**DISCLAIMER**

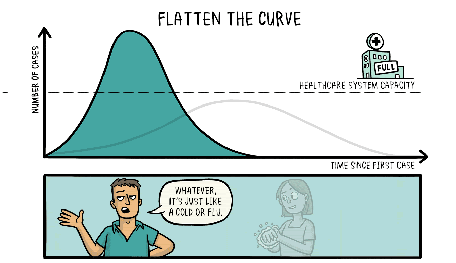
The simulation results in this blog post, or any other results produced by the R code described in it, should **not** be used as actual estimates of mortality or any other aspect of the COVID-19 pandemic. The simulations are intended to explain principles, and permit exploration of the potential effects of various combinations and timings of interventions on spread. Furthermore, the code for these simulations has been written hurriedly over just a few days, it has not yet been peer-reviewed, and is considered *alpha* quality, and the simulation parameterisation presented here has not yet been validated against real-world COVID-19 data.

**Introduction**

We looked at the use of some R packages developed by the [**R** **E**pidemics **Con**sortium (RECON)](https://www.repidemicsconsortium.org/) to undertake epidemiological analyses COVID-19 incidence data scraped from various web sources.

Undertaking such value-adding analyses of COVID-19 incidence data, as the full horror of the pandemic unfolds, is a worthwhile endeavour. But it would also be useful to be able to gain a better understanding of the likely effects of various public health interventions on COVID-19 spread.

[“Flattening-the-curve”](https://www.fastcompany.com/90476143/the-story-behind-flatten-the-curve-the-defining-chart-of-the-coronavirus) infographics such as the one shown below are now everywhere. They are a useful and succinct way of communicating a key public health message – that social distancing and related measures will help take the strain off our health care systems in the coming months.



Source: [Siouxsie Wiles and Toby Morris](https://thespinoff.co.nz/society/09-03-2020/the-three-phases-of-covid-19-and-how-we-can-make-it-manageable/)

However, as pointed out by several commentators, many of these infographics miss a crucial point: that public health measures can do more than just **flatten** the curve, they can also **shrink** it, thus reducing the total number of cases (and thus serious cases) of COVID-19 in a population, rather than just spread the same number of cases over a longer period such that the area under the curve remains the same.

This crucial point was beautifully illustrated using R in a [post by Michael Höhle](https://staff.math.su.se/hoehle/blog/2020/03/16/flatteningthecurve.html), which is highly recommended reading. Michael used a dynamic model of disease transmission, which is based on solving a system of ordinary differential equations (ODEs) with the tools found in base R.

Such mathematical approaches to disease outbreak simulation are elegant, and efficient to compute, but they can become unwieldy as the complexity of the model grows. An alternative is to use a more computational approach. In this post, we will briefly look at the individual contact model (ICM) simulation capability implemented in the excellent [EpiModel](http://www.epimodel.org/) package by Samuel Jenness and colleagues at the Emory University Rollins School of Public Health, and some extensions to it. Note also that this post is based on [two more detailed posts](https://timchurches.github.io/blog/) that provide more technical details and access to relevant source code.

**The EpiModel package**

The EpiModel package provides facilities to explore three types of disease transmission model (or simulations): dynamic contact models (DCMs) as used by [Michael Höhle](https://staff.math.su.se/hoehle/blog/2020/03/16/flatteningthecurve.html), stochastic individual contact models (ICMs) and stochastic network models. The last are particularly interesting, as they can accurately model disease transmission with shifting social contact networks – they were developed to model HIV transmission, but have been used to model transmission of other diseases, including ebola, and even the propagation of *memes* in social media networks. These network models potentially have application to COVID-19 modelling – they could be used to model shifting household, workplace or school and wider community interactions, and thus opportunity for transmission of the virus. However, the networks models as currently implemented are not quite suitable for such purposes, modifying them is complex, and they are also very computationally intensive to run. For these reasons, we will focus on the simpler ICM simulation facilities provided by EpiModel.

Interested readers should consult the extensive tutorials and other documentation for EpiModel for a fuller treatment, but in a nutshell, an EpiModel ICM simulation starts with a hypothetical population of individuals who are permitted to be in one of several groups, or *compartments*, at any particular time. Out-of-the-box, EpiModel supports several types of models, including the popular SIR model which uses **S**usceptible, **I**nfectious and **R**ecovered compartments. At each time step of the simulation, individuals randomly encounter and are exposed to other individuals in the population. The intensity of this population mixing is controlled by an *act rate* parameter, with each “act” representing an opportunity for disease transmission, or at least those “acts” between susceptible individuals and infectious individuals. Recovered individuals are no longer infectious and are assumed to be immune from further re-infection, so we are not interested in their interactions with others, nor are we interested in interactions between pairs of susceptible individuals, only the interactions between susceptible and infectious individuals. However, not every such opportunity for disease transmission will result in actual disease transmission. The probability of transmission at each interaction is controlled by an *infection probability* parameter.

It is easy to see that decreasing the act.rate parameter is equivalent to increasing social distancing in the population, and decreasing the inf.prob parameter is equates to increased practice of hygiene measures such as hand washing, use of hand sanitisers, not touching one’s face, and mask wearing by the infectious. This was what is explored in some detail.

**Extensions to EpiModel**

However, the SIR model type is a bit too simplistic if we want to use the model to explore the potential effect of various public health measures on COVID-19 spread. Fortunately, EpiModel provides a plug-in architecture that allows more elaborate models to be implemented. The full details of my recent extensions to EpiModel, but the gist of it is that several new compartment types were added, as shown in the table below, with support for transition between them as shown in the diagram below the table. The dashed lines indicate infection interactions.

| **Compartment** | **Functional definition** |
| --- | --- |
| S | Susceptible individuals |
| E | Exposed **and** infected, not yet symptomatic but potentially infectious |
| I | Infected, symptomatic **and** infectious |
| Q | Infectious, but (self-)quarantined |
| H | Requiring hospitalisation (would normally be hospitalised if capacity available) |
| R | Recovered, immune from further infection |
| F | Case fatality (death due to COVID-19, not other causes) |

Another capability that has been added is the ability to specify time-variant parameters, as a vector of the same length as there are time steps in the simulation. This allows us to smoothly (or step-wise) introduce, and withdraw, various interventions at arbitrary times during the course of our simulation.

We won’t cover here the details of how to obtain these extensions, which at the time of writing should still be considered *alpha* quality code

**Baseline simulation**

First we’ll run a baseline simulation for a hypothetical population of 10,000 people, in which there are just three infectious COVID-19 cases at the outset. We’ll run it for 365 days, and we’ll set a very low rate at which infectious individuals enter self-quarantine (thereby dramatically lowering their rate of interactions with others) after they become symptomatic (or have been tested and found positive), and thus aware of their infectivity. Because it is stochastic, the simulation is run eight times, using parallel processing if available, and the results averaged.

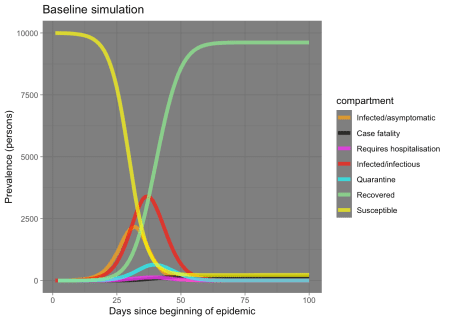
tic()

baseline\_sim <- simulate(ncores = 4)

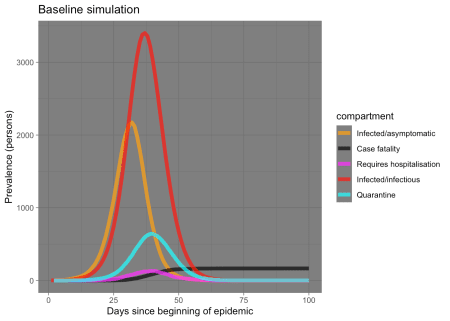
toc()

## 58.092 sec elapsed

Let’s visualise the results as a set of time-series of the daily count of our 10,000 individuals in each compartment.



OK, that looks very reasonable. Note that almost the entire population ends up being infected. However, the **S** and **R** compartments dominate the plot (which is good, because it means humanity will survive!), so let’s re-plot leaving out those compartments so we can see a bit more detail.



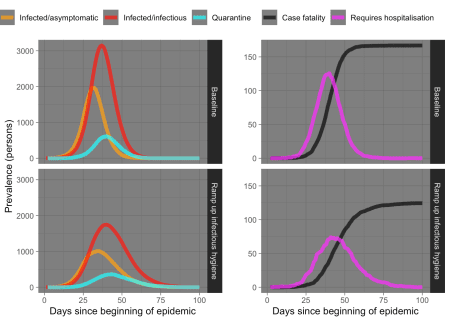
Notice that the **I** compartment curve lags behind the **E** compartment curve – the lag is the incubation period, and that the **Q** curve lags still further as infected people only reluctantly and belatedly quarantine themselves (in this baseline scenario).

**Running intervention experiments**

Now we are in a position to run an experiment, by altering some parameters of our baseline model.

Let’s model the effect of decreasing the infection probability at each exposure event by smoothly decreasing the inf.prob parameters for the **I** compartment. The infection probability at each exposure event (for the **I** compartment individuals) starts at 5%, and we’ll reduce it to 2% between days 15 and 30. This models the effect of symptomatic infected people adopting better hygiene practices such as wearing masks, coughing into their elbows, using hand sanitisers, not shaking hands and so on, perhaps in response to a concerted public health advertising campaign by the government.

Let’s examine the results of experiment 1, alongside the baseline for comparison:



We can see from the plots on the left that by encouraging hygiene measures in symptomatic infectious individuals, we have not only substantially “flattened the curve”, but we have actually shrunk it. The result, as shown in the plots on the right, is that demand for hospital beds is substantially reduced, and only briefly exceeds our defined hospital capacity of 40 bed. This results in a substantially reduced mortality rate, shown by the black line.

**More experiments**

We can now embark on a series of experiments, exploring various interventions singly, or in combination, and with different timings.

**Experiment 2**

Let’s repeat experiment 1, but let’s delay the start of the hygiene campaign until day 30 and make it less intense so it takes until day 60 to achieve the desired increase in hygiene in the symptomatic infected.

infectious\_hygiene\_delayed\_ramp <- function(t) {

ifelse(t < 30, 0.05, ifelse(t <= 60, 0.05 - (t - 30) \* (0.05 -

0.02)/30, 0.02))

}

infectious\_hygiene\_delayed\_ramp\_sim <- simulate(inf.prob.i = infectious\_hygiene\_delayed\_ramp(1:366))

**Experiment 3**

Let’s repeat experiment 1, except this time instead of promoting hygiene measures in the symptomatic infected, we’ll promote, starting at day 15, prompt self-quarantine by anyone who is infected as soon as they become symptomatic. By “prompt”, we mean most such people will self-quarantine themselves immediately, but with an exponentially declining tail of such people taking longer to enter quarantine, with a few never complying. Those in self-quarantine won’t or can’t achieve complete social isolation, so we have set the act.rate parameter for the quarantined compartment to a quarter of that for the other compartments to simulate such a reduction in social mixing (an increase in social distancing) in that group.

quarantine\_ramp <- function(t) {

ifelse(t < 15, 0.0333, ifelse(t <= 30, 0.0333 + (t - 15) \*

(0.3333 - 0.0333)/15, 0.333))

}

quarantine\_ramp\_sim <- simulate(quar.rate = quarantine\_ramp(1:366))

**Experiment 4**

Let’s add a moderate increase in social distancing for everyone (halving the act.rate), again ramping it down between days 15 and 30.

social\_distance\_ramp <- function(t) {

ifelse(t < 15, 10, ifelse(t <= 30, 10 - (t - 15) \* (10 -

5)/15, 5))

}

soc\_dist\_ramp\_sim <- simulate(act.rate.i = social\_distance\_ramp(1:366),

act.rate.e = social\_distance\_ramp(1:366))

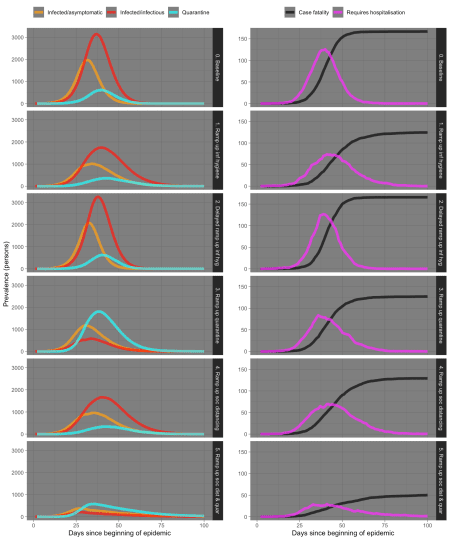
**Experiment 5**

Let’s combine experiments 3 and 4: we’ll add a moderate increase in social distancing for everyone, as well as prompt self-quarantining in the symptomatic.

quar\_soc\_dist\_ramp\_sim <- simulate(quar.rate = quarantine\_ramp(1:366),

act.rate.i = social\_distance\_ramp(1:366), act.rate.e = social\_distance\_ramp(1:366))

Now let’s examine the results.



**Discussion**

The results of our experiments almost speak for themselves, but a few things are worth highlighting:

* Implementing interventions too late is almost worthless. Act early and decisively. You can always wind back the intervention later, whereas a failure to act early enough can never be recovered from.
* Prompt self-quarantining of symptomatic cases is effective. In practice that means everyone with COVID-19-like symptoms, whether they actually have COVID-19 or something else, should immediately self-quarantine. Don’t wait to be tested.
* A moderate increase in social distancing (decrease in social mixing) in everyone is also effective, mainly because it reduces exposure opportunities with both the asymptomatic-but-infected and the symptomatic infected.
* Combining measures is even more effective, as can be seen in experiment 5. In fact, there are theoretical reasons to believe that the effect of combined measures is partially multiplicative, not just additive.
* Public health interventions don’t just flatten the curve, they shrink it, and the result is very substantially reduced mortality due to COVID-19.

None of these insights are novel, but it is nice to be able to independently confirm the recommendations of various expert groups, such as the WHO Collaborating Centre for Infectious Disease Modelling at Imperial College London (ICL) who have recently released a [report on the impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand](https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/news--wuhan-coronavirus/) which recommends similar strategies to those we have just discovered from our modest simulations in R.

**Two more experiments**

What happens if we dramatically increase social distancing through a two week lock-down, which is then relaxed? We’ll use a step function to model this. We test such a lock-down lasting from day 15 to 30, and separately a lock-down from day 30 to day 45 instead. We’ll model the lock-down by reducing the act.rate parameters for all compartments from 10 to 2.5.

twoweek\_lockdown\_day15\_vector <- c(rep(10, 15), rep(2.5, 15),

rep(10, 336))

twoweek\_lockdown\_day30\_vector <- c(rep(10, 30), rep(2.5, 15),

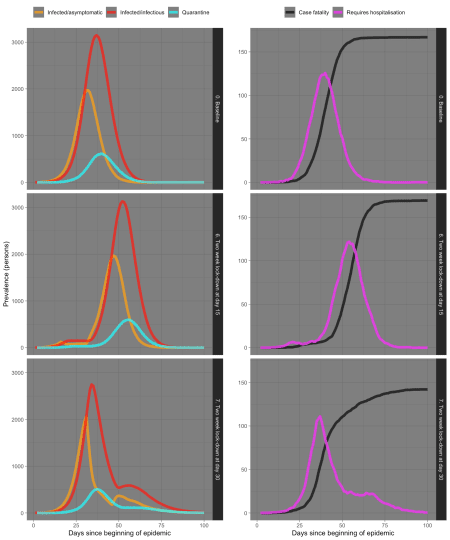
rep(10, 321))

twoweek\_lockdown\_day15\_sim <- simulate(act.rate.i = twoweek\_lockdown\_day15\_vector,

act.rate.e = twoweek\_lockdown\_day15\_vector)

twoweek\_lockdown\_day30\_sim <- simulate(act.rate.i = twoweek\_lockdown\_day30\_vector,

act.rate.e = twoweek\_lockdown\_day30\_vector)



Wow, that’s a bit surprising! The two week lock-down starting at day 15 isn’t effective at all – it just stops the spread in its tracks for two weeks, and then it just resumes again.  
But a two-week lock-down starting at day 30 is somewhat more effective, presumably because there are more infected people being taken out of circulation from day 30 onwards. But the epidemic still partially bounces back after the two weeks are over.

What this tells us is that single lock-downs for only two weeks aren’t effective. What about a lock-down for a whole month, instead, combined with prompt quarantine with even more effective isolation and hygiene measures for those quarantined?

fourweek\_lockdown\_day15\_vector <- c(rep(10, 15), rep(2.5, 30),

rep(7.5, 321))

fourweek\_lockdown\_day30\_vector <- c(rep(10, 30), rep(2.5, 30),

rep(7.5, 306))

fourweek\_lockdown\_day15\_sim <- simulate(act.rate.i = fourweek\_lockdown\_day15\_vector,

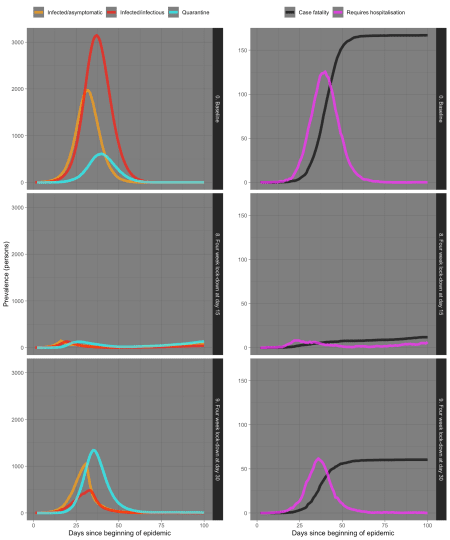
act.rate.e = fourweek\_lockdown\_day15\_vector, quar.rate = quarantine\_ramp(1:366),

inf.prob.q = 0.01)

fourweek\_lockdown\_day30\_sim <- simulate(act.rate.i = fourweek\_lockdown\_day30\_vector,

act.rate.e = fourweek\_lockdown\_day30\_vector, quar.rate = quarantine\_ramp(1:366),

inf.prob.q = 0.01)



Well, that’s satisfying! By acting early, and decisively, we’ve managed to stop COVID-19 dead in it tracks in experiment 8, and in doing so have saved many lives – at least, we have in our little simulated world. But even experiment 9 provides a much better outcome, indicating that decisive action, even if somewhat belated, is much better than none.

Of course, the real world is far more complex and messier, and COVID-19 may not behave in exactly the same way in real-life, but at least we can see the principles of public health interventions in action in our simulation, and perhaps better understand or want to question what is being done, or not done, or being done too late, to contain the spread of the virus in the real world.

**Conclusion**

Although there is still a lot of work yet to be done on the extensions to EpiModel demonstrated here, it seems that they offer promise as a tool for understanding real-world action and inaction on COVID-19, and prompting legitimate questions about such actions or lack thereof.

One would hope that governments are using far more sophisticated simulation models than the one we have described here, which was built over the course of just a few days, to plan or inform their responses to COVID-19. If not, they ought to be.