

UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA

Dipartimento di Scienze Teoriche e Applicate - DiSTA

BACHELOR OF COMPUTER SCIENCE



Development of a tool to study
COVID-19 contagiousness via $R_0(t)$ in Italy
at national, regional and provincial level

Alice Schiavone

Thesis advisor: Davide Tosi

Academic Year: 2019/20

Abstract:

On December 30, 2019, Li Wenliang, ophthalmologist at Wuhan Central Hospital, shared in an online chat with a group of friends the news of suspected SARS-like coronavirus illness at his hospital. On January 3, 2020, Wuhan police asked Li to stop spreading fake news online. Li went back to work and became infected with the new virus. He later died of it on February 7, at 33 years old.

As time passed, the new virus spread. Originated in China, it quickly became a global health emergency. Hospitals were overwhelmed by the contagiousness of the virus, because sick people often require to be hospitalized in intensive care, assisted by a ventilator (machine that moves breathable air in and out of the lungs), as this virus affects mainly the respiratory system.

Governments of the world tried different strategies to contain the spread of the virus to their citizens, relying on different mathematical models, that show the progress and outcome of an epidemic. One of these models is the basic reproduction number (Time-Dependant Method) $R_0(t)$, that measures how contagious is a disease over a period of time.

The numbers about the pandemic in Italy are released by the Italian Department of Civil Protection everyday at 18:00. With the help of two data analysis-oriented programming languages and a package released by epidemiologists for disease outbreaks, the project discussed in this paper aims at developing a tool that will take advantage of the basic reproduction number to show the trend of the contagiousness level of this virus, particularly in Italy as a nation, and its regions and provinces.

Following the instructions, it is possible to download the data and export $R_0(t)$ for each of the 129 subdivisions of Italy, from February 24 to the current day, as a PDF file, including a graphical representation of the contagiousness trend per area.

The tool development continues through a project of visualization and representation of the data elaborated by this tool, expanding the original design to better depict the numbers.

The hope is to help in understanding the development of the virus, to take the necessary action in order to keep the highest number of people safe and go back to our normal lives as soon as possible.

Contents

Abstract:	2
Introduction	4
Covid-19: brief history	4
How to contain an epidemic	5
What is R0?	6
History.....	6
How is R0 calculated?	7
Aim and objectives	8
Design and development	9
Solution Design	9
Managing data	11
Activity diagram.....	14
Test Suite	15
Expected results	15
Runtime example.....	16
Conclusions.....	17
Results	17
Threads to Validity	19
Further development	19
Appendices	20
User's Manual	20
Source Code	22
Bibliography	24

Introduction

Covid-19: brief history

Coronaviruses have been classified as a different virus genus in the 1930s.

First, it was discovered in chickens and since then coronaviruses have naturally infected pigs, rats, and other animals, causing disorders that effect multiple organs.

It is known to affect men since the 1960s, when different strains of coronaviruses were isolated from people suffering from upper respiratory diseases. [1]

On December 31, 2019, the WHO China Country Office (global organization that directs international health among countries who belong to the United Nations' system) reported that Chinese authorities had identified a new type of coronavirus, [2] and isolated a first cluster in Wuhan, in a so-called 'wet market', a place where large collections of open-air stalls sell fresh food, often slaughtering live animals before customers eyes. These markets often sell wild animals, to be eaten or used as ingredients for Chinese traditional medicine. A virus can jump to workers or customers through close interaction with these animals, and they have caused multiple disease outbreaks in humans. [3] For example, Ebola was firstly developed by a two-year-old boy who was killed by the virus, and who lived fifty meters from a tree which once was home to thousands of bats. [4] It is also believed that HIV crossed from chimps to humans, when hunters killed and eaten the animals. [5]

Although the market was closed on January 1, 2020, and the Chinese authorities' intensive surveillance and active case finding, Wuhan (population: 19 million people) quickly became the first infected area of the world. On January 20, the Chinese National Health Commission confirmed that the virus could be transmitted from human to human and told people to practice social distancing and to avoid travelling. This happened during Chinese New Year festival, probably the most important festivity in the country, which brings home thousands of Chinese people emigrated all over the world. This brought the Chinese government to instigate the total city lockdown on January 25, stopping travel in and out of Wuhan. Containment is a common strategy in the control of an outbreak, among screening and mitigation.

Unfortunately, this didn't stop the spread of the virus, which in the meantime has been named 'SARS-Cov-2' (severe acute respiratory syndrome - coronavirus - 2) and the disease caused by the virus as 'COVID-19' (where "CO" stands for corona, "VI" for virus, "D" for disease and "19" indicates the year in which it occurred).

By the first week of April, more than half the world's population was under some form of lockdown. [6]

How to contain an epidemic

The term 'quarantine' spread in Europe during the XVII century in every European language, but its origin it's likely to be found in Italy, in mediaeval times. It translated to "forty-day period" (from 'quaranta', forty in Italian), but later in time it has been used to refer to a "period of application of health measures", with a variable number of days. [7]

During the Black Death, several European port cities required ships to isolate their crew for forty days before landing, to "separate sick persons from those who are healthy to prevent spread of disease". [8]

Today, forms of quarantine have been taken to contain and mitigate the virus effect on the healthcare system, trying to avoid an epidemic peak that would overwhelm hospitals' capacity to treat patients, and to wait for a vaccine development while flattening the epidemic curve. This, however, hasn't stopped the virus from spreading and results in thousands of deaths and considerable social and economic costs. [9]

What is still discussed among experts today is how virus spreads, but most agree that "current evidence suggests that SARS-CoV-2 is primarily transmitted between people via respiratory droplets and contact routes [...] and that transmission of COVID-19 is occurring from people who are pre-symptomatic or symptomatic to others in close contact." [10] Isolating infected people, however, may not be enough, as WHO continues saying that "transmission can also occur from people who are infected and remain asymptomatic, but the extent to which this occurs is not fully understood and requires further research as an urgent priority."

Although lockdowns helped to decrease the rapid spike of daily new cases, as the official number of COVID-19 related deaths closes to one million worldwide, in autumn the world is getting ready to manage a second wave of infections. What many countries are hoping for is the development of a vaccine that will protect the population from the disease, and there is currently a multibillion-dollar race for a vaccine for the virus. Even though the United States are promising to distribute doses as soon as November 2020 [11], it is becoming clear that governments worldwide will have to deal with this pandemic for a long period of time. Israel was the first country to re-establish a national lockdown on September 19, on Jewish New Year celebration, because the rise of new cases worried Prime Minister Netanyahu, as many people planned to celebrate with their families that night. [12]

Knowing that the virus affected and will affect the world for a long period of time, how can the world fight the pandemic?

What is R_0 ?

History

To study the epidemic trend of malaria in 1952, George MacDonald constructed population models of the spread of the disease by the application of the basic reproduction number. [13] Denoted R_0 , and pronounced R nought, it is the expected number of cases in a homogeneous population directly generated by one infected individual. For example, if a sick person infects two others, the R_0 is 2.

R_0 is not affected entirely by the infectiousness of the virus, but also on the "the proportion of susceptible people at the start and the density of the population" and "the rate of disappearance of cases by recovery or death, the first of which depends on the time for which an individual is infective". [14]

Managing the spread of COVID-19

For the New York Times, " R_0 is expected to shape our world in the coming months and possibly years as governments and health experts treat it as the closest thing to a compass in navigating the pandemic." [15] That is because, as further explained in the article, governments are increasingly using R_0 as a "metric for whether their country's cases are growing faster than they can manage". That is, in fact, the only mathematical metric currently available to see if lockdowns and other policies are working.

We know that the seasonal flu has an R_0 value of 1.3, and that the SARS outbreak in 2003 was stopped by bringing R_0 down to 0.3 [16], but what is R_0 for COVID-19?

On January 24, a research suggested that R_0 for the disease ranged from 2.0 to 3.1. [17] As other teams produced their R_0 estimates, it became clear that an epidemic was close, as none of the results were below 1.

For the purpose of this research, the Time-Dependant Method is the most precise when the epidemic is still ongoing, because it considers the temporal progress of daily new infected people. In this case, we refer to $R_0(t)$. [15]

Using $R_0(t)$ as a metric of lockdown efficiency in stopping the spread of the virus, we take Lombardy as an example, the most affected region of Italy. At the end of February, $R_0(t)$ was close to 3. Mobility restrictions were established on March 8 and Italy stayed under total lockdown until May 4, when some restrictions were lifted. $R_0(t)$ for Lombardy was at 0.75 at the time, but only two weeks later, on May 17, after people were free to move again, $R_0(t)$ rose again at 0.8, and this trend affected other Italian areas as well. [18]

How is R0 calculated?

Professor Davide Tosi of Università degli Studi di Varese developed a script in R (a programming language for statistical and graphic computing) to calculate $R_0(t)$ for Italy, some of its regions and provinces, using the Italian Civil Protection [open data set](#) and Pierre-Yves Boelle and Thomas Obadia's [R0 package](#) for R, 'a toolbox to estimate reproduction numbers of epidemic outbreaks'. [18] [19]

This CRAN ('Comprehensive R Archive Network') package was published in 2012 and collects several generic methods that estimate transmission parameters during a pandemic. These methods were commonly used during the 2009 N1H1 influenza pandemic, which is estimated to be associated with 151,700 to 575,400 deaths worldwide during the first year it circulated. [20] The methods are divided into two categories, those estimating the initial reproduction number, and those estimating a time dependent reproduction number. We are interested in the second category, in particular at the Time-Dependant Method (TD) proposed by Wallinga & Teunis, published in 2004 while studying SARS, another coronavirus.

As summarised on the R0 package main page, « *this method computes $R_0(t)$ by averaging over all transmission networks compatible with observations. The probability p_{ij} that case i with onset at time t_i was infected by case j with onset at time t_j is calculated as*

$$p_{ij} = \frac{N_i w(t_i - t_j)}{\sum_{i \neq k} N_i w(t_i - t_k)}$$

The effective reproduction number for case j is therefore $R_j = \sum_i p_{ij}$ and is averaged as

$$R_t = \frac{1}{N_t} \sum_{\{t_j=t\}} R_j$$

over all cases with the same date of onset. The confidence interval for R_t can be obtained by simulation. Correction for real time estimation, where not yet observed secondary cases are taken into account is possible. It is possible to account for importation cases during the course of the epidemic. » [21]

Aim and objectives

The aim of this project is the development of a dashboard to visualize $R_0(t)$ in Italy. More specifically, we take in consideration provinces, regions and the whole nation, to later analyse data at various levels. Until now, the download of data and its relative use in functions to calculate $R_0(t)$ has been done manually. This takes a lot of time and it's not efficient. To speed up the process, we developed a tool able to download the necessary data and compute the respective $R_0(t)$. This will help further studies on the topic, by automating a – otherwise – tedious task. In particular, this project is divided into three main tasks:

- Download all the necessary data about daily new cases made available by the Italian Department of Civil Protection through [GitHub](#).
- Run the relative R script in RStudio for each area, to calculate the Time Dependent basic reproduction number for each one of them.
- Gather all the results (plots and daily $R_0(t)$ values) from the previous step and output a PDF file, to visualize the trend of each area.

In order to achieve the aim of the project, two programming languages are required: Python and R. By developing the right script, it would be possible to reach the intended goal.

It's also important to highlight that the developed tool needs to be cross-platform and an Internet connection is required to download new data daily. There aren't any other specific software requirements, because the process shouldn't need many resources and it would likely run for personal use only.

The project was completed in the period of time between July and September 2020 and it is reviewed by Professor Davide Tosi.

Design and development

Solution Design

Used Technologies

As programming languages, we used Python3 and R. At the first stage of my development the tool also included Java code, but later we were able to eliminate it completely, because everything could be handled more efficiently by R alone.

R and Python are both open-source programming languages oriented towards data science. R is a very specific language, built by statisticians for statistical computing and graphics, as “an environment within which statistical techniques are implemented”. That is why, using the R0 package from CRAN, it was easy to calculate the desired numbers and plots.

Python became very popular lately because it’s very easy to learn and because of its several applications, but in its early days Python didn’t have the data analysis libraries it has today. Fortunately, with the release of packages like Pandas in the early 2010s, this general-purpose language and its flexibility gave developers the opportunity to write simple and understandable code oriented towards data management.

It is also worth mentioning that the code was uploaded and shared on GitHub, a Git repository hosting service. Git is a distributed version control system that allows developers to work and share code remotely.

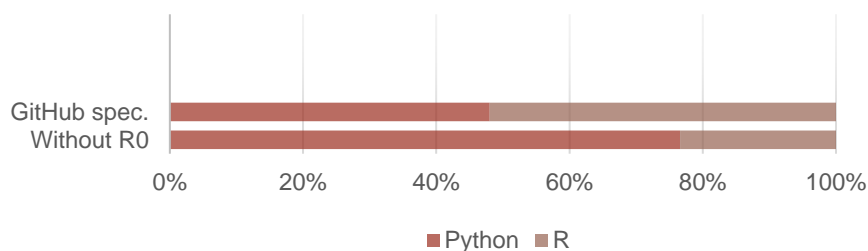


Figure 1, code analysis: used programming languages

In the end, as seen by the GitHub specification graph (Figure 1), the majority of the lines of code in this project are in R, but the actual code developed in R language is far less if we take out the script from R0 package (est.R0.TD.R).

More about the programming languages is discussed in the User’s Guide Appendix.

Task analysis

Following the project requirements, the code has been developed using the following language for the respective task to handle the data:

<i>Download</i>	Python	
<i>Processing</i>	R	
<i>Output</i>	R	Output plots and R0(t) values
	Python	Export results as a .pdf file

As mentioned, at first, Java was involved in the project. This was because we needed a way to run the scripts in the right order, as my Python script only downloaded and saved data as single R files. This way, to output R0(t) for all the areas, we had to source the R0 package method, source 129 R files to import the data (the total number of areas), source a temporary R script that calculated R0(t) for each area, 129 times. This was very time consuming and didn't work for some areas that were randomly left out, as the code had a tendency to jam because of the inefficient process. This issue was solved by using R and by saving the entire data frame instead of only single files for each area.

Project folder structure

The final version of my code is structured as follows:

<i>est.R0.TD.R</i>	R0 package, Time-Dependent Method	
<i>getalldata.py</i>	Downloads all the necessary data and outputs a folder named '_R0(t)data' in the user home directory, containing:	
	_dataframe	Main data frame
	_zones_list	A list of the names of the areas
	zone.2020.R	R file for each area, where 'zone' is the name of said area
<i>mainscript.R</i>	R main script which calls getalldata.py and outputs the desired results in the project folder as:	
	'Plots' folder	All the plots saved as .jpeg
	Log.txt	Console log
	R0t-table.csv	R0(t) values of the day
<i>makeboard.py</i>	Outputs two PDF files with the results	
	R0(t)_values.pdf	List of R0(t) values
	R0(t)_values_plots.pdf	List and trend plots

To see the project workflow, see the [Test Suite](#) section.

Managing data

What was challenging about this project was the data retrieval, because what was relevant to achieve the goal, was the number of daily new cases of COVID-19. That wasn't a problem for the national and regional data, whereas it was for the provincial data.

By looking at the Italian Department of Civil Protection open data set of [national data](#), we notice that the 9th column is named 'nuovi_positivi', which translates to (daily) new cases. The same can be said for the [regional data](#) (13th column).

That is not the case for the [provincial data](#), because this column (and its data) is missing. This required to write an algorithm that calculated this essential data and, as further noticed in the development, and code that handled the data structure construction efficiently. The data used was the 'totale_casi' column (total number of cases from February 24 to the current day). See how this algorithm was written in the [Source Code](#) Appendix.

In *figure 2*, you can see what happens trying to retrieve the data for each day since February 24, by downloading each province data of total positives to the current day. To download data for every one of the 107 provinces of only 4 days, it required 15 seconds, which means that to download data from February 24 to (hypothetically) October 1, the process could take up to 15 minutes. Of course, this is just to download data, which needs to be processed to get the daily number of new cases.

	20200731	20200730	20200729	20200728
denominazione_provincia				
L'Aquila	259	257	256	256
Teramo	638	638	635	635
Pescara	1615	1614	1613	1613
Chieti	841	841	840	837
Potenza	192	192	192	192
...
Belluno	1212	1212	1212	1212
Treviso	2920	2850	2768	2758
Venezia	2839	2816	2801	2788
Padova	4242	4224	4211	4204
Rovigo	459	458	458	458
[107 rows x 4 columns]				
Elapsed time during the whole program in seconds: 0:00:15.100027				

Figure 2, data download for 107 provinces for 4 days (early code)

At the first stage of the *totalToDaily()* function that takes care of the 'totale_casi' column to output the daily new cases number, the time required to output one province data was around 24 seconds (see *Figure 3*). That means that to output the data for every province required around 43 minutes.

```
[0, 0, 0, 0, 3, 1, 0, 0, 3, 4, 6, 6, 4, 5, 12, 6, 25, 23, 27, 33, 26, 18, 32, 31, 45, 28, 21, 27, 35, 29, 18, 34, 209, 5
7, 44, 54, 27, 44, 65, 83, 63, 43, 102, 33, 22, 143, 98, 44, 30, 48, 102, 71, 69, 68, 85, 52, 38, 55, 51, 38, 36, 31, 60
, 29, 72, 51, 48, 38, 10, 68, 55, 53, 127, 55, 75, 16, 18, 14, 77, 29, 20, 13, 41, 3, 13, 9, 12, 10, 23, 44, 16, 5, 9, 1
7, 12, 41, 4, 25, 3, 10, 5, 6, 28, 3, 19, 7, 14, 10, 12, 14, 5, 27, 19, 22, 6, 13, 7, 5, 3, 18, 2, 0, 1, 8, 10, 3, 5, 3,
1, 5, 4, 0, 3, 2, 1, 1, 5, 3, 2, 1, 1, 3, 0, 4, 4, 3, 1, 1, 1, 6, 15, 1, 3, 8, 0, 4, 2, 11, 5, 0]
Elapsed time during the whole program in seconds: 0:00:24.125745
```

Figure 3, totalToDaily() execution for one province (early code)

If the code wasn't optimised, it would have been necessary to download data for about one hour. Of course, these tests differ depending on Internet download speed (for comparison the Internet speed of the machine that was used for these tests offers about 44Mbps in download).

This process has been optimized, and in the end, the *getalldata.py* Python script, which downloads all the data, takes about 6 seconds to finish.

What started as the most time-consuming task, ended up as the fastest, as now it is the 'est.R0.TD.R' script from the R0 package that takes up most of the process runtime, for a total of about 2.5 minutes. (Figure 4) This time could vary a lot depending on the computing power of one's machine, but the project aim was achieved in an overall optimal solution.

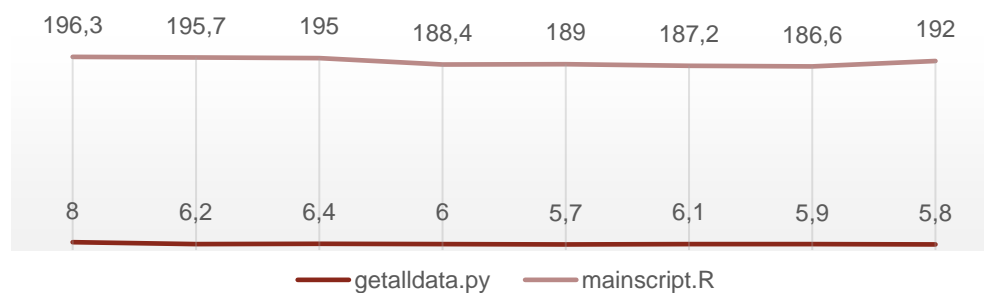


Figure 4, execution time in seconds

Pandas, a Python library, is able to handle data, and in particular data frames, efficiently. As mentioned before, Python wasn't initially made to operate on huge amounts of data, and the standard matrix is a list of lists.

A Pandas DataFrame is a "two-dimensional, size-mutable, potentially heterogeneous tabular data". It is a powerful tool that allows to handle data easily. For this project, it was useful to translate the CSV files uploaded to GitHub into a table. Pandas also gives the opportunity to efficiently clear data and select columns, rows and cells as needed (e.g. eliminate duplicates and merge two DataFrames).

Although the final data frame is only made by only 129 rows and a variable number of columns depending on the current day (for 220 days, 28.380 elements), the provincial data set (at the time I'm writing this section) is made of 11 columns and 28728 rows. This makes a total of 316.008 elements, a number growing daily for 107 provinces, which wasn't easy to handle with standard Python lists.

Code Libraries

Code libraries are a collection of routines to reference in your code. They provide different methods based on the required function by the program during execution. For this project, the following libraries are used:

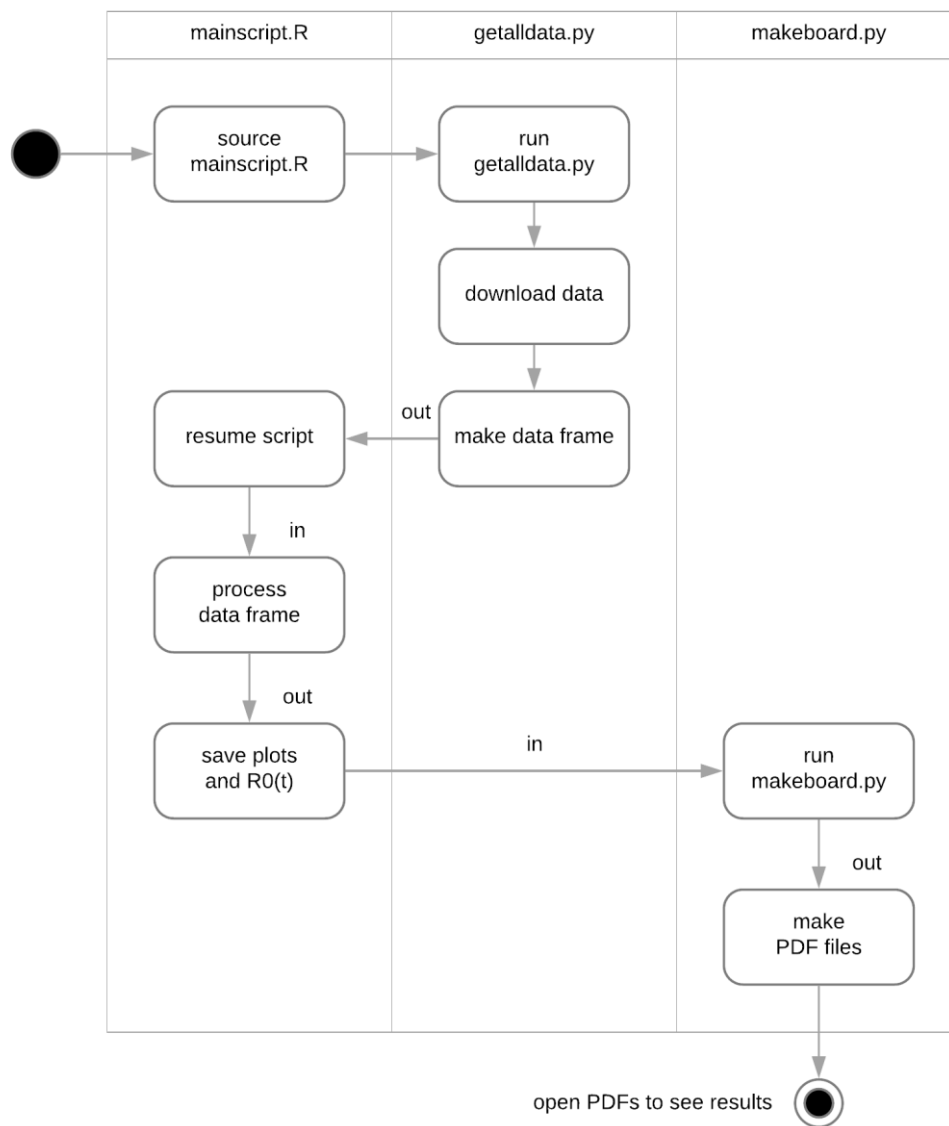
<i>Python libraries</i>	
<i>NumPy</i>	Although not directly implemented in the code, Pandas is built on the NumPy package ('Num' for numerical, 'Py' for Python), which provides objects as fast multi-dimensional arrays.
<i>Pandas</i>	A data manipulation tool to store and elaborate data, Pandas provides a high-performance table-like object called DataFrame, and relative methods to export and import sets of data.
<i>FPDF</i>	Ported from PHP's FPDF class, PyFPDF is a high-level PDF generator for Python.
<i>os, sys, pathlib, datetime, time</i>	These modules provide different functions to handle directories, the user's operating system name acquisition, current time retrieval for time stamps and dates.
<i>R libraries</i>	
<i>R0</i>	Documented methods to estimate the basic reproduction number for disease outbreak study as a R package. It provides different plots options based on the chosen method.
<i>reticulate</i>	To run Python scripts, import Python modules, translate Python objects in R, the reticulate package embeds a Python session within R.
<i>rstudioapi</i>	Easily access RStudio API, for example to handle working directory changes during R sessions.

To read instructions on how to install some of this libraries to support code execution, see the [User's Manual](#) Appendix.

Activity diagram

A UML Activity diagram is a flowchart that represents the flow from one activity to another, from a start point (full black dot) to an end point (lined black dot).

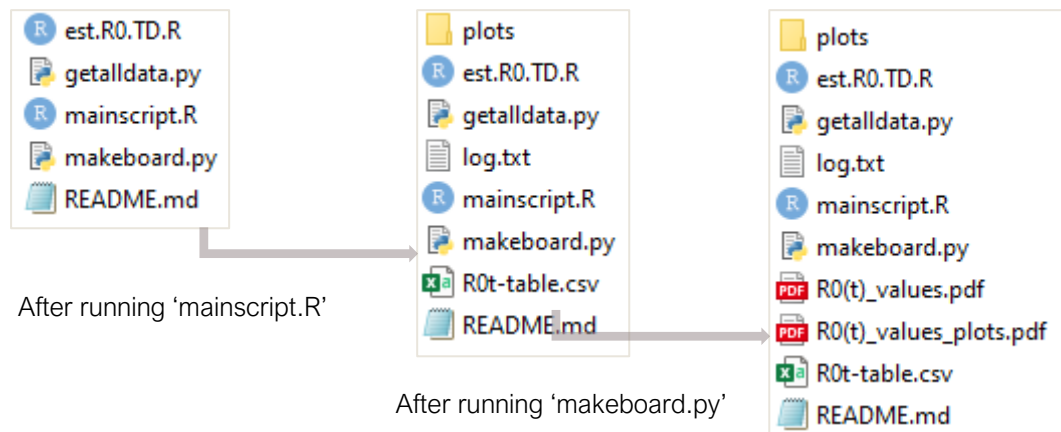
In the following diagram you can see the workflow of the three scripts and how they interact with each other in order to complete the process.



Test Suite

Expected results

After running the code, if the process was completed successfully, the project folder will look like this:



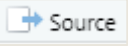
The plots of the R0 trend of each area are saved into the 'plots' folder. In the file 'R0t-table.csv' is saved R0(t) for all 129 areas. The areas are 129 because we consider the entire nation, plus 20 regions, plus 107 provinces, for a total of 128 areas, but region Trentino-Alto Adige is divided into Provincia Autonoma di Bolzano and Provincia Autonoma di Trento because decision-making power is exercised directly at provincial level.

In 'log.txt' you can see if the script processed every row of the data frame as intended, by writing an error line for those who couldn't be calculated correctly. In 'R0(t)_values_plots.pdf' it is possible to see the results of this project in PDF format.

It's expected to have 129 JPEG images in 'plots', 129 rows in 'R0t-table.csv', 129 lines of the type '`[1] "1 Italy"`' in 'log.txt' and 129 lines of the type '`ID 1, Italy - R0(t) = x.xxx`' in 'R0(t)_values_plots.pdf' (where '1' and 'Italy' are different for each area).

See how to set up your machine to run this project in the [User's Manual](#) Appendix.

Runtime example

- Source 'mainscript.R'. You can either click on the  icon or simultaneously press CTRL + SHIFT + S.
- Your global environment tab should look like Figure 5.

dataframe	129 obs. of 213 variables
df	65 obs. of 2 variables
	x.Italy. : chr "Italy" "Abruzzo" "Calabria" "Ca
	x.O.909478164568386.: chr "0.909478164568386" "
names	129 obs. of 1 variable
values	
counter	129L
end_time	2020-09-23 16:54:32
lastTD	1.34580208019893
or_path	"C:/Users/Alice/Desktop/covid"
os	Named chr "windows"
path	"C:/Users/Alice/Documents/_R0(t)da
start_time	2020-09-23 16:51:53
v	chr [1:2] "Padova" "1.345802080198
val	129L
Functions	
createPlot	function (val)
est.R0.TD	Large function (1 MB)

Figure 5, Environment tab on RStudio

- A folder called 'plots' will be created. Here you can find plots named after the area they refer to and an index (out of 129). You can see the progress of the process by looking at the images being created. (Figure 6)

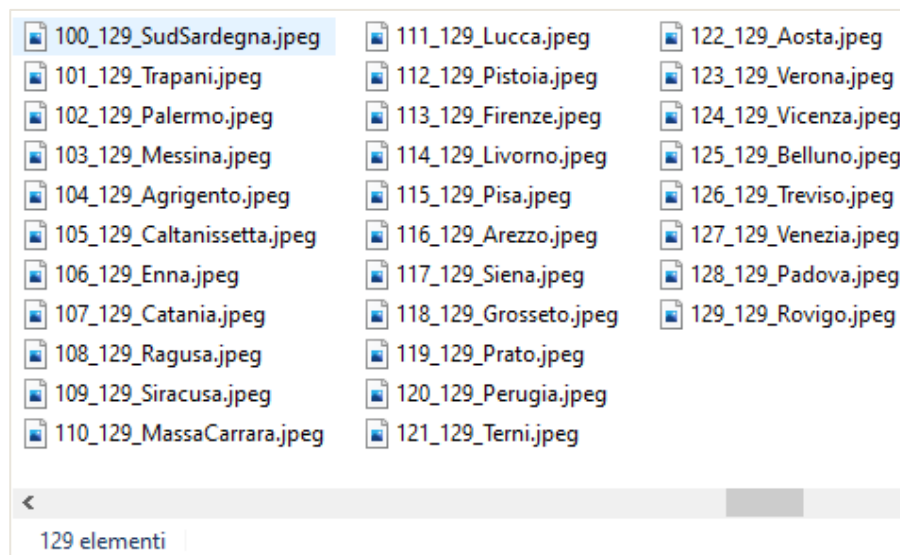


Figure 6, 'plots' folder at the end of 'mainscript.R' execution

- In 'log.txt' you can see what happened during runtime.
- If you want to export your results in PDF files, you can run 'makeboard.py', either by clicking on it or running it through command line.

Conclusions

Results

This thesis project aimed at developing a tool able to download, process and export data about the contagiousness level of COVID-19 in Italy, through the calculation of $R_0(t)$. This number shows how many people can be infected by a person with the disease. The assignment was divided into three main tasks which, at the end, have been merged into a single automatic process, and it is possible to see the results in dedicated files. (Figures 7-8)

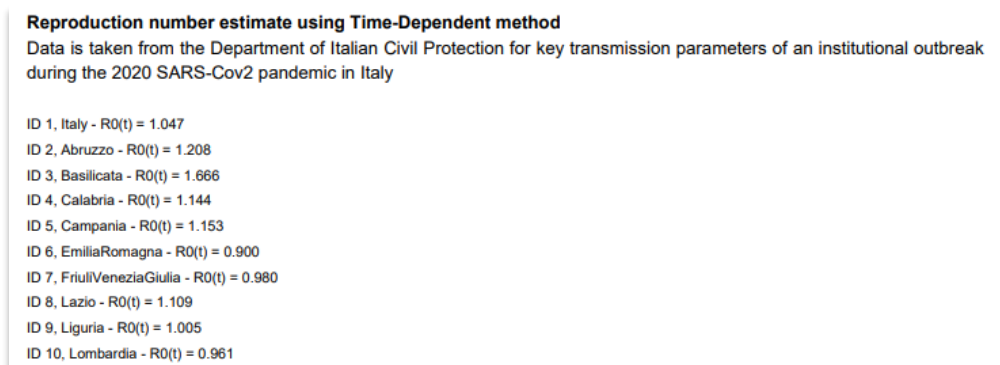


Figure 7, $R_0(t)$ in 'R0(t)_values.pdf'

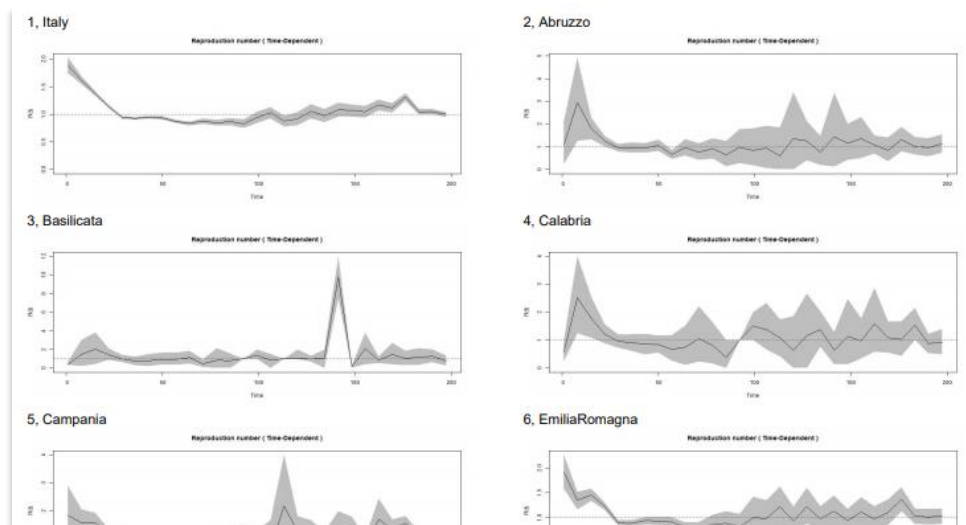


Figure 8, plots in 'R0(t)_values_plots.pdf'

The results differ based on the chosen number of days, because the plots are based on the aggregation of initial data by longest time unit, such as weekly incidence. We can change this number and plots will be different. As the longest time unit grows, the spikes will be less and the line smoothed. However, as explained in the [Threads to Validity](#) section, this number is set at 7.

It is possible to say that the project requirements have been met, although there is still room for improvement.

Plot comparison when changing longest time unit

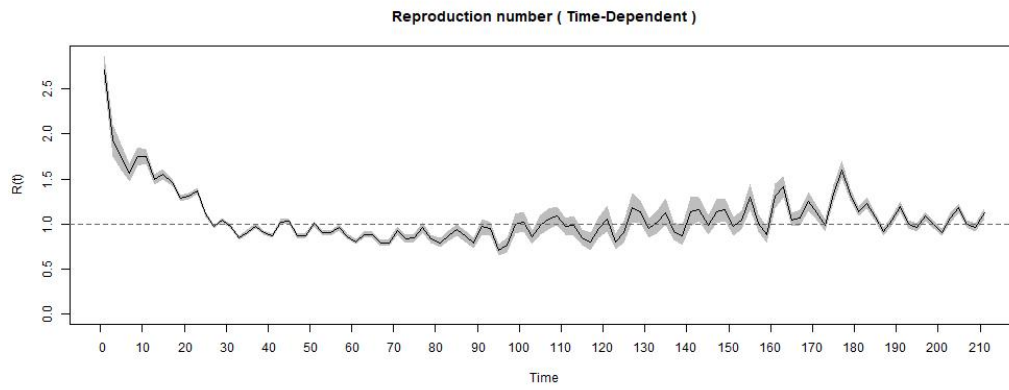


Figure 9, number of days: 2

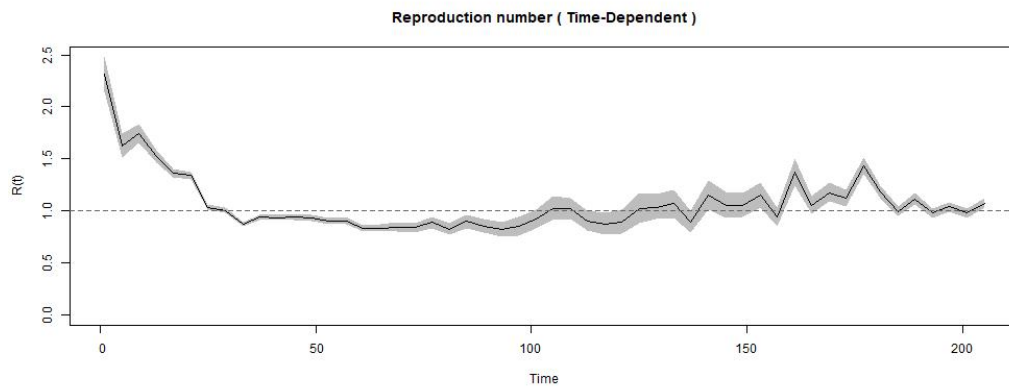


Figure 10, number of days: 4

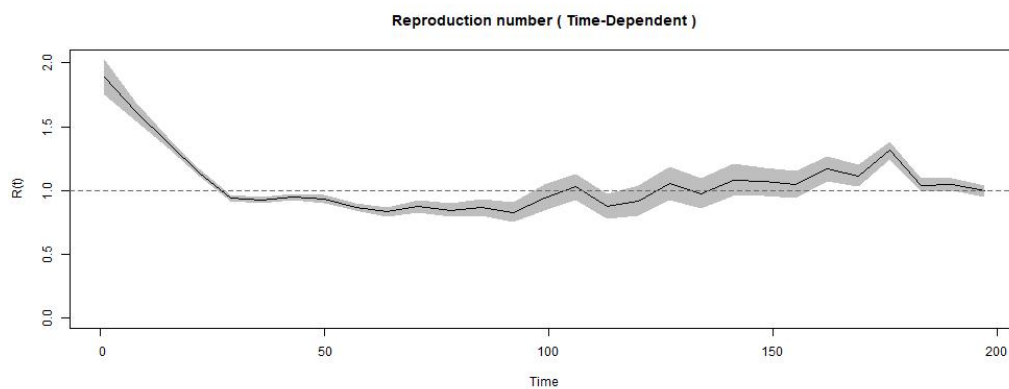


Figure 11, number of days: 7

Threads to Validity

In order to achieve the desired set of results, data had to be manipulated. In fact, the data frame created by `getalldata.py` is not equal to what it downloads from the Civil Protection data set. That is because the Time-Dependant method didn't work with a lot of areas. It wasn't possible to determine the exact cause of this problem, but the best configuration of parameters requires to have a longest time unit number equal to 7 (Figure 11) and a number of consecutive zeros equal or less than 6, to be interrupted by a single 1.

There was also a problem with areas that registered a number of cases less than 3 on the first day, so if the row began with a 0, 1 or 2, it was changed to 3.

This manipulation doesn't make the representation of the daily new cases per area totally accurate. However, for the purpose of this project, a compromise has been reached.

It is possible to change these parameters, but it is not recommended, because the tool will not guarantee an output of results for a number of areas. The names of these areas can be read in the log file, with its relevant warning message.

Another issue was with the compatibility among different operating systems. To ensure the tool performance and reliability, several tests and fixes were made. In the end, this tool was tested for Windows, MacOS and Linux.

Further development

The possible development of this project is to find the right data manipulation algorithm, that makes possible to process all the data for the desired length of days for the longest time unit number. This number is strictly correlated to the accepted sequence of zeros, but it needs more tests to reveal the source of the problem, and not a trail-and-error approach.

This project will be expanded by Alessandro Riva, by developing an online service to show plots about the COVID-19 pandemic.

His code will automatize the visualization and representation of the data downloaded and processed by the project discussed in this thesis. To elaborate the plots, Riva will use Plotly, and Dash for the management of the dashboard.

Currently results can be seen on the website covid19-italy.it, developed and hosted by Prof. Davide Tosi and Alessandro Riva.

Appendices

User's Manual

Software requirements

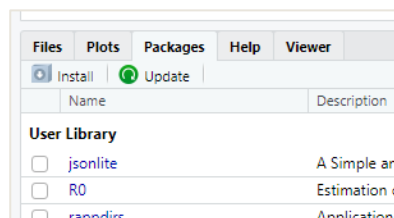
This project was developed using the following versions of the required software. It may not be needed, but in order to successfully run this project, it is recommended to use the specified version or higher.

- [Python](#), 3.8.5
- [R](#), 4.0.2
- [RStudio \(Desktop\)](#), 1.3.1073

RStudio is an *Integrated Development Environment* (IDE) for R. It is available as a regular desktop application or as a cloud version that runs on a remote server, accessible via web browser. Its development studio has no connection to the R Foundation that supports the development of the R language as a non-profit organisation. In fact, RStudio is available in open source (free to use) and commercial editions.

Installation

- Open `mainscript.R` in RStudio.
- In RStudio, go to Packages and Install.



- Download `Reticulate` package.
- Uncomment line 11: `#py_install("pandas")`
Run it. (CTRL + Enter)

```
7 #Loading packages
8 library(R0)
9 library(reticulate)
10 library(rstudioapi)
11 #py_install("pandas")
12
```

- When running, RStudio will ask to download `Miniconda`: press 'Y' and enter.
- Comment the previous line. You can now source the file.
- To run '`makeboard.py`' install 'FPDF' via command line by writing

```
pip install fpdf
```

You can now run the script and output two PDF files.

Test run

You can source the file '*mainscript.R*' by pressing CTRL + SHIFT + S or the 'Source' button.

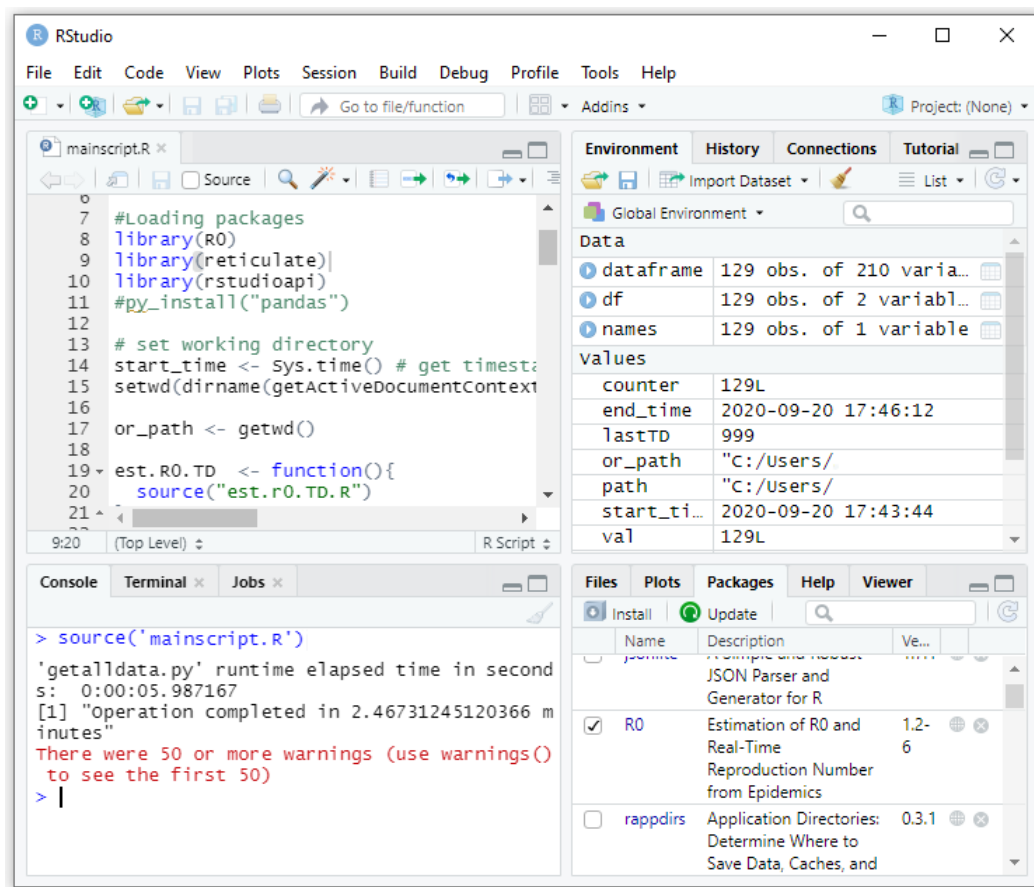


Figure 12, RStudio after sourcing 'mainscript.R'

Your RStudio window should look like Figure 12 when the script has ended.

In the first quadrant there is the code. In the second quadrant under the label 'Environment' all the variables and data sets processed by the script will be displayed.

You will be notified of the end of the process by the console in the third quadrant, with the phrase 'Operation completed in [...] minutes'. If something went wrong, like some area wasn't processed correctly, a phrase in red will notify of a warning. If you want to inspect which area, you can check the console log output saved in 'log.txt' inside your folder.

In the fourth and last quadrant, you can see your packages and how the R0 package is checked.

(Nota Bene: this image was altered to exclude machine-dependent paths)

Source Code

The following [code](#) snippet (Figure 13) demonstrates how we were able to extract the number of daily new cases for provinces, a number that wasn't provided by the Italian Department of Civil Protection data set.

```
# provinces .csv files don't have a daily cases column,  
# to return a list with these values, this function makes this subtraction:  
# 1) total cases _minus_ 2) cases of yesterday  
# _minus_  
# 3) total cases of the day before yesterday  
  
def totalToDaily(prov):  
    sub = 0 # 3) total cases of the day before yesterday - starts at 0  
    returnList = []  
    for d in range(0, len(dates_vector)):  
        dif = p_df.loc[prov, d] # 1) total cases  
        fullsub = dif - sub # 2) total cases _minus_ cases of yesterday  
        if fullsub < 0: # if it is below zero, set it at 0  
            fullsub = 0  
        returnList.append(fullsub)  
        sub = dif # sets new sub of the day for the next day  
    return returnList
```

Figure 13, code snippet from 'getalldata.py'

This function is called by *ProvinciaDaily()*, because it returns a list of numbers of daily new cases starting from February 24, which is later translated as a String of numbers for *makeFile()*, which outputs the single R file. This list is appended to the main frame as a new row.

It takes as arguments the name of the province, while the number of total cases is stored in the 'p_df' data frame (p as 'provincial' and df as 'data frame'), under the date this number was released).

To obtain the number of daily new cases, this function makes two subtractions for a number of iterations equal to the number of days. This is the logical statement behind the algorithm:

To obtain the number of today new cases, we need to subtract to the number of total cases of today the number of cases of yesterday, and the number of total cases of the day before yesterday.

Here is an example of 5 iterations of the algorithm for the province of Venice from February 24 to February 28, 2020. (Figure 14)

	Date	Variable	Result	
1)	24/2	Sub starts at 0	0	Number of new cases on 24/2 in the prov. of Venice is 0
		dif = p_df.loc['Venezia', 0]	0	
		fullsub = 0 - 0	0	
		sub = dif	0	
2)	25/2	dif = p_df.loc['Venezia', 1]	7	Number of new cases on 25/2 in the prov. of Venice is 7
		fullsub = 7 - 0	7	
		sub = 7	7	
3)	26/2	dif = p_df.loc['Venezia', 2]	8	Number of new cases on 26/2 in the prov. of Venice is 1
		fullsub = 8 - 7	1	
		3)total cases of the day before yesterday sub = dif	8	
4)	27/2	dif = p_df.loc['Venezia', 3]	14	Number of new cases on 27/2 in the prov. of Venice is 6
		2)cases of yesterday fullsub = 14 - 8	6	
		sub = dif	14	
5)	28/2	1)total cases of today dif = p_df.loc['Venezia', 4]	15	Number of new cases on 28/2 in the prov. of Venice is 1
		fullsub = dif(15) - sub(14)	1	
		sub = dif	15	

Figure 14, example of totalToDaily() function iteration

As you can see, the orange number is the **total number of cases for each day**, the value that is stored in p_df. The red number is the desired **result of the algorithm**, the number of new cases on that day.

February 27 was highlighted as an example. The number of total cases of today (15, on February 27), minus the number of cases of yesterday (6, on February 26), minus the number of total cases of the day before yesterday (8). The result is 1, which is the right number.

Bibliography

1. McIntosh K. (1974) Coronaviruses: A Comparative Review. In: Arber W. et al. (eds) Current Topics in Microbiology and Immunology / Ergebnisse der Mikrobiologie und Immunitätsforschung. Current Topics in Microbiology and Immunology / Ergebnisse der Mikrobiologie und Immunitätsforschung, vol 63. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-65775-7_3
2. Statement by WHO, 'Disease outbreak news : Update' on 12 January 2020
3. Article by National Geographic, 'Wet markets' likely launched the coronavirus. Here's what you need to know' by Dina Fine Maron, on April 15, 2020.
4. Article by Dina Fine Maron for Scientific American, "Where Does Ebola Come From?", December 30, 2014
5. Article by Avert, 'Origin Of Hiv & Aids', last updated on October 30 2019
6. News article by EuroNews, "Coronavirus: Half of humanity now on lockdown as 90 countries call for confinement", 3 April 2020.
7. Response by Lorenzo Tomasin for Accademia della Crusca, 20 March 2020
8. Etymologia: Quarantine. Emerg Infect Dis. 2013;19(2):263. <https://dx.doi.org/10.3201/eid1902.et1902>
9. Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand. <https://doi.org/10.25561/77482>
10. Scientific brief by WHO, "Transmission of SARS-CoV-2: implications for infection prevention precautions", 9 July 2020
11. Article by The Guardian, "CDC tells health officials to expect a coronavirus vaccine by November". September 2, 2020
12. News article by BBC, "Coronavirus: Israel marks Jewish New Year with second lockdown", September 19, 2020
13. Macdonald G. The analysis of equilibrium in malaria. Trop Dis Bull. 1952;49(9):813-829. <https://pubmed.ncbi.nlm.nih.gov/12995455/>
14. Article by Oxford COVID-19 Evidence Service Team, "When will it be over?: An introduction to viral reproduction numbers, R0 and Re", April 14, 2020
15. Article by Max Fisher for the New York Times, "R0, the Messy Metric That May Soon Shape Our Lives, Explained", April 23, 2020
16. Article by Knvul Sheikh, Derek Watkins, Jin Wu and Mika Gröndahl for
17. The New York Times, "How Bad Will the Coronavirus Outbreak Get? Here Are 6 Key Factors", February 28, 2020
18. Davide Tosi, "Andamento del Tasso di Contagiosità R0(t) in Italia, Regione Lombardia e Province Simbolo", May 22, 2020
19. Davide Tosi, Alessandro Siro Campi, "How Data Analytics and Big Data can Help Scientists in Managing COVID-19 Diffusion: A Model to Predict the COVID-19 Diffusion in Italy and Regione Lombardia"
20. Article by Centers for Disease Control and Prevention, National Center for Immunization and Respiratory Diseases (NCIRD), 'Ten Years of Gains. A Look Back at Progress Since the 2009 H1N1 Pandemic', June 11, 2019
21. Jacco Wallinga, Peter Teunis, 'Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures', American Journal of Epidemiology, Volume 160, Issue 6, 15 September 2004, Pages 509–516, <https://doi.org/10.1093/aje/kwh255>