Covid 19

1 Assignment Description

These assignment sheets describe the modelling of Covid 19. They provide a brief description of the topic and set problems, some of which must be solved using python programs.

The assignment consists in writing an essay that contains an answer to each question and which expands on the material included in these notes. The references are there to provide you with extra sources of information and they are not exhaustive.

The essay must be readable on its own without reference to the assignment sheets. The essay needs not necessarily follow the same order as the notes and does not need to answer the questions in the same order either. As a matter of fact, using a different structure and order is a bonus. Do describe the interpretation or consequences of the results that you obtain. We also expect you to focus more on some aspects of the material presented in these notes. This could be a more detailed discussion of the derivation of the equations or algorithm described in the notes, or to better explain the interpretation of the results. You can also solve problems which are not set and which answer some further questions that you might have.

Some parts of the assignment sheet will need to be included in your essay, but do not copy these sections verbatim, use your own words. The equations do not need to be changed, but you can provide more detailed explanations or derivations when appropriate.

You have to submit 3 files: 1 pdf file for the essay and 2 python programs. The essay must be typeset in LaTeX.

2 Introduction

Covid 19 is a one in a century epidemic event which has taken the world by surprise in 2020. With a high infection rate and an estimated 1% fatality rate[1], drastic measures needed to be taken to prevent a sanitary disaster. To make their decisions, governments asked mathematicians and epidemiologists to model the epidemic and forecast the impact of possible measures that one could take.

The aim of this project is to model the epidemic mathematically during its onset, before measures were taken to slow its progression.

3 Simple SIR Model

The simplest epidemiological model of a disease is one where individuals become infected and then recover or die after a known length of time. Moreover, we assume individuals are infectious throughout the illness.

We then split the population in 4 groups: *susceptible*: those who have not been infected yet; *infectious*: those who are ill and infectious and then *recovered* or *fatalities i.e.* those who have recovered or those who haven't.

People become infected when they interact with other people. The probability to become infected then depends on the number of people they meet as well as the duration and conditions of these meetings. It also depends on how easily the virus can be transmitted between individuals. Infection is also proportional to the probability that a given person is infected.

If I_i is the total number of people who are infectious on day i, then the number of people who will become infected in one day is proportional to the number of susceptible persons on that day, S_i, the probability that a given person is infectious, I_i/N where N is the total population, and then a coefficient K which captures all the other parameters together.

This is described mathematically by

$$\begin{aligned} \Delta I_i &= K S_i I_i / N\,, \\ S_{i+1} &= S_i - \Delta I_i\,, \\ I_{i+1} &= I_i + \Delta I_i\,. \end{aligned} \tag{1}$$

where ΔI_i is the number of people who become infected on day i. Notice that the number of susceptible people decrease by the same amount than the number of exposed people increases.

After d days of infection, the people recover or die and we can write

$$\begin{aligned} R_{i+1} &= R_i + (1 - K_f)\Delta I_{i-d}\,, \\ F_{i+1} &= F_i + K_f \Delta I_{i-d}\,, \\ I_{i+1} &= I_i - \Delta I_{i-d}\,, \end{aligned} \tag{2}$$

where K_f is the fatality rate.

Putting these together we get

$$\begin{aligned} S_{i+1} &= S_i - \Delta I_i\,, \\ I_{i+1} &= I_i + \Delta I_i - \Delta I_{i-d}\,, \\ R_{i+1} &= R_i + (1 - K_f)\Delta I_{i-d}\,, \\ F_{i+1} &= F_i + K_f \Delta I_{i-d}\,. \end{aligned} \tag{3}$$

Question 1:

> At this stage we need to perform a test of our model. We know that all individuals must be accounted for at all times. Check that the total population $S_i + I_i + R_i + F_i$ does not vary over time.

To solve this model, we also need to decide how it starts. The simplest is to assume that S_0 is the total population being considered, say 66 millions for the UK. We then need at least 1 infected person to kick start the epidemic, so we take $I_0 = 1$. Finally, we have $R_0 = F_0 = 0$.

To describe the rate of an epidemic, epidemiologists use the number of individuals that, on average, an infected person contaminates. This parameter is usually called R. In our model, as the infection lasts d days, the relation between K and R is simply

$$K = \frac{R}{d}. \tag{4}$$

This model is similar to the population dynamic models we have seen so far in the course except that we need to use the population levels several days in the past. Implementing this in Python is not difficult. First of all, we can use a list for the 4 categories of people as well as for ΔI and initialise them. We should also initialise the different model parameters

```python
d = 7  # Infection duration
Rpar = 2 # R is used below for the recovered population
K= Rpar/d
S = [66e6]
I = [1]
R = [0]
F = [0]
DI = [0]                  # a list for all the Delta I
Pop=S[-1] + I[-1] + R[-1] # the population still alive
```

We then need to use (3) to evolve the population. Instead of using index i explicitly, we use negative index, so that we access the list elements from the end. Index -1 corresponds to index i in (3). We can then compute the right hand side of (3) and append each value to the corresponding list:

```
DeltaI = K*S[-1]*I[-1]/Pop
S.append(S[-1]-DeltaI)
I.append(I[-1]+DeltaI-DI[-d])
R.append(R[-1]+(1-Kd)-DI[-d])
F.append(R[-1]+Kd-DI[-d])
DI.append(DeltaI)
```

We have a slight problem though: when the program starts, we need to access element d of DI from the end, but it does not exists. The solution is simple, we just initialise DI with at least d zeros `DI = [0]*d` (this creates a list containing d zeros). We also need to take into account that the first infected person was infected in the past and must be included in the history of infection DI. As a result we should use `DI = [0]*d+[I[-1]]` to initialise DI.

The program `simple_SIR.py` contains all the code described above where the model iteration is included in a loop. It also include some code to display the Infected, Recovered populations and Fatalities as a function of time.

Question 2:

> Run the program `simple_SIR.py` for R (the model parameter) taking the values, $3, 2, 1$ and 0.9. What can you conclude from the results? Copy the program `simple_SIR.py` into `simple_SIR_log.py` and replace the function `plot` by `semilogy` to generate logarithmic plots. This makes it easier to read the data. You can also insert the line
>
> ```
> plt.savefig("SIR_R{}_tmax{}.pdf".format(Rpar,tmax))
> ```
>
> before the line `plt.show()` to save the graph into a file using a name containing the value of R as well as the duration of the simulation.

This model is very simple, but there are a couple of aspects that it does not take into account: being infectious is not an all or nothing property, it evolves with time and between individuals. Epidemiologists have identified probability distributions which characterises the probability of infecting someone else as a function of time since infection. Similarly, people do not recover or die exactly after a fixed number of days. This very much varies between individuals and is also described by a probability distribution. We will now take these distributions into account.

3.1 Covid 19 SIR Model

A general model of infection consists in 4 stages: individual who are initially non-infected are described as *susceptible* to contract the virus. If exposed to infected person, they can contract the virus and are then described as *exposed*: they carry the virus but do not contaminate others yet. After a few days they become *infectious* meaning that they can contaminate other people. Ultimately, they either recover, a state called *recovered* or do not recover (a state we refer to as *fatalities*).

To make matters worse, being exposed or infected is not the same as exhibiting symptoms. With Covid 19 many people do not have symptoms at all and those who do have some usually experience them after they have become infectious. As a matter of fact we have no easy way to tell if someone has been exposed or even if they are infectious. Testing can detect people who are infected, but only a few days after the infection has happened. People who have been infected then stop being infectious before they recover, so we would in principle need an extra state to cover these people.

What this means is that the difference between *exposed* and *infectious* is somewhat academic from a modelling point of view because we don't really know when people become infectious once infected. We will thus consider the time when people become infected instead.

Now, the probability to be infected on day d will be proportional to the probability to meet someone who was infected on day $d-1$ time the probability, P_{Inf} that a person is infectious after 1 day, plus the probability that they were infected on day $d-2$ time the probability of infection after 2 days, plus ... In other words

$$\Delta I_i = R S_i \sum_{d=1}^{n_i} \frac{\Delta I_{i-d}}{N} P_{Inf}(d) \tag{5}$$

where n_i is the number of days such that $P_{inf}(d) = 0$ if $d > n_i$. In [3] it was shown that the distribution of recovery and death were very similar[4], we can then use the same distribution for recovery and death after infection and we call it P_r. We then have

$$\begin{aligned} \Delta R_i &= (1 - K_f) \sum_{d=1}^{n_r} \Delta I_{i-d} P_r(d), \\ \Delta F_i &= K_f \sum_{d=1}^{n_r} \Delta I_{i-d} P_r(d). \end{aligned} \tag{6}$$

Here n_r is the number of days such that $P_r(d) = 0$ if $d > n_r$. We can now insert (5) and (6) into (3) to get a more realistic model:

$$\begin{aligned} S_{i+1} &= S_i - \Delta I_i\,, \\ I_{i+1} &= I_i + \Delta I_i - \Delta R_i - \Delta F_i.\,, \\ R_{i+1} &= R_i + \Delta R_i\,, \\ F_{i+1} &= F_i + \Delta F_i\,. \end{aligned} \tag{7}$$

We now need expressions for the probabilities P_{Inf} and P_R. In [1] the following was argued:

- The probability distribution of infecting others during the days following infection is $P_{Inf} = \mathrm{Gamma}(6.5, 0.62)$.
- The probability distribution of incubation, *i.e.* of showing the first symptoms, is $P_{inc} = \mathrm{Gamma}(5.1, 0.86)$ [2][1].
- The probability distribution of recovering after *incubation* is $P_R = \mathrm{Gamma}(18.8, 0.45)$.

Gamma is a lump-like distribution which we illustrate graphically below. Python will compute it for us, so we do not need to worry about its analytical expression. The first argument is the average value, the second describes how spread out it is.

The probability distribution of recovering after *infection* if then given by

$$P_r(d) = \sum_{n=0}^{d} P_{Inc}(n)\, P_R(d - n). \tag{8}$$

It can be read as follows: the probability that an individual recovers 3 days after being infected, is the probability that they incubated for 3 days times the probability that they recovered immediately after incubation, plus the probability that they incubated for 2 days and recovered 1 day later, plus the probability that they incubated for 1 days and recovered 2 days later, plus the probability that they incubated 0 day and recovered after 3 days.

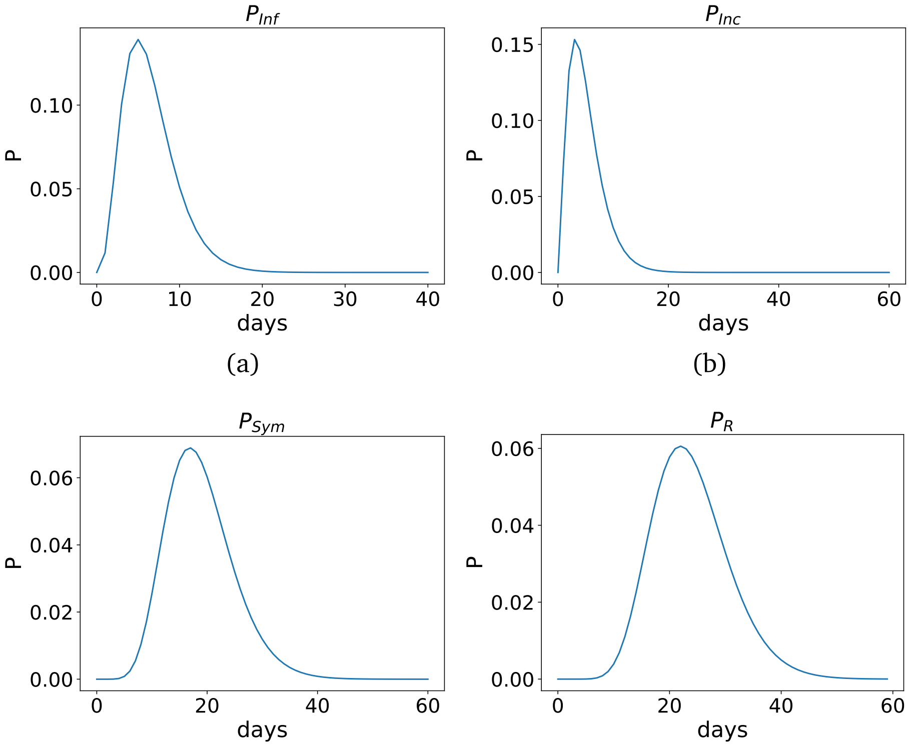
You do not need to worry about which functions these distributions correspond to as they are already coded in the provided programs. What matters more is what they look like. The actual distributions are shown on figure 1.

Question 3:

> Assuming that $P_{Inc}(d)$ and $P_R(d)$ are 0 outside the domain $d \in [0, N]$ for some N, show that $P_r(d)$ is 0 for $d < 0$. What is the largest value of d for which $P_r(d)$ can be non-zero? Then show that it is a probability distribution *i.e.* that $P_r(d) \geq 0$ for all d and $\sum_{d=0}^{\infty} P_r(d) = 1$. Solve the problem explicitely without refering to any known theorem. Hint : the probability distribution domains can be extended by setting all values on the extended domain to zero.

While epidemiologists use the infection rate R to characterise epidemics another parameters one can use, which is easy to determine from data, is the time needed for the number of fatalities to double. If the number of fatalities is of the form $F = A\exp(\lambda t)$ the doubling time τ is defined as the value for which

$$A\exp(\lambda(t + \tau)) = 2A\exp(\lambda t). \tag{9}$$



(a) (b)

(c) (d)

Figure 1: Probability distributions: a) Infectiousness after infection b) Showing symptoms after infection c) Recovery from symptoms d) Recovery from infection after incubation.

Cancelling out the common terms we have $\exp(\lambda\tau) = 2$ and so

$$\tau = \frac{\log(2)}{\lambda}. \tag{10}$$

We are now ready to solve the model equations and to compare them with real data. One issue we have is to chose the correct data set. Two sets have been widely published for every country in the world: the number of cases recorded and the number of fatalities. The problem with the number of recorded cases is that it depends very much on the number of tests that were performed and the protocols that were used to select the people who were tested. This varied greatly between countries and over time. Moreover, many people were not tested because they never developed symptoms. Fatalities on the other hand were recorded more systematically, partly because there already exists procedures to do so, partly because these were the extreme cases. The main problem from a predicting point of view is that there is a 2 to 3 weeks lag between the time of infection and the time of death meaning that monitoring the epidemic with fatalities is not easy. Nevertheless for this project we will use the fatalities data during the onset of the epidemic, *i.e.* before confinement measures were taken.

We will look at 2 countries: the United Kingdom, because this is were we live, and Brazil because few measures were taken at the epidemic onset and, as a results, it fits our model quite well.

Coding task 1:

The supplied program `run_lin_regression.py` loads the fatalities data from the two files `data_UK_tot.txt` and `data_BR_tot.txt`. It uses a fit function from the module `lin_regression.py` which is provided as a skeleton file, and which you need to complete.

Complete the definition of the function `fit(data, dmin, dmax, Country)` in `lin_regression.py` so that it fits the data with the function $f(t) = A\,\exp(\lambda t)$. The fitting must be performed using the **log** of the data, selecting the days between dmin and dmax (excluding that last point). It must then generate a logarithmic graph (using the semilogy function) of the number of fatalities as well as the fitted curve within the range dmin, dmax.

Finally, the fit function must return a tuple (country, A, lambda, doubling_time), where the last value is the doubling time τ of the epidemic.

The file to submit is lin_regression.py.

At the onset of the epidemic, the number of infected people is much smaller than the total population and we can thus assume that S is constant and equal to N. Moreover, the number of people who have recovered or passed away will also be very small and (7) reduces to

$$I_{i+1} \quad = \quad I_i + \Delta I_i. \tag{11}$$

If we consider that the infection lasts exactly d days and neglect the recoveries, then the number of people who become infected on day i is the number of people who became infected on day $i-d$ times R. Now, the number of people who became infected on day $i-d$ is the total number of people infected on day $i-d$ less the number of people infected the day before. In other words, we have

$$I_{i+1} \quad = \quad I_i + R(I_{i-d} - I_{i-d-1}). \tag{12}$$

Question 4:

Assume a solution of the form $I_i = A\exp(\lambda i)$, substitute it in (12) and derive an expression for R as a function of λ and d.

Use the program run_lin_regression.py to estimate λ for both the United Kingdom and Brazil. Taking $d = 6.5$ estimate the values of R for the 2 countries.

4 Solving the SIR Covid 19 model

The program covid19.py solves the model equations (7) with (5) and (6). It defines the class Covid which is a subclass of Covid_base defined in the module covid19_base which contains most of the functionality to solve the model equation.

The program run_covid19.py uses the class Covid to solve the model equation for specific parameters and displays the solutions graphically.

The program run_covid19.py proceeds as follows:

- Sets the duration of the integration dmax.
- Create an instance of Covid specifying Rpar which is the epidemic rate R, the fatality rate Kf and the total population.
- Sets the initial condition. dmax specifies the longest integration one can perform (to create arrays sufficiently large) and I0 is the initial number of infected people.
- Read a data file containing actual data.
- Plots the profile of the different probability distributions. (This can be commented out as we only need to do this once)
- Integrate the equation for the specified duration.
- Generate on the same graph a logarithmic plot of a) the infected model population as a red line; b) the recovered model population as a blue line; c) the number of fatalities for the model as a black line; d) the number of actual fatalities as black stars.

  When these graphs are generated, the values smaller than 1 are removed as they correspond to a very small probability of having someone in that category.

In the program, the class variables `S`, `I`, `R`, `F`, `Drd`, `Dr` and `Df` are all arrays for which item `d` corresponds to the value for the corresponding day. `DI` is also an array, but it contains `self.pad` extra elements to allow the computation of the sums (5) and (6), which need elements preceding the first day. So the value of `DI` for that `d` is `DI[self.pad+self.d]`. Notice that all these extra elements of `DI` are zero except for the last one, `DI[self.pad-1]`, which is set to the initial number of infected people.

Coding task 2:

The program `covid19.py` contains only the definition of the function `step`. It calls the function from the parent class wich computes the equations (7), with the terms (5) and (6). The sums stored in `DI` and `Drd` in the program in these last 2 equations are evaluated using a loop each. Rewrite the function `step` in `covid19.py` so that it uses numpy to compute `DI` and `Drd` without using a loop. Notice that `self.Pi` and `self.Pr` are both arrays. You are also expected to improve the comments in the code.

Question 5:

Use the program `run_covid19.py` to find the parameter R and $I(0)$ of the epidemic in the United Kingdom and Brazil. Use the value of R estimated in question 4 for the 2 countries as the initial value of R. For the United Kingdom start with $I(0) = 1$ but for Brazil take $I_0 = 10000$. Adjust manually the value of R until the black curve is parallel to the black stars. Once this is achieved. Adjust the value of $I(0)$ until the black curve overlaps with the data. Then fine-tune R and $I(0)$ until you get a good fit.

The population of the United Kingdom and Brazil are respectively 66 million and 209 million. Use also `dmax = 70` for the UK and `dmax = 60` for Brazil.

Which values of R and $I0$ do you get for each countries? Include the figures of the fit generated by the program for these parameter values.

Question 6:

The profile of fatalities for the United Kingdom and Brazil are very different: on the logarithmic plot it is pretty much a straight line from the onset for the UK while for Brazil it is curving down. Explain where this difference comes from.

4 References

[1] Flaxman *et. al*, *Report 13 – Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 European countries* https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/covid-19/report-13-europe-npi-impact/

[2] Stephen A. Lauer *et. al*, *The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application* Ann. Intern. Med. doi:10.7326/M20-0504

[3] Sung-mok Jung *et. al*, *Real-Time Estimation of the Risk of Death from Novel Coronavirus (COVID-19) Infection: Inference Using Exported Cases* J. Clin. Med. 2020, 9, 523; doi:10.3390/jcm9020523

[4] Fei Zhou *et. al*, *Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study* www.thelancet.com doi: 10.1016/S0140-6736(20)30566-3

5 To submit:

- One pdf file called essay.pdf for the essay. Ensure all your figures have axis labels which are not tiny. Give all your figures a caption describing their content and refer to them in the text by number.
- Python code lin_regression.py
- Python code covid19.py