Simulation Paper

Benjamin Vandersmissen[1], Lars Van Roy[2],
Evelien Daems[3], and Frank Jan Fekkes[4]

[1] benjamin.vandersmissen@student.uantwerpen.be
[2] lars.vanroy@student.uantwerpen.be
[3] evelien.daems@student.uantwerpen.be
[4] franciscus.fekkes@student.uantwerpen.be

Abstract. In this paper we will examine the Stride tool and discover its functionalities. We will discuss some findings about the use of different parameters, populations and more. In the end there is a brief discussion of the performance of the program, a very important topic within computer related problems.

1 Simulation

1.1 Stochastic variation

We use the Stan (STochatsic ANalysis) controller to examine the influence of stochasticity on the results obtained from the simulation.

In Figure 1 the number of cummulative cases per time-step is plotted. Here we can observe an exponential grow of the number of cases througout time. This is not surprisingly because it can be deducted from common reasoning. If per time more people are affected, a larger contactpool is possibly infected. These people who are now new carriers of the disease will enter their personal contactpool and again more people will be reached.

Towards the end a flattening of the curve occurs. This is not something totally unexpected because the population is obviously not infinite. At one point anyone who can be infected will effectively become a carrier of the disease.

The same reasoning can explain the curve in Figure 2. Now the cases are not the cummulative ones but the number of new cases in each time-step. A similar course of the curve can be observed.

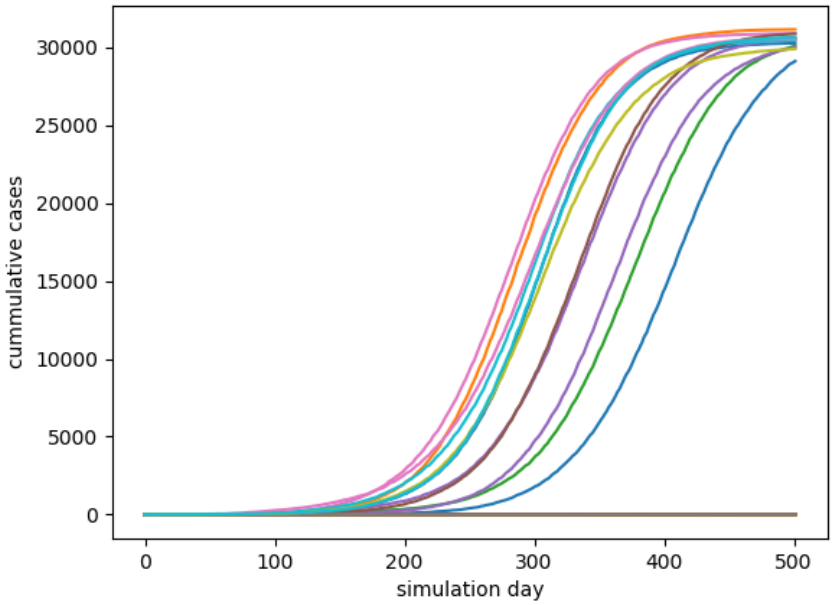

Fig. 1. Result of a number of stochastic runs. The figure displays the distribution of the number of cummulative cases per time-step.

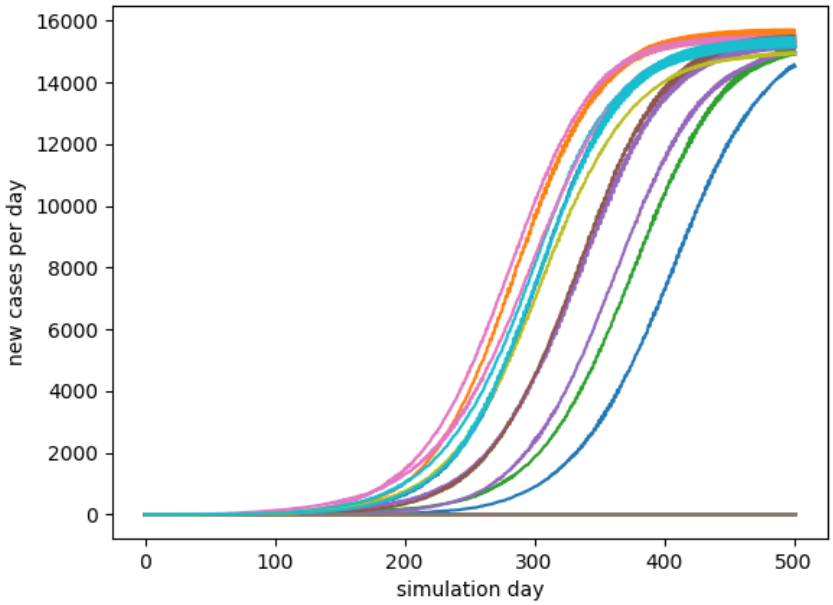

Fig. 2. Result of a number of stochastic runs. The figure displays the distribution of the number of new cases per time-step.

1.2 Determining an extinction threshold

Looking at figure 1 it is clear that in some cases an outbreak doesn't occur. These cases can be called extinctions of the outbreak or disease. Determining if a simulation is an extinction or an outbreak is neccessary to be able to divide the two possible outcomes. If we can find some threshold where there is a clear difference between large outbreaks and extinctions we can seperate the two scenarios.

After running fifty simulations we can plot the total number of infected cases and their frequency. We used the file "stochastic_analysis.xml" for the simulations. After running the simulator the outcome is plotted in the histogram in Figure 3 where the frequency of the amount of infected cases is plotted.

There is a clear distinction between large outbreaks and smaller ones. The smaller ones are again plotted in the second histogram "extinction_small.jpg". There it can be noticed that small outbreaks are really small (20 maximum). Which can be called an extinction afterIt is neccessary to be able to exclusively look at outbreaks. 500 days. The threshold for this example can be set between than 50 and 25 000. Either of those thresholds should eliminate all extinctions in this case.

A very low threshold might allow some extinctions to be passed while a high threshold might eliminate an outbreak. What can be noticed is the total lack of simulations between 100 and 25000 infected cases. But there can still be exceptions in the infected cases. A threshold of 1000 would be more than adequate. It will eliminate all extinctions while keeping the outbreaks.

It should be noted that this threshold will change for a lot of variables. For example: time and population will affect the threshold. A new threshold should be determined for each set of simulations.

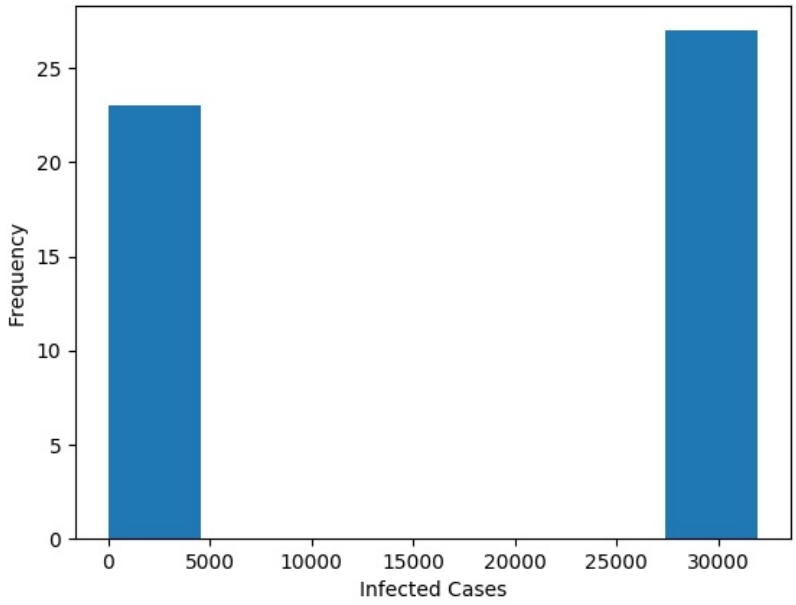

Fig. 3. Above is an histogram with the frequency of total infected cases for 50 simulations. See fig 4 for an enlargement of the left spike.

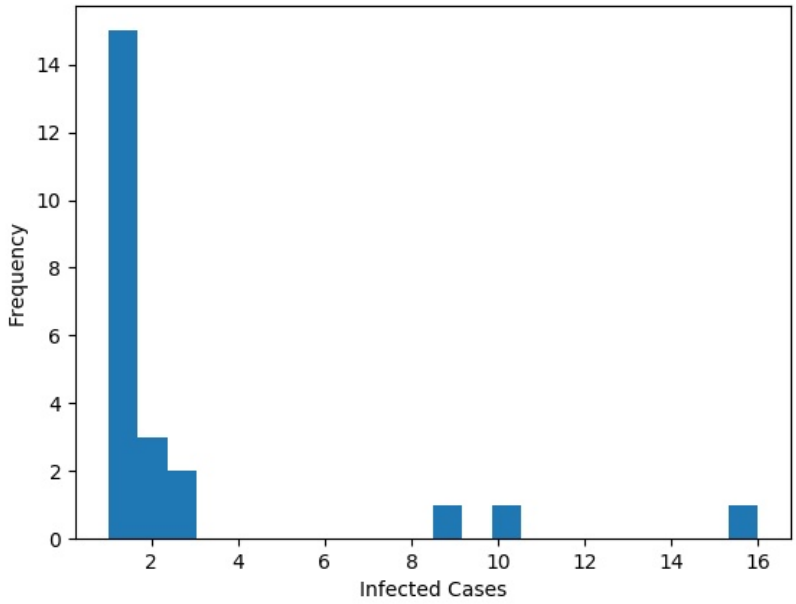

Fig. 4. A zoom on the left spike of figure 3. Instead of 0 to 5000 it is actually 0 to 16.

1.3 Estimating the immunity level

For this assignment we had to estimate the percentage of people who were immune to the disease given the following graph.

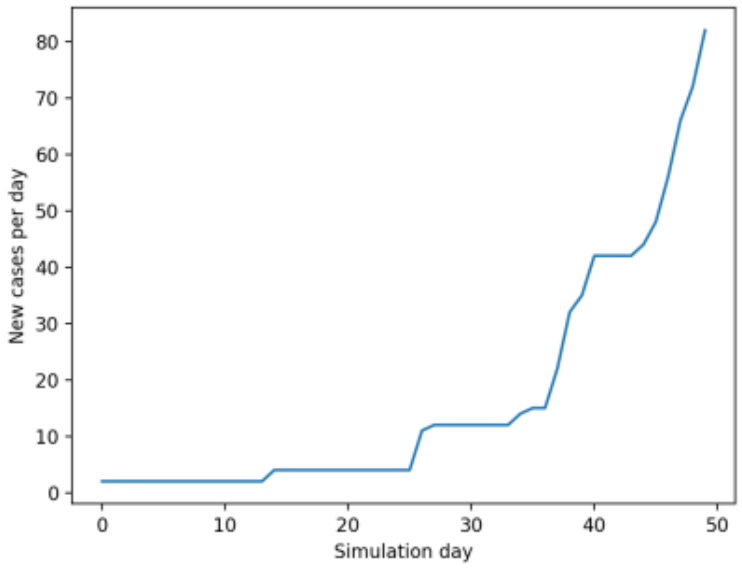

Fig. 5. New cases observed per day during the outbreak

As a first approximation we performed 10 simulations with immunity levels ranging from 0 to 90% as seen in the next graph. As we can clearly see, all immunity levels lower then 50% are unrealistic, compared to the desired graph. As a next step we decided to drop off the unrealistic immunity levels and generated a zoom of the realistic immunity levels with the same offset.

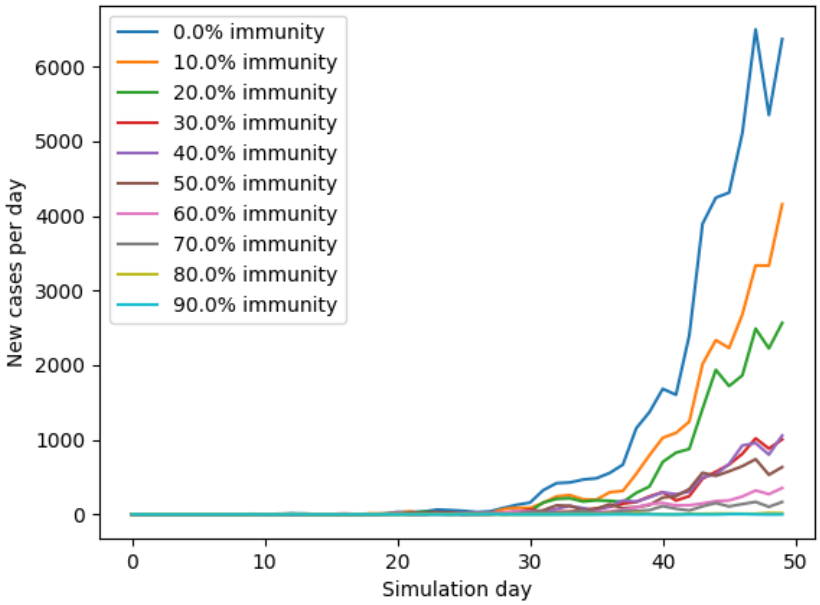

Fig. 6. first estimate of outbreaks

By dropping all percentages lower than 60 and higher than 90% we get the following graph, this graph is a lot closer to the desired graph (since the highest number of new cases is now only 110 compared to over 6000), but is still far from accurate. We can now see that the desired immunity rate should lie somewhere between 70 and 80% as 70 is too high and 80 is too low.

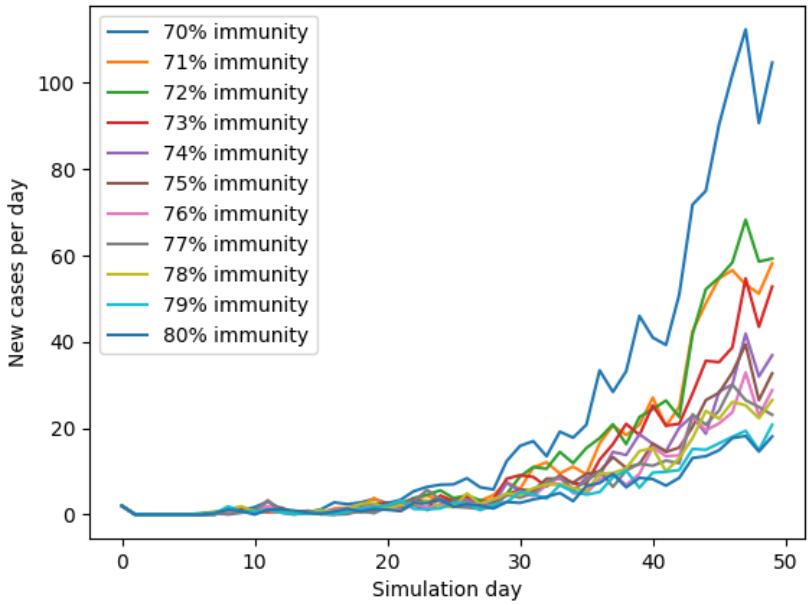

Fig. 7. narrowed down estimate of outbreaks based on 10 simulations

Finally, we zoom in between 70 and 75% and we can see that the immunity percentage is somewhere around the 70% mark. Curiously enough, with an immunity rate of 71% and an average of 25 simulations, there are more infected than with an immunity rate of 70%. The only problem with this 70 % graph is that there is a sudden drop in infections around day 45 while the original graph doesn't feature that drop.

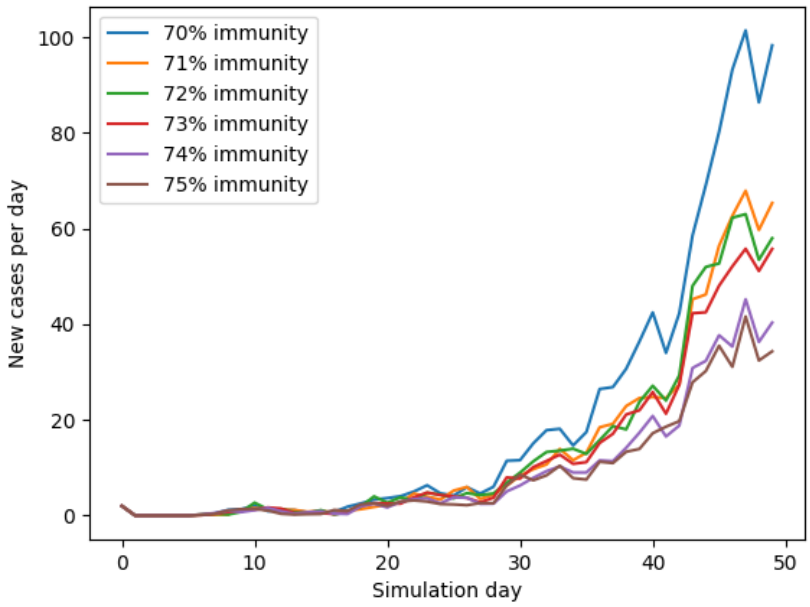

Fig. 8. final estimate of outbreaks based on an average of 25 simulations

1.4 Estimating R_0

Here, we need to approximate the same graph but now he have to use an extra parameter, R_0, the basic reproduction number of a disease, indicating how many people a single infected person will infect on average.

We will use the same technique we used in the previous assignment, only this time, we need to repeat it for R0 = 12 .. 18.

R0	estimated immunity rate
12	65 %
13	68
14	69
15	70
16	73
17	74
18	76

Table 1. Estimated Immunity Rates based on 25 simulations

Generally speaking, the immunity rate is dependent of the parameter R0, which is to be expected. However, the parameter R0 doesn't seem to matter that much because the immunity rates are still pretty close to eachother.

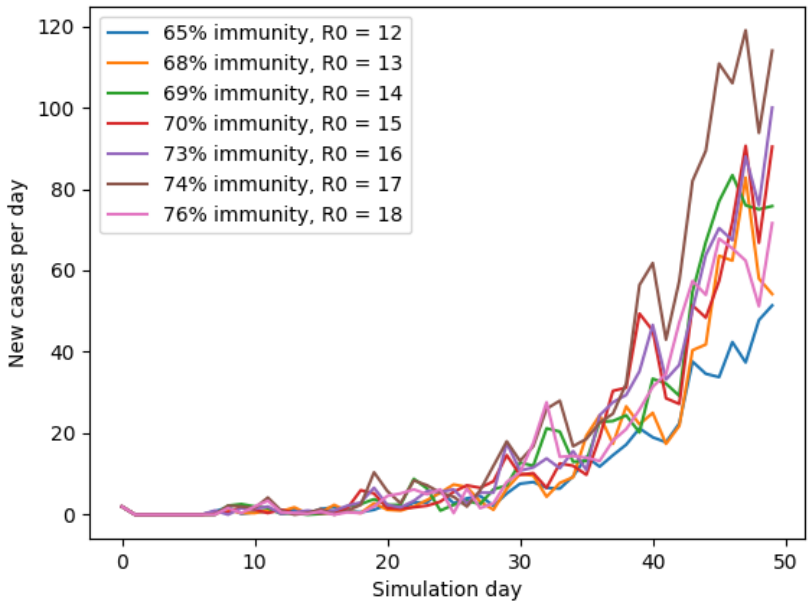

Fig. 9. 25 simulations for found immunity values

2 Population generation

2.1 Investigating the influence of demography on epidemics

We generate populations for two regions with different age distributions. We need to find which region is more likely to have an outbreak, region A, which has more younger people or region B, which has more elderly people. One might assume that the older region will have more chance because of the lower immunity the elderly might have.

In case of population A there is around an 80% chance for an outbreak to occur, in the simulations with population B there is 73% chance for an outbreak. Population B is the older population in respect to population A, this means that an older population does not mean an increase in the likelihood of an outbreak, but a decrease. This could be explained by the fact that older people are commuting less.

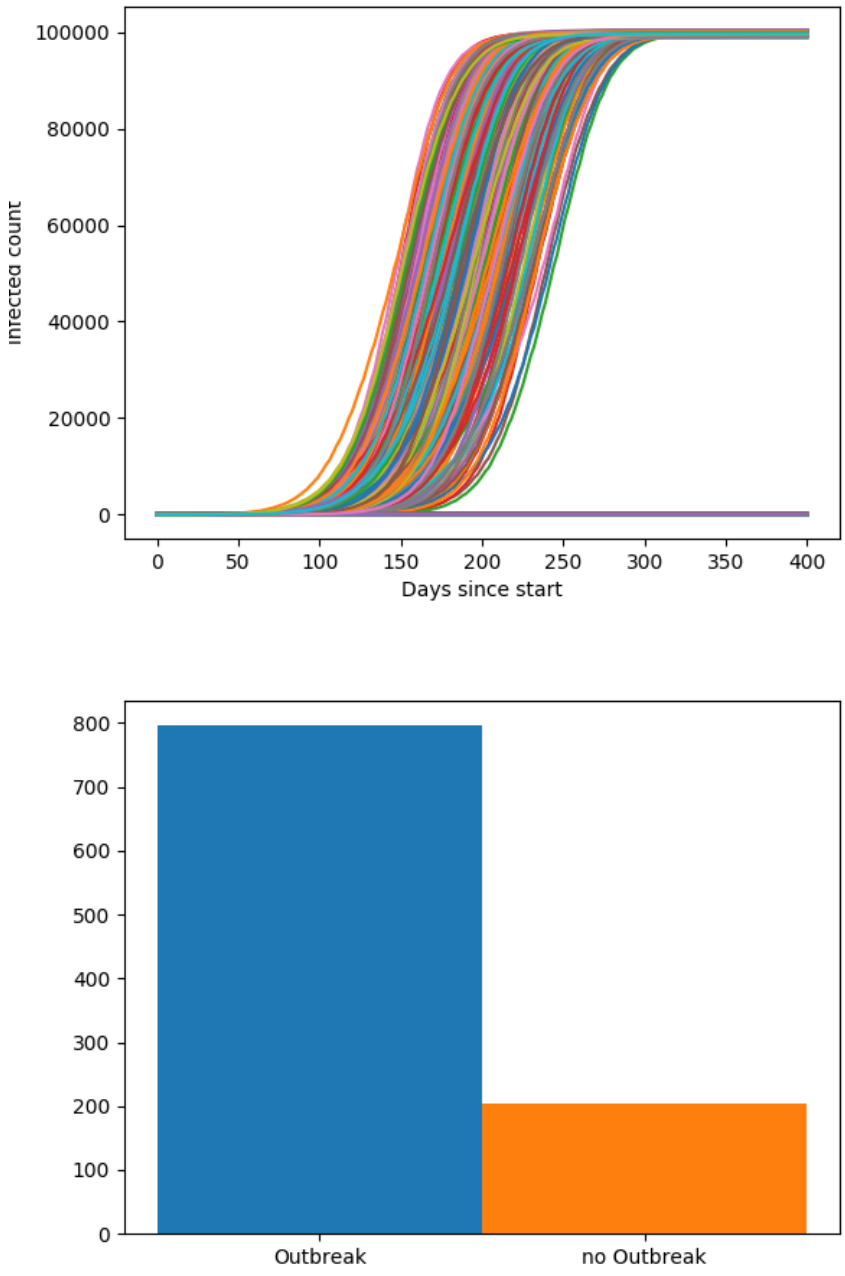

Fig. 10. Results of 1000 simulations with population A

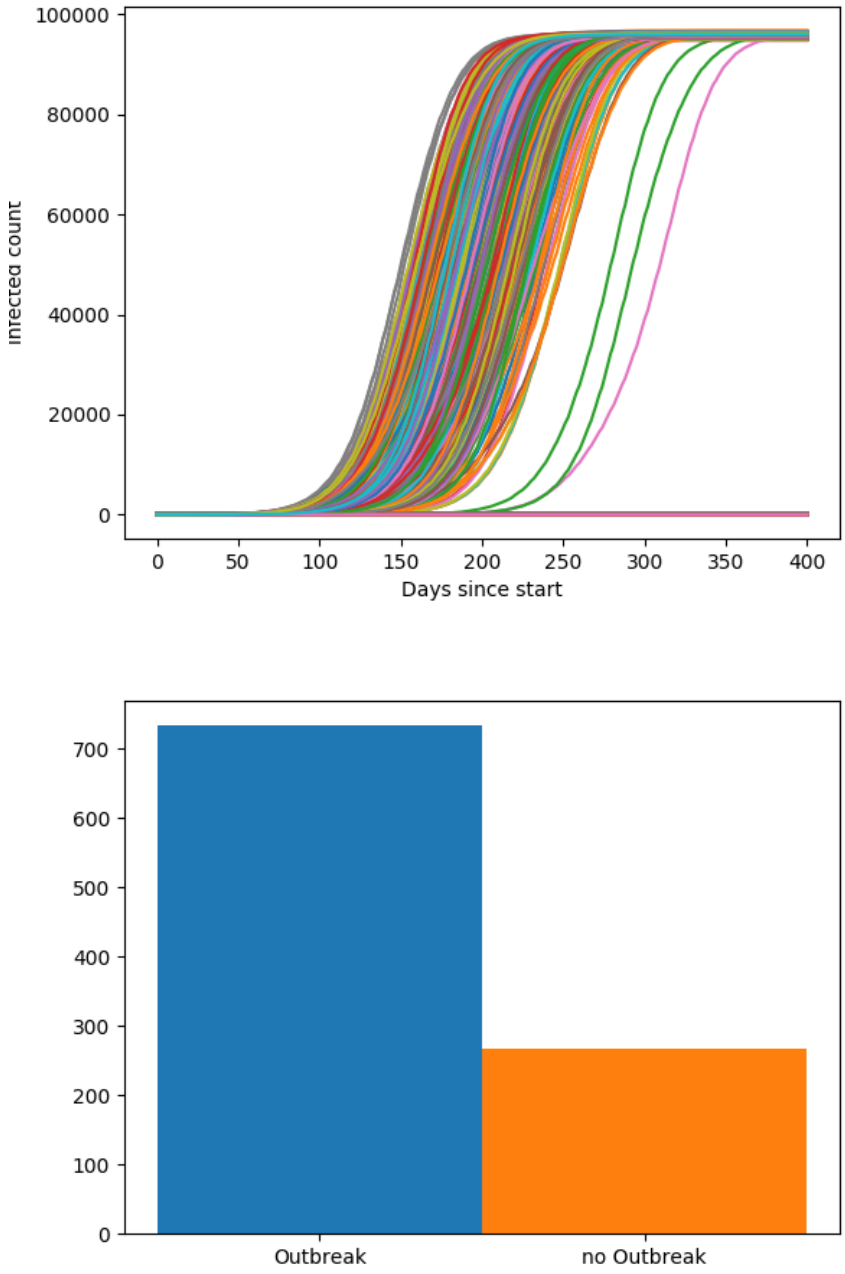

Fig. 11. Results of 1000 simulations with population B

2.2 Vaccinating on campus

Vaccinations are often given at a young age. But not everyone gets vaccinated, there might be particular groups of people who are more susceptible. Here it is simulated what would happen if student were such a group. A student will be defined as a person between the age of 18 and 26 who are in the pool "College". A population will be generated where the age group of 18 to 26 will have a lower immunity. To test if vaccinating the students during an outbreak will aid in the surpression of it, two scenario's will be tested. One where the students will be vaccinated a week after an infected individual is introduced and one where the students aren't vaccinated.
A total of 100 simulations have been run. 50 Without vaccinations and 50 with vaccinations given seven days after the first infected individuals are introduced. A total of 1200 people will be infected to start the outbreak. The cumulative infected cases have been tracked from day to day and averaged over all simulations. The results are shown on fig 12. The results are averaged because there wasn't much deviation between the runs but a single run isn't a good representation.

It is clear that even vaccinating during an outbreak helps in surpressing said outbreak. With only around 3500 students needing a vaccination in a student-body of around 60000 and a total population of ten times that, the cost saving is quite hight and the amount of infected is reduced.
In conclusion, vaccinating students who still need their vaccinations, is worth it during an outbreak in this scenario.

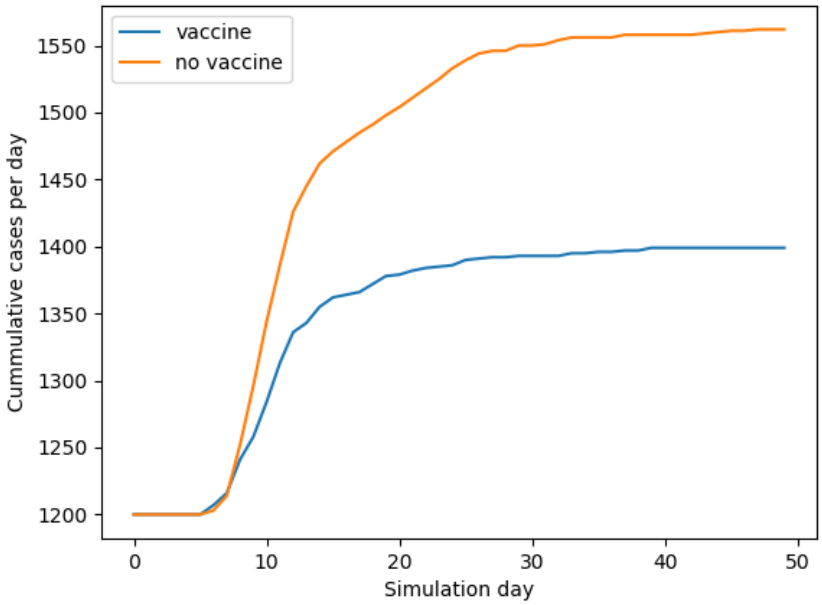

Fig. 12. Above is a plot of average cumulative cases of 50 simulation each seen day by day. People between the age of 18 and 26 have a lowered immunity against the disease. During the blue scenario people in college are vaccinated after seven days. They are not in the orange scenario.

2.3 Is commuting to work important for disease spread?

One could easily assume that working at a different location affects the rate at which a disease spreads, as it enhances it's reach. To get an idea of the effect of the parameter we did a simulation for 11 different percentages, ranging from 0 to 100. The displayed data is an average derived from 50 different runs for each percentage.
As we can clearly see, there is little to no difference between the different commuting percentages and, even though the graph is kinda clouded by the amount of different graphs drawn on top of each other. We can clearly see that the fraction of workplace commuters is negligible. It might still have a slight effect when we were to zoom in, but it is soo small that it can not be considered to be a determining factor for the spread of diseases.

3 Performance profiling of sequential code

To study the performance of the code we will discuss a few parameters. We used the GPROF tool to profile the code. Based on these result we could see the influence of different parameters on the runtime.
First we will choose a random number of days to run a simulation and look at the time needed to complete the algorithm. We considered in total 19 different amount of days in an interval of 10 to 2000 days.

As could be expected, there is an increase in execution time when we take a larger amount of days. The number of days determines the number of loops in a simulation, hence this logically affects the needed time for a simulation by quite a big margin. This parameter will cause the largest change in perfomance in comparison with the other ones. Especially the sorting of the members en getting the total amount of infected people took up most of the computing time.

Table 2. Relation between the number of days and the runtime

Number of days	Time needed
10	00:00:03:946:037
50	00:00:06:885:956
100	00:00:10:964:261
500	00:00:36:137:411
1000	00:01:07:248:430
2000	00:02:06:124:493

Next, we vary the parameter of population size. From the following table it is clear that the larger the population the longer the simulation needs to finish. From the GPROF analysis we notice that the most work is done in getting the count of the infected and soritng the members. It can be said that the size of the population delivers the most influence on the total time. One can argue that the number of days had the most impact but it is not completely realistc to study outbreaks over a number of years what takes up the most computation time. A larger population is more common than a large number of days.
In the previous part we considered the impact of the number of days. And as could be expected a large number of days combined with a large population will result in a large execution time.

Table 3. Relation between population size and simulation runtime with 50 days

Population size	Time needed
1000	00:00:00:429:476
50000	00:00:00:620:702
100000	00:00:00:954:298
500000	00:00:03:904:541
1000000	00:00:07:272:251

When considering the time needed to generate a geo-based population and write it to a file without performing a simulation, the tendency of an increasing time with a larger population can be observed. The most computingpower went to writing the contact pools for the population.

Table 4. Relation between population size and generating runtime with 50 days

Population size	Time needed
1000	00:00:00:262:863
10000	00:00:00:308:607
100000	00:00:00:336:115
500000	00:00:01:130:983
1000000	00:00:02:386:551

When varying the immunity rate, there is no significant difference in runtime for different configurations. In order for this variable to have an influence on the final result, it is necessary to give other parameters different values. As mentioned earlier, most of the time is used to sort and analyze the population, a factor like immunity rate has no effect on this process. In the table below you can see the runtimes for different immunity rates and a period of 50 days. When performing this ananlysis with a higher number of days of course this took some more time. But changing the immunity itself has no effect.

Table 5. Relation between immunity rate and runtime for 50 days

Immunity rate	Time needed
0.2	00:00:04:869:367
0.4	00:00:04:873:811
0.6	00:00:04:966:409
0.8	00:00:05:035:361
0.99	00:00:04:921:399

Seeding rate has a slight impact, but this impact is minimal. Seeding rate has no effect on the computation needed to sort and analyze the population, which is the major faction in a simulation. The following observations were made with a number of days equal to 50. A larger amount

Table 6. Relation between the seeding rate and the runtime

Seeding rate	Time needed
0.000001	00:00:06:210:017
0.00001	00:00:06:188:449
0.0001	00:00:06:294:636
0.001	00:00:06:659:908
0.01	00:00:07:437:776
0.1	00:00:07:406:320

The contact log mode has a significant impact on the running time of a simulation. When the standard algorithm is used (all or susceptibles) it requires a lot more time to complete the simulation. It forms a large contrast with the running time needed when using the optimized algorithm with all the members of the contact pool sorted. By reducing the number of loops in the algorithms, the necessary time to complete the algorithm can be reduced along with it.

Table 7. Relation between the contact log mode and time needed to perform a simulation with 50 days.

Contact log mode	Time needed
All	00:18:38:106:723
Susceptibles	00:18:20:352:080
None	00:00:06:620:969
Transmissions	00:00:06:793:748

3.1 Conclusion

In general we can conclude that there are three parameters that have quite an impact on the running time of a simulation: the size of the population, the amount of days and the algorithm that was used to run the simulation. The larger the size and the number of days, the longer it takes to complete one simulation. Thus in combination with various combinations of the other parameters the runtime can differ a lot.