Bachelor Dissertation

Kasper Engelen, Jonathan Meyer, Dawid Miroyan, and Igor Schittekat

University of Antwerp

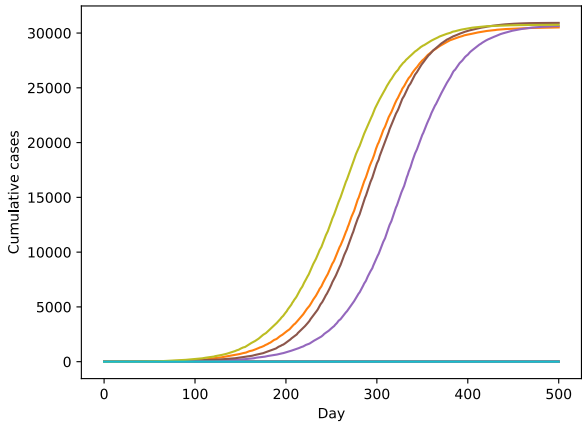
Abstract. This document reports our findings regarding the final dissertation. The sections and subsections correspond to the assignments given to us. In this project we worked with a simulator called Stride, developed at the University of Antwerp. We explore various concepts within computational epidemiology through the use of this program.

Keywords: Computational Epidemiology · Dissertation

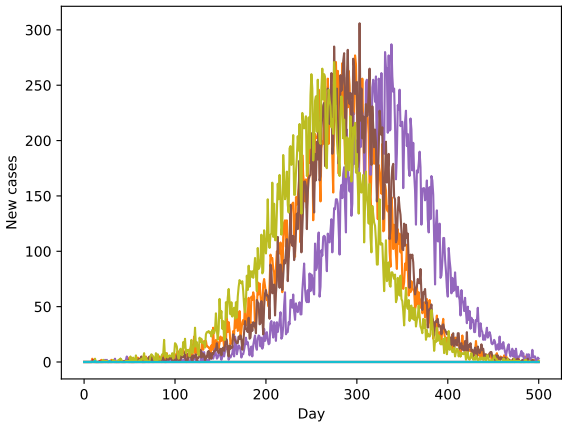
1 Simulation

1.1 Stochastic Variation

The first topic we consider is stochastic variation. Since the simulation uses a pseudo-random number generator, it's useful to inspect the influence of this stochasticity on the results of the simulation. Using the Stan tool which is provided with Stride, we collected data on 100 simulations with an identical configuration file. The only difference between executions is the RNG seed. When plotting the results, it becomes apparent that the amount of new cases per day follows a normal distribution. For the data we collected, we calculated a mean of 33.33 new cases per day, and a variance of 1196.99. Looking at the plot for the cumulative cases per day, two 'categories' can be distinguished: 'outbreak' scenarios and 'extinction' scenarios. These cases are quite evenly spread, which explains the high variance for new cases per day. In the former, the curve has a sigmoid shape, indicating that the disease succesfully spread among the population. The latter scenario corresponds to the curves that are almost constant and are bounded by a value well under the population size. In these cases, the disease did not manage to spread, resulting in only a few infected people at the end of the simulation. This is likely explained by the initial infectation: if the first person to be infected is sufficiently isolated (either socially or by being surrounded by people who are immune), the disease doesn't have a chance to spread.
Figure 1 shows the different scenarios. In the cumulative cases plot the 'extinction' senario is clearly visible at the bottom. The plot for new cases per day is very jagged, which can also be explained by the stochastic nature of the simulation.



(a) Cumulative cases per day.



(b) New cases per day.

Fig. 1: Plots using different seeds

1.2 Extinction Threshold

As discussed previously, it might be the case that only very few people become infected over the course of a simulation. This is referred to as extinction. There is a clear distinction between outbreaks and extinctions, so in this subsection we attempt to find an *extinction threshold*.

Figure 2 gives the frequencies of the amount of infected people at the end of the simulations. The distinction between outbreaks and extinctions is quite clear

from this histogram. There is one peak on the lower end of the x-axis, and there is a cluster on the higher-end. It is hard to find a good concrete value for the threshold, but the total amount of infected people in an extinction is always significantly under the total population size. Therefore, a fraction of the total population could be used as treshold. In our simulations the amount of cases after extinction never exceeded 50, so a threshold of 0.01% of the population seems reasonable.

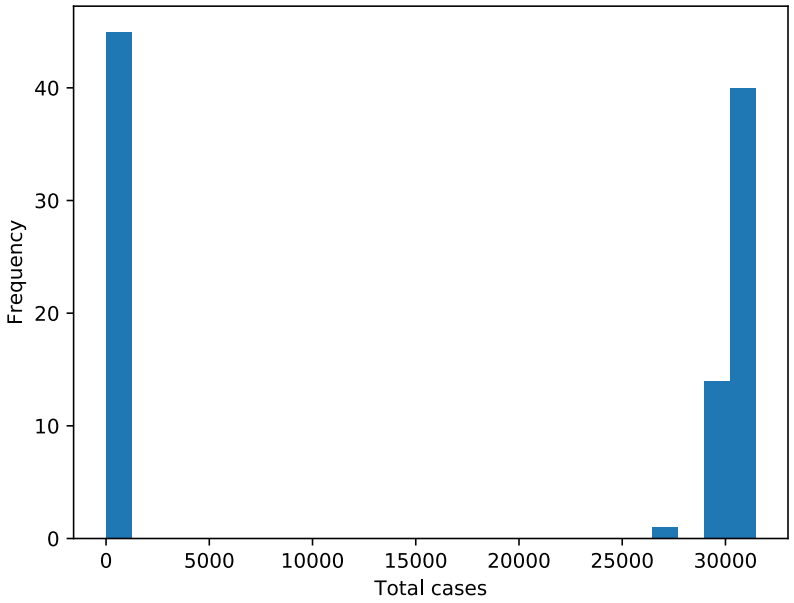


Fig. 2: Histogram showing the total amount of cases.

1.3 Immunity Level

In order to make assumptions about the population's immunity level, we look at the simulated results for different values. Upon experimentation, it becomes apparent the immunity level is approximately 70% of the population. Figure 3 shows the average new cases per day for different immunity levels. The reference curve is also included. This plot was generated by taking the average of new cases per day over 20 simulations per immunity level. Using PyStride to simulate outbreaks, we narrowed down the immunity level I to $0.705 \leq I \leq 0.7175$.

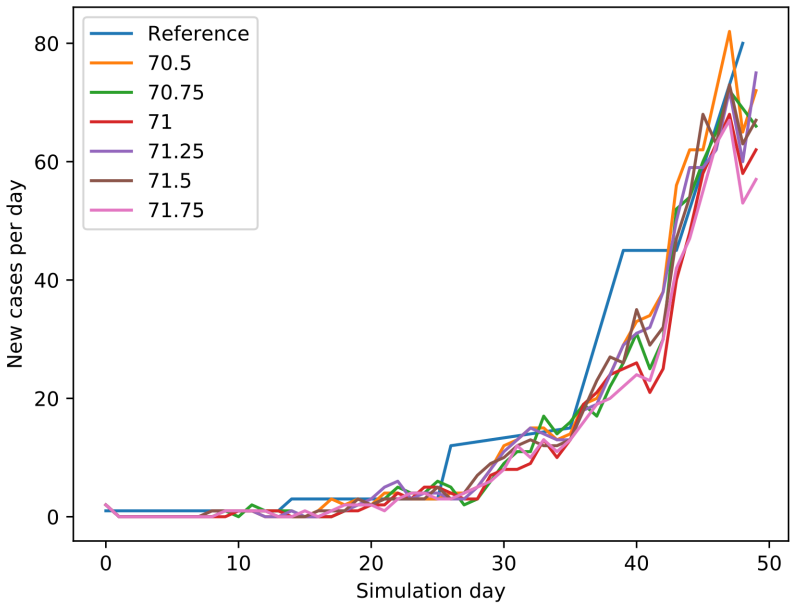


Fig. 3: New cases per day for different immunity levels.

1.4 Estimating R_0

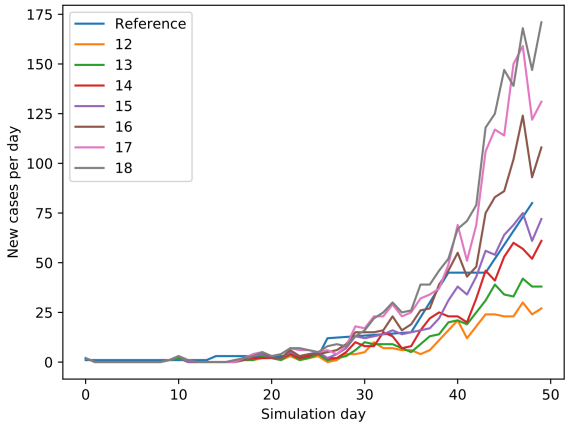
Now that we have a relatively decent estimate for the immunity level of the population, we can do an analogous exercise to estimate R_0. For this data, we fixed the immunity level to 70.5%. Figure 4 (a) shows the plot for each possible R_0 in the range $[12, 18]$ (average over 20 simulations). It's clear that the best candidates are 14, 15 or potentially 16. When we plot a more detailed plot for just those valued (average over 30 different simulations), it seems 15 is the best estimate for R_0.

2 Population generation

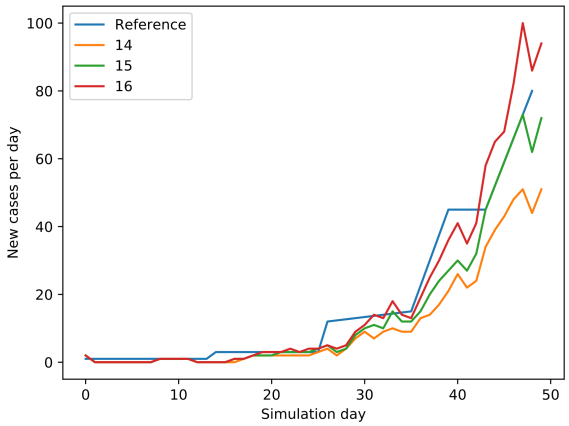
2.1 Influence of demography

2.2 Vaccinating on campus

By vaccinating the students, quickly after infected individuals appear in the population, we want to test whether so called 'catch-up' campaigns have a noticeable impact on an outbreak.
For these simulations, we generated a population where 60% of people between the age of 18 and 26 attend high education. We then simulate scenarios where all



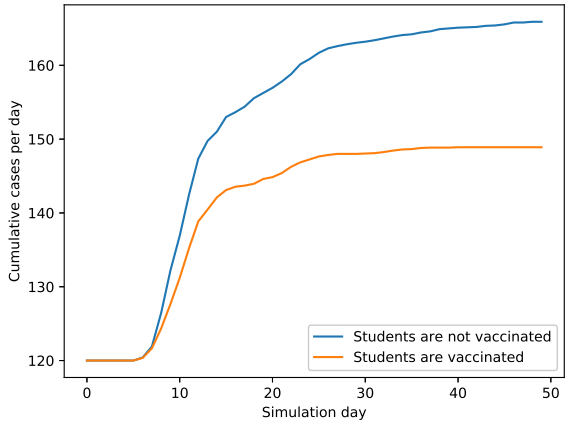
(a) Overview of all potential R_0's.



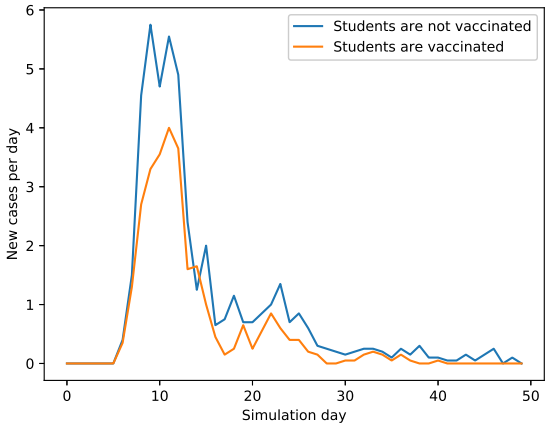
(b) More detailed look at R_0 candidates.

Fig. 4: Plots for various values for R_0

students are either vaccinated on day 7 or not vaccinated at all. When looking at Figure 5, the amount of new cases is the same in both scenarios on day 5 and 6. As the students are not vaccinated in both scenarios, some of them become infected and can spread the disease. Starting from day 7, where in one of the scenarios the susceptible students are vaccinated, the amount of new cases is lower than when not vaccinating.



(a) Cumulative cases per day



(b) New cases per day.

Fig. 5: Plots showing the impact of vaccinating students

2.3 Commuting to work

Another factor that is interesting to examine is the impact of commuting on disease spread.

3 Performance profiling

Using GProf, we will look at the performance in different scenarios. By varying parameters, we try to see which parts of the code they have an influence on, and which parts take up the most time.
For the first 4 parameters

- amount of days
- population size
- immunity rate
- seeding rate

the actual sorting and analyzing of the population takes up most of the time.

3.1 Number of days

By increasing the number of days to be simulated, the total execution time gets longer as well. This should be expected as more days means more times simulating what goes on in a day.

Number of days	Time
50	00:00:04
100	00:00:09
150	00:00:14
200	00:00:18
500	00:00:38

Table 1: Number of days

3.2 Population size

Generating a new population depends on the given size, which was expected. The generation however is very fast.

Population size	Time
10000	00:00:00:421
50000	00:00:00:816
100000	00:00:01:332
200000	00:00:02:249
500000	00:00:04:926

Table 2: Population size

3.3 Immunity rate

Varying the immunity rate does not seem to affect the total execution time.

Immunity rate	Time
0.2	00:00:05:817
0.4	00:00:05:750
0.6	00:00:05:643
0.8	00:00:05:580
0.99	00:00:05:648

Table 3: Immunity rate

3.4 Seeding rate

A bigger seeding rate slightly increases the time of the execution. This happens as more people have to be initially infected.

Seeding rate	Time
0.02	00:00:06:599
0.002	00:00:05:589
0.0002	00:00:05:134
0.00002	00:00:04:865

Table 4: seeding rate

3.5 Contact log mode

The mode of the contact log has a very large impact on the execution time. Logging every contact between people takes a long time.
At day 50 in the simulation, only 20000 people out of 600000 where infected. When logging the susceptible people, you actually log almost 580000 people at each day which is very close to logging all people. This is very fast when the mode is set to Transmissions as you would only log once for each newly infected person.

Contact log mode	Time
All	00:26:48:251
Transmissions	00:00:06:730
Susceptibles	00:27:06:444
None	00:00:05:824

Table 5: Contact log mode