

Computer aided simulations and performance evaluation lab N4

Matteo Latino S269317

1 Introduction: SIR model

The SIR model is a mathematical model of an epidemic. It is composed of a system of differential equations, on which the population is divided into Susceptible S , Infected I and Removed R . $S(t)$, $I(t)$, and $R(t)$ describe the evolution of the epidemic in the population and $\forall t \ S(t) + I(t) + R(t) = N$ which is the population's size. The aim of this laboratory is to implement a SIR model through an agent-based approach. In particular we will assume that:

- The population is made of $N = 10^3$ people.
- The infection period is 14 days.
- Each individual gets in contact with 0.2 people per day.

2 Numerical SIR model

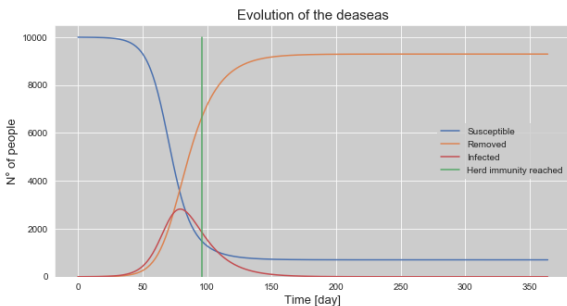
The differential equation of the SIR model can be solved by using a numerical approach. Starting from the definition of derivative, we can find that:

$$\begin{aligned} S(t+1) &= S(t) - \frac{\beta}{N} S(t)I(t) \\ I(t+1) &= I(t) + \frac{\beta}{N} S(t)I(t) - \gamma I(t) \\ R(t+1) &= R(t) + \gamma I(t) \end{aligned} \tag{1}$$

So, the values at the time $t+1$ can be computed by starting from the values at the time t . The main parameter of the *SIR* model are:

- β : it indicates the infection transmission rate, per capita. In this case beta is $0.2 \left[\frac{1}{day} \right]$
- γ : it is the recovery rate. In this case gamma is $\frac{1}{14} \left[\frac{1}{day} \right]$

An important index that can be obtained, starting from the previous parameter is the reproduction number R_0 , it indicates the average number of infections that occurs when an infected person is introduced into a fully susceptible population. It can be computed as $\frac{\beta}{\gamma}$, and in this case is equal to: 2,8.



This graph shows the evolution of the epidemic during 1 year, through the values of $S(t)$, $I(t)$ and $R(t)$. The maximum number of infected people is 2823, and it occurs 80 days after the first infection. From that day on, the number of infected people starts to decrease until the epidemic ends. I think that the last day of the epidemic could be matched with the establishment of the herd immunity in the population. It is a type of indirect protection that occurs when the majority of the population is immune to the virus, in this way the chain of infection is disrupted frequently and the probability that a non-immune individual comes into contact with the disease, becomes quite low. It is the point where the disease reaches an endemic steady state, so the infection level will remain constant.

The exact herd immunity threshold depends on the type of disease, and can be defined starting from the value of the effective reproduction number R_e which is obtained by the product of the basic reproduction number R_0 , and S (the proportion of the population who are susceptible to infection).

At the endemic steady state $R_e = S \cdot R_0 = 1$

Now, S can be rewritten as $(1 - p)$ where p is the proportion of the population that is immune, thus: $p = 1 - \frac{1}{R_0}$ indicate the portion of the population needed to establish the herd immunity. R_0 works as a measure of contagiousness, so low R_0 values are associated with lower p , whereas higher R_0 s result in higher p . In this case p is 64,28%, so we need at least 6428 immune people in the population. In the *SIR* model the immune people correspond to the removed people, so from $R(t)$ it is possible to see that the herd immunity is reached at the day 96.

3 Simulative SIR model

The objective is to mimic the behavior shown in the previous section, by using an agent-based simulation which will evolve through a discrete time period.

3.1 Input variables

The input variables of the simulator are:

- The number of days for which the simulation must be performed.
- N : the size of the population.
- γ and β as defined before.
- The initial seed, the confidence level and the number of runs.

3.2 Output metrics

The output metrics of the simulator are:

- $S(t)$, $I(t)$ and $R(t)$ for each day.
- The max number of infected people, the day on which it is reached and the relative confidence interval.
- The last day of the epidemics and its confidence interval. This day is computed by considering as before the herd immunity.
- The evolution of the R_t in function of the day and the relative confidence interval.

3.3 Structure of the simulator

3.3.1 Data structure

The data structure is composed of 2 main classes: *Person* and *Environment*.

The class Person: It represents the agent that moves inside the environment and whose behavior influences the outcome of the simulation. Its main attribute are:

- γ : it's the recovery rate of the person.
- *Day_start* and *day_end*: are the days on which he becomes infected, and the day of the healing.
- *N_infected*: Number of people infected by that person.
- *day_infection*: is the list of days on which that person infected another one. If on the same day more than 1 person has been infected, that day will be repeated more times.
- state: indicate the actual state of the person, it may have only 3 possible value:
 - 0: the person is susceptible to the disease but is still healthy.
 - 1: the person is infected by the virus, and could also infect other people.
 - 2: the person has recovered from the disease and can not be infected anymore.

The main methods implemented by this class are:

- *Become_infected*: change the state of the person to 1 and update the *day_start* attribute.
- *Recover*: check if the person may recover from the disease. The duration of the disease is distributed like a Geometric distribution with mean equal to $\frac{1}{\gamma}$, on which the success is represented by the healing of the patient. It can be simulated by using a Bernoulli with probability $p = \gamma$ each day until the person does not recover.

The class Environment: It represents the environment on which the person will move during the simulation, since the *SIR* model assumes an homogeneous mixing of the population (each individual may come into contact with any other person in the population) none boundaries are given for the contact among people.

To instantiate an object of this class are required: the size of the population (N), the transmission rate (β) and the recovery rate (γ) of the population. The first parameter is necessary to dimension an array which will represent the population, each element of the array will be an instance of the class *Person*; the other 2 are the main parameters of the disease that will influence the behaviour of the simulation. Other attribute of the *Environment* class are:

- S , I and R : are the actual number of Susceptible, Infected and Removed people.
- Day : is the actual day on which is the environment.

The main methods implemented by this class are:

- *Infect*: It simulates the spread of the disease among the population. For each person, firstly is determined how many people he will meet on that day by using a Poisson distribution with an expected rate of occurrences equal to β , then is analyzed if a transmission of the virus may occur. The values of S and I are updated consequently. The Poisson distribution will represent the number of meetings that may occur within the end of the day.
- *Recover*: It simulates the recovery of the people, for each individual, is called the recover function of the class *Person*, and the value of I and R are updated consequently.
- *Next_day*: update the day for the environment.

3.3.2 Algorithm

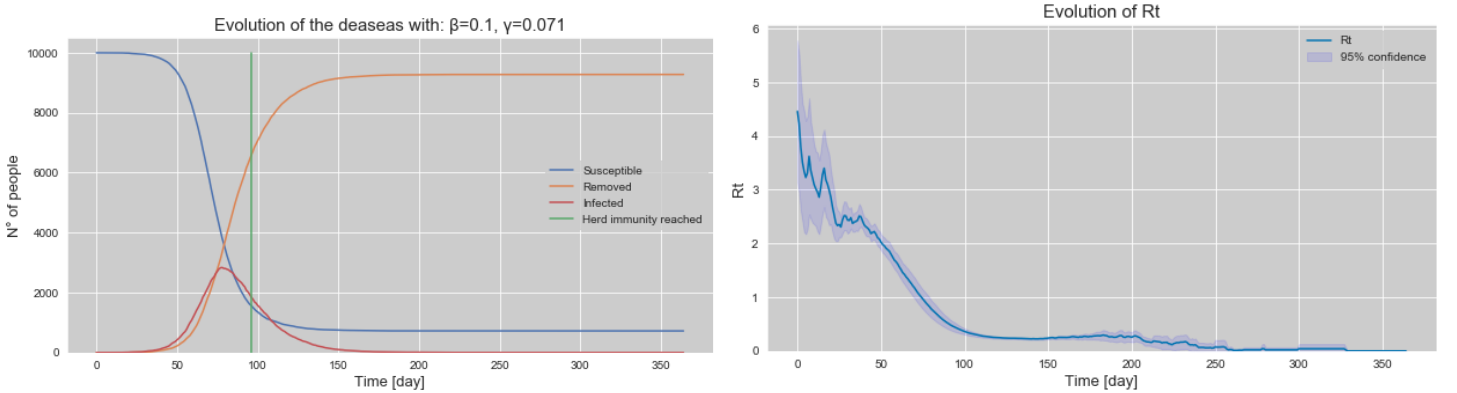
To simulate the spread of the disease during a given period of time, in this case is 1 year, first is instantiate an object of the Environment class that by default will generate the first positive case in the population, then by using the *infect*, *recover* and *next_day* methods described before, the environment is moved forward one day at time.

For each day, the number of infected and removed people are stored, then the day with the max number of infected people and the last day of the epidemic are determined.

Once the simulator ends, it is possible to compute the R_t value for each day, by using all the information stored during the simulation. Since R_t does not take into account the past history but only the future, before calculating the value for R_t at the $t + 1$, all the infections at the time t are removed from the counter inside the class *Person*, by using the information contained in the *day_infection* array.

This process is repeated, for each single run of the simulation, and finally are computed the final results and the confidence intervals.

3.4 Results



The first graph shows the result of 1 non degenerative run of the simulator with $\beta = 0.1$ and $\gamma = 1/14$. Can be observed that, the value for β is the half that the one used in the numerical *SIR* model, because during the simulation must be take into account that, if a person A meet a person B with a probability β , there is also another person C that can meet A with the same probability, as consequence the average number of contact made by a person result to be doubled.

In the table below is possible to see more in detail the result obtained with multiple runs of the simulator. Can be noted that some runs result in a degenerative outcome (the disease does not spread across the population) even if this behavior is not predicted by the numerical model. These runs cause a huge variation on the values of the output metrics, to obtain a more precise result I decided to remove the values that come from a degenerative run, before computing the final results and the confidence intervals.

	Max I(t)	Max I(t) 95% C.I.	Day Max I(t)	Day max I(t) 95% C.I.	Day end	Day end 95% C.I.
Numerical SIR model	2823	-	80	-	96	-
Simul. SIR model W/	2208.57	[1788.53 ; 2628.6]	66.57	[53.78 ; 79.35]	80.3	[65.61 ; 94.99]
Simul. SIR model W/O	2760.42	[2726.33 ; 2794.5]	82.83	[78.65 ; 87.02]	99.125	[95 ; 103.25]

The value of the numerical *SIR* model results to be quite similar to the one obtained with the simulation, in particular the day with the max number of infected people and the day of end of the epidemic fall inside the confidence interval, while the max number of people infected results to be lower than the one predicted by the numerical model. The second graph, instead, shows the trends of the R_t value during the epidemic, as expected it has a decreasing behavior, until it reaches 0 when there is no one infected in the population. Is possible to see that during the initial phase of the epidemic the R_t value is quite unstable, and as consequence the confidence intervals are wider. Then it becomes more stable, until the epidemic enters in the final period, where another small oscillation appears.

4 Extension of SIR model

The main objectives are to remove some criticism of the *SIR* model, like the tele-transport mobility, and increase the level of detail of the simulator. In particular:

- To avoid the tele-transport of the people, the contact among them may happen only if they are in the same location, like people who went to the same market or park. A person can move toward a location with a probability P_{out} , otherwise he is in none location and it's like he is at home and he can neither infect nor be infected.
- In each location the probability to have a contact could be different, for example, in a small shop, the probability of having contact with another person could be greater with respect to a big park, where it is easier to keep the distancing. As a consequence each location could have a different β . Must be noted that this β , will not indicate the transmission rate per capita (which is related to the person) but the transmission rate in that location. The relation between these two β s is: $\beta_{person} = \frac{P_{out}}{N_{location}} \cdot \sum_{i=0}^{N_{location}} \beta_{location,i}$
- The people are divided into different categories, each category has a different recovery rate. For example, younger and healthy people will have a greater recovery rate, while older and sick people will have a slower recovery rate. For the distribution of the people among the different categories I have assumed an uniform distribution.

4.1 Input variables

The input variables of the simulator are almost the same of before, the main difference are:

- γ : now is a vector of all the possible recovery rates that a person could have.
- $N_{location}$: is a new input variable, it indicates the number of possible locations on the environment, as a consequence beta will be a vector of length $N_{location}$.
- P_{out} : is the probability that a person will move to one location.

4.2 Output metrics

The output metrics of the simulator are the same as the previous one.

4.3 Structure of the simulator

4.3.1 Data structure

The data structure is composed of 3 main classes: *Person*, *Environment* and *Location*. The class *Person* is the exactly the same described before, while the *Environment* class was modified to introduce the concept of location described before. In particular:

- To the attributes is added the *locations* array, it contains all the locations present in the environment.
- In the method *infect*, the contact among the people is simulated location by location, and then is checked among all the resulting pairs of people if an infection may occur.
- Is added the method *move*: it assigns each person to a given location. Each individual will move outside according to a *Bernoulli* distribution with a probability $p = P_{out}$, and the location will be chosen by using a *Uniform* distribution.
- Is added the method *set_restriction* which allows to modify the P_{out} probability and the different β s for the locations.
- The methods *recover* and *next_day* are the same as before.

Regarding instead the *Location* class, it represents a given location on which the people can meet each other. The attributes of this class are:

- The ID of the location and β as defined before.
- The *people_index* array: it contains the ID of all the people that are in that location.

The main method of this class is the *meet* method. It simulates the meeting between the different people that are into this location. The simulation approach is the same described before, the only difference is that, can be chosen only the individuals contained in the *people_index* array.

4.3.2 Algorithm

As before, first must instantiate an object of the *Environment* class, that will: assign each person to a given category, instantiate all the Location objects and generate the first positive case into the population. Then to move the environment forward in time, are called, in the following order, the methods: *move*, *infect*, *recover* and *next_day*.

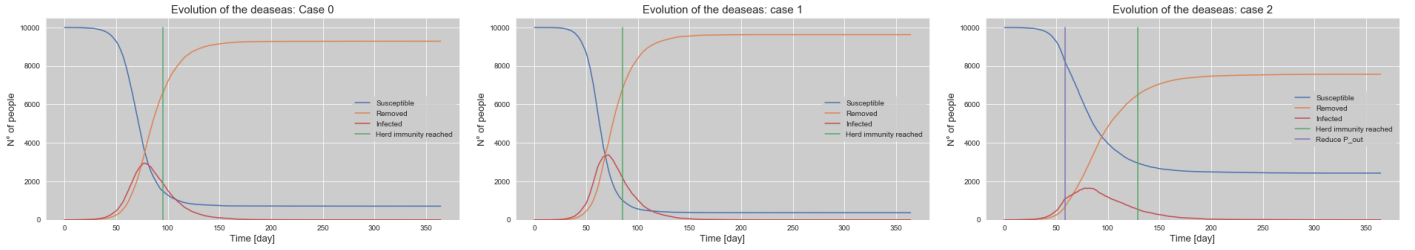
As before, for each day, the number of infected and removed people are stored, and the day with the max number of infected people and the last day of the epidemic are determined. Once the simulator ends, it is possible to compute the R_t value for each day, by using the same method described before.

First, the simulator is executed following the assumption done in the introduction (Case 0), then some modifications are added to show the effect of the different parameters into the behavior of the epidemic. In particular:

- Case 1: the $\beta_{location}$ of 1 location is doubled.
- Case 2: Once there are more than 1000 infected people, the P_{out} probability is reduced to 0,5.

4.4 Results

The graph below shows 1 non degenerative run for each case, while the table gives more detailed results about the output metrics.



	Max I(t)	Max I(t) 95% C.I.	Day Max I(t)	Day max I(t) 95% C.I.	Day end	Day end 95% C.I.
Case 0	2886.35	[2857.37 ; 2915.32]	80.77	[77.11 ; 84.43]	97.35	[93.83 ; 100.86]
Case 1	3523.79	[3485.38 ; 3562.2]	66.89	[61.77 ; 72.02]	80.31	[75.3 ; 85.33]
Case 2	1618.04	[1589 ; 1646.08]	83.92	[80.17 ; 87.68]	127.85	[124.29 ; 131.4]

As expected, the case 0 is the closest to the numerical *SIR* model, it includes both, the day with the max number of infected people and the day of end of the epidemics, inside its confidence interval, just like the previous simulator. The main difference with respect to the previous simulator can be observed in the max number of people infected, which in this case result to be greater (and not lower as before) than the value obtained by the numerical model.

In case 1, it is possible to see the effect, caused by increasing the $\beta_{location}$ of only 1 location, into the behaviour of the epidemic. The bell shaped curve of the infected people becomes higher and narrowed, as consequence the peak of infection is reached before and also the epidemics end before respect to the other cases, however this can be considered the worst case scenario because higher is the peak, higher is the load on the sanitary system and care for the sick becomes more difficult. In the last case, the objective is to limit the spread of the disease by reducing the number of people that will move across the locations, it is possible to see that in this way the peak of infection is smaller and is reached later in time, as consequences also the epidemic will end later. Moreover, it can be noted that the effects of the reduction of P_{out} are not immediate but shown themselves over time.

References

- [1] https://en.wikipedia.org/wiki/Herd_immunity
- [2] https://en.wikipedia.org/wiki/Mathematical_modelling_of_infectious_disease#Endemic_steady_state