Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Δ.Π.Μ.Σ. Επιστήμης Δεδομένων
και Μηχανικής Μάθησης

Προγραμματιστικά Εργαλεία και Τεχνικές για Επιστήμη Δεδομένων

Exploratory Data Analysis Using R

Νικόλαος Μανιάτης
Μεταπτυχιακός Φοιτητής
Αριθμός Μητρώου: 03400097

Επιβλέπων: Δημήτρης Φουσκάκης
Καθηγητής Ε.Μ.Π.

January 12, 2021

First Section

In the first section of our data analysis, we will perform the ten tasks mentioned in our assignment.

```
library(data.table);library(ggplot2);library(dplyr);library(lubridate)

set1 <- fread('https://raw.githubusercontent.com
/CSSEGISandData/COVID-19/master/csse_covid_19_data/
csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
set2 <- fread('https://raw.githubusercontent.com
/CSSEGISandData/COVID-19/master/csse_covid_19_data
/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')
############TASK 1:############ Remove columns:
set1 <- set1[, !c('Province/State','Lat','Long')]
set2 <- set2[, !c('Province/State','Lat','Long')]
############TASK 2:############ Convert data from wide to long format.
dt1 <- melt(set1, id.vars = "Country/Region")
dt2 <- melt(set2, id.vars = "Country/Region")
############TASKS 3-4:############ Rename Variables
setnames(dt1, c("Country/Region","variable","value"),
    c("Country","Date","Confirmed"))
setnames(dt2, c("Country/Region","variable","value"),
    c("Country","Date","Deaths"))
############TASK 5:############ Convert date format
dt1$Date <- mdy(dt1$Date)
dt2$Date <- mdy(dt2$Date)
############TASK 6:############ Group by country and date
dt1 <- dt1[order(Country, Date)]
dt2 <- dt2[order(Country,Date)]
############TASK 7:############ Merge the two datasets into one.
#SUM OVER COUNTRIES AND DATES
#SO WE DONT HAVE MULTIPLE ENTRIES FOR THE SAME DAY AND COUNTRY
#BECAUSE WE DROPPED THE STATES VARIABLE
dt1 <- dt1[,sum(Confirmed), by = .(Country, Date)]
dt2 <- dt2[,sum(Deaths), by = .(Country, Date)]
#Since the two datasets have the same 2 columns,
#and are grouped the same way,
#we can just bind the deaths column.
dt <- cbind(dt1, dt2$V1)
names(dt) <- c("Country","Date","Confirmed","Deaths")
############TASK 8:############ Calculate counts for the whole world
#this keeps only the latest date for every country,
#because it has the maximum confirmed cases and deaths
grouped_<- dt %>% group_by(Country) %>% slice_max(order_by = Date,n = 1)
grouped_ <- as.data.table(grouped_)
#so we use that variable to sum over all of the countries
#and get a number for the counts and deaths all over the world
```

```
dt_clean <- as.data.table(grouped_)
world <- apply(dt_clean[,3:4],2,sum)
############TASK 9:############ sort (again) by country and date.
dt <- dt[order(Country, Date)]
############TASK 10:############ Create two extra variables:
#confirmed.ind and deaths.inc with the daily
#confirmed cases and daily deaths respectively
#calculate the daily confirmed cases and deaths
dt$Confirmed.ind <- dt$Confirmed - lag(dt$Confirmed, n = 1)
dt$Deaths.inc <- dt$Deaths - lag(dt$Deaths, n = 1)
#set the first daily confirmed cases and deaths as zero,
#because the lag function calculates another country's numbers.
dt[Date == '2020-01-22']$Confirmed.ind <- 0
dt[Date == '2020-01-22']$Deaths.inc <- 0

#We don't lose any important data this way,
#with: dt %>% filter(Date=='2020-01-22') %>% filter(Confirmed>0)
#only 6 countries had non-zero confirmed cases
#and only China had more than 2.
```

On task 8 we calculate counts for the whole world, that is, a number for the worldwide confirmed cases and one for the worldwide confirmed deaths. As of 11 January 2021, the worldwide confirmed cases are 90,281,429 while the worlwide deaths are 1,934,791.

Just to take a peek of how our data looks like on its final form, we show the results for our Country, and then the whole dataset.

```
>tail(dt[Country == 'Greece'])
   Country       Date Confirmed Deaths Confirmed.ind Deaths.inc
1:  Greece 2021-01-05    141453   5051           927         40
2:  Greece 2021-01-06    142267   5099           814         48
3:  Greece 2021-01-07    142777   5146           510         47
4:  Greece 2021-01-08    143494   5195           717         49
5:  Greece 2021-01-09    144293   5227           799         32
6:  Greece 2021-01-10    144738   5263           445         36
```

```
          Country       Date Confirmed Deaths Confirmed.ind Deaths.inc
    1: Afghanistan 2020-01-22         0      0             0          0
    2: Afghanistan 2020-01-23         0      0             0          0
    3: Afghanistan 2020-01-24         0      0             0          0
   ---
67803:    Zimbabwe 2021-01-08     19660    468           985         22
67804:    Zimbabwe 2021-01-09     20499    483           839         15
67805:    Zimbabwe 2021-01-10     21477    507           978         24
```

Throughout this paper, we refer to this data table (dt) as defined above.

Second Section

1. Worldwide Perspective

In this section, we begin our exploratory data analysis of the Covid-19 data. We will begin with a global perspective of the situation, starting with plot 1, a world map heatmap about the cumulative cases that each country has.

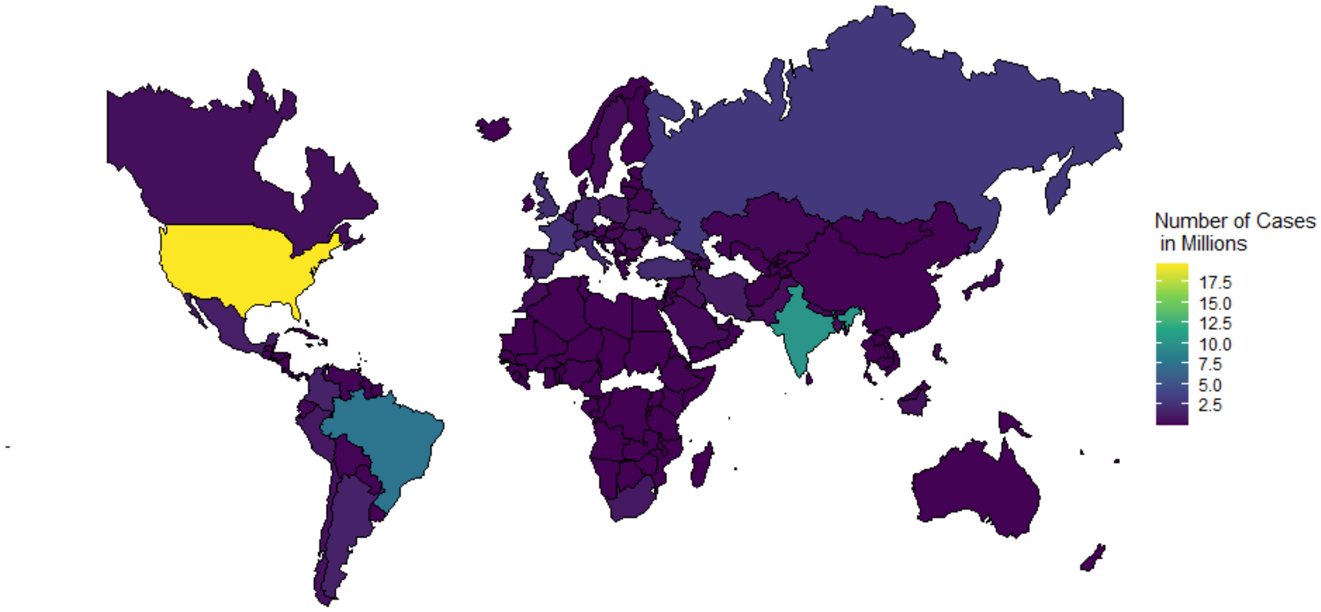

Fig. 1: Heatmap for Cumulative Cases

This gives us only a rough estimate of the confirmed cases, and at a first glance we can see that USA,Brazil and India have the most cases. This plot was generated by the following code:

```
library(ggplot2);library(data.table);library(lubridate);library(dplyr)
library(grid);library(rworldmap);
worldMap <- getMap()
Cases <- dt %>% slice_max(order_by = Date, n = 1)#get max date
#to fix inconsistency between names
Cases[Country == 'US']$Country <- 'United States'
countries <- c(unique(Cases$Country))
#get the countries for which i have data
indworld <- which(worldMap$NAME%in%countries)
coords <- lapply(indworld, function(i){ #extract lat and long
  df <- data.frame(worldMap@polygons[[i]]@Polygons[[1]]@coords)
  df$region =as.character(worldMap$NAME[i])
  colnames(df) <- list("long", "lat", "region");return(df)})
```

```
coords <- do.call("rbind", coords); coords <- as.data.table(coords)
coords <- setnames(coords,'region','Country')
coords <- coords[order(Country)]#order both sets
Cases <- Cases[order(Country)]#to prepare for merging
merged <- merge(coords, Cases, all = TRUE)
breaks <- c(seq(0,max(Cases$Confirmed),2.5e+06))
P <- ggplot() + geom_polygon(data = merged, aes(x = long, y = lat,
  group = Country, fill = Confirmed/1e+06),
  colour = 'black',size = 0.1,show.legend = TRUE) +
  coord_map(xlim = c(-170, 170),  ylim = c(-50, 100))+
  scale_fill_continuous(name = 'Number of Cases \n in Millions',
       type='viridis',breaks=breaks/1e+6)
```

Here on plot 2, we classify our countries by their respective continents, calculate the cumulative cases per continent and plot them for each quarter of the year 2020.

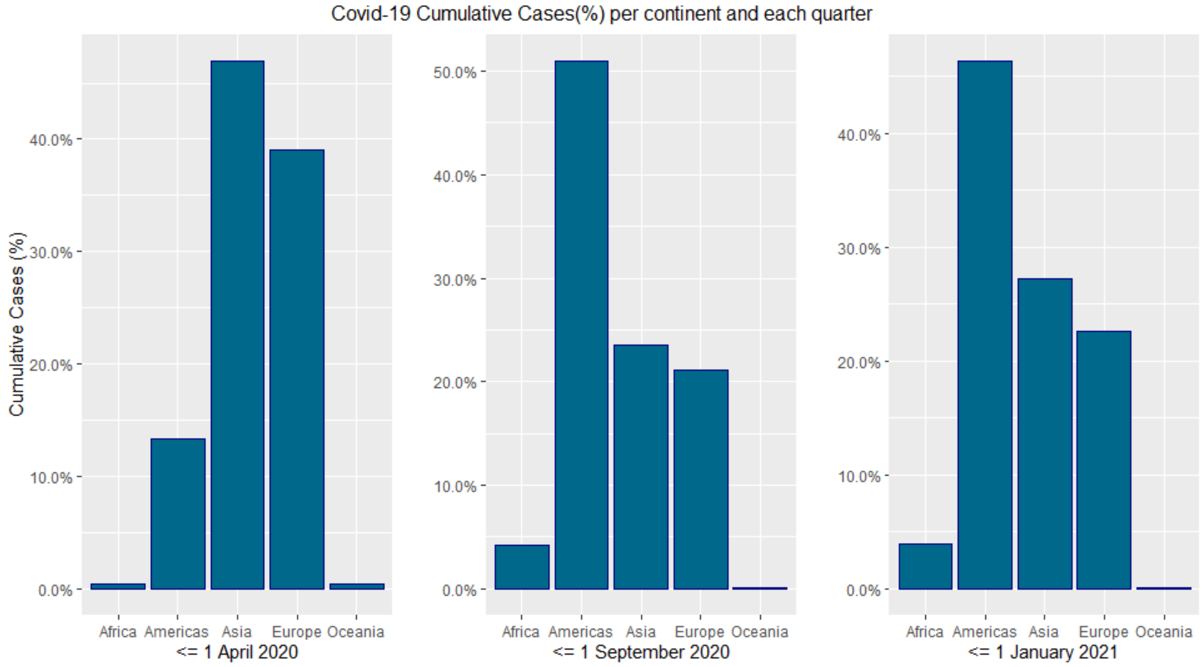

Fig. 2: Barplots for each continent and quarter of 2020.

It is obvious here that on the first quarter, Asia has the most cases worldwide, and we expect that since the virus started from Wuhan in China, but in the second and third quarters America leads the way. This plot is useful because it shows the distribution of confirmed cases on each continent. It's important to note that Africa has a very low percentage and Oceania barely has any[1]. The following histogams is another way to visualize the distribution of cases and deaths on each continent. We can also deduce that the Confirmed cases are about 90 millions, which makes it around 1% of the world population, and deaths are about 1.9

[1] I also made an animation available on this link, that shows the growth of confirmed cases for each continent through time, you can ignore it if it's considered out of topic.

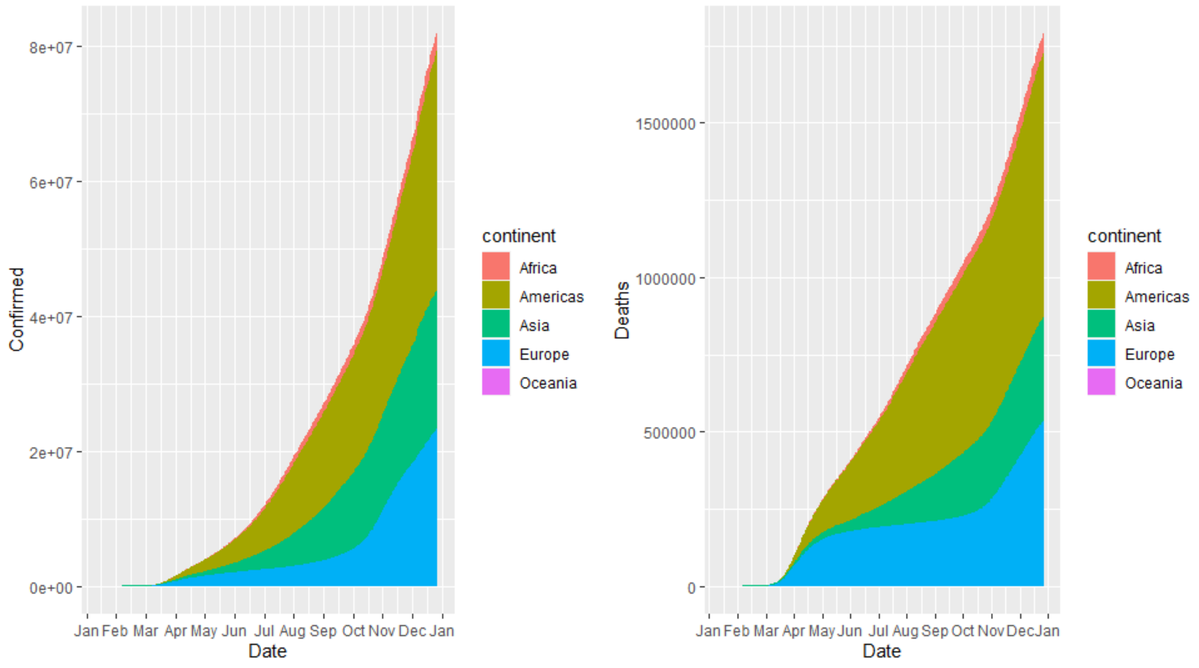

Fig. 3: Histograms showing Cumulative Cases and Deaths per continent.

millions, which makes it the 0.02% of the population. Just to gain some perspective, the Spanish flu of the 1918 pandemic infected about a third of the world's population at the time [2].

```
library(countrycode);library(scales);library(gridExtra)
dt <- dt[Country != 'Diamond Princess']#Not a country!
dt <- dt[Country != 'MS Zaandam']#Not a country!
dt$continent <- countrycode(sourcevar =
    dt$Country, origin = "country.name", destination = 'continent')
dt[Country == 'Kosovo']$continent <- 'Europe'####this was a leftover.
dt <- as.data.table(dt)
dt_1st_qt <- dt[Date <= '2020-04-01']
dt_2nd_qt <- dt[Date <= '2020-09-01']
dt_3rd_qt <- dt[Date <= '2021-01-01']
cont_1 <- dt_1st_qt[,sum(Confirmed), by = .(continent)]
cont_2 <- dt_2nd_qt[,sum(Confirmed), by = .(continent)]
cont_3 <- dt_3rd_qt[,sum(Confirmed), by = .(continent)]
names(cont_1) <- c("Continent","Cumulative_cases")
names(cont_2) <- c("Continent","Cumulative_cases")
names(cont_3) <- c("Continent","Cumulative_cases")
cont_1$Per <- round(cont_1$Cumulative_cases/
                    sum(cont_1$Cumulative_cases),3)
cont_2$Per <- round(cont_2$Cumulative_cases/
                    sum(cont_2$Cumulative_cases),3)
cont_3$Per <- round(cont_3$Cumulative_cases/
                    sum(cont_3$Cumulative_cases),3)#MORE CODE BELOW
```

[2] source: https://en.wikipedia.org/wiki/Spanish_flu

```
bar1 <- ggplot(cont_1) +
  geom_bar(stat = 'identity' ,aes(x = Continent, y = Per),
           color="navyblue", fill="deepskyblue4") +
 xlab("<= 1 April 2020") + ylab("Cumulative Cases (%)") +
   scale_y_continuous(labels=percent)
bar2 <- ggplot(cont_2) +
  geom_bar(stat = 'identity' ,aes(x = Continent, y = Per),
           color="navyblue", fill="deepskyblue4") +
  xlab("<= 1 September 2020")+ylab("")+
  scale_y_continuous(labels=percent)

bar3 <- ggplot(cont_3) +
  geom_bar(stat = 'identity' ,aes(x = Continent, y = Per),
           color="navyblue", fill="deepskyblue4") +
  xlab("<= 1 January 2021")+ ylab("")+ scale_y_continuous(labels=percent)
grid.arrange(bar1,bar2,bar3, nrow = 1,top='Covid-19 Cumulative Cases(%)
per continent and each quarter')

hists <- ggplot(dt, aes(x = Date, y = Confirmed, fill = continent))+
  geom_histogram(stat='identity')+
  scale_x_date(date_breaks = '30 day', date_labels = '%b')
hists2 <- ggplot(dt, aes(x = Date, y = Deaths, fill = continent))+
  geom_histogram(stat='identity')+
  scale_x_date(date_breaks = '30 day', date_labels = '%b')
grid.arrange(hists,hists2,nrow = 1)
```

Below, on plot 4, we have a plot of the total confirmed cases and the total deaths, shown in semi-logarithmic axes.

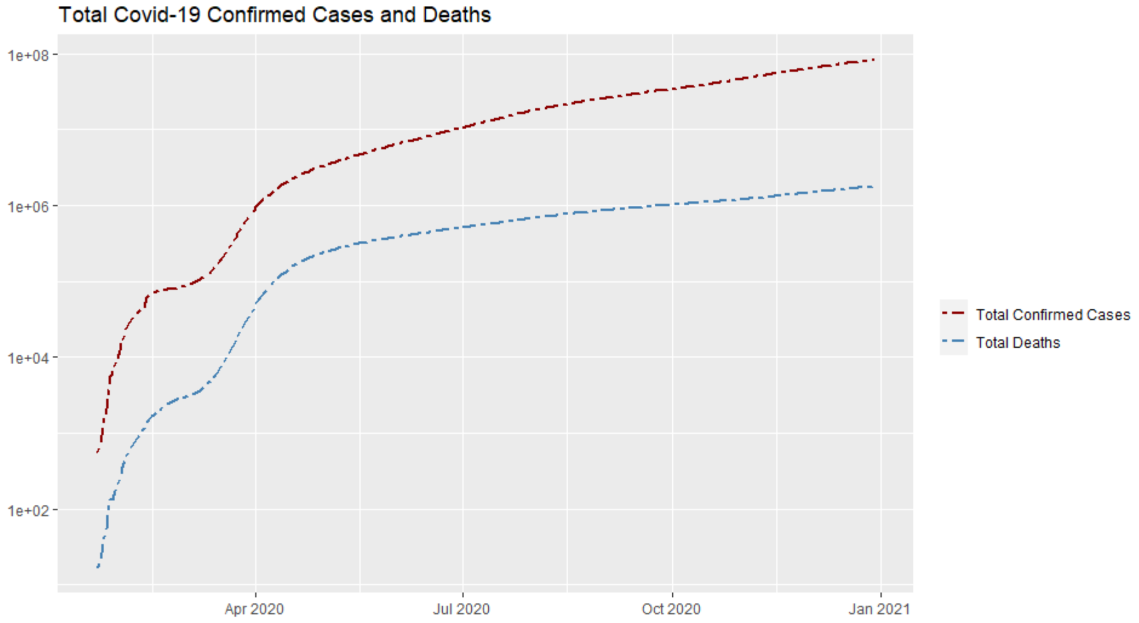

Fig. 4: Total Worldwide Confirmed Cases and Deaths

It is noticed that the total cases and deaths follow a similar trend. This trend, is exponential on the first few months, following the rapid spread of the virus, and is followed by a so-called ”elbow”, which is probably the result of government responses, by putting infected people on quarantines, imposing lockdown in cities etc. The following relaxation of lockdowns and the public’s loosening of precautionary behaviours caused a rise in cases and deaths worldwide, and a second wave followed. At the rest of the year we can see the curve slightly flattening.

There is a straightforward question that most people would like answered. If someone is infected with Covid-19, how likely is that person to die? An attempt to answer this question, is by presenting a fatality rate plot, on plot 5. The Fatality Rate is defined as the ratio between the total deaths and the total confirmed cases. It merely reflects the percentage of the infected people, who end up dying. As it is obvious from the plot, this ratio is not constant and changes over time. It has an average of approximately 4. The probability that someone dies from Covid-19 doesn’t just depend on the disease itself, but also on the treatment they receive, and on the patient’s own ability to recover from it. This means that the Fatality Rate can decrease or increase over time, as responses change; and that it can vary by location and by the characteristics of the infected population, such as age, or sex.

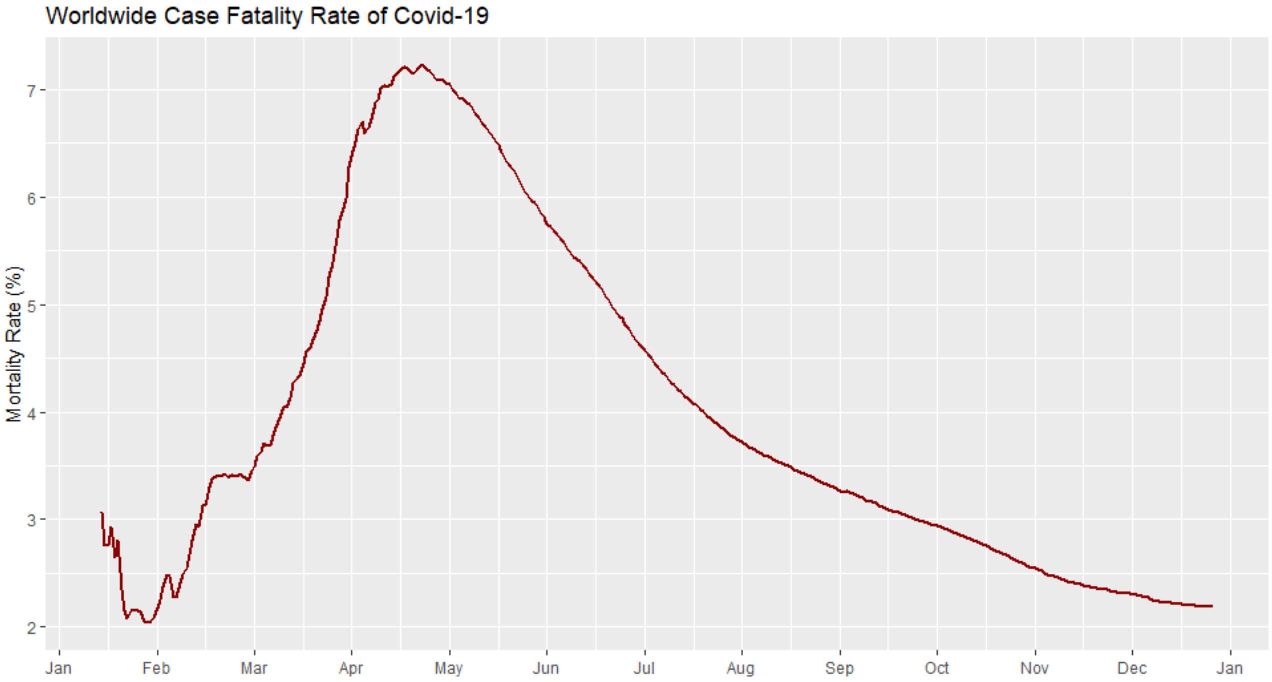

Fig. 5: Covid-19 Fatality Rate

Plots 4 and 5 were generated by the following code.

```
conf <- dt[,sum(Confirmed), by = Date]#Sum - Total Cases
death <- dt[,sum(Deaths),by = Date]#Sum - Total Deaths
all <- cbind(conf,death$V1) #Merge
names(all) <- c("Date","Total_Confirmed_Cases","Total_Deaths")
all$Fatality_Rate <- 100*all$Total_Deaths/all$Total_Confirmed_Cases
all$Date <- as.Date(all$Date)

fatal_rate <- ggplot(all, aes(x=Date, y = Fatality_Rate)) +
      geom_line(linetype = 1, color = 'darkred', size = .9)+
      xlab("") +ylab("Mortality Rate (%)")+
      scale_x_date(date_breaks = '30 day', date_labels = '%b')+
      labs(title = 'Worldwide Case Fatality Rate of Covid-19')
both <-  ggplot(all,aes(x=Date))+
  geom_line(aes(y=Total_Confirmed_Cases,color = 'darkred'),
        size=.9,linetype="twodash") +
  geom_line(aes(y=Total_Deaths,colour='steelblue'),
        size=.9,linetype="twodash")+
  scale_y_log10() + ylab("") +xlab("") +
  labs(title='Total Covid-19 Confirmed Cases and Deaths')+
  scale_color_identity(name="",breaks=c("darkred","steelblue"),
  labels=c("Total Confirmed Cases","Total Deaths"),guide='legend')
```

2. Continent-wide Perspective

After our analysis of the Covid-19 outbreak from a worldwide perspective, it’s time to take a closer look to Europe, which has a population of around 741 millions, which is around the 9.5% of the world population.

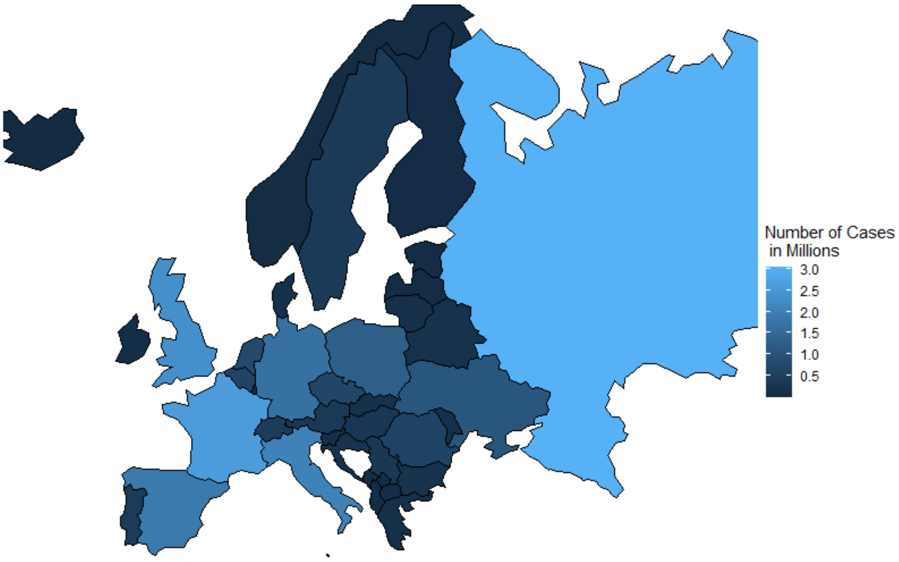

Fig. 6: Europe Heatmap of Cumulative Cases

We begin with plot 6, which is another heatmap, this time just for Europe. It's clear that Russia, England, France, Germany and Italy have the most confirmed cases. The obvious reason behind that is that these countries, are the most populated areas in Europe. The code for this map, is similar to the one in plot 1, and is presented below.

```
library(rworldmap);library(ggplot2);library(dplyr);library(data.table)
worldMap <- getMap()
#we made the continent column on a previous plot
Europe_continent <- dt[continent == 'Europe']#Fix inconsistencies
Europe_continent[Country == 'Czechia']$Country <- 'Czech Rep.'
Europe_continent[Country=='North Macedonia']$Country<-'Macedonia'#Ouch
Europe_cases <- Europe_continent %>% slice_max(order_by = Date, n = 1)
countries <- c(unique(Europe_continent$Country))
indEU <- which(worldMap$NAME%in%countries)
# Extract long and lat border's coordinates of members states of E.U.
europeCoords <- lapply(indEU, function(i){
  df <- data.frame(worldMap@polygons[[i]]@Polygons[[1]]@coords)
  df$region =as.character(worldMap$NAME[i])
  colnames(df) <- list("long", "lat", "region");return(df)})

europeCoords <- as.data.table(do.call("rbind", europeCoords))
europeCoords <- europeCoords[order(region)]
europeCoords <- setnames(europeCoords,'region','Country')
Europe_cases <- Europe_cases[order(Country)]
breaks <- c(seq(0,max(Europe_cases$Confirmed),0.5e+06))
try <- merge(europeCoords, Europe_cases, all = TRUE)
E <- ggplot() + geom_polygon(data = try, aes(x = long, y = lat,
     group = Country, fill = Confirmed/1e+06),
     colour = "black", size = 0.1,show.legend = TRUE) +
     coord_map(xlim = c(-19, 60),  ylim = c(32, 69)) +
     scale_fill_continuous(name = 'Number of Cases \n in Millions',
     breaks=breaks/1e+06)+theme_void()
```

As discussed previously, countries with the highest population have the most Covid-19 cases. So it would be unfair to state that ”X country is doing better than Y country” if they have a different population. As we don't have any data for each country's population, we will normalize counts (and deaths) for each country by substracting the mean for all europe, and dividing by the standard deviation. We will compensate for that in the next section, where we will analyze specific countries with similar population levels.

On plots 7 and 8 we have two barplots for the normalized cases and deaths, respectively. The red bars show the countries with cases (or deaths) above Europe's average, and the blue ones show the ones below Europe's average.

From plot 8, it's interesting to notice that Italy has the highest death toll in Europe. The fact that Italy's population is one of the oldest populations in the world [3], may explain the high amount of deaths, but it depends on other factors too, such as the effectiveness of the government responses, the healthcare system etc.

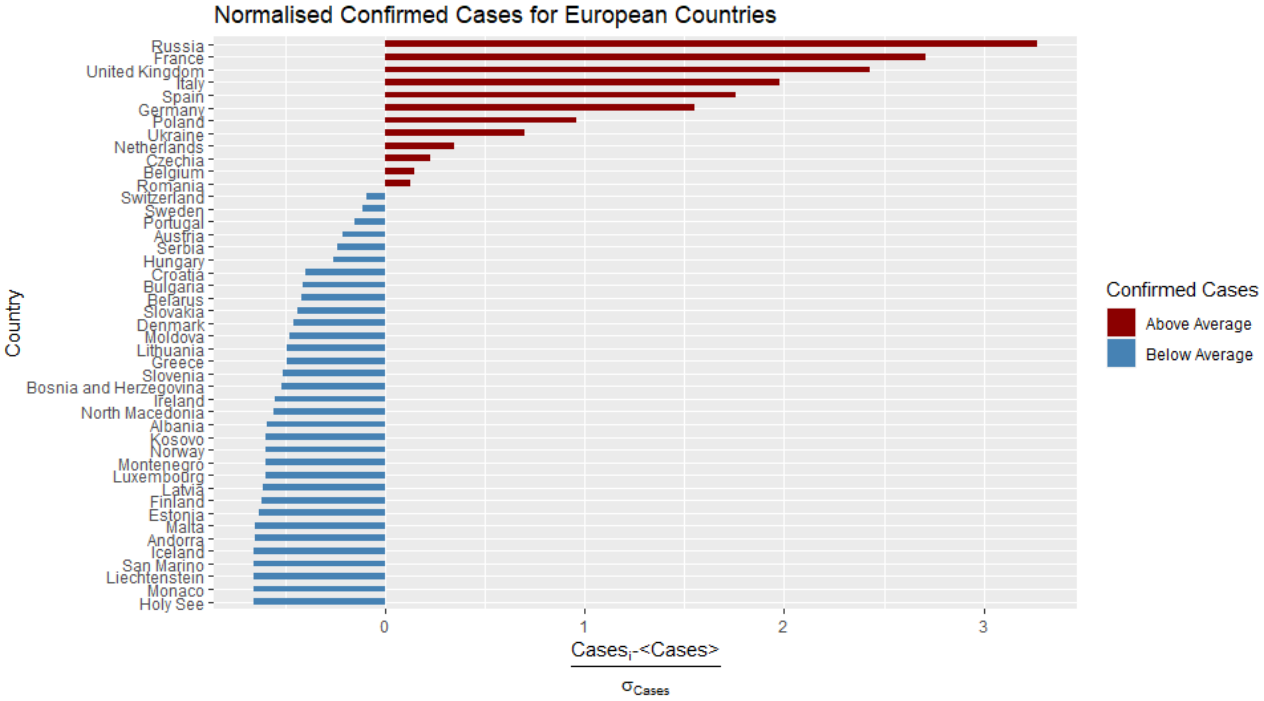

Fig. 7: Normalized confirmed cases for Europe

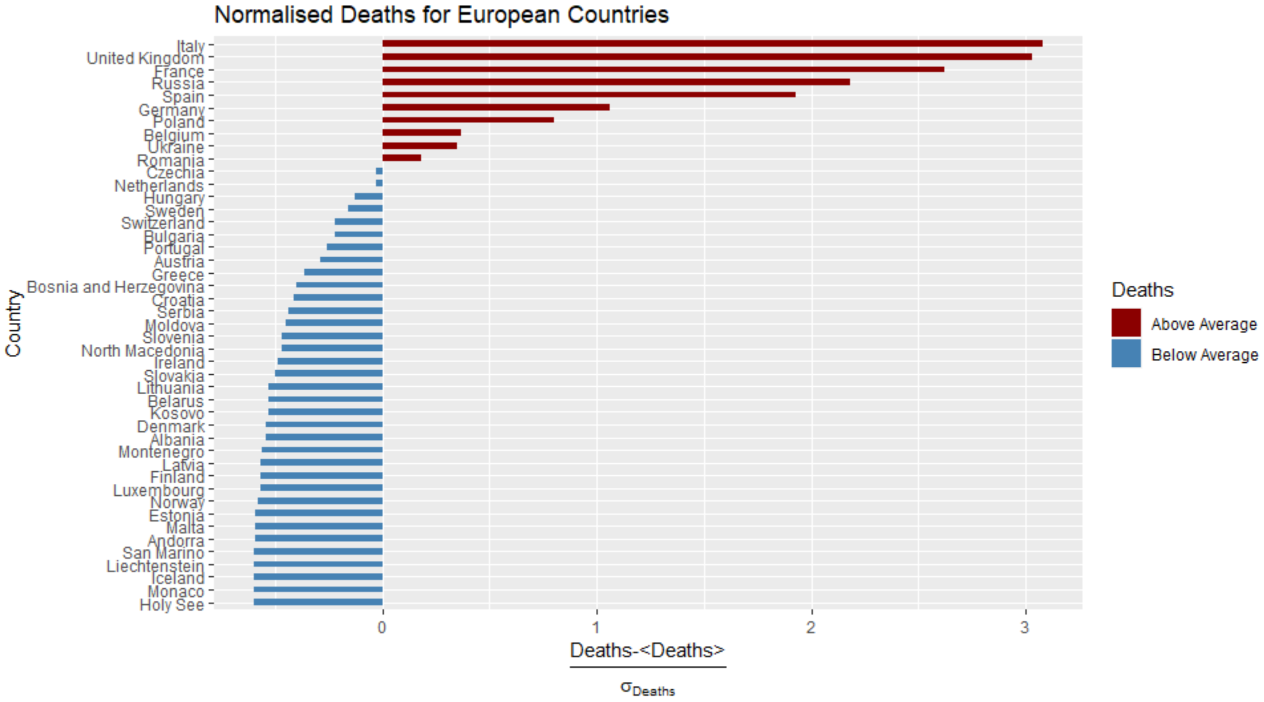

Fig. 8: Normalized confirmed deaths for Europe

[3] Source:This link from www.statista.com

```
library(latex2exp)#LaTeX expressions for labels
Europe_continent <- dt[continent == 'Europe']
Europe_cases <- Europe_continent %>% slice_max(order_by = Date, n = 1)
Europe_cases$Confirmed_normalized <-
    round((Europe_cases$Confirmed-mean(Europe_cases$Confirmed))/
    sd(Europe_cases$Confirmed),2)
Europe_cases$flag_cases <-
    ifelse(Europe_cases$Confirmed_normalized < 0, "Below","Above")
Europe_cases <- Europe_cases[order(Europe_cases$Confirmed),]
Europe_cases$Country <- factor(Europe_cases$Country,
levels = Europe_cases$Country)
as_above_so_below <- ggplot(Europe_cases, aes(x=Country, #FirstPlot
y = Confirmed_normalized,label = Confirmed_normalized))+
  geom_bar(stat = 'identity', width = .5, aes(fill = flag_cases))+
  scale_fill_manual(name = 'Confirmed Cases',
  labels = c("Above Average", "Below Average"),
  values = c("Above"="darkred", "Below"="steelblue")) +
  labs(title="Normalised Confirmed Cases for European Countries") +
  ylab(TeX("$\\frac{Cases_i-< Cases >}{\\sigma_{Cases}}$"))+
  coord_flip()
Europe_deaths <- Europe_continent %>% slice_max(order_by = Date, n = 1)
Europe_deaths$Deaths_normalized <-
    round((Europe_deaths$Deaths-mean(Europe_deaths$Deaths))/
    sd(Europe_deaths$Deaths),2)
Europe_deaths$flag_deaths <-
    ifelse(Europe_deaths$Deaths_normalized < 0, "Below","Above")
Europe_deaths <- Europe_deaths[order(Europe_deaths$Deaths),]
Europe_deaths$Country <- factor(Europe_deaths$Country,
    levels = Europe_deaths$Country)
as_above_so_below_2 <- ggplot(Europe_deaths, aes(x=Country,
    y = Deaths_normalized, label = Deaths_normalized))+
  geom_bar(stat = 'identity', width = .5, aes(fill = flag_deaths))+
  scale_fill_manual(name = 'Deaths',
  labels = c("Above Average", "Below Average"),
  values = c("Above"="darkred", "Below"="steelblue")) +
  labs(title="Normalised Deaths for European Countries") +
  ylab(TeX("$\\frac{Deaths-< Deaths >}{\\sigma_{Deaths}}$"))+
  coord_flip()
as_above_so_below_2
```

3. Country-wide Perspective

As dicussed on the previous section, to overcome the population discrepancies, we will hand-pick a few European countries with a population of 10-11 millions, similar to our country's. The countries we choose are: Greece, Sweden, Azerbaijan, Portugal and Czech Republic (Czechia).

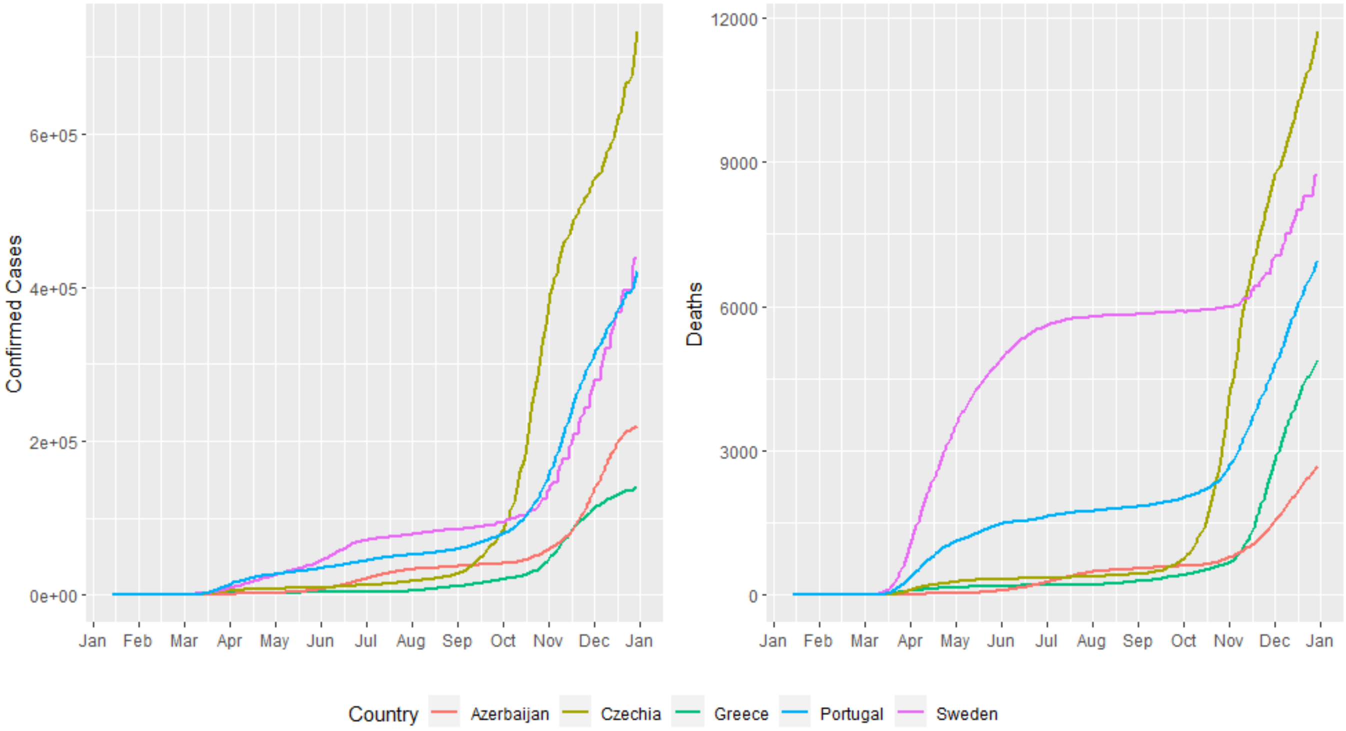

Fig. 9: Confirmed Cases and Deaths, for selected countries with population of 10M to 11M.

Below is the code for the plots (9) above.

```
plotcases <- ggplot(dt, aes(colour = Country),legend=FALSE)+
geom_line(data=dt[Country=='Greece'],aes(x=Date,y=Confirmed),size=.8)+
geom_line(data=dt[Country=='Azerbaijan'],aes(x=Date,y=Confirmed),size=.8)+
geom_line(data=dt[Country=='Czechia'],aes(x=Date,y=Confirmed),size=.8)+
geom_line(data=dt[Country=='Sweden'],aes(x=Date,y=Confirmed),size=.8)+
geom_line(data=dt[Country=='Portugal'],aes(x=Date,y=Confirmed),size=.8)+
xlab("")+ylab("Confirmed Cases")+
scale_x_date(date_breaks = '30 day', date_labels = '%b')

plotdeaths <- ggplot(dt, aes(colour = Country))+
geom_line(data=dt[Country=='Greece'],aes(x=Date,y=Deaths),size=.8)+
geom_line(data=dt[Country=='Azerbaijan'],aes(x=Date,y=Deaths),size=.8)+
geom_line(data=dt[Country=='Czechia'],aes(x=Date,y=Deaths),size=.8)+
geom_line(data=dt[Country=='Sweden'],aes(x=Date,y=Deaths),size=.8)+
geom_line(data=dt[Country=='Portugal'],aes(x=Date,y=Deaths),size=.8)+
xlab("")+ylab("Deaths")+
scale_x_date(date_breaks = '30 day', date_labels = '%b')
library(ggpubr)
ggarrange(plotcases,plotdeaths,nrow=1,
common.legend = TRUE, legend="bottom")
```

On plot 9, we present two line plots on the confirmed cases and deaths of the said countries. Since the countries have similar population, the differences on the number of confirmed cases and deaths are due to differences between the countries. For example, during the first wave of the pandemic, we can clearly see how Sweden has the most deaths. That's due to the fact that Sweden's government from the onset of the Covid-19 pandemic embarked on a herd immunity approach, allowing community transmission to occur relatively unchecked. We can also notice that Czech Republic's numbers went up after October, and the government's response was a two week lockdown starting at the 22th of October, which didn't seem to flatten the curve.

Another useful parameter we can use to visualize the pandemic's growth, is the growth rate percentage. For a number of daily cases on day i, $cases_i$, the growth rate is calculated as: $\frac{cases_i - cases_{i-1}}{cases_{i-1}}$. We can do the same to calculate the death growth rate.

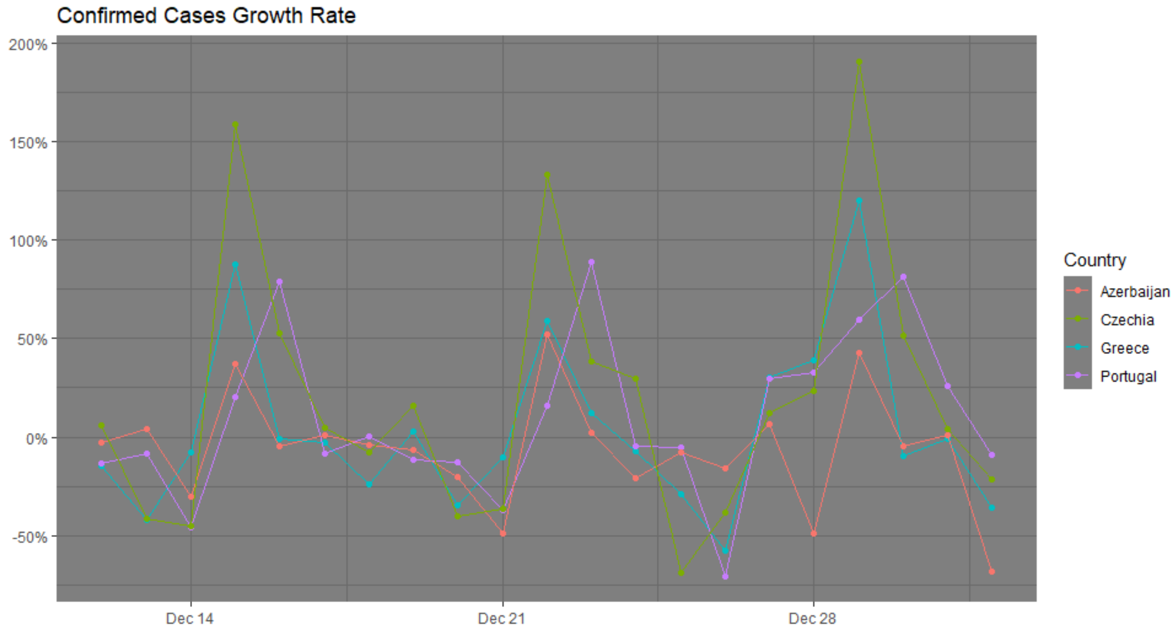

Fig. 10: Confirmed cases growth rate for selected countries. Date range: 12 December 2020 - 03 January 2021

This shows us a daily percentage of the growth or decrease of the confirmed cases. We plotted only the 3 weeks prior to the time of writing, because such a plot becomes more and more confusing the bigger the time window is. We can see that all four countries follow the same trend, which is probably a weekly trend, because less Covid-19 tests happen during the weekends. The peaks on plot 10 happen on Tuesdays, which means most people wait during the weekend and go to get tested on Monday.

Plot 10 was generated by the following code.

```
dt$Growth_factor_cases <-
  (dt$Confirmed.ind-lag(dt$Confirmed.ind, n = 1))/
  lag(dt$Confirmed.ind, n = 1)
dt$Growth_factor_cases[is.infinite(dt$Growth_factor_cases)] <- 0
#we set infinite growth factor as 0. Because of division by 0,
#when daily cases are 0.

growthrate <- ggplot(dt,aes(colour = Country),na.rm=TRUE)+
  geom_point(data=dt[Country == 'Greece' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),na.rm=TRUE)+
  geom_line(data=dt[Country == 'Greece' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),linetype='solid') +
  geom_point(data=dt[Country == 'Portugal' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),na.rm=TRUE)+
  geom_line(data=dt[Country == 'Portugal' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),linetype='solid')+
  geom_point(data=dt[Country == 'Azerbaijan' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),na.rm=TRUE)+
  geom_line(data=dt[Country == 'Azerbaijan' & Date >= '2020-12-12'],
  aes(x = Date, y = Growth_factor_cases),linetype='solid')+
  geom_point(data=dt[Country == 'Czechia' & Date >= '2020-12-12'],
             aes(x = Date, y = Growth_factor_cases),na.rm=TRUE)+
  geom_line(data=dt[Country == 'Czechia' & Date >= '2020-12-12'],
            aes(x = Date, y = Growth_factor_cases),linetype='solid')+
  xlab("")+ylab("")+labs(title='Confirmed Cases Growth Rate')+
  scale_y_continuous(labels = scales::percent,
  breaks = c(seq(-0.5,2,0.5)))+theme_dark()
```

To summarize, it's important to note, that since there is not protocol for how every country's government counts its cases and deaths, there may be inconsistencies. For example, if someone dies from a heart attack while having tested positive to Covid-19, some countries may count that event as a Covid-19 related death, while some others may not.

As vaccination has already started, until approximately the 70% of the population is vaccinated, we can't say the pandemic has come to a stop.

Until then, stay positive, test negative.

If you want to examine the code in detail, it is uploaded on this repository on GitHub.