# Stats 1B – Introduction to Causal Analysis
## LAB SESSION #01
Differences-in-Differences

Revised May 9, 2025

### Introduction

In this session, we will perform some very basic "differences-in-differences" analysis (DiD). The data are redacted from what Julia Brandão used in her master's thesis and consist of municipality level observations of standardized test scores from 2005, 2007, 2009, and 2011, as well as several other variables also measured at the level of the municipality. The data include all municipalities in Ceará and neighboring states, and have been modified from the original source for instructional purposes.

The goal is to use these data to assess impacts on score tests of a change in the way tax revenue were distributed on educational performance (measured by municipality/year). This change was enacted in 2008 only in Ceará, and conditioned a significant portion of tax transfers from the state to the municipalities on educational indicators of performance. You will only use municipalities on either side of Ceará's border in this analysis.

### The Data

The dataset used in this session and assignment are provided in "wide" format (each municipality appears in one row and multiple observations for each municipalities are represented in additional columns), and consists of the following variables:

- ibgecode: municipality code
- mun: municipality name
- uf: state
- divregiao: dummy = 1 if municipality is in a border micro-region (shaded area in map)
- distcap: distance from capital city (in 10kms)
- mtmYYYY: math scores in year YYYY
- lpYYYY: Portuguese scores in year YYYY
- idebYYYY: ideb composite score in year YYYY
- pibpcYYYY: gdp per capita in year YYYY
- popYYYY: population in year YYYY
- idhYYYY: human development index in year YYYY

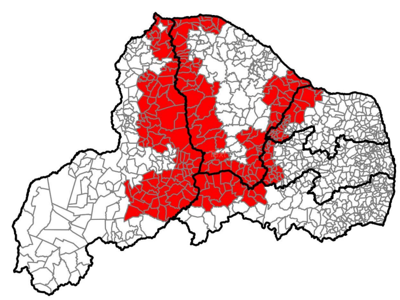


Figure 1: Ceará and its neighbors

## Lab Instructions

### Step 0: Load libraries

Before anything, load some libraries. For our labs, it's always a good idea to load *tidyverse*, *texreg*, and *estimatr*, as they will always be used.

```
library(tidyverse)  # plotting, data cleaning/processing functions
library(estimatr)   # robust linear models
library(texreg)     # display regressions
```

### Step 1: Load, Review and Organizing the Data

Your first step should always be loading the data, taking a glimpse at it, and making small tweaks before actually interacting with it.

To load the Lab data, download the lab file (*lab01-2023-DATAtwoperiod.csv*) to your computer from E-Class. Put it in a folder that you can identify easily. The best folder is always the same folder that your R Project is in (i.e. your "directory"). If that is the case, you can simply provide the name of the file when loading it in R. If you prefer to save the file somewhere else, make sure to copy/paste the exact and complete pathway.

In this example, I placed my file within a folder inside my directory called "Lab 1":

```
# In your computer:
data <- read.csv("data/lab01-2023-DATAtwoperiod.csv")
```

Once the file is loaded (it pops up in your "Global Environment"), take a glimpse of what the data looks like. Don't forget to load the packages, as glimpse() is inside the *tidyverse*:

```
glimpse(data)
```

```
## Rows: 925
## Columns: 16
## $ ibgecode    <int> 2200103, 2200202, 2200251, 2200277, 2200301, 2200400, 2200~
## $ mun         <chr> "AGRICOLANDIA", "AGUA BRANCA", "ALAGOINHA DO PIAUI", "ALEG~
## $ uf          <chr> "PI", "PI", "PI", "PI", "PI", "PI", "PI", "PI", "PI", "PI"~
## $ div.regiao  <lgl> FALSE, FALSE, TRUE, TRUE, TRUE, FALSE, FALSE, FALSE, FALSE~
## $ distcap     <int> 80, 91, 296, 321, 68, 38, 384, 128, 110, 454, 160, 176, 21~
## $ ideb.2007   <dbl> 2.5, 4.2, 3.1, 3.7, 2.8, 3.2, 2.5, 2.6, 4.2, 2.5, 3.8, 2.6~
## $ ideb.2009   <dbl> 3.8, 4.0, 2.9, 3.2, 3.1, 3.8, 3.5, 2.8, 3.7, 3.0, 3.8, 2.5~
## $ mtm.2007    <dbl> 178.36, 200.76, 177.62, 196.34, 165.15, 176.09, 163.78, 15~
## $ mtm.2009    <dbl> 199.48, 195.78, 166.74, 179.58, 175.67, 187.56, 171.81, 16~
## $ lp.2007     <dbl> 147.05, 190.55, 168.79, 172.23, 153.90, 163.31, 142.72, 13~
## $ lp.2009     <dbl> 168.82, 174.03, 154.17, 161.64, 149.76, 170.22, 148.01, 15~
```

```
## $ pop.2007    <int> 4980, 15887, 7447, 4482, 13612, 38328, 4976, 17316, 6609, ~
## $ Dpibpc.0007 <dbl> -1.2695565, 4.5713747, 6.0626337, 7.4256281, 1.7024275, 0.~
## $ pibpc.2007  <dbl> 0.4959746, 0.3152169, 0.4491912, 1.2058777, 4.9633680, 5.3~
## $ pibpc.2009  <dbl> 0.5494404, 0.3284275, 0.5330582, 1.5477571, 5.1741514, 5.5~
## $ idh.2000    <dbl> 0.447, 0.506, 0.368, 0.431, 0.367, 0.424, 0.403, 0.460, 0.~
```

Most of the data looks fine, but we do not want the variable *ibgecode* as a number since it does not represent a quantity but rather a unique code to identify a municipality. Change that column to a character vector.

```r
data$ibgecode <- as.character(data$ibgecode)
```

Other than that, variables that we would predict to be numeric are "dbl" (i.e. double, another word for numeric), same applies for character (chr) and logical (lgl) vectors.

#### Step 2: Table of observations

Now we can address our research concerns. First, check how many municipalities are in the dataset by creating a table of observations by state and by whether municipality is on the border between two states.

```r
# Total observations by state
table(data$uf, data$div.regiao)
```

```
##
##      FALSE TRUE
##   CE    94   89
##   PB   181   23
##   PE   166   17
##   PI   116   79
##   RN   126   34
```

Always stop to reflect: is there a problem with my sample? Do I need to do anything before running some analyses?

In this case, we need to subset the data to only municipalities in the border. In addition, our DiD requires that we have two groups, CE and non-CE, not five separate states.

#### Step 3: Subset data to use only "border" municipalities

Start by filtering the data.

```r
data <- filter(data, div.regiao == T)  # the filter() function keep rows that match a conditio
```

### Step 4: Create a dummy variable for municipalities in Ceará

Then, create a dummy variable indicating whether a given municipality is in Ceará. This works as our *group* variable, with 0 being control and 1 being treatment.

```r
data$group <- ifelse(data$uf == "CE", 1, 0)
```

### Step 5: Compute DiD manually - This is for understanding only!!!

Now, we can already compute our DiD! Let's do it "by hand" first.

Calculate the average IDEB scores for municipalities in CE ($group == 1$) and non-CE ($group == 0$) for both 2007 ($t_0$) and 2009 ($t_1$). Since our treatment occured in 2008, we can think about 2007 as our *pre* and 2009 as our *post* treatment observations.

In short, the DiD is the difference between control and treatment groups and between the pre and post periods.

```r
# Average IDEB non-CE in 2007
mean(data$ideb.2007[data$group == 0])
```

```
## [1] 3.09085
```

```r
# Average IDEB non-CE in 2009
mean(data$ideb.2009[data$group == 0])
```

```
## [1] 3.482353
```

```r
# Average IDEB in CE in 2007
mean(data$ideb.2007[data$group == 1])
```

```
## [1] 3.406742
```

```r
# Average IDEB in CE in 2009
mean(data$ideb.2009[data$group == 1])
```

```
## [1] 3.997753
```

```r
# Difference non-CE pre/post
mean(data$ideb.2009[data$group == 0]) - mean(data$ideb.2007[data$group == 0])
```

```
## [1] 0.3915033
```

```r
# Difference CE pre/post
mean(data$ideb.2009[data$group == 1]) - mean(data$ideb.2007[data$group == 1])
```

```
## [1] 0.5910112
```

```r
# Separate Differences
diff1 <- (mean(data$ideb.2009[data$group == 1]) - mean(data$ideb.2007[data$group == 1]))
diff2 <- (mean(data$ideb.2009[data$group == 0]) - mean(data$ideb.2007[data$group == 0]))

# Differences-in-Differences
diff_diff <- diff1 - diff2

diff_diff
```

```
## [1] 0.199508
```

This value means that, between 2007 and 2009, municipalities in Ceará increased their IDEB score by 0.2 points more than those outside Ceará over the same period.

Now, let's do this same process using linear regression and adding different controls.

#### Step 6: Reshape data into "long" format

In data science, most analyses are done with the data organized in a "long" format—meaning that each variable is represented in just one column. Thus, instead of having a column for *ideb.2007* and another for *ideb.2009*, we should have one column *year* (which is either 2007 and 2009) and another column with the *ideb* score.

This makes it so that each municipality now appears in two separate rows. One row represents the IDEB score (and all of our other variables) in 2007, and the other one in 2009.

Follow this logic and reshape the dataset so that you have a "long" format with: (1) the dependent variable IDEB, (2) an indicator for belonging to the treatment group (group), and (3) indicators for time (dummy for 2007 or 2009). Note that because you're working with two times only, you may (or may not) create a single indicator for time, which you can call post.

```r
# reshape data
data_long <- reshape(data = data,
                     varying = c(6:11, 14,15),           # columns we want to transform
```

```r
                     direction = "long",                    # we want long format
                     sep = ".",                             # variable names separated by .
                     idvar = c("ibgecode","mun","uf"))     # variables we keep constant

# create variable "post"
data_long$post <- ifelse(data_long$time == 2009, 1, 0)
```

#### Step 7: Estimate DiD by regression

Now, estimate the DiD model by regression and compare this to what was obtained by hand.

For this step we will use the lm_robust() function of the *estimatr* package. To better visualize the regression, use the screenreg() function of the *texreg* package.

Estimate a few variants of the basic model.

1. A basic model.

$$Y = \beta_0 + \beta_1 group_i + \beta_2 post_t + \beta_3 group_i * post_t + \epsilon_{it}$$

```r
m1 <- lm_robust(ideb ~ group + post + group*post,
          data = data_long)

screenreg(m1,
          custom.model.names = "m1")
```

```
##
## =========================
##              m1
## -------------------------
## (Intercept)    3.09 *
##              [3.02; 3.16]
## group          0.32 *
##              [0.20; 0.43]
## post           0.39 *
##              [0.27; 0.51]
## group:post     0.20 *
##              [0.01; 0.39]
## -------------------------
## R^2            0.27
## Adj. R^2       0.26
## Num. obs.    484
## RMSE           0.52
## =========================
## * 0 outside the confidence interval.
```

Notice how you can substitute the interaction term $group_i * post_t$ by a single variable that represents whether a unit was treated or not. You can create this by doing exactly what we did in the function: multiply *group* by *post*. The results will always be the same.

$$Y = \beta_0 + \beta_1 group_i + \beta_2 post_t + \beta_3 treat_{it} + \epsilon_{it}$$

```
data_long$treat <- data_long$group * data_long$post

m1_2 <- lm_robust(ideb ~ group + post + treat,
          data = data_long)

screenreg(m1_2,
          custom.model.names = "m1.2")
```

```
##
## =========================
##              m1.2
## -------------------------
## (Intercept)    3.09 *
##              [3.02; 3.16]
## group          0.32 *
##              [0.20; 0.43]
## post           0.39 *
##              [0.27; 0.51]
## treat          0.20 *
##              [0.01; 0.39]
## -------------------------
## R^2            0.27
## Adj. R^2       0.26
## Num. obs.    484
## RMSE           0.52
## =========================
## * 0 outside the confidence interval.
```

Most academic papers simplify their regressions using a treatment variable.

2. A model with control for pre-treatment variables (non-changing or slow changing), with and without clustering standard errors by municipality.

   Why clusters?
   We have two years of data for each municipality, so the observations are not independent; this affects the standard-error calculation, and clustering by municipality adjusts the error bars so we don't underestimate them.

   Why se_type = "stata"
   It uses Stata's cluster-robust formula so the standard errors match those computed in Stata.

```r
# without clustered SEs
m2_1 <- lm_robust(ideb ~ group + post + treat + distcap,
                  data = data_long)

# with clustered SEs
m2_2 <- lm_robust(ideb ~ group + post + treat + distcap,
                  clusters = mun,
                  se_type = "stata",
                  data = data_long)

screenreg(list(m2_1, m2_2),
          custom.model.names = c("m2.1", "m2.2"))
```

```
##
## ===========================================
##              m2.1            m2.2
## -------------------------------------------
## (Intercept)    3.26 *          3.26 *
##              [ 3.11;  3.41]  [ 3.08;  3.43]
## group          0.30 *          0.30 *
##              [ 0.18;  0.42]  [ 0.18;  0.42]
## post           0.39 *          0.39 *
##              [ 0.27;  0.51]  [ 0.30;  0.49]
## treat          0.20 *          0.20 *
##              [ 0.01;  0.39]  [ 0.07;  0.33]
## distcap       -0.00 *         -0.00 *
##              [-0.00; -0.00]  [-0.00; -0.00]
## -------------------------------------------
## R^2            0.28            0.28
## Adj. R^2       0.27            0.27
## Num. obs.    484             484
## RMSE           0.52            0.52
## N Clusters                   241
## ===========================================
## * Null hypothesis value outside the confidence interval.
```

3. A model that controls for a time-varying variable *ln_pibpc*, with and without clustering standard errors by municipality.

The histogram shows that `pibpc` is strongly right-skewed. Applying a log transformation reduces this skewness, making the variable more normally distributed and better suited for linear regression. This also helps meet model assumptions and allows for easier interpretation of coefficients in terms of percentage changes."

```r
# hist(data_long$pibpc)

# Let's apply a log transformation to the pibpc variable and create a new ln_pibpc variable.
```

```r
data_long$ln_pibpc <- log(data_long$pibpc)

# hist(data_long$ln_pibpc)
```

```r
m3_1 <- lm_robust(ideb ~ group + post + treat + ln_pibpc,
          data = data_long)

m3_2 <- lm_robust(ideb ~ group + post + treat + ln_pibpc,
          clusters = mun,
          se_type = "stata",
          data = data_long)

screenreg(list(m3_1, m3_2),
          custom.model.names = c("m3.1", "m3.2"))
```

```
##
## =========================================
##              m3.1           m3.2
## -----------------------------------------
## (Intercept)    3.09 *         3.09 *
##              [ 3.02; 3.17]  [ 3.02; 3.17]
## group          0.27 *         0.27 *
##              [ 0.15; 0.40]  [ 0.14; 0.40]
## post           0.39 *         0.39 *
##              [ 0.27; 0.51]  [ 0.30; 0.48]
## treat          0.20 *         0.20 *
##              [ 0.01; 0.39]  [ 0.07; 0.33]
## ln_pibpc       0.03           0.03
##              [-0.00; 0.07]  [-0.01; 0.07]
## -----------------------------------------
## R^2            0.27           0.27
## Adj. R^2       0.26           0.26
## Num. obs.    480            480
## RMSE           0.52           0.52
## N Clusters                  239
## =========================================
## * Null hypothesis value outside the confidence interval.
```

#### Step 8: Estimate models with unit fixed-effects

4. A model with explicit municipality dummies for fixed-effects and no controls.

When we use **unit fixed effects**, we're estimating how the outcome (IDEB) changes *within* each municipality (`ibgecode`) over time. This effectively means controlling for all time-invariant differences across municipalities—so the `group` variable becomes redundant, as it's absorbed by the fixed effects. We still include the `treat` variable, because it captures the *within-municipality* variation over time (i.e., who was treated post-intervention).

To implement unit fixed effects, we use `0 + ibgecode` in the regression formula to remove the intercept. Removing the intercept allows us to include a dummy for **each municipality**, estimating each one relative to zero instead of relative to a base category.

We also include **time fixed effects**—in this case, just a dummy for the `post` period—so we're controlling for any period-specific shocks that affect all municipalities equally.

```
m4_1 <- lm_robust(ideb ~ 0 + ibgecode + post + treat,
          data = data_long)
```

Notice that when you print the regression there will be a coefficient for every ibgecode:

Each coefficient represents the average ideb for that municipality when *post* == 0 (i.e, before the treatment).

```
screenreg( m4_1,
          custom.model.names = "m4.1")
```

```
##
## ==============================
##                   m4.1
## ------------------------------
## ibgecode2200251    2.80 *
##                  [2.22; 3.39]
## ibgecode2200277    3.25 *
##                  [2.37; 4.14]
## ibgecode2200301    2.75 *
##                  [2.65; 2.86]
## ibgecode2200905    3.60 *
##                  [3.21; 3.99]
## ibgecode2201051    2.20 *
##                  [2.16; 2.25]
## ibgecode2201176    3.00 *
##                  [2.96; 3.05]
## ibgecode2201556    3.20 *
##                  [2.80; 3.61]
## ibgecode2201572    3.45 *
##                  [3.15; 3.76]
## ibgecode2201739    3.05 *
##                  [2.75; 3.36]
## ibgecode2201945    3.05 *
##                  [2.95; 3.16]
## ibgecode2202000    2.90 *
##                  [2.71; 3.10]
## ibgecode2202026    3.95 *
```

```
##                    [1.87; 6.04]
## ibgecode2202083    2.80 *
##                    [2.22; 3.39]
## ibgecode2202091    3.70 *
##                    [2.92; 4.49]
## ibgecode2202109    2.60 *
##                    [2.00; 3.21]
## ibgecode2202117    2.55 *
##                    [2.07; 3.04]
## ibgecode2202133    3.45 *
##                    [2.97; 3.94]
## ibgecode2202208    3.80 *
##                    [3.20; 4.41]
## ibgecode2202406    3.15 *
##                    [2.65; 3.66]
## ibgecode2202455    3.10 *
##                    [2.52; 3.69]
## ibgecode2202539    2.80 *
##                    [1.23; 4.38]
## ibgecode2202554    2.80 *
##                    [2.59; 3.02]
## ibgecode2202604    4.35 *
##                    [3.85; 4.86]
## ibgecode2202653    3.65 *
##                    [2.56; 4.75]
## ibgecode2202703    2.50 *
##                    [2.46; 2.55]
## ibgecode2202711    3.75 *
##                    [3.25; 4.26]
## ibgecode2202729    4.20 *
##                    [3.40; 5.00]
## ibgecode2203271    2.75 *
##                    [2.65; 2.86]
## ibgecode2203420    3.40 *
##                    [2.01; 4.80]
## ibgecode2203503    2.90 *
##                    [2.69; 3.12]
## ibgecode2203859    3.15 *
##                    [3.04; 3.27]
## ibgecode2204006    2.95 *
##                    [2.47; 3.44]
## ibgecode2204303    4.25 *
##                    [4.15; 4.36]
## ibgecode2204659    3.15 *
##                    [2.86; 3.45]
## ibgecode2204709    3.75 *
##                    [3.64; 3.87]
## ibgecode2204907    2.65 *
```

```
##                    [1.57; 3.73]
## ibgecode2205201      2.60 *
##                    [2.21; 2.99]
## ibgecode2205276      3.10 *
##                    [2.89; 3.32]
## ibgecode2205516      2.65 *
##                    [1.95; 3.36]
## ibgecode2205573      2.35 *
##                    [2.05; 2.66]
## ibgecode2205599      2.65 *
##                    [1.97; 3.34]
## ibgecode2205706      2.85 *
##                    [2.37; 3.34]
## ibgecode2205953      2.85 *
##                    [2.56; 3.15]
## ibgecode2206357      3.05 *
##                    [2.94; 3.17]
## ibgecode2206506      3.20 *
##                    [3.01; 3.40]
## ibgecode2206696      3.00 *
##                    [2.02; 3.99]
## ibgecode2206753      3.50 *
##                    [3.31; 3.70]
## ibgecode2206902      2.65 *
##                    [2.36; 2.95]
## ibgecode2207207      3.95 *
##                    [3.66; 4.25]
## ibgecode2207306      3.05 *
##                    [2.95; 3.16]
## ibgecode2207702      3.45 *
##                    [3.16; 3.75]
## ibgecode2207777      4.05 *
##                    [2.17; 5.94]
## ibgecode2207801      2.70 *
##                    [2.30; 3.11]
## ibgecode2207900      2.40 *
##                    [2.36; 2.45]
## ibgecode2207934      2.60 *
##                    [2.21; 2.99]
## ibgecode2207959      3.00 *
##                    [2.61; 3.39]
## ibgecode2208205      3.50 *
##                    [3.29; 3.72]
## ibgecode2208304      3.75 *
##                    [3.46; 4.05]
## ibgecode2208601      3.10 *
##                    [2.32; 3.89]
## ibgecode2208650      2.25 *
```

```
##                  [2.14; 2.37]
## ibgecode2209153    3.30 *
##                  [3.11; 3.50]
## ibgecode2209401    3.30 *
##                  [2.50; 4.10]
## ibgecode2209500    3.70 *
##                  [3.31; 4.09]
## ibgecode2209609    3.40 *
##                  [3.21; 3.60]
## ibgecode2209658    2.85 *
##                  [2.37; 3.34]
## ibgecode2209872    2.80 *
##                  [2.41; 3.19]
## ibgecode2209906    2.80 *
##                  [2.02; 3.59]
## ibgecode2210003    3.00 *
##                  [2.81; 3.20]
## ibgecode2210052    3.30 *
##                  [3.26; 3.35]
## ibgecode2210300    3.45 *
##                  [3.15; 3.76]
## ibgecode2210409    2.70 *
##                  [2.49; 2.92]
## ibgecode2210656    2.65 *
##                  [2.36; 2.95]
## ibgecode2210706    3.60 *
##                  [3.41; 3.80]
## ibgecode2210805    2.85 *
##                  [2.15; 3.56]
## ibgecode2210904    2.45 *
##                  [2.34; 2.57]
## ibgecode2211308    3.05 *
##                  [2.76; 3.35]
## ibgecode2211407    2.90 *
##                  [1.91; 3.90]
## ibgecode2211506    2.60 *
##                  [2.41; 2.80]
## ibgecode2211605    3.95 *
##                  [3.85; 4.06]
## ibgecode2300101    4.70 *
##                  [2.32; 7.09]
## ibgecode2300200    3.55 *
##                  [3.26; 3.85]
## ibgecode2300408    3.95 *
##                  [3.84; 4.07]
## ibgecode2300705    4.10 *
##                  [4.06; 4.15]
## ibgecode2301109    3.65 *
```

```
##                    [3.36; 3.95]
## ibgecode2301257    3.20 *
##                    [2.81; 3.59]
## ibgecode2301307    3.70 *
##                    [3.49; 3.92]
## ibgecode2301505    3.30 *
##                    [2.32; 4.29]
## ibgecode2301604    3.30 *
##                    [2.91; 3.69]
## ibgecode2301703    2.95 *
##                    [2.66; 3.25]
## ibgecode2301802    2.95 *
##                    [2.85; 3.06]
## ibgecode2301901    3.60 *
##                    [3.02; 4.19]
## ibgecode2302008    3.15 *
##                    [3.05; 3.26]
## ibgecode2302057    3.40 *
##                    [3.36; 3.45]
## ibgecode2302305    3.65 *
##                    [3.34; 3.96]
## ibgecode2302503    3.55 *
##                    [2.46; 4.65]
## ibgecode2302602    3.90 *
##                    [3.71; 4.10]
## ibgecode2302701    3.30 *
##                    [2.91; 3.69]
## ibgecode2303402    3.20 *
##                    [2.80; 3.61]
## ibgecode2303600    3.70 *
##                    [3.49; 3.92]
## ibgecode2303808    2.90 *
##                    [2.71; 3.10]
## ibgecode2303907    3.30 *
##                    [3.11; 3.50]
## ibgecode2304103    3.45 *
##                    [3.34; 3.57]
## ibgecode2304202    3.25 *
##                    [3.15; 3.36]
## ibgecode2304236    3.60 *
##                    [3.41; 3.80]
## ibgecode2304251    4.25 *
##                    [4.14; 4.37]
## ibgecode2304277    2.95 *
##                    [2.85; 3.06]
## ibgecode2304459    3.95 *
##                    [3.47; 4.44]
## ibgecode2304707    3.30 *
```

```
##                    [3.11; 3.50]
## ibgecode2305001    3.30 *
##                    [3.26; 3.35]
## ibgecode2305308    3.45 *
##                    [3.34; 3.57]
## ibgecode2305332    3.05 *
##                    [2.95; 3.16]
## ibgecode2305357    3.25 *
##                    [3.15; 3.36]
## ibgecode2305407    2.95 *
##                    [2.66; 3.25]
## ibgecode2305506    3.60 *
##                    [3.41; 3.80]
## ibgecode2305605    3.90 *
##                    [3.50; 4.31]
## ibgecode2305654    3.65 *
##                    [2.75; 4.56]
## ibgecode2305704    2.10 *
##                    [1.71; 2.49]
## ibgecode2305803    3.25 *
##                    [2.94; 3.56]
## ibgecode2305902    2.75 *
##                    [2.27; 3.24]
## ibgecode2306009    3.90 *
##                    [3.86; 3.95]
## ibgecode2306207    4.60 *
##                    [4.39; 4.82]
## ibgecode2306553    3.80 *
##                    [3.59; 4.02]
## ibgecode2307007    3.25 *
##                    [3.14; 3.37]
## ibgecode2307106    3.30 *
##                    [3.11; 3.50]
## ibgecode2307205    3.65 *
##                    [3.54; 3.77]
## ibgecode2307254    4.35 *
##                    [4.25; 4.46]
## ibgecode2307304    3.50 *
##                    [3.11; 3.89]
## ibgecode2307502    2.50 *
##                    [2.31; 2.70]
## ibgecode2307601    3.80 *
##                    [3.61; 4.00]
## ibgecode2307809    3.15 *
##                    [2.86; 3.45]
## ibgecode2307908    3.90 *
##                    [3.71; 4.10]
## ibgecode2308104    3.20 *
```

```
##                    [2.60; 3.81]
## ibgecode2308302     3.20 *
##                    [2.62; 3.79]
## ibgecode2308401     2.85 *
##                    [2.56; 3.15]
## ibgecode2308609     2.65 *
##                    [2.54; 2.77]
## ibgecode2308708     2.70 *
##                    [2.49; 2.92]
## ibgecode2308906     3.35 *
##                    [2.85; 3.86]
## ibgecode2309201     3.90 *
##                    [3.71; 4.10]
## ibgecode2309300     2.80 *
##                    [2.41; 3.19]
## ibgecode2309409     3.60 *
##                    [3.20; 4.01]
## ibgecode2309508     2.85 *
##                    [2.75; 2.96]
## ibgecode2310001     3.20 *
##                    [2.60; 3.81]
## ibgecode2310308     3.75 *
##                    [3.27; 4.24]
## ibgecode2310605     3.75 *
##                    [3.07; 4.44]
## ibgecode2310803     3.45 *
##                    [3.34; 3.57]
## ibgecode2310951     3.55 *
##                    [3.26; 3.85]
## ibgecode2311009     2.95 *
##                    [2.66; 3.25]
## ibgecode2311108     3.25 *
##                    [3.15; 3.36]
## ibgecode2311207     3.15 *
##                    [2.45; 3.86]
## ibgecode2311231     3.00 *
##                    [2.96; 3.05]
## ibgecode2311264     3.45 *
##                    [3.16; 3.75]
## ibgecode2311355     3.50 *
##                    [2.92; 4.09]
## ibgecode2311504     3.25 *
##                    [3.14; 3.37]
## ibgecode2311702     3.50 *
##                    [3.46; 3.55]
## ibgecode2311801     3.55 *
##                    [3.24; 3.86]
## ibgecode2311900     2.90 *
```

```
##                   [2.71; 3.10]
## ibgecode2311959    2.70 *
##                   [2.66; 2.75]
## ibgecode2312106    3.65 *
##                   [3.54; 3.77]
## ibgecode2312304    3.60 *
##                   [3.41; 3.80]
## ibgecode2312502    3.80 *
##                   [2.80; 4.81]
## ibgecode2313104    3.30 *
##                   [3.26; 3.35]
## ibgecode2313203    3.05 *
##                   [2.95; 3.16]
## ibgecode2313302    3.75 *
##                   [3.46; 4.05]
## ibgecode2313401    3.80 *
##                   [3.59; 4.02]
## ibgecode2313609    3.60 *
##                   [3.00; 4.21]
## ibgecode2313708    2.70 *
##                   [2.49; 2.92]
## ibgecode2313955    3.50 *
##                   [3.46; 3.55]
## ibgecode2314102    3.35 *
##                   [3.04; 3.66]
## ibgecode2400406    3.35 *
##                   [3.06; 3.65]
## ibgecode2400505    3.20 *
##                   [2.99; 3.42]
## ibgecode2401008    3.90 *
##                   [3.50; 4.31]
## ibgecode2401107    3.00 *
##                   [2.79; 3.22]
## ibgecode2401453    3.00 *
##                   [1.41; 4.59]
## ibgecode2402303    3.10 *
##                   [3.06; 3.15]
## ibgecode2402907    3.05 *
##                   [2.95; 3.16]
## ibgecode2403202    3.50 *
##                   [3.11; 3.89]
## ibgecode2403301    3.15 *
##                   [2.86; 3.45]
## ibgecode2403707    3.70 *
##                   [2.90; 4.50]
## ibgecode2404309    2.75 *
##                   [2.45; 3.06]
## ibgecode2404408    2.90 *
```

```
##                   [2.71; 3.10]
## ibgecode2404903     3.35 *
##                   [2.85; 3.86]
## ibgecode2406007     3.25 *
##                   [2.75; 3.76]
## ibgecode2407005     2.90 *
##                   [2.51; 3.29]
## ibgecode2407252     3.35 *
##                   [3.25; 3.46]
## ibgecode2407302     2.55 *
##                   [2.07; 3.04]
## ibgecode2408003     3.55 *
##                   [2.66; 4.45]
## ibgecode2409407     2.85 *
##                   [2.74; 2.97]
## ibgecode2410009     3.10 *
##                   [2.52; 3.69]
## ibgecode2410207     2.80 *
##                   [2.41; 3.19]
## ibgecode2410504     3.40 *
##                   [3.00; 3.81]
## ibgecode2410702     2.30 *
##                   [1.52; 3.09]
## ibgecode2410801     3.70 *
##                   [3.66; 3.75]
## ibgecode2411007     3.30 *
##                   [2.13; 4.48]
## ibgecode2411056     2.65 *
##                   [2.55; 2.76]
## ibgecode2411908     2.50 *
##                   [2.10; 2.91]
## ibgecode2412500     3.55 *
##                   [3.26; 3.85]
## ibgecode2413359     3.50 *
##                   [3.31; 3.70]
## ibgecode2413607     3.00 *
##                   [2.02; 3.99]
## ibgecode2413805     3.40 *
##                   [2.03; 4.78]
## ibgecode2414100     3.25 *
##                   [3.14; 3.37]
## ibgecode2414753     2.90 *
##                   [2.32; 3.49]
## ibgecode2414902     2.90 *
##                   [2.51; 3.29]
## ibgecode2500700     2.65 *
##                   [2.55; 2.76]
## ibgecode2502052     2.70 *
```

```
##                    [2.30; 3.11]
## ibgecode2502102     2.55 *
##                    [2.44; 2.67]
## ibgecode2502201     2.85 *
##                    [2.15; 3.56]
## ibgecode2502409     3.20 *
##                    [2.80; 3.61]
## ibgecode2503308     2.90 *
##                    [2.69; 3.12]
## ibgecode2503704     3.10 *
##                    [2.89; 3.32]
## ibgecode2504108     3.90 *
##                    [3.50; 4.31]
## ibgecode2504405     2.85 *
##                    [2.55; 3.16]
## ibgecode2505303     3.00 *
##                    [2.96; 3.05]
## ibgecode2505600     4.00 *
##                    [2.83; 5.18]
## ibgecode2507002     2.75 *
##                    [2.46; 3.05]
## ibgecode2509602     2.70 *
##                    [2.12; 3.29]
## ibgecode2511004     3.15 *
##                    [3.05; 3.26]
## ibgecode2512036     2.55 *
##                    [2.44; 2.67]
## ibgecode2512077     3.30 *
##                    [2.11; 4.50]
## ibgecode2513307     3.75 *
##                    [2.26; 5.25]
## ibgecode2513356     2.25 *
##                    [1.17; 3.33]
## ibgecode2513653     3.10 *
##                    [3.06; 3.15]
## ibgecode2514305     2.80 *
##                    [2.41; 3.19]
## ibgecode2514503     3.10 *
##                    [2.71; 3.49]
## ibgecode2516805     2.75 *
##                    [2.64; 2.87]
## ibgecode2516904     2.80 *
##                    [2.61; 3.00]
## ibgecode2601102     3.25 *
##                    [2.36; 4.15]
## ibgecode2602001     2.60 *
##                    [2.56; 2.65]
## ibgecode2604304     3.15 *
```

```
##                    [2.65; 3.66]
## ibgecode2605301     2.80 *
##                    [2.20; 3.41]
## ibgecode2606309     2.95 *
##                    [2.84; 3.07]
## ibgecode2607307     3.10 *
##                    [2.70; 3.51]
## ibgecode2609303     2.45 *
##                    [1.95; 2.96]
## ibgecode2609907     2.70 *
##                    [2.51; 2.90]
## ibgecode2610400     3.00 *
##                    [2.96; 3.05]
## ibgecode2612208     3.75 *
##                    [3.65; 3.86]
## ibgecode2612455     2.90 *
##                    [2.86; 2.95]
## ibgecode2612554     2.65 *
##                    [2.54; 2.77]
## ibgecode2613503     3.25 *
##                    [2.96; 3.55]
## ibgecode2614006     3.10 *
##                    [2.70; 3.51]
## ibgecode2614303     3.20 *
##                    [3.01; 3.40]
## ibgecode2615607     3.15 *
##                    [2.85; 3.46]
## ibgecode2616100     2.65 *
##                    [2.36; 2.95]
## post                0.39 *
##                    [0.30; 0.48]
## treat               0.20 *
##                    [0.07; 0.33]
## -----------------------------
## R^2                 0.99
## Adj. R^2            0.99
## Num. obs.         484
## RMSE                0.38
## =============================
## * 0 outside the confidence interval.
```

Of course, the table is unnecessarily large due to so many coefficients. You can solve this by hiding the unit fixed-effects from the regression table while the effect is still measured.

```
screenreg(m4_1,
          omit.coef = "ibgecode",
          custom.model.names = "m4.1")
```

```
##
## ========================
##              m4.1
## ------------------------
## post           0.39 *
##              [0.30; 0.48]
## treat          0.20 *
##              [0.07; 0.33]
## ------------------------
## R^2            0.99
## Adj. R^2       0.99
## Num. obs.    484
## RMSE           0.38
## ========================
## * 0 outside the confidence interval.
```

We can do the same model now but clustering by municipality.

5. A model with explicit municipality dummies for fixed-effects and no controls, with no intercept, and with clustering standard errors by municipality.

```r
m4_2 <- lm_robust(ideb ~ 0 + ibgecode + post + treat,
          clusters = mun,
          se_type = "stata",
          data = data_long)

screenreg(m4_2,
          omit.coef = "ibgecode",
          custom.model.names = "m4.2")
```

```
##
## =========================
##               m4.2
## -------------------------
## post            0.39 *
##               [0.26; 0.52]
## treat           0.20 *
##               [0.01; 0.39]
## -------------------------
## R^2             0.99
## Adj. R^2        0.99
## Num. obs.     484
## RMSE            0.38
## N Clusters    241
## =========================
## * 0 outside the confidence interval.
```

We can also use the *fixed_effects* argument of *lm_robust()* and specify a formula with *ibgecode*. This is the same as writing *+ ibgecode* in the function, but this tool can be used to more complex fixed-effect models.

```
# with clustering
m4_3 <- lm_robust(ideb ~ post + treat,
          fixed_effects = ~ ibgecode,
          data = data_long)

# without clustering
m4_4 <- lm_robust(ideb ~ post + treat,
          fixed_effects = ~ ibgecode,
          clusters = mun,
          se_type = "stata",
          data = data_long)

screenreg(list(m4_3, m4_4),
          omit.coef = "ibgecode",
          custom.model.names = c("m4.3", "m4.4"))
```

```
##
## ======================================
##             m4.3          m4.4
## --------------------------------------
## post          0.39 *        0.39 *
##             [0.30; 0.48]  [0.26; 0.52]
## treat         0.20 *        0.20 *
##             [0.07; 0.33]  [0.01; 0.39]
## --------------------------------------
## R^2           0.80          0.80
## Adj. R^2      0.60          0.60
## Num. obs.   484           484
## RMSE          0.38          0.38
## N Clusters                241
## ======================================
## * Null hypothesis value outside the confidence interval.
```

We can keep adding controls to these models if we feel that the DiD may still include confounders, such as gdp per capita.

6. A model with municipality fixed-effects and with control for the time-varying variable.

```
m5_1 <- lm_robust(ideb ~ post + treat + ln_pibpc,
          fixed_effects = ~ ibgecode,
          data = data_long)

m5_2 <- lm_robust(ideb ~ post + treat + ln_pibpc,
```

```r
          fixed_effects = ~ ibgecode,
          clusters = mun,
          se_type = "stata",
          data = data_long)

screenreg(list(m5_1, m5_2),
          omit.coef = "ibgecode",
          custom.model.names = c("m5.1", "m5.2"))
```

```
##
## =========================================
##              m5.1           m5.2
## -----------------------------------------
## post           0.41 *         0.41 *
##              [ 0.30; 0.51]  [ 0.26; 0.55]
## treat          0.21 *         0.21 *
##              [ 0.07; 0.34]  [ 0.02; 0.39]
## ln_pibpc      -0.24          -0.24
##              [-0.77; 0.28]  [-0.97; 0.49]
## -----------------------------------------
## R^2            0.80           0.80
## Adj. R^2       0.59           0.59
## Num. obs.    480            480
## RMSE           0.38           0.38
## N Clusters                  239
## =========================================
## * Null hypothesis value outside the confidence interval.
```

7 - Trying to control for variables that do not change over time when using fixed effects. Spoiler: it won't work!

```r
# This is just to show that it does not work!
m5_3 <- lm_robust(ideb ~ post + treat + ln_pibpc + distcap,
                  fixed_effects = ~ ibgecode,
                  data = data_long)

screenreg(m5_3,
          custom.model.names = "m5.3")
```

```
##
## ========================
##             m5.3
## ------------------------
## post          0.41 *
##             [ 0.30; 0.51]
## treat         0.21 *
```

```
##              [ 0.07; 0.34]
## ln_pibpc      -0.24
##              [-0.77; 0.28]
## distcap
##
## ------------------------
## R^2            0.80
## Adj. R^2       0.59
## Num. obs.    480
## RMSE           0.38
## ========================
## * 0 outside the confidence interval.
```

The error message "1 coefficient not defined because the design matrix is rank deficient" means that distcap was perfectly collinear with the fixed effects (ibgecode). When we include fixed effects, all time-invariant variation across units (in this case, municipalities) is absorbed. As a result, it becomes impossible to estimate the effect of any variable that does not change over time, because its variation is already captured by the fixed effects.

### Step 9: Interpret the Fixed-Effects Models

Now, you should interpret all of the coefficients and how fixed-effects change their interpretation.

Combining all regressions into a table can be helpful. Note how the different parameters of *screenreg* can be used:

```
screenreg(list(m4_1, m4_2, m4_3, m4_4, m5_1, m5_2),
          omit.coef = "ibgecode",
          custom.model.names = c("m4_1",
                                 "m4_2",
                                 "m4_3",
                                 "m4_4",
                                 "m5_1",
                                 "m5_2"),
          custom.header = list("FE in Formula" = 1:2,
                               "FE in Function" = 3:4,
                               "Control Time-Varying" = 5:6),
          custom.coef.names = c("Post",
                               "Treat",
                               "GDPpc (log)"),
          custom.gof.rows = list("Fixed-effects" = c(T, T, T, T, T, T),
                                 "Clustering" = c(F, T, F, T, F, T),
                                 "FE in formula" = c(T, T, F, F, F, F),
                                 "FE in parameter" = c(F, F, T, T, T, T)),
          groups = list("Main Variables" = 1:2,
                        "Control" = 3),
```

```
        column.spacing = 2,
        center = TRUE)
```

| | FE in Formula | | FE in Function | | Control Time-Varying | |
|---|---|---|---|---|---|---|
| | m4_1 | m4_2 | m4_3 | m4_4 | m5_1 | m5_2 |
| Main Variables | | | | | | |
| Post | 0.39* | 0.39* | 0.39* | 0.39* | 0.41* | 0.41* |
| | [0.30; 0.48] | [0.26; 0.52] | [0.30; 0.48] | [0.26; 0.52] | [0.30; 0.51] | [0.26; 0.55] |
| Treat | 0.20* | 0.20* | 0.20* | 0.20* | 0.21* | 0.21* |
| | [0.07; 0.33] | [0.01; 0.39] | [0.07; 0.33] | [0.01; 0.39] | [0.07; 0.34] | [0.02; 0.39] |
| Control | | | | | | |
| GDPpc (log) | | | | | −0.24 | −0.24 |
| | | | | | [−0.77; 0.28] | [−0.97; 0.49] |
| Fixed-effects | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |
| Clustering | FALSE | TRUE | FALSE | TRUE | FALSE | TRUE |
| Fixed-effects in formula | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE |
| Fixed-effects in parameter | FALSE | FALSE | TRUE | TRUE | TRUE | TRUE |
| $R^2$ | 0.99 | 0.99 | 0.80 | 0.80 | 0.80 | 0.80 |
| Adj. $R^2$ | 0.99 | 0.99 | 0.60 | 0.60 | 0.59 | 0.59 |
| Num. obs. | 484 | 484 | 484 | 484 | 480 | 480 |
| RMSE | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 |
| N Clusters | | 241 | | 241 | | 239 |

* Null hypothesis value outside the confidence interval.

Table 1: Statistical models

### Step 10: Plotting

Produce a graph of the data showing the changes in average outcome values in control and treatment group.

First, we need to get the average *ideb* for both *groups* in *pre* and *post* treatment periods. Your bread and butter is usually using *group_by* and *summarise* from the *dplyr* package within the *tidyverse*.

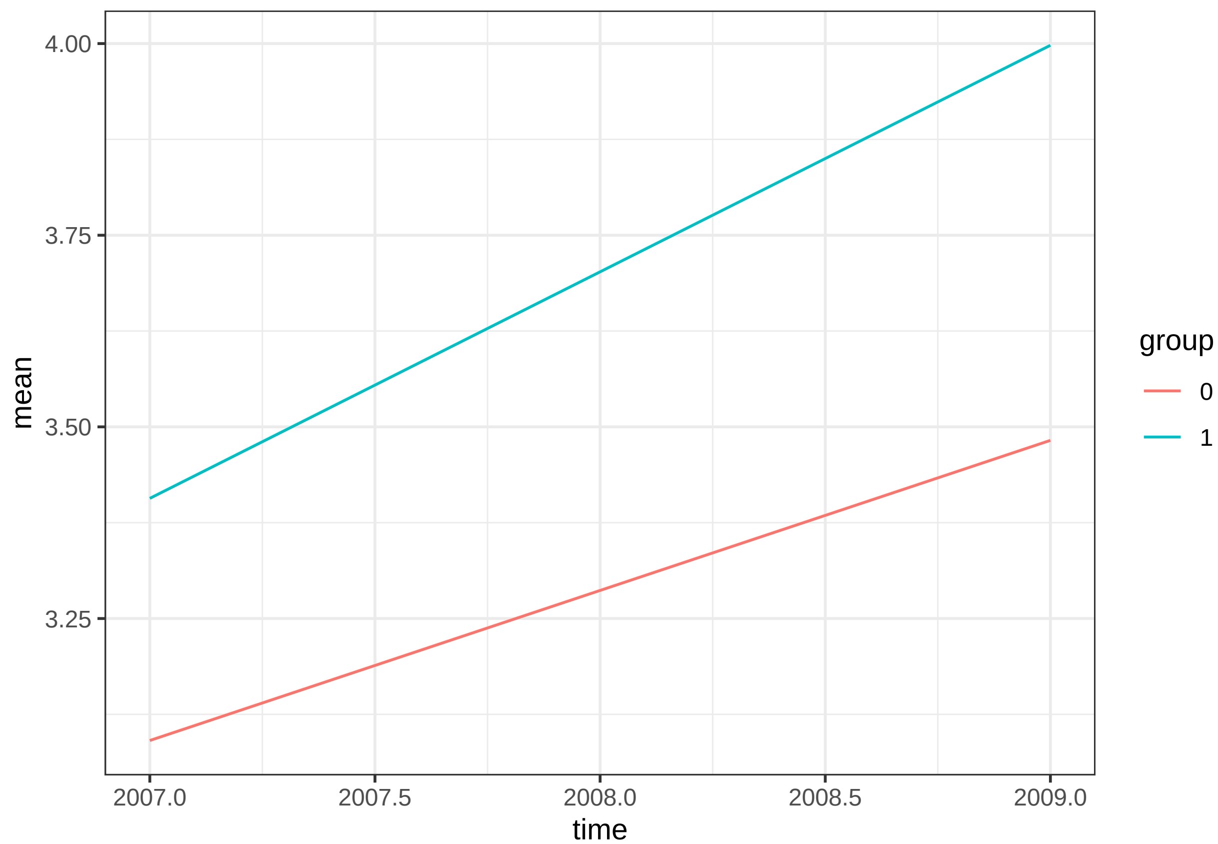
For plotting purposes, we can use the variable *time* (instead of post) to represent the before/after. This is because *time* is written as 2007 and 2009, which look better on the x-axis in comparison to the 0 and 1 of the *post* variable. However, you can easily do the same plot with *post* and manually change the x-axis with *ggplot*, also in the *tidyverse*.

```
# Group and summarize
data_group <- data_long %>%
    group_by(time, group) %>%
    summarise(mean = mean(ideb)) %>%
    mutate(group = as.factor(group))
```

Simple plot

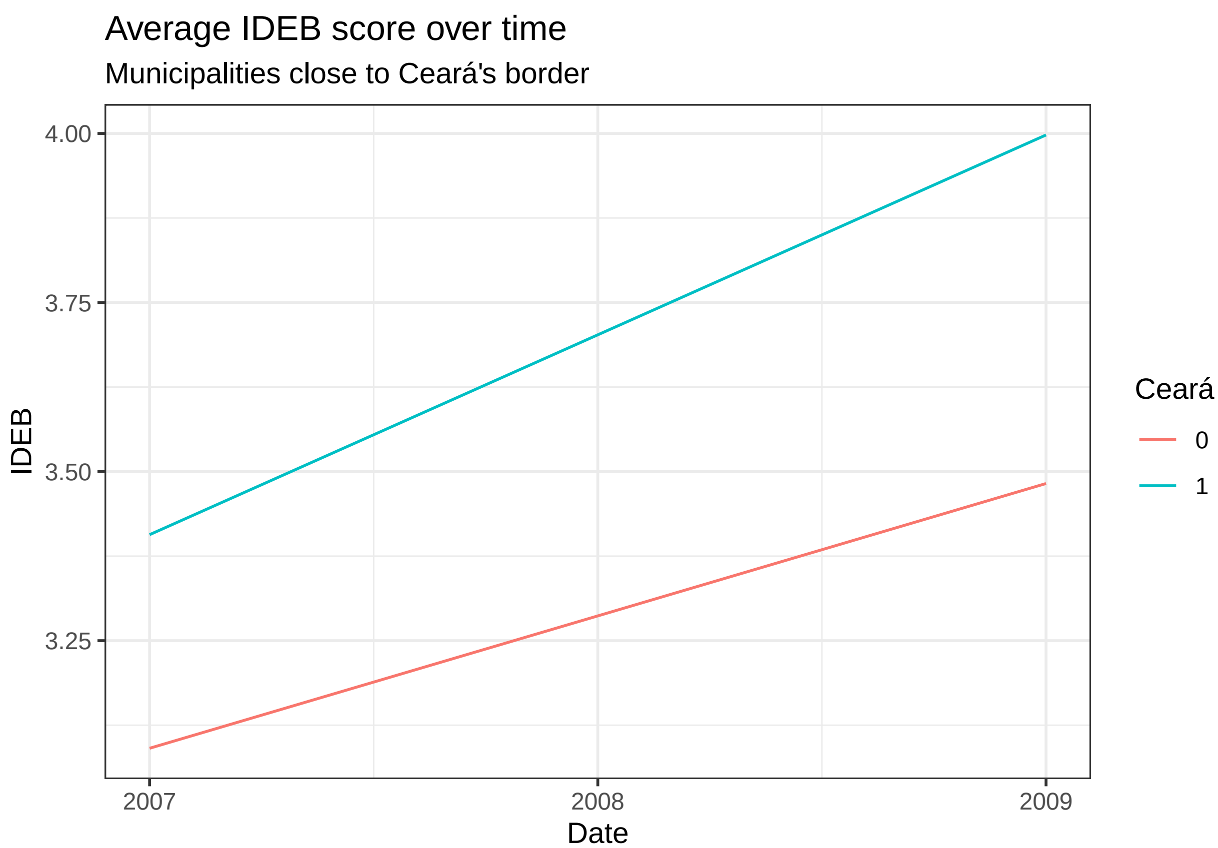
```
data_group %>%
    ggplot(aes(x = time, y = mean, color = group)) +
```

```r
    geom_line() +
    theme_bw()
```




Edit the plot as needed

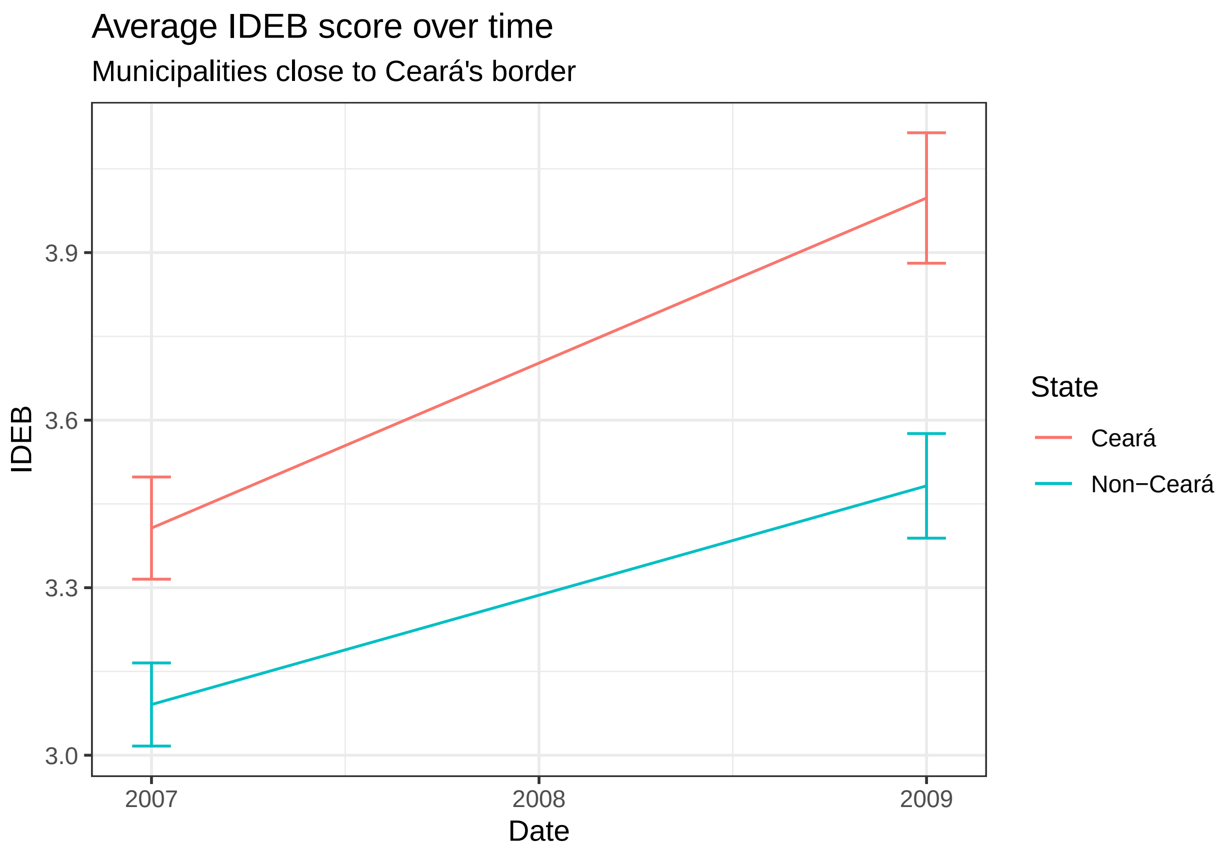
```r
data_group %>%
  ggplot(aes(x = time, y = mean, color = group)) +
  geom_line() +
  labs(title = "Average IDEB score over time",
       subtitle = "Municipalities close to Ceará's border",
       x = "Date",
       y = "IDEB",
       color = "Ceará") +
  scale_x_continuous(breaks = c(2007,2008,2009)) +
  theme_bw()
```




Advanced plot (w/ standard errors)

```
data_long %>%
  group_by(time, group) %>%
  summarise(n = n(),
            mean = mean(ideb),
            sd = sd(ideb)) %>%
  mutate(se = sd / sqrt(n))  %>%
  mutate(ci = se * qt((1 - 0.05)/2 + 0.5, n - 1)) %>%
  mutate(group = as.factor(group)) %>%
  ggplot(aes(x = time, y = mean, color = group)) +
  geom_line() +
  geom_errorbar(aes(ymin = mean - ci,
                    ymax = mean + ci),
                width = 0.1) +
  labs(title = "Average IDEB score over time",
       subtitle = "Municipalities close to Ceará's border",
       x = "Date",
       y = "IDEB") +
  scale_x_continuous(breaks = c(2007, 2008, 2009)) +
  scale_color_discrete(name = "State",
                       limits = c("1", "0"),
                       labels = c("Ceará", "Non-Ceará")) +
```

```r
theme_bw()
```



Average IDEB score over time
Municipalities close to Ceará's border
IDEB
3.9
3.6
3.3
3.0
2007
2008
2009
Date
State
Ceará
Non−Ceará

## Your Assignment

Your goal is to re-analyze the data, this time using a dataset with four time periods. You will produce a "lab write-up" with no more than **2 thousand words** long, which includes a maximum of **two tables** and **one plot** and that should be uploaded on E-Class until Friday (May 16th at 17:29). You can work with colleagues to perform the analysis, but you should produce your own document to be turned in. If you work with others, please indicate in your assignment who you worked with. You do not need to report all results in the order below.

1. Download the data file *lab01-2023-DATAfourperiod.csv* from the "Lab 1" folder on E-Class' "Content" section. Note: this is not the same file you used in the lab exercise, which was *lab01-2023-DATAtwoperiod.csv*.
2. Reshape the dataset so that you have a "long" format, with the dependent variable IDEB, the treatment indicator, and indicators for time. Make sure you create a time varying variable for income per capita, which you can call *gdppc*. For the treatment indicators, you can create a post-treatment indicator to be interacted with a treatment group indicator, but we recommend you use the "simpler" setup and create a treatment indicator $treat_{ij}$ that takes on the value of 1 for post-treatment observations in the treatment group, and 0 otherwise.
3. Estimate the "full" four-period DiD regression model for IDEB (i.e. include an indicator for treatment group $group_i$, all the four time terms $time_t$ and interactions between time and group indicators). In a single table, report estimates for the specifications below. Keep the table simple: You do not need to report the estimates for the control variables and for fixed-effects in the table, just indicate which specifications included them. Cluster standard errors by municipality in all but the basic grouped model.
   a A very basic "full" model (model 1a)
   b Add (to model 1b) unit fixed-effects to the model
   c Add (to model 1c) a control for a time changing variable
4. Interpret the results as completely as possible and discuss any evidence of pre-treatment parallel trends.
5. Estimate the "typical" DiD model, as seen in Bechtel & Hainmueller and Bertrand et al. In a second table, report estimates for the following variants. Report only the "causal effects" (with clustered SEs where applicable and p-values), but identify each model while keeping the table as simple as possible.
   a A very basic "typical" model (model 2a)
   b Add (to model 2b) only time fixed-effects
   c Add (to model 2c) only unit fixed-effects
   d Add (to model 2d) both time and unit fixed-effects
   e Add (to model 2e) the time changing variable
6. Interpret all coefficients, explain the substantive interpretation of the results, and contrast them with those of the "full" model.
7. Create a plot that in your judgement best summarizes the data.

All files used in the lab sessions are available on E-Class. **You are free to use whatever software you prefer.**