Cross Section Assignment

Jacques Rossouw[a]

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

Table of Contents

1 Introduction **2**

2 Data Cleaning/Feature Selection **2**

2.1 Converting to Cumulative Values 2

2.2 Variable Features Scaling 2

2.3 Fixed Effects Feature Scaling 4

3 Correlation Analysis **7**

4 Regressions **9**

4.1 OLS Regression 9

4.2 Fixed- and Random Effects Regression 12

5 References **14**

6 Appendix **14**

6.1 Functional Code 15

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

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

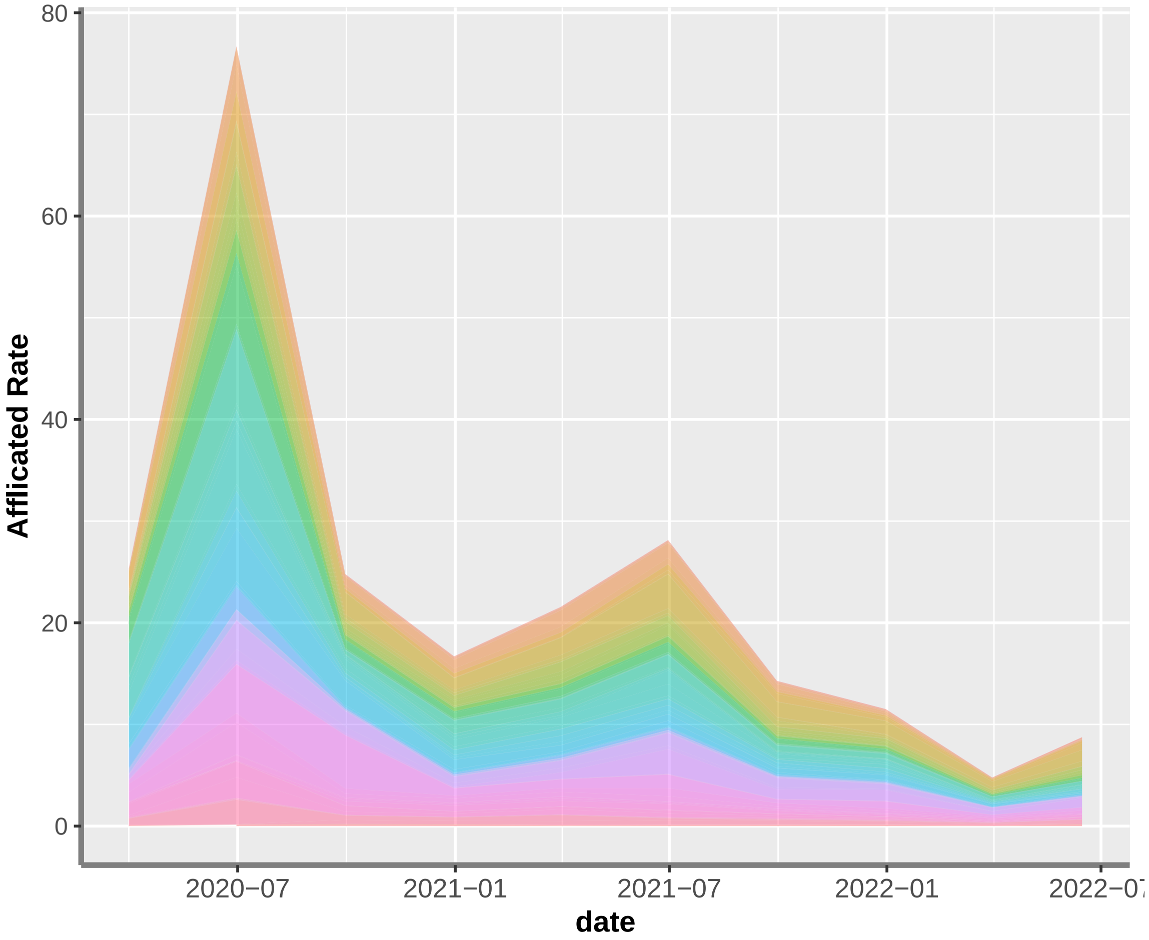
```
# See appendix
world_df <- world_df %>% scale_bigs_cumsum(.)
```

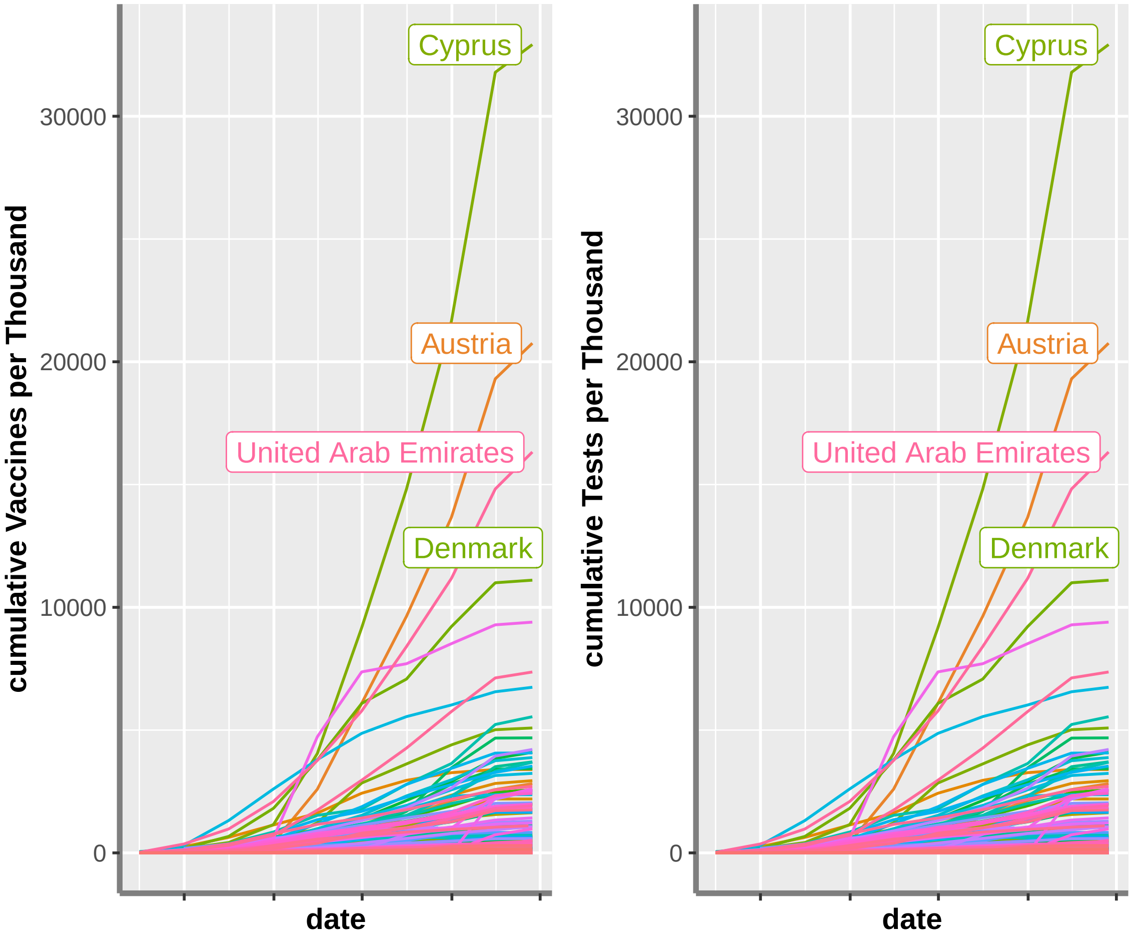
2.2. Variable Features Scaling

```
##                       mean      sd   min      max    range
## afflicted_rate       23.37   88.83  0.00  1479.96  1479.96
## 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.





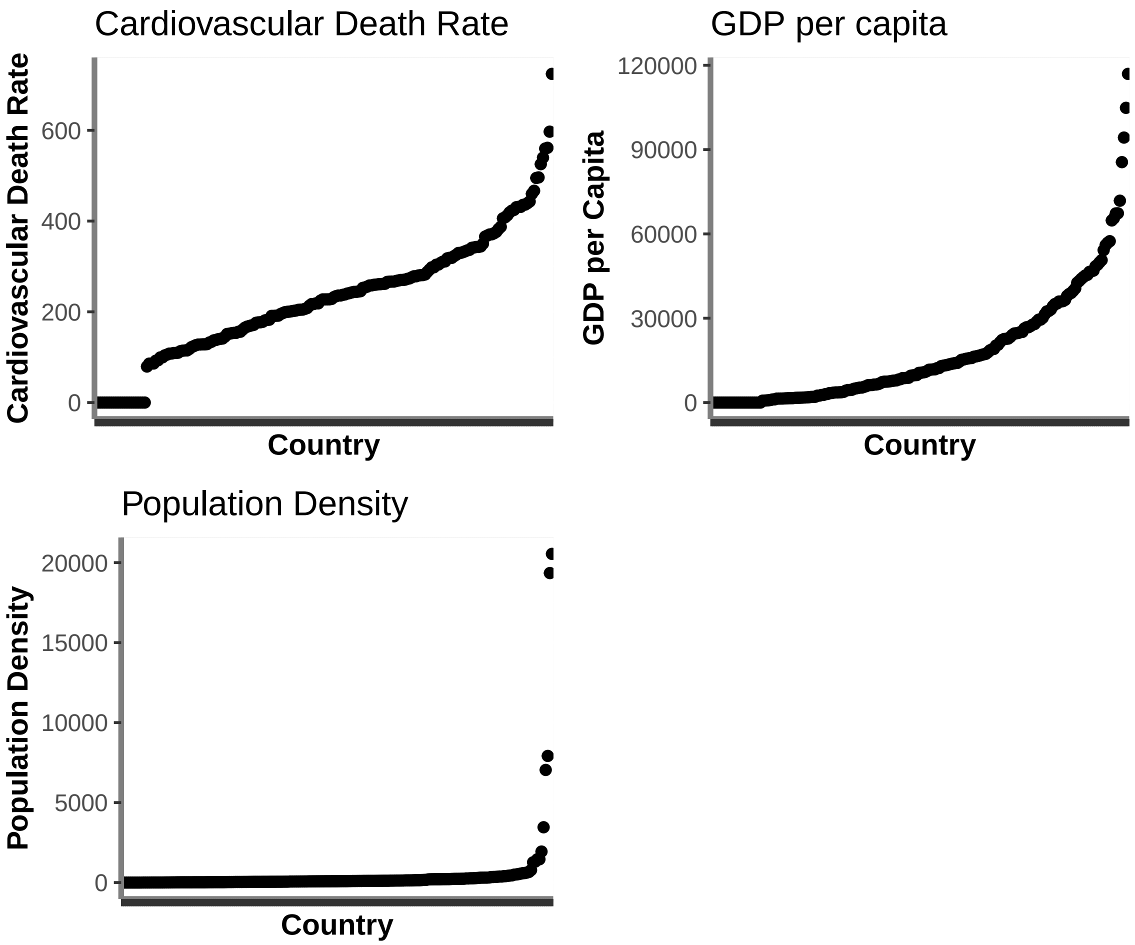
```
world_df <- world_df %>% scale_bigs_scale(.)
```

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`



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

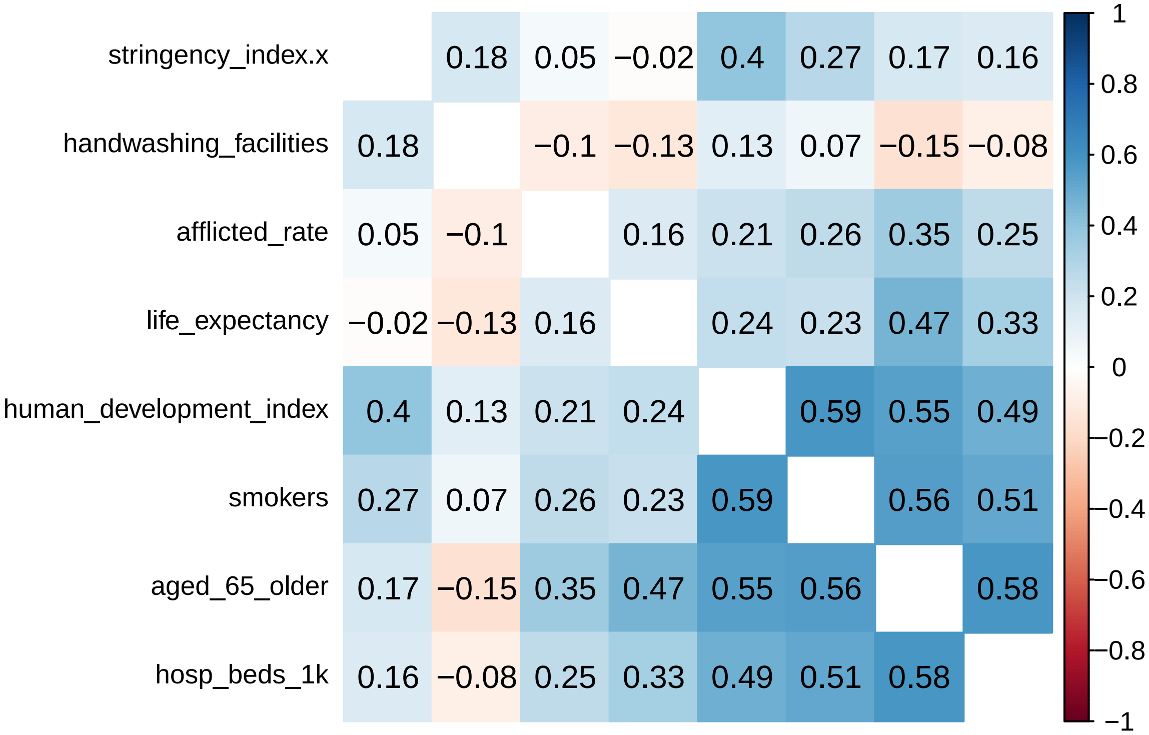
Now we can check all the descriptive stats for all the columns

```
##                                   mean     sd   min      max   range
## afflicted_rate                23.37 88.83  0.00 1479.96 1479.96
## reproduction_rate              0.77  0.44 -0.01    2.06    2.08
## stringency_index              44.81 25.15  0.00   99.06   99.06
## median_age                    27.58 12.79  0.00   48.20   48.20
## aged_65_older                  7.90  6.48  0.00   27.05   27.05
## extreme_poverty                7.83 16.76  0.00   77.60   77.60
## diabetes_prevalence            7.52  4.59  0.00   23.36   23.36
## handwashing_facilities        21.89 32.72  0.00   99.00   99.00
## hosp_beds_1k                   2.38  2.51  0.00   13.80   13.80
## life_expectancy               73.36  9.08  0.00   86.75   86.75
## human_development_index       62.94 28.48  0.00   95.70   95.70
## smokers                       14.38 12.75  0.00   45.95   45.95
## new_vaccinations_cum_per_1000  0.00  1.00 -0.49    4.95    5.45
## new_tests_cum_per_1000         0.00  1.00 -0.27   18.13   18.40
## population_density_norm        0.00  1.00 -0.21    9.60    9.81
## cardiovasc_death_rate_norm     0.00  1.00 -1.67    3.69    5.36
## gdp_per_capita_log             8.21  3.22  0.00   11.67   11.67
```

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.

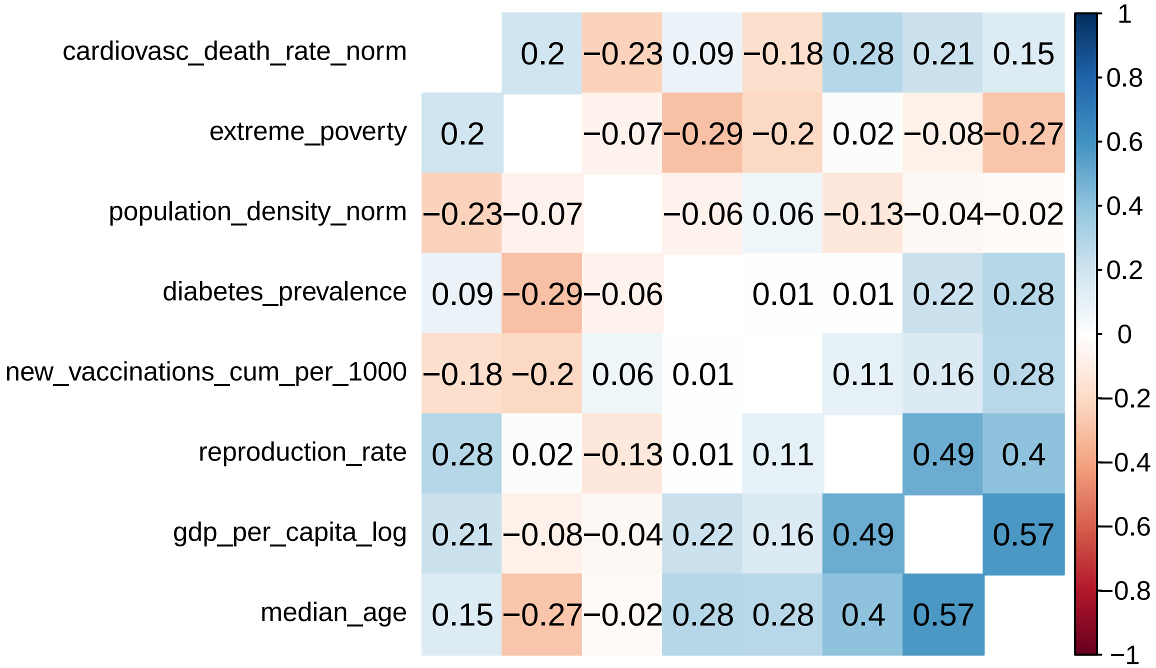
```r
world_df <- world_df %>%
    group_by(location) %>%
    mutate(across(date, function(x) floor_date(x, unit = "quarters"))) %>%
    ungroup()

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



stringency_index.x		0.18	0.05	−0.02	0.4	0.27	0.17	0.16
handwashing_facilities	0.18		−0.1	−0.13	0.13	0.07	−0.15	−0.08
afflicted_rate	0.05	−0.1		0.16	0.21	0.26	0.35	0.25
life_expectancy	−0.02	−0.13	0.16		0.24	0.23	0.47	0.33
human_development_index	0.4	0.13	0.21	0.24		0.59	0.55	0.49
smokers	0.27	0.07	0.26	0.23	0.59		0.56	0.51
aged_65_older	0.17	−0.15	0.35	0.47	0.55	0.56		0.58
hosp_beds_1k	0.16	−0.08	0.25	0.33	0.49	0.51	0.58	

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



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

The components that will form a part of the larger OLS regression is therefore the - stringency index `stringency_index.x`

- average life expectancy `life_expectancy`
- 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

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

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

mod_1sls <- plm(stringency_index.x ~ reproduction_rate
                + new_vaccinations_cum_per_1000,
                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 + diabetes_prevalence
   + life_expectancy + human_development_index + hosp_beds_1k,
   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))
```

Table 4.1

	Dependent variable:		
	afflicted_rate		
	(1)	(2)	(3)
stringency_index.x	0.235***	−0.086	
	(0.071)	(0.077)	
stringency_hat			−0.363**
			(0.185)
smokers		0.598*	0.596*
		(0.330)	(0.328)
handwashing_facilities	−0.297**	−0.140	−0.139
	(0.136)	(0.128)	(0.129)
aged_65_older		3.865***	3.841***
		(0.798)	(0.800)
diabetes_prevalence		−1.085*	−1.170*
		(0.650)	(0.664)
life_expectancy		−0.055	−0.112
		(0.228)	(0.239)
human_development_index		0.042	0.107
		(0.145)	(0.145)
hosp_beds_1k		1.499	1.628
		(1.749)	(1.774)
Constant	18.558***	−2.965	10.755
	(4.213)	(15.943)	(19.295)
Observations	1,844	1,844	1,844
R^2	0.014	0.139	0.140
Adjusted R^2	0.012	0.135	0.136
F Statistic	12.659*** (df = 2; 1841)	37.040*** (df = 8; 1835)	37.244*** (df = 8; 1835)
Note:			*p<0.1; **p<0.05; ***p<0.01

4.2. Fixed- and Random Effects Regression

```
mod_fe_1 <- plm(afflicted_rate ~ stringency_index.x + smokers
   + handwashing_facilities + aged_65_older + diabetes_prevalence
   + life_expectancy + human_development_index + hosp_beds_1k,
   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 + diabetes_prevalence
                   + human_development_index
                  + reproduction_rate,
                   data = world_df,
                   index = c("location"),
                  model = "random")

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

```
stargazer(mod_fe_1, mod_re_1, header = F, font.size = "small", se = list(robustse_re1))
```

Table 4.2

	Dependent variable:	
	afflicted_rate	
	(1)	(2)
stringency_index.x	−0.279	−0.122
	(0.104)	(0.107)
smokers		0.694**
		(0.331)
handwashing_facilities		−0.129
		(0.102)
aged_65_older		3.923***
		(0.653)
diabetes_prevalence		−1.301*
		(0.709)
human_development_index		0.226
		(0.165)
reproduction_rate		−15.521**
		(6.856)
Constant		−0.982
		(9.131)
Observations	1,844	1,844
R^2	0.003	0.060
Adjusted R^2	−0.123	0.057
F Statistic	4.371** (df = 1; 1637)	117.832***
Note:	*p<0.1; **p<0.05; ***p<0.01	

It is important to note that with a lot of Covid data, the reliability of measurement error and false estimates is questionable. The fixed, and random effects might exacerbate this problem. In this case, it seems that OLS pooled regression, might perform better in explaining the behaviour in the defined 'afflicted_rate' variable.

5. References

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

6. 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)
```

6.1. Functional Code

```
## 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_text(siz
##         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)
## }
## <bytecode: 0x000001b91772aee0>
```

```
## 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,
##         reproduction_rate, new_tests_cum_per_1000, new_vaccinations_cum_per_1000,
##         median_age, extreme_poverty, diabetes_prevalence)) %>%
##         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,
##         reproduction_rate, new_vaccinations_cum_per_1000, median_age,
##         extreme_poverty, diabetes_prevalence)) %>% 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: 0x000001b9024fb030>
```

```
## 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: 0x000001b9247548e8>
```

```
## 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")
##     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: 0x000001b91219a2e0>


## 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)/new_cases) * 100, .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: 0x000001b910ba3f90>


## 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, population)) %>%
##         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)
## }
```