BA Project Simulation Assignments

February 19, 2019

1 Simulation

1.1 Stochastic variation

The Stan (STochastic ANalysis) controller makes it possible to investigate the influence of stochasticity on simulation results. It runs a given number of simulations, for the same parameter configuration, but with different random number seeds. The output for the different runs is aggregated in a single .csv file. This file contains, for each random number seed (columns), the cumulative number of infected cases per time-step (rows). The first row specifies the random number seed used for each run.

The Stan controller can be addressed from the command line, in the following way:

```
$ ./bin/stride -e sim -c [CONFIG FILE] --stan [COUNT]
```

For this assignment, use the configuration file *stochastic_analysis.xml* (present in the *config* folder of Stride). [COUNT] indicates the number of simulation runs that should be executed. You can choose the value of [COUNT] yourself.

Using the configuration file that was supplied, run a number of stochastic simulations. Next, use the aggregated output to plot the distribution over the different stochastic runs of:

- The number of cumulative cases per time-step
- The number of new cases per time-step

What can you say about the distribution of cases per time-step? Does chance seem to be an important factor in determining the outcome of a simulation? Can you think of an explanation for the distribution of cases per day that you observe?

1.2 Determining an extinction threshold

The introduction of an infected individual in a partially susceptible population does not always lead to an outbreak. In some cases, only a few or even no secondary cases are observed, while in other cases a large outbreak occurs. Often, we are only interested in the latter. If this is the case, simulation runs where no outbreak is observed could skew our results - as we have observed in the previous assignment.

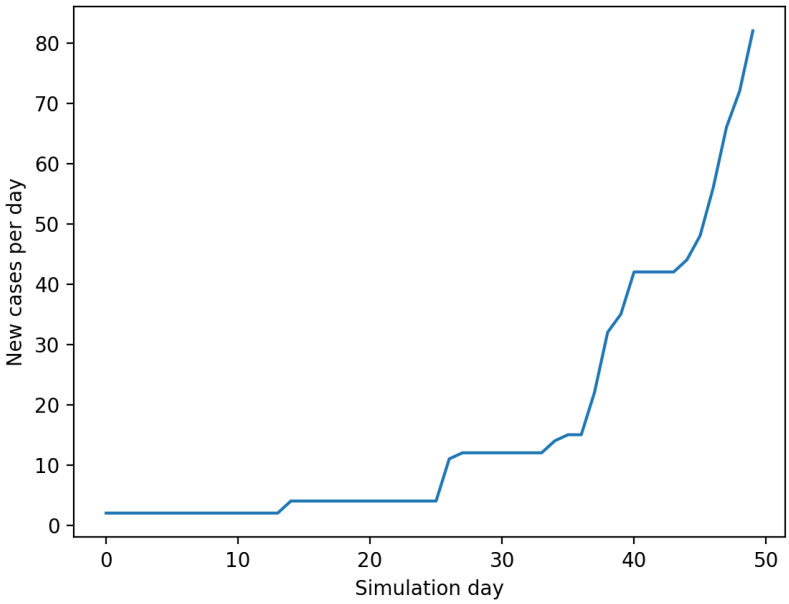

Figure 1: New cases observed per day during the outbreak.

When there are only few or even no secondary cases, we say that extinction occurs. To interpret the results of our simulations, we need to be able to make a distinction between cases where extinction occurs, and cases in which there is an outbreak. For this purpose, we need to determine a threshold. We can do this by looking at the final number of infected cases at the end of each stochastic simulation for a given configuration of parameters.

Use the configuration file *stochastic_analysis.xml* to run a number of stochastic simulations. You can choose this number yourselves. Create a histogram of the frequencies of the final number of infected cases over the different simulations. Describe what you see. Is it possible to determine a threshold distinguishing extinction cases from outbreak cases? What threshold would you choose in this case?

1.3 Estimating the immunity level

Suppose we have data about the evolution of a recent measles outbreak. Over the course of 50 days, data was collected on the number of new cases that was observed each day. The result can be seen in Figure 1.

We already have a good estimate of most relevant parameters. These are described in the configuration file outbreak_2019_estimates.xml. However, we do not have any information about the number of individuals that was immune to measles prior to the recorded outbreak. Using Stride, we could try to estimate the immunity level of this population.

Vary the parameter that determines the immunity level in the population,

and try to estimate which was the case in the original population. Remember that:

- Stochasticity plays a role in Stride
- The data we have are from an *outbreak* of measles

It is useful to use the PyStride environment for this assignment. However, it is not required.

1.4 Estimating R_0

In the previous assignment, we assumed that the parameter R_0 was fixed at 14. R_0, often also called the basic reproduction number of a disease, is the number of individuals one infected person would infect in a completely susceptible population. This only takes into account direct infections - so not secondary, tertiary, etc infections.

However, this number is only an estimation. For measles, it is generally assumed that the basic reproduction number lies between 12 and 18. To test whether a conclusion is stable, we need to see if it holds for different values of R_0.

In Stride, R_0 is supplied as an input parameter. From this value, the transmission probability in the event of contact between an infectious and a susceptible individual is calculated. The formula used to this is calibrated based on the population, disease characteristics and social contact patterns used.

Re-examine the question of the previous assignment, but take different values of R_0 into account. What do you notice? Is your conclusion dependent upon your estimation of R_0?

2 Population generation

2.1 Investigating the influence of demography on epidemics

When there is an outbreak of an infectious disease in a population, not all age groups are equally affected. The demography of a population has an impact on the risk for oubreaks of a disease.

We have data for populations in Region A and Region B, which are, for the most part, quite similar. However, there are significant differences between the age distributions of both populations. The population of Region A is significantly younger than that of Region B. A sample of the households in both regions was taken, and the age distributions present in these samples can be seen in Figure

The sample of households from Region A can be found in the file households_regionA.csv, while that of Region B can be found in households_regionB.csv. Both files can be found in the *data* folder of Stride. Other parameters relating to the consitituion of both populations can be assumed to be similar to those found in the run_generate_default.xml configuration file.

Generate populations for both regions, and investigate how the risk for an outbreak is different between the two regions. To save time, you can generate both populations once, and then re-use the generated populations by setting the *population_type* parameter to *imported* and specifying the correct file.

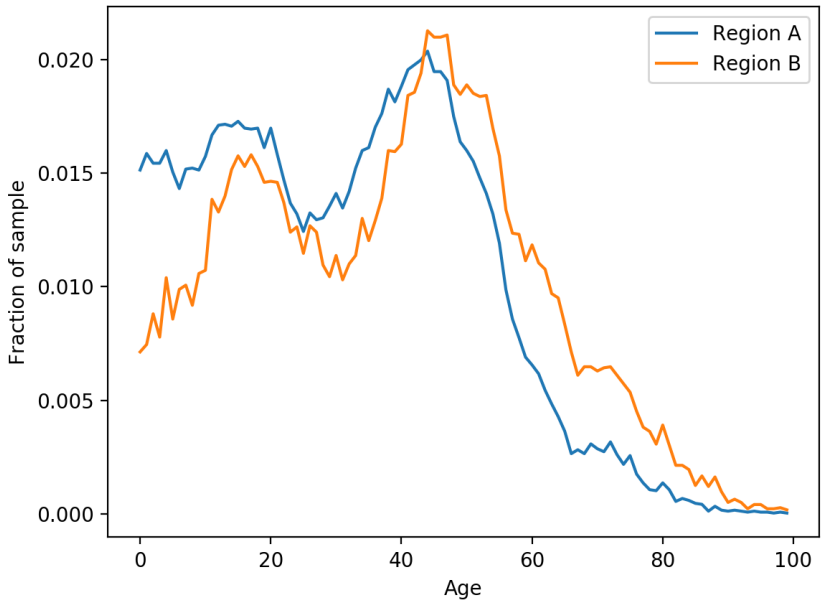

Figure 2: Comparison of age distributions in household samples from region A and region B.

For each population, determine the percentage of runs that results in an outbreak. Which population has a higher chance of outbreaks occurring?

2.2 Vaccinating on campus

Focus is often placed on the vaccination of infants and young children, but older individuals also risk being infected when left unvaccinated. So-called 'catch-up' campaigns address this issue by vaccinating older persons who had not yet been completely vaccinated.

Generate a population where 60% of 18 to 26 year olds attend higher education. Use an age-dependent immunity profile, as supplied in *lower_student_immunity.xml*. In this scenario, 18 to 26 year olds are insufficiently vaccinated against measles. Other parameters can be assumed to be the same as those in the run_generate_default.xml configuration file.

Investigate what the effect would be if, a week after an individual infected with measles is introduced in the population, all students attending higher education would be vaccinated. Compare this to a situation where no action is taken. To do this, you can write a call-back function and register it through the PyStride environment.

2.3 Is commuting to work important for disease spread?

When individuals commute to another city every day to attend their workplaces, this might have an influence on how fast a disease spreads. Investigate the

impact of the percentage of working persons that commutes to another city on the risk of outbreaks, and their size. Furthermore, examine whether the 'peak' of the epidemic - this is the day on which most newly infected cases are observed - is impacted by the number of commuting individuals in the population.

Generate different populations, varying the percentage of the population that commutes to another city for their work. Other parameters can be assumed to be similar to those in the run_generate_default.xml. Discuss the outbreak occurrence, sizes, and evolution over the different scenarios.

3 Performance profiling of sequential code

When writing scientific software, it is important to have an idea of its performance in different scenarios. For this purpose, one should keep track of which parts of the code take up the most time, and how this varies in relation to different parameters. Analyze, for population generation as well as simulation, how much time each procedure takes up. The most interesting result can be obtained by recording the time used by each procedure that is called, instead of aggregating the run-times of smaller procedures called by a larger procedure. To perform these analyses, you can turn of OpenMP, so that all code is run sequentially.

Several tools are available to perform this analysis: GProf, Valgrind, ... You can choose yourselves which tool to use.

Analyze the impact that the following parameters have on the performance profile:

- Number of days
- Population size
- Immunity rate
- Seeding rate
- Contact log mode