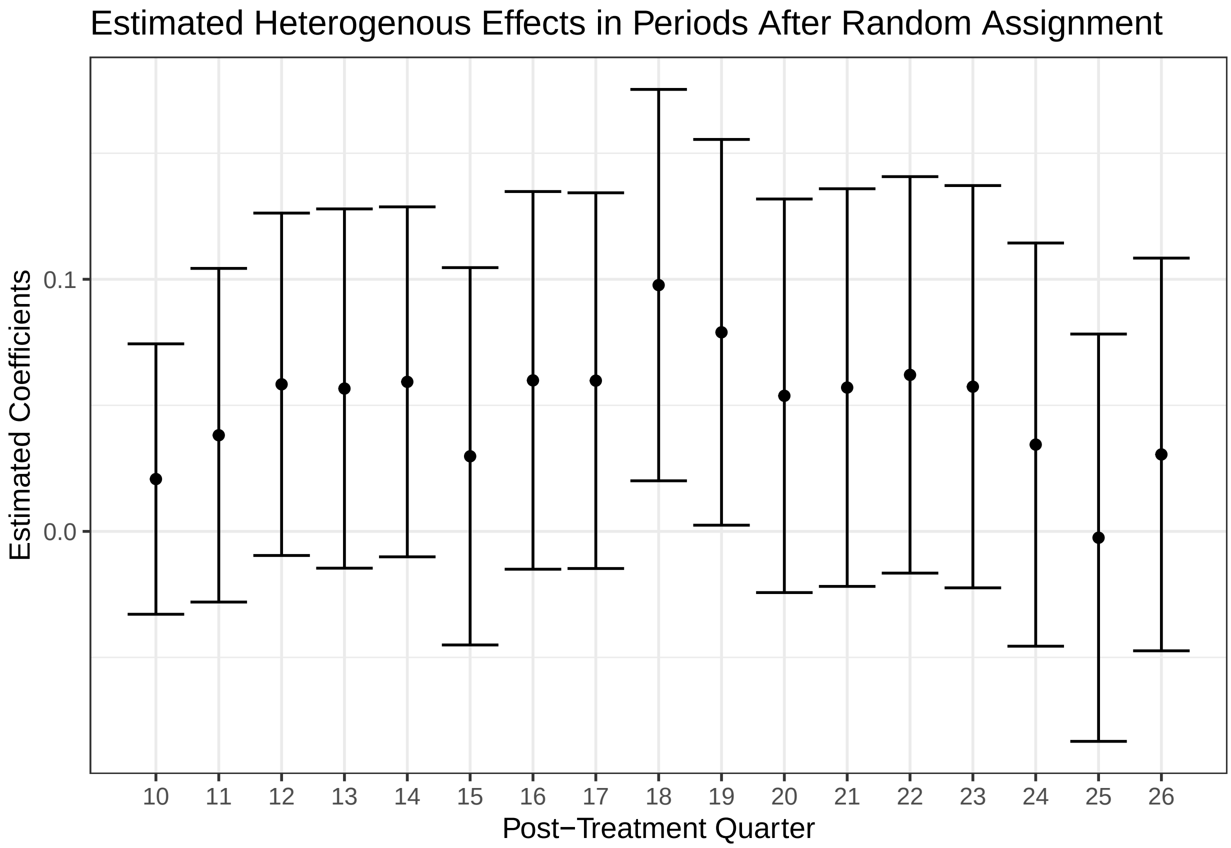
As we can see from the output above, the estimate remains the same at roughly 0.05, but the standard error has increased from roughly 0.0106 to 0.02. As the standard error has increased, we are less certain of our estimate. If we hadn't clustered standard errors at the individual level we potentially overstate/overestimate the significance of our results by (falsely) assuming greater precision in our estimates.

In terms of interpretation, the interaction between `TLyes` and `post_treat` tells us the average effect of treatment on the treated assuming our assumptions hold (as we saw above, the pre-treatment parallel trends assumption was met). If an individual believes that their benefits are time-limited, and they are in the post-treatment period, this is – on average – associated with a roughly 5% increase in the rate of being employed. Statistically, this difference is meaningful as the p-value is less than our 0.05 threshold.

Heterogenous Treatment Effects in the Post-Period

Now we'll let treatment effects vary in the post-treatment period. We'll plot the point estimates in each quarter of the post-period similar to what we did when we tested for pre-treatment parallel trends.

```r
# Run for loop to store a vector containing the names of our coefficients of interest
post_treat_estimates <- c()
for (number in 1:17) {
  var_name <- str_c("quarterempq", number , ":TLyes")
  post_treat_estimates <- c(post_treat_estimates, var_name)
}

# Cluster standard errors by sampleid and return a dataframe containing the
# post-treatment estimates
post_treat_results <- coeftest(
  period_effects,
  vcovHC(period_effects, type = "HC1", cluster = "group")
) %>%
  tidy(conf.int = TRUE) %>%
  filter(term %in% post_treat_estimates)

# Plot estimates
post_treat_results %>%
  mutate(
    period = case_when(
      term == "quarterempq1:TLyes"  ~ 10,
      term == "quarterempq2:TLyes"  ~ 11,
      term == "quarterempq3:TLyes"  ~ 12,
      term == "quarterempq4:TLyes"  ~ 13,
      term == "quarterempq5:TLyes"  ~ 14,
      term == "quarterempq6:TLyes"  ~ 15,
      term == "quarterempq7:TLyes"  ~ 16,
      term == "quarterempq8:TLyes"  ~ 17,
      term == "quarterempq9:TLyes"  ~ 18,
      term == "quarterempq10:TLyes" ~ 19,
      term == "quarterempq11:TLyes" ~ 20,
      term == "quarterempq12:TLyes" ~ 21,
      term == "quarterempq13:TLyes" ~ 22,
      term == "quarterempq14:TLyes" ~ 23,
      term == "quarterempq15:TLyes" ~ 24,
      term == "quarterempq16:TLyes" ~ 25,
      term == "quarterempq17:TLyes" ~ 26,
    )
  ) %>%
  mutate(period = as.numeric(period)) %>%
  ggplot(aes(x = period, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  theme_bw() +
  scale_x_discrete(breaks = c(10:26), labels = c(10:26), limits = c(10:26)) +
  labs(
    x     = "Post-Treatment Quarter",
    y     = "Estimated Coefficients",
    title = "Estimated Heterogenous Effects in Periods After Random Assignment"
  )
```



With the exception of `empq9` (“18” on the x-axis) and `empq10` (“19” on the x-axis) none of the period specific estimates are statistically significant as all of the confidence intervals encapsulate zero. However, this does not concern me. The DiD estimator is concerned with the significance of the average effect of the entire sample period. While there appears to be positive heterogeneous effects in each post-period, the question is whether or not they are enough to make the average effect in the post-period significant by imposing the constraint of heterogeneous effects even when there are more than two time periods.