Do lockdown policies reduce AQI?

Replication of He et al., 2020.

Jiahuan Li, Yifei Lu, Jiangtian Guan, Jack Green, and Tilly Yao

Abstract

- A recent study suggests that China's lockdown policies to prevent COVID-19 transmission escalation directly cause air quality improvement, especially in colder and northern cities. To examine this causal relationship, we replicate the authors' data analyses. Having reproduced their main result, we also discover irregularities in their data processing. In addition, we analyze the factors the authors overlooked that could have impacted the Air Quality Index (AQI), including location and China's restrictive regulations on the use of coal.
- We have two main findings from replication and extended analyses.
    - First, it takes time for lockdown policies to kick in. Studies suggest that air circulation takes about four days to reduce pollutants. However, AQI reduction was only significant after two weeks in our target study.
    - Second, exceptions to the aforementioned causal relationship are present. Hubei Province implemented the strictest lockdown policies but only achieved slight reductions in AQI, while northeastern provinces with lenient policies achieved disproportionately significant reductions.

Background

- Air pollution is a pressing environmental and public health challenge facing our society (Bruce *et al.*, 2000). While decades of research on air pollution control have made significant progress, lockdown policies that emerged during this global pandemic bring new perspectives (Goolsbee & Silverson, 2020; Shalal & Crossley, 2020; Schwela *et al.*, 2012). Studies around the world, including those in Indian, the United States, and European countries confirm large decreases in air pollutant concentrations in major cities under lockdown measures (European Environmental Agency, 2020; Holcombe & O'Key, 2020; Mahato et al., 2020). The European Environmental Agency (2020) attributes such decreases to the reduced road traffic/transportation, which might be affected by the reduced economic activities.
- Inspired by literature, we hypothesize that air quality may also be improved during lockdowns in China. To test our hypothesis, we replicate the data analyses in a recent paper published on Nature Sustainability (He *et al.* 2020). The research question of this paper is: what are the short-term impacts of COVID-19 lockdown on urban air pollution? The authors found lockdown policies have a direct causal relationship with the improvement in urban air quality in China, especially in colder and northern cities (He *et al.* 2020).

Method

Data source and pre-analysis

- The original data includes the following four parts: air quality data (Ministry of Ecology and Environment), Weather data (NOAA), local governments' lockdown information (Wikipedia page), and cities' socio-economic status (the 2017 China City Statistical Yearbook). The author creates the city-by-day level data by calculating the distance from a center to all monitoring stations within the corresponding city and then aggregates station-level data to city-level data using the inverse distance weights.
- Based on the processed data, we do a data pre-analysis. We need a parallel trend test to confirm if changes are caused by the lockdown policy before replicating the DiD model. Due to city code encryption protection, we match it by comparing the GDP in the 2017 China City Statistical Yearbook to the specific city. Then we organize the data into weekly data according to the lead and lag lockdown policy.
- All the details of our replication work are recorded in the replication section.

DiD model

- The DiD model allows us to control various confounding factors that may affect air pollution levels and determine the reasonable causal effects of virus containment measures. In the baseline regression, we use the following model to estimate the relative change in air pollution levels between the treated city and the control city. Treat city is the lockdown city and control city on the contrary.

$$Y_{it} = 1[city\ lockdown]_{it} \times \beta + X_{it} \times \alpha + \mu_i + \pi_t + \epsilon_{it} \tag{1}$$

- Where,
  - Y_{it} : the level of air pollution in city i on date t.
  - $1[city\ lockdown]_{it}$: whether a lockdown is enforced in city i on date . It takes the value 1 if the city is locked down and 0 otherwise.
  - X_{it} : control variables, including temperature, temperature squared, precipitation, and snow depth.
  - μ_i : city fixed effects. City fixed effects are a set of city-specific dummy variables that can control the time-invariant confounding factors specific to each city. For example, by introducing urban fixed effects, the geographic conditions, short-term industrial economic structure, income and natural endowments of the city can be controlled.
  - π_t : date fixed effects. The date fixed effect is a set of dummy variables used to explain the shocks common to all cities on a given day, such as national holiday policies, macroeconomic conditions, and changes in national air pollution over time.
- Since the regression includes both the location fixation effect and the time fixation effect, the coefficient β estimates the difference in air pollution between treatment cities and control cities before and after implementing the city lockdown policy. Due to the spillover effect, β measures the relative effect of the city lockdown on air pollution between the two groups of cities rather than the absolute impact.

Replication

Pre-analysis Preparation

1. Convert .dta File to .csv

- Problem: some group members have failed to open the .dta files in Rstudio.
- The read function in R cannot read a Stata version 5-12 .dta file.

```python
import pandas as pd
import os

path_0 = r"C:\Users\Lijh\Desktop\statistics\project\data\test data"  # path of original files (folder)
path_1 = r"C:\Users\Lijh\Desktop\statistics\project\data\csv data"  # path of restoration
filelist = os.listdir(path_0)  # the list of files under the folder

f1 = r'C:\Users\Lijh\Desktop\statistics\project\data\test data\city_yb.dta'
file = pd.read_stata(f1)
print(file)
csv_file = r'C:\Users\Lijh\Desktop\statistics\project\data\csv data\city_yb.csv'
file.to_csv(csv_file)
```

2. Decode the City code

Problem

- We do not know the meaning of the field ‘city code’, which is important for our further discussion.
- We have written to the author of this paper but he did not answer our question.

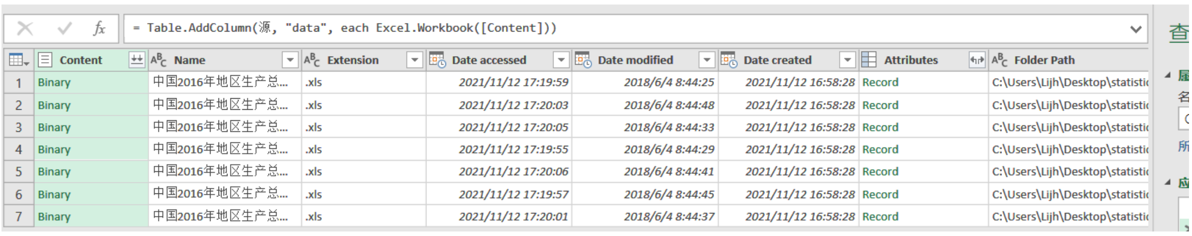
Solution

- Use other datasets with the information of the cities to decode the data of this paper.

Merging plenty of the Excel files (Power Query)

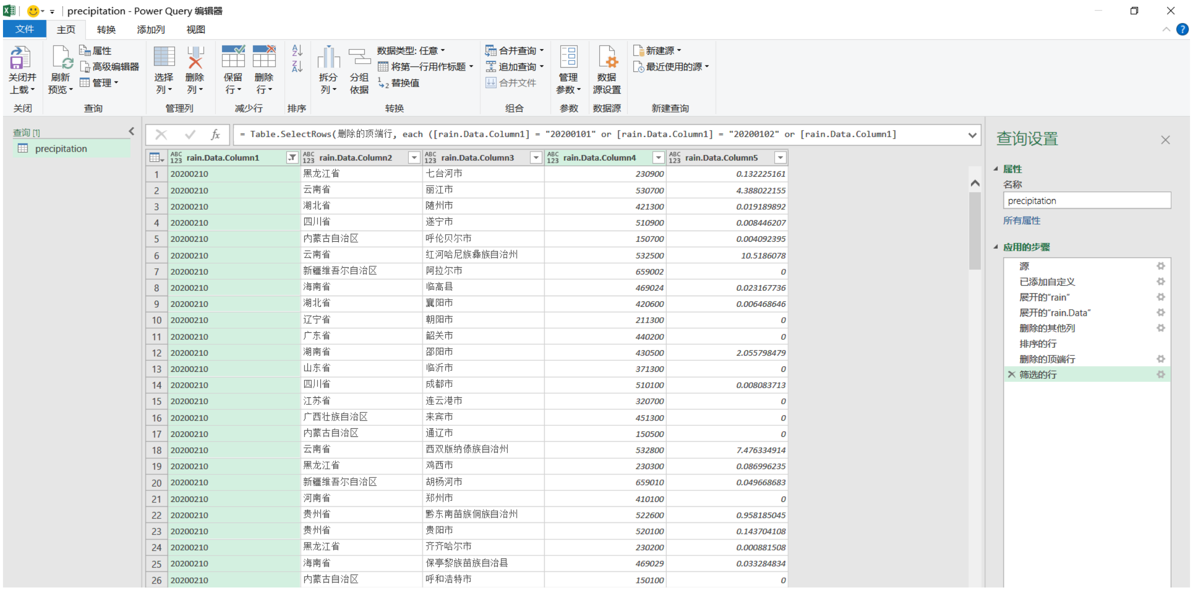
- The advanced tool developed and integrated in Excel is especially helpful.

Fig 1. Interface of the operation windows of Power Query



= Table.AddColumn(源, "data", each Excel.Workbook([Content]))

	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:19:59	2018/6/4 8:44:25	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
2	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:20:03	2018/6/4 8:44:48	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
3	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:20:05	2018/6/4 8:44:33	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
4	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:19:55	2018/6/4 8:44:29	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
5	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:20:06	2018/6/4 8:44:41	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
6	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:19:57	2018/6/4 8:44:45	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti
7	Binary	中国2016年地区生产总...	.xls	2021/11/12 17:20:01	2018/6/4 8:44:37	2021/11/12 16:58:28	Record	C:\Users\Lijh\Desktop\statisti



- **Code of the operation**

```
= Folder.Files("C:\Users\Lijh\Desktop\statistics\project\city code\rainfall data\precipitation")
= Table.AddColumn(源, "rain", each Excel.Workbook([Content])) ## Open the content of each files
= Table.ExpandTableColumn(已添加自定义, "rain", {"Name", "Data", "Item", "Kind", "Hidden"},
= Table.ExpandTableColumn(#"展开的“rain”", "rain.Data", {"Column1", "Column2", "Column3", "Column4", "Column5"},
= Table.SelectColumns(#"展开的“rain.Data”",{"          ## Expanding the content
= Table.Sort(删除的其他列,{{"rain.Data.Column1", Order.Descending}})  ## Deleting other useless information
= Table.Skip(排序的行,370)
= Table.SelectRows(删除的顶端行, each Text.StartsWith([rain.Data.Column1], "2019") or Text.EndsWith([rain.Data.Column1], "200314"))       ## Data selection
= Table.Sort(筛选的行,{{"rain.Data.Column1", Order.Ascending}})
```

Precipitation or GDP

- We begin with the hope to match the data with other weather dataset. Unluckily, the results did not match at all!

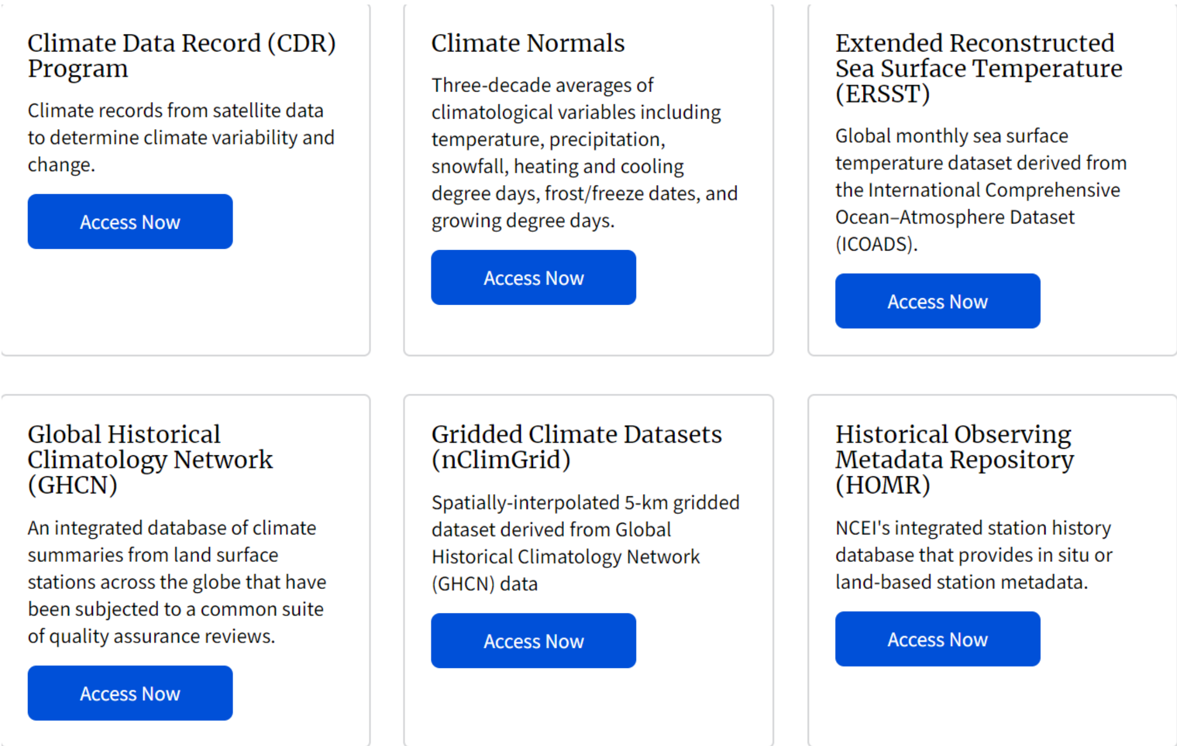
Fig 2. Results display (the precipitation data of cities in Hubei Province on 2019.01.08)

	A	B	C	D	E
	date	prec	hubei	north	city_code
09	20190108	71.10404904	1	0	62661
34	20190108	68.61392805	1	0	44739
27	20190108	66.81590902	1	0	91051
47	20190108	66.36756978	1	0	99238
19	20190108	62.75631322	1	0	55521
79	20190108	59.75485271	1	0	81523
34	20190108	57.17490908	1	0	100430
80	20190108	47.75761757	1	0	36764
27	20190108	47.09848866	1	0	39466
48	20190108	38.02981829	1	0	70859
90	20190108	37.7064739	1	0	16871
28	20190108	34.39166252	1	0	35583
41	20190108	30.91534147	1	0	47088
33	20190108	18.0592261	1	0	54348
53	20190108	5.647376259	1	0	19482
72					

rain.Data.Column1	rain.Data.Column2	rain.Data.Column3	rain.Data.Column4	rain.Data.Column5
20190108	湖北省	黄石市	420200	6.470420268
20190108	湖北省	咸宁市	421200	5.583927596
20190108	湖北省	天门市	429006	5.285030842
20190108	湖北省	鄂州市	420700	4.836530175
20190108	湖北省	仙桃市	429004	4.123194739
20190108	湖北省	武汉市	420100	3.917989268
20190108	湖北省	潜江市	429005	3.785714102
20190108	湖北省	黄冈市	421100	3.646492487
20190108	湖北省	荆州市	421000	3.28750483
20190108	湖北省	孝感市	420900	3.260554862
20190108	湖北省	荆门市	420800	2.040794233
20190108	湖北省	宜昌市	420500	1.063040235
20190108	湖北省	恩施土家族苗族自治州	422800	0.89394192
20190108	湖北省	随州市	421300	0.413395565
20190108	湖北省	襄阳市	420600	0.129905459
20190108	湖北省	神农架林区	429021	0.027924084
20190108	湖北省	十堰市	420300	0.00227192

- Then we examined the data sources and found that not only the data is from an international dataset, but also it is a result of complex calculation (the inverse distance weights).

Fig 3. Confusing weather datasets on the Global Historical Climatology Network (GHCN) webset



- Therefore, we decided to turn to the GDP data, which clearly comes from the **2017 China City Statistic Yearbook**.

Table 1. Part of the reference data of GDP

城市	地区生产总值(当年价格)(万元)		人均地区生产总值(元)		地区生产总值增长率(%)	
	全市	市辖区	全市	市辖区	全市	市辖区
北京市	256691300	256691300	118198	118198	6.80	6.80
天津市	178853900	178853900	115053	115053	9.10	9.10
石家庄市	59277293	32148250	55177	67493	6.80	7.30
唐山市	63548675	33238336	81239	93110	6.80	7.40
秦皇岛市	13493526	9340031	73755	56805	7.00	7.40
邯郸市	33370903	13663695	35265	38365	6.08	5.99
邢台市	19757460	3143341	27038	33372	7.10	6.70
保定市	34771269	11416785	29992	40087	7.20	5.40

Data Matching in Excel

- Simply use the command of sorting to sort the GDPs data in the same order.
- Use the command of conditional format filtering to illustrated the **unique value** of column A & D with the **orange color**, which represents the missing of the cities' data.
- The **missing value** issue also exists, as shown in **A 286 - A 288**.

Fig 4. Result of GDP matching

	A	B	C	D
1	gdp_city	city_code	城市	地区生产总值(当年价格)(万元)
2	281786500	93997	上海市	281786500
3	256691300	54613	北京市	256691300
4	195474420	36093	广州市	195474420
5	194926012	53747	深圳市	194926012
6	178853900	6124	天津市	178853900
7	177405900	5993	重庆市	177405900
8	154750900	64738	苏州市	154750900
9	121702335	80196	成都市	121702335
10	119126100	55521	武汉市	119126100
11	113137223	33655	杭州市	113137223
12	105030200	44998	南京市	105030200
13	100112900	39012	青岛市	100112900
14	93569088	84786	长沙市	93569088
15	92100200	93779	无锡市	92100200
16	86864911	70802	宁波市	86864911
17	86300002	84914	佛山市	86300002
18	81139666	19948	郑州市	81139666
19	69256587	99066	烟台市	69256587
20	68276868	92478	东莞市	68276868
21	68101998	90724	大连市	68101998
22	67682000	73969	南通市	67682000
23	66466294	85766	泉州市	66466294
24	65361165	78395	济南市	65361165
25	63548675	20585	唐山市	63548675
26	62743777	67987	合肥市	62743777
27	62571800	29636	西安市	62571800
28	61976395	80521	福州市	61976395
29	61016096	2414	哈尔滨市	61016096
30	59864200	100304	长春市	59864200
282	2166414	65987	七台河市	2166414
283	2078152	38294	金昌市	2078152
284	1877546	11788	日喀则市	1877546
285	1534089	81617	嘉峪关市	1534089
286		80814	泰州市	41017800
287		45983	宿迁市	23511200
288		39902	营口市	11562477
289			崇左市	7662005
290			来宾市	5891105
291	This are unmatched values		中卫市	3391289
292			儋州市	2577835

Translation of the cities' names

- command in Excel

`=FILTERXML(WEBSERVICE("http://fanyi.youdao.com/translate?&i="&A1&"&doctype=xml&version"),"//translation")`

- **translation results** shown in column C, which is not very accurate.

Fig 5. Result of the translation of the city names to English version

C5 =FILTERXML(WEBSERVICE("http://fanyi.youdao.com/translate?&i="&B5&"&doctype=xml&version"),"//translation")

	A	B	C
1	city_code	城市	city
2	93997	上海市	Shanghai
3	54613	北京市	Beijing
4	36093	广州市	guangzhou
5	53747	深圳市	shenzhen
6	6124	天津市	tianjin
7	5993	重庆市	chongqing
8	64738	苏州市	suzhou
9	80196	成都市	chengdu
10	55521	武汉市	Wuhan city
11	33655	杭州市	hangzhou
12	44998	南京市	nanjing
13	39012	青岛市	Qingdao
14	84786	长沙市	Changsha city
15	93779	无锡市	Wuxi city
16	70802	宁波市	ningbo
17	84914	佛山市	Foshan city,
18	19948	郑州市	Zhengzhou city
19	99066	烟台市	yantai
20	92478	东莞市	Dongguan city,

Data Evaluation

- The data of the social and economic fields may be too **old**. The paper published in 2020 but used the data in yearbook of 2017, which summarizes the GDP and pGDP in 2016.
- One of our group member is a native of Hubei province, She knows shennongjia “神农架” and Enshi “恩施” exist in Hubei province, but the GDP and other data of them are not included in the national statistical yearbook, so the author's paper did not includ them either. She thinks that is strange.

Fig 6. Screenshot of the yearbook

湖北省	**Hubei**						
武汉市	Wuhan	119126100	96306028	111469	125463	7.80	8.00
黄石市	Huangshi	13055500	6166200	53033	69459	7.20	7.40
十堰市	Shiyan	14291500	9334567	42083	68048	8.90	10.14
宜昌市	Yichang	37093600	15930190	89978	108701	8.80	8.20
襄阳市	Xiangyang	36945100	18588198	65663	80691	8.53	7.91
鄂州市	Ezhou	7978200	7978200	74983	74983	8.00	8.00
荆门市	Jingmen	15210000	5138400	52470	74551	8.50	8.30
孝感市	Xiaogan	15766900	2877351	32236	31128	7.90	8.00
荆州市	Jingzhou	17267500	5699800	30305	46163	7.30	8.40
黄冈市	Huanggang	17261700	2031609	27373	52280	7.60	7.00
咸宁市	Xianning	11079300	2591100	44027	439321	7.60	8.70
随州市	Suizhou	8521800	3930900	38801	62217	8.00	8.20

3. Matching city with province

- Three days after we decoded the city_code using GDP data from China City Statistical Yearbook 2017, the author replied the email with the attachment "city_id_and_code.dta" and "city_list.dta".

```r
library(haven)
city_id_and_code <- read_dta("Desktop/city_id_and_code.dta")
View(city_id_and_code)
```

```r
library(haven)
city_list <- read_dta("Desktop/city_list.dta")
View(city_list)
```

Fig 7. Screenshots of the city_code and city_list

	city_code2010 city_code2010	city_code
1	2301	2414
2	2105	3547
3	3402	4870
4	3412	4956
5	5000	5993
6	3405	5996
7	1200	6124
8	1308	6145
9	5107	6270
10	1506	7765
11	6109	8027
12	2113	8131
13	1409	8615
14	3710	8775
15	5305	9121
16	2112	9715

	city_code2010 city_code2010	city_name2010 NL_NAME_2
1	1100	北京|北京
2	1200	天津|天津
3	1301	石家庄市
4	1302	唐山市
5	1303	秦皇岛市
6	1304	邯郸市
7	1305	邢台市
8	1306	保定市
9	1307	张家口市
10	1308	承德市
11	1309	沧州市
12	1310	廊坊市
13	1311	衡水市
14	1401	太原市
15	1402	大同市
16	1403	阳泉市

- The authors did some masking work as we noticed.
- We found the cities which were the missing values when we decoding the city_code using GDP data by Excel. In addition, some cities are not included in the China City Statistical Yearbook 2017. Most of them are minority autonomous prefectures, such as Yushu Tibetan Autonomous Prefecture in Qinghai province.
- We use the old ISO-3166 subdivision code to code the province.

Fig 8. Screenshot of the province_code

city_code	city	province_coc	province
2414	Harbin	230	Heilongjiang
3547	Benxi	210	Liaoning
4870	Wuhu	340	Anhui
4956	Fuyang	340	Anhui
5993	Chongqing	500	Chongqing
5996	Maanshan	340	Anhui
6124	Tianjin	120	Tianjin
6145	Chengde	130	Hebei
6270	Mianyang	510	Sichuan
7765	Ordos	150	Nei Mongol
8027	Ankang	610	Shaanxi
8131	Chaoyang	210	Liaoning
8615	Xinzhou	140	Shanxi
8775	Weihai	370	Shandong
9121	Baoshan	530	Yunnan
9715	Tieling	210	Liaoning

- **Reflection**: We did the correction and matching process by hand using Excel. Therefore, ther are some spelling mistakes. Fortunately, it does not influence the latter exploration. There should be more efficient ways.
- We matched the "city" "province_code" "province" with the original data "wf" using "for loop" in R.

```
# read wf.csv and make subset of wf.csv to select 60 days from 20200101 to 20200301
wf <- read.csv("/Users/tilly/Desktop/wf.csv")
newdata <- subset(wf, daynum >= 8401 & daynum <=8461)

# province code
# created and edited in the Excel
# also adjusted the error in the translation of city
# saved as "province_code_reference.csv"
province_data = read.csv("/Users/tilly/Desktop/province_code_reference.csv")

# city & province_code & province
for (m in 1:20130){
  if (is.na(newdata$city_code[m])){
    next
  }
  for (n in 1:330){
    if (newdata$city_code[m]==province_data$city_code[n]){
      newdata$city[m] = province_data$city[n]
      newdata$province_code[m] = province_data$province_code[n]
      newdata$province[m] = province_data$province[n]
    }
  }
}
write.csv(newdata, file = ('wf_city_province.csv'))
```

Fig 9. Screenshot of the data matched with the province code

	day	month	year	daynum	north	city_code	city	province_code	province
0	1	1	2020	8401	1	54613	Beijing	110	Beijing
0	1	1	2020	8401	1	6124	Tianjin	120	Tianjin
0	1	1	2020	8401	1	71005	Shijiangzhuang	130	Hebei
0	1	1	2020	8401	1	20585	Tangshan	130	Hebei
0	1	1	2020	8401	1	41928	Qinhuangdao	130	Hebei
0	1	1	2020	8401	1	93580	Handan	130	Hebei
0	1	1	2020	8401	1	50965	Xingtai	130	Hebei
0	1	1	2020	8401	1	98240	Baoding	130	Hebei
0	1	1	2020	8401	1	94313	Zhangjiakou	130	Hebei
0	1	1	2020	8401	1	6145	Chengde	130	Hebei
0	1	1	2020	8401	1	39753	Cangzhou	130	Hebei
0	1	1	2020	8401	1	77331	Langfang	130	Hebei
0	1	1	2020	8401	1	97330	Hengshui	130	Hebei
0	1	1	2020	8401	1	50167	Taiyuan	140	Shanxi
0	1	1	2020	8401	1	66511	Datong	140	Shanxi
0	1	1	2020	8401	1	22294	Yangquan	140	Shanxi
0	1	1	2020	8401	1	85903	Changzhi	140	Shanxi
0	1	1	2020	8401	1	14529	Jincheng	140	Shanxi
0	1	1	2020	8401	1	79806	Suzhou	140	Shanxi
0	1	1	2020	8401	1	97735	Jinzhong	140	Shanxi

4. Aggregate Data to Week Level

Problem

- The picture of the main findings shows the results aggregated at the week level.
- We also need to modify the data and include the week dummies in the analysis. The dummies indicate **the number of lead and lag weeks** of the start of the city lock down for every record.
- The definition of the week dummy should be the same as in the original paper, which we got wrong at the first time.

Fig 10. Wrong understanding of the week dummy

date	aqi	treat	hubei	day	month	year	daynum	north	city_code	judge	week
20200106	77.08334	0	0	6	1	2020	8406	1	54613	1	-5
20200113	22.375	0	0	13	1	2020	8413	1	54613	1	-4
20200120	27.83333	0	0	20	1	2020	8420	1	54613	1	-3
20200127	203.125	0	0	27	1	2020	8427	1	54613	1	-2
20200203	31.45833	0	0	3	2	2020	8434	1	54613	1	-1
20200210	161	1	0	10	2	2020	8441	1	54613	1	0
20200217	22.5	1	0	17	2	2020	8448	1	54613	1	1
20200224	87.875	1	0	24	2	2020	8455	1	54613	1	2

Fig 11. Correct understanding of week dummies

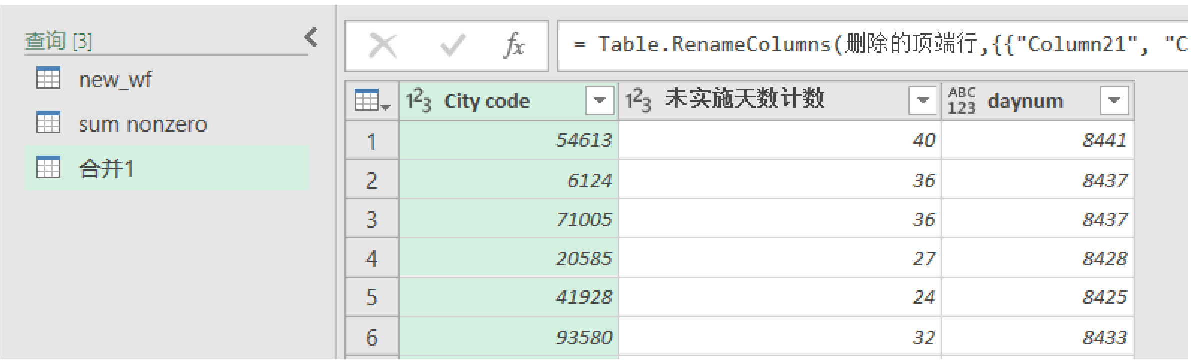
date	aqi	treat	daynum	city_code	w4_lead	w3_lead	w2_lead	w1_lead	w0	w1	w2	w3	w4
20200118	191.3333	0	8418	54613	1	0	0	0	0	0	0	0	0
20200119	61.95833	0	8419	54613	1	0	0	0	0	0	0	0	0
20200120	27.83333	0	8420	54613	0	1	0	0	0	0	0	0	0
20200121	59.5	0	8421	54613	0	1	0	0	0	0	0	0	0
20200122	90.94737	0	8422	54613	0	1	0	0	0	0	0	0	0
20200123	74	0	8423	54613	0	1	0	0	0	0	0	0	0
20200124	103.4167	0	8424	54613	0	1	0	0	0	0	0	0	0
20200125	182.9583	0	8425	54613	0	1	0	0	0	0	0	0	0
20200126	190.6667	0	8426	54613	0	1	0	0	0	0	0	0	0
20200127	203.125	0	8427	54613	0	0	1	0	0	0	0	0	0
20200128	212.875	0	8428	54613	0	0	1	0	0	0	0	0	0
20200129	97.70834	0	8429	54613	0	0	1	0	0	0	0	0	0

Solution

The Lockdown Cities & Dates (Power Query)

- Firstly, we deleted the records with the NA values in the ‘city_code’ field.
- Then, we grouped all the records according to the ‘city_code’ field and summarized the sum of all the values in the ‘treat’ field of each group.
- Next, the groups were deleted if their sum of the values in ‘treat’ field equals to zero. That is, these cities had not been locked down (treated) in the time frame of the study. Through this step, we could find the city codes of 95 cities (of 330 in total) which had been locked down.
- At last, we created a new field named ‘daynum’ to restore the dates of lockdown of the 95 cities. The dates were got by adding the sum of the rows with ‘treat = 0’ to the baseline day number 8401 (the number indicating January 1).

Fig 12. Calculation of the dates of the lockdown



	City code	未实施天数计数	daynum
1	54613	40	8441
2	6124	36	8437
3	71005	36	8437
4	20585	27	8428
5	41928	24	8425
6	93580	32	8433

The Lead & Lag Week Dummies (R)

Fig 13. Screenshot of the modified dataset with week dummies

daynum	north	city_code	period_sum	lock_date	lock_sum	w4_lead	w3_lead	w2_lead	w1_lead	w0	w1	w2	w3	w4
8401	1	54613	1	41	-40	1	0	0	0	0	0	0	0	0
8402	1	54613	2	41	-39	1	0	0	0	0	0	0	0	0
8403	1	54613	3	41	-38	1	0	0	0	0	0	0	0	0
8404	1	54613	4	41	-37	1	0	0	0	0	0	0	0	0
8405	1	54613	5	41	-36	1	0	0	0	0	0	0	0	0

```r
newwfdata =
read.csv("C:/Users/Lijh/Desktop/statistics/project/data/csv/new_wf.csv")
weekcode =
read.csv("C:/Users/Lijh/Desktop/statistics/project/data/week_code.csv")

### data pre-analysis:add the fields of lead & lag dummies

newwfdata$period_sum = 0

# baseline value: if the city is in control group, values of these two
fields will be zero.
newwfdata$lock_date = 0
newwfdata$lock_sum = 0

i = 1
m = 1
## i,m = row numbers in the datasets 'newwfdata' & 'weekcode'

## baseline value: for January 1 to March 1, the period_sum will be 1 to
61.
for (m in 1:20130) {
  if (is.na(newwfdata$city_code[m])) {
```

```r
    next
  }
  newwfdata$period_sum = newwfdata$daynum - 8400
}

## only 95 cities of 330 have been locked down and the 'lock_date' of them have been recorded in dataset 'weekcode'. Then these dates are subtracted by 8400 to facilitate subsequent calculations.
for (m in 1:20130) {
  if (is.na(newwfdata$city_code[m])) {
    next
  }
  for (i in 1:95) {
    if (newwfdata$city_code[m] == weekcode$City_code[i]) {
      newwfdata$lock_date[m] = weekcode[i, 2] - 8400

    }
  }
}

## calculate the order of the days related to the lock date of each city
for (m in 1:20130) {
  for (i in 1:95) {
    if (newwfdata$city_code[m] == weekcode$City_code[i]) {
      newwfdata$lock_sum[m] = newwfdata$period_sum[m] - newwfdata$lock_date[m]

    }
  }
}

## calculate all the lead and lag dummies of each weeks
newwfdata$w4_lead = 0
newwfdata$w3_lead = 0
newwfdata$w2_lead = 0
newwfdata$w1_lead = 0
newwfdata$w0 = 0
newwfdata$w1 = 0
newwfdata$w2 = 0
newwfdata$w3 = 0
newwfdata$w4 = 0

## calculate the differences between the fields 'period_sum' & 'lock_sum' and compare it with the multiples of 7 to decide which week dummy of a certain record will get the value of 1.
for (m in 1:20130) {
  for (i in 1:95) {
    if (newwfdata$city_code[m] == weekcode$City_code[i]) {
      if (newwfdata$lock_sum[m] < -21) {
        newwfdata$w4_lead[m] = 1
      }
      if (-21 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= -15) {
        newwfdata$w3_lead[m] = 1
      }
      if (-14 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= -8) {
        newwfdata$w2_lead[m] = 1
```

```
      }
      if (-7 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= -1) {
        newwfdata$w1_lead[m] = 1
      }
      if (0 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= 6) {
        newwfdata$w0[m] = 1
      }
      if (7 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= 13) {
        newwfdata$w1[m] = 1
      }
      if (14 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= 20) {
        newwfdata$w2[m] = 1
      }
      if (21 <= newwfdata$lock_sum[m] &&
          newwfdata$lock_sum[m] <= 27) {
        newwfdata$w3[m] = 1
      }
      if (28 <= newwfdata$lock_sum[m]) {
        newwfdata$w4[m] = 1
      }
    }
  }
}

# save the modified dataset
write.table(
  newwfdata,
  "corrected_weekly_data.csv",
  row.names = FALSE,
  col.names = TRUE,
  sep = ","
)
```

Replication of the Author’s Result

Replication process & result

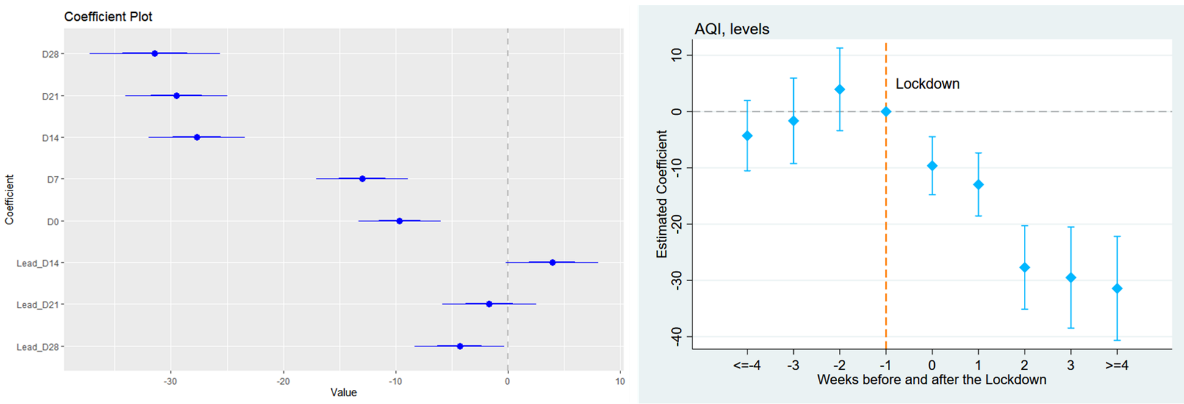
- Using R programming, we can replicate the Fig. 3 in the paper.

```
library(haven)

# use the original dataset generated through the authors' code
wf_stata_generated <- read.dta("C:\Users\Lijh\Desktop\statistics\project\data\wf_stata_generated.dta")

temp_2 = wf_stata_generated$temp * wf_stata_generated$temp

# the author's method (delete treat, entire dataset, delete Lead_D7 manually)
didreg_week = lm(
  aqi ~ temp + temp_2 + prec + snow
  + Lead_D28 + Lead_D21 + Lead_D14 + D0 + D7 + D14 + D21 + D28 + base
```

```
  + as.factor(city_code) + as.factor(daynum),
  data = wf_stata_generated
)
summary(didreg_week)

# visualization of the result
library(coefplot)
coefplot(
  didreg_week,
  predictors = c(
    "D0",
    "D7",
    "D14",
    "D21",
    "D28",
    "Lead_D14",
    "Lead_D21",
    "Lead_D28",
    "base"
  )
)
```

- Through the above process, we could generate the graph with the same features as the image in the article.

Fig 14. The replicated result (left) and the original result (right)



Evaluation — the lag of the lock down effect on air quality

- As we can see from both graphs, although the significant decrease in the air quality index (AQI) happened immediately after the lock down, it took **2 weeks** for the decrease to become stable and more significant.
- However, this lag of effect does not coincide with the common knowledge of the air circulation and refreshment. It usually takes **1 day to 1 week** for the convection of the pollutants from the atmospheric boundary layer to the free atmosphere. Therefore, the more significant and stable effect of the lock down is expected to appear in an **earlier** stage.

Problem — multicollinearity

We found the multicollinearity problem need to be considered during the replication process.

- **Definition**: multicollinearity (also collinearity) is a phenomenon in which one independent variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy. In this situation, the coefficient estimates of the multiple regression may change erratically in response to small changes in the model or the data.

Solution

- To avoid the multicollinearity problem caused by the introduction of the week dummies, we need to make some adjustments to the typical DiD model:

  1. Delete the independent variable 'treat'

     The multiple fields of week dummies are actually the **mutually independent subsets** of the field 'treat'. Therefore, 'treat', the 'sum' field, must be strongly collinear with all the week dummies.

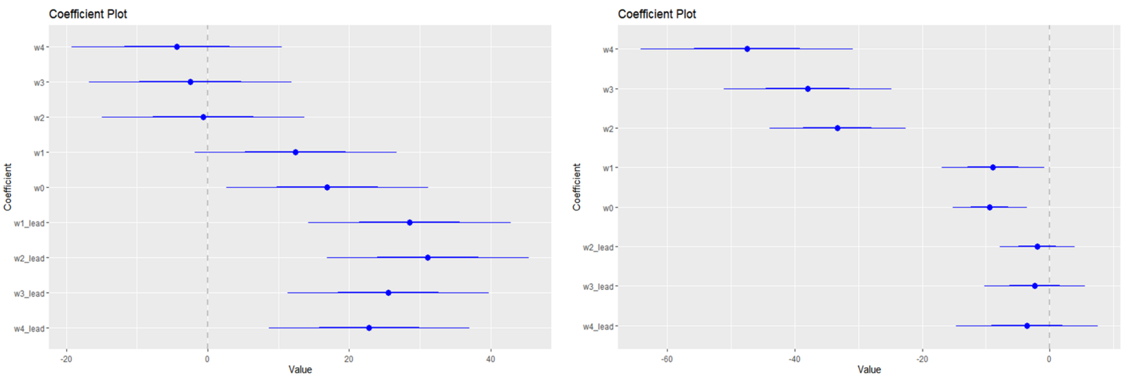
  2. Omit one of the week dummies

     The time frame of the study (2 months) is divided into 9 parts (roughly 9 weeks before and after the lock down policy). So, only 8 week dummies are needed in the regression. If we use 9 dummies, the collinearity problem will definitely happen. In this case, we'd better **omit the dummy of the lock down eve week** (w1_lead) and make it serve as the **base**. If we do not do that, the computer will automatically omit one dummy because the multicollinearity problem happens.

  3. Omit the data from the control group (cities without lock down policy)

     If the lock down policy does not exist, the concept of weeks before and after the lock down does not exist, either. Therefore, there is **no appropriate value** for the week dummies of the records from the control group. If we include these records into the regression, they will be identified as the wrong base and disturb the result.

Fig 15. Our result using the entire dataset (left) vs. using the data of the treated group (right)

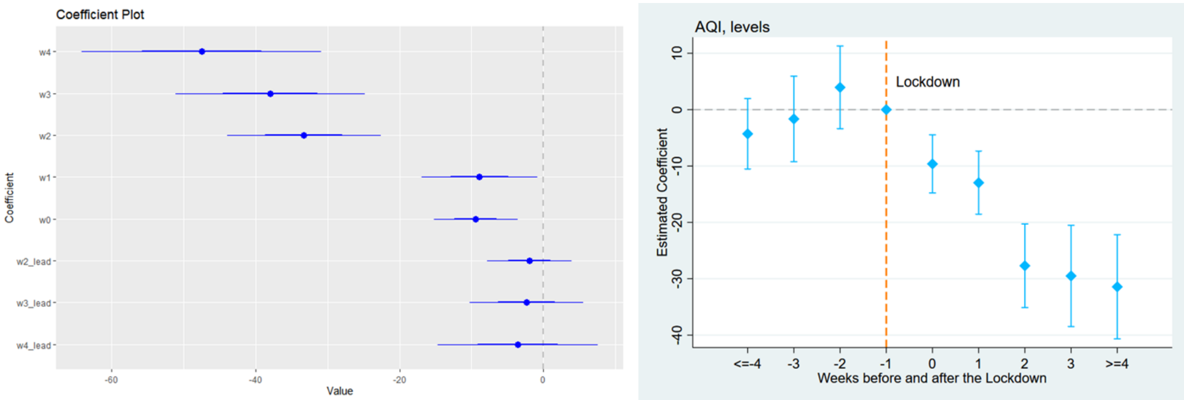


- After these adjustments, we finally get our result of this analysis.

```
# corrected finding (delete 'treat', lockdown cities only, delete 'w1_lead')

week_data <- read.csv("C:/Users/Lijh/Desktop/statistics/project/data/csv/corrected_weekly_data.csv")
week_treated <- subset(week_data, t_asign == 1)
temp_2 = week_treated$temp * week_treated$temp

didreg_week1 = lm(
  aqi ~ temp + temp_2 + prec + snow
  + w4_lead + w3_lead + w2_lead + w0 + w1 + w2 + w3 + w4
  + as.factor(city_code) + as.factor(daynum),
  data = week_treated
)
summary(didreg_week1)
```

```r
# Note that the multicollinearity should appear and can be proved if the 'w1_lead' is added into the above formula. If so, we can find no estimate coefficient (NA) for 'w4', which is exactly the dummy automatically deleted by the computer.

library(coefplot)
coefplot(
  didreg_week1,
  predictors = c(
    "w4_lead",
    "w3_lead",
    "w2_lead",
    "w0",
    "w1",
    "w2",
    "w3",
    "w4"
  )
)
```

Fig 16. Our result (left) and the original result (right)



Doubts in the authors' result

1. Irregularity of the authors' dataset
   - For the treated group, the sum of all the week dummies of each record should be **1**. That is, every certain day should only belongs to **one** certain week before or after the date of lock down. However, we can observe some irregularities in the original dataset generated by the authors' code.
   - Therefore, we generate our own dataset to produce our regression result using the method mentioned in the pre-analysis section. The irregularity problem has not been found in our dataset.

Fig 17. proofs of the data irregularity in the original dataset from two aspects

	temp2	l_aqi	l_pm	t_sum	D0	D7	D14	D21	D28	Lead_D7	Lead_D14	Lead_D21	Lead_D28	base
41	143.5367	4.943427	4.489573	7	1	0	0	0	0	0	0	0	0	0
42	26.0944	4.471639	4.39342	8	1	1	0	0	0	0	0	0	0	0
43	1.733874	4.668536	4.301246	9	1	0	0	0	0	0	0	0	0	0
44	33.67875	4.50673	4.455316	10	0	1	0	0	0	0	0	0	0	0

```
> table(stata_treated$Lead_D28 + stata_treated$Lead_D21 + stata_treated$Lead_D14 + stata_treated$Lead_D7 + stata_treated$D0 +
 stata_treated$D7 + stata_treated$D14 + stata_treated$D21 + stata_treated$D28)

   1    2
5701   94
```

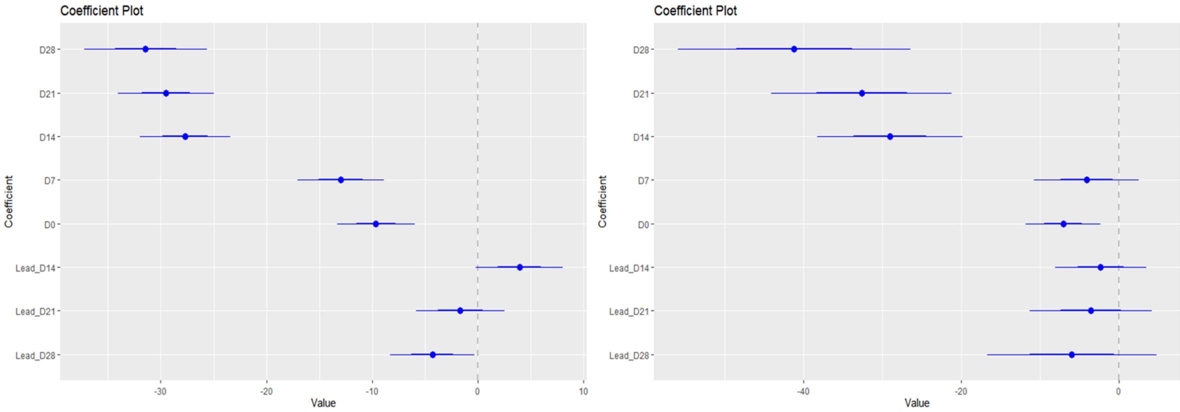
Fig 18. proof of the validity of the data generated by us

```
> week_data <- read.csv("C:/Users/Lijh/Desktop/statistics/project/data/csv/corrected_weekly_data.csv")
> week_treated <- subset(week_data, daynum >= 8401 & daynum <=8461 & t_asign == 1)
> table(week_treated$w4_lead + week_treated$w3_lead + week_treated$w2_lead + week_treated$w1_lead + week_treated$w0 + week_treated$w1 + week_treated$w2 + week_treated$w3 + week_treated$w4)

   1
5795
```

2. Retainment of the data from the control group (cities without lock down policy)

    - We found that the data from the control group has not been deleted when the replication result is obtained.

Fig 19.replication result using the entire authors' dataset (left) vs. using the data of the treated group (right)



Extension & Discussion

Treat coefficient at a provincial level

- 95 cities were locked down during the pandemic. They are disproportionately distributed in different provinces. To study the treat coefficient of each province, we add the province code in the original data set. First, we use the old ISO 3166-2 to code the province, e.g., “130” for Hebei. The city code and province code are matched manually in Excel (Figure 20).
- The original dataset (called “wf”) covers cities in 32 provincial administrative regions, including provinces, 5 autonomous regions, and 4 municipalities. The province code reference list is matched to the “wf” in R using “for loop”. Then, we did the regression for each province to get the treat coefficient.
    - code 1 in extension_province.R (available at https://github.com/Artemis20123/Statistics-870K/blob/exploration/extension_province.R)
- 16 provinces out of 32 have the treat coefficient. There are two cases for which provinces have no treat coefficient available. There is only one city for municipalities, but contrasts should be applied only to factors with 2 or more levels (Figure 21). For some provinces, no city was locked down during the pandemic. For example, a lockdown was not implemented in Hunan. Therefore, the treat coefficient is NA (Figure 22).

Fig 20.Screenshots of province code reference list sorted by city code (left) and sorted by province code (right)

city_code	city	province_code	province
2414	Harbin	230	Heilongjiang
3547	Benxi	210	Liaoning
4870	Wuhu	340	Anhui
4956	Fuyang	340	Anhui
5993	Chongqing	500	Chongqing
5996	Maanshan	340	Anhui
6124	Tianjin	120	Tianjin
6145	Chengde	130	Hebei
6270	Mianyang	510	Sichuan
7765	Ordos	150	Nei Mongol
8027	Ankang	610	Shaanxi
8131	Chaoyang	210	Liaoning
8615	Xinzhou	140	Shanxi
8775	Weihai	370	Shandong
9121	Baoshan	530	Yunnan
9715	Tieling	210	Liaoning
11242	Wuhai	150	Nei Mongol
11788	Shigatse	540	Tibet
11975	Putian	350	Fujian

city_code	city	province_code	province
54613	Beijing	110	Beijing
6124	Tianjin	120	Tianjin
6145	Chengde	130	Hebei
20585	Tangshan	130	Hebei
39753	Cangzhou	130	Hebei
41928	Qinhuangdao	130	Hebei
50965	Xingtai	130	Hebei
71005	Shijiangzhuang	130	Hebei
77331	Langfang	130	Hebei
93580	Handan	130	Hebei
94313	Zhangjiakou	130	Hebei
97330	Hengshui	130	Hebei
98240	Baoding	130	Hebei
8615	Xinzhou	140	Shanxi
14529	Jincheng	140	Shanxi
22294	Yangquan	140	Shanxi
45700	Yuncheng	140	Shanxi
50167	Taiyuan	140	Shanxi
66511	Datong	140	Shanxi

Fig 21.Regression for Tianjin municipality

```
> prov_120 <- subset(wfprovince, province_code == 120)
> didreg_120 <- lm(aqi ~ treat + as.factor(daynum) + as.factor(city_code),
+                  data = prov_120)
Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) :
  contrasts can be applied only to factors with 2 or more levels
```

Fig 22.Regression for Hunan

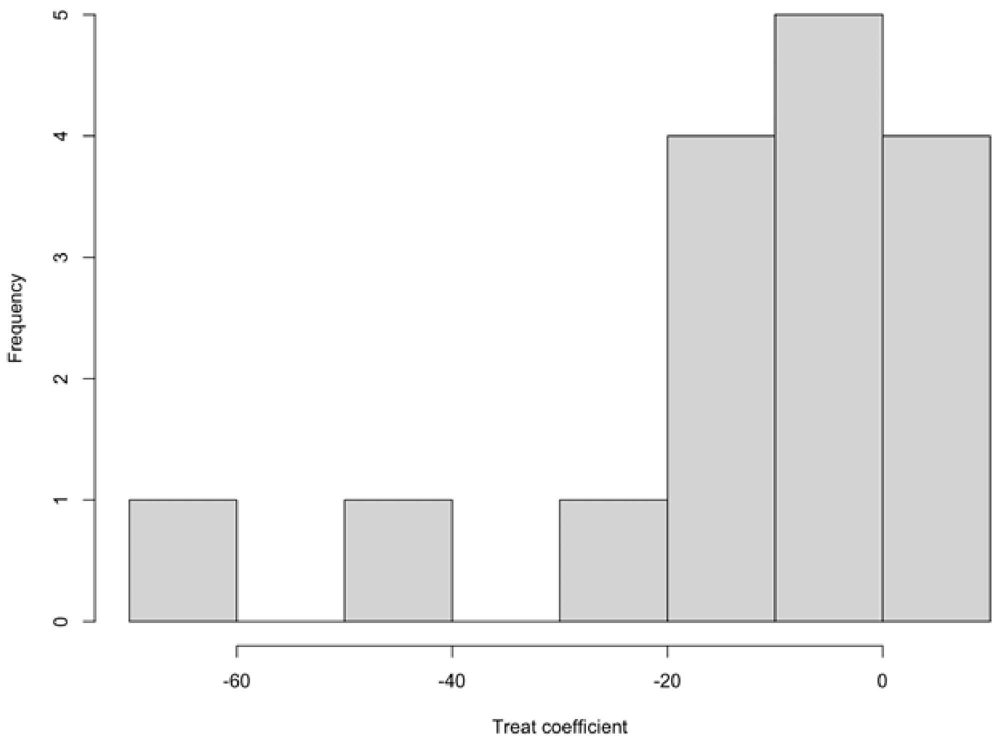
```
> prov_430 <- subset(wfprovince, province_code == 430)
> didreg_430 <- lm(aqi ~ treat + as.factor(daynum) + as.factor(city_code),
+                  data = prov_430)
> summary(didreg_430)

Call:
lm(formula = aqi ~ treat + as.factor(daynum) + as.factor(city_code),
    data = prov_430)

Residuals:
    Min      1Q  Median      3Q     Max
-59.350  -8.855  -0.452   7.810  63.047

Coefficients: (1 not defined because of singularities)
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                  91.008      4.492  20.261  < 2e-16 ***
treat                            NA         NA      NA       NA
```

- The treat coefficient of every province was collected in Excel (Table 2). The histogram is generated to show the distribution of the treat coefficient at a provincial level (Figure 23).
    - code 2 in extension_province.R (available at https://github.com/Artemis20123/Statistics-870K/blob/exploration/extension_province.R)

Table 2.Treat coefficient of each province

province	province_code	treat_beta
Beijing	110	
Tianjin	120	
Hebei	130	-18.20
Shanxi	140	
Nei Mongol	150	
Liaoning	210	-40.43
Jilin	220	
Heilongjiang	230	-61.37
Shanghai	310	
Jiangsu	320	-12.64
Zhejiang	330	-4.01
Anhui	340	-10.50
Fujian	350	0.93
Jiangxi	360	-1.31
Shandong	370	1.58
Henan	410	-27.10
Hubei	420	-13.23
Hunan	430	
Guangdong	440	-0.01
Guangxi	450	-2.40
Hainan	460	
Chongqing	500	
Sichuan	510	1.24
Guizhou	520	
Yunnan	530	4.01
Tibet	540	
Shaanxi	610	
Gansu	620	
Qinghai	630	
Ningxia	640	-1.95
Xinjiang	650	

Fig 23.Histogram of the treat coefficient at the provincial level



- The lockdown policy has a positive impact on air quality in 12 provinces. In comparison, there are 4 provinces that did not experience an improvement in air quality in lockdown cities, i.e. Fujian, Shandong, Sichuan, and Yunnan (Table 2, Figure 23).

- There are three worth noting observations from our data analysis. First, the treat coefficient in Hubei is -13.23. The treat coefficient nationwide is -18.27. All cities in Hubei were locked down. The lockdown policies are strictly implemented. The time for lockdown is the longest in Hubei, especially Wuhan. It is irregular that the treat coefficient of Hubei is not significant compared to the nationwide effects.
- Second, the lockdown policy should have spillover effects. According to spillover effects, the AQI of provinces (i.e. Henan, Anhui, Jiangxi, Hunan, Sichuan, and Shanxi) around Hubei will be affected by the lockdown policy in Hubei. However, spillover effects cannot be seen from the coefficients of surrounding provinces. The coefficient in Sichuan is even positive (Figure 24).

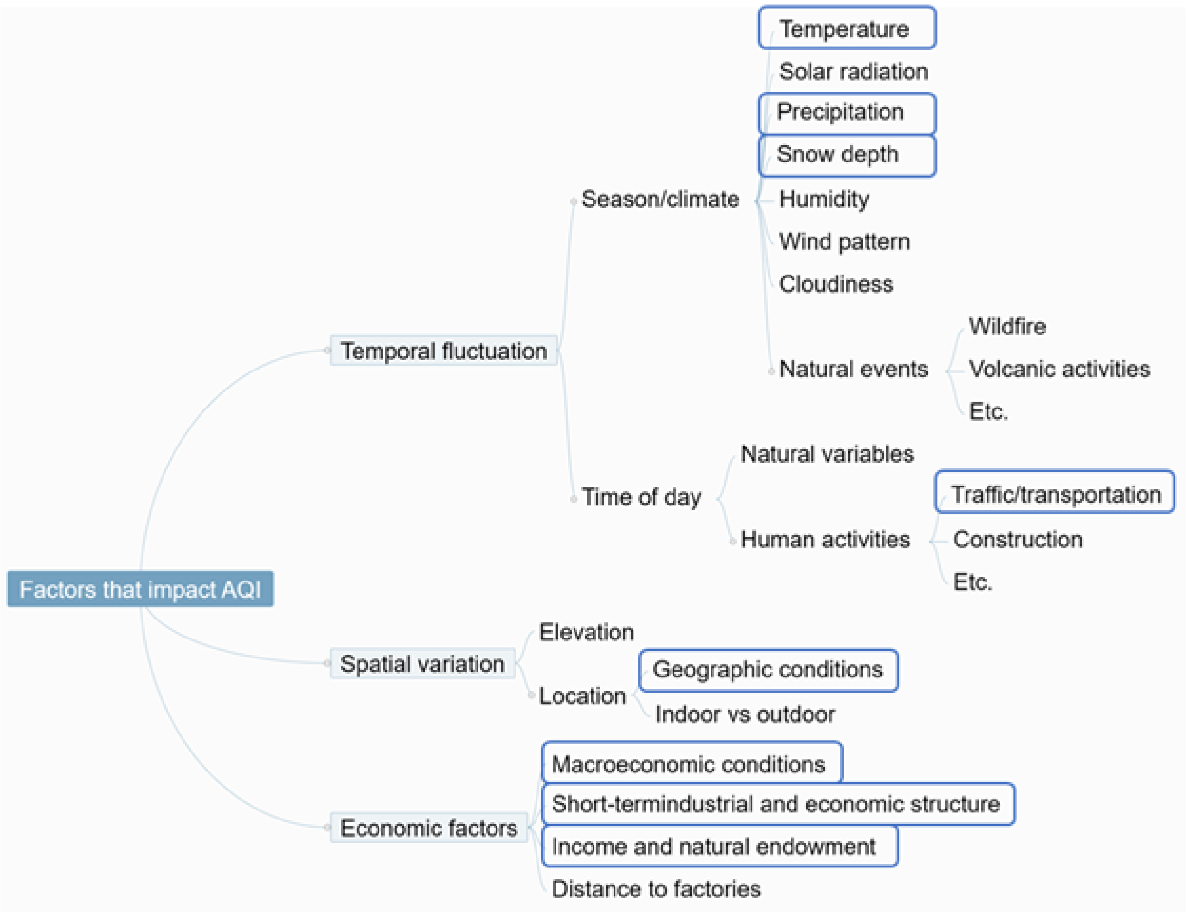
Fig 24.Spatial distribution of the treat coefficient

- Third, from the magnitude of the coefficient, the AQI of northeastern provinces, i.e. Heilongjiang and Liaoning, have dropped significantly (Figure 24). It is partly due to the reduced use of the centralized winter heating system relying heavily on coal burning. However, only 1 city in Heilongjiang was locked down. It is arbitrary to attribute better air quality to the lockdown effects.
- These findings are not consistent with the hypothesis that lockdown policy has a positive impact on air quality. Some other factors affecting the air quality might be overlooked by the authors. Future studies are required to identify the causes of three findings.

AQI Factors

- Over time, researchers have developed various tools to measure air quality. One widely used quantitative measurement is AQI. Its value falls between the range of 0 to 300, and reduction in AQI is associated with improvement in air quality. AQI is calculated based on 5 criteria: carbon monoxide, nitrogen dioxide, ozone, sulfur dioxide, and particulate matter (EPA, 2014). Many factors, both natural and artificial, may alter the value of AQI. As a result of the literature review, we summarize those factors into 3 main categories: temporal fluctuation, spatial variation, and economic factors (EPA, 2014; Han *et al.*, 2019; Liu *et al.*, 2017).
- Figure 25 shows a list of the most common factors that may impact AQI. Although it is not a comprehensive list, it does include twice as many factors as the authors did, which are shown in the blue boxes (He *et al.*, 2020). One particularly notable variable is location; namely, whether an AQI value is recorded indoor or outdoor. Indoor AQI tends to have higher values due to an inherent difference in ventilation (Lawrence & Fatima, 2014). The authors analyzed data collected outdoor. However, most people are kept indoor for almost the entire time during lockdowns. Therefore, indoor AQI is more relevant in this setting, while outdoor AQI is minimally practically meaningful. Future research on lockdown and air quality could consider prioritizing indoor AQI.

Fig 25.Factors that may impact AQI



"Coal to gas"

- Studies reveal a significant influence on air pollution control by changing the fuel from coal to gas in thermal power and central heating systems. *Ceteris paribus*, cities that underwent this change reduced AQI by an average of 15.97 in 5 years (Xiong *et al.*, 2021).
- One quintessential example is Beijing. With 97.4% conversion from coal to gas, Beijing reduced 93.5% SO2 and 20.5% PM (Fang *et al.*, 2021).

- Backed by such evidence, China implemented blunt force regulations on the use of coal since 2015 (Van Der Kamp, 2017). In provinces like Hebei, the government subsidizes residents for upgrading to gas-fueled heating systems. However, the natural gas supply becomes drastically insufficient in winter (Jiang, 2021). The combination of strict regulations on coal usage and shortage in gas makes a compelling contributing factor to air quality improvement during winter, especially in northern provinces.
- The authors of our selected paper did not consider this factor, hinting at possible directions for future research.

References

1. Bruce, N., Perez-Padilla, R., & Albalak, R. (2000). Indoor air pollution in developing countries: a major environmental and public health challenge. *Bulletin of the World Health Organization*, 78, 1078-1092.
2. European Environmental Agency. (2020, 25 March). *Air pollution goes down as Europe takes hard measures to combat coronavirus.* https://www.eea.europa.eu/highlights/air-pollution-goes-down-as
3. EPA. (2014). A guide to air quality and your health. *U.S. Environmental Protection Agency*.
4. Fang, C., Yin, L., and Liu, M. (2021). Discussion on the influence of changing coal to gas in the field of thermal power and central heating on air pollution control in Beijing. *Energy Conservation & Environmental Protection* (10), 78-80.
5. Goolsbee, A., & Syverson, C. (2021). Fear, lockdown, and diversion: Comparing drivers of pandemic economic decline 2020. *Journal of Public Economics*, 193, 104311.
6. Han, X., Li, H., Liu, Q., Liu, F., & Arif, A. (2019). Analysis of influential factors on air quality from global and local perspectives in China. *Environmental Pollution*, 248, 965-979.
7. He, G., Pan, Y., & Tanaka, T. (2020). The short-term impacts of COVID-19 lockdown on urban air pollution in China. *Nature Sustainability*, *3*(12), 1005-1011.
8. Holcombe, M. & O'Key, S. (2020, 23 March). *Satellite images show less pollution over the US as coronavirus shuts down public places.* CNN. https://www.cnn.com/2020/03/23/health/us-pollution-satellite-coronavirus-scn-trnd/index.html
9. Jiang, Y. (2021). Summary of implementation and operation effect of "burning gas instead of coal" project. (姜懿芸. (2021). "煤改气"项目实施及运行效果. (eds.) 2021供热工程建设与高效运行研讨会论文集(pp.30-36). 《煤气与热力》杂志.)
10. Lawrence, A., & Fatima, N. (2014). Urban air pollution & its assessment in Lucknow City—the second largest city of North India. *Science of the total environment*, 488, 447-455.
11. Liu, H., Fang, C., Zhang, X., Wang, Z., Bao, C., & Li, F. (2017). The effect of natural and anthropogenic factors on haze pollution in Chinese cities: A spatial econometrics approach. *Journal of cleaner production*, 165, 323-333.
12. Mahato, S., Pal, S., & Ghosh, K. G. (2020). Effect of lockdown amid COVID-19 pandemic on air quality of the megacity Delhi, India. *Science of the total environment*, 730, 139086.
13. Shalal, A., & Crossley, G. (2020, October 22). *Analysis: China and U.S. economies diverge over coronavirus response.* Reuters. https://www.reuters.com/article/health-coronavirus-usa-china-analysis/analysis-china-and-u-s-economies-diverge-over-coronavirus-response-idUSKBN277066.
14. Schwela, D., Haq, G., Huizenga, C., Han, W. J., Fabian, H., & Ajero, M. (2012). *Urban air pollution in Asian cities: status, challenges and management*. Routledge.
15. Van der Kamp, D. (2017). Clean air at what cost? The rise of blunt force regulation in China (Doctoral dissertation). *University of California Berkeley*.
16. Xiong, Y., Liao, W., and Wang, L. (2021). A study on air pollution governance effect of the "coal-to-gas/electricity" policy. *Collected Essays on Finance and Economics* (3), 103-112.

Data and code availability

- Original data and code will be available at the public repository (https://github.com/yhyhpan/COVID19_LOCKDOWN).
- All data and code in our replication and extension will be available at the public repository (https://github.com/Artemis20123/Statistics-870K). The code for each part is in different branches.