Ulm University

Faculty of Mathematics and Economics

CHINA'S POLITICAL PROMOTION INCENTIVES TO CONTROL RIVER BORDER POLLUTION – AN INTERACTIVE ANALYSIS WITH R

Master Thesis

in Management and Economics

of

Maier; Simon
Matriculation Number: 919636

submitted on 01.04.2022

Reviewer

Prof. Dr. Sebastian Kranz

Supervisor

Dr. Alexander Rieber

Table of Contents

Problem Set: China’s Political Promotion Incentives to Control River Border Pollution

Introduction

Author: Simon Maier

Welcome! I am glad you found your way to this interactive analysis in R. This problem set is part of my master thesis at Ulm University and is based on the paper **“Water Pollution Progress at Borders: The Role of Changes in China’s Political Promotion Incentives”** written by Matthew E. Kahn, Pei Li and Daxuan Zhao. The study examines the role of political promotion incentives for local officials in combating river pollution that spills across political boundaries. The study has been published in 2015 in the *American Economic Journal: Economic Policy*. Both the article and the data can be found here[1].

Note: Please open all links in a new tab by right clicking and selecting “Open link in new tab”. This will prevent you from losing any progress you may have made in the problem set.

The problem set is published here:

- GitHub: https://github.com/SimonMaier1995/Chinas-Political-Promotion-Incentives
- ShinyApps: https://simonmaier.shinyapps.io/Chinas-Political-Promotion-Incentives

The free-rider problem is a type of market failure. It occurs when individuals are able to benefit from resources without paying for them. As a consequence, purely individualistic mechanisms will prevent the generation of the optimal amount of the corresponding public good (Varian, 2014).

We encounter this problem, for example, with regard to water pollution of rivers crossing political borders. The social costs of contaminating activities are borne by downstream neighbors while possible pollution avoidance costs and social gains such as through economic growth remain within the political boundaries. Applied to provinces, this leads to a degree of water pollution that is above the national’s social optimum. Significant free riding issues regarding river border pollution have been documented in several studies (Cai, Chen and Qing, 2013; Sigman 2002, 2005; Sandler, 2006).

For instance, it was found that dirty firms in the Chinese province Hebei were more likely to set up in border than in interior counties (Duvivier, 2013). As a result, the central government acted by motivating local officials to invest more in water pollution prevention as well as stricter control and monitoring of river border pollution. Orientation to the central government’s guidelines and fulfillment of their plan targets has a direct impact on the promotion prospects of provincial officials.

Hence, we aim to evaluate in this problem set whether the government’s action has been successful in order to reduce excessive water pollution at the borders. We do this by thoroughly explaining the core methods applied by the researchers as well as representing the results step by step.

[1] https://www.aeaweb.org/articles?id=10.1257/pol.20130367

Exercise Content

Overview

Let us start by learning some basics about how China's political system is structured as well as introducing the free-rider problem regarding river border pollution the paper is based on. Afterwards, we will have a first glance at the data set and analyze measurements from several water pollution indicators descriptively. Then, the centerpiece of the paper follows: We introduce the econometric background regarding the *difference in differences (DiD)* approach and learn step by step how the regression analysis is put together by introducing cluster-robust standard errors, fixed effects and control variables. In addition to the authors' analysis, I implemented a *triple difference* analysis by regression that you will find in the appendix. In the fourth part, we examine the link between provincial officials' career concerns and their ambition in implementing the new environmental policy. In the exercise that follows, we have a closer look at the location of pulp and paper plants as one of the main contaminators regarding *chemical oxygen demand (COD)*, the indicator in focus of the new pollution targets. In particular, we are interested where new factories have been opened with regard to the proximity of the border before and after the regime change. In the last section, we conclude the problem set, but note that in several alternative regression specifications are appended to check the robustness of the results.

How to Work on the Problem Set

When solving the exercises, you do not have to follow the structure of the problem set strictly. Nevertheless, I recommend this approach since it is organized like a tour through the entire study. However, within the exercises you still have to follow the order of the tasks. The essence of the problem set are the code chunks, where you will encounter three types:

- empty code chunk where you have to find the solution completely by yourself;
- code chunks with gaps where you should complete the existing code by replacing ___;
- ready to run code chunks where you just check the chunk, the entire code is already given;

In case you want to work on a code chunk press edit first and in case you need some advice, press the hint button. However, it is possible to show the correct code by the solution button. If you want to execute the code without checking it, press simply run. Finally, to verify and complete the task, click on check.

In addition to code blocks, there are also quizzes you can work on. In some cases, you can simply guess the correct answer, while other questions will help you to gain a deeper understanding and test your knowledge of the topics covered.

As soon as you have finished a (sub-)exercise, click on Go to next exercise... to continue.

Exercise 1 – Motivation

The first chapter is intended to give a first impression of the topic covered in this problem set and why it is important.

Therefore, we learn about the characteristics and peculiarities of the Chinese political system that are relevant to this study in the first sub-exercise. The focus is on the hierarchical relationships between the levels of government, particularly between the central and provincial governments. We also briefly summarize the relationships between the state organs and the ruling *Communist Party of China (CPC)*.

In the second part, we explain the free-rider problem in river border pollution by visualizing the problem using a map. We also show, using a descriptive analysis and a regression model, that the pollution levels for several indicators at the borders are above the inland levels over the entire observation period.

Note: Since I only want to motivate in this chapter, all the *R* code is already included in the chunks. So, you do not need to worry if you do not yet understand everything that happens in the code chunks. Just click check to execute and complete the tasks. However, you should still answer the quiz questions.

Structure

1.1 Relationship Between China’s Central and Provincial Governments

1.2 Above-average Pollution at Provincial Borders

Exercise 1.1 – Relationship Between China's Central and Provincial Governments

The underlying paper examines the impact of a policy measure aimed at reducing higher-than-average water pollution at provincial borders. To better understand and motivate the topic, we start with some information about the structure of China's political system. If you like, you can take a look at the China Internet Information Center's website. It provides basic knowledge and is authorized by the Chinese administration. Similar official information can be found on the website of China's state media CGTN. A very comprehensive work is *China's Political System*, written by Sebastian Heilmann.

Let us start by examining the role of the ruling *CPC*, the relationship between its organs and the state organs. The CPC was established in 1921 and is, with a membership of more than 95 million, the second largest political party in the world. The members are building branches and are considered as the **Grass-Root Level** of the party.

Those branches select delegates which make up congresses at the municipal, county, city and provincial levels, whereby the delegates of the lowest two levels are elected by the people. The congresses, in turn, elect the government of the respective administrative level, which, however, is bound by the instructions of the committees appointed by the CPC. They manifest the **Local Level**.

On the **Central Level** we find the party's *National Congress* that is voted by forty electoral units. It consists out of over 2000 members which elect the *Central Committee* via a candidate list that is prepared by the *Politburo* and its *Standing Committee*. During a plenary meeting, that occurs at least once per year, the **Central Committee** holds elections for composing the *Politburo* and its *Standing Committee*. The latter exercises functions and powers of the *Central Committee* between the just mentioned plenary sessions. Furthermore, it nominates the members of the committee's working body, the *Secretariat*.

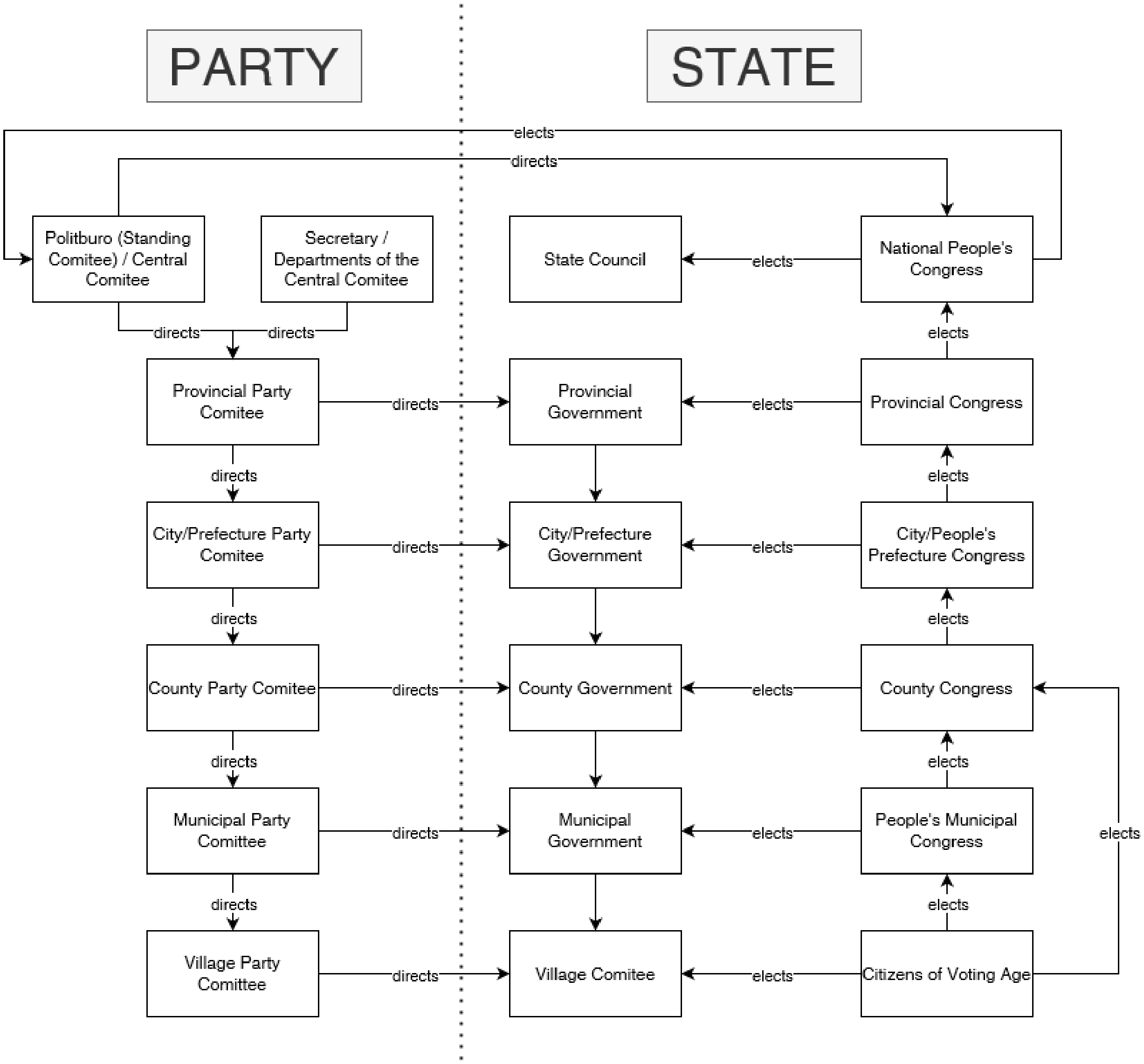
Summarizing, there are three top bodies of the party, which centralize a great deal of power:

- *Central Committee*
- *Politburo*
- *Politburo Standing Committee*

Let us now outline how the institutions of the party are linked to the state organs:

By constitution, the *State Council* is equal to the *Government of China*. It is dominated by members of the CPC's *Central Committee* and is the top administrative authority of China. The *State Council* is administered to elaborate and implement plans for development of the country. The *Five-Year Plans* are one of the main characteristics of the *Socialism with Chinese Characteristics* and contain detailed guidelines for the economic and social development of all regions. For instance, the river border pollution targets which became effective in 2006 are part of the *Eleventh Five-Year-Plan*.

Structure of China's Political System



Adapted from: "Das politische System der Volksrepublik China by S. Heilmann

Quiz: Do you think China's political system is rather centralized or federal?

- Centralized. [x]
- Federal. [ ]

One of the major organizational principles regarding state organs of the Chinese constitution is called *Democratic Centralism*. Aside from the constitution, the statute of the *Communist Party of China (CPC)* demonstrates that the country is a highly centralized unitary state. Supervision is exercised by upper administrative organs and leaders over lower ones. This is done by canceling orders and decisions as well as evaluating lower organ's work and results and awarding or penalizing them accordingly by superior units. Depending on the decade, the reins were tightened sometimes more and sometimes less (Heilmann, 2004).

Nevertheless, it is significant that China has undergone economic policy decentralization in recent decades. Montinola, Qian and Weingast (1996) describe this model as *market-preserving federalism*. Although the constitution does not grant federalism, the central government limits itself to monetary and regulatory control. By refraining from interfering with decentralized economic regulation, it creates a highly competitive environment among the provinces. Local rulers have a strong interest in promoting the competitiveness, productivity and profitability of local businesses rather than burdening them with regulation. This is also closely related to the free-rider problem discussed in the paper.

After learning something about the highest administrative organ, let us continue with the levels below. Just have a guess when answering the following questions.

Quiz: Which is the correct order of the administration levels (from highest to lowest)?

- Province, prefecture, county. [x]
- Prefecture, county, province. [ ]
- Prefecture, province, county. [ ]
- County, province, prefecture. [ ]

The highest administrative level, which is also the focus of this work, is the provincial level. Therefore, entities on this level are directly accountable to the central government for achieving the plan goals.

Quiz: What is the function of a governor in China?

- Head of the central government. [ ]
- Head of the government of a province. [x]
- Head of the government of a prefecture. [ ]
- Head of the government of a county. [ ]

The governor is the head of a province, while still not being the highest ranking executive due to his or her subordination to the secretary of the provincial communist party. However, considering previous environmental incidents, the governors seem to be more responsible for the local government's performance. As a consequence of the aniline spill in Changzhi (Shanxi province, December 2012) and the cadmium contamination in Longjiang (Guangxi province, February 2012) only the governors but not the party secretaries have been punished by the central government (Kahn, 2015).

While the central government set targets in five-year-plans, provincial governments are encouraged and incentivized to meet them. Hence, the governors are evaluated by those goals, and the more successful they are it the higher are their promotion chances. During 1978- 2005, one third of the provincial officials were promoted as leaders of central authorities, with a rising trend (Xianbin, 2008).

Quiz: What is the number of entities at the provincial level in China? Just guess.

- 34 [x]
- 4 [ ]
- 211 [ ]
- 1051 [ ]

The following types of entities at the provincial level exist:

- *Province* (23, for example Sichuan).
- *Autonomous Region* (5, for example Xinjiang).
- *Municipality* (4, for example Beijing).
- *Special Administrative Region* (Hong Kong and Macao).

The population of the divisions at the provincial level range from 552,300 for Macau, which is comparable to Luxembourg, up to 126,012,510 for Guangdong, which is roughly the population size of Japan. In terms of both number of inhabitants and area, the median province can be rather thought of as a medium-sized country.

Summary

The governors, together with the local party secretaries, run the provincial governments. The 34 entities at the provincial level, where a single one can be as large as Japan, are the highest administrative units in China. Hence, governors are directly subordinated and accountable to the central government that sets targets. Achieving these goals is therefore critical for provincial officials to receive promotions, which boosts economic competition. Without being responsible for water pollution reaching neighboring provinces, governors could have directed to locate heavily contaminating industries at the downstream boundaries, thus causing above-average river border pollution.

Let us compare water pollution at the border and in the interior of the provinces in the next sub-exercise.

Exercise 1.2 – Above-average Pollution at Provincial Borders

Significant free-rider issues regarding river pollution at borders have been documented worldwide in several studies, as we learnt in the introduction already (Cai, Chen and Qing, 2013; Sigman 2002, 2005; Sandler, 2006).

Local officials of provinces are evaluated by a set of targets. Especially, governors are in charge of achieving those goals. While economic, environmental and social goals are set by the central government, provincial leaders have some freedom to choose the means to fulfill the plans as we learnt in the first part of the exercise. In order to maximize chances of their own promotion, officials of the provinces could have aimed at positioning dirty industries at the border to make the neighbor province bearing the costs of pollution while fostering the economic growth of the province they are responsible for.

Quiz: Let us assume governors are evaluated by their province's interior environmental and economic situation only. What is the most rational way for them dealing with polluting industries, taking into account this simplification?

- Do not invest in pollution prevention measures and locate them in the interior of the province. [ ]
- Invest in pollution prevention measures and locate them in the interior of the province. [ ]
- Do not invest in pollution prevention measures and locate them at the provincial border. [x]
- Invest in pollution prevention measures and locate them in the interior of the province. [ ]

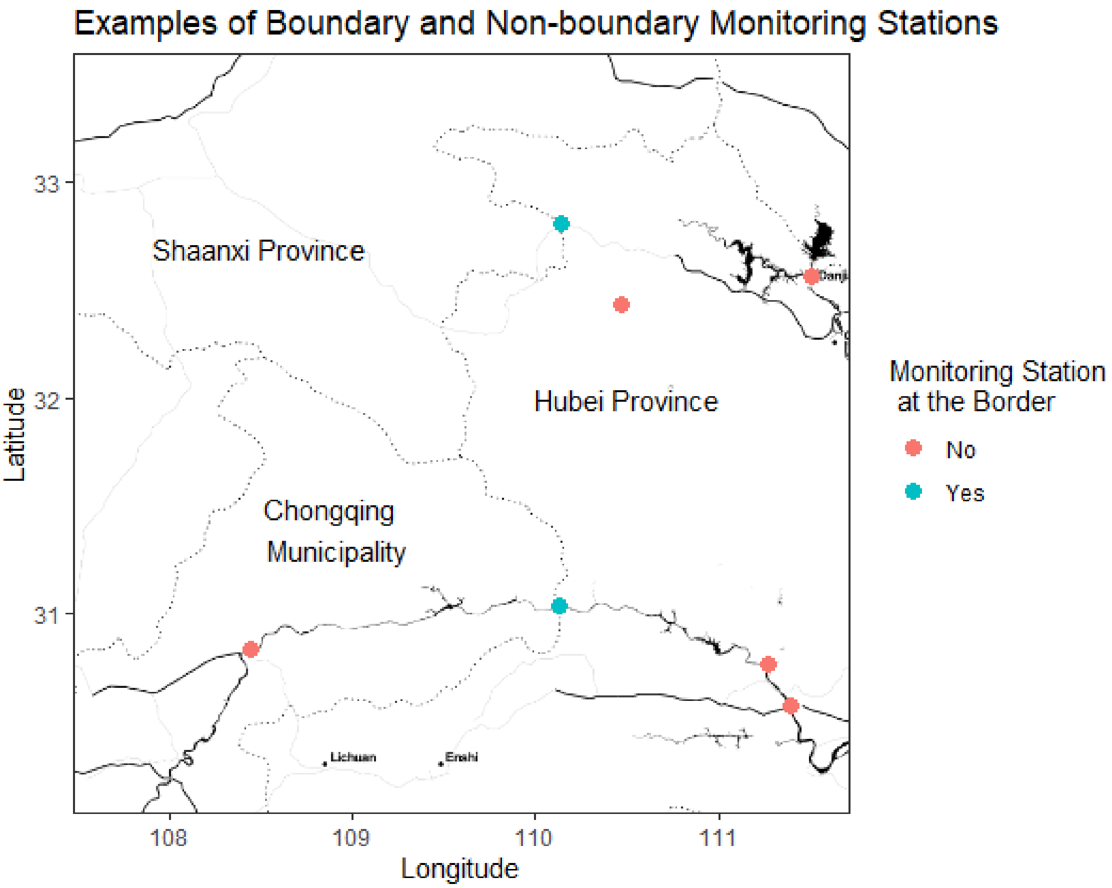
Investing in pollution prevention measures would possibly burden the economic performance of a province. On the other hand, deteriorating water quality in the interior of the province would decrease the rating of the governor. Solving this trade-off leads to strong incentives locating polluting industries at provincial boundaries, more precisely downstream. As a result, excessive water pollution at borders can be observed.

This is why the *Eleventh Five-Year-Plan*, which came in effect in 2006, makes upstream provinces responsible for the pollution that is experienced in downstream provinces. As a consequence, the new environmental policy should not have only motivated the officials of the provinces to reduce water pollution in general, but also discourage to free-ride at the cost of downstream provinces. Hence, water pollution at borders should decrease, too, and due to catch-up effects even stronger than in the inner part of the provinces.

To implement this policy, not only monitoring stations in the interior but also at the provincial boundaries were needed.

Task: To show some examples of border and non-border monitoring stations, just check the following chunk:

```
#read in prepared data set
dat = readRDS("dat.RDS")
#load map
map = readRDS("map.boundary.RDS")
#show map
ggmap(map) +
  geom_point(data=dat, aes(x=x, y=y, color = as.factor(boundary)), size=3) +
  annotate("text", x = 108.9, y = 31.4, label = 'Chongqing \n Municipality', size = 4) +
  annotate("text", x = 110.5, y = 32, label = 'Hubei Province', size = 4) +
  annotate("text", x = 108.5, y = 32.7, label = 'Shaanxi Province', size = 4) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Examples of Boundary and Non-boundary Monitoring Stations") +
  scale_color_discrete(labels = c("No", "Yes"), name = "Monitoring Station \n at the Border") +
  theme_bw()
```



The map section depicts a central part of China and shows the border region between the provinces Shaanxi and Hubei as well as the municipality of Chongqing. The dotted lines represent the provincial boundaries, while the solid lines stand for rivers. Monitoring stations at the borders are displayed as blue dots, while those in the interior of a province are colored in red.

Task: Now, let us find out how the pollution levels differ between boundary and non-boundary stations in 2005, the last year before the new environmental policy took effect. To do this, just check the following code chunk.

```
dat %>%
  filter(year == "2005") %>%
  group_by(boundary) %>%
  summarise_at(dplyr::vars(cod:mercury), mean) %>%
  mutate(across(c(cod:mercury), ~ round(.x, digits = 2))) %>%
  mutate(boundary = recode_factor(boundary, "0" = "No", "1" = "Yes")) %>%
  kbl(align = "c", caption = "Average River Pollution at the Border and the Interior of the Provinces in 2005",
col.names = c("Boundary", "COD (mg/l)", "BOD (mg/l)", "NH (mg/l)", "Petroleum (µg/l)", "Phenol (µg/l)",
"Mercury (µg/l)")) %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Average River Pollution at the Border and the Interior of the Provinces in 2005

Boundary	COD (mg/l)	BOD (mg/l)	NH (mg/l)	Petroleum (µg/l)	Phenol (µg/l)	Mercury (µg/l)
No	7.05	6.14	2.09	12.37	0.65	3.40
Yes	10.59	8.30	3.44	19.37	0.85	3.83

COD stands for chemical oxygen demand, *BOD* for biological oxygen demand and *NH* for ammonia nitrogen. In case you want to learn more about the different pollutants, have a look at the appended exercise A.3 COD Dynamics versus Other Indicators of Water Pollution.

Quiz: Is the average pollution measured at borders higher than it is at interior stations?

- Yes [x]
- No [ ]

The water pollution levels measured at borders are indeed jointly higher for all indicators the study observes.

Before we conclude this exercise, let us compare the COD pollution not only with help of descriptive statistics, but also by using a regression. The advantage is that we get to know not only the mere difference but also its significance. You do not have to know yet how a regression is performed in *R* since I will introduce it in the later exercises. So, the following task only serves as a motivation. Just as in the previous tasks, we observe measurements from 2005 only, the last year before the policy change came into force:

Task: Just check the chunk to perform a regression with COD as response and boundary as explanatory variable.

```
#run regression
reg = lm(cod ~ boundary, data = dat, subset = (year == 2005))
#show regression results
modelsummary(list("COD Pollution" = reg), group = model ~ term, coef_rename = c("(Intercept)"="Interior Pollution", "boundary" = "Additional Pollution \n at Borders"),
             stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
             gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE|F|RMSE",
             title = "COD Pollution in the Interior and at the Boundary") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

COD Pollution in the Interior and at the Boundary

	Interior Pollution	**Additional Pollution at Borders**
COD Pollution	7.046***	3.545***
	(0.682)	(1.344)
*** p < 0.1, ** p < 0.05, *** p < 0.01**		

We see that the coefficient of the boundary dummy that I labelled as Additional Pollution at Borders is 3.5452 and thus relatively large. If you scroll up and have a look at the table displaying the average river pollution in 2005, you will note that it matches the difference between COD pollution at boundary and non-boundary stations. It stands for the additional COD pollution that monitoring stations at the border captured, while the intercept that I labelled as Interior Pollution, 7.0460 represents the average COD measurement in the interior. Hence, the sum 10.5912 equals the average COD measurement at boundary monitoring stations. Furthermore, it is noteworthy, that the coefficient measuring the difference is highly significant at the one percent level.

Note: The values in parentheses are the standard errors of the estimated coefficients. This applies to all regression results from now on, unless otherwise noted.

Summary

In the last sub-exercise, we have illustrated the free-rider problem on a map. Moreover, we have compared the situation regarding water pollution at border and non-border stations in 2005, the year before the policy change became effective. Both, the descriptive and regression analysis have clearly shown that the contamination at borders was significantly more intense.

Let us dive deeper into the data set in the next exercise.

Award: Motivation

You have successfully completed the first exercise and are now motivated to learn more!

Exercise 2 – Data Overview

In the first exercise, we mainly acquired background knowledge helpful for a better understanding of the paper’s issue and why it is relevant. We dealt with the basics of the Chinese political administration and learnt about the environmental goals of the *Eleventh Five-Year Plan* for the development period from 2006 to 2010, which are important for our analysis.

Our data set is based on the years between 2004 and 2010 and focuses on the ten river systems listed in the *China Environmental Yearbooks*. Now, we are mainly interested how the data set is structured. It consists mainly of three parts:

First, six different metrics that are good indicators of water quality, including COD, that has been in focus of the regime change.
Second, it provides different control variables related to economic strength and the climate surrounding the according monitoring stations as possible confounding factors.
And last but not least, the data set contains information about the governors and secretaries of the provinces where the monitoring stations are located, including name and year of birth.

In this section, we will learn about how the data set is organized. The first sub-exercise serves as an introduction, while the second one focuses on river systems and monitoring stations. In the last sub-exercise, we analyze the six water pollution indicators descriptively, especially how the pollution levels change over time.

Structure

2.1 Introduction to the Data Set

2.2 River Systems and Monitoring Stations

2.3 Descriptive Statistics on Water Quality

Exercise 2.1 – Introduction to the Data Set

In order to later find our way around this data set with 26 variables and a total of over 3000 entries, we will first learn how it is structured.

Before we start, let us load the original data set the authors analysis is based on. The corresponding file is called *Regression_Data.dta* and is stored in the *.dta* file format. It is the primary format used by *Stata 15* through *Stata 17*. Since the *R* base package does not offer a way to read it, we use the read_dta() function provided by the haven package. If you need some help on how it works, feel free to read the info box below.

Info: How to load and save data with read_dta().

To read and save files of the .dat format we need the read_dat() function of the haven package. Therefore, we basically proceed in three steps.

First, we **load the required haven package**. This is done with the function library. We have to pass the name of the package as parameter.

```
library(haven)
```

Second, we **call the function read_dat**. We pass it as parameter the name and path of the file to be read in quotes. Let us assume that the name of the file is *file_name.dta* and that it is located in the current working directory.

```
read_dta("file_name.dta")
```

Third, we **assign a variable** to our read-in command to be able to access the data set at any time in the corresponding exercise. Let us say we would like to name the variable datset.name.

```
dataset.name = read_dta("file_name.dta")
```

Finally, we successfully read the file *file_name.dta* and assigned the result the variable dat.

Note that you find documentations to virtually all packages and their associated functions. For instance, if you want to learn more about the function we have covered here, have a look at read_dta() documentation.

Task: Start by loading the haven package using library().

```
#use library() to load the haven package
library("haven")
```

After successfully loading the library haven, let us use the function read_dta() to read the data set.

Task: Use read_dta() to read the file *Regression_Data.dta* and save it in a variable called dat.

```
#use read_dta() to load "Regression_Data.dta" and save it in dat
dat = read_dta("Regression_Data.dta")
```

After successfully loading the data set into the variable dat, we would like to explore how it is organized. *R* provides some basic functions that make it easier for us to have a first glance on data frames. One of them is head(). It simply displays the first six rows of the passed data set.

Task: Apply head() to the data set you have stored in the previous task into dat.

```
#use head() to show the first six rows of dat
head(dat)

##   sn station boundary year prov cod bod   nh petroleum phenol mercury  gdpg
## 1  5    H001        0 2008   63 1.8 0.5 0.19      0.01  0.001    0.01 0.131
## 2  4    H001        0 2007   63 2.1 1.5 0.24      0.01  0.001    0.01 0.140
## 3  1    H001        0 2004   63 2.4 1.0 0.13      0.01  0.001    0.05 0.117
## 4  2    H001        0 2005   63 2.7 1.0 0.14      0.01  0.001    0.02 0.113
## 5  6    H001        0 2009   63 1.7 1.0 0.15      0.01  0.001    0.01 0.136
## 6  3    H001        0 2006   63 1.3 1.0 0.21      0.01  0.001    0.01 0.141
##   temperature lightbuffer10km lightbuffer5km tpost riversystem_time riversystem
## 1         1.8               0              0     3       Yellow2008      Yellow
## 2         2.0               0              0     2       Yellow2007      Yellow
## 3         1.5               0              0     0       Yellow2004      Yellow
## 4         1.9               0              0     0       Yellow2005      Yellow
## 5         2.5               0              0     4       Yellow2009      Yellow
## 6         2.4               0              0     1       Yellow2006      Yellow
##   boundary_distance  gdpp        g_name g_yob     s_name s_yob       x       y
## 1           138.849 0.755   Song Xiuyan  1955 Qiang Wei  1953 100.155 35.4996
## 2           138.849 0.687   Song Xiuyan  1955 Qiang Wei  1953 100.155 35.4996
## 3           138.849 0.916 Yang Chuantang 1954  Zhao Leji  1957 100.155 35.4996
## 4           138.849 0.527   Song Xiuyan  1955  Zhao Leji  1957 100.155 35.4996
## 5           138.849 0.832   Song Xiuyan  1955 Qiang Wei  1953 100.155 35.4996
## 6           138.849 0.593   Song Xiuyan  1955  Zhao Leji  1957 100.155 35.4996
```

What do we see here? These are the first six rows of our data set and all 26 columns it consists of. One row represents the measurement values of six pollution metrics in one specific year for one specific measurement station. Additionally, it shows the corresponding information about the governor and secretary of the province the measurement station is in, economic and environmental data.

River Systems and Monitoring Stations

The column riversystem_time is the combination of the observation's year stored in year and the river system the station belongs to, saved in riversystem. The variable sn is a unique identifier for each observation while every monitoring station can be identified by station. If the station is at the border, the dummy variable boundary is equal to 1 The variable boundary_distance on the other hand stores the exact distance between the monitoring station and the border. Logically, for a boundary station it will take the value 0. The latitude and longitude of the station's geographical position is represented by x and y.

Water Pollution Indicators

The six pollution indicators are cod, bod, nh, petroleum, phenol, mercury which stand for the levels of chemical oxygen demand (COD), biological oxygen demand (BOD), ammonia nitrogen (NH), petroleum, mercury and phenol, in this order. While the first three water quality indicators are measured in milligrams per liter, the latter are given in micro grams per liter. They reflect the man-made pollution well and have the potential to travel far downstream. Therefore, we will measure the effects of upstream pollution on many downstream monitoring stations. The lower these values are, the better the water quality at the according measurement station is. In case you want to learn more about the mentioned substances, you might check out the info box in the appendix exercise A3. COD Dynamics versus Other Indicators of Water Pollution.

Economic and Environmental Data

Economic data is stored in gdpg and gdpp and is brought us by the *China Statistical Yearbook for Regional Economy*. The gross domestic product (GDP) per capita of cities in the river basin is stored in gdpp, while the growth rate of the GDP of the cities in the river basin is captured in gdpg.

Environmental variables are temperature that stands simply for the temperature, provided by the closest meteorological station through the *China Meteorological Data Sharing Service System* as well as lightbuffer10km and lightbuffer5km, which represent the luminosity level. While COD and BOD are linked to the temperature, the luminosity level tells us something about the economic and urban development of the region around the station and comes from the *Defense Meteorological Satellite Program (DMSP)*. For instance, we assume that an increased level of luminosity in the evening corresponds with a higher population density that is connected with the pollution intensity of rivers nearby.

Provincial Officials

In addition, the researchers used several public sources like *China Vitae* to gather biographical information about the party secretaries and governors of the provinces the monitoring stations are in. The names and years of births of the governors are contained in g_name and g_yob, while s_name and s_yob stand for the secretaries' information, correspondingly.

Summary

In this sub-exercise we had a look at the data set for the first time and already learnt some basic commands in *R*. Furthermore, we understood the motivation behind including the individual variables and the meaning behind them.

Let us now analyze some selected variables. In the next sub-exercise, we are going to learn more about the river systems and monitoring stations with help of the data set.

Exercise 2.2 – River Systems and Monitoring Stations

In this section, we are going to analyze data that is related to river systems and monitoring stations. To do so, let us load the data set again first:

Task: Just click on check to read the data set again and save it in dat.

```
dat = read_dta("Regression_Data.dta")
```

Each monitoring station is assigned to a specific river system. First of all, we want to find out which river systems are considered in the study. Just displaying the column riversystem would list all rows it consists of and therefore many duplicates. In order to show every single river system only once, we employ the function unique().

Info: How to get a distinct list of the values in a vector/column of a data frame.

This function is used for example when we want to learn about the different manifestations of a categorical variable. To do so, unique() removes all duplicate values from a vector or data frame. It is part of the R Base package so there is no need to load a library. The only parameter you are required to pass for needs is the vector, data frame or array you want to remove duplicates from.

Assume we want to know all distinct values from the column colors in dat.

```
unique(dat$colors)
```

If you are interested in how to specify the other, optional parameters, have a look here: Documentation unique()

Task: Use unique() to display the examined river systems in the study. Note that they are saved in the column riversystem in dat.

```
#apply unique() to the column riversystem in the data set dat
unique(dat$riversystem)

##  [1] "Yellow" "Hai"    "Huai"   "Liao"   "ZM"     "XG"     "YX"     "Song"
##  [9] "YTZ"    "Pearl"
```

While for some of the river systems the names are directly given, for four other river systems only the abbreviations are available. To find out what *ZM*, *XG*, *YX* and *YTZ* stand for, we will create a map with representations of the monitoring stations, coloring the points according to their belonging river systems in combination with a brief internet research.

Now, let us draw a map now to have an overview over the distribution of the monitoring stations. To be able to identify the affiliation of the stations to the river systems, we want to assign each a different color. The package we use here is ggmap.

As you can see in the code below, we use the latitude coordinates stored in x as well as the longitude coordinates in y from our data set. To download the map, we use get_map() from the ggmap package. If you are interested in how it works in detail, please have a look at the info box. In short, we need to supply the function with the coordinates of the map to download it from *Stamen Maps*.

Info: How to visualize spatial data with help of the ggmap package

Sometimes we wish to visualize spatial data in form of coordinates regarding longitude and latitude. In this case, the functions get_map() and ggmap()from the ggmap might be very helpful. Basically, you download a map first that fits the coordinates you would like to plot. Afterwards you are able to combine it with a ggplot2 function like geom_point() to plot the spatial data on the map you just got.

Now, let us say we would like to download a map of Egypt.

First of all, we **load the necessary packages** using library(). These are ggmap and ggplot2.

```
#load libraries
library(ggmap, ggplot2)
```

Second, we **download the map** using get_map(). In this case we pass the location by defining a left/bottom/right/top bounding box vector with the following boundaries. To get the boundary coordinates, have a look at BBoxfinder and draw a rectangle around Egypt. At the bottom of the website, you find the desired boundaries for the box.

```
#define boundaries
bounding.box = c(24.609375,21.759500,37.529297,31.728167)
#use get_map() to download the map and save it to a variable called `map.egypt`
map.egypt = get_map(location = bounding.box)
```

Usually, we further define some scaling parameters like zoom and scale within the get_map() function.

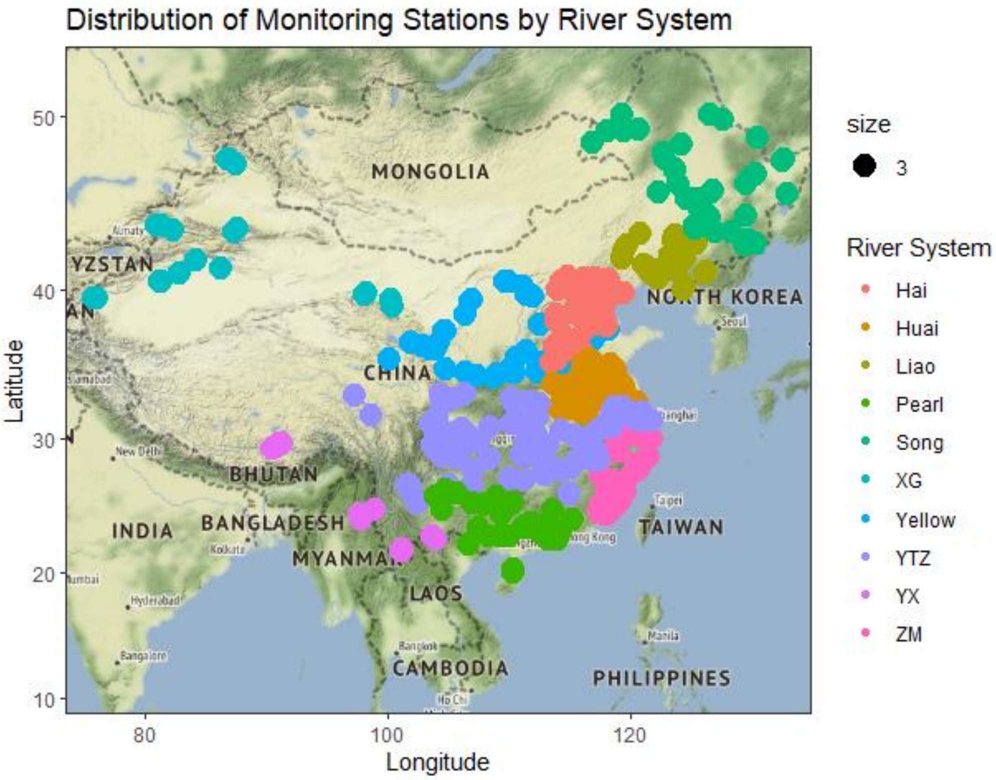
Third, we are able to **combine ggmap() with a function from the ggplot2 package** in order to plot spatial data. Assume we saved some coordinates concerning longitude in the columns long and latitude in lat belonging to the data frame called dat. As an example, you can use geom_point() to add these to the downloaded map.

```
#ggmap() in combination with geom_point()
ggmap(map.egypt) +
  geom_point(data = dat, aes(x = long, y = lat))
```

In the following task I added the get_map() command as a comment. Instead, we load the map from an already deposited .RDS file named map.china.RDS with help of readRDS(). We do so because files we download from servers may not be available at a later point in time.

Task: Just check the following chunk to create the map.

```
#load library
library(ggmap)
#command to download the map:
#get_map(location = c(73.5, 8.84, 134.78, 53.56), zoom = 4)
#load and show map
readRDS("map.china.RDS") %>%
ggmap() +
  geom_point(data=dat, aes(x = x, y = y, color = riversystem, size=3)) +
  xlab("Longitude") +
  ylab("Latitude") +
  theme_bw() +
  labs(title = "Distribution of Monitoring Stations by River System", col = "River System")
```



We can see that, apart from the so-called *Inland River Basin*, virtually all measuring stations are located in the eastern half of the country. This is not very surprising as the major river systems are almost exclusively located there.

As an example, try to find out which river system *ZM* stands for. You may have a look on the map provided in the following paper:

China’s River Systems - Map

After you did some research on your own, try to answer the following question:

Quiz: Which river system the categorical variable *ZM* stands for?

- Inland Rivers Basin [ ]
- Southwest Rivers Basin [ ]
- Southeast Rivers Basin [x]
- Yangtze River Basin [ ]

Furthermore, XG refers to the *Inland Rivers Basin*, YTZ to the *Yangtze Rivers Basin* and finally YX corresponds to the *Southwest Rivers Basin*.

Let us replace the abbreviations from above with more intuitive names. Helpful functions are mutate, recode() and recode_factor() from the dplyr package.

Task: Load the library dplyr first.

```
#load the library "dplyr"
library(dplyr)
```

In case you do not know yet, how to combine mutate() and recode() or recode_factor(), have a look at the info box below.

Info: How to recode a variable of a data set by combining mutate() and recode() or recode_factor().

Sometimes we are not satisfied with the manifestations of a variable, for instance if abbreviations are not really intuitive or we wish to replace them by what they stand for. However, there is a simple solution how to recode a variable. Recoding means to replace a set of manifestations with another manifestation. Let us look at the following example.

```
presidents = data.frame(c("Friedrich", "Vladimir", "Zemin", "Anwar", "Recep", "Gamal", "Jintao", "Hosni",
"Kemal"), c("GER", "RUS", "CHN", "EGY", "TRY", "EGY", "CHN", "EGY", "TRY"))
colnames(presidents) = c("name", "country")
presidents

##      name       country
## 1    Friedrich  GER
## 2    Vladimir   RUS
## 3    Zemin      CHN
## 4    Anwar      EGY
## 5    Recep      TRY
## 6    Gamal      EGY
## 7    Jintao     CHN
## 8    Hosni      EGY
## 9    Kemal      TRY
```

In the data set presidents created above, we list a few presidents' first names with their respective country. As you will notice, instead of the full names of the countries only abbreviations are given. Let us say, we wish to replace the abbreviations with the full names, that is, recode the column country.

In short, we overwrite the column using mutate() with the recoded column created by recode(). In general, we just need to supply recode() with the name of the column to recode and the manifestations we want to recode as follows:

- *GER: Germany*
- *RUS*: *Russia*
- *CHN: China*
- *EGY*: *Egypt*
- *TRY*: Turkey

```
presidents.recoded = presidents %>%
  mutate(country = recode(country, "GER" = "Germany", "RUS" = "Russia", "CHN" = "China", "EGY" =
"Egypt", "TRY" = "Turkey"))
```

Let us have a look at what has changed:

```
#original
presidents

##      name        country
## 1    Friedrich   GER
## 2    Vladimir    RUS
## 3    Zemin       CHN
## 4    Anwar       EGY
## 5    Recep       TRY
## 6    Gamal       EGY
## 7    Jintao      CHN
## 8    Hosni       EGY
## 9    Kemal       TRY

#recoded
presidents.recoded

##      name        country
## 1    Friedrich   Germany
## 2    Vladimir    Russia
## 3    Zemin       China
## 4    Anwar       Egypt
## 5    Recep       Turkey
## 6    Gamal       Egypt
## 7    Jintao      China
## 8    Hosni       Egypt
## 9    Kemal       Turkey
```

As you can see, recode() has successfully replaced the abbreviations of all values in country with their full names. If you want to learn more about this function, have a look at its documentation.

If you do not know yet about the mutate() function from the dplyr package have a look here.

Task: Replace ___ with the name of the column we wish to recode.

```
# dat = dat %>%
#   mutate(___ = recode_factor(___, "XG" = "Inland", "YTZ" = "Yangtze", "YX" = "Southwest", "ZM" = "Southeast"))

dat = dat %>%
  mutate(riversystem = recode_factor(riversystem, "XG" = "Inland", "YTZ" = "Yangtze", "YX" = "Southwest", "ZM" = "Southeast"))
```

Just by looking on the map it is hard to say how many monitoring stations belong to each river basin. To calculate which river system the most monitoring stations are assigned to, we will group the data by river system and count the number of rows. This will give us a list of river systems with the corresponding number of monitoring stations.

We start by loading the dplyr package which provides the group() function to group the data set by riversystem. Then, we apply the summarise() function from the same package in order to count the number of monitoring stations per group. Furthermore, arrange it decreasingly by this column.

If you need help with the two functions, don’t hesitate to have a look into the info box below:

Info: How to perform operations with summarise() on groups defined with group_by().

Often, we are interested in calculations based on specified groups in data set. For instance, if we have a data set of employees, we are probably not only interested in the total number of employees, but rather in the number of males and females. Therefore, grouping by sex would create two subsets on which we would like to perform the n() operation on.

To do so, we need to perform two steps basically, for which we make use of summarise() and group_by() provided by the dplyr package. Furthermore n() is used to give the group size.

First, we group by the variable sex we want to use for creating subsets on our data set dat. In general, we are able to group by several variables. In this example we save the grouped data in dat_grouped. Don’t forget to load the dplyr package in advance.

```
#load the dplyr package
library(dplyr)
#group `dat` by `sex`
dat.grouped = group_by(dat, sex)
```

Second, we are now able to perform operations on the group level with help of the grouped data set dat_grouped. Use summarise() to do so. The operation we want to apply on dat_grouped is n(), which returns the size per group. If we are interested in other characteristics of the groups, we might also apply mean(), min() or max() for example to the variable specified within the parentheses.

```
summarise(dat.grouped, n())
```

To learn more about the possibilities of aggregating data sets, read here more about group_by()

Documentation group_by()

and summarise()

Documentation summarise()

As you probably notice in the prepared coded below, I used the operator %>%. It is used to pipe output from a previous function as input to the following function. To learn more, read the info box below.

Info: Pipe data using the %>% operator.

Instead of working with a lot of intermediate results, the pipe operator %>% from the dplyr package enables us to pass output from a preceding to a subsequent function.

For instance, we have the following code, creating lots of intermediate results, we don't actually reuse later:

```
dat = mutate(dat, var_1 = var_1*2)
dat = group_by(dat, var_1)
dat = filter(dat, var_2 == 5)
summarise(dat, no_var = n())
```

With help of the pipe operator %>% it can be rewritten as follows:

```
dat %>%
  mutate(var_1 = var_1*2) %>%
  group_by(var_1) %>%
  filter(var_2 == 5) %>%
  summarise(no_var = n())
```

The result of both code statements is equivalent.

Try to find out to which river basin the most monitoring stations are assigned to in 2005, the last year before the new environmental policy became effective. For a better understanding, the analysis is split into two parts, grouping and summarizing.

Task: Replace ___ with the correct parameter to group dat by riversystem and save it to dat_grouped. The command to filter for observations from 2005 is already given.

```
# #group data by "riversystem"
# dat.grouped = group_by(dat, ___) %>%
#   filter(year == 2005)
#
dat.grouped = group_by(dat, riversystem) %>%
  filter(year == 2005)
```

After we grouped and filtered the data, let us calculate the summary statistic.

Task: Replace ___ with the correct parameter to calculate the number of monitoring stations per river system. The command to sort by the number of stations is already given.

```
# #count the number of monitoring stations (per river system)
# summarise(___, no.stations = n()) %>%
#   arrange(desc(no.stations))
#


summarise(dat.grouped, no.stations = n()) %>%
  arrange(desc(no.stations))
```

```
##    riversystem no.stations
## 1      Yangtze         104
## 2         Huai          86
## 3          Hai          65
## 4       Yellow          44
## 5         Song          42
## 6         Liao          37
## 7    Southeast          32
## 8        Pearl          31
## 9       Inland          28
## 10   Southwest          17
```

The summarise() function returns a tibble consisting of two columns. The first column lists all river systems, while the second one is the sum of all observations per river systems in 2005. As we used arrange() in combination with desc() in the last row of the code statement, the river system on the top is the one the most monitoring stations are assigned to.

Quiz: Which river system the most monitoring stations are assigned to?

- Huai River Basin [ ]
- Southwest Rivers Basin [ ]
- Yangtze River Basin [x]
- Yellow River Basin [ ]

As we see, the most stations - 104 - belong to the Yangtze River Basin, while the fewest - 17 - belong to the Southwest Rivers Basin. The first-mentioned river system's namesake, the Yangtze, is also the longest river in Asia.

We conclude this exercise by examining how many provinces the individual river systems run through. Helpfully, we already prepared the data we need in dat.grouped. In order to properly calculate the number of provinces per river system we will shrink the data set to the columns riversystem and prov. Before we calculate the summary statistics, we want to count the unique provinces only which is done by applying unique().

Task: Replace ___ with the function that returns the current group size. The group size is the number of observations per group.

```
#
# dat.grouped %>%
#   select("riversystem", "prov") %>%
#   unique() %>%
#   summarise(no.provinces = ___)
#


dat.grouped %>%
  select("riversystem", "prov") %>%
  unique() %>%
  summarise(no.provinces = n())
```

```
##        riversystem no.provinces
## 1      Inland         2
## 2      Yangtze       14
## 3      Southwest      2
## 4      Southeast      2
## 5      Yellow         8
## 6      Hai            6
## 7      Huai           4
## 8      Liao           3
## 9      Song           3
## 10     Pearl          5
```

Not very surprisingly, the river system the most monitoring stations are assigned to, the Yangtze River Basin, is also the one that runs through the largest number of provinces, 14. For the other river systems, the number is between two and ten. An important insight is, that every river system in the data set is exposed to the river border pollution difficulty.

Summary

With help of a map, we got to know that most of the river systems are located in the Eastern half of China as well as the monitoring stations are. The most monitoring stations are assigned to the Yangtze river system, that is also the namesake of the longest river in Asia. What is essential to our analysis is that each river system crosses at least two up to six provinces.

After we have familiarized ourselves with the river system variables, we would like to analyze the pollution indicators descriptively. In particular, we want to compare the water pollution trends at boundary and non-boundary stations over the observation period from 2004 until 2010.

Exercise 2.3 – Descriptive Statistics on Water Quality

In this section, we will take a closer look at water pollution at non-border compared to border-stations as well as at the time trend of the pollution levels between 2004 and 2010.

If you want to read up on the meaning of the key figures, I recommend to have a look at the info box *Description of Water Pollution Indicators* in the appendix exercise A3. COD Dynamics Versus Other Indicators of Water Pollution.

Before, we start let us load the data set.

Task: Read the file Regression_data.dta with help of read_dta() and save it in dat.

```
#read "dat.RDS" with help of readRDS()
dat = read_dta("Regression_data.dta")
```

How many non-boundary stations existed in 2005? Let us find out:

Task: Just check the following code to find out how many non-border and border-monitoring stations exist in 2005:

```
no.stations = dat %>%
  filter(year == 2005)
table(no.stations$boundary)

##
##   0   1
## 361 125
```

The variable boundary hereby is an indicator variable that is 1 for a monitoring station at the border and 0 otherwise.

Quiz: How many interior stations did exist in the year 2005?

- 361 [x]
- 125 [ ]
- 486 [ ]
- 231 [ ]

While 125 boundary monitoring stations existed, there have been 361 stations in the interior of the provinces. Hence, one out of four stations lies at the provincial borders.

We already learnt about the motivation for the passages in the 11th Five-Year Plan imposing a stricter control of river border pollution and this study in the first exercise. The water quality at the provincial borders in 2005 is far worse than in the interior of the provinces. While interior stations exhibited an average value of 7.05 mg per Liter, the COD level at the border is 10.59 mg per Liter. If governors and party secretaries are held responsible for environmental conditions in the province and not for negative externalities occurring in other provinces due to their actions, they maximize their benefits by optimizing their own economic development only, ignoring spillover effects on other provinces.

However, this was only a snapshot. The reform in 2005, through which the central government focused its attention on COD, should lead to an overall pollution reduction since the main contributors of COD emit other pollution indicators, too. Hence, a shrinking difference between measurements of other indicators at the border and in the interior of the provinces should have been taken place. Consequently, this difference should not have decreased to the same extent for other pollution indicators aside from COD, as there were still no incentives set for this by the superordinated central government.

To further analyze the development of the water pollution indicators over time, we need to prepare the data set first. Before we start, let us create a data set that is limited to the data we need.

If we would not like to have a look on the first six rows, but on randomly chosen ones, we might apply slice_sample(). The first parameter to set is the data set, while the second defines the number of rows to display. If you do not know how, please have a look at the info box below.

Info: How do you get a first overview of your dataset with slice_sample()?

Sometimes you do not want to show the first six rows of a data set, but rather view a randomized sample. Slice_sample() is a simple but nonetheless helpful function, which allows you to draw a specific number of randomly chosen rows from the data set you want to get an overview of. Basically, it asks for two parameters. The first one is the name of the data frame, while the second one, n is the number of rows to draw. Since it's part of the dplyr package, so do not forget to load it in advance:

```
library(dplyr)
```

Let us say we want to draw eight random rows from a data set called dat. The corresponding command would look like that:

```
slice_sample(dat, n = 8)
```

For further information have a look at the `slice_sample()` documentation.

Sometimes it is far more convenient to work with the variables you need only. In the following task, we will shrink the data set to ten columns.

Task: Replace ___ to show ten randomly chosen rows using slice_sample(). Then, check the chunk to create a more compact data set called dat.comp.

```
#
# #create reduced data set
# dat.comp = dat[, c(2:11)]
# #show head
# slice_sample(dat.comp, n = ___)
#


dat.comp = dat[, c(2:11)]
slice_sample(dat.comp, n = 10)
```

```
##    station  boundary year  prov  cod  bod   nh   petroleum phenol mercury
## 1    L015       0     2007   21  13.5 30.8 6.09    0.160    0.013   0.02
## 2   HU015       0     2006   41   4.8  3.8 1.22    0.050    0.001   0.02
## 3   HA023       1     2010   41  12.3  8.5 4.74    0.165    0.004   0.03
## 4    H005       1     2008   62   2.6  2.4 0.33    0.010    0.001   0.03
## 5   HA065       0     2009   13  34.7  4.6 1.45    0.030    0.002   0.08
## 6    L001       1     2009   21   2.3  2.1 0.21    0.000    0.001   0.00
## 7    S005       0     2010   23   5.9  1.6 0.18    0.025    0.000   0.00
## 8    O038       0     2007   65   2.8  1.4 0.18    0.010    0.001   0.02
## 9    O046       0     2010   65   2.4  1.5 0.18    0.010    0.001   0.03
## 10  HU027       0     2007   34   8.4  6.7 4.32    0.230    0.001   0.08
```

In order to apply the combination of group_by() and summarise() to the data set, we need key-value pairs. This is done by converting the data to the long-format. In the next task we want to apply pivot_longer() to do so. If you want to learn more about it, have a look at the info box below, before you continue.

Info: The long format - how pivot_longer() helps

In order to properly aggregate or plot data, it is given in so called key-value pairs ideally - that is the long format. To convert data from the wide to the long format, we apply pivot_longer.

After loading the package tidyr you have to specify at least two parameters: * data: The data set you wish to convert * cols: The columns which contains the values

Nonetheless, it is possible to set the new column names already: * names_to: the name of the new column containing the *keys* * values_to: the name of the new column containing the *values*

Let us assume the values are in the columns val_1 to val_5 of the data set dat.

```
#Load `tidyr`
library(tidyr)
#apply pivot_longer
pivot_longer(data = dat, cols =  val1:val5)
```

In order to rename the name and value column, just set the names_to and values_to columns accordingly.

Further information can be found here:

- Example and
- Documentation of `pivot_longer()`

Task: Replace ___ with the columns to pivot into the longer format. If you do not know what to do, maybe the info box above will help you. Note that the values are the measurements of all six pollution indicators.

```
# dat.long = dat.comp %>%
#   pivot_longer(cols =  ___, names_to = "substance", values_to = "measurement")


dat.long = dat.comp %>%
  pivot_longer(cols =  cod:mercury, names_to = "substance", values_to = "measurement")
```

The *key* variable is substance and the *value* column is measurement. Let us have a look at what has changed by comparing dat.comp and dat.long:

Task: Use head() twice in total to view the first six rows of dat.comp and dat.long.

```
#show the first six rows of dat.comp
#show the first six rows of dat.long
head(dat.comp)

##   station boundary year  prov cod  bod   nh   petroleum phenol mercury
## 1    H001        0  2008   63   1.8  0.5  0.19      0.01      0.001    0.01
## 2    H001        0  2007   63   2.1  1.5  0.24      0.01      0.001    0.01
## 3    H001        0  2004   63   2.4  1.0  0.13      0.01      0.001    0.05
## 4    H001        0  2005   63   2.7  1.0  0.14      0.01      0.001    0.02
## 5    H001        0  2009   63   1.7  1.0  0.15      0.01      0.001    0.01
## 6    H001        0  2006   63   1.3  1.0  0.21      0.01      0.001    0.01

head(dat.long)

##     station boundary year prov substance   measurement
## 1    H001        0   2008   63       cod          1.800
## 2    H001        0   2008   63       bod          0.500
## 3    H001        0   2008   63        nh          0.190
## 4    H001        0   2008   63 petroleum       0.010
## 5    H001        0   2008   63    phenol        0.001
## 6    H001        0   2008   63   mercury        0.010
```

Compare both data sets - what has changed? Now, all measurements are in the measurement column, while substance tells us which water pollution indicator the observation belongs to.

As you can see, the second column still contains the dummy variable regarding the location of the measuring station and the third column indicates the year of the measurement. However, the six columns containing the pollution measurements have been disappeared. Instead, in their place are the columns substance and measurement. The categorical variable substance is indicating to which indicator the measurement value in the measurement column belongs to. Thus, pivot_longer() has changed the organization of the data set, but no single data point has been lost. This makes it easier to plot, for example, a line chart with ggplot2.

We have already used the combination of group_by() and summarise() before. Just check the next code to apply it onto the data set we just prepared.

Task: Check the chunk in order to finish this task.

```
dat.mean = dat.long %>%
  group_by(boundary, year, substance) %>%
  summarise(measurement = mean(measurement)) %>%
  ungroup()
```

When applying aggregation operations as shown before through the combination of group_by() and summarise() we usually convert data into the long format. However, it is often not very convenient to interpret data that is given in the form of key-value pairs. In that case we would like to transform tables to the wide format. The tidyverse package helps us with the pivot_wider() function. If you want to learn how to use this function, check out the info box below.

Info: The wide format - how pivot_wider() helps

In order to conveniently interpret data, you usually don't prefer key-value pairs. However, sometimes data is given in the long format. To convert data from the long to the wide format, we apply pivot_wider.

After loading the package tidyr you have to specify at least three parameters. * data: The data set you wish to convert * names_from: The column(s) which contain(s) the names of the output columns * values_from: The column which contains the cell values of the output columns

Let us assume the values are in the column val of the data set dat. Furthermore, the columns names.1 and names.2 contain the names of the output columns of the widening process.

```
#Load `tidyr`
library(tidyr)
#apply pivot_wider
pivot_wider(data = dat, names_from = c(names.1, names.2), values_from = val)
```

In order to rename the name and value column, just set the names_to and values_to columns accordingly.

Further information can be found here: Example Documentation of `pivot_wider()`

In the next step, we apply pivot_wider() to dat.mean. We aim at having one column per water pollution indicator that contains the measurement value per year and boundary dummy.

Task*:* Replace ___ with the name of column of dat.mean we want to get the cell values from. Then, check the chunk to complete the task.

```
# #replace ___ with the correct column name that contains the required cell values
# dat.wide = dat.mean %>%
#   pivot_wider(names_from = c(substance, boundary), values_from = ___)
# #show the wide table
# dat.wide

#replace ___ with the correct column name that contains the required cell values
dat.wide = dat.mean %>%
  pivot_wider(names_from = c(substance, boundary), values_from = measurement)
#show the wide table
dat.wide
```

```
##   year   bod_0   cod_0 mercury_0    nh_0 petroleum_0 phenol_0   bod_1    cod_1
## 1 2004 7.16944 7.42222   0.04261 2.28656     0.15847  0.00767 9.01360 12.59440
## 2 2005 6.14349 7.04598   0.03398 2.08921     0.12369  0.00650 8.29920 10.59120
## 3 2006 6.77734 7.05382   0.03652 2.14246     0.10708  0.00669 7.38361  9.47951
## 4 2007 5.59169 6.27452   0.02978 2.00291     0.10551  0.00420 7.56529  9.36942
## 5 2008 3.75442 5.17210   0.02859 1.72597     0.06113  0.00374 6.80738  8.06311
## 6 2009 3.37064 4.64626   0.02947 1.54346     0.05706  0.00282 5.08917  6.42583
## 7 2010 3.35983 4.49834   0.02961 1.35812     0.05006  0.00215 4.77967  5.77236
##   mercury_1    nh_1 petroleum_1 phenol_1
## 1   0.03432 3.42232     0.19968  0.00633
## 2   0.03826 3.43710     0.19374  0.00846
## 3   0.03566 2.90434     0.14525  0.00950
## 4   0.02417 3.18736     0.16264  0.00819
## 5   0.03164 2.68000     0.12869  0.01483
## 6   0.02900 2.51133     0.11225  0.00343
## 7   0.02992 2.19943     0.07533  0.00255
```

Let us create a table that is easier to interpret. For a more beautiful HTML rendering we make use of the cableExtra package. If you want to learn more about this useful extension, have a look at the info box below.

Info: cableExtra

The cableExtra package helps you to create complex HTML tables. Thanks to a similar grammar that for example the ggplot2 package is using, it is very convenient to add several *layers* with help of the pipe operator %>% by passing the result along the chain. As a result you are able to create more beautiful tables for HTML applications.

Let us briefly explain the most important parameters and functions used in the problem set:

- align: set where you want to position the table - on the left, in the center or on the right
- caption: the name of the table
- col.names: the names of the columns
- kable_classic(): a customized table theme
- add_header_above(): adds an additional row containing labels above the table
- kable_styling(): allows to manually change the appearance of the table

Read more about the package at its documentation.

Let us create a more beautiful and intuitive table here. Just check the following chunk. You do not have to understand everything what is happening in the code chunk.

Task: Just check the following code in order to display a more beautiful representation of dat_wide. First, it arranges the columns, second, it rounds the values to two digits and finally it applies the kableExtra package for a more beautiful representation of the table.

```r
#arrange columns
dat.wide = dat.wide[,order(colnames(dat.wide))]
dat.wide[,c(13,1:12)] %>%
#round values
  mutate(across(where(is.numeric), round, digits = 2)) %>%
#set table style
  kbl(align = "c", caption = "Pollution Levels at Boundary and Non-boundary Stations from 2004 - 2010", 
col.names = c("Year", rep(c("Nonborder","Border"), 6))) %>%
  kable_classic() %>%
  add_header_above(c(" " = 1, "COD(mg/L)" = 2, "BOD(mg/L)" = 2, "NH(mg/L)" = 2, "Petroleum(ug/L)" = 
2, "Mercury(ug/L)" = 2, "Phenol(ug/L)" = 2)) %>%
  kable_styling(full_width = T)
```

Pollution Levels at Boundary and Non-boundary Stations from 2004 - 2010

	COD(mg/L)		**BOD(mg/L)**		**NH(mg/L)**		**Petroleum(ug/L)**		**Mercury(ug/L)**		**Phenol(ug/L)**	
Year	**Non-border**	**Border**	**Non-border**	**Border**	**Non-border**	**Border**	**Non-border**	**Border**	**Non-border**	**Border**	**Non-border**	**Border**
2004	7.17	9.01	7.42	12.59	0.04	0.03	2.29	3.42	0.16	0.20	0.01	0.01
2005	6.14	8.30	7.05	10.59	0.03	0.04	2.09	3.44	0.12	0.19	0.01	0.01
2006	6.78	7.38	7.05	9.48	0.04	0.04	2.14	2.90	0.11	0.15	0.01	0.01
2007	5.59	7.57	6.27	9.37	0.03	0.02	2.00	3.19	0.11	0.16	0.00	0.01
2008	3.75	6.81	5.17	8.06	0.03	0.03	1.73	2.68	0.06	0.13	0.00	0.01
2009	3.37	5.09	4.65	6.43	0.03	0.03	1.54	2.51	0.06	0.11	0.00	0.00
2010	3.36	4.78	4.50	5.77	0.03	0.03	1.36	2.20	0.05	0.08	0.00	0.00

In the table, we find the average concentrations of COD, BOD, NH in milligrams per liter, while the concentrations of petroleum, mercury and phenol are given in micrograms per liter. We distinguish between the values measured at the measuring stations in the interior of a province, which is the left column for each pollution indicator, and those detected at the border, in the right column. The results from 2004 to 2010 are arranged in descending order.

We clearly see a gap between all the pollution metrics. This always indicates - except in the case of the mercury content - a stronger pollution of the rivers near the border. Over the observation period, however, the values decrease steadily. Furthermore, we can observe that the gap between the measurements at boundary and non-boundary stations is shrinking over time.

Before we start introducing the DiD approach and using regressions in the third section, we might have a look on the development of the different pollution indicators graphically in addition. To do so, we make use of the facet_wrap() function. Supplying it with the parameter substance, it will create one line diagram per water pollution indicator.

Info: How to use facet_wrap() to create several plots at once by grouping a data set through defining faceting groups.

In case you want to create a series of plots that differ by one faceting variable with several levels, facet_wrap() is a good choice. It uses screen space better than facet_grid(). Just add the command to a typical ggplot2 command series as shown in the following example:

```
#ggplot function
ggplot() +
  #add the plot function of your choice, for instance geom_line()
  geom_line(mapping = aes(x = x_values, y = y_values), data = dat) +
  #add faceting function and variable as a formula
  facet_wrap(~ facet_variable)
```

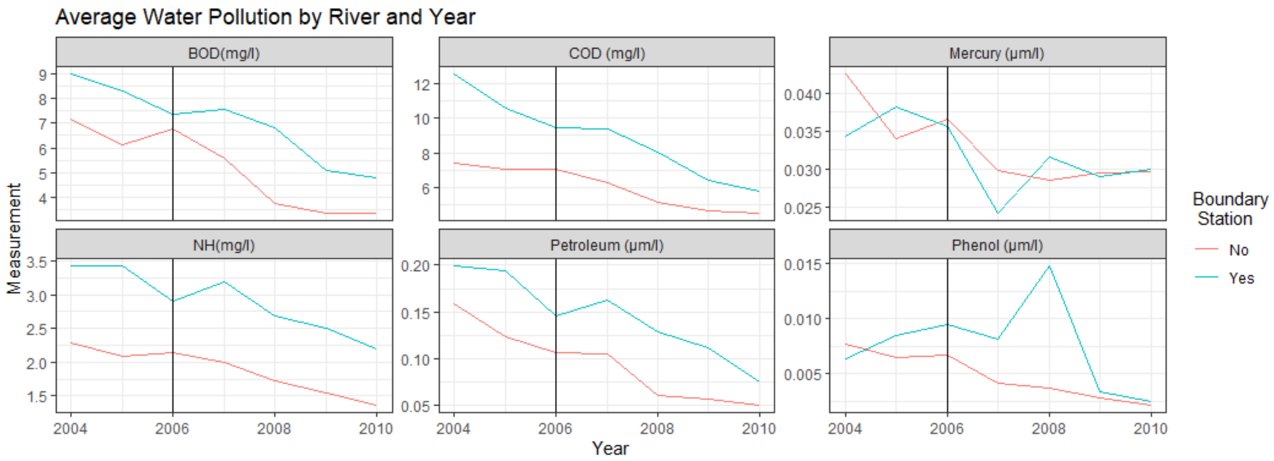
Furthermore, we set two parameters in the task below:

- scale: Either you fix all scales or set one of them or both free.
- labeller: Here you set labels for the faceted diagrams.

Have a look at the documentation of facet_wrap() if you want to learn more.

Task: Supply facet_wrap with the correct parameter through replacing ___ with the name of the corresponding column name in pol_trend_mean to create one plot per indicator.

```
#
# ggplot() +
#   geom_line(mapping = aes(x = year, y = measurement, color = as.factor(boundary), group =
as.factor(boundary)), data = dat.mean))9 +   facet_wrap(~ ___, scales = "free_y", labeller =
labeller(substance = c(cod = "COD (mg/l)", bod = "BOD(mg/l)", nh = "NH(mg/l)", petroleum = "Petroleum
(µm/l)", mercury = "Mercury (µm/l)", phenol ="Phenol (µm/l)"))) +
#   scale_color_discrete(labels = c("No", "Yes"), name = "Boundary \n Station") +
#   xlab("Year") +
#   ylab("Measurement") +
#   ggtitle("Average Water Pollution by River and Year") +
#   geom_vline(xintercept = 2006) +
#   theme_bw()
#


ggplot() +
  geom_line(mapping = aes(x = year, y = measurement, color = as.factor(boundary), group =
as.factor(boundary)), data = dat.mean) +
  facet_wrap(~ substance, scales = "free_y", labeller = labeller(substance = c(cod = "COD (mg/l)", bod =
"BOD(mg/l)", nh = "NH(mg/l)", petroleum = "Petroleum (µm/l)", mercury = "Mercury (µm/l)", phenol
="Phenol (µm/l)"))) +
  scale_color_discrete(labels = c("No", "Yes"), name = "Boundary \n Station") +
  xlab("Year") +
  ylab("Measurement") +
  ggtitle("Average Water Pollution by River and Year") +
  geom_vline(xintercept = 2006) +
  theme_bw()
```



Finally, you find six line diagrams here, one per water pollution indicator. They represent the average measurement of each substance per year and are grouped into boundary (blue) and non-boundary stations (red). As seen before in the table, we can easily spot the gap between non-border and border stations for the concentrations of COD, BOD, NH and petroleum. While the development for those indicators is quite similar, mercury and phenol deviate.

Summary:

What we saw in both the table and the line diagrams is that the overall water quality has improved dramatically. However, it is still hard to derive whether COD emissions declined more than others due to the regime change. Since we want to find our whether the regime change had a stronger effect on the water quality at borders compared to the interior, it is not sufficient to analyze the data descriptively.

To find out whether there is indeed substantial stronger COD water pollution progress at river boundaries compared to the interior of the provinces, we need further methods and models. Therefore, we set up the DiD approach manually and by regression in the next exercise to examine the effectiveness of the environmental regulation reforms defined in the *Eleventh Five-Year-Plan*.

Exercise 3 – Water Pollution Dynamics at Borders

In the last exercise, we saw that the COD level indeed decreased after 2006. But how can we be sure that this is the consequence of the policy change induced by the *Eleventh Five-Year-Plan*? How can we find out whether there is a causal relationship? How can we exclude other causes that could be causal? This leads us to the approach we will present in Exercise 3.

The study is based on regression analyses, using the difference in differences (DiD) approach. In this section, we will assemble the main regressions performed by the authors step by step. Therefore, we will first present the method by applying it manually before performing a DiD estimation with the OLS procedure.

The DiD method tests whether the modification of the promotion incentives for governors in 2005 had a stronger effect on the water pollution reduction at borders compared to the interior of the provinces. Hence, it compares the pollution measurements before and after 2005, between boundary and non-boundary stations.

We will check whether the assumptions made to apply the DiD approach are fulfilled and implement the method not only manually but also by regression in the second sub-exercise. In this way, we benefit from automatically calculated standard errors and other statistics as well as from the options to include fixed effects and control variables. It is important to note, that the data set is clustered in several dimensions. Hence, standard errors must be calculated in a cluster robust fashion. We will cover both in the third and fourth sub-exercise.

Structure

3.1 Introduction to the DiD Approach

3.2 Implementation of the DiD Approach by Regression

3.3 Clustered Standard Errors

3.4 Fixed Effects and Control Variables

Exercise 3.1 – Introduction to the DiD Approach

The DiD estimator is used to estimate the causal effect of a treatment by calculating the change of the difference between a treatment and control group after starting an experiment. It heals the problem of often not being able to control for all background changes and empowers us with getting quite consistent estimators even under weak conditions (Angrist, 2008, p.221).

In the study we examine, the model is set up as follows:

- Treatment group: Border monitoring stations.
- Control group: Non-border monitoring stations.
- Pre-experimental period: 2004 and 2005 (before the change in promotion incentives).
- Experimental period: 2006 until 2010 (after the change in promotion incentives).

The stations at the border are considered as treated, since the policy change should set strong political promotion incentives to reduce water pollution at borders, while monitoring stations in the interior of the provinces are used as control group. We assume that incentives to reduce non-border pollution should not have changed afterwards. Since the incentives changed after 2005, our pre-experimental period lasts for two years, 2004 and 2005. It is worth discussing whether two years are sufficient to state that the pollution trends would have followed parallel paths in case the environmental law would not have changed. The years after the policy change, 2006 until 2010 represent the experimental period.

The basic idea of the DiD approach is that if no treatment would have taken place, both the group that is considered as treated as well as the control group would have followed parallel paths over time (Angrist, 2008, p.230). At one point in time, an event happens that affects only one group and changes the course of its path. This group is considered as the treated group and the effect we measure is assumed to be causal for the change in the group’s path. Since the assignment of the study objects or individuals is done randomly, it mimics a natural experiment in order to estimate causal effects (Cunningham, 2021, p.409). Hence, it is a quasi-experimental identification strategy (Cunningham, 2021, p.406).

Let us think about what this means to the study here. If the government would not have tightened the environmental law regarding river border pollution, no change in the difference between the extent of water pollution at borders and non-borders - in other words treatment effect - should have been observed. This is the premise under which the approach can work in the first place.

Quiz: If we assume that the water pollution at borders indeed decreased stronger than in the interior in the years after 2005, what does it mean to the treatment effect?

- The treatment effect is negative. [x]
- There is no treatment effect. [ ]
- The treatment effect is positive. [ ]

If the water pollution changed as described above, the treatment effect is negative, meaning that after the policy change became effective it led to a stronger reduction of non-boundary stations.

Now that you know the correct answer to the preceding question, what does it imply about the value that the DiD estimator takes?

Quiz: What is the range of values of the DiD estimator if the treatment effect is negative?

- The DiD estimator must be positive. [ ]
- The DiD estimator must be zero. [ ]
- The DiD estimator must be negative. [x]

A stronger pollution reduction at borders means a negative treatment effect, leading to a negative DiD estimator as well. Let us calculate the DiD estimator manually now as follows:

Manual Calculation of the DiD-Estimator

The formula to compute the Difference-in-Difference Estimator looks as follows:

$$DiD = \left(\acute{y}_{exp,tr} - \acute{y}_{exp,co}\right) - \left(\acute{y}_{pre,tr} - \acute{y}_{pre,co}\right)$$

with the average value of

- the treatment group during the experimental period $\acute{y}_{exp,tr}$;
- the control group during the experimental period $\acute{y}_{exp,co}$;
- the treatment group during the pre-experimental period $\acute{y}_{pre,tr}$;
- the control group during the pre-experimental period $\acute{y}_{pre,co}$.

Now we want to apply the formula to our example manually. To do so, let us calculate the means of chemical oxygen demand (COD) for the treatment and experimental group during the pre-experimental as well as during the experimental period, that is, four results in total. We split this procedure into two parts. First, we will calculate the means in the formula above in order to calculate the estimator afterwards.

Before we start, we need to load our data set again. From now on, we will only load a modified version of the author's data set which I have already adapted to the needs of this problem set. This save us not having to make the same changes over and over again in each new exercise.

I saved the data set in the *R*-related file type .*RDS*. Hence, we will need the function readRDS() from *R*'s base package. Just type in the name of the file you want to read as a parameter without loading an additional library. However, if you still need any help, try to retrieve the help file of the function by running the command ?readRDS.

Task: Just check to save the data set in dat.

```
#read dat.RDS with readRDS() and save it in dat
dat = readRDS("dat.RDS")
```

The variables in the following task are equivalent to the following components of the formula above:

- y_et $\hat{=}$ $\acute{y}_{exp,tr}$
- y_ec $\hat{=}$ $\acute{y}_{exp,co}$
- y_pt $\hat{=}$ $\acute{y}_{pre,tr}$
- y_pc $\hat{=}$ $\acute{y}_{pre,co}$

In the next task, we will calculate those estimators.

Task: Replace ___ either with 0 or 1 according to the description above to calculate the components of the formula correctly.

```
# y_et = dat %>%
#   filter(boundary == ___ & year >= 2006) %>%
#   select(cod) %>%
#   unlist() %>%
#   mean()
# y_ec = dat %>%
#   filter(boundary == ___ & year >= 2006) %>%
#   select(cod) %>%
#   unlist() %>%
#   mean()
# y_pt = dat %>%
#   filter(boundary == ___ & year < 2006) %>%
#   select(cod) %>%
#   unlist() %>%
#   mean()
# y_pc = dat %>%
#   filter(boundary == ___ & year < 2006) %>%
#   select(cod) %>%
#   unlist() %>%
#   mean()

y_et = dat %>%
  filter(boundary == 1 & year >= 2006) %>%
  select(cod) %>%
  unlist() %>%
  mean()
y_ec = dat %>%
  filter(boundary == 0 & year >= 2006) %>%
  select(cod) %>%
  unlist() %>%
  mean()
y_pt = dat %>%
  filter(boundary == 1 & year < 2006) %>%
  select(cod) %>%
  unlist() %>%
  mean()
y_pc = dat %>%
  filter(boundary == 0 & year < 2006) %>%
  select(cod) %>%
  unlist() %>%
  mean()
```

We used filter() with a variation of the filter argument to compute the means of the COD indicator of the according subsets. Hereby, & combines two expressions, one regarding the boundary dummy and one checking the year variable. Both must return TRUE to select the corresponding row.

Let us have a look at the intermediate results.

Task: Click on check to show the values of the formula`s components:

```
data.frame("PreExp.Treat" = round(y_pt, 2), "Exp.Treat" = round(y_et, 2), "PreExp.Cont" = round(y_pc, 2),
"Exp.Cont" = round(y_ec, 2)) %>%
  kbl() %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

PreExp.Treat	Exp.Treat	PreExp.Cont	Exp.Cont
11.59	7.82	7.23	5.52

As you can see, the average pollution levels after 2005 decreased at borders - from 11.59 to 7.82 - as well as in the interior of the provinces - from 7.23 to 5.52.

Task: Just choose the right variable names from above to complete the formula and calculate the DiD estimator.

```
# #First calculate the difference between the means of the two groups during the experimental period:
# y_e = ___ - ___
# #Second calculate the difference between the means of the two groups during the pre-experimental period:
# y_p = ___ - ___
# #Last but not least, put the two results you just wrote down the formula for into the following gaps to
correctly calculate the difference-in-difference estimator according to the formula above:
# DiD = ___ - ___
# DiD

y_e = y_et - y_ec
y_p = y_pt - y_pc
DiD = y_e - y_p
DiD

## [1] -2.060259
```

The DiD estimator has the value -2.06. Reconsider the formula and think about how we can interpret our result.

Quiz: How would you interpret the result of -2.06 we have got here when estimating the DiD estimator?

- After the government realized changes in promotional incentives, that is, from 2006 to 2010, the average COD level at border stations decreased by -2.06 more compared to a world in which the government would not have implemented those changes. [x]
- The overall pollution level in 2006 decreased by 2.06 compared to 2005. [ ]
- After the government realized changes in promotional incentives, that is, between 2006 and 2010, the average COD level at both border- and non-border stations decreased by -2.06 more compared to a world in which the government would not have implemented those changes. [ ]
- The overall pollution level in 2006 increased by 2.06 compared to 2005. [ ]

After the government changed the political promotion incentives to fight river border pollution the average COD level at borders decreased by - 2.06 more than it would have decreased if no action would have been taken. Remember, the average COD pollution during the pre-treatment phase at border stations has been 11.59. Since the DiD estimator is negative, it supports the thesis the political action has been successful.

Summary

In this sub-exercise we learnt that the DiD approach is an elegant way to circumvent problems that, for instance, arise from not being able to control sufficiently for unobservable confounding factors. Furthermore, we calculated the DiD estimator manually and interpreted it as the difference of how COD pollution has changed at borders and the interior of provinces after the policy change came into effect.

The more common way to compute the DiD estimator is to estimate it via linear regressions. By doing so, one benefits for instance from standard errors giving a hint about the statistical significance of the estimated effect. In addition, it allows to add fixed effects and control variables to overcome deficiencies regarding assumptions we need to make later. Therefore, we learn in the next section about how to prepare the data and conduct the estimation by a linear regression.

Exercise 3.2 – Implementation of the DiD Approach by Regression

After introducing the DiD approach and calculating the estimate manually, we learn how to implement the DiD estimation by regression. But why do we prefer estimating causal effects via linear regressions over the manual method shown in Exercise 3.1?

First, we are not only interested in the mere value of the estimated coefficients but also in the significance. Joyfully, **standard errors** and other statistics are estimated automatically as part of the linear regression functions in *R*. As a result, we can learn about the coefficient of interest's significance at a glance.

Second, recall which factors we considered in the previous sub-exercise. We calculated the DiD estimator such that the pollution only depends on the location - border or non-border - and time - before or after the regime change became effective. Although this is consistent with the basic idea of the DiD approach, possible weak assumptions recommend to consider other factors that may have a substantial influence on water pollution. Just think about varying levels of economic development or environmental factors between the monitoring stations or provinces. Hence, we would like to control for these effects. In contrast to the manual method, taking **fixed effects** and **control variables** into account can be done easily with help of linear regression functions.

This sub-exercise covers the first part we just talked about. The second part is introduced in Exercise 3.4. So let us start by loading the data set again.

Task: Load the data set dat.RDS and save it into a variable called dat with help of readRDS().

```
#use readRDS() to load "dat.RDS" in dat
dat = readRDS("dat.RDS")
```

Now we want to add a dummy variable that indicates whether an observation belongs to the pre- or post-experimental period. Any observation made in 2006 or later belongs to the period after the policy change and vice versa.

Task: Set the correct parameters in order to create a dummy variable called post06 that indicates whether an observation is made before or after the government changed the promotion incentives after 2005. Note that 1 shall indicate the post- and 0 the pre-experimental period.

```
#
# dat = dat %>%
#   mutate("post06" = ifelse(year>2005, yes = ___, no = ___))
#



dat = dat %>%
  mutate("post06" = ifelse(year>2005, 1, 0))
```

In this part of the third exercise, we will demonstrate how the manually calculated DiD estimator from the previous parts of the exercise can be obtained by using a linear regression. We will not go into the topic of linear regression itself, but rather focus on paper-specific problems such as panel data, fixed effects and clustered standard errors.

Let us have a look at the formula for the simple regression we focus on in this sub-exercise:

$$y_{it} = ß_0 + ß_1 T_t + ß_2 D_i + ß_3 (T_t \times D_i) + \varepsilon_{it}$$

where

- i indicates the individual
- t indicates the time
- y_{it} is the dependent variable
- T_t is a dummy variable that is 1 for an observation that is made after the experiment started and otherwise 0
- D_i is a dummy variable that is 0 for individuals from the control group and 1 for those being considered as treated
- ϵ_{it} is the error term at time t for individual i

Our aim is to calculate the effect of changes in political promotion incentives - that became effective from 2006 - on water pollution levels at the border. Let us apply the formula from above to our study.

Quiz: Which variable is the dependent one?

- the dummy variable indicating the post-experimental period [ ]
- the dummy variable indicating the group considered as treated [ ]
- the interaction between the two dummy variables [ ]
- one of the water pollution indicators [x]

We already learnt that the dependent variable is one of the water pollution indicators. On which side of the regression formula do you usually find the dependent variable?

Quiz: The dependent variable can be found on the …

- left side of the regression formula. [x]
- right side of the regression formula. [ ]

While the dependent variable is on the left side of the regression formula, it is straight forward for the independent variable(s) to be on the right side. The next step is to find out what our independent variables in the linear regression are. Note that we would like to get the same results as in the manually computed DiD estimator, so we exclude any fixed effects and control variables.

Quiz: Which variables are the independent ones?

- two dummy variables, one indicating the treatment status and another one the period [ ]
- two dummy variables, one indicating the treatment status and another one the period, further one interaction term between those two dummies [x]
- one dummy variable indicating the treatment status and the pollution indicator [ ]
- two dummy variables, one indicating the treatment status and another one the period, further one interaction term between those two dummies and the pollution indicator [ ]

In total, three independent variables are required for the regression formula. One dummy variable indicates whether the observation is considered as treated or not, which is the border dummy. Furthermore we have another dummy that indicates the period in which the observation is made. That's is the dummy we created in the previous section and indicates whether the observation is made after or before the political action took place. That means it says something about the period.

Let us summarize our findings on the regression formula:

$$pol.level_{it} = ß_0 + ß_1 post06_t + ß_2 border_i + ß_3(post06_t \times border_i) + \varepsilon_{it}$$

where

- i indicates the individual monitoring station
- t indicates the year
- $pol.level_{it}$ is the measured value of a certain water pollution indicator at monitoring station i in year t
- $post06_t$ is a dummy variable that is 0 before the experimental period started in 2006 and 1 during the experimental period from 2006 - 2010
- $border_i$ is a dummy variable that is 1 for station i being at the border and otherwise 0
- $post06_t \times border_i$ is the interaction term between the boundary and experimental period dummy
- ϵ_{it} is the error term at time t for monitoring station i

After learning about the regression formula we would like to apply it.

Task: Now regress the variables post06, boundary as well as the interaction between both variables on cod according to the formula above. Use the data set dat for doing so and save the regression results in reg. Note that we have chosen chemical oxygen demand here as an exemplary water pollution level.

If you do not know how to include interactions in lm(), read the info box below, please:

Info: Interaction Terms in lm()

Let us assume we wish to perform the following regression in R:

$$y = ß_0 + ß_1 x_1 + ß_2 x_2 + ß_3(x_1 \times x_2)$$

If you only want to include the interaction of two variables itself, you use :.

If you want to include the interaction of two variables as well as the variables themselves, you use *.

Hence, we are free to choose out of two options when implementing the regression formula from above:

```
#including both the interaction as well as the variables themselves implicitly
lm(y ~ x1*x2 )
#including the interaction only while adding the variables themselves explicitly
lm(y ~ x1 + x2 + x1:x2 )
```

```
reg = lm(cod ~ post06 + boundary + post06:boundary, data = dat)
```

After saving the regression results in reg, we can have a look on it using the function modelsummary() from the same name package. If you are further interested in this package, check out the info box below. Remember the result we have got when manually calculating the DiD estimator - -2.06. Compare it to the coefficient of the interaction term below.

Info: How to create customized tables to summarize one or more statistical models with help of the modelsummary package.

Modelsummary offers myriad ways of creating and modifying statistical models to your needs. In our case, we will use it to show the regression results in a manner that is easy to understand.

As there are too many parameters to mention here, I will just explain the most important parameters, which are,

- models: set of statistical models to show, in our case the names of the regression models;
- coef_omit: the names of coefficients we want to exclude from the table;
- coef_rename: for better comprehensibility we may want to modify some of the variable's names;
- gof_omit: the names of goodness-of-fit statistics we want to remove from the table and;
- title: the name of the table will be added as headline.

Have a look at the documentation to learn more about the options the package provides.

Task: Just check the chunk to show the regression results nicely.

```
modelsummary(list("COD Pollution" = reg), group = model ~ term, coef_rename = c("boundary" =
"Boundary", "post06" = "Post2006", "post06:boundary" = "Post2006 x Boundary"), stars = c('*' = .1, '**' =
0.05 ,'***' = .01), gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE|F|RMSE", title = "COD Discharges and Its
Determinants - Linear Regression") %>%
  kable_classic(full_width = TRUE) %>%
  kable_styling(font_size = 15)
```

COD Discharges and Its Determinants - Linear Regression

	(Intercept)	Post2006	Boundary	Post2006 x Boundary
COD Pollution	7.234***	-1.712***	4.359***	-2.060**
	(0.404)	(0.478)	(0.796)	(0.945)
*** p < 0.1, ** p < 0.05, *** p < 0.01**				

In the table you see the intercept and three further coefficients. These are the boundary and the post06 coefficient as well as the interaction term between the latter variables. The interaction term's coefficient is estimated the same as the result we got from the manual DiD calculation in the exercise before. It is significant on the five percent level. Now, we would like to know which of the coefficient measures the effect of the policy change on river border pollution.

Quiz: Which variable's coefficient stands for the effect of the change in promotion incentives on water pollution at borders?

- Boundary [ ]
- Post2006 [ ]
- Boundary x Post2006 [x]

The coefficient we are mainly interested in is that one documenting the effect of the interaction between the dummy variables boundary and post06, $ß_3$. It measures the effect of how the pollution indicator deviates after the new incentives regime came into force for border measurement stations compared to a world where the incentives have not been changed.

Quiz: Assuming the government's action in order to reduce the river border pollution has been successful, what should be the range of value the coefficient for the interaction term ß3 takes?

- The value of the coefficient must be greater than zero. [ ]
- The value of the coefficient must be smaller than one. [ ]
- The value of the coefficient must be zero. [ ]
- The value of the coefficient must be smaller than zero. [x]

The coefficient must be negative. In our regression, $ß_3$ results to -2.060 which indicates the regime change to be successful in preferably reducing river border pollution.

Quiz: What does the coefficient ß0 , that takes a value of 7.234 , stand for?

- The average COD value. [ ]
- The average COD value, but only at non-border monitoring stations. [ ]
- The average COD value, but only during the time before 2006. [ ]
- The average COD value, but only at interior monitoring stations before 2006. [x]

The first regression coefficient, $ß_0$ represents the intercept of the regression. Hence, it equals the average COD value measured at border monitoring stations in the years 2004 and 2005.

Quiz: What does ß1 , that is -1.712,, stand for?

- The difference between the average COD value measured at non-boundary and boundary stations before 2006. [ ]
- The difference between the average COD value measured at non-boundary and boundary stations after 2006. [ ]
- The difference between the average COD value measured at non-boundary stations before and after 2006. [x]
- The difference between the average COD value measured at boundary stations before and after 2006. [ ]

The second coefficient, $ß_2$ represents the gap between the average COD level measured at border compared to non-border monitoring stations before 2006.

In summary, this means that we can calculate four different scenarios using the four estimators:

- Average COD value at non-border monitoring stations before 2006: $ß_0 = 7.234$;
- Average COD value at non-border monitoring stations after 2006: $ß_0 + ß_1 = 7.234 - 1.712 = 5.522$;
- Average COD value at boundary monitoring stations before 2006: $ß_0 + ß_2 = 7.234 + 4.359 = 11.593$;
- Average COD value at boundary monitoring stations after 2006: $ß_0 + ß_1 + ß_2 + ß_3 = 7.234 - 1.712 + 4.359 - 2.060 = 7.821$;

Summary

The regression model examined here only included the dummy variables indicating whether an observation is made after 2006 or not, at the border or in the interior of the province. Besides that, we included the interaction term which is under special attention with regard to the research question of the paper. We estimate a value of -2.06 which is significant at the five percent level. However, it is questionable whether this model without any controls and fixed effects is sufficient with regard to possible deficiencies of the model's assumptions, for instance keeping the short pre-experimental period in mind. Hence, we will extend the regression model in the next sections.

Exercise 3.3 – Cluster-Robust Standard Errors

To refine the analysis, we introduce cluster-robust standard errors. This is crucial, since the default standard error calculation procedure underestimates the true variation due to some special characteristics in the data. In this sub-exercise, we want to explore these peculiarities and learn how to solve this issue. For this purpose, we focus on the regression model from *Exercise 3.2* applied to COD:

$$COD_{it} = ß_0 + ß_1 post06_t + ß_2 border_i + ß_3 (post06_t \times border_i) + \varepsilon_{it}$$

Before we start, we load the data set used before for our regressions.

Task: Just check the code to load the data set.

```
dat = readRDS("dat.RDS")
```

Furthermore, let us run the simple regression from the second sub-exercise again and store it in reg.

Task: Check the following chunk to estimate the simple regression again and store it in reg:

```
reg = lm(cod ~ post06 + boundary + post06:boundary, data = dat)
```

Standard errors tell us something about the quality of the estimated parameter. It allows us to derive how precise the estimators of the regression coefficients are. It is calculated as follows (Greene, 2002, p.75):

$$\widehat{SE}(\hat{\beta}_k) = \sqrt{[s^2(X^T X)^{-1}]_{kk}}$$

Where $\hat{\beta}_k$ is the estimator k, s the standard error of the regression, and X the covariance matrix.

The derivation of this formula is done under the assumptions of

- independence
- identical distribution

What is relevant to us is that it allows to make assumptions about the covariance matrix X. To go into detail, it is assumed that only the numbers on the diagonal (variances) are non-zero. Conversely, if the two conditions above are not fulfilled, there are non-zero values aside from the diagonal (covariances), too. As a consequence, the result will underestimate the true standard error (Abadie, 2017).

The study is based on panel data. It contains several units (stations) which can be grouped into clusters (river systems) and are observed several times (once per year). For panel data, especially the **independence** assumption is often violated (Petersen, 2008).

Let us try to check those assumptions graphically. First, we calculate the residuals and save them into an additional column in dat. The formula to calculate the residuals is as follows - it is derived by rearranging the terms from the simple regression formula shown above:

$$\varepsilon_{it} = COD_{it} - ß_0 - ß_1 * post06_t - ß_2 * boundary_i - ß_3 * (post06_t \times boundary_i)$$

Luckily, the stat package, that does not need to be loaded manually, provides a function called rstandard(), that calculates standardized residuals. We just need to set one parameter, the name of the regression we want to calculate the residuals of.

Task: Replace ___ with the function and argument that calculates the standardized residuals of the regression reg. The function cbind() adds the column containing the results to dat.

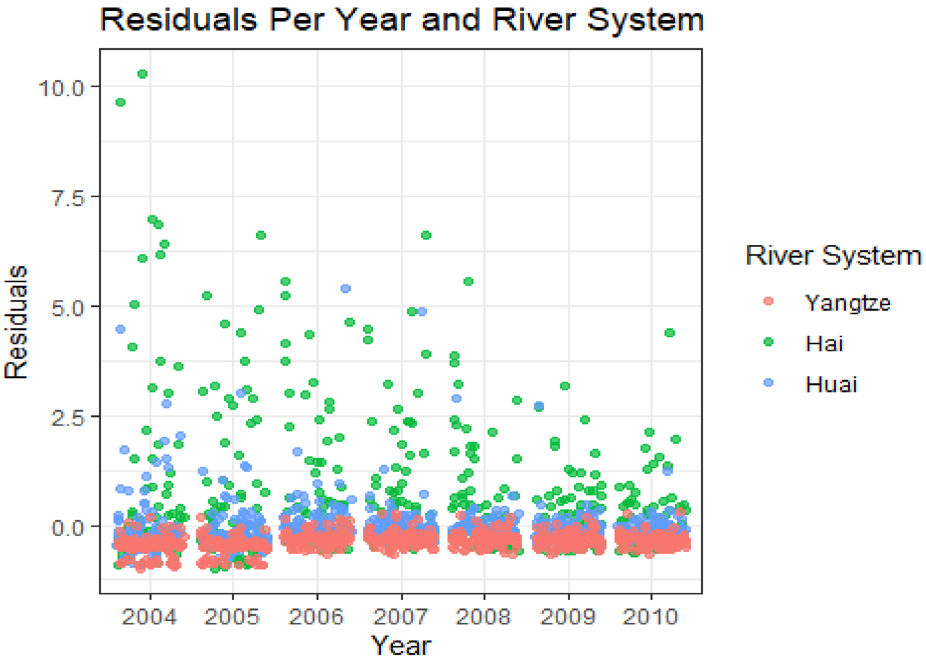
```
# dat = dat %>%
#   cbind(res = ___)

dat = dat %>%
  cbind(res = rstandard(reg))
```

Note that the name of the column containing the residuals is res.

If the residuals are indeed identically and independently distributed, the residuals of each river system should be randomly distributed with mean 0 for each cluster. To find out, let us plot the residuals res against year and color the data points according to the river system they belong to.

Task: Just check the next chunk to plot the residuals.

```
dat %>%
  filter(riversystem %in% c("Hai", "Yangtze", "Huai")) %>%
  ggplot(mapping = aes(x = as.factor(year), y = res, color = riversystem)) +
  geom_jitter(alpha = 0.7) +
  xlab("Year") +
  ylab("Residuals") +
  ggtitle("Residuals Per Year and River System") +
  scale_color_discrete(name = "River System") +
  theme_bw()
```

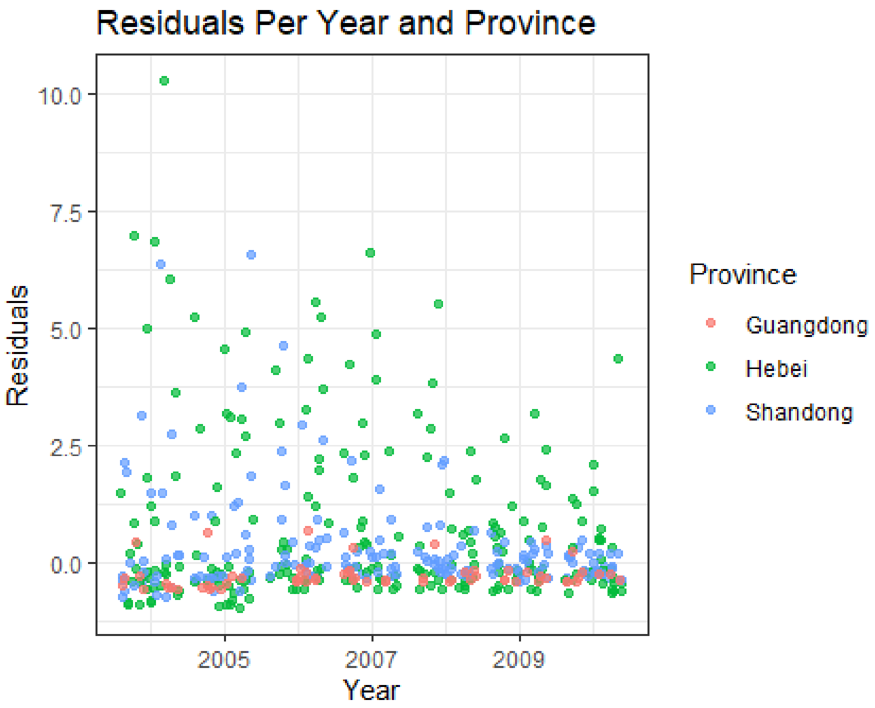


The residuals of the Hai He and Huai river systems are more scattered than for example those of the Yangtze river system. Furthermore, the extent of dispersion for both the Hai He and Huai river system seems to decrease by the years. Hence, we expect the error terms of different observations within the same clusters to be serially as well as spatially correlated across clusters. As a consequence, the *iid* assumption is violated what makes us interested in cluster-robust standard errors.

Incidentally, a similar observation must be made if we divide the residuals by province instead of river system. The reason for this is that each river system is tied to one or more specific provinces. Thus, if the residuals depend on the river systems and the river systems depend on the provinces, the residuals also depend on the provinces. Supplementary to the paper we will test this conjecture:

Task: Just check the next chunk to plot the residuals.

```
dat %>%
  mutate(riversystem = as.character(riversystem)) %>%
  filter(prov %in% c("Hebei", "Shandong", "Guangdong")) %>%
  ggplot(mapping = aes(x = year, y = res, color = as.factor(prov))) +
  geom_jitter(alpha = 0.7) +
  xlab("Year") +
  ylab("Residuals") +
  ggtitle("Residuals Per Year and Province") +
  scale_color_discrete(name = "Province") +
  theme_bw()
```



In the plot above the residuals for three provinces are shown exemplary. What we observe is similar to the results before, when we categorized by river system and confirms the we presumed before. Due to linkage between river systems and province, we see that the residuals of Hebei and Shandong are far more scattered than those of Guangdong. Hence, we see further evidence for violations of the *iid* assumption - the standard errors are clustered by province, too. For all regressions conducted in the paper, the researchers need to deal with potential heteroskedasticity and spatial correlation. So they decided to cluster the standard errors along two dimensions, by monitoring station and river system/year to allow for series and spatial correlation. Luckily, the command feols() can be instructed easily to cluster standard errors by one or more dimensions. Just have a look into the info box below to learn how to set the parameters correctly.

Info: Clustered Standard Errors with feols()

The most important parameters in case you want to cluster standard errors using the feols() command are

- cluster

The parameter is to be provided with a list of dimensions on the basis of which clustering is to be performed.

- se

This defines the mode according to which clustering is to take place. In our case it is twoway, and is the default parameter. Therefore we do not need to specify the setting explicitly.

- ssc

The abbreviation stands for small sample correction; hence, you define here how the small sample correction is done. We make use of this parameter since we want to emulate the ssc procedure the researchers applied.

A more detailed explanation can be found here

Task: Here, you find the regression model from *Exercise 3.2* again. Adapt the parameters cluster and se. If you do not know how, have a look at the info box above. In our case we want to cluster along station and riversystem_time.

```
#
# reg.clust = feols(cod ~ post06 + boundary + post06:boundary, cluster = c("___", "___"), se = "___", data = dat)
#


reg.clust = feols(cod ~ post06 + boundary + post06:boundary, cluster = c("station", "riversystem_time"), data = dat)
```

Still, we do not get the same results as the researchers did. The reason is that the original regressions have been performed in Stata, where - thanks to addons - several methods exist of how to calculate standard errors. However, feols() from the fixest package allows to imitate the way the small sample correction works in xtivreg2, the function used by the researchers. Let us first compare the two ways small sample correction is done by default:

COD Discharges - Regression With Clustered Standard Errors

	Conventional SE	Default Cluster Robust SE	Paper Cluster Robust SE
(Intercept)	7.234***	7.234***	7.234***
	(0.404)	(1.263)	(1.253)
Post2006	-1.712***	-1.712	-1.712
	(0.478)	(1.248)	(1.239)
Boundary	4.359***	4.359**	4.359**
	(0.796)	(1.898)	(1.883)
Post2006 x Boundary	-2.060**	-2.060	-2.060
	(0.945)	(1.310)	(1.300)
Num.Obs.	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**			

Parameter/Function	**feols**	**xtivreg2**
small sample correction of the form $(n-1)/(n-K)$	TRUE	FALSE
$G/(G-1)$ correction is performed with G the number of cluster values	TRUE	FALSE
degrees of freedom of the Student t distribution	minimum size of the clusters with which the variance-covariance matrix has been clustered	number of observations minus the number of estimated variables

While the default method to obtain the coefficients in feols() is based on Berge (2018), *xtivreg2*'s procedure relies on the cluster-robust covariance matrix for the fixed-effects model introduced by Arellano (1987). However, *xtivreg2* is not an official Stata command, but an addon created by Mark E Schaffer. According to the author, the method the official *Stata* command applies is more conservative.

```
reg.paper = feols(cod ~ post06 + boundary + post06:boundary, cluster = c("station", "riversystem_time"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

After we conducted the regression with clustered standard errors, let us compare the results:

Task: Again, press the button to compare the regression results.

```
modelsummary(list("Conventional SE" = reg, "Default Cluster Robust SE" = reg.clust, "Paper Cluster Robust SE" = reg.paper),
          coef_rename = c("post06" = "Post2006", "boundary" = "Boundary", "post06:boundary" = "Post2006 x Boundary"),
          stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
          gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE|F|RMSE",
          title = "COD Discharges - Regression With Clustered Standard Errors") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

The first column contains the conventional standard errors, while both the second and the third column show standard errors that are cluster-robust. The column in the middle contains the results of the standard feols() procedure and the last one's standard errors are the same as in the paper, imitating the xtivreg2()'s method.

Quiz: Compare the regression results. What has changed?

- The estimators of the regression coefficients. [ ]
- The standard errors of the estimators of the regression coefficients. [x]
- Nothing. [ ]

While all of the estimated regression coefficients stayed exactly the same, their standard errors have increased massively when calculating in a cluster-robust way. The post06 dummy, formerly significant on the 0.1 % level, is not significant anymore. Same applies for the interaction term between the post06 and the boundary dummy. Hence, when interpreting regression results from clustered data, it is crucial to calculate cluster robust standard errors. Otherwise, one will probably far overestimate the significance of the estimated coefficients Due to the - in the small sample bias correction context - slightly less conservative procedure of xtivreg2(), the standard errors are jointly a little bit smaller in the last column.

Summary

As we saw, it is essential to calculate cluster-robust standard errors when performing regressions on a clustered data set. Otherwise, we will overestimate the significance of estimated parameters by a large extent. However, the procedure how standard errors can be adapted in additional ways, for example by adjusting for small samples. That is way, the default cluster robust standard errors calculated in feols() differ slightly from the paper's results. For all analysis done from now, we will only consider the standard errors clustered by monitoring station and river/year, imitating the procedure of the addon used by the authors, xtivreg2().

That is not the only obstacle to overcome in this study. One of the main assumptions of the DiD approach is the *Parallel Trends* assumption. We will learn about it and how to heal model deficiencies by introducing fixed effects and control variables in the next sub-exercise.

Exercise 3.4 – Fixed Effects and Control Variables

In the previous chapter, we introduced a regression examining the link between changes in China's political promotion incentives and relative changes in water pollution at borders. We focus on chemical oxygen demand since provincial officials are evaluated using this pollution indicator. Usually, it is not sufficient just to include the variables of interest but rather it is necessary to control for fixed effects and other variables that capture variance that is not explained yet by the model. Actually, the DiD approach is designed to avoid the need of control variables and fixed effects. However, the premise under which this approach works are parallel trends during the time before the experiment started. Let us examine this assumption in this chapter.

First and foremost, let us read the data set from Regression_Data_dta and add the post06 dummy.

Task: Just check the chunk below to load the data set.

```
dat = readRDS("dat.RDS")
```

Assumption: Parallel trend

The parallel trend assumption is the core assumption of the DiD approach. We assume that if no experiment took place, the difference between the treatment and the control group would not have changed. The course of the value relevant for us, the COD averages, would therefore have taken a parallel path for both groups. We can check this assumption graphically by looking at the course of the curve during the pre-experimental period.

Let us check this here. In our example, the parallel trend assumption means, that the gap between the measurements at border and non-border stations would not have changed, if China had not experienced the policy change in 2006. We check this by examining the trends of the years 2004 and 2005, exemplary for COD. However, this pre-experimental period is relatively short, since this means we do not know anything about the courses prior to these years.

The average pollution values per year and location(boundary/non-boundary) for all indicators from *Exercise 2.3* are stored in dat.mean.RDS.

Before we start, load the data set dat.mean.RDS with help of readRDS().

Task: Use readRDS() to load the average pollution values per year and location from dat.mean.RDS and assign it to a variable called dat.mean.

```
dat.mean = readRDS("dat.mean.RDS")
```

The data set contains measurements for all pollution indicators. Since we want to examine the COD values only, we need to filter dat.mean.

Task: Replace ___ to filter dat.mean by cod using the column substance. Save only rows containing information about cod in a variable called dat.cod.

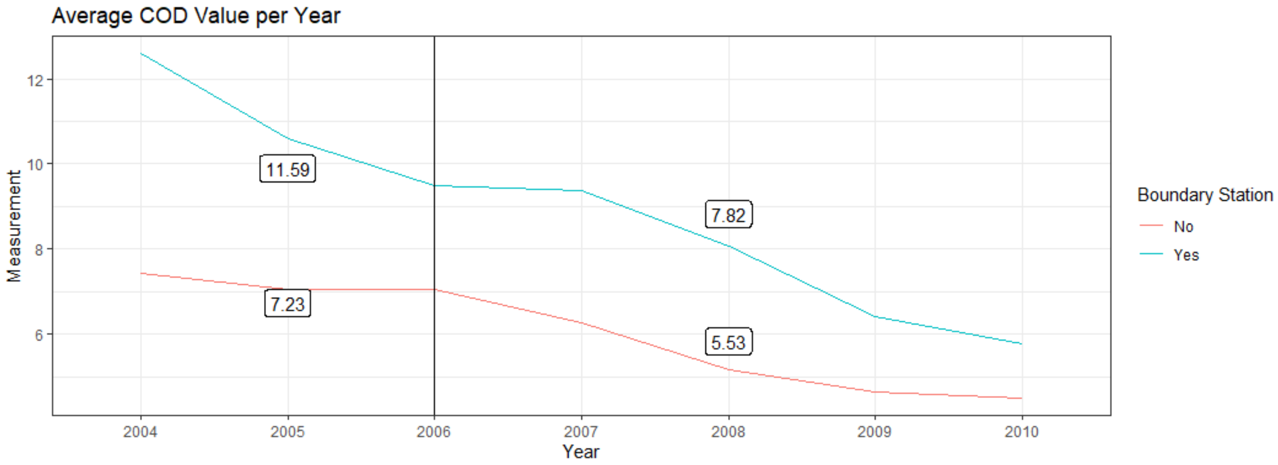
```
# ___ = ___ %>%
#   filter(___ == ___) %>%
#   select(-substance)

dat.cod = dat.mean %>%
  filter(substance == "cod") %>%
  select(-substance)
```

What we have got now is a data set containing the average COD pollution values per year separated for border and non-border monitoring stations.

Task: Replace ___ in geom_line() with the correct dummy variable in dat.cod. Remember, we want to compare the average values measured at boundary and non-boundary stations.

```
# #annotations
# y.pre.tr = filter(dat.cod, year < 2006, boundary == 1)$measurement %>%
#   mean() %>%
#   round(2)
# y.exp.tr = filter(dat.cod, year > 2005, boundary == 1)$measurement %>%
#   mean() %>%
#   round(2)
# y.pre.co = filter(dat.cod, year < 2006, boundary == 0)$measurement %>%
#   mean() %>%
#   round(2)
# y.exp.co = filter(dat.cod, year > 2005, boundary == 0)$measurement %>%
#   mean() %>%
#   round(2)
# #plot
# ggplot() +
#   geom_line(mapping = aes(x = as.factor(year), y = measurement, color = ___, group = boundary), dat = dat.cod) +
#   geom_vline(xintercept = as.factor(2006)) +
#   xlab("Year") +
#   ylab("Measurement") +
#   scale_color_discrete(labels = c("No", "Yes"), name = "Boundary Station") +
#   ggtitle("Average COD Value per Year") +
#   theme_bw() +
#   annotate("label", x = as.factor(2005), y = y.pre.tr - 1.7, label = y.pre.tr) +
#   annotate("label", x = as.factor(2008), y = y.exp.tr + 1, label = y.exp.tr) +
#   annotate("label", x = as.factor(2005), y = y.pre.co - 0.5, label = y.pre.co) +
#   annotate("label", x = as.factor(2008), y = y.exp.co + 0.3, label = y.exp.co)
#
```

```
#annotations
y.pre.tr = filter(dat.cod, year < 2006, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.exp.tr = filter(dat.cod, year > 2005, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.pre.co = filter(dat.cod, year < 2006, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
y.exp.co = filter(dat.cod, year > 2005, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
#plot
ggplot() +
  geom_line(mapping = aes(x = as.factor(year), y = measurement, color = boundary, group = boundary), dat =
dat.cod) +
  geom_vline(xintercept = as.factor(2006)) +
  xlab("Year") +
  ylab("Measurement") +
  scale_color_discrete(labels = c("No", "Yes"), name = "Boundary Station") +
  ggtitle("Average COD Value per Year") +
  theme_bw() +
  annotate("label", x = as.factor(2005), y = y.pre.tr - 1.7, label = y.pre.tr) +
  annotate("label", x = as.factor(2008), y = y.exp.tr + 1, label = y.exp.tr) +
  annotate("label", x = as.factor(2005), y = y.pre.co - 0.5, label = y.pre.co) +
  annotate("label", x = as.factor(2008), y = y.exp.co + 0.3, label = y.exp.co)
```



Have a look at the graph above. The red line represents stations in the interior of the province, while the blue line stands for monitoring stations at the border. The vertical black line in 2006 symbolizes the start of the treatment phase. Furthermore, I added the average values per group and period in four black-framed boxes that are attached to the respective lines.

Quiz: Based on the graph, do you think the parallel trends assumption is fulfilled?

- Yes, it is definitely fulfilled. [ ]
- No, it is rather not fulfilled. [x]

We cannot really state that the pre-trends are parallel. Hence, let us add another robustness check to the analysis. In the next task, we logarithmize the COD measurements using the log function, before calculating the averages and comparing the pre-trends again.

It is possible that a measurement of COD is zero, which logarithmized value is undefined. Consequently, such an operation in R would return an -Inf entry, the value it converges to, that would complicate further operations on the respective variable, vector or data set. Hence, we remove rows from dat containing an -Inf in cod.log with help of the filter() function.

In the following, we add a column called cod.log that contains the logarithmized COD measurements from the column cod to dat. The filter() operation removes rows that contain an -Inf entry in cod.log. This could occur if a measured value of 0 is logarithmized.

Task: Just check the chunk to perform both operations at once.

```
dat = dat %>%
  mutate(cod.log = log(cod)) %>%
  filter(cod.log != "-Inf")
```

Task: Just check the chunk in order to calculate the COD average by year and location, but this time based on logarithmized measurements.

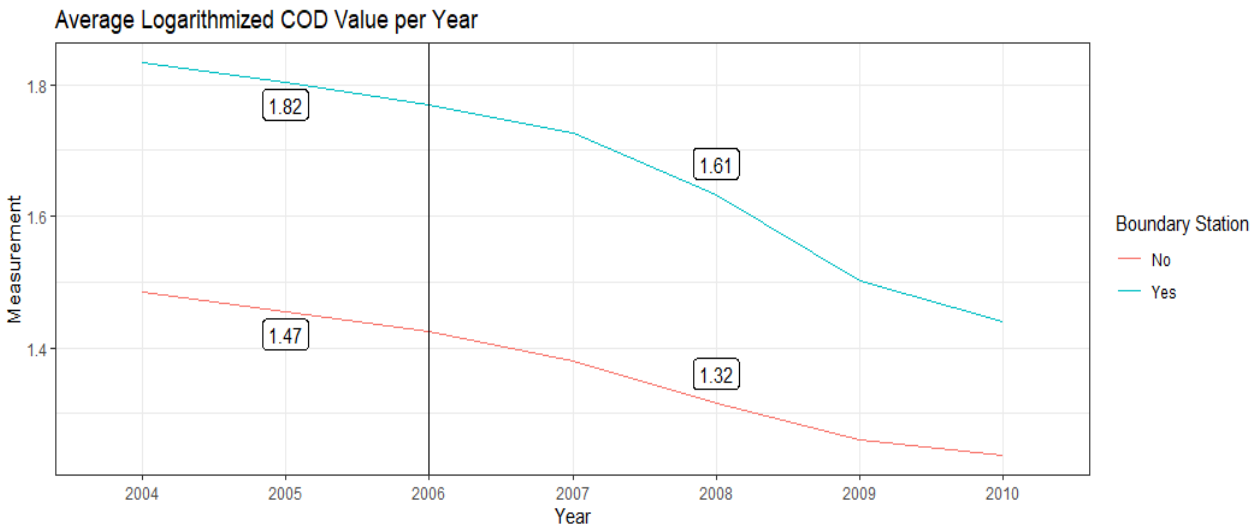
```
dat.log = dat %>%
  select(cod.log, boundary, post06, year) %>%
  group_by(boundary, year) %>%
  summarise(measurement = mean(cod.log)) %>%
  ungroup() %>%
  mutate(boundary = as.factor(boundary))
```

Let us compare the results:

Task: Just check the chunk to display both the plot from the previous task as well as the plot using logarithmized values.

```
#annotations
y.pre.tr = filter(dat.log, year < 2006, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.exp.tr = filter(dat.log, year > 2005, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.pre.co = filter(dat.log, year < 2006, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
y.exp.co = filter(dat.log, year > 2005, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
```

```
#plot
ggplot() +
  geom_line(mapping = aes(x = as.factor(year), y = measurement, color = boundary, group = boundary), dat =
dat.log) +
  geom_vline(xintercept = as.factor(2006)) +
  xlab("Year") +
  ylab("Measurement") +
  scale_color_discrete(labels = c("No", "Yes"), name = "Boundary Station") +
  ggtitle("Average Logarithmized COD Value per Year") +
  theme_bw() +
  annotate("label", x = as.factor(2005), y = y.pre.tr - 0.05, label = y.pre.tr) +
  annotate("label", x = as.factor(2008), y = y.exp.tr + 0.07, label = y.exp.tr) +
  annotate("label", x = as.factor(2005), y = y.pre.co - 0.05, label = y.pre.co) +
  annotate("label", x = as.factor(2008), y = y.exp.co + 0.04, label = y.exp.co)
```



Again, the measurements from boundary stations are blue, those in the interior of the provinces are red. The vertical black line stands for the year when the altered promotion incentives became effective and the black framed boxes display the average values for the respective group and period.

After logarithmizing the measurements and plotting the average values again, the pre-experimental trends look almost perfectly parallel although a longer time span would have been desirable. In the experimental period, we can indeed detect that the average pollution level at borders is decreasing by larger amount than in the interior.

We are actually interested whether the change in promotion incentives was able to reduce COD values at borders by a larger amount than in the interior of the province. But what if the pre-experimental trends are parallel for 2004 and 2005 but not for the years before? Therefore, the trends we observe in the plot could be just a statistical fluke.

Imagine, if the change in promotion incentives in 2005 was already known before, officials of the provinces could have carried out anticipative actions until 2006, to be well prepared. Ignoring that possibility may lead us on a wrong track, which is why we do not leave it at the graphical analysis.

Then, another problem might be unobserved background changes. For example, some provinces with only a few monitoring stations at borders might have experienced a stronger economic development than others and were therefore polluting more extensively. This would lead to a closing gap between non-border and border-stations while everything else held equal. Adding control variables might help to explain observed deviations from the parallel trends and therefore heal possible weak points - at least to some extent. The authors controlled for instance for differences in the economic development or climatic conditions.

However, not only the pollution at boundary stations decreased a lot, but also in the interior of the provinces. The five-year-plans must have not only set incentives to prevent excessive border pollution, but also to increase overall water quality. Therefore governors must have handled a trade-off between border- and non-border pollution reduction.

We cannot be sure that the pollution pre-trends of the control and treatment group have been parallel all the time. Hence, we cannot trust that all background changes have been captured by the model yet. Through fixed effects and control variables the authors back against possible deficiencies. The fixed effects, the authors make use of, are year, station and river while they additionally account for external factors that may have an influence on the water pollution indicators. This could be, for example, the temperature or varying degrees of settlement concerning the altitude of the measuring points over the observation period. Therefore, the risk of an **omitted variable bias** is reduced. It occurs when a variable is correlated with the dependent variable but excluded from the regression (Greene, 2008, p.868). Hence, this procedure is crucial because we do not want to test for a mere temporal relationship, but for a causal effect.

Fixed Effects

We continue by adding fixed effects to the regression. Thus, effects that are fixed for a group of certain units as observations made in a specific year are factored out. Hence, they enable us for instance to control for factors we do not have a concrete data source for and capture their effects on the dependent variable.

Before we continue, we verify one prerequisite of using fixed effects: We check whether the panel data is balanced. The data set is not balanced if data availability correlates with dynamics in water pollution. This is avoided if for roughly all stations and in every year measurements exist.

In order to visualize the number of stations by how many observations exist, let us create a histogram. If you want to learn more about how histograms are created in *R* with help of geom_histogram() from the ggplot2 package, have a look at the info box below.

Info: Creating Histograms for visualizing the distribution of a single continuous variable with help of geom_histogram() from the ggplot2 package

Assume you have one variable that is continuous. Often it is ideal to visualize that kind of data with help of histograms. The idea is to create bins of equal size into which the individual manifestations of the variables are sorted. Finally the number of observations per bin is counted and displayed in a bar chart.

The package ggplot we have already introduced before, offers the function geom_histogram() to create such a plot.

Note how to define the bins:

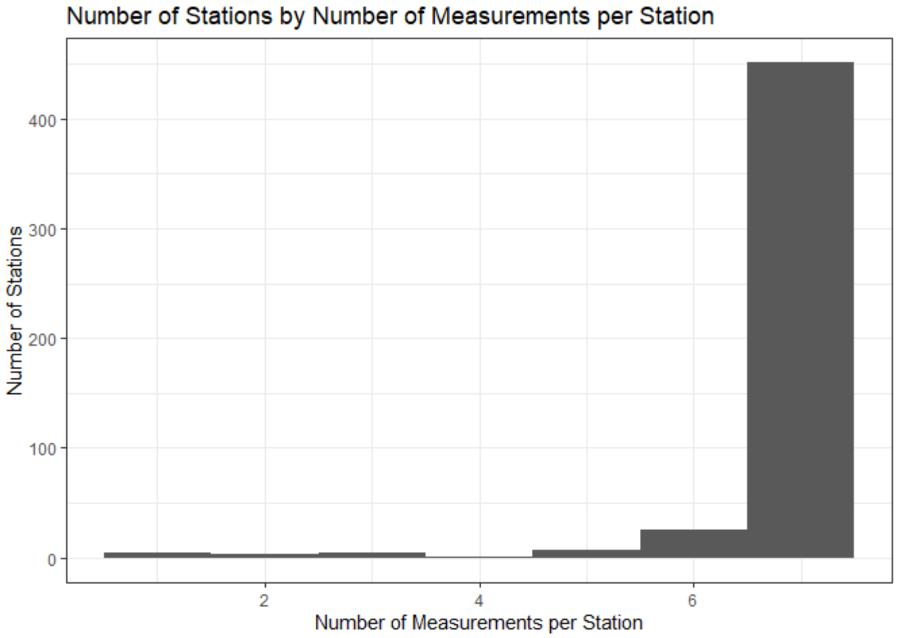
- bins: the number of bins you wish to sort the individual manifestations of the variable in
- binwidth: overrides the bins parameter and set the width of each bin
- breaks: overrides bins and breaks and is supplied with a vector containing all bin boundaries

If you want to learn more about how to create histograms with help of geom_histogram(), please have a look here.

In the following histogram we wish to count the number of observations per station.

Task: Replace the first ___ with the correct grouping variable and the second ___ with the right summarizing function. Then check the chunk in order to display the number of measurements per station.

```
# #fill in the gaps to count the number of observations per station
# dat %>%
#   group_by(___) %>%
#   summarise("no" = ___) %>%
#   ggplot(mapping = aes(x = no)) +
#   geom_histogram(binwidth = 1) +
#   xlab("Number of Measurements per Station") +
#   ylab("Number of Stations") +
#   ggtitle("Number of Stations by Number of Measurements per Station") +
#   theme_bw()

#show histogram
dat %>%
  group_by(station) %>%
  summarise("no" = n()) %>%
  ggplot(mapping = aes(x = no)) +
  geom_histogram(binwidth = 1) +
  xlab("Number of Measurements per Station") +
  ylab("Number of Stations") +
  ggtitle("Number of Stations by Number of Measurements per Station") +
  theme_bw()
```



A measurement series for a station is complete if the number of observations is seven. That is, for every year a measurement exists for the station. We can easily see in the histogram that for almost all stations the observation series is complete. Therefore the risk that the panel data is imbalanced at a large extent is low.

In order to account for possible nation-wide trends and shocks, the authors included **year** fixed effects. Thereby they control for an unobserved effect that is constant across entities but varies over years. In other words, they control for factors changing each year that are common to all monitoring stations, both at the border and in the interior of a province for a given year. For instance, this could be a more stringent environmental policy at the national level over the years that affects all factories regardless of their location.

On the other hand, the researchers added **station** fixed effects. Similarly, they control for an observed effect that is constant across years but varies over entities. This means, they account for factors that do not change over years but are different from station to station. To give an example, we could think about specific locational factors like altitude that affect only the corresponding station. but virtually do not vary over time.

After including the fixed effects for year and station., the regression formula looks as follows:

$$pol.level \quad = ß_1 post06_t + ß_2 boundary_i + ß_3 boundary_i \times post06_t + Y_t + S_i + u$$

In this formula, Y_t represents a vector of $t-1$ year dummies and S_t stands for a vector of $i-1$ station dummies, both with a corresponding vector of estimated coefficients. Recall what we would expect about $ß_3$ and answer the following quiz:

Quiz: In case the policy change was successful with regard to the previously discussed objective, ß3 will be …

- zero. [ ]
- positive. [ ]
- negative. [x]

Instead of putting the plain R base function lm to use, we want to make use of the feols function from the fixest package. This allows us to define fixed effects specifically and later to adapt the standard errors to the panel form of our data. We will deal with this in more detail later in the problem set. If you want to dive deeper into the fixest package as well as the feols function, have a look at the info box below, please.

There are some advantages of defining fixed effects specifically. First, we do not have to convert the fixed effect variable as a factor. Furthermore, if we call summary() on the estimated regression the myriad coefficients of the corresponding categories - which are usually not of interest - are not shown to improve comprehensibility.

Info: Advanced Regressions with help of feols() from the fixest package

While the standard lm() from the *R* base package can be sufficient for many purposes, sometimes we want to adapt our regressions to fixed effects and cluster in the data.

The function feols() serves our need for advanced regressions.

So how can one add fixed effects to our regression formula? Let us have a small example. Assume we have the dependent variable y as well as two independent variables x and z and the fixed effects a and b. Additionally, x and z interact with each other.

The feols function would look as follows:

```
#load the library
library(fixest)

#option a)
#set up the regression while not including the variables x and z implicitly
feols(y = x + z + x:z | a + b)

#option b)
#set up the regression while including the variables x and z implicitly
feols(y ~ x*z | a + b)
```

Task: First of all, please load the package fixest that enables us to use the feols() function.

```
library(fixest)
```

When you add a categorical or factor variable to a regression in feols, it will automatically add all required dummy variables. So there is no need to create a dummy column for each manifestation the variable could take. To be precise, exactly one less dummy is created than the categorical variable can take on different values.

Let us explain the reason with an example using the categorical variable year. If we were to create a dummy variable for each year for our regression, one variable would correspond to an exact linear combination of other independent variables, namely when the year dummies from 2005 until 2010 would be jointly equal to zero. Thus, we omit one manifestation of the year dummy, for example 2004, which then serves as our reference level. The same applies to station fixed effects, but in total only one dummy variable has to be omitted, be it a station or year variable. This is all done automatically by the function feols().

In case you are wondering why the intercept $ß_0$ disappeared from the regression formula: This is a consequence of including fixed effects. The intercept is now part of the fixed effects, if we wanted to keep it, we would have to omit a total of two dummy variables to prevent multicollinearity (Kennedy, 2008, 192ff).

Additionally, when including year and station fixed effects, we refrain from including the post06 dummy as well as the boundary dummy from the regression formula. Otherwise we would cause another case of perfect multicollinearity. While the boundary dummy is part of the information the station fixed effects contain, the post06 dummy is already implied by the year fixed effects.

An example will clarify this phenomenon: If one of the year dummies indicating the years 2006 - 2010 is 1, we can directly state that the post06 dummy will be 1, too. On the other hand, if those dummies are jointly zero, post06 will be 0, too. Hence, post06 will not provide any additional explanatory power to the model. Same counts for the boundary dummy.

As a consequence, we do not include the interaction effect by applying post06*boundary, but by the expression post06:boundary that does not add the variables themselves implicitly to the formula.

Control variables

As announced before, we want to include further variables that help us overcoming possible deficiencies regarding the parallel trends assumption and in that sense preventing us from leaving unexplained variance not taken into account yet through our model.

Let us add the following controls:

- *temperature*: Water temperature can affect biological activity and chemical conditions
- *gdpg*: city-level gross domestic product growth rate to control for possible economic growth-environmental degradation relationship
- *gdpp* city-level gross domestic product per capita to control for possible pollution-income relationship
- *lightbuffer5km*: Luminosity data from satellite images obtained at night to identify urban settlements around

Task: Add the control variables from above to the regression formula below by replacing ___ and click on check. You do not have to modify anything else.

```
# reg.fe.ctrl = feols(cod.log ~ post06:boundary + ___ + ___ + ___ + ___| station + year, cluster =
c("station", "riversystem"), ssc = ssc(adj=FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)

reg.fe.ctrl = feols(cod.log ~ post06:boundary + temperature + gdpg + gdpp + lightbuffer5km | station + year,
cluster = c("station", "riversystem"), ssc = ssc(adj=FALSE, cluster.adj = FALSE, t.df = "conventional"), data
= dat)
```

Verify Parallel Trend Assumption with Fixed Effects and Control Variables

We want to check whether the parallel trends assumption still holds after including control variables and fixed effects. However, that is not as simple as in the previous tasks without any controls. Luckily, Sebastian Kranz describes a convenient way on his *Economics and R Blog* how to virtually assess the parallel trends assumption when including control variables and fixed effects. Although he created a package called ParallelTrendsPlot that contains the functions needed to apply his approach, we will perform the steps by hand.

The approach comprises the following steps:

1. Estimate the DiD Regression including all control variables. Note that fixed effects variables have to be added as control variables, too.
2. Then, we set all control and fixed effect variables in the data set constant, besides the experimental period and the treatment group indicator. That is, all continuous variables may be set to zero for instance, while discrete variables can be set equal to any value from the respective set.
3. Predict outcomes for treatment and control groups using the constant control and fixed effects variables from 2) and add the residuals of the original regression.
4. Plot the predicted outcomes.

In the following, most of the code is already given. Nonetheless, I still want to you to fill out some parameters to complete the statements.

Step 1: We already added the logarithmized COD measurements in the column cod.log of dat. Let us adapt the regression above according to 1. We need to include station and year as controls instead of fixed effects. Maybe you wonder why we need to add the post06 and boundary dummy variable although it does not add any information that station and year could not imply. Remember the second step: Following the instructions, we set year as well as station equal to zero. This way we lose the indicator information about the experimental period and the treatment group. To avoid this, we add, in contrast what we have discussed before, the post06 and boundary dummy variable to the regression.

Task: Perform the regression from above, but this time adding the fixed effects variables as control variables, as well as bringing back the dummy variables post06 and boundary. However, cod.log is still the dependent variable. In order not to change the original data set, dat is saved into dat.pt. Replace ___ to complete the code.

```
# #save dat into dat.pt
# ___ = ___
# #save the regression result into reg.pt
# ___ = feols(___ ~ post06 + boundary + post06:boundary + temperature + gdpg + gdpp + lightbuffer5km
+ station + as.factor(year), data = dat.pt)

dat.pt = dat
reg.pt = feols(cod.log ~ post06 + boundary + post06:boundary + temperature + gdpg + gdpp +
lightbuffer5km + station + as.factor(year), data = dat.pt)
```

Step 2: Let us create a new data frame where all control variables are set constant. The columns cod, post06 and boundary remain unchanged.

Task: Set all controls constant. Replace ___ with zero to complete the code.

```
# #fill in the gaps to set the variables to zero
# dat.0 = mutate(dat.pt, temperature = ___, gdpg = ___, gdpp = ___, lightbuffer5km = ___, station =
"H001", year = "2004")

dat.0 = mutate(dat.pt, temperature = 0, gdpg = 0, gdpp = 0, lightbuffer5km = 0, station = "H001", year =
"2004")
```

Step 3: Replace the column cod.log in dat.pt with the corresponding predictions. Furthermore, add the residuals of reg.

Task: Replace ___ according to the instructions above to complete the code. Do not change anything else.

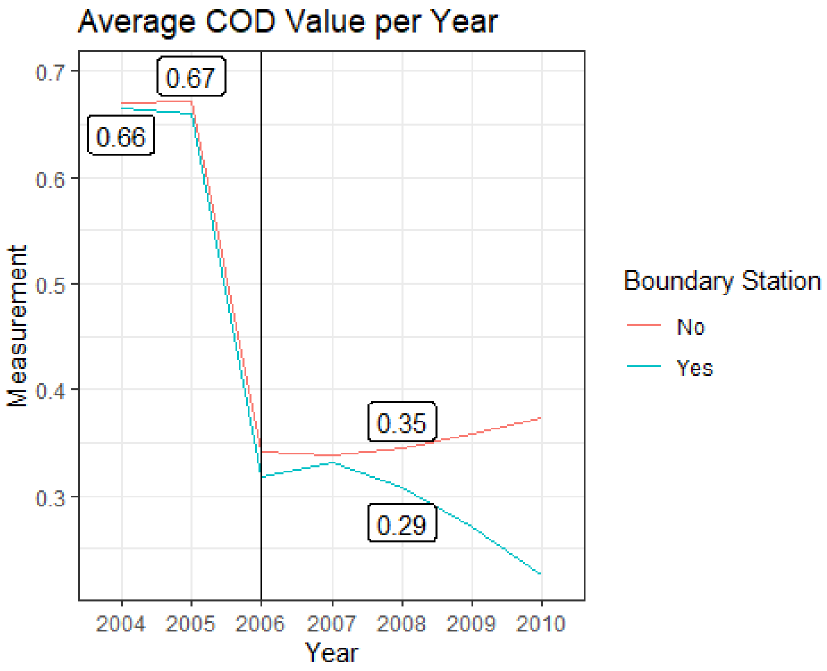
```
# #fill in the gaps by following the instructions
# ___ = predict(reg.pt, dat.0) + resid(___)
#

dat.pt$cod.log = predict(reg.pt, dat.0) + resid(reg.pt)
```

Step 4: In the next chunk I have combined two steps. First, the preparing the data set for the plot as we already did two times before, Second, creating and showing a plot to virtually assessing the parallel trends assumption.

Task: Just check to prepare the data set and show the plot.

```
#prepare
dat.pt.mean = dat.pt %>%
  select(cod.log, boundary, post06, year) %>%
  group_by(boundary, year) %>%
  summarise(measurement = mean(cod.log)) %>%
  ungroup()
```

```r
#annotations
y.pre.tr = filter(dat.pt.mean, year < 2006, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.exp.tr = filter(dat.pt.mean, year > 2005, boundary == 1)$measurement %>%
  mean() %>%
  round(2)
y.pre.co = filter(dat.pt.mean, year < 2006, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
y.exp.co = filter(dat.pt.mean, year > 2005, boundary == 0)$measurement %>%
  mean() %>%
  round(2)
#plot
ggplot() +
  geom_line(mapping = aes(x = as.factor(year), y = measurement, color = as.factor(boundary), group =
as.factor(boundary)), data = dat.pt.mean) +
  geom_vline(xintercept = as.factor(2006)) +
  xlab("Year") +
  ylab("Measurement") +
  scale_color_discrete(labels = c("No", "Yes"), name = "Boundary Station") +
  ggtitle("Average COD Value per Year") +
  theme_bw() +
  annotate("label", x = "2004", y = y.pre.tr - 0.02, label = y.pre.tr) +
  annotate("label", x = "2008", y = y.exp.tr - 0.015, label = y.exp.tr) +
  annotate("label", x = "2005", y = y.pre.co + 0.025, label = y.pre.co) +
  annotate("label", x = "2008", y = y.exp.co + 0.02, label = y.exp.co)
```



The red line stands for monitoring stations at the boundary, while the blue line represents stations in the interior. The vertical black line symbolizes the time when the new environmental policy came into effect. The black-framed boxes report the average values for the respective group over the period they are in. Controlling for several variables, both lines are almost overlapping and quite parallel in the pre-experimental period. However, it is still subjective: between 2004 and 2005 the lines do not run perfectly in the same angle. In the experimental period, the lines diverge over time. While the average value for non-boundary stations is increasing, the value for boundary stations is decreasing. From the 2005 perspective this means the following: Everything else held equal, the logarithmized COD value is increasing over the years in the interior, but decreasing at borders. This is consistent with the insights we gained before.

After verifying the parallel trends assumption, let us display both the results of the regression model from *Exercise 3.2* and the model with fixed effects and control variables in the next task.

Task: Just check the following chunk to compare the two previous regression results with the results from the regression extended by the above mentioned controls.

```
#run regression
reg = feols(cod ~ post06*boundary, cluster = c("station", "riversystem"), ssc = ssc(adj=FALSE, cluster.adj =
FALSE, t.df = "conventional"), data = dat)
#compare results
modelsummary(list("Conventional Regression" = reg, "FE Regression with Controls" = reg.fe.ctrl),
        coef_rename = c("post06" = "Post2006", "boundary" = "Boundary", "post06:boundary"  = "Post2006
x Boundary"),
        coef_omit = "year|temp|light|gd",
        stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
        gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE|F",
        title = "COD Discharges and Its Determinants Without And With Station and Year Fixed Effects And
Controls") %>%
        kable_classic() %>%
        kable_styling(font_size = 15)
```

COD Discharges and Its Determinants Without And With Station and Year Fixed Effects And Controls

	Conventional Regression	**FE Regression with Controls**
(Intercept)	7.244***	
	(1.665)	
Post2006	-1.713***	
	(0.549)	
Boundary	4.395**	
	(1.976)	
Post2006 x Boundary	-2.106*	-0.052***
	(1.160)	(0.019)
Num.Obs.	3372	3372
*** p < 0.1, ** p < 0.05, *** p < 0.01**		

The Conventional Regression in the first regression does not consider fixed effects or control variables. The second column shows a regression including both fixed effects and control variables as described above. Here, the coefficient of the interaction term between the Boundary and Post2006 dummy is smaller, but shows a higher significance than in the regression without controls and fixed effects.

The estimated coefficient of -0.052 can be interpreted as follows: If no experiment had taken place - which means the government would not have changed promotion incentives regarding river border pollution - the COD concentration at borders after 2006 would be on an average -0.052 higher. Said the other way around, this is the amount the water quality improved thanks to the regime change. However, the interpretability depends on the significance of the coefficients that is on the one percent level and therefore highly significant. Interestingly, the magnitude of the coefficient has shrunk a lot. Hence, parts of the reduction must be caused by other factors, aside from the government's action.

Summary

Initially, we could not detect any parallel trends - the essential prerequisite for the application of the DiD approach - in the pre-experimental phase. These appeared only after logarithmizing the measured data. But how about the situation before 2004? Since there are no data points here, we make our analysis somewhat more robust using control variables and fixed effects. As a result, the magnitude of the estimator has shrunk considerably, but the direction of the effect is still the same and is now significant at the one percent level.

Note: Complementing the authors' analysis, I added another differentiating component with the different pollution indicators. Since the new environmental policy considers COD only, this pollution indicator must have decreased to a greater extent compared to the other substances - at least if they are not perfectly correlated. This approach is called *triple difference*, because we observe the difference (pollution indicator) in difference (time) in difference (location). You will find the analysis in the appendix exercise A5 Triple Difference by Regression.

Award: Improvement

You have successfully completed the third exercise and learnt that the water quality measured in COD indeed improved at borders after the change in promotion in incentives!

Exercise 4 – Leader Career Concerns and Pollution Dynamics

Party secretaries and governors are responsible for the provincial development, with the latter in charge of detailed government affairs. The researchers collected biographical data about both secretaries and governors to find out whether younger officials are more ambitious to meet environmental target goals due to greater career concerns. We will explore this conjecture in this exercise to establish a direct link between promotion incentives and controlling river border pollution.

Before we start, let us load the data set.

Task: Just check the chunk to read the data set from dat.RDS and save it in dat.

```
dat = readRDS("dat.RDS")
```

Since we will be looking at the ages of governors and secretaries in the upcoming regressions, we want to calculate some summary statistics on this variable for the entire study period. However, only birth years are included in the data set.

Use the following variables to calculate how old the provincial leaders were when each measurement was made:

- year, the year in which the observation was made;
- g_yob, the year of birth of the respective governor when the observation was made;
- s_yob, the year of birth of the respective secretary when the observation was made.

Task: Complete the formula by replacing ___ to calculate the age of the governors g_age and the secretaries s_age.

```
# #fill in to calculate how old the governors>/secretaries were when the observation has been made
# dat = dat %>%
#   mutate(g_age = ___ - ___) %>%
#   mutate(s_age = ___ - ___)

dat = dat %>%
  mutate(g_age = year - g_yob) %>%
  mutate(s_age = year - s_yob)
```

After calculating the age of the provincial officials, let us analyze the age distribution. We use summary() that is part of the R base package. The function calculates the minimum, the maximum, the median, the mean value as well as the first and third quartile.

Task: Just check the chunk in order to show summary statistics for both governor's and secretaries' ages.

```
#show summary of governor's ages
summary(dat$g_age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    45.0    56.0    59.0    58.2    61.0    65.0
```

```
#show summary of secretaries' ages
summary(dat$s_age)

##    Min.  1st Qu.   Median    Mean  3rd Qu.   Max.
##   47.00   57.00    60.00    59.55   63.00   68.00
```

As we see, the age of the governors over the observation period ranges from 47 to 68, while half of them reached the age of 59 already. The secretaries exhibit a very similar age structure. The relatively large age span of around twenty years will be helpful later in our regressions to account for the heterogeneity of career concerns. We will assume the latter one correlating with the age of the corresponding person in charge.

The majority of the governors and secretaries are between 55 and 65 years old. What we learn from this is that most party secretaries and governors are promoted to their posts at the age of 55 and usually relinquish their posts at the age of 65 at the latest. So another promotion should take place in that short time frame, ideally before the official reaches the age of 60.

The government adjusted the promotion conditions for political leaders at the provincial level, the governors and party secretaries, intending to make them pay more attention to water pollution in downstream areas near the border.

Furthermore, two legislative changes are of interest for our analysis. First, the Decision to Build a Retiring Scheme for Senior Cadres (1982) precluded the reappointment of state officials over the age of 65, and second, the Temporary Regulations on Terms of Cadre of Communist Party and Government (2006) established a maximum term of five years for governors and party secretaries.

These norms imply the following with regard to our analysis: Through improving the chances for relatively younger leaders to achieve advanced leadership positions, we hypothesize they will be more ambitious in order to fulfill environmental goals, such as the COD target levels at borders, as a consequence of their longer career horizons.

To test the hypothesis, we apply the following regression:

$$\begin{aligned} pol.level \quad &= ß_1(Boundary_i \times TimeTrend_t) + ß_2(Age_i \times Boundary_i \times TimeTrend_t) \\ &\quad + ß_3 Age_i + Y_t + S_i + C \quad + \varepsilon \end{aligned}$$

The researchers decided to deploy the boundary dummy $Boundary_i$ and the time trend $TimeTrend_t$. Note that we add an interaction term between the age of the secretaries, the boundary dummy, the time trend as well as another age term. The variable Age_i stands hereby for the age of the upstream provincial leader. Y_t and S_t are year and station fixed effects with their corresponding regression coefficients. C represents the list of control variables as well as their coefficients we already introduced in Exercise 3.4.

What is new here is the time trend variable. It is defined as follows:

Time Trend

The variable we use to represent the time trend is tpost. It is a number indicating how many years have passed since the policy change came into effect. This setting allows a more precise localization of the relative water quality improvement with regard to the year it has taken place. Have a look at the table that follows by clicking on check to find out what it means:

Year	tpost
2004	0
2005	0
2006	1
2007	2
2008	3
2009	4
2010	5

Have a look at the regression formula again and try to answer the next quiz:

Quiz: Assume older officials are less ambitious in reducing COD levels at borders according to the new environmental policy. What does this imply for the coefficient ß2?

- It takes a negative value. [ ]
- It takes a positive value. [x]
- Nothing. [ ]
- It is equal to zero. [ ]

The coefficient ß$_2$ belongs to the interaction term between age of the person in charge and the dummy variable indicating whether the monitoring station is next to the provincial border as well as the time trend. Hence, if the coefficient takes a positive value, older leaders are less ambitious in reducing the COD level according to the new environmental policy. This causes a higher pollution at their provincial borders compared to the pollution level at borders of provinces with younger officials.

In total, we will run three regressions, one for the secretaries, one for the governors, and then for both groups together.

Note: In the following, the researchers clustered by station and riversystem. In contrast, they clustered by station and riversystem_time in the regressions before. The authors did not further comment their reasons. Hence, we will just follow their ideas in order to reproduce the results from the paper.

Task: Just check the following code chunk in order to estimate the three regressions mentioned.

```
#secretaries
reg.sec = feols(cod ~ tpost:boundary + s_age:boundary:tpost + s_age + gdpg + gdpp + temperature +
lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj =
FALSE, t.df = "conventional"), data = dat)
#governors
reg.gov = feols(cod ~ tpost:boundary + g_age:boundary:tpost + g_age + gdpg + gdpp + temperature +
lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj =
FALSE, t.df = "conventional"), data = dat)
#secretaries and governors
reg.both = feols(cod ~ tpost:boundary + g_age:boundary:tpost + g_age + s_age:boundary:tpost + s_age +
gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc =
ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

The variable reg.sec contains the regression concerning the secretaries, reg.gov the regression concerning the governors and reg.both includes variables of both, the secretaries and governors together.

Task: Just check the chunk to show the regression results.

```
modelsummary(list("Secretaries" = reg.sec, "Governors" = reg.gov, "Secretaries and Governors" = reg.both),
            coef_omit = "year|temp|light|gd",
            coef_rename = c("tpost:boundary" = "Time Trend x Boundary", "s_age" = "Secretary Age", "g_age"
= "Governor Age", "tpost:boundary:s_age" = "Time Trend x Boundary x Secretary Age",
"tpost:boundary:g_age" = "Time Trend x Boundary x Governor Age"),
            stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
            gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE",
            title = "Career Concerns and Water Pollution") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Career Concerns and Water Pollution

	Secretaries	Governors	Secretaries and Governors
Secretary Age	-0.047		-0.052
	(0.058)		(0.062)
Time Trend x Boundary	-1.273	-4.199***	-4.988**
	(1.509)	(1.212)	(2.224)
Time Trend x Boundary x Secretary Age	0.012		0.012
	(0.023)		(0.022)
Governor Age		-0.008	-0.004
		(0.067)	(0.071)
Time Trend x Boundary x Governor Age		0.063***	0.065***
		(0.017)	(0.018)
Num.Obs.	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**			

While we find no significant coefficients when observing the secretaries only, the regression concerning the governors exhibits highly significant coefficients of the interaction terms between the boundary dummy and the time trend as well as between the governor’s age, the boundary dummy and the time trend. This finding is consistent with the idea of governors being responsible for the province’s performance while secretaries being rather supervisors.

Let us think about how to interpret the significant results:

Quiz: The estimated coefficient $ß_2$ of the interaction between the time trend variable and the boundary dummy in the second regression is -4.199. How can you interpret the result?

- The overall water quality improved over time. [ ]
- The overall water quality deteriorated over time. [ ]
- The water quality in the interior relatively improved over time. [ ]
- The water quality at borders relatively deteriorated over time. [ ]
- The water quality at borders relatively improved over time. [x]

Since tpost stands for the number of years since the government has changed promotion incentives, average COD measurements at the borders decreased by an average of $4{,}199$ per year since the enactment of the new environmental legislation, holding the governor's age equal. This estimator is consistent with the results from the last exercise, where we saw that the water quality at borders improved after 2006.

Quiz: The estimated coefficient $ß_3$ of the interaction between the time trend variable, the boundary dummy and the governor's age in the second regression is 0.063. How can you interpret the result, considering the result from the previous quiz?

- The water quality improvement over time at borders is weaker when the governor is older. [x]
- The water quality improvement over time at borders is weaker when the governor is younger [ ]
- The water quality improvement over time in the interior is weaker when the governor is older. [ ]
- The overall water quality improvement over time is weaker when the governor is older. [ ]

The coefficient $ß_3$ is positive, therefore the water quality improvement is weaker when the governor is older. This can be interpreted as relatively younger governors being more interested in achieving COD goals due to potential career opportunities.

Let us take a look at the third regression, too, where we included both, the secretary's as well as the governor's terms. Again, the coefficient of the interaction term between the boundary dummy and the time trend variable is significant and negative. Therefore it can be interpreted as before. Furthermore, all terms referring to the secretaries' age are insignificant and thus shall only play a minor role with regard to its influence on the COD pollution. However, the coefficient of the interaction term between the governors' age, the boundary dummy and the time trend variable is highly significant and positive again. That is, even controlling for the secretary, the governors are in the decisive role for the extent of COD reduction at the border over time.

Summary

The central government incentivized governors through promotion perspectives to reduce river border pollution. It seems like governors are rather responsible for the performance of a province than secretaries, who serve first and foremost as supervisors. It is plausible to ascribe more ambitions to younger governors and that is consistent to the results we have got in the regressions: The younger the governor was, the more the water quality improved over time. This is an indication that the mechanism implemented by the central government has indeed contributed to the reduction of above-average pollution of rivers at borders by setting effective incentives.

But how might governors have managed to reduce pollution at borders? Let us discover one possible way in the next exercise.

Note: If you are interested in further robustness checks have a look at the appended exercise A4 Leader Career Concerns and Pollution Dynamics: COD vs. Other Pollution Indicators. We repeat the regressions from this exercise, but replacing COD by other water pollution indicators.

Award: Ambitions

You have successfully completed the fourth exercise and know now how to motivate governors to perform at their best!

Exercise 5 – The Location of Pulp and Paper Plants

In the last exercise, we learnt that governors that care about their promotion prospects were more ambitious in reducing the excessive COD pollution at borders. But how did they manage to reduce the contamination?

The above-average water pollution levels near border areas suggest that there must be a concentration of heavy contaminators. There are several ways to reduce excessive river border pollution. Shutting down pollution-incentive industries in border areas or encouraging investment in better technological equipment regarding water conditioning are just two examples. Apart from this, if the requirements regarding the pollution level of rivers near the border become relatively more stringent, this should also have an impact on the settlement of new potential contaminators. Governors could exert their influence by designating land or setting incentives for relocation accordingly, having the trade-off between border and non-border pollution goals in mind. In this exercise, we explore this question and compare settlement of new companies before and after tightening the environmental legislation in 2006.

According to the authors, there are three major sources of COD discharges, which are

- Industrial activity;
- Domestic activity and;
- Agricultural runoff.

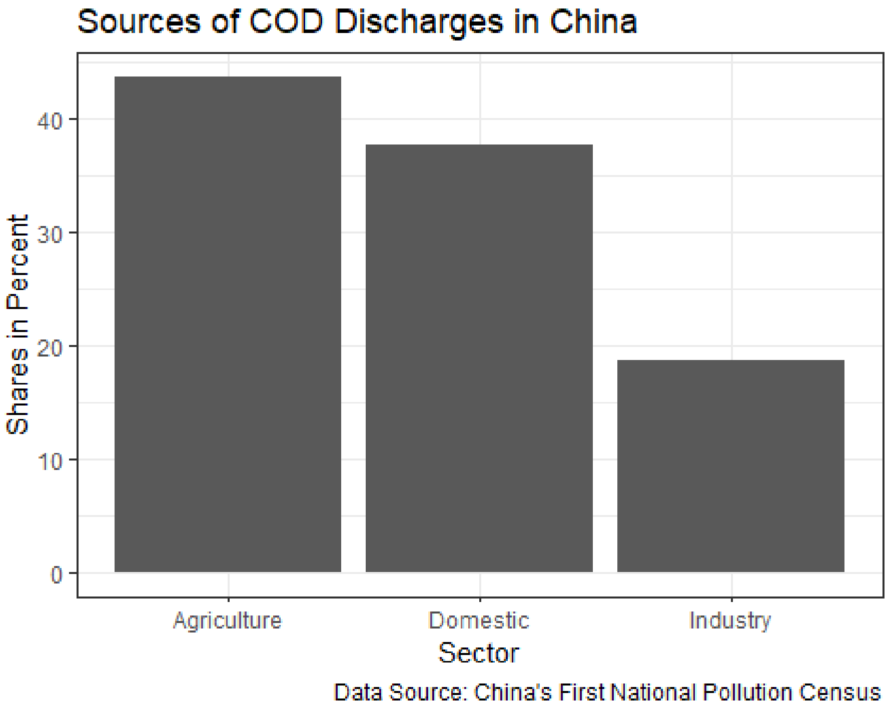
What would you guess are the percentages of each sector? Simply guess in the next quiz.

Quiz: Rank the sectors in descending order of their share to total COD emissions in China.

- Industrial activity, Domestic activity, Agricultural runoff. [ ]
- Agricultural runoff, Domestic activity, Industrial activity. [x]
- Industrial activity, Agricultural runoff, Domestic activity. [ ]
- Domestic activity, Agricultural runoff, Industrial activity. [ ]

In order to visualize the share of each sector, just check the following chunk.

```
shares = data.frame(c(43.7, 37.7, 18.6), c("Agriculture", "Domestic", "Industry"))
colnames(shares) = c("Share", "Sector")
ggplot() +
  geom_bar(mapping = aes(x = Sector, y = Share), data = shares, stat = "identity") +
  labs(caption = "Data Source: China's First National Pollution Census") +
  ylab("Shares in Percent") +
  ggtitle("Sources of COD Discharges in China") +
  theme_bw()
```



As we can see in the bar chart, the agricultural sector is the strongest driver behind the COD indicator. Households have a similarly strong influence, while industry still contributes with just under a fifth.

However, it is not only which sector is the biggest polluter that matters, but also which sector local governments have the most influence over. Measures that lead theoretically to a strong reduction, but cannot be executed are ultimately not effective. What sector do you think local governments have the greatest control over? Just guess in the quiz below.

Quiz: Which is the sector the local governments have the greatest control over?

- Agriculture [ ]
- Domestic sector [ ]
- Industry [x]

Local governments can exert the greatest influence on industry, compared to the domestic and agricultural sectors. This is done through their sole authority of land development and thus through the targeted designation of industrial zones. The spatial distribution and extent of pollution can thus be regulated by decisions on the location and closure of emission-intensive factories.

However, the reduction of COD load cannot be achieved without costs. As a consequence, the strategy should be to focus on industries that contribute relatively heavily to pollution but relatively little to economic strength. The pulp and paper industry seems to be such a candidate (Laplante, 1996; Grey, 1998).

We want to verify this with help of data provided by the researchers. The data is stored in an Excel file called Figure-3.xlsx. To read this file type we use read_excel() from the readxl package.

If you need some help how to read Excel files using the readxl package, have a look at the info box below.

Info: How to read Excel files by means of the readxl package.

In case you wish to read Excel files stored in either the xls or xlsx format, the function read_excel helps you to do so.

Let us summarize the most important parameters briefly:

- path: file path
- sheet: the name of the sheet you want to read from the file
- range: the cell range you want to read from, for example C3:F12
- col_types: either NULL if you wish the function to guess the types of the columns or you supply a character vector containing one type per column
- col_names: define how to set the names of the columns, a character vector or use the first row in a sheet

If you want to learn more about how to read Excel files with help of read_excel(), have a look at the function's documentation.

Task: Load the library readxl first.

```
#load the readxl library
library(readxl)
```

Task: Read the file Figure-3.xlsx and store it in a variable called dat.sectors. Note, that you should use the parameter range = cell_cols("A:E") in order to read columns A to E only. Afterwards, apply head() to show the first six rows of dat.sectors.

```
#read columns A to E from the file "Figure-3.xlsx" and store it in dat.sectors
#show the first six rows of dat.sectors
dat.sectors = read_excel("Figure-3.xlsx", range = cell_cols("A:E"))
head(dat.sectors)
```

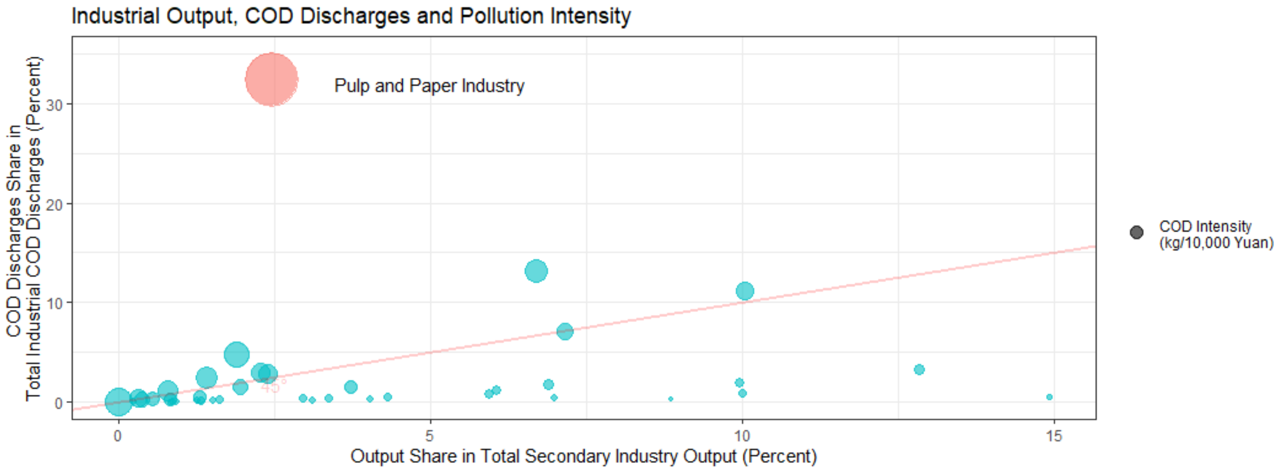
no	Sector	GDP_share	COD_share	COD_intensity
1	Mining and Washing of Coal	3.7328259	1.5567141	0.91700693
2	Extraction of Petroleum and Natural Gas	3.3781878	0.4163978	0.27551599
3	Mining and Processing of Ferrous Metal Ores	0.83215408	0.28970652	0.96578668
4	Mining and Processing of Non-Ferrous Metal Ores	0.78647144	1.0998667	3.3657449
5	Mining and Processing of Nonmetal Ores	0.55529718	0.2718565	1.2030297
6	Mining of Other Ores	0.00409277	0.01388468	6.9881003

The column GDP_share contains the percentage of the gross domestic product generated by the corresponding sector, while COD_share stands for the share of total COD for which the sector is responsible. The last column represents the ratio of COD in kg per 10000 Yuan industrial output.

Let us visualize the data using a scatter plot. You do not need to understand every single parameter set and function used to create the plot below.

Task: Just check the following code to visualize the data from dat.sectors.

```
#prepare data set
dat.sectors %>%
  mutate(point_label = ifelse(Sector=="Paper and Paper Products", "Pulp and Paper Industry", "")) %>%
  mutate(color = ifelse(Sector=="Paper and Paper Products", "green", "orange")) %>%
#plot sectors
  ggplot(aes(GDP_share, COD_share, label = point_label, color = color)) +
  geom_point(aes(size = COD_intensity), alpha = 0.6) +
  xlab("Output Share in Total Secondary Industry Output (Percent)") +
  ylab("COD Discharges Share in \nTotal Industrial COD Discharges (Percent)") +
  ggtitle("Industrial Output, COD Discharges and Pollution Intensity") +
  ylim(0,35) +
  annotate("text", x = 5, y = 32, label = 'Pulp and Paper Industry') +
  geom_abline(slope = 1, color = "red", size = 1, alpha = 0.2) +
  annotate("text", x = 2.5, y = 1.7, label = "45°", color = "red", alpha = 0.2) +
  guides(color = "none") +
  scale_size_continuous(range = c(1,15), breaks = 1, labels = "COD Intensity \n(kg/10,000 Yuan)", name =
NULL) +
  theme(legend.position = c(0.8,0.9), legend.background = element_rect(fill="grey", size=0.1,
linetype="solid")) +
  theme_bw()
```



We have plotted the share in total secondary output in percent on the horizontal axis and share of total industrial COD discharge in percent on the vertical axis. The size of the dots symbolizes the COD intensity. That is, the amount of emitted COD divided by the secondary output of the respective industry. The exceptionally large red colored outlier we can spot easily on the graph is the pulp and paper industry.

Let us find out the exact share for the pulp and paper industry for both COD discharge and secondary industry output. Since we already know that it is the industry with the highest COD discharge, we can just sort the data set accordingly.

Task: Arrange the data set dat.sectors decreasingly by COD_share with help of arrange() to learn more about the industry sector with the highest COD pollution potential. If you do not know how to arrange decreasingly, you might have a look on the function's documentation. Then, pipe the result to head() in order to show the first six rows.

```
arrange(dat.sectors, desc(COD_share)) %>%
  head()

##   no                                          Sector GDP_share COD_share
## 1 16                        Paper and Paper Products   2.46197  32.36453
## 2  7 Processing of Food from Agricultural Products   6.70629  13.17755
## 3 20  Raw Chemical Materials and Chemical Products  10.03886  11.12358
## 4 11                                         Textile   7.17616   6.97265
## 5  9                                       Beverages   1.89445   4.71415
## 6 26       Smelting and Pressing of Ferrous Metals  12.82999   3.23283
##   COD_intensity
## 1      29.10104
## 2       4.55173
## 3       2.54749
## 4       2.06341
## 5       5.43556
## 6       0.59420
```

We see that the Paper Industry is responsible for almost a third of total industrial COD discharges, while it only stands for roughly 2.5 percent of the total output of the secondary industry. This fact recommends officials to focus on this sector. In case you want to learn more about this industry, have a look at the info box below.

Info: The Pulp and Paper Industry

Paper mills are considered as the strongest emitters of COD and BOD worldwide (Laplante, 1996). Let us learn why.

The production of paper consists out of the following steps:

1. Pulp Production;
2. Pulp Processing and Chemical Recovery;
3. Pulp Bleaching;
4. Stock Preparation and;
5. Paper Manufacturing.

The steps of processing pulp are very water-intensive. When done, large amounts of waste water containing residual materials are - under the conditions of a weak water protection legislation - directly released into rivers. In case plants are located further away from rivers, where firms are forced to comply with a stricter environmental protection legislation, production costs are generally higher (Gray, 2004).

The reasons for higher costs are:

- More Equipment for Emission Prevention;
- Higher Operating Costs and;
- More Extensive Servicing.

Hence, under the assumption of profit maximization, paper mills tend to be located near rivers when environmental legislations are weak. As a consequence, negative externalities in the form of increased COD and BOD emissions, have to be borne by society, or more precisely, in our example, by provincial neighbors downstream. However, it is feasible to reduce emissions by appropriate political measures that encourage firms to invest into new technologies (Gray, 1998).

Let us now take a closer look at the geographic distribution of this sector over time, as we expect to see shifts after the new environmental policy takes effect. To do so, let us first read one more data set from another *Stata* file.

Task: Read the .dta file figure_5.dta. Use read_dta() and store the data set in a variable called dat.dis.p. Then, show the first six rows of dat.dis.p with head().

```
#read figure_5.dta and store it in dat.dis.p
#show the first six rows of dat.dis.p
dat.dis.p = read_dta("figure_5.dta")
head(dat.dis.p)

##   newfirm06_08_distance_bmonitor newfirm03_05_distance_bmonitor
## 1                          6.706                          8.151
## 2                          6.950                          8.151
## 3                          7.171                          8.491
## 4                          7.651                          8.745
## 5                          8.936                          8.831
## 6                          8.936                         10.266
```

While the first column represents the distance of new pulp and paper firms to the nearest boundary monitoring station within the province from 2006 to 2008, the second column stands for the period between 2003 and 2005.

That is, we divide the companies into two groups: * Companies that have been opened after including 2006, that is, after the new environmental policy became effective (Column 1) and; * Companies that have been opened before 2006, that is, before the new environmental policy became effective (Column 2).

What do you expect about the average distance of new pulp and paper firms to the next boundary station from 2006 compared to before? Answer the question by choosing the right answer in the quiz below.

Quiz: Assume the environmental policy change is effective from 2006 and the governors focused on the pulp and paper industry to achieve substantial COD reductions at borders. What would happen to the average distance between new companies in the pulp and paper sector and the nearest boundary station compared to prior 2006?

- It should have remained the same. [ ]
- It should have increased. [x]
- It should have decreased. [ ]

If the new environmental policy had been effective, it is likely that the average distance of new companies in the pulp and paper industries to the next boundary measurement station would have increased after including 2006, when the policy change came into force.

To check our assumption, let us display the kernel density function of both sets' distances, those of companies that started operations before and after 2006. Before we start, we need to transform the data set into the long format using pivot_longer().

Note: If you make a mistake and need to load dat.dis.np again, do not hesitate to go to the last chunk and run it again.

Task: Just check the chunk to convert the data into the long format.

```
dat.dis.p.long = dat.dis.p %>%
  pivot_longer(cols = everything(), names_to = "time", values_to = "distance")
```

The column time of distr does now indicate, whether an observation has been made after the policy change came into force or afterwards. The column distance contains the distance of an observed firm to the next boundary station.

We learnt already how to visualize one continuous variable with help of histograms. Let us now use geom_density() from the package ggplot2 in the task below. If you want to learn more about how the function works, do not hesitate to have a look at the info box.

Info: Visualize density functions with help of geom_density()

There are several ways the ggplot2 package offers to visualize one continuous variable. One of those is the function geom_density. It computes and draws the estimate of the kernel density of a variable with help of a large data set. One can also imagine it as a histogram with myriad, extremely small bins that together look like a line chart.

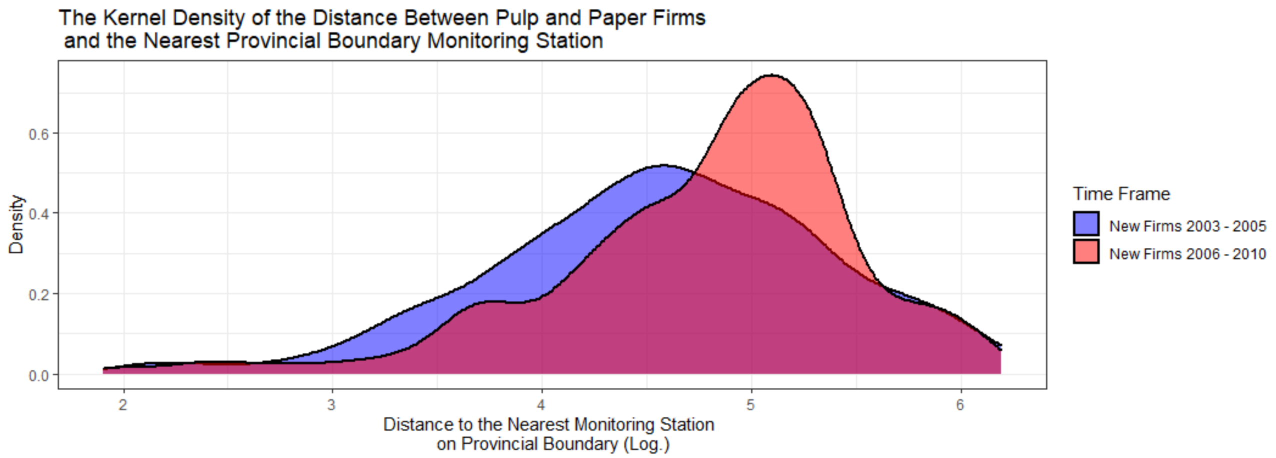
Note that the function allows to adjust the bandwidth manually - the most important parameters are:

- bw: smoothing bandwidth (numeric or character) and;
- adjust: multiplicative bandwidth adjustment (of the default).

If you want to learn more about how to draw kernel densities, have a look at the function's documentation.

Task: Assign the columns time and distance correctly to the corresponding parameter. Then, click on check. Remember, we want to display the density of the distances and fill it differently according to the period when the new firm has been established.

```
# dat.dis.p.long %>%
#   ggplot() +
#   geom_density(size = 2, aes(log(___), fill = ___), alpha = 0.5) +
#   xlab("Distance to the Nearest Monitoring Station \n on Provincial Boundary (Log.)") +
#   ylab("Density") +
#   ggtitle("The Kernel Density of the Distance Between Pulp and Paper Firms \n and the Nearest Provincial Boundary Monitoring Station") +
#   labs(fill = "Time Frame") +
#   scale_fill_manual(labels = c("New Firms 2003 - 2005", "New Firms 2006 - 2010"), values = c("blue", "red")) +
#   theme_bw()

dat.dis.p.long %>%
  ggplot() +
  geom_density(size = 1, aes(log(distance), fill = time), alpha = 0.5) +
  xlab("Distance to the Nearest Monitoring Station \n on Provincial Boundary (Log.)") +
  ylab("Density") +
  ggtitle("The Kernel Density of the Distance Between Pulp and Paper Firms \n and the Nearest Provincial Boundary Monitoring Station") +
  labs(fill = "Time Frame") +
  scale_fill_manual(labels = c("New Firms 2003 - 2005", "New Firms 2006 - 2010"), values = c("blue", "red")) +
  theme_bw()
```



While the density is shown on the vertical, the logarithmic distance to the nearest monitoring station at the provincial boundary is shown on the horizontal axis. Furthermore, the new firms established after the policy change are colored in red, those that started operations before are colored in blue.

It can be clearly seen that pulp and paper plants put in operation between 2006 and 2008 were definitely placed further away from the closest provincial boundary station. This is another indication that the new regulation with regard to the assessment of governors has induced measures that may have led to an actual improvement with regard to excessive border river pollution.

Until now, we focused on plants’ distances to the nearest provincial boundary station. As a consistency check, let us focus now on the distance to the nearest non-provincial boundary station. To do so, start by loading another data set from the file figure_6.dta.

Task: Read figure_6.dta and save it to the variable dat.dis.np. Further, run head to show the first six rows of the data set.

```
#read "figure_6.dta" and save it in a variable called dat.nis.np
#show the head of dat.dis.np
dat.dis.np = read_dta("figure_6.dta")
head(dat.dis.np)

##   newfirm06_08_distance_nbmonitor newfirm03_05_distance_nbmonitor
## 1                           2.003                           1.475
## 2                           2.399                           1.541
## 3                           2.585                           1.648
## 4                           2.647                           1.866
## 5                           3.032                           2.054
## 6                           3.039                           2.167
```

Try to remember what the terms long and wide format regarding data sets stand for. Then, answer the quiz below.

Quiz: What is the format of the data set we just loaded?

- It is the long format. [ ]
- It is the wide format. [x]
- Neither of these. [ ]

As we learnt in previous tasks, data in the long format is represented through key-value pairs. These facilitate visualization, for example. The data set dat.dis.np is in the wide format. Hence, we need to convert it first in the next task.

Note: If you make a mistake and need to load dat.dis.np again, do not hesitate to go to the last chunk and run it again.

Task: Check the chunk to convert the table into the long format.

```
dat.dis.np.long = dat.dis.np %>%
  pivot_longer(cols = everything(), names_to = "time", values_to = "distance")
```

Let us assume the provincial officials made new pulp and paper firms locating further away from the border areas. What should have happened to the average distance between pulp and paper forms and the nearest boundary monitoring station outside of the province? Answer the quiz below to check your guess.

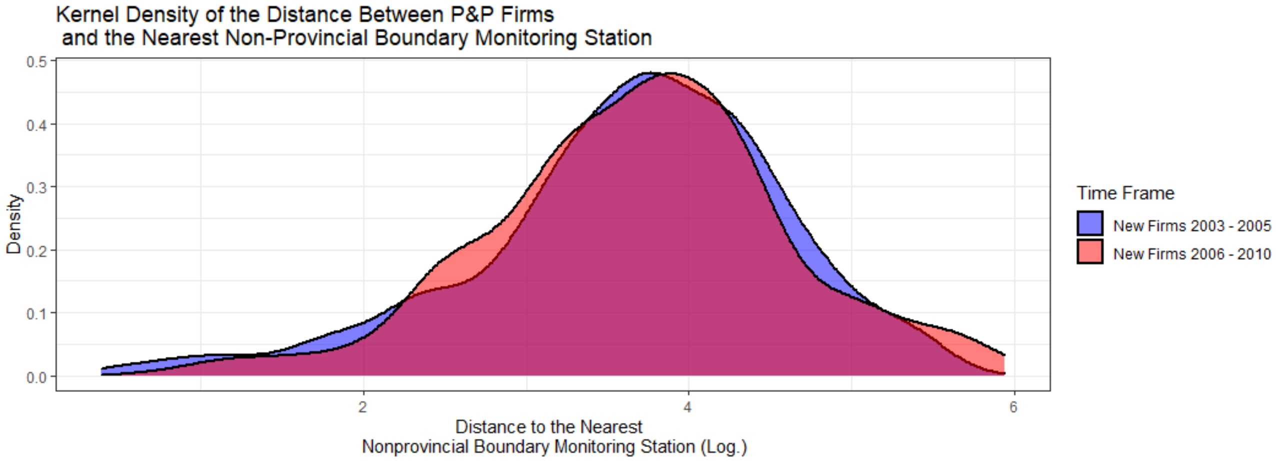
Quiz: What should have happened to the average distance between pulp and paper forms and the nearest boundary monitoring station outside of the province?

- Nothing, it should not have changed by necessity. [x]
- The firms should have moved further away. [ ]
- The firms are now closer. [ ]

If the change in location of new pulp and paper companies was to reduce pollution in the river boundaries, the distance to the nearest non-provincial monitoring station would not necessarily have changed. The reason is that only measurements from provincial monitoring stations are relevant to the governor. In the next task, we show the kernel density of the distance between pulp and paper firms and the next monitoring station outside of the corresponding province before and after 2006.

Task: Just check the chunk in order to visualize the data as before.

```
dat.dis.np.long %>%
  ggplot() +
  geom_density(size = 1, aes(log(distance), fill = time), alpha = 0.5) +
  xlab("Distance to the Nearest \n Nonprovincial Boundary Monitoring Station (Log.)") +
  ylab("Density") +
  ggtitle("Kernel Density of the Distance Between P&P Firms \n and the Nearest Non-Provincial Boundary
Monitoring Station") +
  labs(fill = "Time Frame") +
  scale_fill_manual(labels = c("New Firms 2003 - 2005", "New Firms 2006 - 2010"), values = c("blue",
"red")) +
  theme_bw()
```



Both figures show the kernel density of pulp and paper firms' distances to the closest non-provincial boundary station. The blue one stands for the companies founded between 2003 and 2005, while the red one represents the firms established between 2006 and 2008. The two graphs overlap almost completely.

Combining the results with the plot showing the kernel density of the distance between pulp and paper firms and the nearest provincial boundary monitoring station, we find that definite changes in the location of new companies in the sector only took place with regard to the distance to the provincial boundary station. While those have been placed further away from the border, we cannot find migration patterns like this in the interior of the provinces. Hence, new paper firms must have been positioned purposefully in order to reduce COD pollution at borders.

Summary

What can we learn from these results? It seems like governors indeed intensified their efforts to reduce water pollution at borders. The outcomes of exercise five suggest that the officials successfully prevented companies from the pulp and paper sector being built close to other provinces, intending to reduce river border pollution. On the other hand, one can hypothesize that industrial zones for heavy contaminators have been designated explicitly at provincial river boundaries before 2006, allowing governors to meet environmental and economic plan targets at the same time.

Award: Focus

You have successfully completed the fifth exercise and know what to focus on if you want to reduce pollution effectively!

Exercise 6 – Conclusion

Recapitulation

At this point, we briefly recap what we have covered in the previous chapters. In particular, the methodology and results will be summarized here.

The first exercise motivated for the topic of the paper by first introducing a few features of China's political system relevant to the research question. Furthermore, the relevance of the topic was clarified through presenting first evidence about excessively high water pollution at borders and a first insight into the main data set was provided.

In the second exercise, we gained deeper insights into the data used for the regressions later on. We had a look at the existing river systems, the distribution of monitoring stations and much more. In doing so, we introduced many *R* features. Furthermore, we analyzed the water pollution metrics descriptively. Thus, we learnt that water pollution at borders is mostly heavier than in the interior and that the overall water quality improved massively over the observation period. Nonetheless, we saw that it is necessary to introduce another approach to evaluate the effectiveness of changes in promotion incentives to control river border pollution.

The third exercise consists of several parts and is the centerpiece of our analysis. First, we introduce the difference in differences approach and learnt how to implement it by regression. We built these up step by step by adding fixed effects and control variables to make the analysis more robust against possible deficiencies regarding the parallel trends assumption. Furthermore, we replace the conventional standard errors by cluster-robust standard errors in order to fit the panel data with respect to the dimensions measurement stations and river system/year. Regarding the results, it is important to mention that the COD level has actually decreased significantly stronger over the years at the monitoring stations at the border than in the interior of the provinces. However, there has been no direct link yet between the actions of the provincial officials and the stronger reduction in river border pollution.

In the fourth exercise, we further hypothesize that if the prospect of better promotion opportunities were indeed causal for the stronger COD reduction at borders, younger decision makers would be more likely making efforts in order to fulfill then new environmental plan goals. Again, we run regressions to examine our conjecture. In fact, we note an increased effort by younger governors to favorably reduce COD emissions at border stations. With respect to party secretaries, we cannot find this relationship. This supports our conjectures, since secretaries can be rather seen as supervisors while governors actually being in charge of the government affairs. Nevertheless, we wondered whether it is possible to observe how governors made the water pollution decreasing stronger at border than in the interior of the provinces.

In the fifth exercise, we learn that governors can exert the greatest control over the industrial sector that causes one third of the total COD value measured. Especially, the paper industry plays a very large role regarding COD and should be under special attention. Therefore, we suspect that new factories in this sector were purposefully located further away from the borders after the law was changed. We compare the density function of the distances of paper companies established after 2006 to the nearest measuring station at the border with the distances of companies established from 2003 until the change in the law. We can easily see here that after the change in said promotion incentives, paper companies were clearly placed more likely in more - relative to the border - distant locations.

Results

The paper based on this problem set intensively examined the impact of an amendment by China's central government in 2005 to reduce the free-rider issue regarding extensively high river pollution at provincial boundaries. Evaluation of governors should explicitly consider the pollution level of water flowing into neighboring provinces. The chemical oxygen demand (COD) indicator is used for this purpose. A number of different research methods in the paper yielded results consistent with the authors' thesis. Furthermore they linked the regression results to the promotional ambitions of younger leaders and actual changes where heavy contaminators have been placed. However, the pre-experimental period consisting out of two years only is worth a discussion, although the authors backed their analysis by some extent through introducing fixed effects and additional control variables. Besides that, it is not completely improbable that provincial officials tried to forestall the regime change before it came into effect. In that case, it is hard to observe the true extent of its effect in the study as it is designed.

However, the researchers have shown how the effective implementation of the plan objectives in the five-year plans prepared by the central government can work in the provinces. This is done through a combination of economic federalism and consequent autonomy in the choice of means to achieve the centrally set development goals. By making eligibility for promotion of local officials to the central government directly dependent in the achievement of those goals, as well as limiting the period for promotion through a maximum term limit and maximum age, governors are subjected to competitive pressures. As a result, if specified in the five-year plans, success can be achieved quickly, as demonstrated in this study by addressing higher-than-average levels of water pollution in border regions. Conversely, if not addressed in the central government's plans, problems such as the one discussed in the paper can first be created.

If you are interested in further regression specifications to check the robustness of the results, have a look at the last exercise/appendix. We replace the COD variable with other pollution indicators to test, whether they decreased at a similar extent. Furthermore, instead of including the boundary variable, we want to introduce a continuous measure of distance, proximity to boundary. While we relied on station and year fixed effects in the previous exercises, the same regression model but with river system fixed effects is added.

Related Literature

A similar study is Shen (2017), but using weekly data on pollution measurements. By applying the DiD approach, it compares the impact of overall pollution reduction targets and a stricter water quality assessment system. It proves the influence of changes in environmental policies on boundary pollution. Furthermore, it confirms strategic polluting through positioning contaminating industries near the provincial border, what is consistent with the results from the paper we examined here. Examining how provincial governments try to find a trade-off between border pollution and achieving economic goals is done by Wang (2020) using a threshold model. Interestingly, boundary cities seem not being able to profit from relaxed environmental legislation. Besides that, ranking high in economic growth co-occurs with ambitious environmental protection. Lu (2022) applies a triple difference model to examine the effect of the *Central Environmental Protection Inspection* (CEPI) on excessive water pollution at borders. The CEPI, as described in Wang (2021), is designed to overcame the gap between the central government and lower administration levels that prevents environmental protection laws from becoming fully effective. It is found that the central supervision model achieved significant reductions in border pollution. As we know so far, the pulp and paper industry is one of the main contaminators of COD. A different perspective on the subject is provided by Zhang (2012). With help of scenario models, the researchers examined the effectiveness of several technological approaches to reduce COD pollution. In contrast to raw material substitution, they found that it is highly effective to close small backward paper mills and replacing their capacities by large factories in combination with investments in pollution prevention equipment.

Exercise 7 – References

Bibliography

Abadie, A., Athey, S., Imbens, G. W., & Wooldridge, J. (2017). When should you adjust standard errors for clustering? (No. w24003). National Bureau of Economic Research.

Angrist Joshua D., Pischke Jörn-Steffen. (2008). “Mostly Harmless Econometrics - An Empiricist's Companion.”.

Arellano, M. (1987). Computing robust standard errors for within-groups estimators. Oxford bulletin of Economics and Statistics, 49(4), 431-434.

Babich, H., & Davis, D. L. (1981). Phenol: A review of environmental and health risks. Regulatory Toxicology and Pharmacology, 1(1), 90-109.81.

Bergé, L. (2018). Efficient estimation of maximum likelihood models with multiple fixed-effects: the R package FENmlm (No. 18-13). Department of Economics at the University of Luxembourg.

Brenniman G.R. (1999) Biochemical oxygen demand. In: Environmental Geology. Encyclopedia of Earth Science. Springer, Dordrecht.

Cai Hongbin, Yuyu Chen, and Gong Qing. (2013). “Polluting The Neighbor: The Case of River Pollution in China.” Unpublished.

Cameron, A. C., Gelbach, J. G., & Miller, D. L. (2006). Robust inference with multi-way clustering (Working Paper 09-8). Department of Economics, University of California-Davis.

Cunningham Scott. (2021). “Causal Inference”.

Duvivier, Chloé, and Hang Xiong. 2013. “Transboundary Pollution in China: A Study of Polluting Firms’ Location Choices in Hebei Province.” Environment and Development Economics 18 (4):459–83.

Gray, W. B., & Shadbegian, R. J. (1998). Environmental regulation, investment timing, and technology choice. The Journal of Industrial Economics, 46(2), 235-256.

Gray, W. B., & Shadbegian, R. J. (2004). ‘Optimal’ pollution abatement—whose benefits matter, and how much?. Journal of Environmental Economics and management, 47(3), 510-534.

Greene, W.H. (2008). Econometric Analysis. 6th Edition, Pearson Prentice Hall, Upper Saddle River.

Heilmann, S. (2004). Das politische System der Volksrepublik China.

Hu Z., Grasso D. (2005). WATER ANALYSIS | Chemical Oxygen Demand, 325 - 330.

Kahn, M. E., Li, P., & Zhao, D. (2015). Water pollution progress at borders: the role of changes in China's political promotion incentives. American Economic Journal: Economic Policy, 7(4), 223-42.

Karri, R. R., Sahu, J. N., & Chimmiri, V. (2018). Critical review of abatement of ammonia from wastewater. Journal of Molecular Liquids, 261, 21-31.

Laplante, B., & Rilstone, P. (1996). Environmental inspections and emissions of the pulp and paper industry in Quebec. Journal of Environmental Economics and management, 31(1), 19-36.

Lu, J. (2022). Can the central environmental protection inspection reduce transboundary pollution? Evidence from river water quality data in China. Journal of Cleaner Production, 332, 130030.

Ma, J., Pan, F., He, J., Chen, L., Fu, S., & Jia, B. (2012). Petroleum pollution and evolution of water quality in the Malian River Basin of the Longdong Loess Plateau, Northwestern China. Environmental Earth Sciences, 66(7), 1769-1782.

Montinola, G., Qian, Y., & Weingast, B. R. (1995). Federalism, Chinese style: the political basis for economic success in China. World politics, 48(1), 50-81.

Petersen, M. A. (2009). Estimating standard errors in finance panel data sets: Comparing approaches. The Review of financial studies, 22(1), 435-480.

Sandler, T. (2006). Regional public goods and international organizations. The Review of International Organizations, 1(1), 5-25.

Shen, M., & Yang, Y. (2017). The water pollution policy regime shift and boundary pollution: Evidence from the change of water pollution levels in China. Sustainability, 9(8), 1469.

Sigman, H. (2002). International spillovers and water quality in rivers: do countries free ride?. American Economic Review, 92(4), 1152-1159.

Sigman, H. (2005). Transboundary spillovers and decentralization of environmental policies. Journal of environmental economics and management, 50(1), 82-101.

Thompson, S. B. (2011). Simple formulas for standard errors that cluster by both firm and time. Journal of financial Economics, 99(1), 1-10.

Varian, H. R. (2014). "Intermediate microeconomics : a modern approach" (9th ed.) W.W. Norton & Company.

Water Quality Criterion for the Protection of Human Health: Methylmercury (2001), accessed 19 October 2021, https://www.epa.gov/sites/default/files/2019-02/documents/wqc-final-methylmercury-factsheet-2001.pdf.

Wang, Q., Fu, Q., Shi, Z., & Yang, X. (2020). Transboundary water pollution and promotion incentives in China. Journal of Cleaner Production, 261, 121120.

Wang, M. (2021). Environmental governance as a new runway of promotion tournaments: campaign-style governance and policy implementation in China’s environmental laws. Environmental Science and Pollution Research, 28(26), 34924-34936.

Xianbin, W. (2008). Sources, Whereabouts and Tenures of Local Officials and Economic Growth. Management World, 3.

Zhang, C., Chen, J., & Wen, Z. (2012). Alternative policy assessment for water pollution control in China's pulp and paper industry. Resources, Conservation and Recycling, 66, 15-26.

R Packages

Berge L (2018). “Efficient estimation of maximum likelihood models with multiple fixed-effects: the R package FENmlm.” CREA Discussion Papers.

D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

Hadley Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Hadley Wickham and Evan Miller (2021). haven: Import and Export ‘SPSS’, ‘Stata’ and ‘SAS’ Files. R package version 2.4.3. https://CRAN.R-project.org/package=haven

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. https://CRAN.R-project.org/package=readxl

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). dplyr: A Grammar of Data Manipulation. R package version 1.0.7. https://CRAN.R-project.org/package=dplyr

Hadley Wickham (2021). tidyr: Tidy Messy Data. R package version 1.1.4. https://CRAN.R-project.org/package=tidyr

Hao Zhu (2021). kableExtra: Construct Complex Table with ‘kable’ and Pipe Syntax. R package version 1.3.4. https://CRAN.R-project.org/package=kableExtra

Sebastian Kranz (2020). RTutor: Interactive R problem sets with automatic testing of solutions and automatic hints. R package version 2020.11.25.

Vincent Arel-Bundock (2021). modelsummary: Summary Tables and Plots for Statistical Models and Data: Beautiful, Customizable, and Publication-Ready. R package version 0.9.4. https://CRAN.R-project.org/package=modelsummary

Exercise Appendix – Further Robustness Checks

To avoid overloading the regular exercises, I appended alternative regression specifications in order to verify the robustness of the results. The authors modified the regression models by replacing COD with other pollution indicators, the binary boundary dummy variable with a continuous measure, proximity to boundary as well as the binary dummy Post2006 with a time trend variable. Furthermore, river system fixed effects are introduced. Last but not least, I developed a triple difference approach by regression to complement the authors analysis. The appendix is structured as follows:

Structure

A1. COD Discharges and Its Determinants: Time Trend and Proximity to Boundary

A2. COD Discharges and Its Determinants: River Fixed Effects

A3. COD Dynamics versus Other Indicators of Water Pollution

A4. Leader Career Concerns and Pollution Dynamics - COD versus Other Pollution Indicators

A5. Triple Difference Approach

Exercise A1 – COD Discharges and Its Determinants: Time Trend and Proximity to Boundary

In Exercise 3.2, we included a dummy variable called boundary to indicate whether an observation was made at a border monitoring station. Furthermore, we used post06 equal to 1 for measurements that have been made after the new environmental policy came into effect in 2006. In this exercise, we will compare the results from Exercise 3.4 with those from regression models replacing the dummy variables boundary and post06 through continuous measures of proximity to boundary and a time trend variable. Additionally, we examine what happens if we go without economic controls.

First of all, we need to load the data set.

Task: Check the chunk to save the data set in dat.

```
dat = readRDS("dat.RDS")
```

In the following, you find brief definitions of old and new variables.

Economic Controls

While we included all controls at once in Exercise 3.4, we want to present the results by excluding the following variables from the regression models:

- GDPG
- GDPP
- Luminosity

Time Trend

The variable we use to represent the time trend is tpost. It is a number indicating how many years have passed since the policy change came into effect. This setting allows a more precise localization of the relative water quality improvement with regard to the year it has taken place. Have a look at the table that follows to find out what it means:

```
data.frame(Year = c(2004:2010), tpost = c(0,0,1,2,3,4,5)) %>%
  kbl() %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Year	tpost
2004	0
2005	0
2006	1
2007	2
2008	3
2009	4
2010	5

Proximity to Boundary

This continuous variable measures the distance between the monitoring station where the observation has been made and the provincial border. Introducing a continuous measure may allow to capture the effect of the policy change better. Before we continue let us modify the boundary distance as the authors did. Note that by doing so, the interpretation of the boundary_distance coefficient changes.

$$BoundaryDistance_{mod} = (500 - BoundaryDistance)/10$$

Task: Modify boundary_distance according to the formula below. Store the result in a new column named prox in the data set dat.

```
# dat = dat %>%
#   mutate("prox" = ___ )

dat = dat %>%
  mutate("prox" = (500 - boundary_distance)/10)
```

The higher the observed value of prox, the closer the monitoring station is to the border, while it has the maximum value for boundary stations. Hence, a positive regression coefficient means that the closer the station is to the border, the higher the pollution measured.

Results

In the next task, I outlined all regression models the authors presented in the paper:

- Model 1: Conventional regression;
- Model 2: Without economic controls;
- Model 3: Time trend;
- Model 4: Proximity to boundary;
- Model 5: Time trend and proximity to boundary.

Task: Check the chunk to estimate all of these regression models at once:

```
#1
reg.1 = feols(cod ~ boundary:post06 + gdpg + gdpp + temperature + lightbuffer5km | station + year, cluster =
c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#2
reg.2 = feols(cod ~ boundary:post06 + temperature | station + year, cluster = c("station", "riversystem"), ssc =
ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#3
reg.3 = feols(cod ~ boundary:tpost + gdpg + gdpp + temperature + lightbuffer5km | station + year, cluster =
c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#4
reg.4 = feols(cod ~ prox:post06 + gdpg + gdpp + temperature + lightbuffer5km | station + year, cluster =
c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#5
reg.5 = feols(cod ~  prox:tpost  + gdpg + gdpp + temperature + lightbuffer5km | station + year, cluster =
c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

After saving all regressions, let us show the results in a neatly arranged fashion:

Task: Check the chunk to compare the regression results of all models estimated before.

```
modelsummary(list(reg.1, reg.2, reg.3, reg.4, reg.5),
            coef_omit = "year|temp|light|gd",
            coef_rename = c("boundary:post06" = "Boundary x Post2006", "boundary:tpost" = "Boundary x
Time Trend", "prox:post06" = "Proximity to Boundary x Post2006", "prox:tpost" = "Proximity to Boundary x
Time Trend"),
            add_rows = data.frame(c("Station Dummy", "Year Dummy", "Temperature", "Economic Controls"),
                    c("Yes", "Yes", "Yes", "Yes"),
                    c("Yes", "Yes", "Yes", ""),
                    c("Yes", "Yes", "Yes", "Yes"),
                    c("Yes", "Yes", "Yes", "Yes"),
                    c("Yes", "Yes", "Yes", "Yes")),
            stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
            gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|F",
            title = "Cod Discharges and Its Determinants With Station Fixed Effects, Time Trend and Proximity
to Boundary") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Cod Discharges and Its Determinants With Station Fixed Effects, Time Trend and Proximity to Boundary

	Model 1	Model 2	Model 3	Model 4	Model 5
Boundary x Post2006	-2.012*	-2.138*			
	(1.192)	(1.271)			
Boundary x Time Trend			-0.543		
			(0.342)		
Proximity to Boundary x Post2006				-0.129**	
				(0.057)	
Proximity to Boundary x Time Trend					-0.052**
					(0.023)
Num.Obs.	3377	3377	3377	3377	3377
Station Dummy	Yes	Yes	Yes	Yes	Yes
Year Dummy	Yes	Yes	Yes	Yes	Yes
Temperature	Yes	Yes	Yes	Yes	Yes
Economic Controls	Yes		Yes	Yes	Yes
*** p < 0.1, ** p < 0.05, *** p < 0.01**					

The first column represents the regression with station and year fixed effects and all control variables from Exercise 3.4. The next column is the same specification but without economic controls. Model 3 replaces the post06 dummy with the time trend variable and Model 4 replaces the boundary dummy with proximity to boundary. In the last column, both dummies are substituted with their continuous counterparts.

Excluding economic controls in Model 2 does not change the magnitude or the significance of the interaction’s coefficient much. Including the time trend variable but keeping the boundary variable in the third model makes the coefficient insignificant. In the last two models, were we replaced the boundary dummy through proximity to boundary, the coefficient is negative and becomes significant at the five percent level.

Summary

Regardless of the specification, all estimated coefficients are negative which is consistent with the results we have got in the main analysis. It implies that a stronger reduction of COD pollution took place at borders and after 2006. Furthermore, the significance of the result increases when introducing the more precise variable that measures proximity to boundary.

Exercise A2 – COD Discharges and Its Determinants: River Fixed Effects

In Exercise 3.4 as well as in the previous appendix section, we included station and year fixed effects in the regression. Instead, we apply river system fixed effects in the following. Aside from that, it is the same formula as we put together in Exercise 3.4, including the controls and fixed effects mentioned there.

Note: While applying xtivreg2()'s procedure to calculate cluster-robust standard errors in all other exercises, the authors decided to use the approach cluster2 coded by Mitchell A. Peterson based on papers written by Thompson (2011) and Cameron (2006). The main difference between xtivreg()'s approach is the way *small sample correction (ssc)* is applied. If you are interested in how *ssc* differs by the regression functions, read the info box below.

Info: Small Sample Correction - Comparison

Parameter/ Function	**feols**	**xtivreg2**	**cluster2**
ssc of the form $(\boldsymbol{n-1})/(\boldsymbol{n-K})$	TRUE	FALSE	TRUE
$\boldsymbol{G/(G-1)}$ correction is performed	TRUE	FALSE	TRUE
degrees of freedom of the Student t distribution	minimum size of the clusters with which the variance-covariance matrix has been clustered	number of observations minus the number of estimated variables	minimum size of the clusters with which the variance-covariance matrix has been clustered
$\boldsymbol{G}$ in $\boldsymbol{G/(G-1)}$	use smallest $G_i = G_{min}$ for all matrices	use smallest $G_i = G_{min}$ for all matrices	use G_i for the i-th sandwich matrix

with * i being the number of unique dimensions, that is cluster variables (in our example with station and riversystem_time: $i = 2$) and * G_i being the number of groups belonging to dimension i (for instance, the number of all unique manifestations of riversystem_time)

Task:: Check the chunk in order to load both the data set and the regression from Exercise 3.4, where station and year fixed effects have been included.

```
#load data set
dat = readRDS("dat.RDS")
#save regression with station and year fixed effects from exercise 3.4
reg.ctrl = feols(cod ~ post06:boundary + temperature + gdpg + gdpp + lightbuffer5km | station + year, cluster
= c("station", "riversystem_time"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data
= dat)
```

River Fixed Effects

In the following, we would like to run the same regression but with river fixed effects. The idea is being able to probably capture hidden effects and background changes better than through station and year fixed effects.

Task: Just check the chunk in order to re-estimate the regression, but this time with river fixed effects.

```
reg.riv = feols(cod ~ post06 + boundary + post06:boundary + gdpg + gdpp + temperature + lightbuffer5km |
riversystem, cluster = c("station", "riversystem_time") , ssc = ssc(adj = TRUE, cluster.adj = TRUE, cluster.df
= "conventional"), data = dat)
```

Proximity To Boundary

As we did in the previous exercise, we would like to estimate the regression from the previous task again, but this time with proximity to boundary prox instead of the dummy variable boundary. If the policy change had been effective, we expect that monitoring stations closer to the border must have experienced a stronger water quality improvement after 2006 compared to stations that are more distanced to the provincial boundary.

Task: Let us take the regression formula from task 4.2.5 and replace the respective terms.

```
reg.riv.prox = feols(cod ~ post06 + boundary + post06:boundary + temperature + gdpg + gdpp +
lightbuffer5km | riversystem, cluster = c("station", "riversystem_time") , ssc = ssc(adj = TRUE, cluster.adj =
TRUE, cluster.df = "conventional"), data = dat)



reg.riv.prox = feols(cod ~ post06 + prox + post06*prox + gdpg + gdpp + temperature + lightbuffer5km |
riversystem, cluster = c("station", "riversystem_time") , ssc = ssc(adj = TRUE, cluster.adj = TRUE, cluster.df
= "conventional"), data = dat)
```

Let us add the results to our table in order to compare them.

Task: Just check the following chunk.

```
modelsummary(list("Station and Year FE" = reg.ctrl, "River System FE - Boundary Dummy" = reg.riv,
"River System FE - Proximity to Boundary" = reg.riv.prox),
        coef_rename = c("post06:boundary" = "Post2006 x Boundary", "post06" = "Post2006", "boundary" =
"Boundary", "prox" = "Proximity to Boundary", "post06:prox" = "Post2006 x Proximity to Boundary"),
        coef_omit = "year|temp|light|gd",
        stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
        gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|F",
        title = "COD Discharges and Its Determinants: Station and Year Versus River Fixed Effects") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

COD Discharges and Its Determinants: Station and Year Versus River Fixed Effects			
	Station and Year FE	**River System FE - Boundary Dummy**	**River System FE - Proximity to Boundary**
Post2006 x Boundary	-2.012*	-1.888**	
	(1.119)	(0.751)	
Post2006		-1.528***	3.139*
		(0.137)	(1.869)
Boundary		2.814**	
		(1.398)	
Proximity to Boundary			0.169**
			(0.074)
Post2006 x Proximity to Boundary			-0.112**
			(0.045)
Num.Obs.	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**			

The first column represents the regression with station and year fixed effects and all control variables from Exercise 3.4. In the second regression, station and year fixed effects are replaced by river fixed effects. The last column contains the same regression as in the second, but instead of the boundary dummy the continuous measure proximity to boundary is applied

In the second regression, the coefficient on the interaction term between the post06 and the boundary dummy is negative and now significant at the five percent level. This indicates a stronger COD reduction at borders after the policy change came into effect and is consistent with the result we have got when applying station and year fixed effects. The coefficient on the interaction term between the post06 and prox in the third regression is significant on the ten percent level only, but still negative and thus at least not contradicting the results of the other two regressions.

Summary:

Replacing station and year fixed effects through river system fixed effects makes the relevant interaction’s coefficient even significant at a higher level. When replacing the boundary dummy through proximity to boundary, the result is significant at the ten percent level, but still consistent with the other results.

Exercise A3 – COD Dynamics versus Other Indicators of Water Pollution

The government's change in promotion criteria targets the water pollution with regard to chemical oxygen demand (COD) only. Hence, we assume that pollution measured by this indicator should have decreased by a larger amount compared to other pollution indicators at borders. To test this hypothesis, we perform a series of regressions not only with COD as dependent variable, but also five other pollution metrics. These are of interest in terms of general health, but have not been included by the government in the promotion guidelines. Below, you find a list and description of those indicators.

Info: Description of Pollution Indicators

We find a total of six different measures of water quality levels in our data set. The higher the values, the more dangerous the use of the water is for humans. Below is a brief description and hazards of each indicator. If you want to learn more, have a look at the corresponding links I added.

COD: This abbreviation stands for chemical oxygen demand and results are received by the COD test. COD is hereby the amount of dissolved oxygen that is needed to oxidize, that is chemically break down the chemical organic components in the water sample. For instance, the higher the petroleum concentration is in the sample, the higher the chemical oxygen demand is (Hu, 2005).

BOD: The biological oxygen demand is correlated with COD and is the amount of oxygen required to break up the organic matter biologically by micro-organisms. Again, the higher degree of pollution the more oxygen is required and therefore the higher this key figure the worse the water quality is (Brenniman, 1999).

NH: NH stands for ammonia nitrogen and it pollutes water through agriculture and the certain industry sectors like plastics and paper. Although it is only toxic to humans when surpassing their detoxifying capacities, it is a danger to aquatic life. Therefore we consider lower values as an indicator for less pollution extents. (Karri, 2018)

Petroleum: Contamination with Petroleum is mainly caused by the oil industry. Leaks and oilfield wastewater are just two of several possible sources. Especially the later one is enriched with minerals, radioactive substances, benzenes, phenols, and polycyclic aromatic hydrocarbons being harmful to human and thus making water unsafe to drink. Again, higher concentrations are logically considered worse. (Ma, 2012)

Mercury: Mercury, which is mainly emitted through energy production and other industrial activities is not only considered as a serious risk to wildlife. Due to its ability to accumulate in fish and shellfish, it also poses a danger to humans, who then ingested it through food. It has been shown that developmental brain disorders can occur, especially in unborn children, as a result of the pregnant woman's ingestion (Water Quality Criterion for the Protection of Human Health: Methylmercury, 2001).

Phenol: Phenol is a component of wastewater from a wide range of industries, including, for example, the paper, chemical, coal and oil industries. It is not only harmful to microorganisms in polluted waters. Animal studies also indicate carcinogenic and fertility-damaging behavior.
In higher doses, it is also acutely toxic to humans (Babich, 1981).

Before we start, we load the data set Regression_Data.dta with help of read_dta() and store it in a variable called dat.

Task: Just check the following chunk to load the required data set.

```
dat = readRDS("dat.RDS")
```

For the regression, we apply monitoring station and year fixed effects. Furthermore, to allow the border effect to vary by calendar year, we include an interaction term between each year and the boundary dummy. Therefore, the regression equation looks like this:

$$pol.level_{it} = Y_t + boundary_i + Y_t \times boundary_i + C_{it} + \varepsilon_{it}$$

The C_{it} part of the regression consists out of both the following control variables and their respective regression coefficients:

- gdpp
- gdpg
- lightbuffer5km
- temperature

In sum, we have six interaction terms between the year and the boundary dummy, that is, one combination per year and the boundary dummy variable. As you might notice below, we have one interaction term less than actual years. This is because the year 2004 is the omitted category here.

Let us start by adding a dummy variable for each combination of the dummy variable boundary and a manifestation of the categorical variable year.

Quiz: How many dummy variables are we about to create in this step?

- Five [ ]
- Six [x]
- Seven [ ]
- Eight [ ]

We are about to create one dummy variable per combination of the boundary dummy and year. As the year 2004 serves as omitted category, there is no need for creating a dummy. Hence, one dummy variable per year from 2005 to 2010 amounts to a total number of six.

Task: Just check the following chunk to create the desired dummy variables.

```
dat = dat %>%
  mutate("by05" = boundary * ifelse(year == 2005, 1, 0)) %>%
  mutate("by06" = boundary * ifelse(year == 2006, 1, 0)) %>%
  mutate("by07" = boundary * ifelse(year == 2007, 1, 0)) %>%
  mutate("by08" = boundary * ifelse(year == 2008, 1, 0)) %>%
  mutate("by09" = boundary * ifelse(year == 2009, 1, 0)) %>%
  mutate("by10" = boundary * ifelse(year == 2010, 1, 0))
```

Maybe you wonder, why I have not replaced the interactions between year and boundary simply by year:boundary. As mentioned before, I did not include by04 - the interaction between boundary:year2004 - to avoid collinearity. However, according to the developers of the fixest package, there is no way to select the variable one wishes to omit. Since feols implicitly chooses to omit boundary:year2010, I decided to include the interactions manually to guarantee the same representation of the regression results as in the paper, where `boundary:year2004 is omitted.

In the next step, we are going to include all interaction terms between the boundary and year dummy as well as the controls and year and station dummy. In total, we run six regressions.

Task: Just check the chunk to run the regressions and save the results.

```
#cod
reg.cod = feols(cod ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg + lightbuffer5km +
temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
#bod
reg.bod = feols(bod ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg + lightbuffer5km +
temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
#ammonia nitrogen
reg.nh = feols(nh ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg + lightbuffer5km +
temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
#petroleum
reg.petroleum = feols(petroleum ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
#phenol
reg.phenol = feols(phenol ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg + lightbuffer5km +
temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
```

```
## Variance contained negative values in the diagonal and was 'fixed' (a la Cameron, Gelbach & Miller
2011).
```

```
#mercury
reg.mercury = feols(mercury ~ by05 + by06 + by07 + by08 + by09 + by10 + gdpp + gdpg + lightbuffer5km +
temperature | year + station, cluster = c("station", "riversystem_time"), data = dat)
```

Task: Just check the chunk to show the regression results:

```
modelsummary(list("COD" = reg.cod, "BOD" = reg.bod, "NH" = reg.nh, "Petroleum" = reg.petroleum,
"Mercury" = reg.mercury, "Phenol" = reg.phenol),
          coef_omit = "year|temp|light|gd",
          coef_rename = c("by05" = "Boundary x Year2005", "by06" = "Boundary x Year2006", "by07" =
"Boundary x Year2007", "by08" = "Boundary x Year2008", "by09" = "Boundary x Year2009", "by10" =
"Boundary x Year2010"),
          stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
          gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R|FE",
          title = "Water Pollutants and Its Determinants") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Water Pollutants and Its Determinants

	COD	BOD	NH	Petroleum	Mercury	Phenol
Boundary x Year2005	-1.609	0.463	0.268	2.874	1.285	0.315**
	(1.159)	(0.643)	(0.198)	(3.317)	(1.080)	(0.134)
Boundary x Year2006	-2.737*	-1.114	-0.355	0.040	0.777	0.409
	(1.472)	(0.749)	(0.270)	(3.371)	(1.372)	(0.495)
Boundary x Year2007	-1.958	0.451	0.149	2.028	0.225	0.513
	(1.387)	(0.566)	(0.345)	(4.716)	(1.180)	(0.416)
Boundary x Year2008	-2.358	0.835	-0.310	2.974	1.050	1.212
	(1.529)	(1.186)	(0.336)	(4.187)	(0.998)	(0.911)
Boundary x Year2009	-3.218*	0.024	-0.130	1.389	0.731	0.173
	(1.913)	(1.158)	(0.346)	(4.297)	(1.060)	(0.206)
Boundary x Year2010	-3.942**	-0.903	-0.444	-2.137	0.834	0.101
	(1.944)	(1.203)	(0.382)	(4.444)	(1.052)	(0.248)
Num.Obs.	3377	3377	3377	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**						

We can observe a greater improvement at the border compared to non-border stations for the COD level in the years 2005, 2006, 2008, 2010 at the 10 % and 2007 on the 5 % significance level. In general, the coefficients are jointly negative and increase in absolute values over the years, that is consistent as the water quality is improving over the observation period. The absolute strongest relative improvement can be found in 2010, where the authors suggest that, in order to meet the central government's targets, more resources have been allocated. In contrast we find almost no significant relative improvement regarding the other pollution indicators, except for the BOD level in 2006. Hence, there is virtually no evidence for a comparable stronger water quality improvement at borders for the other indicators, that have not been considered in the new environmental law.

Proximity to Boundary

Similar to the sections of the appendix before, we replace the boundary dummy by its continuous counterpart. In the following, we create interaction terms between the time trend tpost and proximity to boundary prox.

Task: Just check the following chunk to create the desired dummy variables.

```
dat = dat %>%
  mutate("prox05" = prox * ifelse(year == 2005, 1, 0)) %>%
  mutate("prox06" = prox * ifelse(year == 2006, 1, 0)) %>%
  mutate("prox07" = prox * ifelse(year == 2007, 1, 0)) %>%
  mutate("prox08" = prox * ifelse(year == 2008, 1, 0)) %>%
  mutate("prox09" = prox * ifelse(year == 2009, 1, 0)) %>%
  mutate("prox10" = prox * ifelse(year == 2010, 1, 0))
```

In the next step we are going to replace the boundary dummy variable in the regressions with the continuous measure of distance, prox.

Note: The authors clustered by station and riversystem instead of station and riversystem_time as before. Hence, we clustered by station and riversystem in the following regressions, too.

Task: Just check the chunk in order to run the same regressions as in the task before, but replacing boundary with prox.

```
#cod
reg.cod = feols(cod ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#bod
reg.bod = feols(bod ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#ammonia nitrogen
reg.nh = feols(nh ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg + lightbuffer5km
+ temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj =
FALSE, t.df = "conventional"), data = dat)
#petroleum
reg.petroleum = feols(petroleum ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#mercury
reg.mercury = feols(mercury ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#phenol
reg.phenol = feols(phenol ~ prox05 + prox06 + prox07 + prox08 + prox09 + prox10 + gdpp + gdpg +
lightbuffer5km + temperature | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

Task: Just check the chunk to show the regression results:

```
modelsummary(list("COD" = reg.cod, "BOD" = reg.bod, "NH" = reg.nh, "Petroleum" = reg.petroleum,
"Mercury" = reg.mercury, "Phenol" = reg.phenol),
          coef_omit = "year|temp|light|gd",
          coef_rename = c("by05" = "Proximity to Boundary x Year2005", "by06" = "Proximity to Boundary x
Year2006", "by07" = "Proximity to Boundary x Year2007", "by08" = "Proximity to Boundary x Year2008",
"by09" = "Proximity to Boundary x Year2009", "by10" = "Proximity to Boundary x Year2010"),
          stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
          gof_omit = "AIC|BIC|Std.Errors|Log.Lik|F|R",
          title = "Water Pollutants and Its Determinants - Proximity to Boundary") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Water Pollutants and Its Determinants - Proximity to Boundary						
	COD	**BOD**	**NH**	**Petroleum**	**Mercury**	**Phenol**
prox05	-0.011	-0.064**	-0.016**	0.099	0.041	-0.003
	(0.054)	(0.033)	(0.008)	(0.197)	(0.097)	(0.009)
prox06	-0.004	-0.054	-0.037	-0.009	-0.085	0.015
	(0.086)	(0.036)	(0.028)	(0.158)	(0.177)	(0.016)
prox07	-0.072*	-0.089	-0.010	0.001	-0.060	0.022
	(0.040)	(0.061)	(0.013)	(0.208)	(0.137)	(0.021)
prox08	-0.160**	-0.188**	-0.032	0.213	-0.005	0.054
	(0.070)	(0.085)	(0.025)	(0.318)	(0.086)	(0.045)
prox09	-0.214*	-0.265***	-0.037	0.096	-0.011	0.009
	(0.112)	(0.100)	(0.034)	(0.366)	(0.110)	(0.020)
prox10	-0.241*	-0.292**	-0.051	-0.071	0.016	0.008
	(0.126)	(0.122)	(0.033)	(0.318)	(0.099)	(0.021)
Num.Obs.	3377	3377	3377	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**						

The interaction terms regarding the years 2008 until 2010 are jointly significant and negative for COD. The results for BOD are quite similar. Both indicators, COD and BOD depend on the water's amount of organic compounds. Analogically to the results from the previous regression set, we cannot find significant improvement regarding the other pollution metrics - except for ammonia nitrogen in 2005. In contrast to the results from the regressions before, we cannot find immense COD reduction in 2006. The strongest relative improvements occurred between 2007 and 2009. The same is true for BOD.

Summary:

With help of two regression models we examined in which years significant changes in pollution concentrations occurred for all pollution indicators in the set. Aside from COD and its related indicator, BOD, we cannot really find stable significant trend of pollution reduction. Hence, we conclude that the environmental scheme indeed incentivized local officials to improve water quality at borders by targeting COD. Since the new environmental policy focused on COD only, we find differential water pollution progress primarily with regard to this indicator and its related metric, BOD.

Exercise A4 – Leader Career Concerns and Pollution Dynamics - COD versus Other Pollution Indicators

In this additional exercise, you find the analysis regarding the relationship between the age of provincial officials and their efforts to reduce border pollution, but this time extended to all pollution indicators. Before we start, we need to load the data set and add three variables used in the regressions.

Task: Check the chunk to load the required data set.

```
dat = readRDS("dat.RDS")
```

In the following, we run and show the results of the regressions from Exercise 4, not only with COD as dependent variable, but also with other water pollution indicators. Just as in the appendix section before, they use the continuous measure of proximity of a monitoring station to the nearest border prox.

Task: Check the chunk to re-estimate the regression results from Exercise 4, but now for all available water pollution indicators.

```
#cod
reg.cod = feols(cod ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#bod
reg.bod = feols(bod ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#ammonia nitrogen
reg.nh = feols(nh ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#petroleum
reg.petroleum = feols(petroleum ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#phenol
reg.phenol = feols(phenol ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#mercury
reg.mercury = feols(mercury ~ tpost:boundary:post06 + g_age:boundary:post06:tpost + g_age +
s_age:boundary:post06:tpost + s_age + gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster
= c("station", "riversystem"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

Task: Check the chunk to show the regression results.

```
modelsummary(list("COD" = reg.cod, "BOD" = reg.bod, "NH" = reg.nh, "Petroleum" = reg.petroleum,
"Mercury" = reg.mercury, "Phenol" = reg.phenol),
          coef_omit = "year|temp|light|gd",
          coef_rename = c("tpost:boundary:post06" = "Boundary x Time Trend", "g_age" = "Governor Age",
"s_age" = "Secretary Age", "tpost:boundary:post06:g_age" = "Governor Age x Boundary x Time Trend",
"tpost:boundary:post06:s_age" = "Secretary Age x Boundary x Time Trend"),
          stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
          gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE",
          title = "Leader Career Concerns and Pollution Dynamics") %>%
 kable_classic() %>%
 kable_styling(font_size = 15)
```

Leader Career Concerns and Pollution Dynamics

	COD	BOD	NH	Petroleum	Mercury	Phenol
Governor Age	-0.004	0.202	0.010	-0.183	-0.094**	-0.024
	(0.071)	(0.206)	(0.026)	(0.404)	(0.042)	(0.019)
Secretary Age	-0.052	0.054	0.001	0.337	0.092**	0.012*
	(0.062)	(0.063)	(0.007)	(0.206)	(0.045)	(0.007)
Boundary x Time Trend	-4.988**	2.652	-1.991	17.278	-0.730	1.337*
	(2.224)	(2.829)	(1.479)	(11.889)	(1.255)	(0.712)
Governor Age x Boundary x Time Trend	0.065***	-0.048	0.019**	-0.143	0.021	-0.018
	(0.018)	(0.057)	(0.010)	(0.150)	(0.013)	(0.012)
Secretary Age x Boundary x Time Trend	0.012	-0.001	0.013	-0.156**	-0.007	-0.005**
	(0.022)	(0.029)	(0.015)	(0.062)	(0.013)	(0.002)
Num.Obs.	3377	3377	3377	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**						

For comparison, the first column contains the results for COD from Exercise 4. The other columns contain the other five pollution indicators. However, for those, in contrast to the COD result, we cannot find evidence for differential water quality improvement in the years after 2006 at borders when governors are younger. Aside from the COD regression, the only coefficient for the interaction between the governor's age, the boundary dummy and time trend variable being significant at least at the five percent level, is the ammonia nitrogen (NH) regression. Interestingly, the coefficient of the interaction between the secretaries' age, the boundary dummy and the time trend is significant for petroleum and phenol.

In the next task, the authors replaced the discrete boundary dummy with the continuous proximity to the boundary measure and re-estimated the regression set.

Task: Just check the chunk to re-estimate the results, but with the continuous proximity to the boundary measure.

```
#cod
reg.cod.prox = feols(cod ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age + gdpg + gdpp
+ temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#bod
reg.bod.prox = feols(bod ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age + gdpg + gdpp
+ temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#ammonia nitrogen
reg.nh.prox = feols(nh ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age + gdpg + gdpp +
temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj = FALSE,
cluster.adj = FALSE, t.df = "conventional"), data = dat)
#petroleum
reg.petroleum.prox = feols(petroleum ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age +
gdpg + gdpp + temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc =
ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#mercury
reg.mercury.prox = feols(mercury ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age + gdpg
+ gdpp + temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj =
FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
#phenol
reg.phenol.prox = feols(phenol ~ tpost:prox + g_age:prox:tpost + g_age + s_age:prox:tpost + s_age + gdpg +
gdpp + temperature + lightbuffer5km | year + station, cluster = c("station", "riversystem"), ssc = ssc(adj =
FALSE, cluster.adj = FALSE, t.df = "conventional"), data = dat)
```

Task: Check the chunk to show the regression results.

```
modelsummary(list("COD" = reg.cod.prox, "BOD" = reg.bod.prox, "NH" = reg.nh.prox, "Petroleum" =
reg.petroleum.prox, "Phenol" = reg.phenol.prox, "Mercury" = reg.mercury.prox),
          coef_omit = "year|temp|light|gd",
          coef_rename = c("g_age" = "Governor Age", "s_age" = "Secretary Age", "tpost:prox" = "Proximity
to Boundary x Time Trend", "tpost:prox:g_age" = "Governor Age x Proximity to Boundary",
"tpost:prox:s_age" = "Secretary Age x Proximity to Boundary x Time Trend"),
          stars = c('*' = .1, '**' = 0.05 ,'***' = .01),
          gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE",
          title = "Leader Career Concerns and Pollution Dynamics - Proximity to Boundary",
          statistic = "p.value") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Leader Career Concerns and Pollution Dynamics - Proximity to Boundary						
	COD	**BOD**	**NH**	**Petroleum**	**Phenol**	**Mercury**
Governor Age	-0.140***	-0.067**	-0.003	-0.758*	-0.068*	-0.158**
	(0.002)	(0.031)	(0.829)	(0.051)	(0.054)	(0.020)
Secretary Age	-0.138	0.064	0.005	0.323	0.018*	0.180*
	(0.113)	(0.130)	(0.594)	(0.225)	(0.052)	(0.094)
Proximity to Boundary x Time Trend	-0.203***	-0.187**	-0.022	-0.175	-0.004	0.035
	(0.002)	(0.015)	(0.245)	(0.124)	(0.644)	(0.469)
Governor Age x Proximity to Boundary	0.002***	0.002**	0.000	0.004**	0.000*	0.000
	(0.001)	(0.042)	(0.320)	(0.037)	(0.051)	(0.308)
Secretary Age x Proximity to Boundary x Time Trend	0.001**	0.000	0.000	-0.001	0.000	-0.001
	(0.046)	(0.327)	(0.564)	(0.565)	(0.144)	(0.203)
Num.Obs.	3377	3377	3377	3377	3377	3377
*** p < 0.1, ** p < 0.05, *** p < 0.01**						

Again, the result we observe is only slightly different from what we have got before. We observe that BOD, that is related to COD, follows a similar pattern. Besides that, the coefficient of the interaction between the governor’s age, proximity to boundary and the time trend for the petroleum is positive and significant at the five percent level. The interpretation is the same as for COD and BOD in this and the previous regression.

Summary:

Considering both regression models, COD is the only indicator where in both regression specifications together the coefficient of the interaction between the governor’s age, boundary / proximity to boundary and the time trend is positive and highly significant. Hence, we conclude that there is indeed a stronger pollution progress regarding COD at borders after the policy change came into effect and that furthermore younger governors were more ambitious in reducing COD, the indicator in focus of the central government, but not in the same extent regarding the other indicators. From the regressions this means the following: The modified promotion incentives motivated especially those officials that have longer career horizons, that is younger governors, since they are responsible for achieving the plan goals, rather than secretaries. All indications thus point to the success of the policy change.

Exercise A5 – Triple Difference by Regression

This chapter contains an additional robustness analysis that I performed.

The paper's research is based on the DiD approach, that basically calculates the difference between two differences. Here, it is the difference in average pollution measured at border and non-border monitoring stations, before and after the regime change became effective.

However, it is also possible to add another difference dimension. Thus, it is called the *Triple Difference* approach. The monitoring stations do not only capture measurements for COD, but also for other indicators. Remember, the new environmental regulation only affects COD. As a consequence, other pollution indicators should not have decreased by the same extent as COD at borders after the regime change became effective. The difference between the reduction before and after 2006 of the difference between border and non-border stations of the difference between COD and other pollution indicators is the estimator of the triple difference.

The formula of the triple difference regression looks as follows:

$$\begin{aligned} pol.level_{it} &= ß_1 post06_t + ß_2 boundary_i + ß_3 COD_{it} + ß_3 boundary_i \times post06_t \\ &\quad + ß_4\, post06_t \times COD_{it} + ß_5 boundary_i \times COD_{it} \\ &\quad + ß_6 post06 \times boundary_i \times COD_{it} + u_{it} \end{aligned}$$

Note that we added …

- a dummy variable indicating whether the measurement value refers to COD: COD_{it};
- an interaction term between the post06 and COD dummy: $post06_t \times COD_{it}$;
- an interaction term between the boundary and COD dummy: $boundary_i \times COD_{it}$ and;
- an interaction term between the post06, the boundary and COD dummy: $post06 \times boundary_i \times COD_{it}$

First of all, we need to load the data set:

Task: Check the chunk to load the data set in dat.

```
dat = readRDS("dat.RDS")
```

In contrast to the previous regressions, the dependent variable is a measurement value of one of the five selected pollution indicators. Furthermore, a dummy variable serves as an explanatory variable indicating whether the measurement value does refer to COD or not. However, the scales of the individual indicators differ from each other, which would distort the regression results. Hence, we need to normalize the measurement values to make them comparable. One way is to divide each indicator's set of measurements through their means. We choose to apply the mean from 2005 and multiply the result by 100.

Normalization of the measurement values

We divide the normalization procedure into three steps: *Step 1:* filter the data for observations from 2005 *Step 2:* calculate the means of each indicator *Step 3:* multiply measurement value by 100 and divide by the results from *Step 2*

Let us start with *Step 1* and filter dat for the year 2005.

Task: Replace ___ to filter for 2005 and save the result in dat.2005.

```
# dat.2005 = dat %>%
#   filter(year == ___)

dat.2005 = dat %>%
  filter(year == 2005)
```

Step 2: Now, only observations from 2005 are left in dat.2005. Let us calculate the means of the respective indicators and save the results.

Task: Just check the chunk in order to calculate the means of all water pollution indicators at once.

```
cod.mean = mean(dat.2005$cod)
mercury.mean = mean(dat.2005$mercury)
petroleum.mean = mean(dat.2005$petroleum)
nh.mean = mean(dat.2005$nh)
phenol.mean = mean(dat.2005$phenol)
```

In *Step 3*, we use the results from *Step 2* to normalize the measurements. Hereby, we multiply the original measurements by 100 and divide by the respective means we just calculated.

Task: Replace ___ according to the instructions above to normalize the measurement values.

```
# dat = dat %>%
#   mutate(cod.norm = 100*cod/___) %>%
#   mutate(mercury.norm = 100*mercury/___) %>%
#   mutate(petroleum.norm = 100*petroleum/___) %>%
#   mutate(nh.norm = 100*nh/___) %>%
#   mutate(phenol.norm = 100*phenol/___)

dat = dat %>%
  mutate(cod.norm = 100*cod/cod.mean) %>%
  mutate(mercury.norm = 100*mercury/mercury.mean) %>%
  mutate(petroleum.norm = 100*petroleum/petroleum.mean) %>%
  mutate(nh.norm = 100*nh/nh.mean) %>%
  mutate(phenol.norm = 100*phenol/phenol.mean)
```

Long Format

After normalizing the measurement values, we need to convert the data set into the long format. For the regression, all measurement values need to be in the same column, regardless of the pollution indicator they relate to. Furthermore, we add a column called cod.dummy that is TRUE if the measurement value in value is related to COD.

Task: Just check the chunk in order to convert the data set into the long format and add the COD dummy variable.

```
dat.long = pivot_longer(dat, cols = c("cod.norm", "nh.norm", "phenol.norm", "petroleum.norm",
"mercury.norm")) %>%
  mutate(boundary = as.factor(boundary)) %>%
  mutate(year = as.factor(year)) %>%
  mutate(post06 = as.factor(post06)) %>%
  mutate(name = as.factor(name)) %>%
  mutate(cod.dummy = (name == "cod.norm"))
```

Regression Results

Finally, we are able to run the regression on the data set dat.long and show the results.

Task: Check in order to run the regression and show the results.

```
#run regression
reg = feols(value ~ post06*boundary*cod.dummy + temperature + gdpg + gdpp + lightbuffer5km, cluster =
c("station", "riversystem_time"), ssc = ssc(adj = FALSE, cluster.adj = FALSE, t.df = "conventional"), data =
dat.long)
#display regression
modelsummary(list("Results" = reg), group = model ~ term, coef_omit = "year|temp|light|gd", stars = c('*' =
.1, '**' = 0.05 ,'***' = .01),
             coef_rename = c("post061" = "Post2006", "boundary1" = "Boundary", "cod.dummyTRUE" =
"COD", "post061:boundary1" = "Post2006 x Boundary", "post061:cod.dummyTRUE" = "Post2006 x COD",
"boundary1:cod.dummyTRUE" = "Boundary x COD", "post061:boundary1:cod.dummyTRUE" = "Post2006
x Boundary x COD"),
           gof_omit = "AIC|BIC|Std.Errors|Log.Lik|R2|FE",
           title = "Triple Difference Approach",
           statistic = "p.value") %>%
  kable_classic() %>%
  kable_styling(font_size = 15)
```

Triple Difference Approach

	(Intercept)	Post2006	Boundary	COD	Post2006 x Boundary	Post2006 x COD	Boundary x COD	Post2006 x Boundary x COD
Results	123.082***	-36.027**	33.076	-9.024	9.993	11.134	32.544*	-35.002
	(0.000)	(0.032)	(0.111)	(0.418)	(0.623)	(0.251)	(0.073)	(0.111)
*** p < 0.1, ** p < 0.05, *** p < 0.01**								

In contrast to the other regressions results, p-values are shown here in parentheses. The *Post2006* dummy's coefficient is negative and significant on the five percent level. This would mean that there has been an average overall pollution reduction from 123.082 to 87.055 after 2006 in the interior of the provinces. The result is plausible insofar as the governors are not only aiming at improving water quality at the borders, but also in general and independent of the pollution indicator. On the other hand, COD emissions are correlated with other types of pollution. It is noteworthy that the interaction between *Post2006* and *Boundary* lost its entire explanatory power, while the coefficient of the interaction between *Boundary* and *COD* became significant at the five percent level. That is, COD pollution at borders had been extra-ordinarily compared to other indicators, too. However, the most interesting coefficient is the one referring to the interaction between the *Post2006*, *Boundary* and *COD* dummy variables. Although not being significant, the p-value of 0.111 is also not exorbitantly high, but just missed the ten percent level. Interpreting the coefficient anyway would lead to the following result: While the average water pollution in the interior before 2006 for all water pollution indicators has been 123.082, the pollution for COD at borders after 2006 only decreased by an additional amount of 35.002. That is, COD pollution decreased stronger than other pollution indicators and this would be consistent with the results the researchers achieved.

Summary

To contribute an additional robustness check, I decided to apply the *Triple Difference* approach by including the type of pollution indicator as a third dimension into the DiD regression. Nonetheless, although the coefficient is negative and quite large, the result is not significant with a p-value of 0.111. What we can say, however, is that the result does not contradict the authors' findings, but it is rather a bit of support.

Declaration of Honor

I hereby confirm on my honor that I personally prepared the present academic work and carried out myself the activities directly involved with it. I also confirm that I have used no resources other than those declared. All formulations and concepts adopted literally or in their essential content from printed, unprinted or Internet sources have been cited according to the rules for academic work and identified by means of footnotes or other precise indications of source.

The support provided during the work, including significant assistance from my supervisor has been indicated in full.

The academic work has not been submitted to any other examination office authority. The work is submitted in printed and electronic form. I confirm that the content of the digital version is completely identical to that of the printed version.

Date: 30.03.2022 Signature: ______________________________