

EXTENDED COURSEWORK: COMPUTING FOR PHYSICAL SCIENCE

Hand in date: Wednesday 27th January 2021, 3pm

1. Complete the Quiz under the ‘Coursework and Exam Practice’ section of the module Moodle page: <https://moodle.nottingham.ac.uk/mod/quiz/view.php?id=4599452>
There is no time limit for the quiz, but you are only allowed to **submit once**. {25}

2. 2020 was a tough year. The COVID-19 pandemic took the world by surprise and led to major changes in our daily lives. Despite the huge economic and social impact of the pandemic, there remains a surprising level of debate on the seriousness of the disease, with many people believing it is not so much worse than flu and arguing government interventions are unjustifiably harsh. The present question is motivated by this debate.

Let us consider the following situation: a politician argues that COVID-19 is the most severe public health emergency of our times, currently killing more people than all other death causes combined, while another politician claims that the disease is in fact quite low in the list of leading causes of death in the population. Is there a sense in which both statements could be correct? In other words, could both of these statements be based on different (mis)interpretations of the same data? This question invites you to use some of the available data on COVID-19 in the UK, your modelling skills and the power of Python in order to understand what the data are saying and also to study the basic dynamics of the disease. You will be guided through this step by step. After answering the question you will have put your basic Python skills into practice to model something useful, and in doing so you will have also refined your own understanding of the seriousness of this pandemic.

Note that this question is (intentionally) somewhat more open-ended in nature than our usual Computing coursework questions. However, we have allocated marks for the specific subquestions below as a guide, so you know where you stand. For full marks, you should submit a Jupyter Notebook rather than a Python script. After (or even before) you have submitted your code for assessment, you should feel free to use your code as the basis for further exploration. For example, you could download transmission rate data and use them to factor in the effect of government interventions/lockdowns in your model, or extend the model to include additional effects like birth and death in the population, or even use your model as a starting point to implement the (published) full models used by the research groups advising the government.

- (a) Create a Jupyter Notebook and use Markdown to introduce a document title (e.g. Computing Extended Coursework) and some descriptive text. Then introduce a code cell to import NumPy as `np` and PyPlot as `plt`. {3}
- (b) Recall the two apparently contradicting claims above: one politician said COVID-19 is killing more people than all other death causes combined, while the other claimed that the disease is quite low in the list of leading causes. To understand whether any of them (or both!) could in some sense be right, let’s have a look at the data. These are available to download from the gov.uk website: <https://coronavirus.data.gov.uk/details/download>. To work directly with the data you would need to know how to read and write files in Python, which we will cover later in the module (section 11 of the Python Notes). However, there is no need to do that for this question. We have re-organised the published daily data into weekly data and we supply them as arrays in file `data.py`.
 - i. Open file `data.py` (you can do so in CoCalc, or using any text editor if you are working off-line) and copy the first data list, which corresponds to weekly deaths (with a positive covid test within 28 days) from early March until just before Christmas. Use this data list in your Jupyter Notebook to create a NumPy array for these weekly data. Then plot the data against week number, where the first data point corresponds to week 1, the second to week 2, etc. {3}

- ii. The plot would look much nicer if it was displayed in bar form, where each week on the x-axis has a bar with a height corresponding to the weekly data. Fortunately, matplotlib's pyplot has a function `bar()` for making such bar plots. Look up this function online and use it to make a new plot, displaying the data as a bar plot (against week number as before). This `plt.bar()` function has a parameter 'width', which you should use to set the width of your bars to 0.7. Add a plot title and appropriate labels on the x and y axis of your new (bar) plot. **{3}**
- iii. Let us now revisit the two politicians' claims. One thing we can check is how the data compare to average mortality rates in the UK. You can consult the gov.uk website, but a reasonable estimate for the annual number of deaths in the UK is 550,000 corresponding to roughly 10,500 deaths per week. We can also compare our COVID-19 data to typical numbers of deaths from other leading causes, like influenza. This varies significantly from year to year, but a ballpark figure for influenza is 20,000 annual deaths, which is about 400 deaths per week¹. Plot these two numbers (i.e. for weekly deaths from all causes and from influenza), or your own estimates for these two numbers, on your bar plot as horizontal lines with a different linestyle and colour **{2}**.
- iv. You should observe that on the first peak of the pandemic, the weekly deaths are comparable to, but significantly lower than, the average weekly deaths from all causes in a typical year. Thus, the first politician may have exaggerated somewhat, but the essence of their statement is correct. More precisely, near the (first) peak of the pandemic, COVID-19 was responsible for about half as many weekly deaths as all other causes combined. In fact, if we work with daily data, the average number of daily deaths from all causes is about 1500, which is about 3/2 the daily deaths at the first peak, and it is also where we are (fast!) heading in the second peak, at the time this coursework question is being written. (This depends sensitively on when you are doing the coursework: by now we are probably well past this limit – it is a rapidly changing world we find ourselves in!)

What about the second politician's claim? By inspecting your plot, focusing on the horizontal line for influenza, you should conclude that there are two ways in which their claim could be 'correct' in some sense, and in both cases the claim is highly misleading. The first way is if they made the statement sometime between weeks 18 and 29 (when a combination of government interventions and high temperatures ensured relatively low numbers of cases) and only compared daily/weekly/monthly figures, ignoring the first peak. The second way is if they made their statement early in the pandemic, near the first peak, but counted total number of deaths from the beginning of the pandemic until that moment, comparing that to the total (annual!) number of deaths from other leading causes – a highly misleading comparison.

Using array slicing and the array operator `np.sum()`, confirm that the total number of COVID-19 deaths between week 1 and week 6 are smaller than 20,000. Print that value in integer format, along with some relevant text. **{2}**

The general lesson here is that we must be careful when comparing numbers like this; we must look not only at the current values, but also their rate of change, and their integral over time, appropriately normalised.

- (c) Let us now examine the number of infections and try to understand how it grows in time. The initial phase of an epidemic is characterised by exponential growth, which is why early intervention is crucial for controlling the spread of the virus. Do the data show exponential growth? Let's have a look.
 - i. Revisit file `data.py` and copy the second data list, which contains the cumulative sum of confirmed covid infections (number of positive tests) week by week over the

¹This is a rather rough estimate, as there are also large seasonal variations in influenza with many more cases in the winter than in the summer, but it is sufficient for our purposes.

same period, i.e. from early March until just before Christmas. Create a NumPy array from this data list, and plot it against week number (again with the first data point corresponding to week 1, the second to week 2, etc) as a bar plot. **{2}**

- ii. According to a (very crude!) model, the number of infections during the early evolution of an epidemic grows exponentially according to the formula:

$$I(t) \simeq I_0 \exp [(R_0 - 1)\gamma(t - t_0)] \quad (1)$$

where $I_0 = I(t_0)$ is the number of infectious individuals at time t_0 . The quantity R_0 is the basic reproduction number we've been reading about in the news, describing the average number of secondary infections produced by a single infected person. The other important quantity here is γ , which (roughly) describes the average time over which an infected individual remains infectious (γ is the inverse of that time). Typical numbers we have been reading about recently for R_0 are in the range 1.1 – 1.3. Also, based on a 2-week-long self-isolation advice we can take $\gamma \simeq 1/2$ in our units.

Compute an array with predicted values of $I(t)$ from week 28 to week 38, using equation (1) and assuming $R_0 = 1.3$ and $\gamma = 1/2$. You may take I_0 to be equal to the number of cases on week 28, as given by the data in (c)(i) above. Plot this model prediction as a line on your bar plot of cumulative covid infections. You should find that the model closely follows the height of the bars on those weeks. **{3}**

- (d) From your plot in (c)(ii) above you can see that the data are clearly showing exponential growth in the second peak over many weeks in a row and government intervention is necessary. (National lockdown restrictions were just announced as this coursework question was being finalised.) While it is encouraging that with the above reasonable choice of parameters, equation (1) appears to be describing the data quite well, you should not take this model too seriously². In particular, exponential growth cannot be indefinitely sustained even without government intervention: once a significant fraction of the population becomes immune after having been infected by the virus, the infection growth falls and a steady state is reached. This effect is not captured by equation (1), so we need a better model.

The basic dynamical features of the evolution of an epidemic are described by the so-called SIR model. Apart from the number of *infectious individuals* $I(t)$ the model also evolves the number of *susceptible individuals* $S(t)$ (i.e. those who are not immune and can therefore be infected through contact with an infectious individual), and the number of *removed individuals*, $R(t)$. This last quantity describes individuals who have previously been infected and have now either recovered (and assumed to be immune so they are now counted in $R(t)$ rather than $I(t)$) or deceased. Clearly the sum of $S(t)$, $I(t)$ and $R(t)$ is equal to the total population, and ignoring birth and death we have:

$$S(t) + I(t) + R(t) = N, \quad (2)$$

where N is the *total number of individuals* in the population, which is a *constant*. In this approximation, the dynamics of the epidemic is fully described by the following set of differential equations:

$$\frac{dI}{dt} = R_0\gamma\frac{IS}{N} - \gamma I \quad (3)$$

$$\frac{dS}{dt} = -R_0\gamma\frac{IS}{N} \quad (4)$$

$$\frac{dR}{dt} = \gamma I \quad (5)$$

²In fact, we are not even comparing the right quantities...

Notice that the sum of these three equations has its right-hand side equal to zero, guaranteeing that the quantity $I(t) + S(t) + R(t)$ remains constant, in agreement with equation (2). In practice, this means that we don't need to solve all three differential equations (3-5); we may solve only two of them and use equation (2) to express one of the three variables in terms of the other two.

- i. Go back to file `data.py` and copy the code that appears at the bottom of that file. This is a simple implementation of the SIR model described above. Take a few minutes to inspect the code and understand what it is doing, comparing to equations (2) and (3-5). It models a population of 70 million (similar to the UK) over 80 weeks, starting with 100 infectious individuals. Copy this code in a new cell in your Jupyter Notebook and run it for $R_0 = 1.6$ and $\gamma = 1/2$. Then plot $I(t)$, $S(t)$ and $R(t)$ as solid lines of different colour (on the same plot) against week number, starting with week 1 and finishing with week 80. **{2}**

From your plot, it should become apparent that even for such a large value of $R_0 = 1.6$ (so with very limited or no government intervention) herd immunity guarantees that the exponential decay eventually stops and a steady stage is reached. However, this requires a large fraction ($\sim 60\%$) of the population to be infected. Thus, according to the model, without government intervention (which would have the effect of reducing R_0) a large portion of the population would be infected, leading to hundreds of thousands of deaths.

- ii. In the code above, you will have noticed that we evolved the system by solving only two of the differential equations, (3) and (4), also using equation (2). This works because we are working in the approximation that the population is constant (i.e. we probe the system on short timescales over which birth and death in the population can be neglected). You may want to extend your code in the future to move away from that simplifying assumption. To anticipate this, let us modify the code so as to solve all three differential equations (3-5).

Copy the above code from (d)(i), paste it in a new cell and modify it so that $R(t)$ is computed from the differential equation (5) instead of equation (2). Also copy your plotting code from (d)(i) and use it to check that the plot of $I(t)$, $S(t)$ and $R(t)$ from your modified code is identical to the plot in (d)(i). **{3}**

- iii. A University of 30,000 students and staff members quotes 1,500 covid cases, but after taking drastic measures to prevent the spread of the virus records a rapid drop of cases, down to a total of around 100 after 5 weeks. Use your code from (d)(ii) to model this and ask yourself how low R_0 must be for such a rapid drop to occur. (If your attempt at (d)(ii) was unsuccessful, you can use the code from (d)(i) instead.) You may assume that the University is isolated so that the total population can be taken to be $N = 30,000$ with $I(0) = 1,500$. You only need to evolve the system over just 5 time steps (corresponding to 5 weeks). **{2}**

You will find that even for a very low $R_0 \lesssim 0.5$ our model cannot produce such a rapid reduction in the number of infections. However, you should be able to explain the data with a reasonable value $R_0 \lesssim 1$ if you allow for a significant number of recovered individuals in the beginning, and if you take into account that a large portion of infectious individuals may have chosen not to test themselves.

That was the end of the question. Before you submit (or after, if you are running out of time!) it is worth spending some time to think about how you could expand this work. For example, what is the effect of government interventions (limiting social contacts) and how would you model it? What other factors are relevant to include? How much more complicated from your model are the models used by the various research groups who are advising the government?

Upload to Moodle your .ipynb (Jupyter Notebook) file.