An introduction to disease modelling

Alongside medics, biologists and public health experts, a key group of people currently advising the government on the Covid-19 pandemic are mathematicians. Mathematical models allow us to explore how an infectious disease will spread through a population and the impact of possible strategies. While none of us here in SoMaS are involved in this frontline work advising the government, a number of us do have significant experience of studying, teaching and researching disease spread in a range of contexts. We therefore thought it might be useful to provide some resources for you to understand more about the basics of disease modelling and to explore for yourselves some of the dynamics that may occur. We hope this can provide you with increased knowledge to understand some of the terms and ideas you will see discussed in mainstream and social media in the coming months.

We should reiterate that these resources are not intended as complete guides for how we should deal with the current pandemic. Nor are the results we're showing here highly accurate representations. Our hope is merely to give you some academic insight in to how mathematical models help us understand disease spread.

This document will talk you through the basics of disease modelling and introduce some python code for you to play around with. The python code is in an accompanying Jupyter notebook. Instructions for those who haven't used this before (or are rusty!) can be found at the end of this document.

The basic disease model

Infectious diseases are largely modelled with a classic framework called the SIR model (which dates from around the early 1920s). In this model we assume that every individual in a population can be classified in to one of three compartments: Susceptible (not infected, no immunity), Infected or Recovered (with immunity).

We would like to know how the number of individuals in each compartment changes over time. We therefore need to consider the transitions between them. Taking the simplest possible case, we have:

- Everyone starts Susceptible (true for an emerging disease like Covid-19);
- Susceptible individuals become Infected by catching the disease off an infected person;
- Infected become Recovered when they clear the infection and afterwards are immune.

From this flowchart we can turn this in to a mathematical model. Since we want to know how each compartment changes with respect to time, ordinary differential equations are an obvious tool for us to use. The following set of ODEs does this:

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

S, *I*, and *R* are the respective numbers of Susceptible, Infected and Recovered individuals. β controls the transmission coefficient of the disease – notice that this requires an interaction between Susceptible and Infected individuals, perhaps an obvious point but an important one. If you are Susceptible and nobody in the population has the infection, there should be no chance of you catching it; the more people who are Infected, the greater the chance you will come in to contact with someone who has it. For this reason the transmission *rate* is actually βI. The recovery rate is γ, and the reciprocal of this is the time spent infected.

We can simplify this even further. Since in our model nobody ever enters or leaves the system, we actually only need two of the ODEs to fully describe the system. We can therefore model everything just using,

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$

Despite being a simple looking system – there are just two variables and two parameters – it is non-linear, and it is therefore not possible to explicitly solve the system (those of you who have taken MAS222 will know about finding equilibria, determining stability and drawing phase portraits for such systems. In fact, you may wish to try and draw some phase portraits of the system yourself!). Of course, the models being used to advise the government are much more detailed than this. For example, models may include:

- When someone first gets infected they are not yet *infectious*. We can model this with an *Exposed* compartment between *S* and *I*, with infection still only occurring from *I* individuals. This is called the SEIR model.
- Include deaths due to the infection. If we do this we can no longer neglect the *dR/dt* equation as the population size will not be constant.
- We don't yet know whether immunity to Covid-19 is long-lasting. We can include this by assuming recovered individuals lose their immunity and return to being susceptible.
- Split the population up in to high-risk and low-risk individuals, and/or age-groups.
- Include realistic contact networks (currently we assume the risk of infection is linked to the total number of infected in the whole population – a mean-field assumption – but realistically you will catch the infection off someone you are actually in contact with).

For the remainder of this worksheet I will stick with the simple SIR model described above. While simplistic, it is the engine that drives many of the results that we still see in more complex models.

So what will happen here? In the long-term, the only possible outcome is that the infection eventually dies out and we are left with some of the population Susceptible and the others Recovered. That is the natural course of most new infections. However, as is currently clear, just because a disease eventually dies out does not mean we do not worry about the damage it will

cause in the interim. Instead, our concern is whether there will be an *epidemic* – a growth in the number of infections.

Assume that when the disease first emerges, like now, the vast majority of the population is Susceptible with just a few infections, such that $S(0) \approx N, I(0) \approx 0$ (where *N* is the total population size). In that case the dynamics of Infected individuals is given by,

$$\frac{dI}{dt} = \beta N I - \gamma I$$

If this is positive, the number of infected individuals grows and we have an epidemic. Since we know *I* is non-negative, this means we need,

$$\begin{aligned} & \beta N - \gamma > 0 \\ \Rightarrow \quad & \frac{\beta N}{\gamma} > 1 \end{aligned}$$

This fraction, $\beta N/\gamma$, has an important real-world definition. It is called the *basic reproductive ratio* and is denoted by R_0. We can interpret this as the (average) number of new infections one infected individual creates at the start of an epidemic. If $R_0 > 1$ then each Infected individual will pass the disease to more than one person, meaning the infection will grow. If $R_0 < 1$, then each Infected individual will pass the disease to (on average) less than one person and it must die out.

This simple definition, which can be estimated for a disease by keeping track of who has infected who, is what gives it its power. The table below gives a few example values of R_0. The estimates for the current outbreak are very preliminary, but it seems to be more infectious than 'flu, but less so than some other diseases. However, it is rare that a new disease emerges to which nobody has previously been exposed such as is happening now.

Disease	$\boldsymbol{R_0}$
Seasonal 'flu	1 - 2
SARS	2 - 5
Smallpox	5 - 7
Measles	12 - 18
Covid-19	2 – 3?

Visualising an epidemic

How does an epidemic play out in the end? As we have seen, provided $R_0 > 1$ it will spread through the population, but we also know it will eventually die out. In the accompanying Jupyter notebook I have provided code for you to explore this. If you have never used Jupyter notebook (or python) before, please see the quick guide at the end of this document.

The first bit of code (under 'Picturing an epidemic') assumes we have 100 individuals in the population and that $R_0 = 2$. As such, each time step every Infected individual makes contact with two randomly-chosen population members. If the contacts are Susceptible, they catch the disease. If not (so they are either currently Infected or Recovered and immune) they stay as they are. Each Infected individual recovers after 1 week. This example has some randomness to it so the plot will

look slightly different every time you run it (try it!), but we always end up with something like a bell-curve. There is initially rapid growth, the infection peaks at about 20 people infected after about 6 weeks and then dies off again, with no infections after about 12 weeks. Notice that a good number of individuals make it through the epidemic having never been infected. Feel free to try playing around with different values of R_0 (it needs to be an integer for this bit of code to work – sorry, I was doing it quickly!) or population sizes.

The second bit of code now uses the ODEs introduced above, with continuous time and populations. The pattern here looks basically the same as the last one (in fact, it should basically average out all of the randomised trials of the discrete model above). Again, we get a classic bell-curve of infection, peaking at around 20 individuals around week 6 and then dying off. Again, feel free to play with the numbers. Look also at the ongoing reproductive ratio plotted in the 2nd figure. This starts at 2 (R_0) but quickly drops off as the size of the Susceptible populations shrinks, and is equal to 1 exactly at the peak of the epidemic.

Vaccination

There is currently a lot of work going on to develop a vaccine against Covid-19. The idea of vaccination is that you are given a low dose of a (part of) an infection, which means your body develops antibodies to it, and afterwards you are immune to the disease. In terms of our model, we can think of vaccination causing individuals to be taken out of the Susceptible compartment and immediately put in to the Recovered compartment, without having ever got infected.

Assume that we give the vaccination near the start of an epidemic. We know that usually the reproductive ratio of the disease is, $R_0 = \beta N/\gamma$. However, if we have vaccinated some proportion, *v*, of the population, then the size of the Susceptible population is no longer *N* but *N(1-v)*. Therefore the reproductive ratio is now $\beta N(1-v)/\gamma = R_0\,(1-v)$. We know that as long as we can reduce this ratio to be less than 1, we can control the disease. We therefore need,

$$R_0(1-v) < 1$$
$$\Rightarrow v > 1 - \frac{1}{R_0}$$

So, for an infection with $R_0 = 2$, vaccinating over 50% of the population should stop it spreading. It is interesting to put this in the context of real diseases using the table of R_0 values above. Smallpox is the one disease that has been globally eliminated with a vaccination programme, and we can see that this required more than 80% of the population to be vaccinated. For measles the figure is more like 93%, but in the late 1990s we had achieved this in the UK and measles was almost eliminated.

It is an important point that we do not need to vaccinate everyone for this to work. There are many people who cannot be vaccinated because they have a compromised immune system. By enough people getting vaccinated we can protect the most vulnerable individuals. This is known as 'herd immunity'.

Some early statements from the UK government suggested that herd immunity should be encouraged to 'naturally emerge' as a strategy for Covid-19. I believe this was unfortunate language to use. The idea was not based on vaccination but on the results seen above whereby the infection dies out naturally because the Susceptible population shrinks, and that a number of the population

would never catch it. The idea seemed to be to arrange things such that it was the large healthy population that contracted the infection and that the vulnerable could self-isolate and end up as the proportion who make it through not being infected. This is dangerous both because Covid-19 still causes many otherwise healthy people to be hospitalised (and so the overwhelmed hospitals will still come to pass) and (as we will see later) unless we completely cut off the high-risk individuals there will still be a significant epidemic in that population.

A simple model for Covid-19

In the next section of code I have tried to choose parameter values that are loosely based on Covid-19 in the UK. To be clear, **this is a very simplistic model and should not be used as an accurate guide to what will happen**. It is more just to give you the chance to play around and get a rough idea about different aspects of the current pandemic. I have assumed a UK population of 65 million, an R_0 of 2.5 and an infection time of 2 weeks.

The red curve shows the number of infected individuals in this case if there was no control. We see this peaking around day 160 (early June) with around 15 million people simultaneously infected. Once more, let's be clear that this is a very rough and loosely-parameterised model, but it is not a million miles from some of the predictions I've seen. If we assume around 20% of these cases require hospitalisation and 5% require an ICU bed (which are the figures guiding a lot of policy), it is clear that this would be an extremely bad scenario, with probably hundreds of thousands of deaths. The blue line shows what happens if we can reduce R_0 by 25% through some simple social distancing measures (the 25% is chosen at random – I have no clear idea of what reduction current measures can realistically produce). We now see a rather smaller epidemic, peaking at around 8 million cases and occurring much later in the year. This is an example of 'flattening the curve'. The key benefit of this is in bringing the numbers down to something close to what we can cope with (though even this would be way beyond NHS capacity).

Temporary control measures

When I assumed 25% reduction in R_0 above, that was for the full year. Many governments and citizens would prefer to avoid long-term measures like this, so what if we instead assume the control measures are time-limited? In the next section of code it is set up to have a 25% reduction for 60 days starting on day 60. This reveals almost no change from no control, just a slight delay. However, if you experiment with changing the start day to sometime later you should find that the 60 day control can lead to a reduction in the epidemic peak to not too far from the year-long control above. Once more, this is a simplistic model, but it shows that well-timed time-limited interventions might be almost as effective as ongoing ones.

In the next cell I perform the same temporary measures but assume they are more extreme, such that we can get a 75% reduction in R_0. This means that for the period of the control measures we would actually have pushed $R_0 < 1$ and the infection will no longer spread. This works brilliantly while the measures are in place – assuming a start day of 120 we see the epidemic never gets to high levels and may well lead to numbers hospitals can cope with. However, as soon as the control measures stop we see a massive epidemic later in the year. This is because hardly anybody has been infected and, in the absence of a vaccine, most of the population remains susceptible (I think this is in part what drove the herd immunity idea discussed above). While buying ourselves time could be important, simply delaying a massive epidemic to later in the year is obviously not much of an

improvement, and would suggest the need for continued control measures later in the year. Again, play around with the start day and control time to see what an optimal strategy might be.

A model with risk-groups

As a first step towards a more realistic model, I have divided the population up in to low-risk and high-risk groups. In fact there is not necessarily any difference in parameter values (transmission, recovery) between these two groups. I simply assume 20% of the population are high-risk and 80% are low-risk. I then assume there are contacts (leading to infection) both within and between these groups. Our interest here is in what happens if we limit these contacts and therefore lower R_0 as above. I assume four cases: no control at all, control only amongst high-risk individuals (so reduced high-high and low-high contacts, but low-low contacts stay the same), control only amongst low-risk individuals and control in everyone.

The red (no control) and black (full control) curves are exactly the same as our first Covid-19 plots above, as should be expected. The green curve shows what happens if only high-risk individuals reduce contacts (many governments have started with such a policy – ask over-70s and those with underlying conditions to self-isolate). Here we see only a small reduction in the epidemic peak. Reducing contacts only in low-risk individuals leads to a much stronger reduction in the peak. This may seem obvious – the low-risk population is the larger one, so of course this leads to a bigger effect. But notice the 2nd plot which shows cases just in the high-risk group. The pattern is exactly the same – we will have far more cases in the high-risk group if it is only those individuals who reduce their contacts than if it is instead the low-risk individuals. This emphasises the importance of everybody reducing their contacts to limit the effects on the most vulnerable.

Thanks!

We hope this has been a simple guide for you to explore the mathematics behind disease models and get the chance to play around with some models. We will hopefully try and keep working on things ourselves and update you with more code. Please feel free to experiment yourself as well! If you have any questions please feel free to get in touch with me (a.best@shef.ac.uk) but please note with the closing of schools I am not as available as usual and it may take me some time to respond. There are also a number of online models available developed by people who are more expert (and have more time!).

A quick guide on how to use Jupyter Notebook

- Jupyter Notebook is a front-end program for running python code. Anyone who has taken MAS212 will be familiar with it. Anyone who has taken MAS115 will know Spyder – the python code is no different, Jupyter is just a different program to use to run the code. It has the advantage that comments and instructions can be written as markdown text, so it works well for a worksheet like this.
- You cannot just double-click on the code file and open it. It has to be opened from within Jupyter Notebook.
- If you haven't already, you will first need to download and install a package of programs called Anaconda from this link: https://www.anaconda.com/distribution/ This contains the actual python code libraries as well as programs like Spyder and Jupyter to run it.

- Once installed, on a Windows machine you should go to START/Anaconda3/Jupyter_notebook. On a Mac you can either look for Navigator in the Applications folder or type ‘jupyter notebook’ into the command line.
- You will first get a screen in your web browser showing a file directory. Navigate to wherever you have saved the python notebook and then click on it to open.
- The notebook provided consists of a series of cells. Some contain explanatory text in markdown language, others contain code.
- To run each cell of code, click inside it and press <Shift>-<Enter>. While it is running that cell the square brackets to the left will show an asterisk. When it has run that will change to a number.
- The first cell must always be run for anything else lower down to work. The rest of the cells are (I hope!) self-contained, but I have assumed you will work through it in order.
- I have produced this fairly quickly so things like legends won’t immediately update if you change parameter values.
- If at any point you get an error, look carefully at the message you are given. Usually the first and last line of the error message are the most instructive for where your error is.
- Sometimes things get so messed up that you need to stop and start the whole thing again. Click on the menu at the top marked ‘Kernel’ and then ‘restart’. You will then need to run all the cells again.
- Any problems, email me at a.best@shef.ac.uk but please have patience about responses.