Lecture 9: Difference in Differences in Multi-Period and Multi Group-Setting

Vinish Shrestha

Notes.

2025-04-30

1. Multi period and multi group
2. Simulating the data for Multi-group and multi-period setup
3. Estimating TWFE in case of homogeneous treatment effect
4. Heterogeneous treatment effects
5. Event study models

Section 1

Multi period and multi group

From canonical DiD to multi-period and multi-group

- Target: Simulate data for the following pooled cross sectional setting

1. groups (G) = 3
2. T = 20 (20 periods)
   - time period
3. n = 5000
   - every time period we have a sample of 5000

- num. of obs. in data: Group [1] $(T \times n)$ + Group [2] $(T \times n)$ + Group [3] $(T \times n)$

Multi-group and multi-time

- there are 3 groups (unlike 2 of canonical DiD)
- $T = 20$ (unlike 2 of canonical DiD)
- variation in treatment timing
  - set it up so that group 1 gets treated at 6 (early treated)
  - group 2 gets treated at 16 (late treated)
  - group 3 (untreated)

Section 2

Simulating the data for Multi-group and multi-period setup

```r
# set seed
set.seed(1256)
# Set up true parameters
trueeffect <- 10
intercep <- 10
N <- 5000
T <- 20
early <- 6
late <- 16

# A skeleton function
datagen <- function(T, N, group){
  timeT <- rep(1:T, each = N)
  treatT <- rep(1, length(timeT))
  groupT <- rep(group, length(timeT))
  df <- data.frame(time = timeT, treat = treatT, group = groupT)
  return(df)
```

```r
# call the function

dftreat <- datagen(T, N, 1) # early treatment
dftreat2 <- datagen(T, N, 2) # late treatment
dfuntreat <- datagen(T, N, 3) # untreated
```

Group 1

```
data <- rbind(dftreat, dftreat2, dfuntreat)
data[1:20, ]
```

```
##    time treat group
## 1     1     1     1
## 2     1     1     1
## 3     1     1     1
## 4     1     1     1
## 5     1     1     1
## 6     1     1     1
## 7     1     1     1
## 8     1     1     1
## 9     1     1     1
## 10    1     1     1
## 11    1     1     1
```

Group 2

```
data[100001:100020, ]
```

```
##        time treat group
## 100001    1     1     2
## 100002    1     1     2
## 100003    1     1     2
## 100004    1     1     2
## 100005    1     1     2
## 100006    1     1     2
## 100007    1     1     2
## 100008    1     1     2
## 100009    1     1     2
## 100010    1     1     2
## 100011    1     1     2
## 100012    1     1     2
```

Group 3

```
data[200001:200020, ]
```

```
##          time treat group
## 200001      1     1     3
## 200002      1     1     3
## 200003      1     1     3
## 200004      1     1     3
## 200005      1     1     3
## 200006      1     1     3
## 200007      1     1     3
## 200008      1     1     3
## 200009      1     1     3
## 200010      1     1     3
## 200011      1     1     3
## 200012      1     1     3
```

```
table(data$time[data$group == 1])
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000
##   17   18   19   20
## 5000 5000 5000 5000
```

```
table(data$time[data$group == 2])
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000
##   17   18   19   20
## 5000 5000 5000 5000
```

```r
table(data$time[data$group == 3])
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000
##   17   18   19   20
## 5000 5000 5000 5000
```

Staggered policy roll-out

- policy can go into effect at different time periods (unlike canonical did)
- once the policy is “on”, it doesn’t turn off
- In the simulated data
  - early group: 6th period
  - late group: 16th period

```r
# generating policy variables
data <- data %>%
        mutate(policy = 0,
        policy = ifelse(group == 1 & time >= early, 1, policy),
        policy = ifelse(group == 2 & time >= late, 1, policy),
        dumtreat1 = ifelse(group == 1, 1, 0),
        dumtreat2 = ifelse(group == 2, 1, 0),
        dumtreat3 = ifelse(group == 3, 1, 0))
```

```r
# simulate the outcome
e <- rnorm(nrow(data), 0, 5)

data <- data %>% mutate(Y = 1 + trueeffect*dumtreat1*policy +
                         trueeffect*dumtreat2*policy +
                         time + # common time trend
                         dumtreat1*2 + dumtreat2*4 + # time invariant
                         #unobserved heterogeneity
                         e,  # error term
                         Ypot = 1 +
                         time +
                         dumtreat1*2 + dumtreat2*4 + e)
```

Group 1

```r
data[1:20, ]
```

```
##      time treat group policy dumtreat1 dumtreat2 dumtreat3          Y          Y
## 1       1     1     1      0         1         0         0 -0.3668203 -0.3668
## 2       1     1     1      0         1         0         0  8.0325062  8.0325
## 3       1     1     1      0         1         0         0  7.8423163  7.8423
## 4       1     1     1      0         1         0         0  8.9015771  8.9015
## 5       1     1     1      0         1         0         0 11.9657890 11.9657
## 6       1     1     1      0         1         0         0  0.4926646  0.4926
## 7       1     1     1      0         1         0         0  9.9828709  9.9828
## 8       1     1     1      0         1         0         0 -0.3174144 -0.3174
## 9       1     1     1      0         1         0         0  1.5232241  1.5232
## 10      1     1     1      0         1         0         0  1.0805328  1.0805
## 11      1     1     1      0         1         0         0  5.2858838  5.2858
## 12      1     1     1      0         1         0         0  8.1822637  8.1822
```

Group 2

```
data[100001:100020, ]
```

```
##        time treat group policy dumtreat1 dumtreat2 dumtreat3          Y
## 100001    1     1     2      0         0         1         0  6.0919264
## 100002    1     1     2      0         0         1         0 18.9396561
## 100003    1     1     2      0         0         1         0 10.2502914
## 100004    1     1     2      0         0         1         0 11.9784019
## 100005    1     1     2      0         0         1         0  5.1051618
## 100006    1     1     2      0         0         1         0 11.0657456
## 100007    1     1     2      0         0         1         0  7.8192590
## 100008    1     1     2      0         0         1         0  2.0059905
## 100009    1     1     2      0         0         1         0 14.8845040
## 100010    1     1     2      0         0         1         0  6.5452892
## 100011    1     1     2      0         0         1         0 11.9790474
## 100012    1     1     2      0         0         1         0  1.9248714
```

Group 3

```r
data[200001:200020, ]
```

```
##        time treat group policy dumtreat1 dumtreat2 dumtreat3          Y
## 200001    1     1     3      0         0         0         1 -0.8439043
## 200002    1     1     3      0         0         0         1  3.2687880
## 200003    1     1     3      0         0         0         1  5.1045403
## 200004    1     1     3      0         0         0         1  3.9323844
## 200005    1     1     3      0         0         0         1 -0.8597653
## 200006    1     1     3      0         0         0         1  7.6718160
## 200007    1     1     3      0         0         0         1  0.5746743
## 200008    1     1     3      0         0         0         1  1.4134960
## 200009    1     1     3      0         0         0         1  4.2974281
## 200010    1     1     3      0         0         0         1  5.2107459
## 200011    1     1     3      0         0         0         1  2.8431292
## 200012    1     1     3      0         0         0         1  9.4264438
```

Aggregate by group and time

```
datasum <- data.frame(data %>%
            group_by(group, time) %>%
            summarise(meanY = mean(Y), meanYpot = mean(Ypot),
            .groups = "drop")
)
```

View the simulated data

```
datasum
```

```
##    group time     meanY  meanYpot
## 1      1    1  3.961191  3.961191
## 2      1    2  5.059393  5.059393
## 3      1    3  6.064970  6.064970
## 4      1    4  7.077845  7.077845
## 5      1    5  8.057153  8.057153
## 6      1    6 19.099934  9.099934
## 7      1    7 20.019154 10.019154
## 8      1    8 20.993719 10.993719
## 9      1    9 21.960722 11.960722
## 10     1   10 22.964545 12.964545
## 11     1   11 23.934175 13.934175
## 12     1   12 25.070944 15.070944
```

Dimension of the simulated data

- 20 periods and 3 groups

```
dim(datasum)
```

```
## [1] 60  4
```

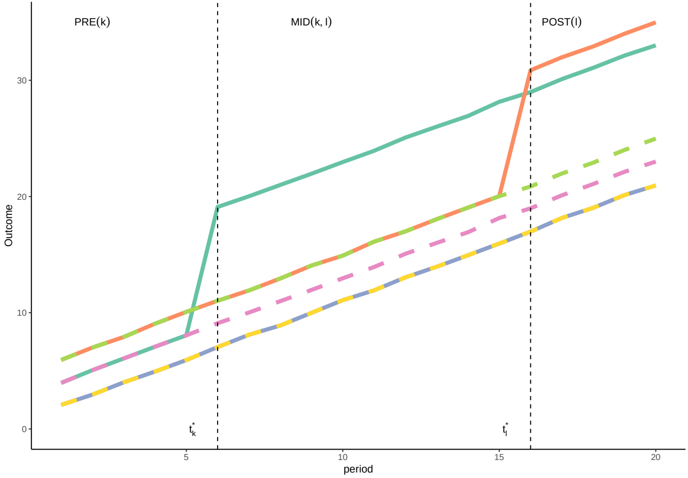
Plot the outcome by group and time

```
# data for potential outcome
datasumpot <- datasum %>% dplyr::select(c(group,
                                          time,
                                          meanYpot)) %>%
                              mutate(group = 10*group) # this would be the gro
                              # 10 for the early group in absence of the treat
                              # 20 for the late group in absence of the treatm
                              # 30 for the untreated group in absence of the t

datasum <- datasum %>% dplyr::select(-c(meanYpot))
colnames(datasumpot) <- c("group", "time", "meanY")

# bind the potential outcome
datasum <- rbind(datasum, datasumpot)
```

Plot

```
vlines <- data.frame(xint = c(6, 16))

datasum$group = factor(datasum$group)

f0  <- ggplot(datasum, aes(x = time, y = meanY, group = group)) +
            geom_line(aes(linetype = group, color = group),size = 2) +
            scale_linetype_manual(name = "Linetype",
                values = c("1" = 1, "2" = 1, "3" = 1, "10" = 2, "20" = 2, "30
            theme(panel.grid.major = element_blank(), panel.grid.minor = eleme
            panel.background = element_blank(), axis.line = element_line(colo
            legend.position = "none") + xlab("period") + ylab("Outcome") +
            geom_vline(data = vlines, aes(xintercept = xint), linetype = "das
            annotate(x = c(5.2, 15.2, 2, 9, 17), y = c(0, 0, 35, 35, 35),
            label = c(bquote(t[k]^"*"), bquote(t[l]^"*"), "PRE(k)", "MID(k,l)
```

Plot



Things to note

- Group 1 is treated at T = 6
- Group 2 is treated at T = 16
- Group 3 is untreated
- True effect is 10
- In other words, treatment effect is homogeneous

Section 3

Estimating TWFE in case of homogeneous treatment effect

Let's use TWFE in this case

```r
# first generate the necessary variables

datasum$treat1  <- ifelse(datasum$group == 1 & datasum$time >= early, 1, 0)
datasum$treat2  <- ifelse(datasum$group == 2 & datasum$time >= late, 1, 0)
datasum$twfe  <- datasum$treat1 + datasum$treat2
```

Group 1

```
##    group time     meanY treat1 treat2 twfe
## 1      1    1  3.961191      0      0    0
## 2      1    2  5.059393      0      0    0
## 3      1    3  6.064970      0      0    0
## 4      1    4  7.077845      0      0    0
## 5      1    5  8.057153      0      0    0
## 6      1    6 19.099934      1      0    1
## 7      1    7 20.019154      1      0    1
## 8      1    8 20.993719      1      0    1
## 9      1    9 21.960722      1      0    1
## 10     1   10 22.964545      1      0    1
## 11     1   11 23.934175      1      0    1
## 12     1   12 25.070944      1      0    1
## 13     1   13 26.008729      1      0    1
## 14     1   14 26.927938      1      0    1
```

Group 2

```
##    group time    meanY treat1 treat2 twfe
## 30     2   10 14.91414      0      0    0
## 31     2   11 16.11539      0      0    0
## 32     2   12 16.99957      0      0    0
## 33     2   13 18.05093      0      0    0
## 34     2   14 19.04691      0      0    0
## 35     2   15 20.03236      0      0    0
## 36     2   16 30.86308      0      1    1
## 37     2   17 31.97818      0      1    1
## 38     2   18 32.90965      0      1    1
## 39     2   19 34.00396      0      1    1
## 40     2   20 34.98348      0      1    1
```

Group 3

```
##    group time     meanY treat1 treat2 twfe
## 41     3    1  2.069679      0      0    0
## 42     3    2  2.954870      0      0    0
## 43     3    3  4.009114      0      0    0
## 44     3    4  4.941765      0      0    0
## 45     3    5  5.927573      0      0    0
## 46     3    6  7.044617      0      0    0
## 47     3    7  8.104446      0      0    0
## 48     3    8  8.901367      0      0    0
## 49     3    9  9.992003      0      0    0
## 50     3   10 11.081951      0      0    0
## 51     3   11 11.927770      0      0    0
## 52     3   12 13.041271      0      0    0
## 53     3   13 13.959321      0      0    0
## 54     3   14 14.946201      0      0    0
```

TWFE specification

```r
reg_twfe  <- lm(meanY ~ twfe + factor(time) + factor(group), data = datasum)

# extract TWFE coefficients for brevity
cat("TWFE estimate: ", coefficients(reg_twfe)[[2]])
```

```
## TWFE estimate:  9.943102
```

```r
cat("Remember the true effect assigned: ", trueeffect)
```

```
## Remember the true effect assigned:  10
```

Section 4

Heterogeneous treatment effects

Heterogeneous treatment effects

- So far, looked at homogeneous treatment effect
- What if treatment effects are heterogeneous?

Two kinds:

1. consider when effects vary by groups
   - Early group, has a treatment effect of 20 and late group of 5
2. effects vary by time
   - dynamic treatment effects

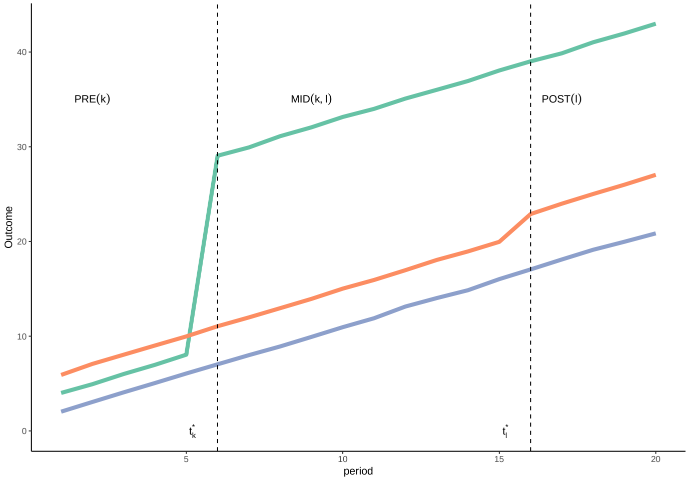
Effects vary by groups

```
trueeffect_early  <- 20 # treatment effect is high for early adopters
trueeffect_late  <- 2   # low for late adopters



# simulate the outcome
e <- rnorm(nrow(data), 0, 5)

data <- data %>% mutate(Y = 1 + trueeffect_early*dumtreat1*policy +
                        trueeffect_late*dumtreat2*policy +
                        time + # common time trend
                        dumtreat1*2 + dumtreat2*4 + # time invariant
                        #unobserved heterogeneity
                        e,  # error term
                        Ypot = 1 +
                        time +
```

Plot

```
vlines <- data.frame(xint = c(6, 16))

datasum$group = factor(datasum$group)

f0  <- ggplot(datasum, aes(x = time, y = meanY, group = group)) +
            geom_line(aes(linetype = group, color = group),size = 2) +
            scale_linetype_manual(name = "Linetype",
                values = c("1" = 1, "2" = 1, "3" = 1, "10" = 2, "20" = 2, "30
            theme(panel.grid.major = element_blank(), panel.grid.minor = eleme
            panel.background = element_blank(), axis.line = element_line(colo
            legend.position = "none") + xlab("period") + ylab("Outcome") +
            geom_vline(data = vlines, aes(xintercept = xint), linetype = "dasl
            annotate(x = c(5.2, 15.2, 2, 9, 17), y = c(0, 0, 35, 35, 35),
            label = c(bquote(t[k]^"*"), bquote(t[l]^"*"), "PRE(k)", "MID(k,l)
```

View the plot



TWFE using all three groups

```r
# first generate the necessary variables

datasum$treat1  <- ifelse(datasum$group == 1 & datasum$time >= early, 1, 0)
datasum$treat2  <- ifelse(datasum$group == 2 & datasum$time >= late, 1, 0)
datasum$twfe  <- datasum$treat1 + datasum$treat2

reg_twfe2  <- lm(meanY ~ twfe + factor(time) + factor(group), data = datasum)

# extract TWFE coefficients for brevity
cat("TWFE estimate: ", coefficients(reg_twfe2)[[2]])
```

```
## TWFE estimate:  11.04637
```

TWFE group 1 (compare with untreated)

```
twfe_group1  <- lm(meanY ~ twfe + factor(time) + factor(group), data = subset
effect_group1  <- coefficients(twfe_group1)[[2]]
# extract TWFE coefficients for brevity
cat("TWFE estimate (early treated only): ", effect_group1)
```

```
## TWFE estimate (early treated only):  20.08684
```

TWFE group 2 (compare with untreated)

```
twfe_group2  <- lm(meanY ~ twfe + factor(time) + factor(group), data = subset
effect_group2  <- coefficients(twfe_group2)[[2]]
# extract TWFE coefficients for brevity
cat("TWFE estimate (late treated only): ", effect_group2)
```

```
## TWFE estimate (late treated only):  1.978138
```

weighted average of effects on group1 and group2

```
num_treat_cell  <- (20 - early + 1) + (20 - late + 1)       # total instance
w_group1  <- (20 - early + 1) / num_treat_cell  # weight for group 1
w_group2  <- (20 - late + 1)  / num_treat_cell  # weight for group 2
cat("weight for early group:", w_group1)
```

```
## weight for early group: 0.75
```

```
cat("weight for late group:", w_group2)
```

```
## weight for late group: 0.25
```

TWFE estimate if far from the ATE estimate

- In case of heterogeneous treatment effect by groups

```
# print ate using weighted version of TWFE estimates
ate  <- w_group1 * effect_group1 + w_group2 * effect_group2
cat("ATE estimate: ", ate)
```

```
## ATE estimate:  15.55966
```

```
# print TWFE with all three groups
cat("TWFE estimate: ", coefficients(reg_twfe2)[[2]])
```

```
## TWFE estimate:  11.04637
```

Effects vary by time

```
# generate year dummies
year_dummies <- model.matrix(~ factor(data$time) - 1)
# True effects for early and late groups
trueeffect_early  <- c(rep(0, (early - 1)), seq(10, length.out = 15))
trueeffect_late  <- c(rep(0, (late - 1)), seq(5, length.out = 5))

# interacting early and late groups with true effects
early_int  <- sweep(year_dummies[1:100000, ], 2, trueeffect_early, "*") # yea
late_int  <- sweep(year_dummies[100001:200000, ], 2, trueeffect_late, "*") #
untreated_int  <- year_dummies[200001:300000, ] * 0

# stacking
stack  <- rbind(early_int, late_int, untreated_int)
colnames(stack)  <- paste0("dumeffect", seq(1, 20, 1))
```

```r
head(stack)
```

```
##   dumeffect1 dumeffect2 dumeffect3 dumeffect4 dumeffect5 dumeffect6 dumeffe
## 1          0          0          0          0          0          0
## 2          0          0          0          0          0          0
## 3          0          0          0          0          0          0
## 4          0          0          0          0          0          0
## 5          0          0          0          0          0          0
## 6          0          0          0          0          0          0
##   dumeffect8 dumeffect9 dumeffect10 dumeffect11 dumeffect12 dumeffect13
## 1          0          0           0           0           0           0
## 2          0          0           0           0           0           0
## 3          0          0           0           0           0           0
## 4          0          0           0           0           0           0
## 5          0          0           0           0           0           0
## 6          0          0           0           0           0           0
##   dumeffect14 dumeffect15 dumeffect16 dumeffect17 dumeffect18 dumeffect19
```

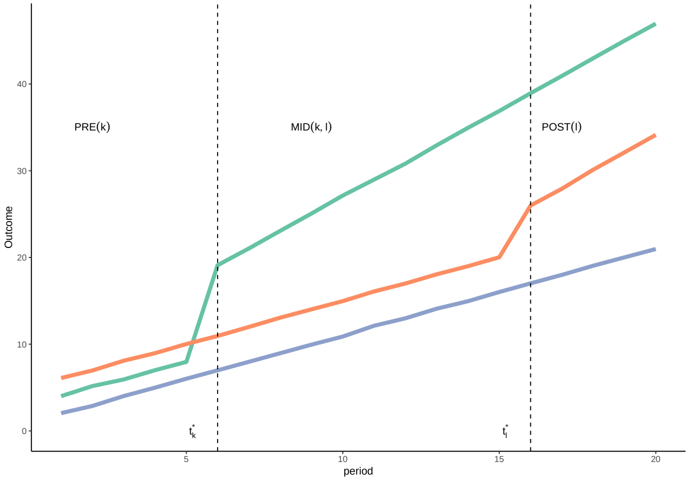
```
stack  <- rowSums(stack)
data$stack  <- stack
```

```
# simulate the outcome
e <- rnorm(nrow(data), 0, 5)

data <- data %>% mutate(Y = 1 + stack +
                        time + # common time trend
                        dumtreat1*2 + dumtreat2*4 + # time invariant
                        #unobserved heterogeneity
                        e,  # error term
                        Ypot = 1 +
                        time +
                        dumtreat1*2 + dumtreat2*4 + e)

# aggregate by group and time
datasum <- data.frame(data %>%
```

Plot

```
f0 <- ggplot(datasum, aes(x = time, y = meanY, group = group)) +
            geom_line(aes(linetype = group, color = group),size = 2) +
            scale_linetype_manual(name = "Linetype",
                values = c("1" = 1, "2" = 1, "3" = 1, "10" = 2, "20" = 2, "30
            theme(panel.grid.major = element_blank(), panel.grid.minor = eleme
            panel.background = element_blank(), axis.line = element_line(colo
            legend.position = "none") + xlab("period") + ylab("Outcome") +
            geom_vline(data = vlines, aes(xintercept = xint), linetype = "dasl
            annotate(x = c(5.2, 15.2, 2, 9, 17), y = c(0, 0, 35, 35, 35),
            label = c(bquote(t[k]^"*"), bquote(t[l]^"*"), "PRE(k)", "MID(k,l)
```

View plot



Run the Twfe

```r
datasum$treat1  <- ifelse(datasum$group == 1 & datasum$time >= early, 1, 0)
datasum$treat2  <- ifelse(datasum$group == 2 & datasum$time >= late, 1, 0)
datasum$twfe  <- datasum$treat1 + datasum$treat2

twfe  <- lm(meanY ~ twfe + factor(time) + factor(group), data = datasum)

# extract TWFE coefficients for brevity
cat("TWFE estimate: ", coefficients(twfe)[[2]])
```

```
## TWFE estimate:  10.01283
```

Section 5

Event study models

Event study specification

$$Y_{it} = \alpha + \sum_{j \in -5}^{10} \beta_j I(t - \tilde{(}t) = j) + \sigma_s + \eta_t + \epsilon_{it}$$

Event study: A handmade approach

```
# assign group time
datasum$group_time  <- ifelse(datasum$group == 1, 6, 0)
datasum$group_time  <- ifelse(datasum$group == 2, 16, datasum$group_time)
# generate policy and treatment dummies
datasum$policy  <- ifelse(datasum$group == 1 & datasum$time >= early, 1, 0)
datasum$policy  <- ifelse(datasum$group == 2 & datasum$time >= late, 1, datasu
datasum$treat  <- ifelse(datasum$group == 1 | datasum$group == 2, 1, 0)
# generate relative time dummies
datasum$r_time  <- datasum$time - datasum$group_time
# bound the relative time (lower -5 and upper 10)
datasum$r_time  <- ifelse(datasum$r_time < -5, -5, datasum$r_time)
datasum$r_time  <- ifelse(datasum$r_time > 10, 10, datasum$r_time)

#datasum$r_time  <- ifelse(datasum$group == 3, 999, datasum$r_time)
```

```r
# generate relative time dummies
r_time_dummies  <- model.matrix(~ factor(datasum$r_time) - 1)
# interact relative time with treatment indicator
r_time_dummies  <- apply(r_time_dummies, 2, function(x) x * datasum$treat)
colnames(r_time_dummies)  <- c(paste0("nr.treat", seq(5, 1, -1)), paste0("r.tr
# bind with datasum
datasum   <- cbind(datasum, r_time_dummies)
```

Set the event study specification

```r
# set a regression formula
# omit the year prior to the policy "nr.treat1"
form  <- meanY ~ nr.treat5+nr.treat4+nr.treat3+nr.treat2+ r.treat0 +
                  r.treat1+r.treat2+r.treat3+r.treat4+r.treat5+
                  r.treat6+r.treat7+r.treat8+r.treat9+r.treat10 +
                  factor(time) + factor(group)

es_reg  <- lm(form, data = datasum)
```

Display results

```r
coefficients(es_reg)[2:16]
```

```
##   nr.treat5   nr.treat4   nr.treat3   nr.treat2    r.treat0    r.treat1
##  1.11009631  0.18055505 -0.05163029  0.06616340  8.45746110  9.14516688
##    r.treat2    r.treat3    r.treat4    r.treat5    r.treat6    r.treat7
##  9.96855154 10.73144106 11.57607619 13.93340013 14.42337881 15.33796067
##    r.treat8    r.treat9   r.treat10
## 16.51247626 17.37309696 21.91729555
```

Compare results with feols

```
library(fixest)
mod  <- feols(meanY ~ i(r_time, treat, (-1)) | time + group , data = datasum)
```

Display results from feols

```
mod
```

```
## OLS estimation, Dep. Var.: meanY
## Observations: 60
## Fixed-effects: time: 20,  group: 3
## Standard-errors: Clustered (time)
##                     Estimate Std. Error   t value   Pr(>|t|)
## r_time::-5:treat  1.110096   1.160272  0.956756 3.5071e-01
## r_time::-4:treat  0.180555   1.306581  0.138189 8.9155e-01
## r_time::-3:treat -0.051630   1.400129 -0.036875 9.7097e-01
## r_time::-2:treat  0.066163   1.377438  0.048034 9.6219e-01
## r_time::0:treat   8.457461   1.244774  6.794373 1.7324e-06 ***
## r_time::1:treat   9.145167   1.352391  6.762221 1.8487e-06 ***
## r_time::2:treat   9.968552   1.432750  6.957636 1.2483e-06 ***
## r_time::3:treat  10.731441   1.568471  6.841977 1.5739e-06 ***
```

Plot event study coefficients

```
coefplot(mod)
```