COVID-19 Evolution in EU Members

Data source

This COVID-19 data repository is developed by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.
Data are retrieved from government/authority sources and basically include the evolution of cases and deaths on a daily basis for almost all countries in the world. The covered timespan starts from January 2020 to the current day (28 November 2021 at time of writing). For this analysis I focused on countries that are currently part of the European Union (EU).

Preliminary steps

The first step is to load all the libraries needed to wrangle the data and carry on the analysis.

```
library(tidyverse)
library(magrittr)
library(lubridate)
library(dplyr)
```

Get current COVID19 data from the two files I need, adding each file name to the common url variable.

```
url_in <-
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
file_names <- c(
  "time_series_covid19_deaths_global.csv",
  "time_series_covid19_confirmed_global.csv")
urls <- str_c(url_in,file_names)
```

Then, all datasets must be saved into dataframes.

```
global_cases <- read_csv(urls[2])
global_deaths <- read_csv(urls[1])
```

Analysis 1: Evolution of cumulative cases and deaths in EU

In order to tidy up the data, columns `Long` and `Lat` have been deleted since they are not useful to the analysis I want to carry on. Moreover I do not need one column for each date, so I re-format the dataframes of both global deaths and cases.

```
global_cases <- global_cases %>% pivot_longer(
  cols = -c (`Province/State`, `Country/Region`, Lat, Long),
  names_to = "date", values_to = "cases") %>% select(-c(Lat,Long))
global_deaths <- global_deaths %>% pivot_longer(
  cols = -c (`Province/State`, `Country/Region`, Lat, Long),
  names_to = "date", values_to = "deaths") %>% select(-c(Lat,Long))
```

Then, it is time to join the deaths and cases dataframes in a single one called global, in which I will combine the cases into deaths per date. In addition, I rename the two columns for Country and Province to get rid of the `/` character and lastly I set column `date` as a Date object using `mdy` function.

```
global <- global_cases %>% full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`, Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

It may be useful to look at the summary of the newly-created dataframe.

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:190120      Length:190120      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-07-09   1st Qu.:     174
##  Mode  :character   Mode  :character   Median :2020-12-26   Median :    3158
##                                        Mean   :2020-12-26   Mean   :  348480
##                                        3rd Qu.:2021-06-14   3rd Qu.:   68150
##                                        Max.   :2021-11-30   Max.   :48554890
##      deaths
##  Min.   :     0
##  1st Qu.:     1
##  Median :    48
##  Mean   :  7761
##  3rd Qu.:  1175
##  Max.   :780140
```

In order to delete the many rows that have 0 cases, I filtered the dataset to retrieve only those rows having cases bigger than 0.

```
global <- global %>% filter(cases > 0)
```

To have a more compact view of each row I collapsed the `Province_State` and `Country_Region` columns into a single one called `Combined_Key`. This will be useful when I will add the population for each Country, since I will consider a new dataset containing exactly the `Combined_Key` column.

```
global <- global %>%
  unite("Combined_Key", c(Province_State, Country_Region),
        sep = ", ", na.rm = TRUE, remove = FALSE)
```

Then, it is useful to add the population for each country. To do so I download the data from the url below and put its content in a new dataset getting rid of the columns I surely do not need.

```
uid_lookup_url <-
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_ISO_FIPS_Lool
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

Next, I join together the `global` and the `uid` dataset to have a unique datsaet with the Population column for each country.

```
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths, Population,
         Combined_Key)
```

I want to focus my analysis on the Countries that are currently members of the European Union (EU). Therefore I create a vector containing all the members, then filter the global dataframe to get only those rows having `Combined_Key` values matching the vector. It is important to notice that in my analysis only the Europe mainland territories are considered (i.e. no overseas or administered territories outside continental Europe).

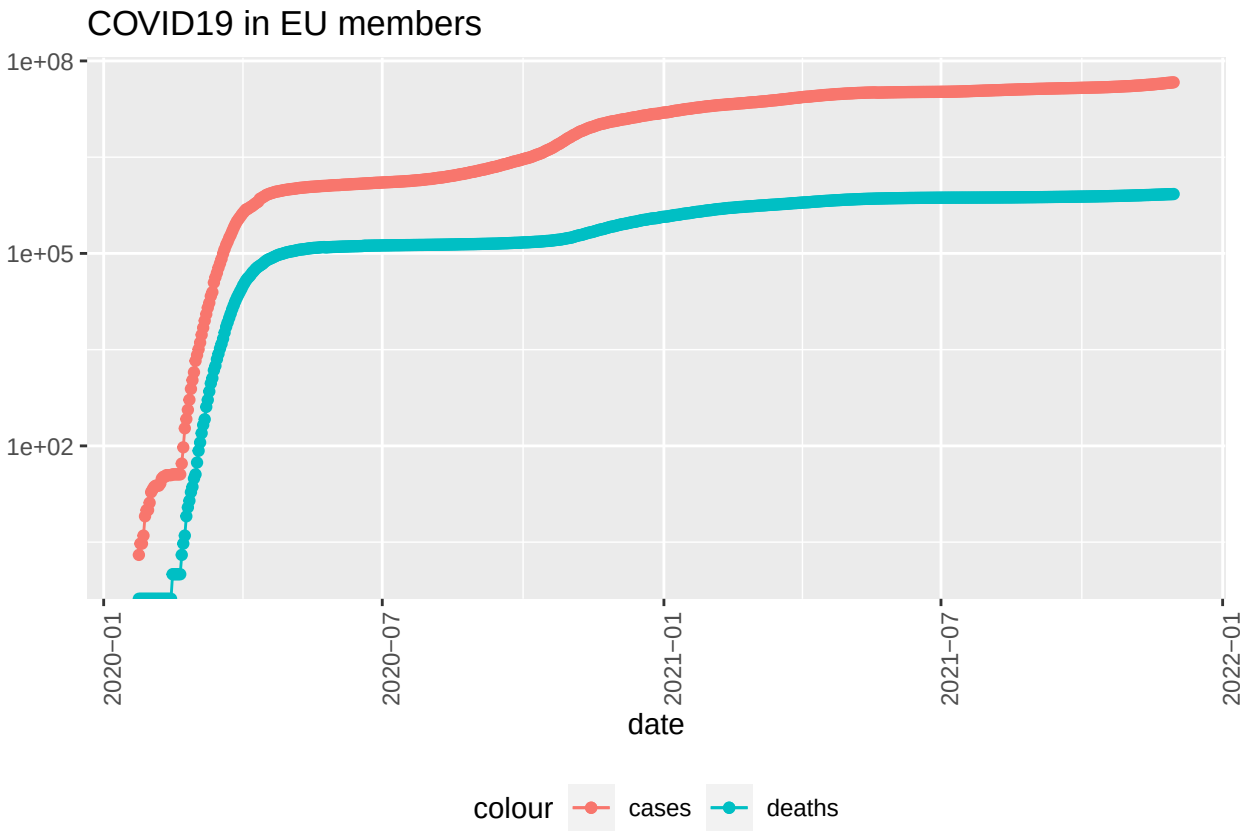
```
ue_members <- c("Austria","Belgium","Bulgaria","Croatia","Cyprus","Czechia","Denmark",
                "Estonia","Finland","France","Germany","Greece","Hungary","Ireland",
                "Italy","Latvia","Lithuania","Luxembourg","Malta","Netherlands","Poland",
                "Portugal","Romania","Slovakia","Slovenia","Spain","Sweden")
europe <- filter(global, Combined_Key %in% ue_members)
```

First of all I want to add to my analysis the parameter `deaths_per_mill`, expressing the deaths per million for each EU member on each tracked day. I also group the data by date to have the cumulative total cases and deaths for all EU on each day.

```
europe_totals <- europe %>%
  group_by(date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  ungroup()
```

To visualize some data I plot the daily evolution of the cumulative cases in EU over the whole timespan.

```
europe_totals %>% filter(cases >0) %>% ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in EU members", y = NULL)
```



The x-axis represents the covered days while the y-axis depicts the cumulative number of cases and deaths up to that day. Anyway this visualization only tells us the general behavior of the cumulative cases and deaths over the observed time range. Indeed, even if at different paces, the evolution is continuously growing.

Analysis 2: Evolution of cases and deaths daily increase in EU and Italy

Now I am interested in observing the daily increase on a daily basis, rather than a cumulative count. As a comparison I create a new dataframe extracting the same data as those in `europe`, but focusing on Italy only.

```
italy <- filter(europe, Combined_Key == "Italy") %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population)
```

Then I add two new columns on all my newly created dataframes to keep track of the daily increase.

```
europe <- europe %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths))
europe_totals <- europe_totals %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths))
italy <- italy %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths))
```

In order to check the coherence of these data I issue a `summary` command on all dataframes.

```
summary(europe)
```

```
##  Province_State     Country_Region          date                cases
##  Length:17488       Length:17488       Min.   :2020-01-24   Min.   :      1
##  Class :character   Class :character   1st Qu.:2020-08-02   1st Qu.:  18013
##  Mode  :character   Mode  :character   Median :2021-01-11   Median : 163018
##                                        Mean   :2021-01-10   Mean   : 664562
##                                        3rd Qu.:2021-06-22   3rd Qu.: 738664
##                                        Max.   :2021-11-30   Max.   :7486179
##
##      deaths          Population       Combined_Key         new_cases
##  Min.   :     0   Min.   :  441539   Length:17488       Min.   :-7486178
##  1st Qu.:   403   1st Qu.: 2722291   Class :character   1st Qu.:      73
##  Median :  3752   Median : 9006400   Mode  :character   Median :     477
##  Mean   : 15485   Mean   :16794527                      Mean   :      69
##  3rd Qu.: 17050   3rd Qu.:17134873                      3rd Qu.:    2268
##  Max.   :133828   Max.   :83155031                      Max.   :  117900
##                                                         NA's   :1
##    new_deaths
##  Min.   :-133828.00
##  1st Qu.:      0.00
##  Median :      6.00
##  Mean   :      0.87
##  3rd Qu.:     36.00
##  Max.   :   1734.00
##  NA's   :1
```

```
summary(europe_totals)
```

```
##       date                cases              deaths         Population
##  Min.   :2020-01-24   Min.   :       2   Min.   :     0   Min.   : 65249843
##  1st Qu.:2020-07-11   1st Qu.: 1310887   1st Qu.:134223   1st Qu.:444545974
##  Median :2020-12-27   Median :14963582   Median :354681   Median :444545974
##  Mean   :2020-12-27   Mean   :17166697   Mean   :399992   Mean   :433829665
##  3rd Qu.:2021-06-14   3rd Qu.:32756089   3rd Qu.:732213   3rd Qu.:444545974
##  Max.   :2021-11-30   Max.   :46383613   Max.   :842280   Max.   :444545974
##
##  deaths_per_mill    new_cases          new_deaths
##  Min.   :   0.0   Min.   :-302748   Min.   :-1583.0
##  1st Qu.: 301.9   1st Qu.:  11477   1st Qu.:  197.8
##  Median : 797.9   Median :  49460   Median :  725.5
##  Mean   : 899.8   Mean   :  68615   Mean   : 1246.0
##  3rd Qu.:1647.1   3rd Qu.: 111398   3rd Qu.: 2102.2
##  Max.   :1894.7   Max.   : 315236   Max.   : 5017.0
##                   NA's   :1         NA's   :1
```

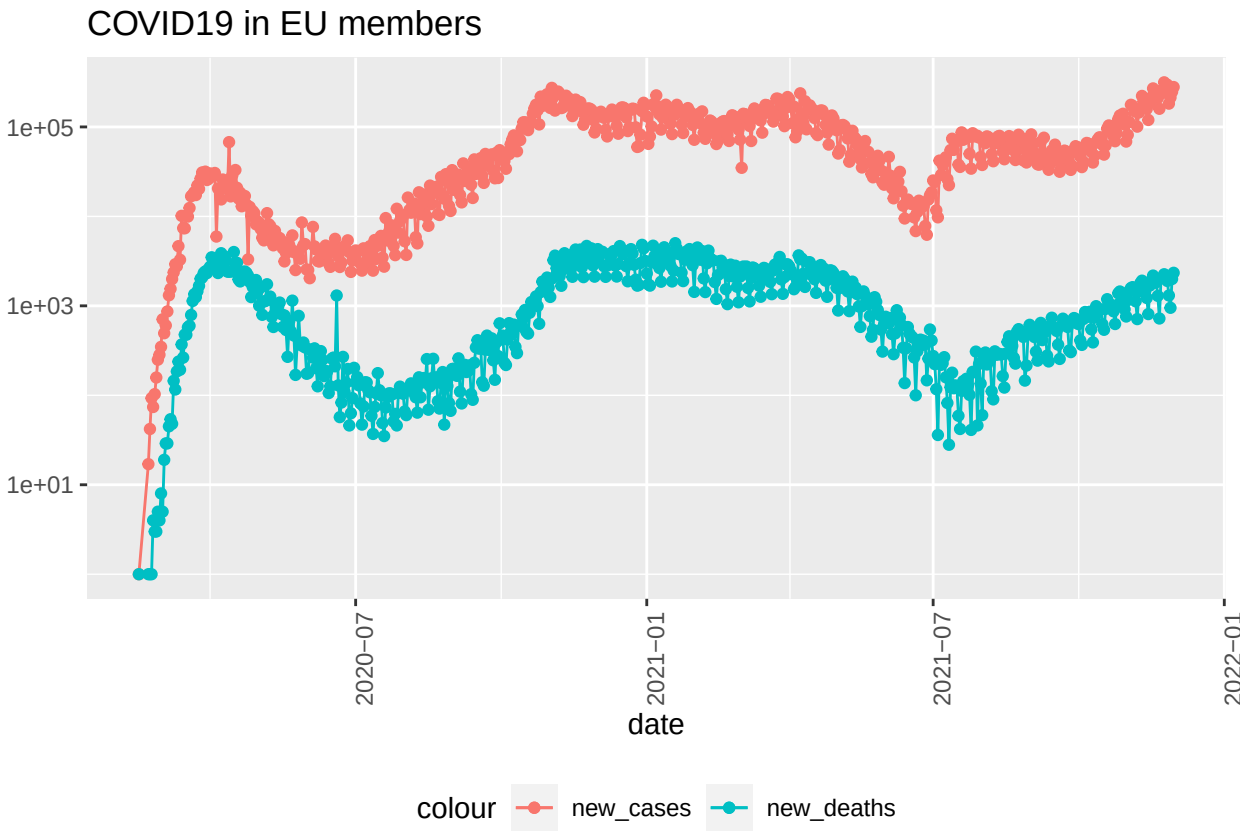
```
summary(italy)
```

```
##  Province_State     Country_Region          date                cases
##  Length:670         Length:670         Min.   :2020-01-31   Min.   :     2
##  Class :character   Class :character   1st Qu.:2020-07-16   1st Qu.: 243794
```

```
##  Mode  :character   Mode  :character   Median :2020-12-30   Median :2095428
##                                        Mean   :2020-12-30   Mean   :2200820
##                                        3rd Qu.:2021-06-15   3rd Qu.:4248082
##                                        Max.   :2021-11-30   Max.   :5028547
##
##      deaths          Population       Combined_Key       deaths_per_mill
##  Min.   :     0   Min.   :60461828   Length:670         Min.   :   0.0
##  1st Qu.: 35020   1st Qu.:60461828   Class :character   1st Qu.: 579.2
##  Median : 73882   Median :60461828   Mode  :character   Median :1222.0
##  Mean   : 74597   Mean   :60461828                      Mean   :1233.8
##  3rd Qu.:127140   3rd Qu.:60461828                      3rd Qu.:2102.8
##  Max.   :133828   Max.   :60461828                      Max.   :2213.4
##
##    new_cases       new_deaths
##  Min.   : -148   Min.   :-31
##  1st Qu.: 1147   1st Qu.: 23
##  Median : 4092   Median : 70
##  Mean   : 7517   Mean   :200
##  3rd Qu.:12450   3rd Qu.:352
##  Max.   :40902   Max.   :993
##  NA's   :1       NA's   :1
```

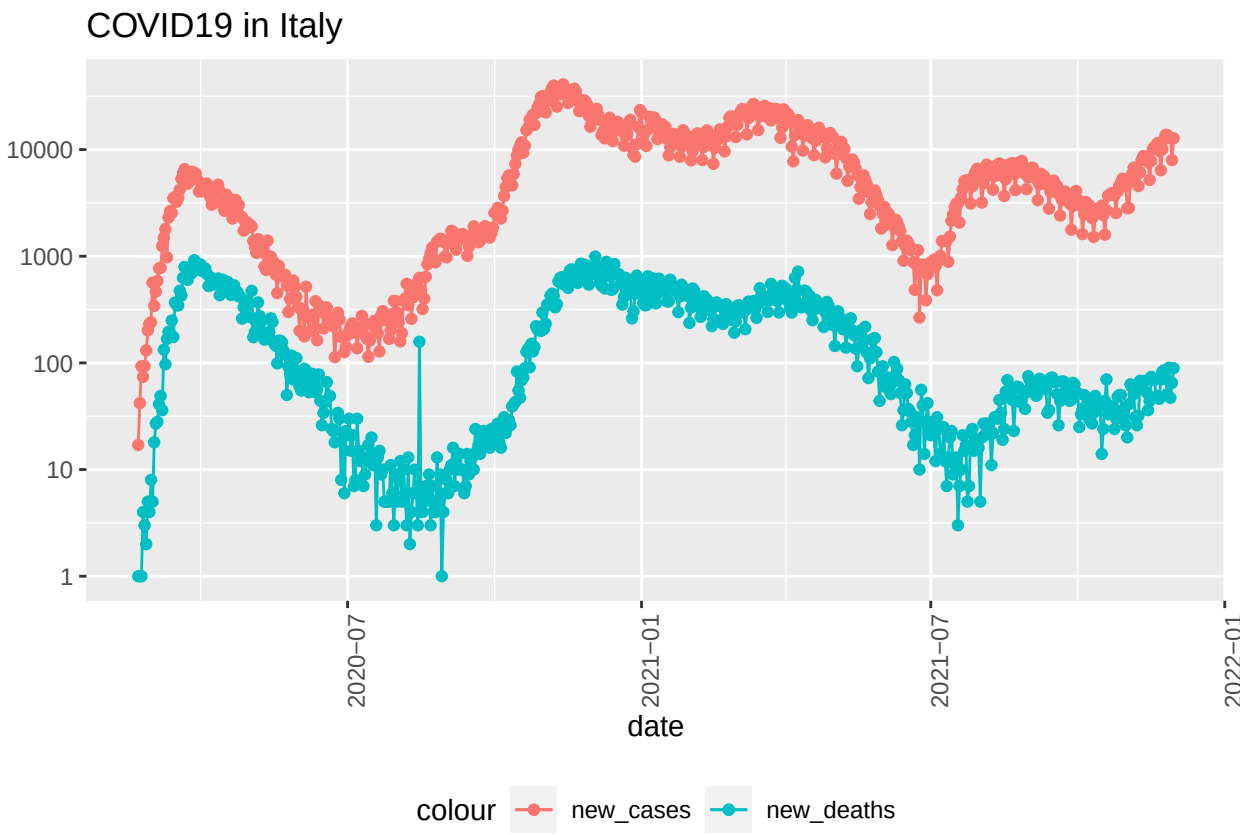
It appears that, for some days, the variables `new_cases` and `new_deaths` are negative, and this is very likely due to a typo in the cumulative variables `cases` and `deaths`. For this reason I will consider only the rows having positive values when plotting data.
Here I plot the daily evolution of `new_cases` and `new_deaths`.

```
europe_totals %>% filter(new_cases > 0, new_deaths > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in EU members", y = NULL)
```



In both curves it is easy to see the lower incidence of new cases and new deaths over the two summer periods of the timespan , while the curve increases steadily as winter approaches. The number of cases per day leveled off around 100000 cases during the first months of 2021 and after a decrease in July is now back to its previous level. The same behaviour holds for the number of deaths per day as well with smaller values between 1000 and 10000.
Now I am interested in evaluate the same graph for a single state, in this case Italy.

```
italy %>% filter(new_cases > 0, new_deaths > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in Italy", y = NULL)
```



In this graph it turns out that both the new cases and new deaths faced a lower increase comparing the same autumn period, over which the incidence of these two variables generally grows. In particular, the current trend of new deaths appears to be significantly lower if compared to the same period of 2020. This may be explained with an increasing percentage of vaccinated people in this Country.

Linear model: Deaths as function of cases

Now I start manipulating the dataframes to have at the and a new dataset grouped by states along with their total cases and death (both cumulatve and per-thousand). All this steps will be useful for the linear model I want to derive.

```
europe_by_state <- europe %>%
  group_by(Combined_Key,date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population,
         cases_per_mill = cases * 1000000 / Population) %>%
  ungroup()
```

It is time to build a new dataset including the new variables `cases_per_thou` and `deaths_per_thou`, indicating the cases and the deaths per one thousand people.

```
europe_state_totals <- europe_by_state %>%
  group_by(Combined_Key) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_per_thou = 1000 * cases / population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases >0, population >0)
```

To derive this linear model I include the columns `deaths_per_thou` and `cases_per_thou` to explore if the number of deaths per thousand can be linearly predicted by the number of cases per thousand.

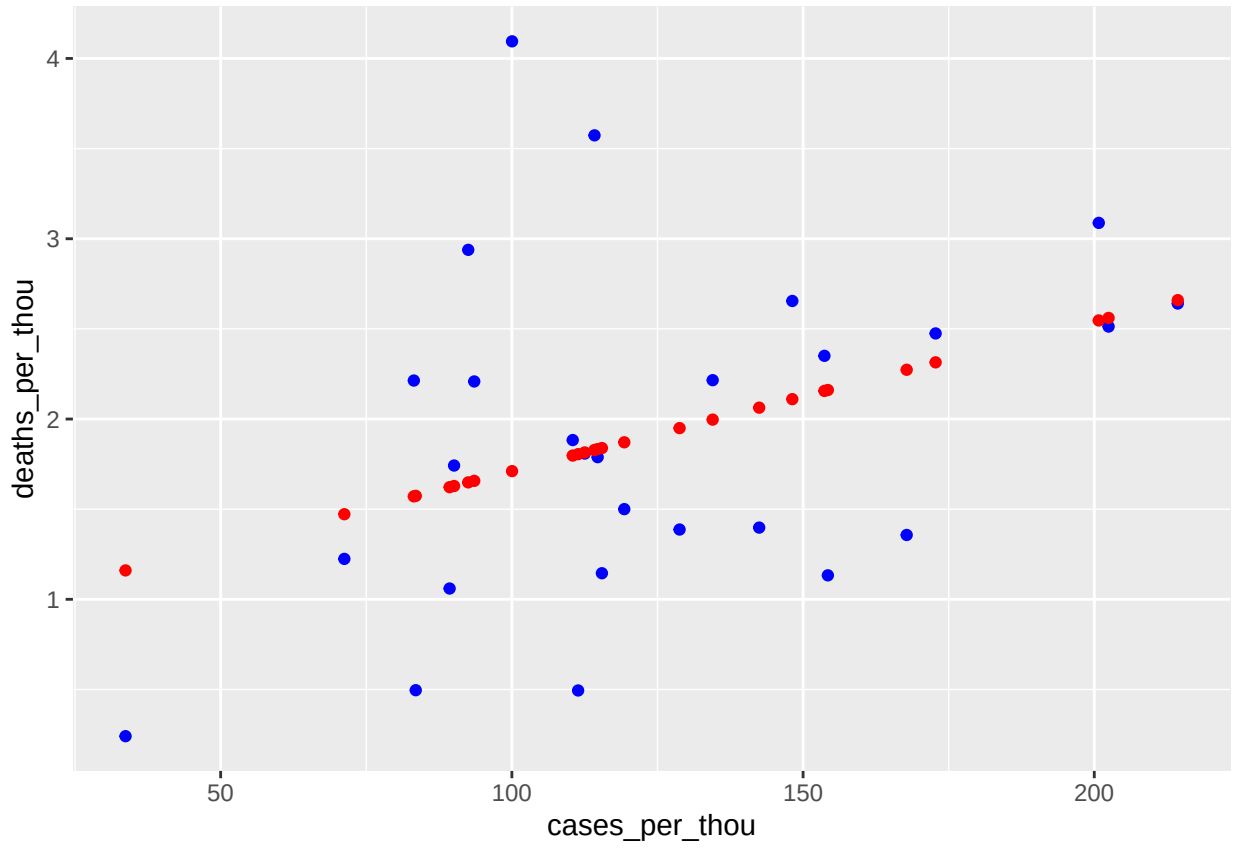
```
mod <- lm(deaths_per_thou ~ cases_per_thou, data = europe_state_totals)
summary(mod)
```

Here I create a new dataframe to include the column with the predicted deaths per thousand according to the linear model evaluated before.

```
europe_state_totals_pred <- europe_state_totals %>% mutate(pred = predict(mod))
```

Now I plot the actual data (as blue dots) in comparison to the predicted ones (as red dots).

```
europe_state_totals_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



The model seems to follow a linearity even though some outliers cannot remain unnoticed. For instance Bulgaria has a death per thousand around 4 times the predicted value, while Finland has an actual value which is almost 5 times smaller than the one predicted by the model.

Identification of bias

Within this kind of analysis, possible sources of bias may be related to political, social or economical factors. To overcome any bias of this kind all EU states have been equally and objectively considered throughout the analysis, regardless of their government orientation as well as their economical situation. As an example of another type of bias, one may think that more cases necessarily means more deaths, but this is not always true. This may be due to other factors, such as the efficiency of vaccination campaigns or the effectiveness of measures against pandemic like personal protection or social distancing.