Cross Section Assignment - The Effect of Lockdowns on the Severity Covid-19

Jacques Rossouw[a]

[a]*Stellebosch, Western Cape, South Africa*

Email address: `gerardrossouw@gmail.com` (Jacques Rossouw)

Table of Contents

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
        fig.width = 6, fig.height = 5, fig.pos="H", fig.pos = 'H',
```

```
        purl = TRUE)

pacman::p_load(tidyverse, dplyr, lubridate, AER,
 broom, plm, corrplot, psych, ggrepel,
 gridExtra, stargazer, formatR)

fmxdat::source_all("./code")
```

1. Introduction

The effects of the covid-19 corona virus has led to global economic strain and the debate over whether strict lockdown rules were necessary inspite of their economic implications is debated by some.

The problem with identifying the effect that covid policy stringency had on covid-19 is with the idiosynctratic differences between countries that also contributed to the severity of how covid would have affected individuals. This suggests that the fixed effects regression technique offers a unique opportunity to account for these fixed effects between countries, and to isolate the effect that policy had on covid. Towards fighting the coronavirus as a different flu from the typical cold, the amount of people hospitalised, those ending up in ICU, or deaths per unit of time is measured relative to the amount of new cases. The typical flu also spreads rapidly but the the problem is set up as mentioned, to isolate the effect over and above being a standard flu.

By using the data from H. Ritchie (2020), the data is first cleaned by filtering for only the country components, and removing the countries for which data does not exist after a certain date.

2. Data Cleaning/Feature Selection

Some transformations are also made to columns to make them more usable for regression. The variables that are distributed on a wider range, or scale, are also scaled to ensure that the OLS estimation is not biased by this. The code in the Appendix illustrates this.

2.1. Converting to Cumulative Values

```
# See appendix
world_df <- world_df %>% scale_bigs_cumsum(.)
```

2.2. Variable Features Scaling

```
##                        mean         sd   min         max       range
## new_cases          261519.32 1220518.92  0.00 25275423.00 25275423.00
## afflicted_rate         26.00      85.81  0.00      931.54      931.54
## reproduction_rate       0.77       0.44 -0.01        2.06        2.08
## new_tests             487.10    1789.23  0.00    32919.30    32919.30
## new_vaccinations      275.41     558.44  0.00     3041.92     3041.92
## stringency_index       44.81      25.15  0.00       99.06       99.06
```

Thus, want to scale: `new_test, new_vaccinations`

Plotting to see whether there is any irregularity in the distribution of the dependent variable.
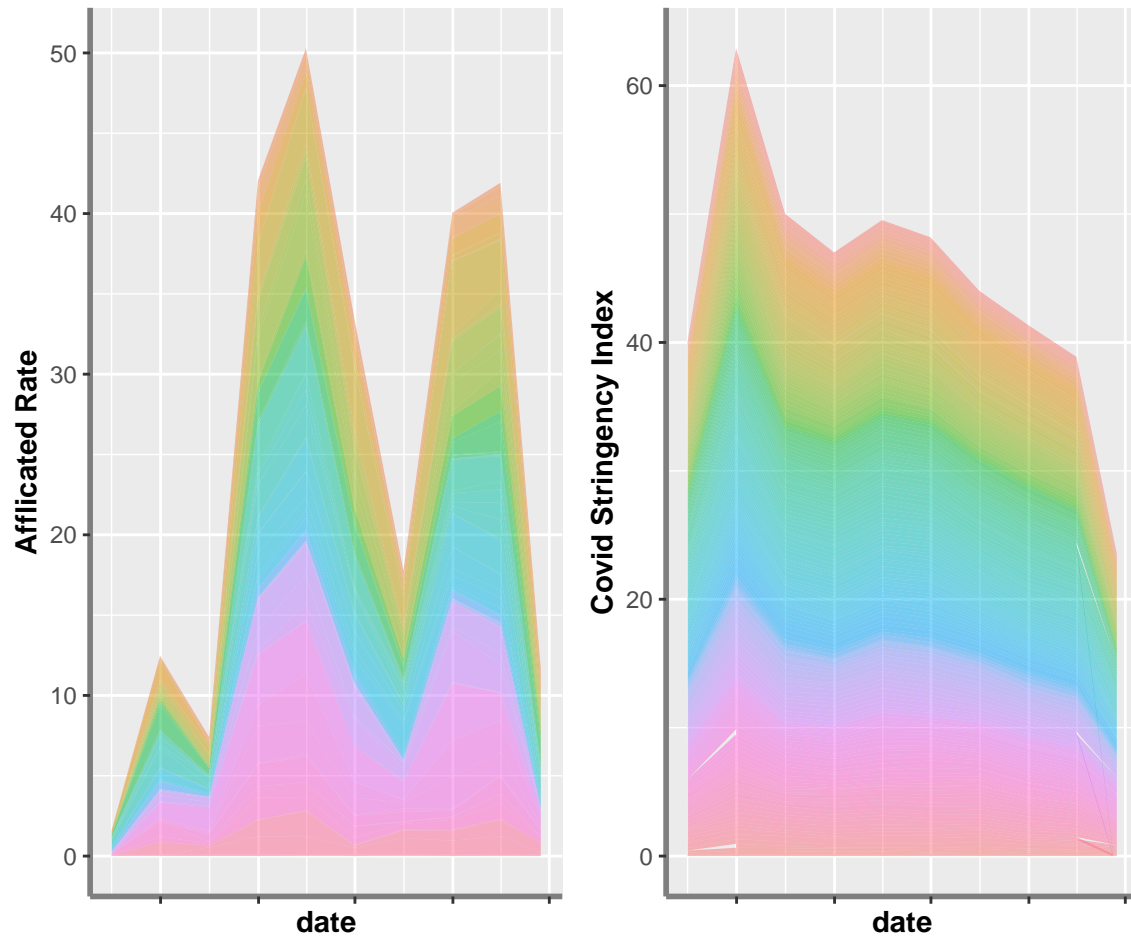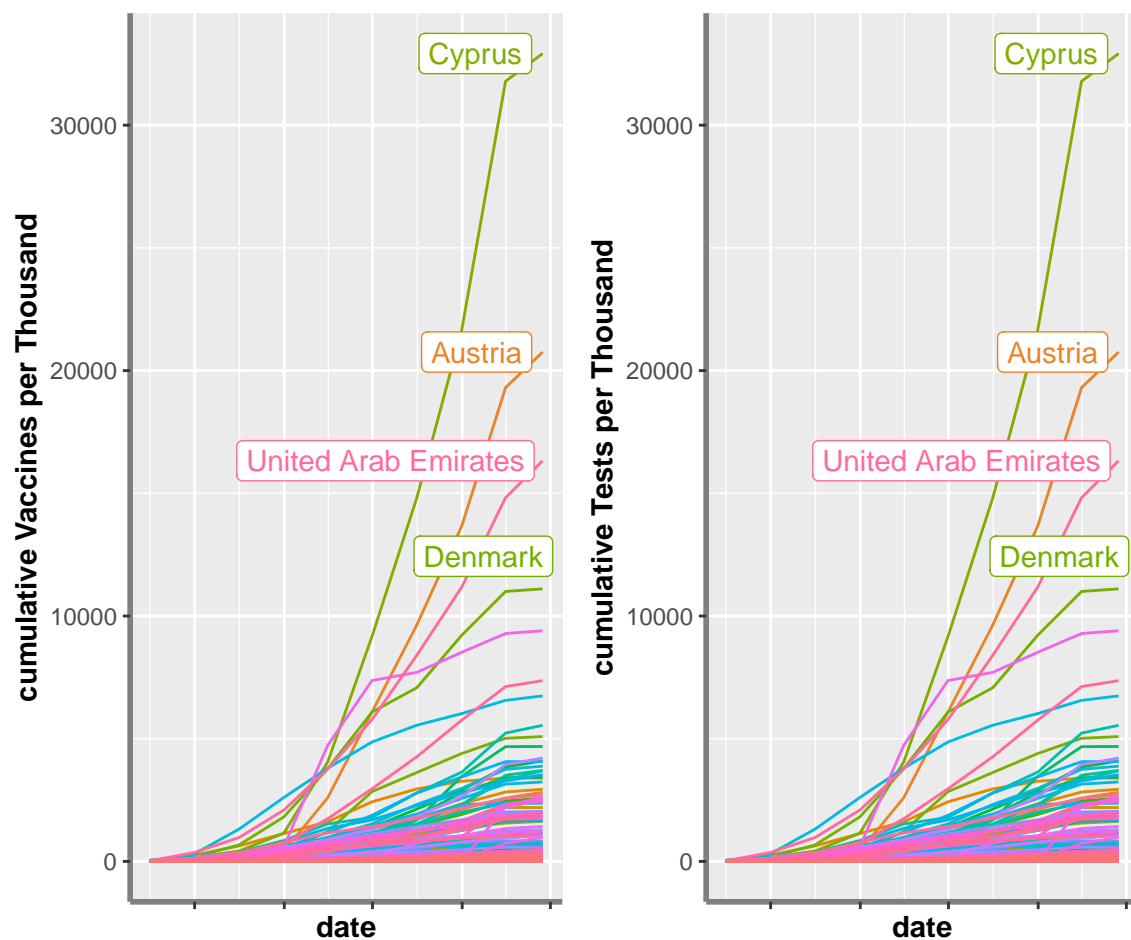
```
world_df <- world_df %>% scale_bigs_scale(.)
```

2.3. Fixed Effects Feature Scaling

Now, to check the scales of the features that remain constant per country:

```
##                        mean       sd min       max      range
## gdp_per_capita     17697.35 20539.28   0 116935.60 116935.60
## population_density   444.44  2094.60   0  20546.77  20546.77
## median_age            27.58    12.79   0     48.20     48.20
## aged_65_older          7.90     6.48   0     27.05     27.05
## extreme_poverty        7.83    16.76   0     77.60     77.60
## cardiovasc_death_rate 226.19  135.19   0    724.42    724.42
## diabetes_prevalence    7.52     4.59   0     23.36     23.36
```

```
## handwashing_facilities      21.89    32.72    0    99.00    99.00
## hosp_beds_1k                  2.38     2.51    0    13.80    13.80
## life_expectancy              73.36     9.08    0    86.75    86.75
## human_development_index       0.63     0.28    0     0.96     0.96
## smokers                      14.38    12.75    0    45.95    45.95
```

Additional features that need to be scaled are this - gdp_per_capita - population_density -
cardiovasc_death_rate
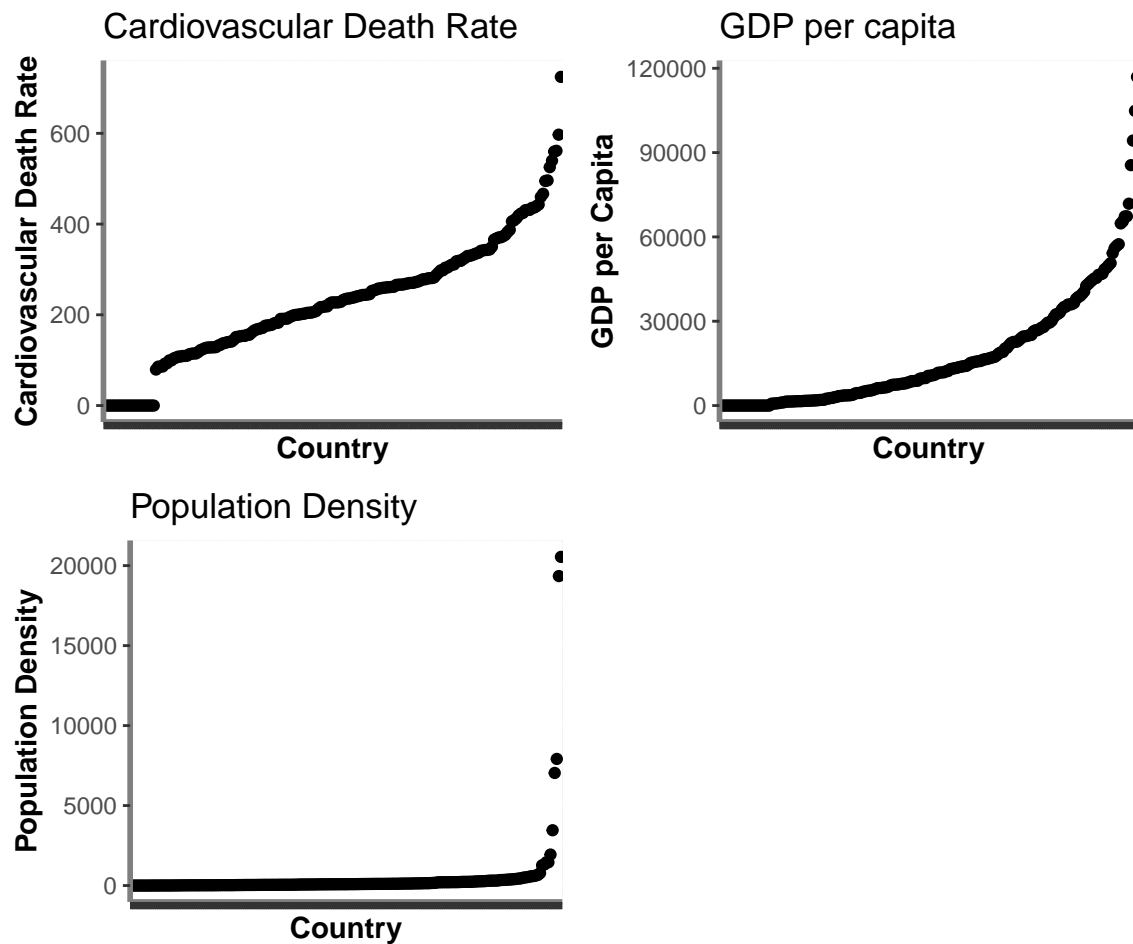
The non-linear distribution of Real GDP suggests a transformation. Presence of outliers and large
range in population_density suggests normalisation scaling. The same goes for the cardiovascular
death rate.

```
world_df <- world_df %>% scale_bigs_constant(.)
```

The value of the stringency index is assumed to have a delayed effect on the coronavirus, therefore, to account for this, one quarter lagged average stringency index value is associated the current with each current period.

```r
world_df <- world_df %>%
    group_by(location) %>%
    mutate(across(date, function(x) floor_date(x, unit = "quarters"))) %>%
    ungroup()

quick_df <- world_df %>%
    select(location, date, stringency_index) %>%
    group_by(location) %>%
    mutate(across(date, function(x) x %m+% months(3))) %>%
    filter(date != last(date))

world_df <- quick_df %>% left_join(world_df, by = c("location", "date")) %>%
    select(-stringency_index.y)
```
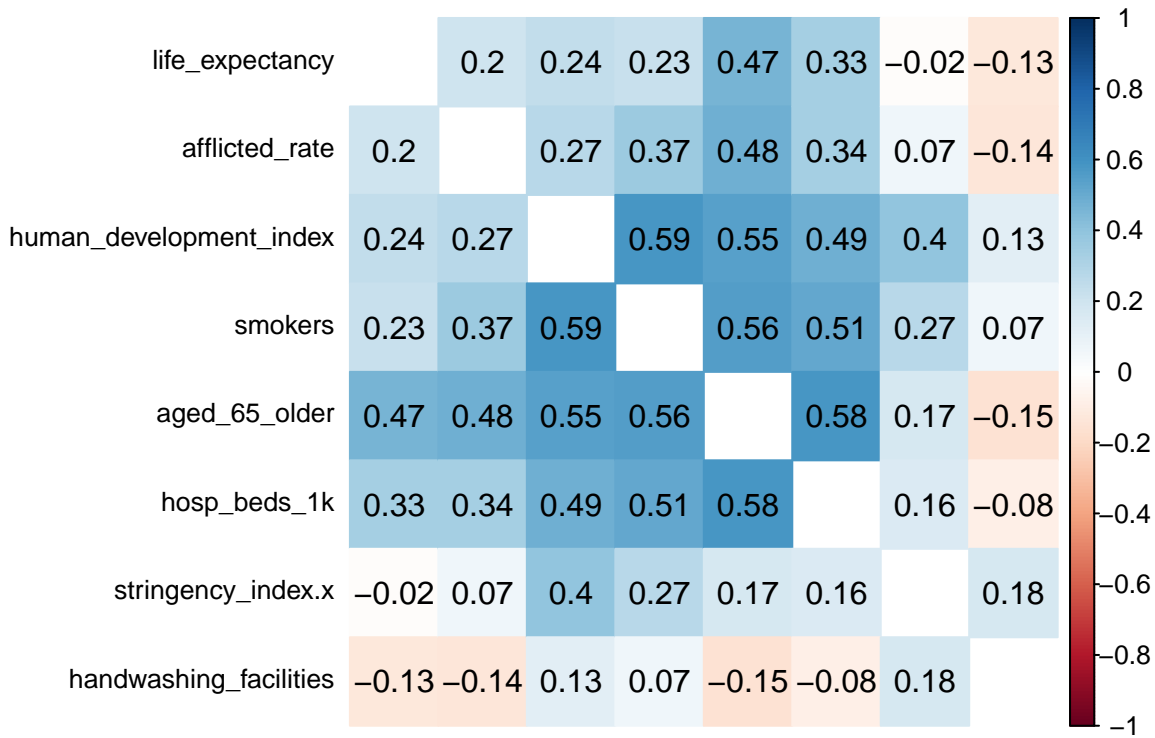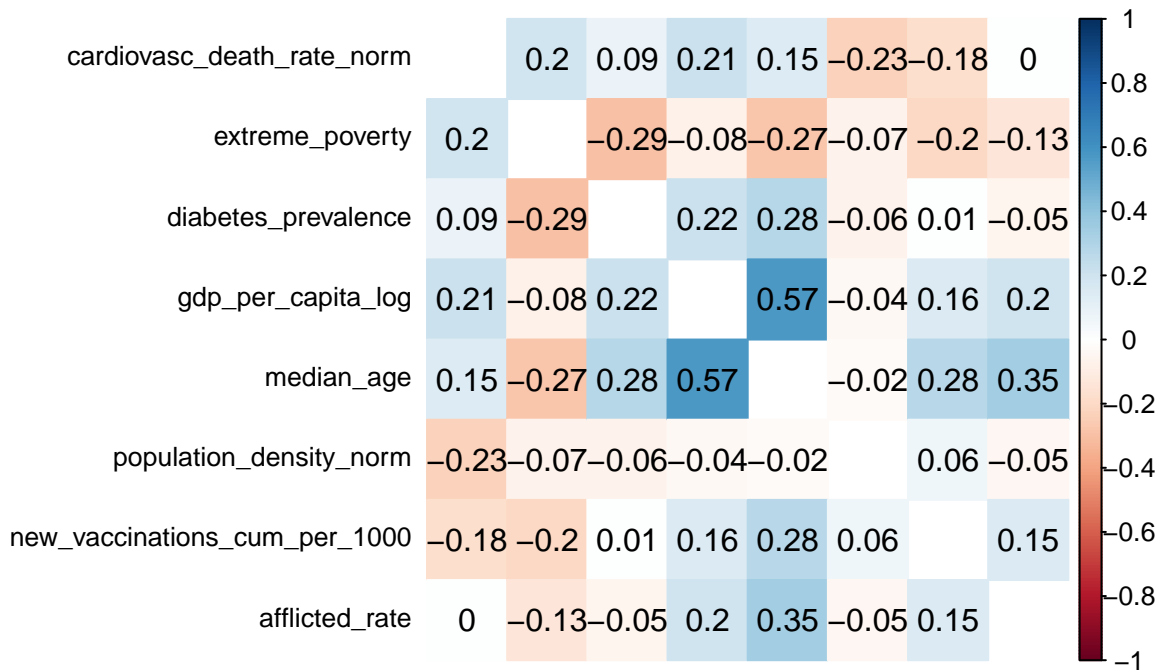
3. Correlation Analysis


```
## [1] "full"
```

	cardiovasc_death_rate_norm	extreme_poverty	diabetes_prevalence	gdp_per_capita_log	median_age	population_density_norm	new_vaccinations_cum_per_1000	afflicted_rate
cardiovasc_death_rate_norm		0.2	0.09	0.21	0.15	−0.23	−0.18	0
extreme_poverty	0.2		−0.29	−0.08	−0.27	−0.07	−0.2	−0.13
diabetes_prevalence	0.09	−0.29		0.22	0.28	−0.06	0.01	−0.05
gdp_per_capita_log	0.21	−0.08	0.22		0.57	−0.04	0.16	0.2
median_age	0.15	−0.27	0.28	0.57		−0.02	0.28	0.35
population_density_norm	−0.23	−0.07	−0.06	−0.04	−0.02		0.06	−0.05
new_vaccinations_cum_per_1000	−0.18	−0.2	0.01	0.16	0.28	0.06		0.15
afflicted_rate	0	−0.13	−0.05	0.2	0.35	−0.05	0.15	

```
## [1] "full"
```

The components that will form a part of the larger OLS regression is therefore the:

- stringency index `stringency_index.x`

- development index `human_development_index`

- proportion of smokers `smokers`

- population over the age of 65 `aged_65_older`

- and hospital beds per 1000 `hosp_beds_1k`

- Availability of hand washing facilities `handwashing_facilities`

Hand washing facilities is added based on intuitive interpretation

4. Regressions

4.1. OLS Regression

```
mod_ols_1 <- plm(afflicted_rate ~ stringency_index.x +
   handwashing_facilities + reproduction_rate,
    index = c("location", "date"), data = world_df,
   model = "pooling")

mod_ols_2 <- plm(afflicted_rate ~ stringency_index.x + smokers
   + handwashing_facilities + aged_65_older + hosp_beds_1k
   + human_development_index + reproduction_rate
   + new_vaccinations_cum_per_1000,
   data = world_df,
   index = c("location", "date"), model = "pooling")

mod_1sls <- plm(stringency_index.x ~ gdp_per_capita_log
               + population_density_norm,
               data = world_df,
               index = c("location", "date"), model = "pooling")

stringency_hat <- fitted.values(mod_1sls)

mod_2sls <- plm(afflicted_rate ~ stringency_hat + smokers
   + handwashing_facilities + aged_65_older + hosp_beds_1k
   + human_development_index + new_vaccinations_cum_per_1000,
   data = world_df,
   index = c("location", "date"), model = "pooling")

# To get robust standard errors
robustse_ols1 <- sqrt(diag(vcovHC(mod_ols_1, type = "HC1")))
robustse_ols2 <- sqrt(diag(vcovHC(mod_ols_2, type = "HC1")))
robustse_2sls <- sqrt(diag(vcovHC(mod_2sls, type = "HC1")))
```

```
stargazer(mod_ols_1, mod_ols_2, mod_2sls, header = F, font.size = "footnotesize",
    se = list(robustse_ols1, robustse_ols2, robustse_2sls), column.labels = c("OLS",
        "OLS", "2SLS"))
```

Table 4.1

	Dependent variable:		
	afflicted_rate		
	OLS	OLS	2SLS
	(1)	(2)	(3)
stringency__index.x	0.072	−0.108	
	(0.096)	(0.112)	
stringency_hat			−0.092
			(0.396)
smokers		1.138***	1.136***
		(0.434)	(0.429)
handwashing__facilities	−0.474***	−0.226	−0.226
	(0.172)	(0.151)	(0.149)
aged__65_older		5.206***	5.151***
		(0.936)	(0.954)
hosp__beds__1k		1.727	1.841
		(2.472)	(2.495)
human__development__index		−0.188	−0.107
		(0.172)	(0.194)
reproduction__rate	39.621***	10.311**	
	(7.000)	(4.408)	
new__vaccinations__cum__per__1000		0.163	0.079
		(2.338)	(2.330)
Constant	3.190	−19.524***	−16.706
	(2.717)	(5.497)	(13.284)
Observations	1,844	1,844	1,844
R^2	0.054	0.256	0.255
Adjusted R^2	0.052	0.253	0.252
F Statistic	34.712*** (df = 3; 1840)	78.935*** (df = 8; 1835)	89.570*** (df = 7; 1836)

Note: *p<0.1; **p<0.05; ***p<0.01

```
mod_fe_1 <- plm(afflicted_rate ~ stringency_index.x + smokers
    + handwashing_facilities + aged_65_older + hosp_beds_1k
    + human_development_index + new_vaccinations_cum_per_1000,
    data = world_df,
    index = c("location"),
  model = "within", effect = "individual")



robustse_fe1 <- sqrt(diag(vcovHC(mod_fe_1, type = "HC1")))
```

```
mod_re_1 <- plm(afflicted_rate ~ stringency_index.x + smokers
    + handwashing_facilities + aged_65_older + hosp_beds_1k
    + human_development_index + new_vaccinations_cum_per_1000,
                  data = world_df,
                  index = c("location"),
                model = "random", effect = "individual")

robustse_re1 <- sqrt(diag(vcovHC(mod_re_1, type = "HC1")))
phtest(mod_fe_1, mod_re_1)
```

```
##
##  Hausman Test
##
## data:  afflicted_rate ~ stringency_index.x + smokers + handwashing_facilities +  ...
## chisq = 0.21695, df = 2, p-value = 0.8972
## alternative hypothesis: one model is inconsistent
```

```
stargazer(mod_fe_1, mod_re_1, header = F, font.size = "small", se = list(robustse_re1),
    column.labels = c("FE", "RE"))
```

Table 4.2

	Dependent variable:	
	afflicted_rate	
	FE	RE
	(1)	(2)
stringency_index.x	−0.088	−0.084
	(0.073)	(0.094)
smokers		1.151***
		(0.414)
handwashing_facilities		−0.218*
		(0.126)
aged_65_older		5.095***
		(0.849)
hosp_beds_1k		1.835
		(2.018)
human_development_index		−0.110
		(0.188)
new_vaccinations_cum_per_1000	1.117***	0.945
	(1.542)	(1.640)
Constant		−16.829*
		(9.858)
Observations	1,844	1,844
R^2	0.001	0.069
Adjusted R^2	−0.126	0.066
F Statistic	0.627 (df = 2; 1636)	137.012***
Note:	*p<0.1; **p<0.05; ***p<0.01	

```
knitr::kable(tidy(pFtest(mod_fe_1, mod_2sls)), caption=
        "Fixed effects test: Ho:'No fixed effects'")
```

Table 4.3: Fixed effects test: Ho:'No fixed effects'

df1	df2	statistic	p.value	method	alternative
200	1636	8.099815	0	F test for individual effects	significant effects

4.3. Interpretations

It is not surprising that the health measures such as proportion the of smokers larger older generations had a significant impact on the amount of individuals that were afflicted from covid. This holds for the OLS, 2SLS, and the RE model. It is surprising, however, that the availability of handwashing facilities had no significant impact, except for in the RE model.

The Hausman test results in a p-value of 0.8877 which implies the null hypothesis cannot be rejected and either model, RE or FE is consistent, however, RE is more efficient.

Table 4.3 illustrates the motivation for the use of a fixed effects model. The `pFtest` compares OLS with Fixed Effects. The result suggests a rejection of the null in favour of the Fixed effects model. That is, there exists fixed effects whitin the data.

Most importantly, independent of which model is chosen, there seems to be no significance between the stringency index and afflicted rate. The models presented here offer variable approaches to analysing the causal effect. The results should not be taken verbatim, however. There are alternative non-parametric models hat can allow for a more accurate anaylisis. Similar attempts have been made, such as the recen paper by Huang, Shao, Xing, Hu, Sin & Zhang (2021). Their methods also suggest a machine learning based model can offer an ideal substitute. Based on the results from above, the lockdown measures did not have a significant impact on those individuals severly affected by Covid-19, over and above the normal effects of a widespread flu.

4.3.1. Measurement Error & Bias

It is important to note that the unexpected nature of Covid-19 resulted in a lot of "on the fly" data collection which can lead to larger measurement errors. The reliability of measurement error and false estimates in this case is questionable. Furthermore, the fixed- and random effects regressions can exacerbate this problem. In that case, it would suggest that a pooled OLS regression offer some particular benefits.

4.3.2. Assumptions of OLS:

There exists an inherent endogeneity in this dataset, since there exists some country specific factors that might have led to lockdowns being more effective or having more resources to withstand the aderse effects of lockowns. For this reason, the last OLS model includes a two stage least squares regression model to mitigate this issue. Additionally, the Random- and Fixed effects models are also fitted. These can more precisely account for country specific factors playing a role.

The Instrumental Variable regression, by method of two stage least squares, requires the following assumptions to hold for an unbiased estimate:

Instrument Variable: real GDP, and population density, such that they affect the afflicted rate through the stringency index

1.   Exclusion Restriction

This requires that the real GDP, and population density only affects the afflicted rate through the stringency index.

This assumption very likely to holds. There is no reason to believe that real GDP affects the afflicted rate per ten thousand individuals in the population. There is however, a possibility that population density has an effect on this.

2.   Random Assignment

This requires that there be no difference between the potential outcomes

3.   Instrument Relevance

This requires that real GDP and population density actually have a significant effect on the the stringency index value. Intuitively, this should be the case since wealthier countries have more resources to apply stricter rules and more densely populated areas ought to require more strict policy to combat the effects if Covid-19.

4.   Monotonicity

Whether any country could be expected to have endorsed less strict policy based on their high real GDP or high population density.

For the latter case, this is unlikely the case. However, some countries might have been more relaxed from having more resources to cope with the effects of Covid.

4.3.3. Assumptions of Fixed- and Random Effects

The assumptions that need to hold here, include:

1.   Type of Unconfoundedness

Conditioning on the fixed effects per country (and other covariates), there should be no other reason for selection into treatment, i.e. having more stringent policy. This does seem likely to hold.

2.   Same Time Trends

There is no way to assert that this assumption holds with certainty.

5. Concluion

There does not seem to be a significant causal effect of the level of strictness regarding covid lockdown and the rate of afflicted individuals per ten thousand people in a population. There exists some room for improvement on the models presented hre. Perhaps a non-linear model such as a machine learning based option would yield more accurate results.

6. References

10 H. Ritchie, L.R.-G., E. Mathieu. 2020. Coronavirus pandemic (COVID-19). *Our World in Data.*

Huang, X., Shao, X., Xing, L., Hu, Y., Sin, D.D. & Zhang, X. 2021. The impact of lockdown timing on COVID-19 transmission across US counties. *EClinicalMedicine.* 38:101035.

7. Appendix

```r
# Checking the date of first observations
start_date()



# fetching and cleaning the dataset
world_df <- extract_all() %>%
    feature_adj_all() %>%
    experiment_aggregate_week() %>%
    experiment_trim() %>%
    relocate(afflicted_rate, .before = reproduction_rate)


world_df
# See appendix
world_df <- world_df %>% scale_bigs_cumsum(.)
world_df <- world_df %>%
    group_by(location) %>%
    mutate(across(date, function(x) floor_date(x, unit = "quarters"))) %>%
    ungroup()

quick_df <- world_df %>%
    select(location, date, stringency_index) %>%
    group_by(location) %>%
    mutate(across(date, function(x) x %m+% months(3))) %>%
    filter(date != last(date))

world_df <- quick_df %>% left_join(world_df, by = c("location", "date")) %>%
    select(-stringency_index.y)
cor_plot2(world_df)$arg$type
names(world_df)
knitr::kable(tidy(pFtest(mod_fe_1, mod_2sls)), caption=
        "Fixed effects test: Ho:'No fixed effects'")
```

7.1. Functional Code

```r
## function (df)
```

```
## {
##     plot <- df %>% ggplot(aes(fill = location, y = afflicted_rate/206,
##         x = date)) + geom_area(position = "stack", stat = "identity",
##         alpha = 0.5) + theme(legend.position = "none") + theme(axis.text.x = element_blank(),
##         axis.title = element_text(face = "bold"), axis.line = element_line(colour = "grey50",
##             size = 1)) + scale_y_continuous("Afflicated Rate")
##     return(plot)
## }


## function (df)
## {
##     plot <- df %>% ungroup() %>% group_by(location) %>% filter(date ==
##         last(date)) %>% ggplot() + geom_point(aes(x = reorder(location,
##         cardiovasc_death_rate, mean), y = cardiovasc_death_rate)) +
##         theme(axis.text.x = element_blank(), axis.title = element_text(face = "bold"),
##             axis.line = element_line(colour = "grey50", size = 1)) +
##         scale_y_continuous("Cardiovascular Death Rate") + scale_x_discrete("Country") +
##         labs(title = "Cardiovascular Death Rate")
##     return(plot)
## }


## function (df, constant_features = c("gdp_per_capita", "population_density",
##     "median_age", "aged_65_older", "extreme_poverty", "cardiovasc_death_rate",
##     "diabetes_prevalence", "handwashing_facilities", "hosp_beds_1k",
##     "life_expectancy", "human_development_index", "smokers"))
## {
##     descriptive_stats <- df %>% ungroup() %>% select(-c(constant_features,
##         date, location)) %>% describe() %>% select(-c(median,
##         mad, se, vars, n, skew, kurtosis, trimmed))
##     return(descriptive_stats)
## }


## function (df, constant_features = c("gdp_per_capita", "population_density",
##     "median_age", "aged_65_older", "extreme_poverty", "cardiovasc_death_rate",
##     "diabetes_prevalence", "handwashing_facilities", "hosp_beds_1k",
##     "life_expectancy", "human_development_index", "smokers"))
## {
```

```
##      descriptive_stats <- df %>% ungroup() %>% select(constant_features) %>%
##          describe() %>% select(-c(median, mad, se, vars, n, skew,
##          kurtosis, trimmed))
##      return(descriptive_stats)
## }


## function (df)
## {
##      plot1 <- df %>% ungroup() %>% select(-c(location, date, gdp_per_capita_log,
##          population_density_norm, cardiovasc_death_rate_norm,
##          new_tests_cum_per_1000, new_vaccinations_cum_per_1000,
##          median_age, extreme_poverty, diabetes_prevalence, new_cases,
##          reproduction_rate, new_cases)) %>% cor(.)
##      plot2 <- plot1 %>% corrplot(., method = "color", order = "hclust",
##          tl.srt = 0, diag = F, tl.col = "black", addCoef.col = "black",
##          tl.pos = "l", tl.cex = 0.8, number.font = 8)
##      return(plot2)
## }


## function (df)
## {
##      plot1 <- df %>% ungroup() %>% select(c(gdp_per_capita_log,
##          population_density_norm, cardiovasc_death_rate_norm,
##          new_vaccinations_cum_per_1000, median_age, extreme_poverty,
##          diabetes_prevalence, afflicted_rate)) %>% cor(.)
##      plot2 <- plot1 %>% corrplot(., method = "color", order = "hclust",
##          tl.srt = 0, diag = F, tl.col = "black", addCoef.col = "black",
##          tl.pos = "l", tl.cex = 0.8, number.font = 8)
##      return(plot2)
## }


## function (df)
## {
##      plot <- world_df %>% ungroup() %>% group_by(location) %>%
##          mutate(label = if_else(date == last(date), as.character(location),
##              NA_character_)) %>% ggplot(aes(x = date, y = new_tests,
##          group = location, col = location)) + geom_line() + theme(axis.text.x = element_blank(
```

```
##           axis.title = element_text(face = "bold"), axis.line = element_line(colour = "grey50",
##               size = 1)) + scale_y_continuous("cumulative Vaccines per Thousand") +
##           geom_label_repel(aes(label = label), nudge_x = 1, na.rm = TRUE) +
##           theme(legend.position = "none")
##       return(plot)
## }
## <bytecode: 0x000002247ea762b0>


## function (df)
## {
##       plot <- df %>% ungroup() %>% group_by(location) %>% mutate(label = if_else(date ==
##           last(date), as.character(location), NA_character_)) %>%
##           ggplot(aes(x = date, y = new_tests, group = location,
##               col = location)) + geom_line() + theme(axis.text.x = element_blank(),
##           axis.title = element_text(face = "bold"), axis.line = element_line(colour = "grey50",
##               size = 1)) + scale_y_continuous("cumulative Tests per Thousand") +
##           geom_label_repel(aes(label = label), nudge_x = 1, na.rm = TRUE) +
##           theme(legend.position = "none")
##       return(plot)
## }
## <bytecode: 0x0000022418a693c8>


## function (df)
## {
##       constant_features <- c("gdp_per_capita", "population_density",
##           "median_age", "aged_65_older", "extreme_poverty", "cardiovasc_death_rate",
##           "diabetes_prevalence", "handwashing_facilities", "hosp_beds_1k",
##           "life_expectancy", "human_development_index", "smokers",
##           "population")
##       mean_cols = c("reproduction_rate", "stringency_index")
##       df <- df %>% replace(is.na(.), 0) %>% select(-excess_mortality) %>%
##           mutate(year_quarter = paste(year(date), quarter(date),
##               sep = "-")) %>% relocate(year_quarter, .before = date) %>%
##           group_by(location, year_quarter) %>% mutate(across(-c(constant_features,
##           mean_cols, date), sum), across(c(mean_cols), function(x) mean(x))) %>%
##           ungroup()
##       return(df)
## }
```

```
## <bytecode: 0x0000022401960be0>


## function (df)
## {
##     df <- df %>% group_by(location, year_quarter) %>% filter(row_number() ==
##         n()) %>% ungroup() %>% select(-c(year_quarter)) %>% group_by(location,
##         date) %>% mutate(afflicted_rate = ((new_deaths + icu_patients +
##         hosp_patients)/population * 10000), .keep = "unused") %>%
##         replace(is.na(.), 0)
##     return(df)
## }


## function (path = "./data/owid-covid-data.csv")
## {
##     names1 <- c(".+smoothed.*", ".+per_million", ".+per_thousand",
##         ".+per_hundred", ".*cumulative.*", ".*weekly.*", "total.+")
##     continents <- extract_continents()$continent
##     df <- read_csv(file = path, show_col_types = F) %>% filter(!location %in%
##         c(continents)) %>% filter(!is.na(continent)) %>% group_by(location) %>%
##         filter(first(date) <= lubridate::ymd(20200430)) %>% ungroup() %>%
##         select(-c(iso_code, continent, tests_units)) %>% rename(hosp_beds_1k = hospital_beds_
##         .[, !grepl(names(.), pattern = paste(names1, collapse = "|"))]
##     return(df)
## }
## <bytecode: 0x0000022403eba928>


## function (path = "./data/owid-covid-data.csv")
## {
##     continents <- read_csv(file = path, show_col_types = F) %>%
##         select(continent) %>% unique()
##     return(continents)
## }


## function (df)
## {
##     df %<>% mutate(new_vaccinations = (new_vaccinations/population) *
##         1000) %>% mutate(new_tests = (new_tests/population) *
```

```
##          1000) %>% select(-c(aged_70_older, people_fully_vaccinated,
##          people_vaccinated, tests_per_case, positive_rate)) %>%
##          group_by(location) %>% mutate(smokers = mean(c(female_smokers,
##          male_smokers)), .keep = "unused")
##      return(df)
## }


## function (df)
## {
##      plot <- df %>% ungroup() %>% group_by(location) %>% filter(date ==
##          last(date)) %>% ggplot() + geom_point(aes(x = reorder(location,
##          gdp_per_capita, mean), y = gdp_per_capita)) + theme(axis.text.x = element_blank(),
##          axis.title = element_text(face = "bold"), axis.line = element_line(colour = "grey50",
##              size = 1)) + scale_y_continuous("GDP per Capita") +
##          scale_x_discrete("Country") + labs(title = "GDP per capita")
##      return(plot)
## }


## function (df)
## {
##      plot <- df %>% ungroup() %>% group_by(location) %>% filter(date ==
##          last(date)) %>% ggplot() + geom_point(aes(x = reorder(location,
##          population_density, mean), y = population_density)) +
##          theme(axis.text.x = element_blank(), axis.title = element_text(face = "bold"),
##              axis.line = element_line(colour = "grey50", size = 1)) +
##          scale_y_continuous("Population Density") + scale_x_discrete("Country") +
##          labs(title = "Population Density")
##      return(plot)
## }


## function (df)
## {
##      df <- df %>% ungroup() %>% mutate(across(c("population_density",
##          "cardiovasc_death_rate"), function(x) scale(x, center = T),
##          .names = "{.col}_norm"), across(c("gdp_per_capita"),
##          function(x) if_else(x == 0, 0, log(x)), .names = "{.col}_log"),
##          across(c("human_development_index"), function(x) x *
```

```
##               100, .names = "{.col}"), .keep = "unused")
##     return(df)
## }


## function (df, big_cols = c("new_tests", "new_vaccinations"))
## {
##     df <- df %>% ungroup() %>% group_by(location) %>% mutate(across(big_cols,
##         function(x) cumsum(x), .names = "{.col}"), .keep = "unused") %>%
##         return(df)
## }


## function (df, big_cols = c("new_vaccinations", "new_tests"))
## {
##     df <- df %>% ungroup() %>% mutate(across(big_cols, function(x) scale(x),
##         .names = "{.col}_cum_per_1000"), .keep = "unused")
##     return(df)
## }


## function ()
## {
##     plot <- read_csv(file = "./data/owid-covid-data.csv", show_col_types = F) %>%
##         group_by(location) %>% filter(date == first(date)) %>%
##         ggplot() + geom_point(aes(x = location, y = date)) +
##         geom_hline(yintercept = lubridate::ymd(20200430), color = "red") +
##         theme(axis.text.x = element_blank(), axis.title = element_text(face = "bold"),
##             axis.line = element_line(colour = "grey50", size = 1)) +
##         scale_y_date("First Date Observation")
##     return(plot)
## }
```