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.
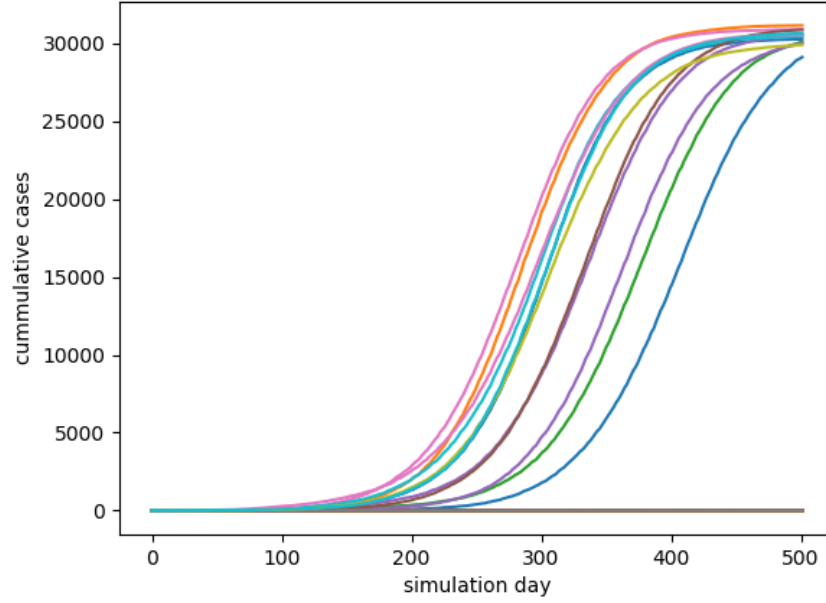
Fig. 1. Result of a number of stochastic runs. The figure displays the distribution of the number of cummulative cases per time-step.
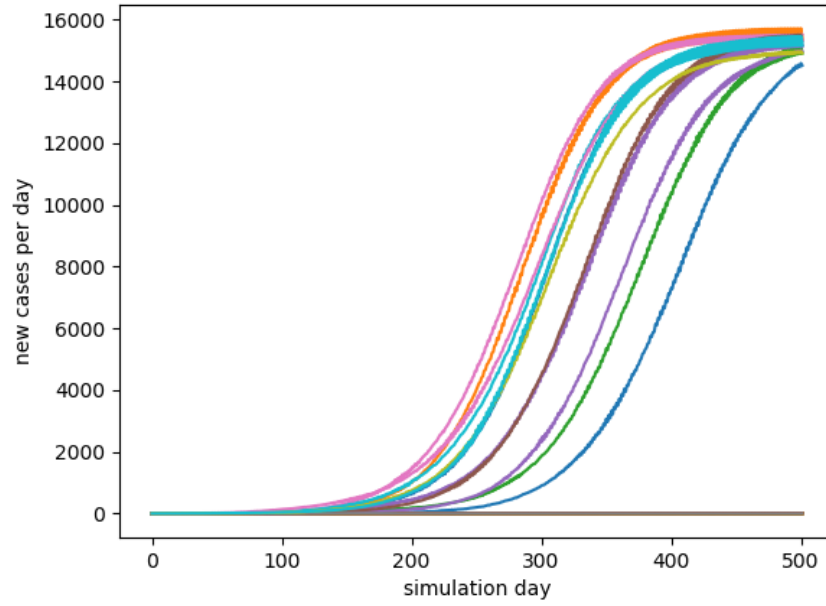
Fig. 2. Result of a number of stochastic runs. The figure displays the distribution of the number of new cases per time-step.

1.2   Determining an extinction threshold

In the previous section 1.1 (Stochastic variation) we found that extinction will influence the outcome. It is neccessary to be able to exclusively look at outbreaks. If we can find some threshold where there is a clear difference between large outbreaks and extinctions we can seperate the two scenarios.

After creating a number of simulations we can plot the total number of infected cases and their occurences. 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 (25 maximum). Which can be called an extinction after 500 days. The threshold 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 al lot of variables. Variables like time and population will affect the threshold. A new threshold should be determined for each simulation.
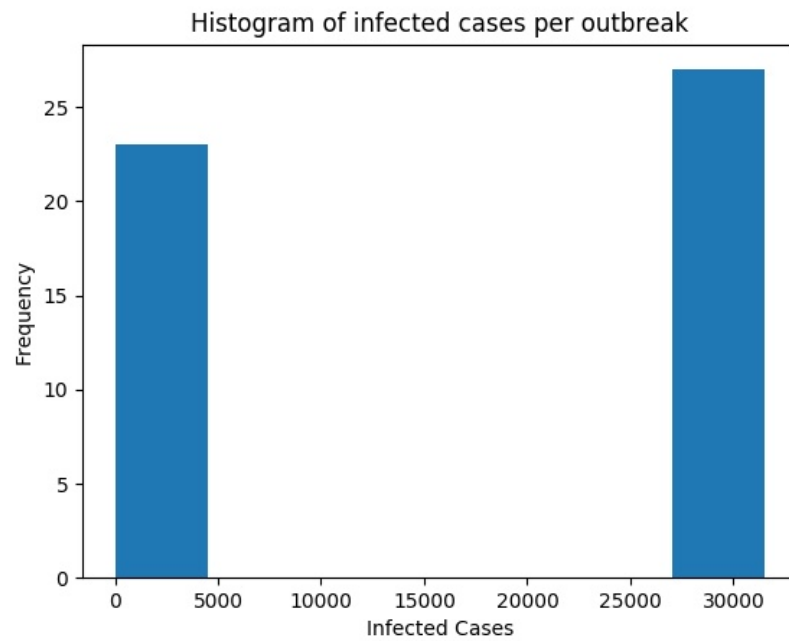
Histogram of infected cases per outbreak

Fig. 3.

Histogram of infected cases in small outbreaks
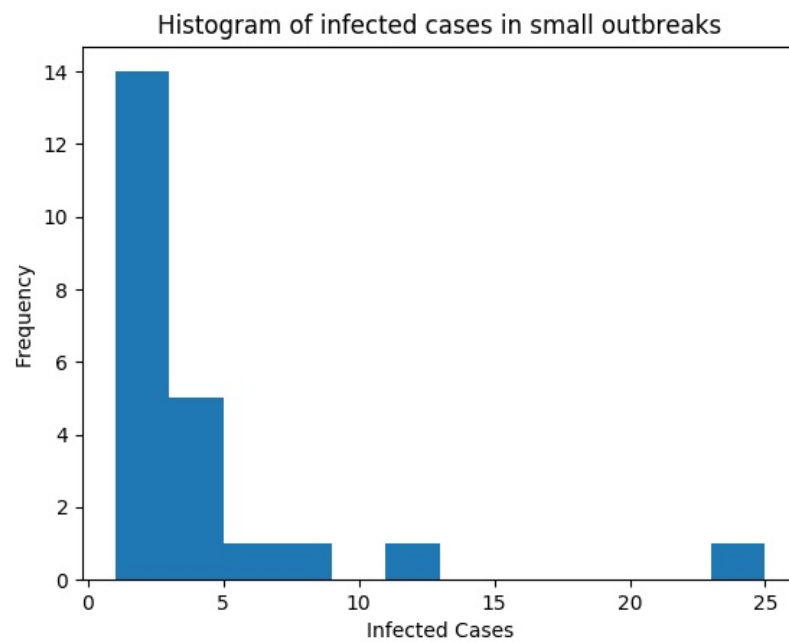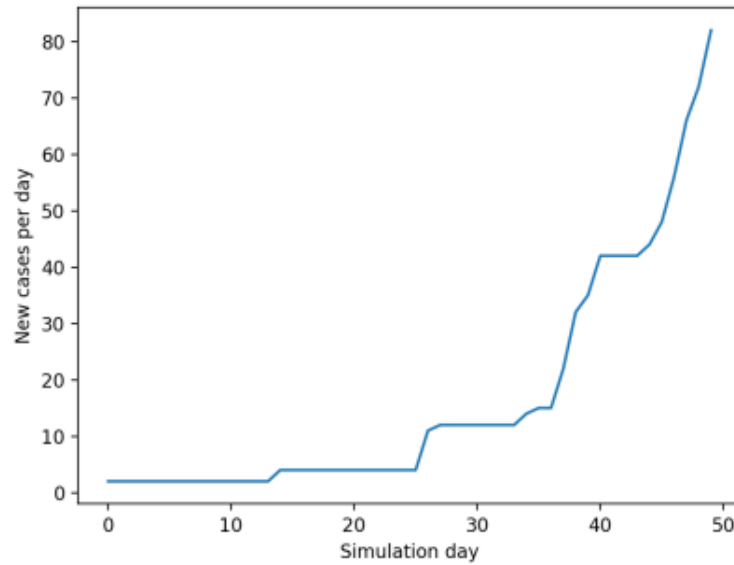

Fig. 4.

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.
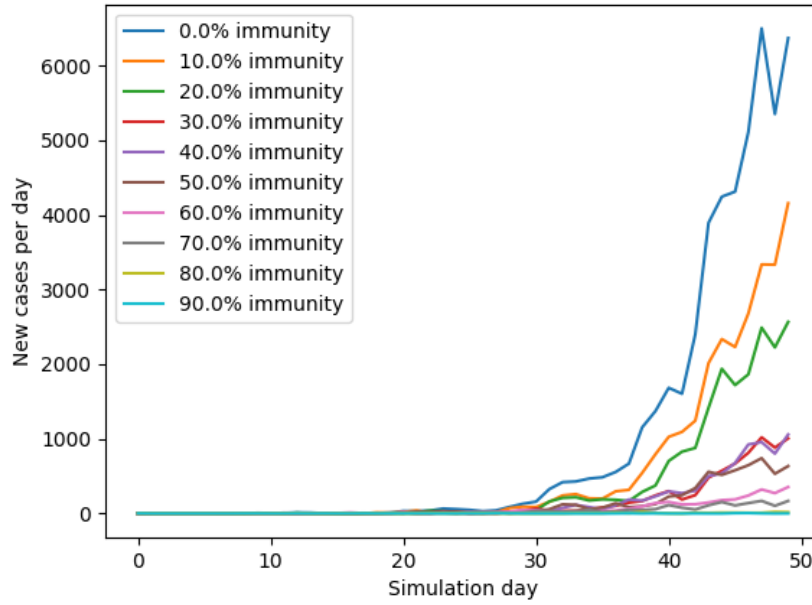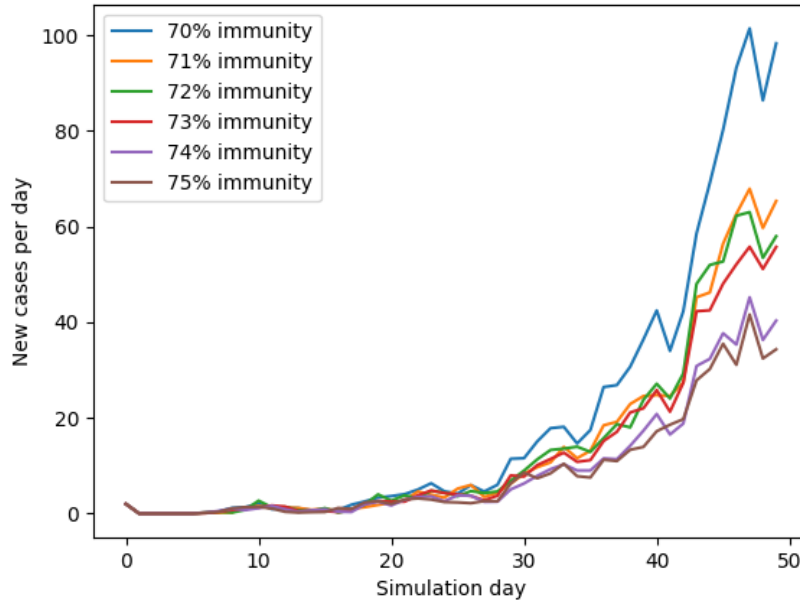

Fig. 6. first estimate of outbreaks

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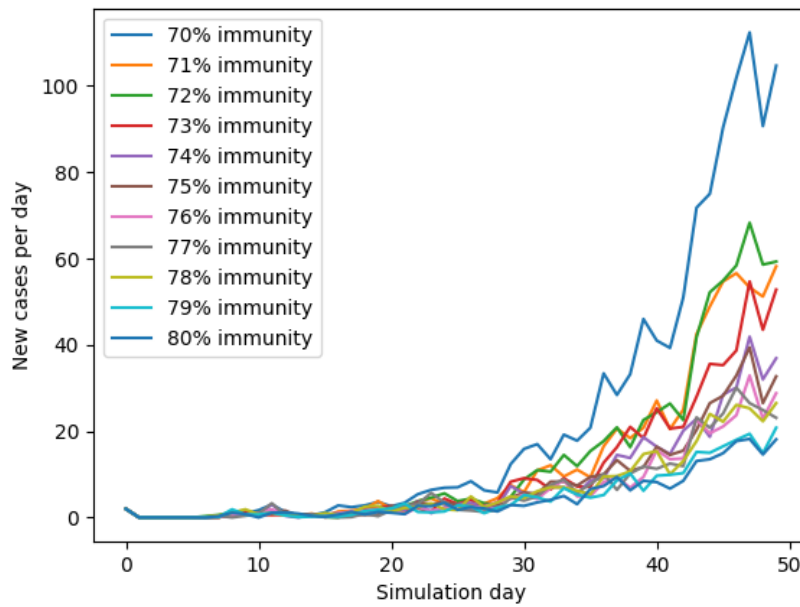
Fig. 7. narrowed down estimate of outbreaks based on 10 simulations

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.

A first estimate is done with the same immunity rate found in the previous experiment and R0 varying between 12 and 18. We should expect that a lower R0 will result in a lower amount of infections. We will then try to replicate the correct immunity rate for each R0 and try to determine which graph fits the best.
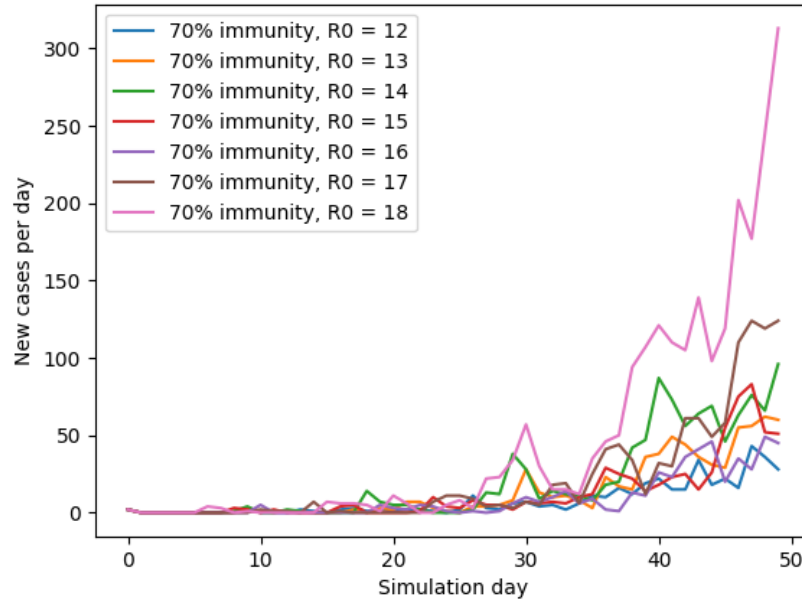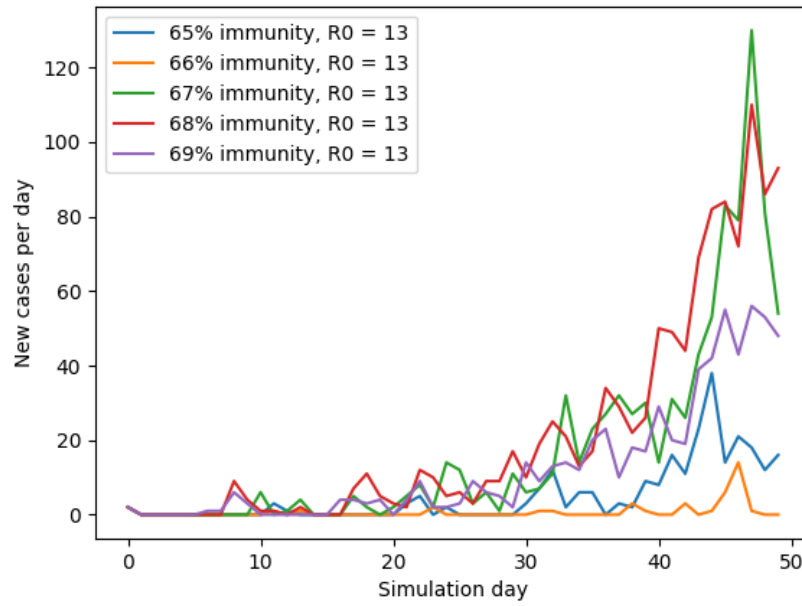

Fig. 9. simulations with an immunity rate of 70%

For R0 = 12, the optimal immunity rate is somewhere around 65%, this will mean that 65% immunity rate is the lower bound for all values of R0 if we want to find the same graph.
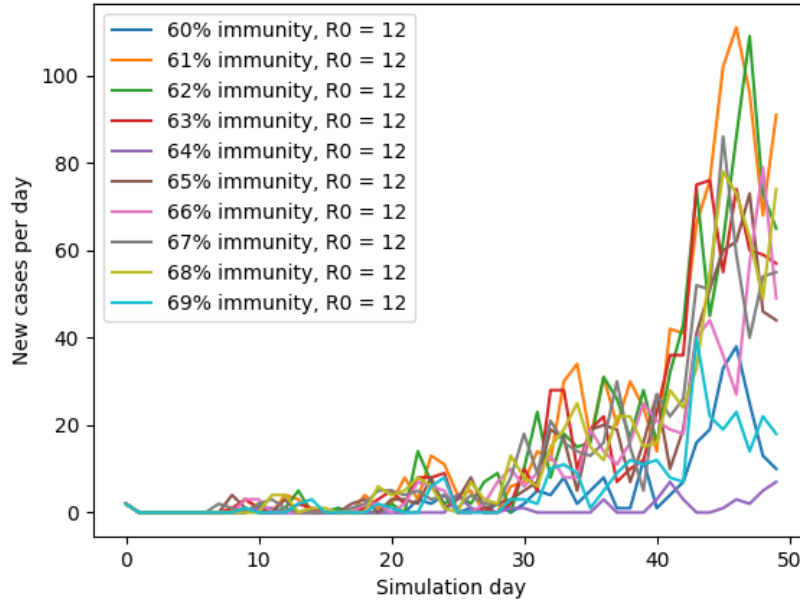
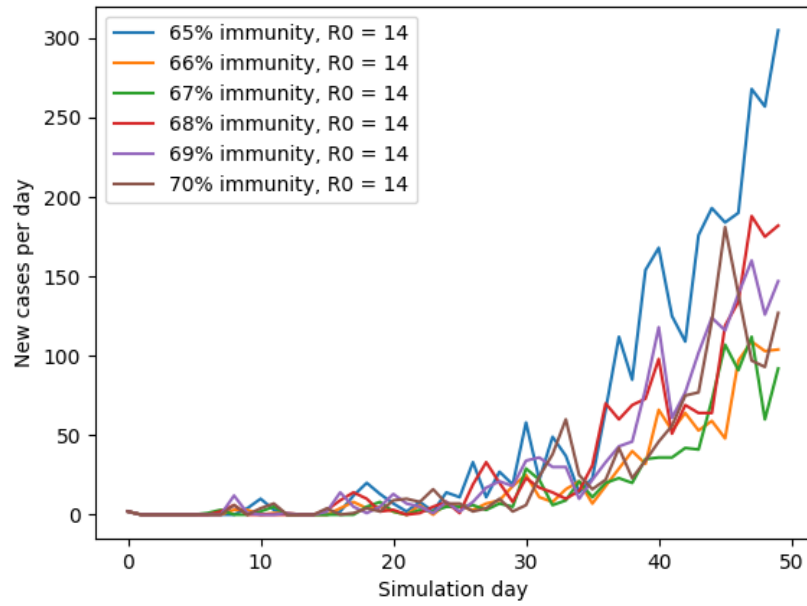**Fig. 10.** simulations with R0=12

For R0 = 13, the estimated immunity rate is around 68%.

Fig. 11. simulations with R0=13

For R0 = 14, the estimated immunity rate is around 69%.

Fig. 12. simulations with R0=14

For R0 = 15, we already know from the previous section that the estimated immunity rate is 70%, so we don't need to recompute that one.

For R0 = 16, the best fit seeems to be with an immunity rate of 73%.
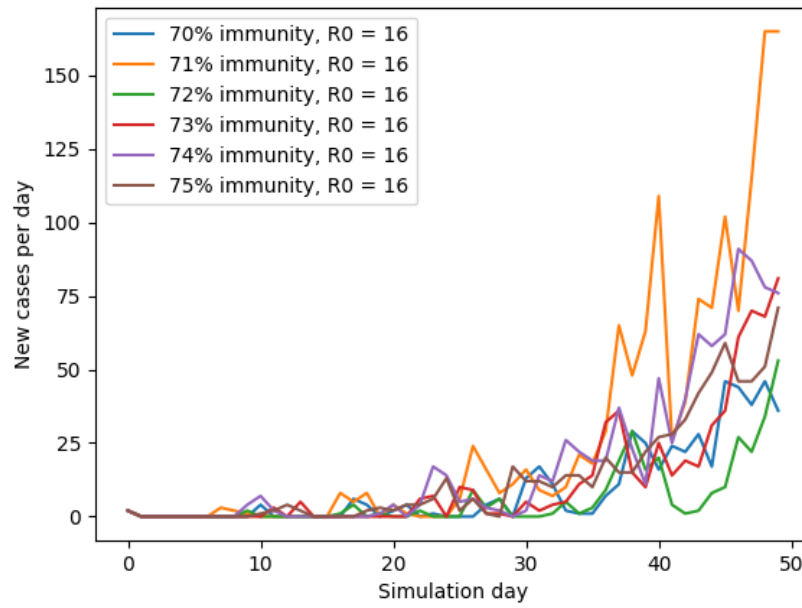
Fig. 13. simulations with R0=16

For R0 = 17, it seems to be that an immunity rate of 74% gives the best results.
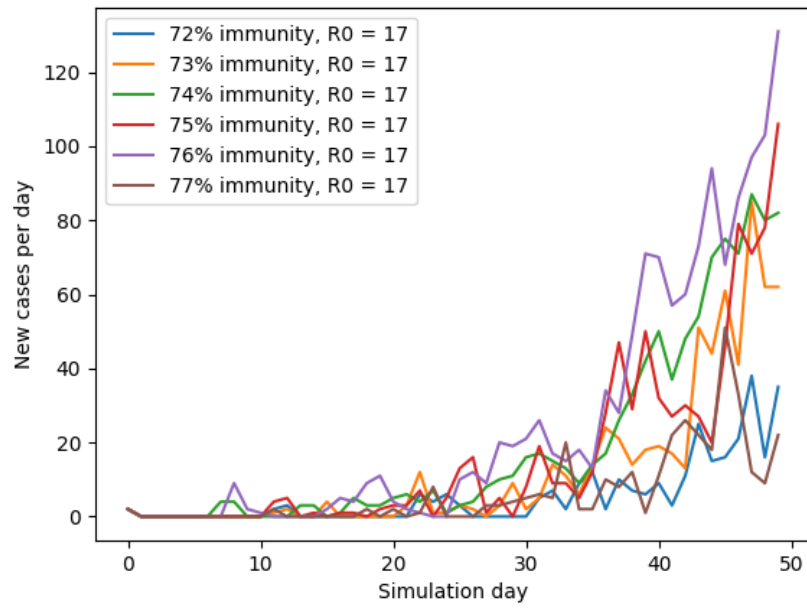
Fig. 14. simulations with R0=17

For R0 = 18, it seems to be that an immunity rate of 76% gives the best results.
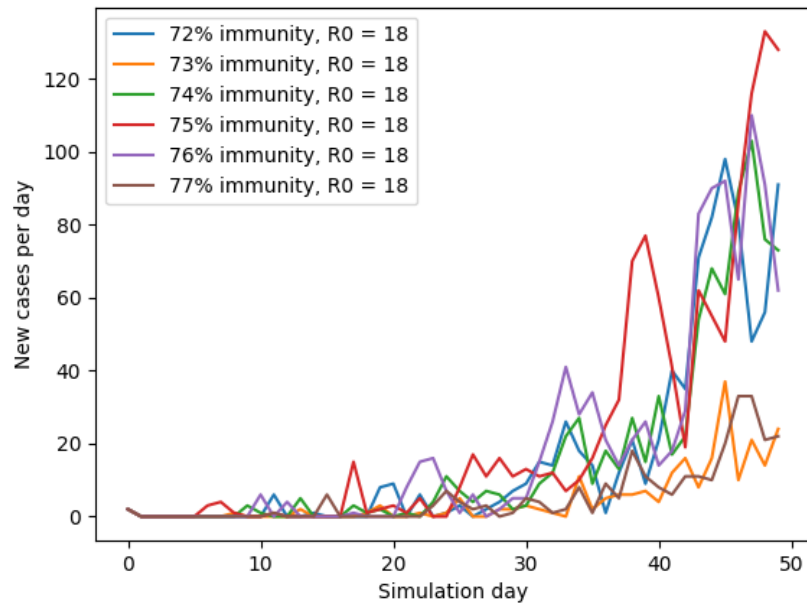
Fig. 15. simulations with R0=18

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.
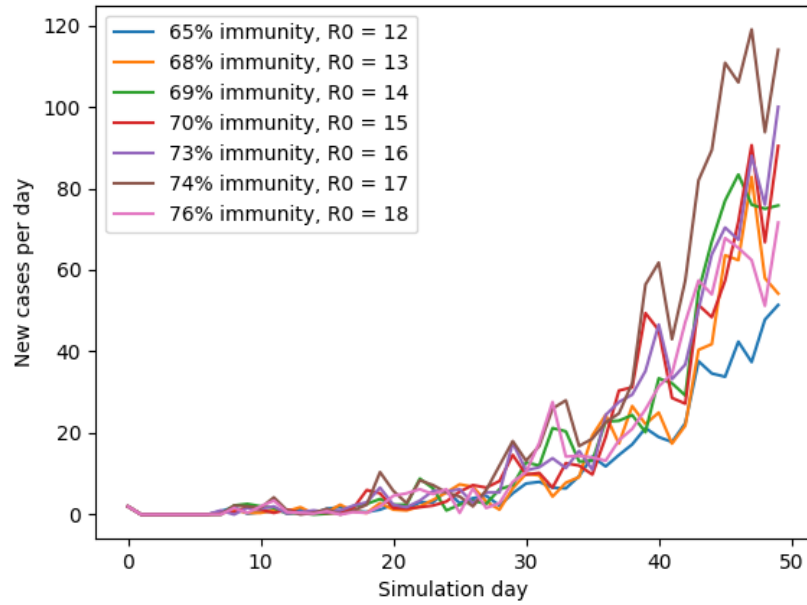
Fig. 16. 5 simulations for found immunity values

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.
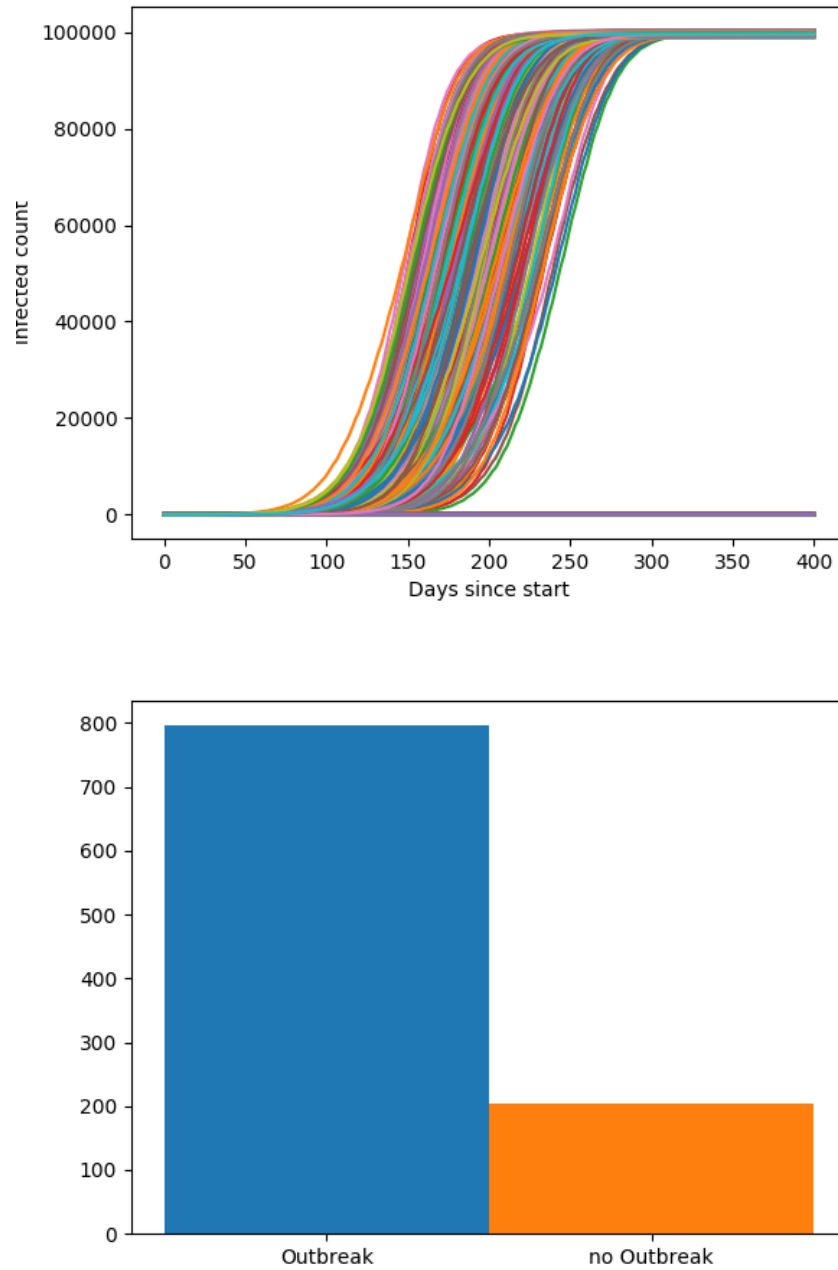
Fig. 17. Results of 1000 simulations with population A

Fig. 18. Results of 1000 simulations with population B

2.2   Vaccinating on campus

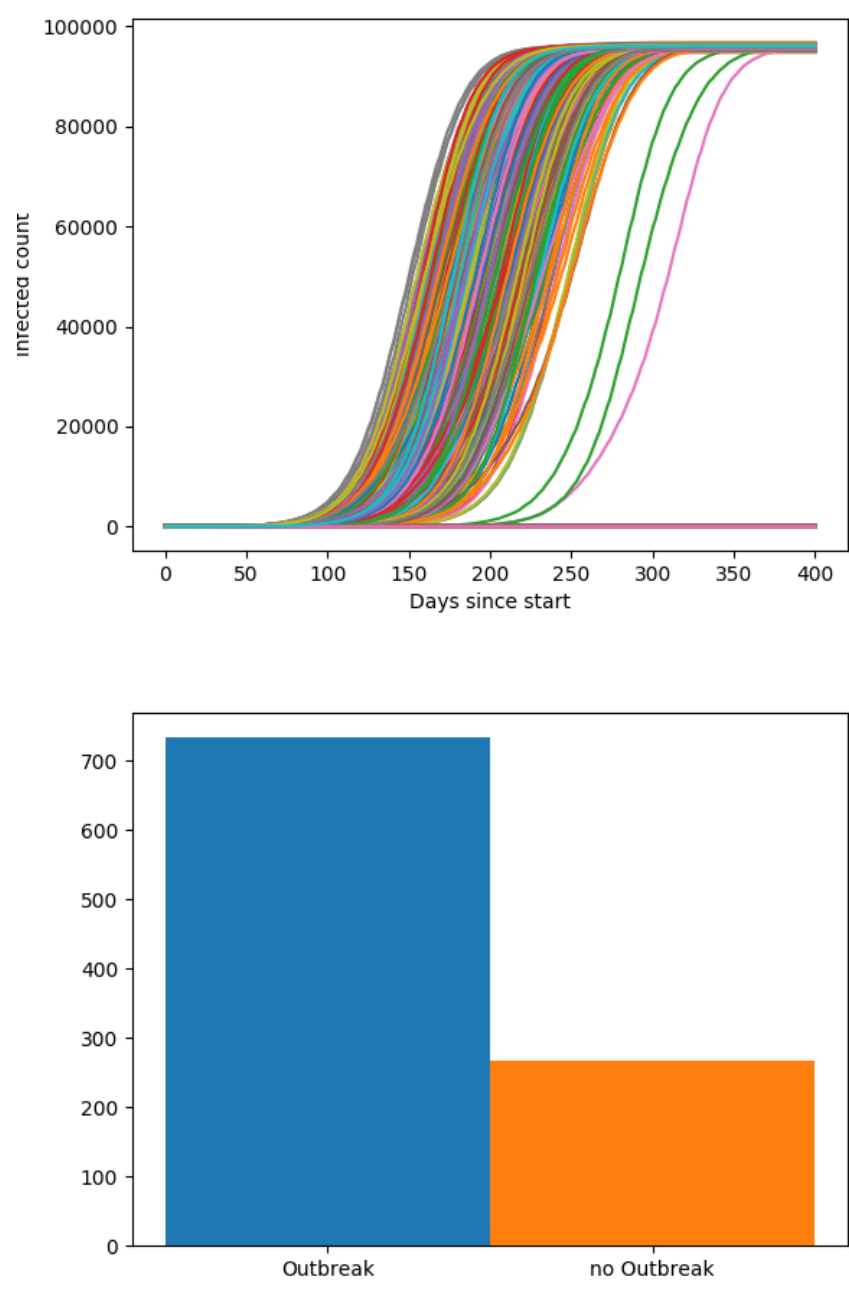Vaccinations are often given at a young age. But not everyone gets vaccinated. In this test we see what happens if students have a lower immunity and if vaccination during the spreading of the disease is useful.

In total 120 simulations have been run. 60 Without vaccinations and 60 with vaccinations given seven days after the first infected is introduced. For both runs a boxplot is created (figure 19 and 20). The y-axis represents the amount of infected cases at the end of the simulation. It is clear to see that even in the worst case, the disease willl spread less with vaccinations than in the best case without vaccinating. With only around 3500 student needing vaccinating in a studentbody of around 60000 and a total population of ten times that, the cost saving is quite high and the amount of infected is heavily reduced.

In conclusion, vaccinating student who still need their vaccinations or who are behind on them, is worth it during an outbreak.
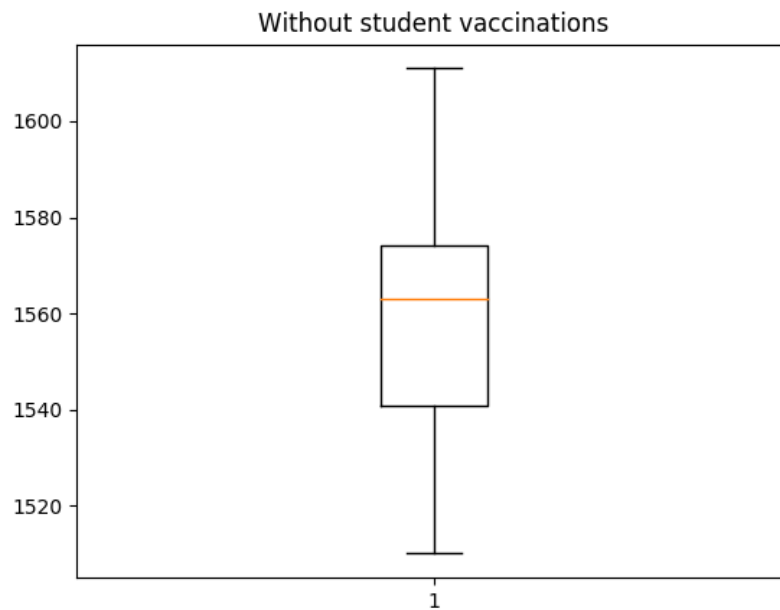
Fig. 19. Boxplot of 60 simulation where the students aren't vaccinated after seven days. The number of total infected people after 50 days is seen on the y-axes
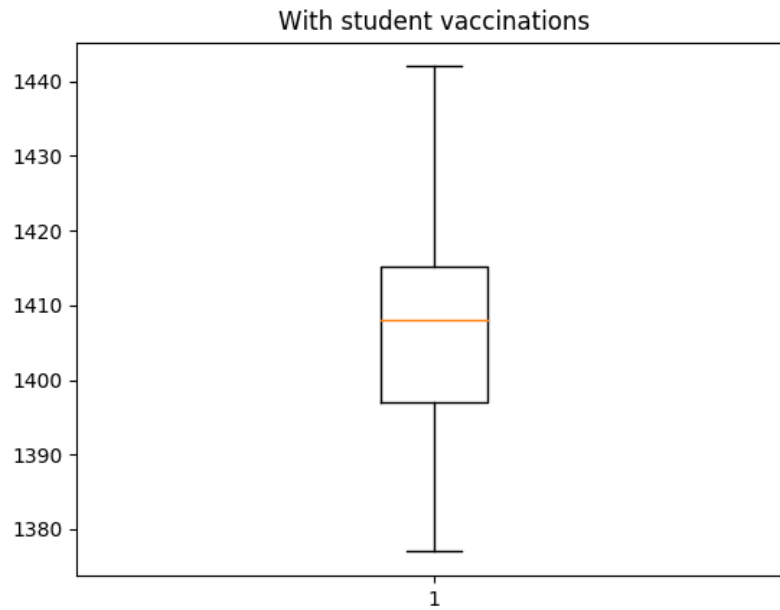
Fig. 20. Boxplot of 60 simulation where the students are vaccinated after seven days. The number of total infected people after 50 days is seen on the y-axes

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. In a first simulation we generated simulations for 5 different commuting percentages. As you can clearly see, it does affect the rate at which the disease spreads, but it has no, or little, effect on whether the disease does or does not spread. In all cases, the entire population got the disease, be it that it took a few days longer to get to that point. Another thing we can remark is that the highest "peak" of newly diseased people is lower the lower the commuting factor gets. A possible cause of the lack of serious effect can come from the fact there are a lot of college commuters who will have the same effect as the workplace commuters.
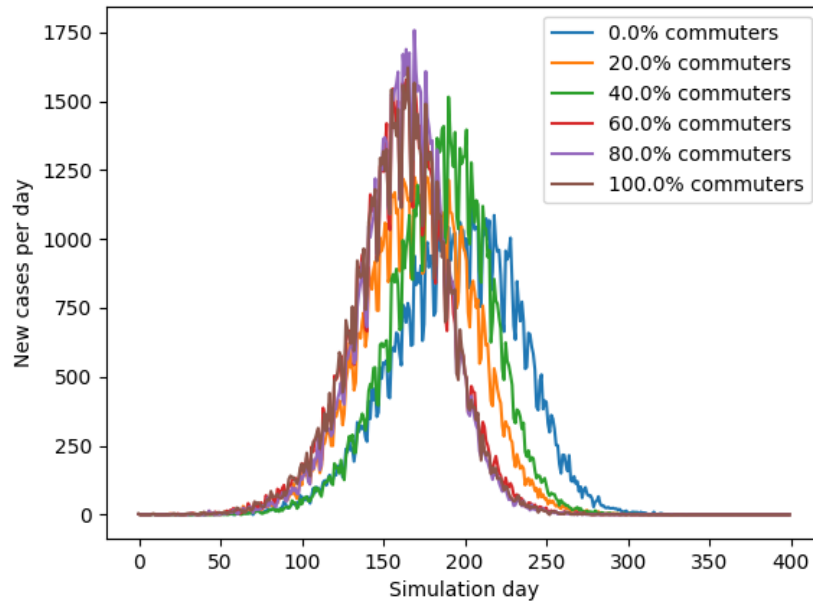
Fig. 21. Results of 6 different percentages of commuters in a range from 0 to 100.

If you watch the top of the graph, you can see that from a certain fraction and onwards, there is little to no difference in their behavior. To get a better view we graphed a closeup of percentages between 30 and 70. In the next graph you can see that the peeks are almost equally in height. You can even notice that some of the higher percentages have their peeks later then the ones with a lower percentage.
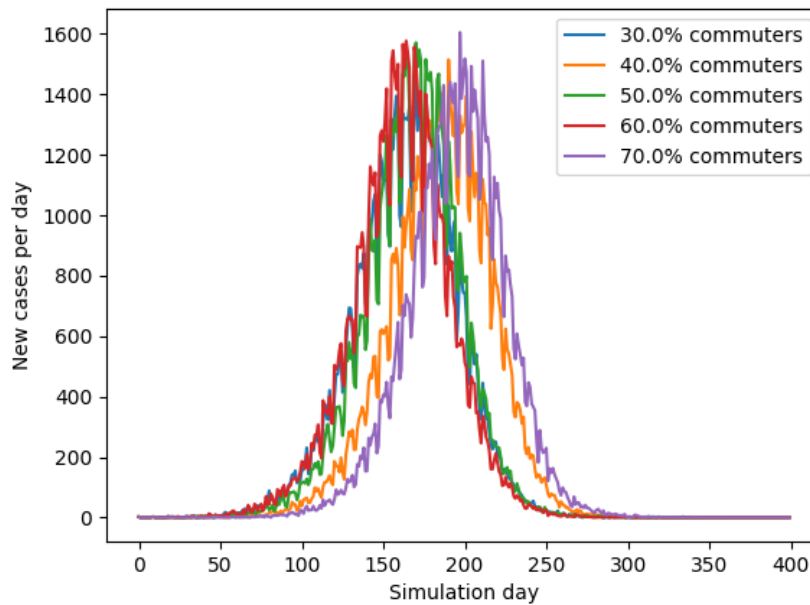
Fig. 22. Results of 5 different percentages of commuters in a range from 30 to 70.

Considering the recorded data we can see that solely changing the percentage of people who commute to work will not affect the spreading of disease in a significant manner. The disease will still spread all the same, but at a slower pace.

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 1. 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 2. 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 3. 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 4. Relation between immunity rate and runtime for a 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 5. 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 6. Relation between teh 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