Project Assignment

Analyzing Covid-19 data with R

Course: Visualization

Mentor: Assistant Professor, Dijana Capeska Bogatinoska, Ph.D.

Students: Hristijan Veleski, MIR, 1039

Dimitar Mitrevski, CSE, 1057

Ohrid, 2022

Table of Contents:

Overview and Purpose of the Project

The COVID-19 pandemic, also known as the coronavirus pandemic, is an ongoing global pandemic (as of January 2022) of coronavirus disease 2019 (COVID-19) caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The novel virus was first identified from an outbreak in the Chinese city of Wuhan in December 2019, and attempts to contain it there failed, allowing it to spread across the globe. The World Health Organization (WHO) declared a Public Health Emergency of International Concern on 30 January 2020 and a pandemic on 11 March 2020.

The purpose of our project is to show some covid analytics and statistics data concerning this issue. We will use some tools for comparison of data among the countries with respect to some variable, but also, we provide detailed information for a specific country regarding its rate for some variables. We want to show how covid affects countries through time series analysis, by using different visualization techniques.

Loading and understanding the dataset

First of all, we need to load and understand the dataset in order to be able to work with the provided data.

```
> covid_data<-read.csv("covid_data.csv")
> head(covid_data)
  iso_code continent    location      date total_cases new_cases
1      AFG      Asia Afghanistan 2/24/2020           5         5
2      AFG      Asia Afghanistan 2/25/2020           5         0
3      AFG      Asia Afghanistan 2/26/2020           5         0
4      AFG      Asia Afghanistan 2/27/2020           5         0
5      AFG      Asia Afghanistan 2/28/2020           5         0
6      AFG      Asia Afghanistan 2/29/2020           5         0
```

This is a huge dataset, so it requires a bit of time before it downloads. Normally, every dataset contains a lot of NA (not available) entries. So does this one.

In order to understand better this dataset, we are providing some basic information for some of the columns that we consider more important than the others.

Variable	Description
total_cases	Total confirmed cases of COVID-19
new_cases	New confirmed cases of COVID-19
new_cases_smoothed	New confirmed cases of COVID-19 (7-day smoothed)
hosp_patients	Number of COVID-19 patients in hospital on a given day

reproduction_rate	Real-time estimate of the effective reproduction rate (R) of COVID-19 (how many people a person can spread the virus to)
total_tests	Total tests for COVID-19
people_fully_vaccinated	Total number of people who received all doses prescribed by the vaccination protocol
iso_code	ISO 3166-1 alpha-3 – three-letter country codes
continent	Continent of the geographical location
location	Geographical location
date	Date of observation
aged_70_older	Share of the population that is 70 years and older in 2015

There are, of course, more variables, but most of their names are descriptive, so it is easy to understand the meaning.

Data Analysis and Visualization using R

A powerful tool in data analysis and visualization when it comes to Covid-19, it is the library 'covid19.analytics'. It has some smooth and well-structured graphs of the data that are easy to understand and to prove a certain conclusion.

We start off by viewing the aggregated data

```r
#AggregatedData
ag = covid19.data(case = 'aggregated')
View(ag)
```

Output:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Combin
1	NA			Afghanistan	2022-01-15 04:21:07	33.93911	67.70995	158639	7376	NA	NA	Afgha
2	NA			Albania	2022-01-15 04:21:07	41.15330	20.16830	228777	3262	NA	NA	Alban
3	NA			Algeria	2022-01-15 04:21:07	28.03390	1.65960	224979	6393	NA	NA	Algeri
4	NA			Andorra	2022-01-15 04:21:07	42.50630	1.52180	29888	142	NA	NA	Andor
5	NA			Angola	2022-01-15 04:21:07	-11.20270	17.87390	93302	1852	NA	NA	Angol
6	NA			Antigua and Barbuda	2022-01-15 04:21:07	17.06080	-61.79640	5246	120	NA	NA	Antigu
7	NA			Argentina	2022-01-15 04:21:07	-38.41610	-63.61670	6932972	117901	NA	NA	Argen
8	NA			Armenia	2022-01-15 04:21:07	40.06910	45.03820	347084	8015	NA	NA	Armen
9	NA		Australian Capital Territory	Australia	2022-01-15 04:21:07	-35.47350	149.01240	21174	16	NA	NA	Austra
10	NA		New South Wales	Australia	2022-01-15 04:21:07	-33.86880	151.20930	738891	846	NA	NA	New S
11	NA		Northern Territory	Australia	2022-01-15 04:21:07	-12.46340	130.84560	5290	1	NA	NA	North
12	NA		Queensland	Australia	2022-01-15 04:21:07	-27.46980	153.02510	177454	20	NA	NA	Queen
13	NA		South Australia	Australia	2022-01-15 04:21:07	-34.92850	138.60070	62511	36	NA	NA	South
14	NA		Tasmania	Australia	2022-01-15 04:21:07	-42.88210	147.32720	17098	13	NA	NA	Tasma
15	NA		Victoria	Australia	2022-01-15 04:21:07	-37.81360	144.96310	589546	1680	NA	NA	Victor
16	NA		Western Australia	Australia	2022-01-15 04:21:07	-31.95050	115.86050	1285	9	NA	NA	Weste
17	NA			Austria	2022-01-15 04:21:07	47.51620	14.55010	1411421	13905	NA	NA	Austri
18	NA			Azerbaijan	2022-01-15 04:21:07	40.14310	47.57690	624208	8497	NA	NA	Azerb

If we want to see a detailed time series of confirmed cases, we can do that by typing

```r
#Time Series of Confirmed Cases
tsc = covid19.data(case = 'ts-confirmed')
View(tsc)
```

It even shows for example regions of the greater countries (like Australia) for better reading and understanding the detailed confirmed cases day-by-day.

Output:

	Province.State	Country.Region	Lat	Long	2020-01-22	2020-01-23	2020-01-24	2020-01-25	2020-01-26	2020-01-27	2020-01-28	202 01-
1		Afghanistan	33.939110	67.709953	0	0	0	0	0	0	0	
2		Albania	41.153300	20.168300	0	0	0	0	0	0	0	
3		Algeria	28.033900	1.659600	0	0	0	0	0	0	0	
4		Andorra	42.506300	1.521800	0	0	0	0	0	0	0	
5		Angola	-11.202700	17.873900	0	0	0	0	0	0	0	
6		Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	
7		Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	
8		Armenia	40.069100	45.038200	0	0	0	0	0	0	0	
9	Australian Capital Territory	Australia	-35.473500	149.012400	0	0	0	0	0	0	0	
10	New South Wales	Australia	-33.868800	151.209300	0	0	0	0	3	4	4	
11	Northern Territory	Australia	-12.463400	130.845600	0	0	0	0	0	0	0	
12	Queensland	Australia	-27.469800	153.025100	0	0	0	0	0	0	0	
13	South Australia	Australia	-34.928500	138.600700	0	0	0	0	0	0	0	
14	Tasmania	Australia	-42.882100	147.327200	0	0	0	0	0	0	0	
15	Victoria	Australia	-37.813600	144.963100	0	0	0	0	1	1	1	
16	Western Australia	Australia	-31.950500	115.860500	0	0	0	0	0	0	0	
17		Austria	47.516200	14.550100	0	0	0	0	0	0	0	
18		Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	
19		Bahamas	25.025885	-78.035889	0	0	0	0	0	0	0	
20		Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	
21		Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	
22		Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	
23		Belarus	53.709800	27.953400	0	0	0	0	0	0	0	
24		Belgium	50.833300	4.469936	0	0	0	0	0	0	0	
25		Belize	17.189900	-88.497600	0	0	0	0	0	0	0	
26		Benin	9.307700	2.315800	0	0	0	0	0	0	0	
27		Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	
28		Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	
29		Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	
30		Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	
31		Brazil	-14.235000	-51.925300	0	0	0	0	0	0	0	
32		Brunei	4.535300	114.727700	0	0	0	0	0	0	0	

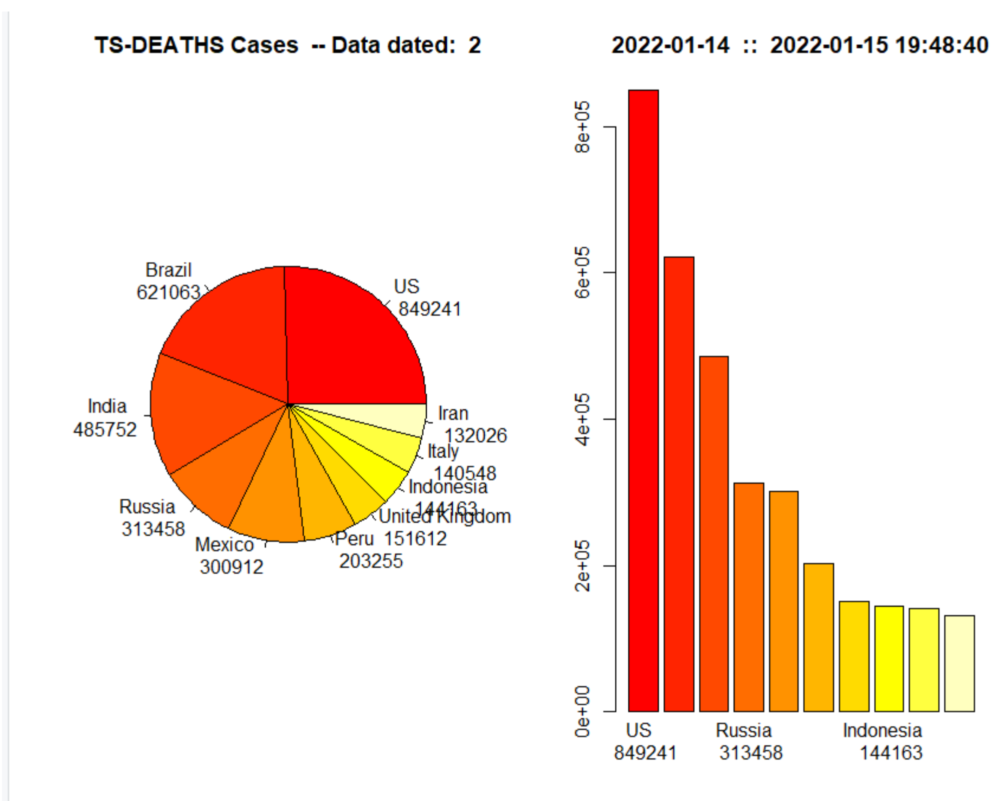
Using the following code we can show the top n countries hit by the pandemic, concerning the death cases. In our case, we set Nentiries=10, but it can be changed to other number, depending how many top-hit-by-death countries we want.

```
#Summary
report.summary(Nentries = 10,graphical.output = T)
```

The first output shows tabular data for the top 10 countries hit by the pandemic, so we can understand more of it.

```
Console   Terminal ×   Jobs ×
R 4.1.2 · ~/
checking for ... Country Province Lat Long
No critical issues have been found.
 >>> checking data consistency...
Warning in consistency.check(data, n0, nf, datasetName, details = details,  :
  Inconsistency of type.II in ts-confirmed data detected -- 92 records (out of 280) show inconsistencies in the data...
################################################################################
  ##### TS-CONFIRMED Cases  -- Data dated:  2022-01-14  ::  2022-01-15 19:34:04
################################################################################
  Number of Countries/Regions reported:  196
  Number of Cities/Provinces reported:  88
  Unique number of distinct geographical locations combined: 280
--------------------------------------------------------------------------------
  Worldwide ts-confirmed  Totals: 323567887
--------------------------------------------------------------------------------
   Country.Region Province.State   Totals GlobalPerc LastDayChange    t-2    t-3    t-7   t-14   t-30
1              US                64917963      20.06        855903 776455 899678 380312 117510 141742
2           India                36850962      11.39        533035      0 442137 159632  27553   7447
3          Brazil                22933289       7.09        111112  97945  87873  49303   3782   3661
4  United Kingdom                15066395       4.66         98578 105679 129544 140566 162572  87565
5          France                13279505       4.10        322648 301647 332476 303669 219126  60556
6          Russia                10565484       3.27         23614  20972  17797  16424  19606  28158
7          Turkey                10273170       3.17             0  75564  77722  66237  36731  18100
8           Italy                 8356514       2.58        200869 184577 196205 197535 141256  26104
9           Spain                 8093036       2.50        162508 159161 179125      0      0  28900
10        Germany                 7943959       2.46         77175  89920  86232  32128   2968  51400
```

While this output shows a pie chart of the “best” hit countries of the pandemic, again, considering the death cases. Next to it is a barplot, ordered from the highest of the top 10 (US), up until the last in the top 10, (Iran).



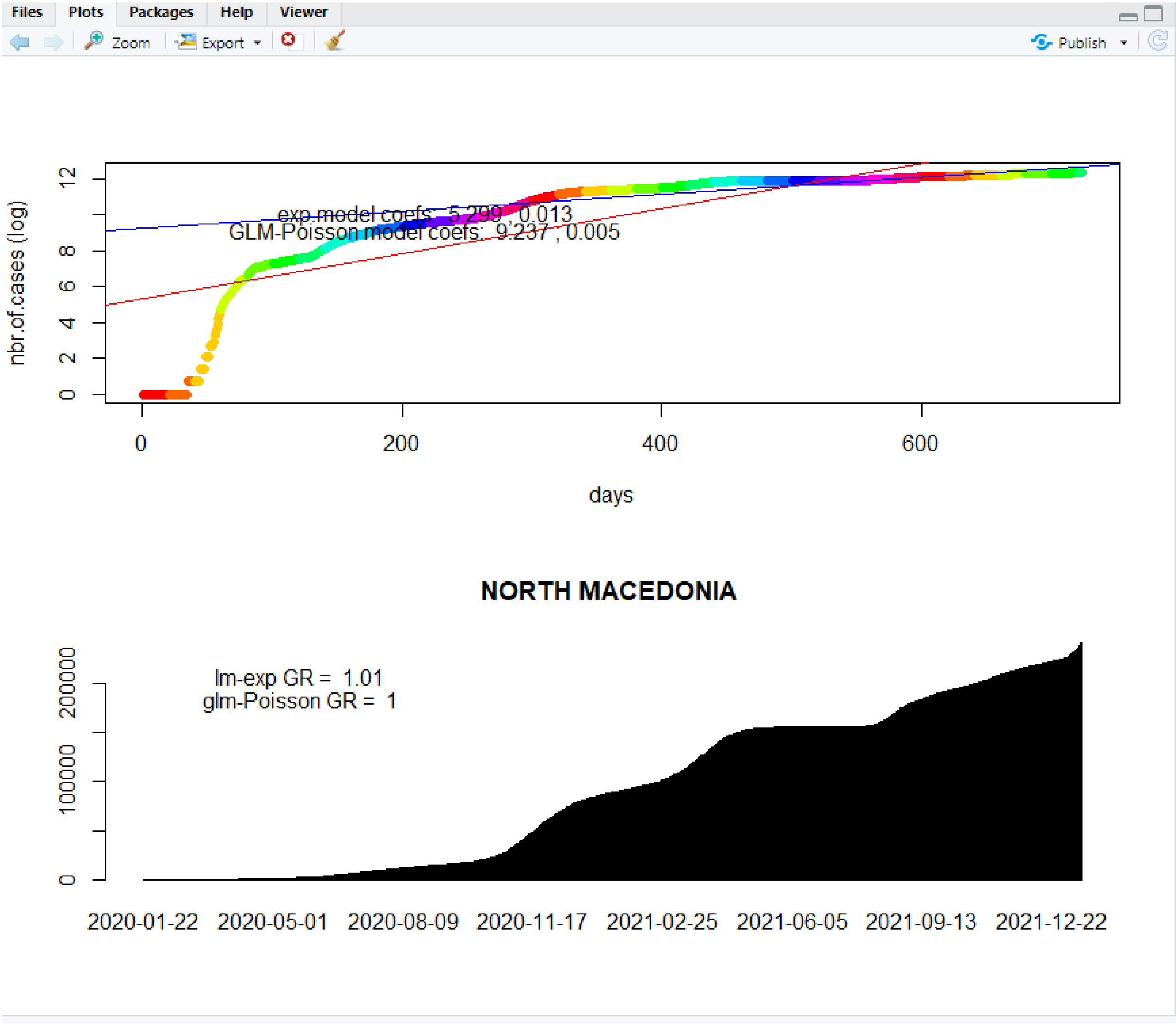
```
#Totals per location
tots.per.location(tsc,geo.loc = 'North Macedonia')
```

Similarly, we can see detailed statistics for a certain country. In our case, we chose our country, Macedonia, but geo.loc can be set to any country.

```
Console   Terminal ×   Jobs ×
R 4.1.2 · ~/
> tots.per.location(tsc,geo.loc = 'North Macedonia')
NORTH MACEDONIA  --  241651
===============================   running models...===============================
  Linear Regression (lm):

Call:
lm(formula = y.var ~ x.var)

Residuals:
   Min     1Q Median     3Q    Max 
-34932  -8273   1251   8530  42288 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -42659.081   1248.628  -34.16   <2e-16 ***
x.var          371.368      2.984  124.45   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16780 on 722 degrees of freedom
Multiple R-squared:  0.9555,	Adjusted R-squared:  0.9554 
F-statistic: 1.549e+04 on 1 and 722 DF,  p-value: < 2.2e-16

--------------------------------------------------------------------------------
  Linear Regression (lm):
```

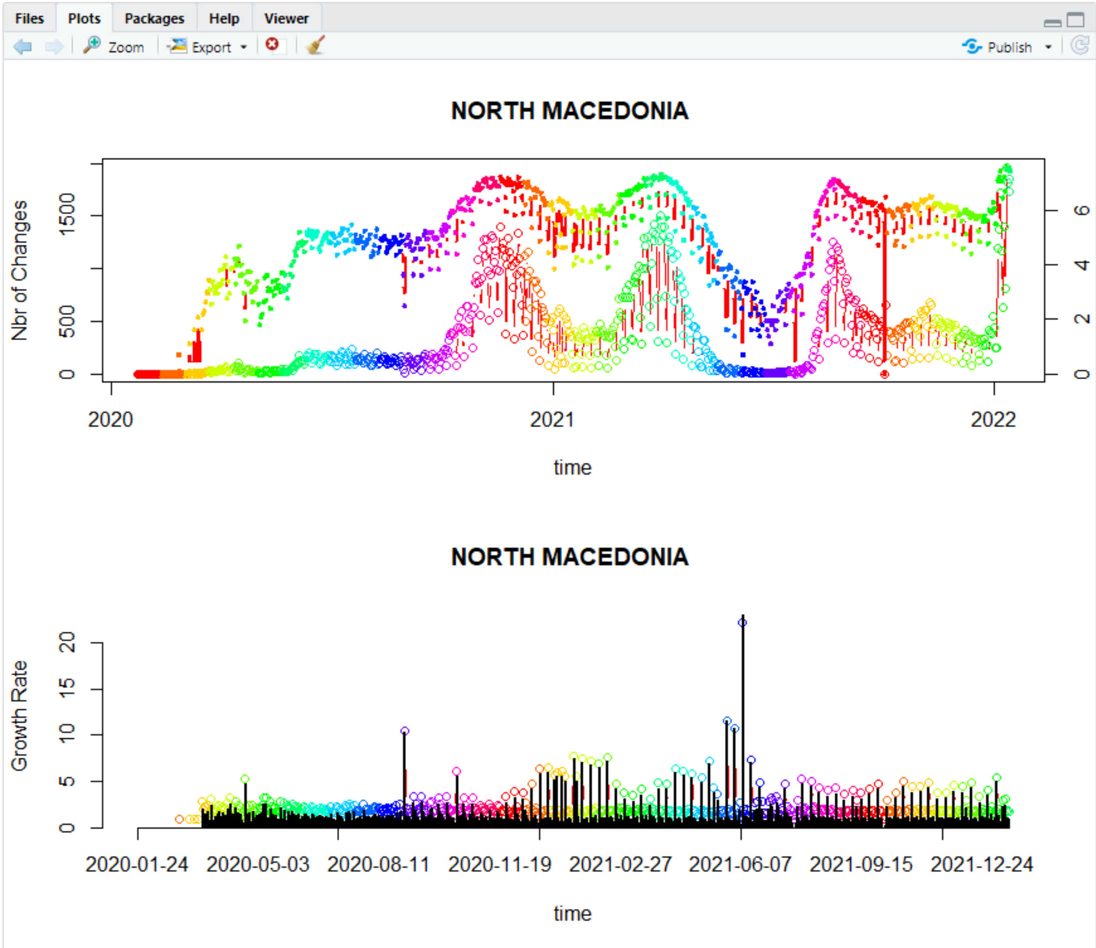
Same like previously, we are showing firstly some descriptive data. Linear regression is shown by default with this package.

In the second graph, we can observe how the total cases changed over time for the country Macedonia



The growth rate is the amount in which the value of an asset, increases over a specific period. The growth rate provides us with important information about the value of an asset as it helps us understand how that asset grows, changes and performs over time.

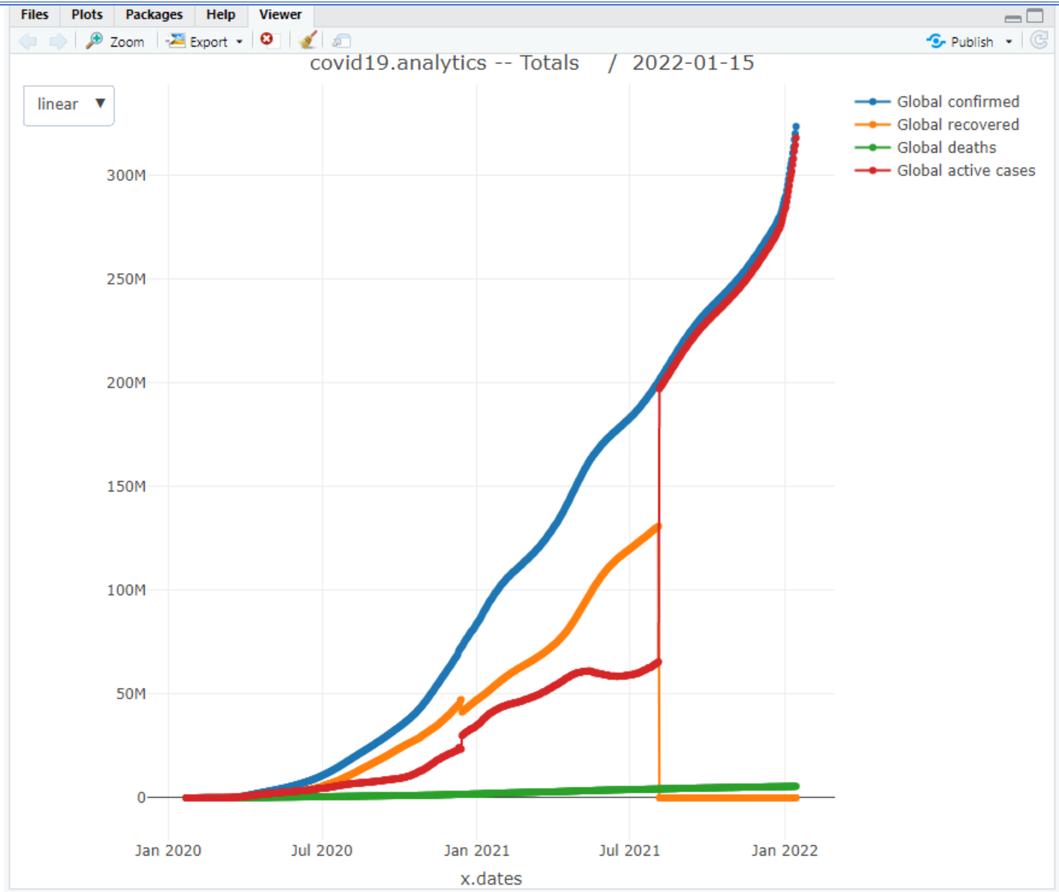
```
#Growth rate
growth.rate(tsc,geo.loc = 'North Macedonia')
```



Next, we want to do a time series analysis of the global data, thus we set case = 'ts-ALL',

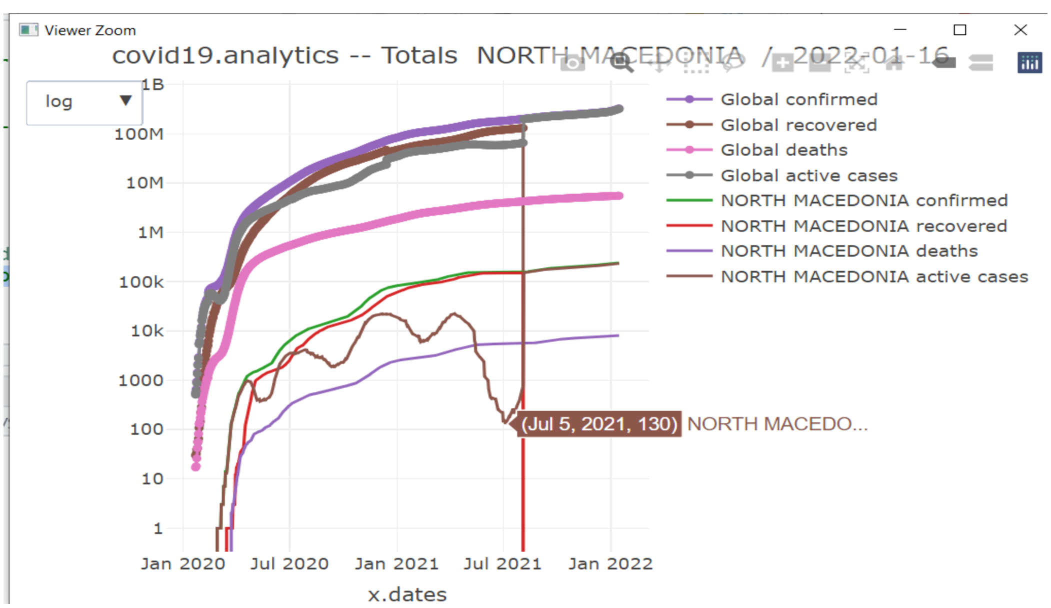
```
#time series global data
tsa = covid19.data(case = 'ts-ALL')
View(tsa)

#Totals plot
totals.plt(tsa)
```

and we are plotting it, so this is the result



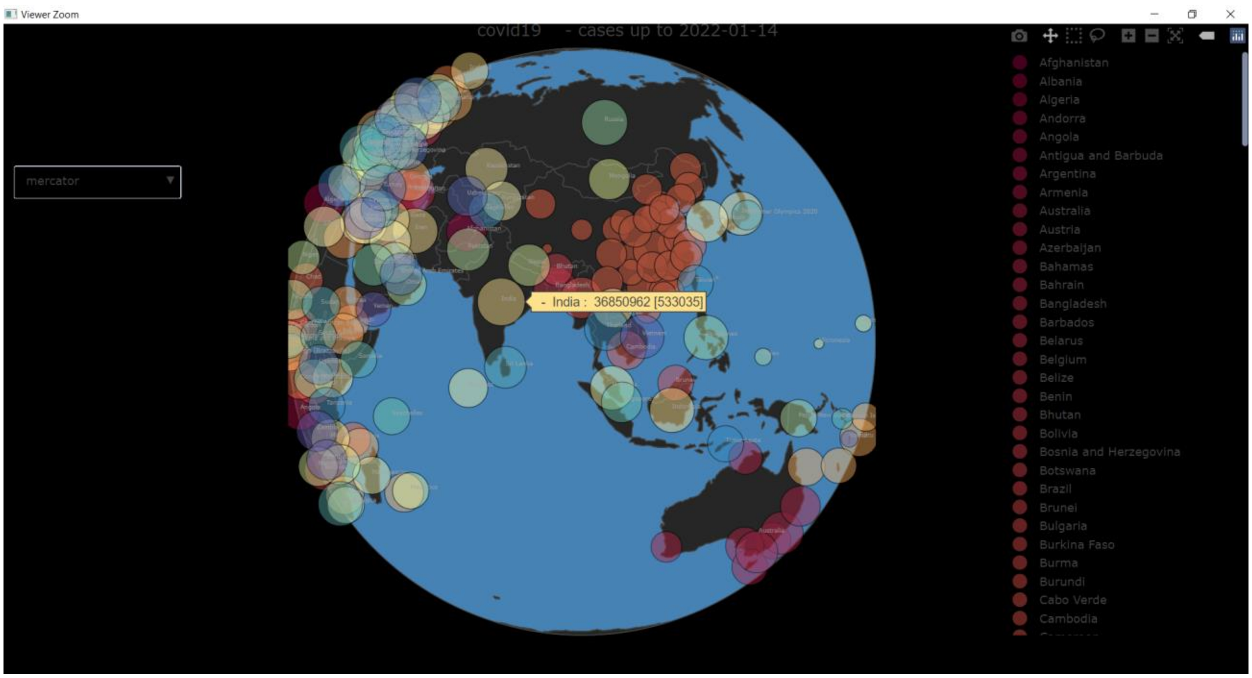
We would like to do the same for the country Macedonia, so here it is the result.

```
#Totals plot compared to Macedonia
totals.plt(tsa,c('North Macedonia'))
```



Some really cool thing we can do with these plots is the globe map, or so-called live map. It is really easy to generate it within this library, and it looks interesting.

```
#global map
live.map(tsc)
```

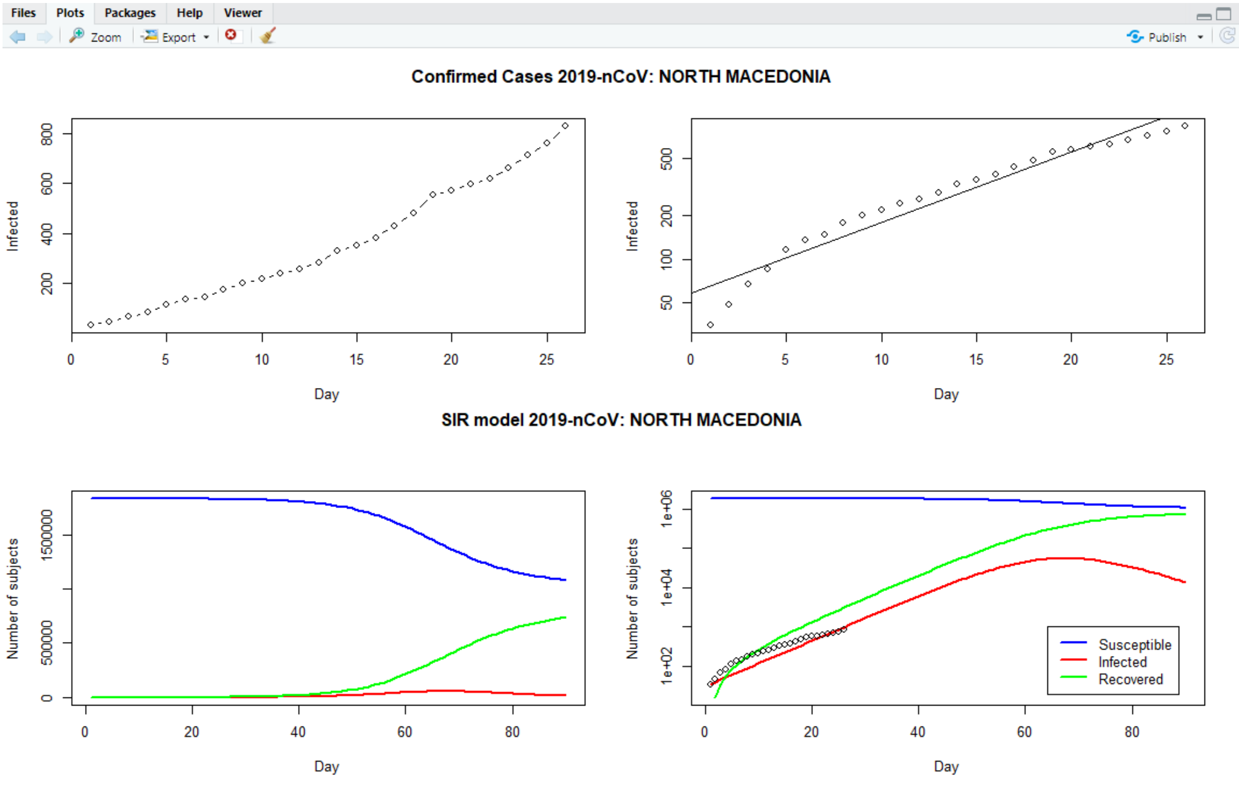


If you hover on a country's circle, it shows its name, total number infected, and deaths in square brackets.

Compartmental models are a very general modelling technique. They are often applied to the mathematical modelling of infectious diseases. The population is assigned to compartments with labels – for example, S, I, or R, (Susceptible, Infectious, or Recovered).

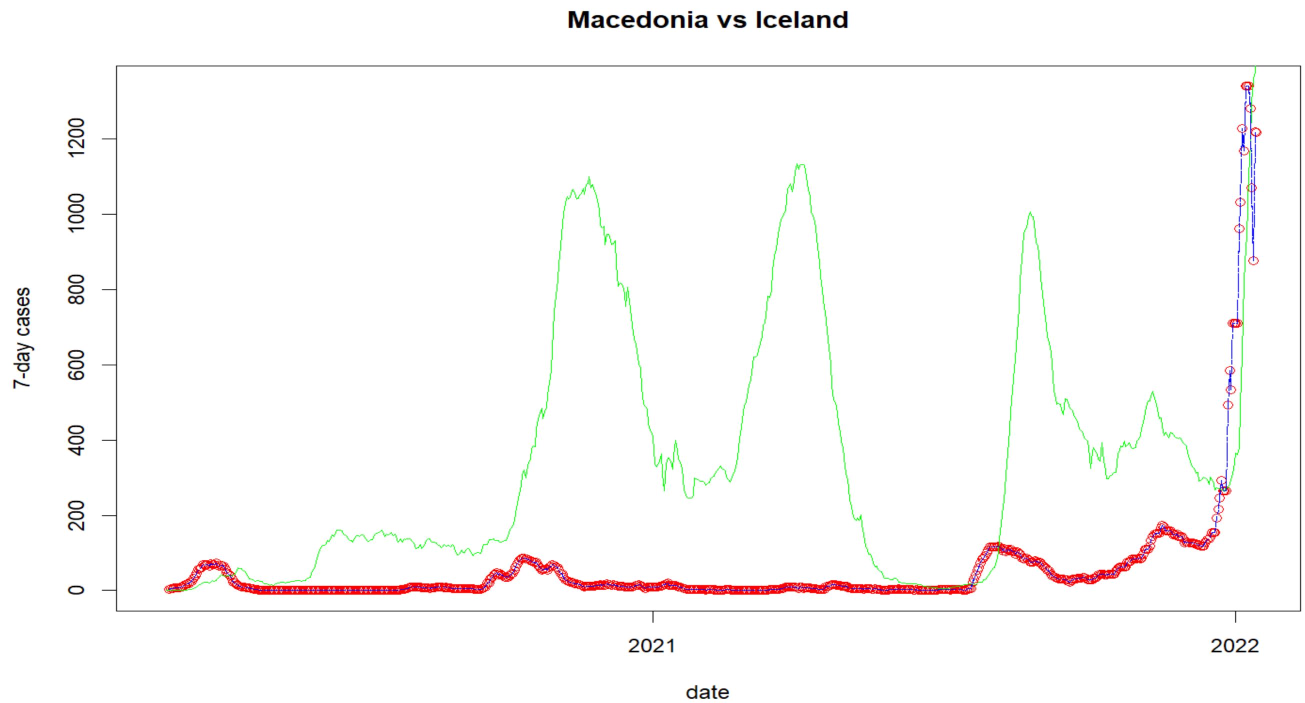
```
#SIR Model
generate.SIR.model(tsc,'North Macedonia',tot.population = 1832696 )
```

The output of the SIR model:



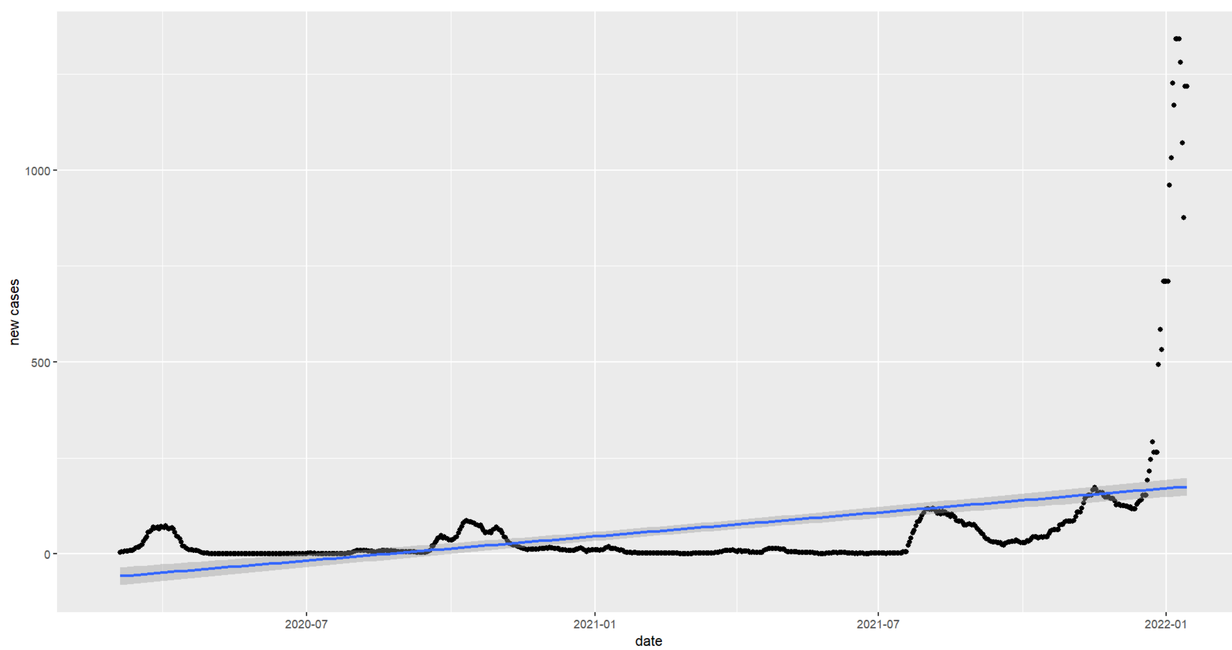
In the following graph, it is shown the difference between Macedonia and Iceland with respect to daily cases. This is made using the base plotting system.

```
plot(as.Date(ice$date, "%m/%d/%Y"), ice$new_cases_smoothed, col="red",
     xlab = "date", ylab = "7-day cases", main = "Macedonia vs Iceland")
lines(as.Date(ice$date, "%m/%d/%Y"), ice$new_cases_smoothed, col="blue", lty=1)
lines(as.Date(mkd$date, "%m/%d/%Y"), mkd$new_cases_smoothed, col="green")
```



Linear model of the data for the country Iceland:

```
gg<-ggplot(ice, aes(as.Date(ice$date, "%m/%d/%Y"),ice$new_cases_smoothed))
gg+geom_point()+geom_smooth(method = "lm")
```



Conclusion

After everything being presented, we can conclude that although Covid-19 pandemic has affected all of us and our health, if we compare it to other diseases, like cancer, other rare disease that there is no cure, or even hunger, we can observe that these things are far more dangerous and deadlier. We should not forget that there is also an ongoing “pandemic” with hunger, that lasts since a long time ago, and it does not seem to be solved in the upcoming years. Based on current GHI projections, the world as a whole—and 47 countries in particular—will fail to achieve a low level of hunger by 2030. [1]

Namely,

During the past two decades, population growth, improvement in incomes and diversification of diets have steadily increased the demand for food. Prior to 2000, food prices were in decline, largely through record harvests. At the same time, however, public and private investment in agriculture, especially in the production of staple food, decreased, which led to stagnant or declining crop yields in most developing countries.

A triple challenge
The current global food crisis is a huge challenge. It will require sustained political commitment at the highest levels for many years if we are to deal with it successfully and prevent further mass pauperization and the rolling back of development gains painfully won.

What the problem is

Food prices began rising in 2004, with a particularly steep increase in 2006. The Food and Agriculture Organization of the United Nations forecasts that the world will spend $1,035 billion on food imports in 2008, about $215 billion more than in 2007. This will severely strain the budgets of Low-Income Food-Deficit Countries whose food bills will soar by more than 40 per cent in 2008. This may also cause inflation, disrupt the balance of payments and increase debt for many low-income countries.

What the crisis can teach us

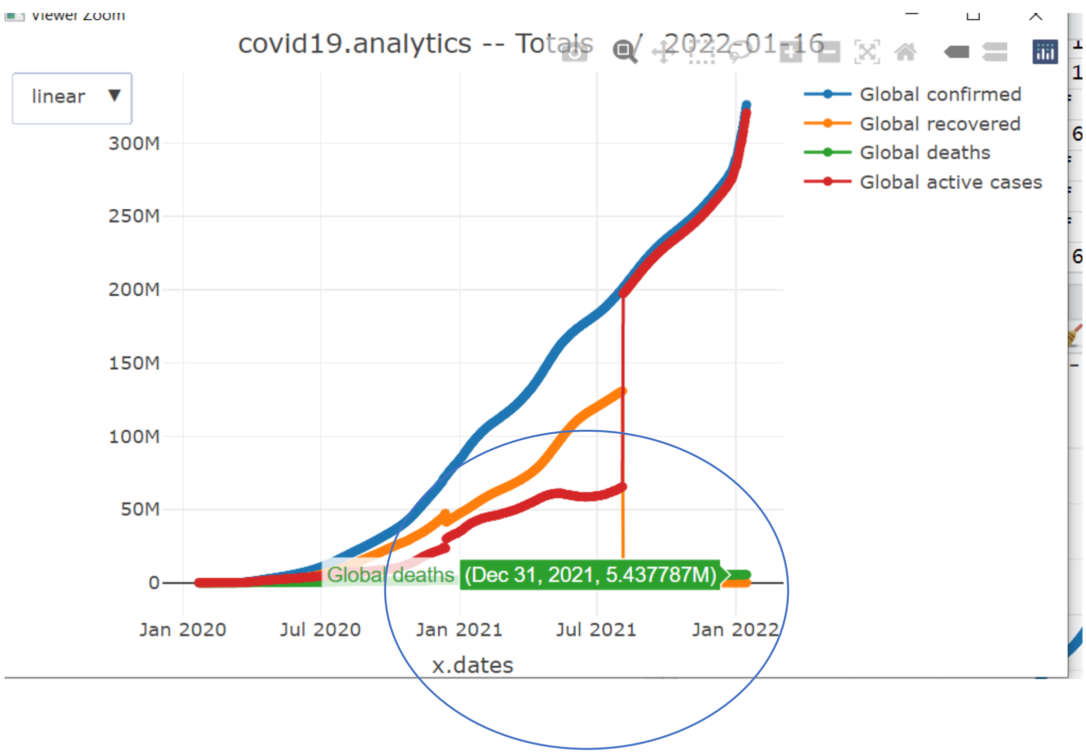
Escalating food prices can benefit smallholder farmers if appropriate assistance is available. Interventions should ensure access to inputs, i.e. seed and fertilizer, rehabilitation of infrastructure and methods to decrease post-harvest losses. This will boost crop yields, improve rural household welfare and local food supply. [2]

The Topic of this article is: *Losing 25,000 to Hunger Every Day*

Making simple calculations, 25k a day, for 365 days is 9.125.000 deaths from hunger, yearly

The Covid Pandemic as we can observe from the graph, has affected 5.43M since the outbreak in December 2019 (two years)

18.250.000 deaths from hunger vs 5.437.787 deaths from Covid-19 for 2 years.



"Well, into the 21st century, hunger is still the world's biggest health problem. And it's about to get worse." [3]

Future work

What can we do?

A child dies from hunger every 10 seconds.

There are many ways to help children in need:

-Don't waste food, or buy unnecessary groceries, instead, if you have some extra food, donate it to someone that really needs it.

- You can donate money to organizations that deal with children in hunger, that way, you can help to children from abroad.

- Spread the word. Convince other people to care for the children, so more people would donate food and/or money.

- Let's not forget to help stop the climate change, because climate affects agriculture, and we all depend on it!

References

[1] https://www.globalhungerindex.org/pdf/en/2021.pdf

[2] https://www.un.org/en/chronicle/article/losing-25000-hunger-every-day

[3] https://www.theworldcounts.com/challenges/people-and-poverty/hunger-and-obesity/how-many-people-die-from-hunger-each-year/story