Demo Class 3

This is the code to implement the work for Demo Class 3

- estimate several TWFE models
- apply cluster robust standard errors
- collapse data to a before/after panel

Introduction

We are going to estimate the following two models

$y_{st} = \alpha_s + \lambda_2 f_t^2 + \lambda_3 f_t^3 + \lambda_4 f_t^4 + \tau d_s p_t + u_{st}$ (R) for $t = 1, ..., 4$

and

$y_{st} = \alpha_s + \lambda_2 f_t^2 + \lambda_3 f_t^3 + \lambda_4 f_t^4 + \omega_2 d_s f_t^2 + \tau_3 d_s f_t^3 + \tau_4 d_s f_t^4 + u_{st}$ (U) for $t = 1, ..., 4$

Estimating model (R) by OLS generates the TWFE estimator of the treatment effect τ. However one should also model (U) to test for common trends and investigate heterogenous treatment effects. Model (U) is labeled an events-study model.

Preparing your workfile

We add the basic libraries needed for this week's work:

```
library(tidyverse)    # for almost all data handling tasks
library(ggplot2)      # to produce nice graphics
library(stargazer)    # to produce nice results tables
library(haven)        # to import stata file
library(ggplot2)      # for graphs
library(AER)          # access to HS robust standard errors
library(plm)          # for panel data methods
library(sandwich)     # for cluster robust se
library(lmtest)
library(coefplot)     # to create coefficient plots
```

You should also save the separately supplied `stargazer_HC.r` file in your working directory. This will make it straightforward to estimate and compare regressions with robust standard errors. Once you have done that you should include the following line into your code which basically makes this function available to you.

```
source("stargazer_HC.r")  # includes the robust regression
```

This has worked if you can see it loaded into your environment as a function.

Data Prep

Read the data.

```
data <- read_dta("did_4.dta")
data <- as.data.frame(data)
```

Let's look at the data file.

```
str(data)
names(data)
summary(data)
```

We will convert some variables to factor (categorical) variables

```
data$year <- as_factor(data$year)
data$w <- as_factor(data$w)
data$d <- as_factor(data$d)
levels(data$d) <- c("control","treated")
data$id <- as_factor(data$id)
```

Investigate the Panel Structure

Let's define the dataset as a panel dataset with `id` as the cross-sectional identifier and `year` as the time identifier.

```
pdata <- pdata.frame(data, index = c("id","year")) # defines the panel dimensions
```

The `plm` library we imported has a useful little function to check whether the panel is balanced.

```
is.pbalanced(pdata)
```

```
## [1] TRUE
```

This has returned `TRUE` indicating that it is indeed balanced. As there are four years of data, this means that we have 513 units of observation (4 x 513 = 2052).

Let's look again at the summary statistics for the variables `w` "treated in a particular year" and `d` "ever treated".

```
summary(data[,c("w","d")])
```

```
##   w              d
##   0:1784   control:1516
##   1: 268   treated: 536
```

You can see that 536 observations belong to individuals ever treated. As we have four years of observations for each individual this implies that $S_1 = 134$ individuals were ever treated. The remainder, $S_0 = 379$ is the size of the control group. The number of observations in treatment are only 268. Exactly half. This is best understood if we look at the data for one of the observations in the treatment group (`id=3591`):

```
pdata[data$id==3591,c("id","year","w","d","post")]
```

```
##             id year w       d post
## 3591-2012 3591 2012 0 treated    0
## 3591-2013 3591 2013 0 treated    0
## 3591-2014 3591 2014 1 treated    1
## 3591-2015 3591 2015 1 treated    1
```

You can see that this individual was treated in two of the years (2014 and 2015). This is the same for all treated individuals. The variable `w` is therefore the equivalent to the "TREATxPOST" or here `d*post` variable.

Estimate the TWFE model (R)

Let us estimate the TWFE model but when we output the result we shall only show the coefficient on `w`, our treated variable.

```
mod1 <- lm(logy~id+year+w, data = pdata)
mod1_ro_se <- sqrt(diag(vcovHC(mod1, type = "HC1")))  # robust standard errors
stargazer(mod1, keep = "w", type="text", se=list(mod1_ro_se),
          digits = 6, notes="HS Robust standard errors in parenthesis")
```

```
## 
## ============================================================
##                                   Dependent variable:       
##                     ----------------------------------------
##                                           logy              
## ------------------------------------------------------------
## w1                                0.185928***               
##                                    (0.019743)               
##                                                             
## ------------------------------------------------------------
## Observations                         2,052                  
## R2                                  0.968607                
## Adjusted R2                         0.958053                
## Residual Std. Error           0.199563 (df = 1535)          
## F Statistic               91.783900*** (df = 516; 1535)     
## ============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
##                     HS Robust standard errors in parenthesis
```

If you want to estimate cluster (here by `id`) robust standard errors we use the following function

```
mod1_cr_se <- sqrt(diag(vcovCL(mod1, cluster = ~ id)))
stargazer(mod1, keep = "w", type="text", se=list(mod1_cr_se),
          digits = 6, notes="Cluster Robust standard errors in parenthesis")
```

```
## 
## =================================================================
##                                      Dependent variable:         
##                     ---------------------------------------------
##                                             logy                 
## -----------------------------------------------------------------
## w1                                   0.185928***                 
##                                      (0.021322)                  
##                                                                  
## -----------------------------------------------------------------
## Observations                            2,052                    
## R2                                     0.968607                  
## Adjusted R2                            0.958053                  
## Residual Std. Error              0.199563 (df = 1535)            
## F Statistic                  91.783900*** (df = 516; 1535)       
## =================================================================
## Note:                                 *p<0.1; **p<0.05; ***p<0.01
##                     Cluster Robust standard errors in parenthesis
```

You can see that there is a small difference in the standard error.

Let us now replace the `id`-level fixed effect by merely adding the ever treated dummy `d`

```
mod2 <- lm(logy~year+d+w, data = pdata)
mod2_cr_se <- sqrt(diag(vcovCL(mod2, cluster = ~ id)))
stargazer(mod2, type="text", se=list(mod2_cr_se), digits = 6)
```

```
## 
## ===============================================
##                         Dependent variable:    
##                     ---------------------------
##                                logy            
## -----------------------------------------------
## year2013                     0.018548          
##                             (0.012448)         
##                                                
## year2014                    0.054975***        
##                             (0.013848)         
##                                                
## year2015                   -0.037914***        
##                             (0.013230)         
##                                                
## dtreated                   -0.344730***        
##                             (0.090445)         
##                                                
## w1                          0.185928***        
##                             (0.018469)         
##                                                
## Constant                    2.502501***        
##                             (0.051841)         
##                                                
## -----------------------------------------------
## Observations                   2,052           
## R2                           0.016432          
## Adjusted R2                  0.014028          
## Residual Std. Error    0.967528 (df = 2046)    
## F Statistic         6.836235*** (df = 5; 2046) 
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

As this model only estimates 6 parameters we can actually look at all estimated coefficients. The standard errors are incorrect as we are actually estimating $S + T - 1 + 1 = 517$ coefficients. The correct standard errors are the ones from `mod1`.

Estimate the events study model (U)

Now we create interactions between the ever treated variable `d` and the years. In order to understand what the following regression does we will actually calculate new variables into the dataset.

```
pdata <- pdata %>%  mutate(d2013 = (year=="2013")*(d=="treated"),
                           d2014 = (year=="2014")*(d=="treated"),
                           d2015 = (year=="2015")*(d=="treated"))
```

Now we estimate the extended TWFE model. First with the individual fixed effects included, producing the correct standard errors.

```
mod3 <- lm(logy~id+year+d2013+d2014+d2015, data = pdata)
mod3_cr_se <- sqrt(diag(vcovCL(mod3, cluster = ~ id)))
```

```
coef_keep = c("year","d2013","d2014","d2015")
stargazer(mod3, type="text", keep = coef_keep, se=list(mod3_cr_se), digits = 6)
```

```
##
## =================================================
##                           Dependent variable:
##                      -----------------------------
##                                  logy
## -------------------------------------------------
## id2014                        -1.662853***
##                                (0.000000)
##
## year2013                        0.021044
##                                (0.016954)
##
## year2014                       0.055244***
##                                (0.017265)
##
## year2015                       -0.035688**
##                                (0.016344)
##
## d2013                           -0.009554
##                                (0.031916)
##
## d2014                          0.184897***
##                                (0.032225)
##
## d2015                          0.177406***
##                                (0.031101)
##
## -------------------------------------------------
## Observations                      2,052
## R2                              0.968610
## Adjusted R2                     0.958004
## Residual Std. Error       0.199681 (df = 1533)
## F Statistic         91.321650*** (df = 518; 1533)
## =================================================
## Note:                 *p<0.1; **p<0.05; ***p<0.01
```

The first coefficient `id2014` is a randomly selected fixed effect coefficient. There are 513 of these but we only show one of them.

Now using the algebraic trick (but incorrect standard errors)

```
mod4 <- lm(logy~year+d+d2013+d2014+d2015, data = pdata)
mod4_cr_se <- sqrt(diag(vcovCL(mod4, cluster = ~ id)))
stargazer(mod4, type="text", se=list(mod4_cr_se), digits = 6)
```

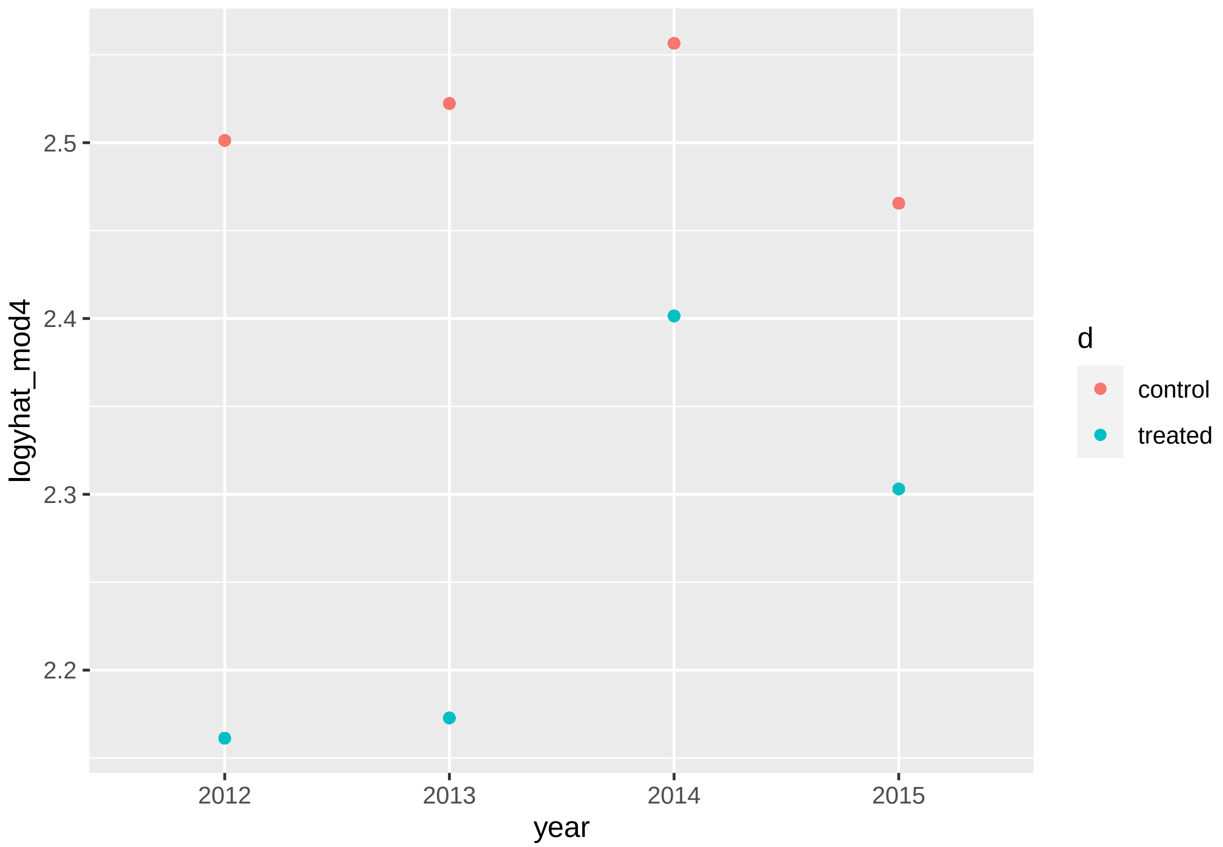
```
##
## ===============================================
##                          Dependent variable:
##                      ---------------------------
##                                 logy
## -----------------------------------------------
## year2013                      0.021044
##                              (0.014682)
##
## year2014                    0.055244***
##                              (0.014952)
##
## year2015                    -0.035688**
##                              (0.014154)
##
## dtreated                   -0.339953***
##                              (0.091424)
##
## d2013                        -0.009554
##                              (0.027640)
##
## d2014                       0.184897***
##                              (0.027908)
##
## d2015                       0.177406***
##                              (0.026934)
##
## Constant                    2.501253***
##                              (0.051993)
##
## -----------------------------------------------
## Observations                   2,052
## R2                           0.016436
## Adjusted R2                  0.013067
## Residual Std. Error    0.967999 (df = 2044)
## F Statistic         4.879383*** (df = 7; 2044)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

From the last model we can get the fitted values.

```
pdata$logyhat_mod4 <- mod4$fitted.values
```

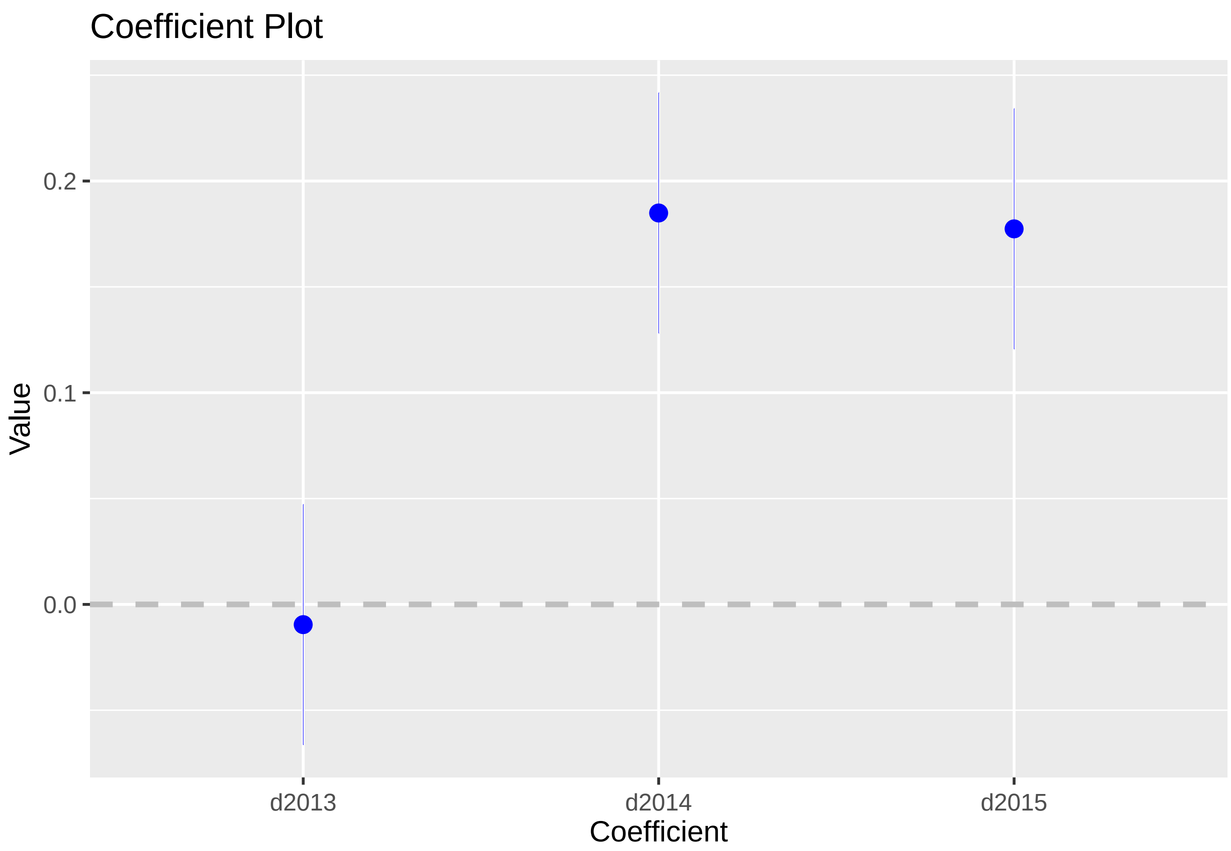
Let us plot the predicted `logyhat`, separate for the treatment and control group. We use the second version, as it basically averages across individuals in year/treatment groups.

```
p1 <- ggplot(pdata,aes(x=year,y=logyhat_mod4,color=d)) + geom_point()
p1
```



The most common way to display these results is by showing the coefficients of the `d` variable interacted with the years.

```
coefplot(mod3, coefficients = coef_keep, innerCI = 0, horizontal = TRUE)
```



Collapse the data to group means

```
data_collapse <- pdata %>% group_by(id,post) %>%
                           summarise(logym = mean(logy),
                                     dm = mean(d=="treated"),
                                     wm = mean(w=="1")) %>%
                           print()
```

```
## # A tibble: 1,026 x 5
## # Groups:   id [513]
##    id     post logym    dm    wm
##    <fct> <dbl> <dbl> <dbl> <dbl>
##  1 13        0  4.12     0     0
##  2 13        1  4.34     0     0
##  3 17        0  4.17     0     0
##  4 17        1  3.90     0     0
##  5 18        0  1.42     0     0
##  6 18        1  1.25     0     0
##  7 45        0  2.67     0     0
##  8 45        1  2.30     0     0
##  9 110       0  2.16     0     0
## 10 110       1  2.37     0     0
## # ... with 1,016 more rows
```

We now only have two observations for every id, pre and post.

Let's calculate the outcome average by pre- and post and treatment and control group.

```
Tab1 <- data_collapse %>% group_by(post,dm) %>%
            summarise(m = mean(logym)) %>% spread(dm,m) %>% print()
```

```
## # A tibble: 2 x 3
## # Groups:   post [2]
##    post   `0`   `1`
##   <dbl> <dbl> <dbl>
## 1     0  2.51  2.17
## 2     1  2.51  2.35
```

The Diff-in-Diff estimator can be calculated from here

$\hat{\tau} = (\bar{y}_T^{post} - \bar{y}_T^{pre}) - (\bar{y}_C^{post} - \bar{y}_C^{pre})$

Plugging in the above values you get (2.352229-2.167045)-(2.511031-2.511775) = 0.185928.

You can see that this is the same as the estimate from `mod1`. However, here we got it from averaged data. When you calculate the estimate as done here from the collapsed data averages you do not get a standard error. However, we can use the collapsed data in a regression

Before proceeding we define this collapsed data set as a panel data-set.

```
# defines the panel dimensions
pdata_collapse<- pdata.frame(data_collapse, index = c("id","post"))
# We now add the differenced y series
pdata_collapse$dy <- diff(pdata_collapse$logym)
```

Now estimate simple first difference regressions

```
mod_fd1 <- plm(dy ~ dm, model = "pooling", data = pdata_collapse)
mod_fd2 <- plm(dy ~ wm, model = "pooling", data = pdata_collapse)
stargazer(mod_fd1,mod_fd2, type="text", digits = 6)
```

```
##
## ======================================================
##                               Dependent variable:
##                           ----------------------------
##                                        dy
##                                (1)            (2)
## ------------------------------------------------------
## dm                         0.185928***
##                             (0.019410)
##
## wm                                        0.185928***
##                                           (0.019410)
##
## Constant                    -0.000744      -0.000744
##                             (0.009920)    (0.009920)
##
## ------------------------------------------------------
## Observations                   513            513
## R2                           0.152231      0.152231
## Adjusted R2                  0.150572      0.150572
## F Statistic (df = 1; 511)  91.758530***  91.758530***
## ======================================================
## Note:                      *p<0.1; **p<0.05; ***p<0.01
```

As you can see these are identical regressions. As we are using only the post period data (the pre data gets lost from the differencing) both `wm` and `dm` give the same value (1 for treated and 0 for non-treated).

The advantage of estimating DiD estimate by a regression is that you get standard errors which allow you to do inference. Here the standard error is 0.0194.