UPC - Complex and Social Networks

Lab7: Simulation of SIS model over networks

Martí Renedo Simon Van den Eynde

January 6, 2017

1 Introduction

We will simulate a SIS epidemic model on networks. We will use a binary tree, a scalefree graph, an Erdos-Renyi graph, a complete graph and a star graph. We will also look at their thresholds (the inverse of the highest eigenvalue of the graph) and try to see how they influence the epidemic behaviour.

2 Results

2.1 Task 1

In this first section we present the results of the simulation using fixed parameters $\gamma = 0.05$ and $\beta = 0.005$ for all 5 networks. Figures 1, 2, 3, 4 and 5.

2.2 Task 2

The following graphs are the result of a simulation on 5 different graphs with 100 timesteps. For every graph we took $\gamma = .1$, then we calculated the thresholdvalue for β. We increased β by 10% (plotted dots) and decreased β by 10% (plotted red line). We started with 5% infected

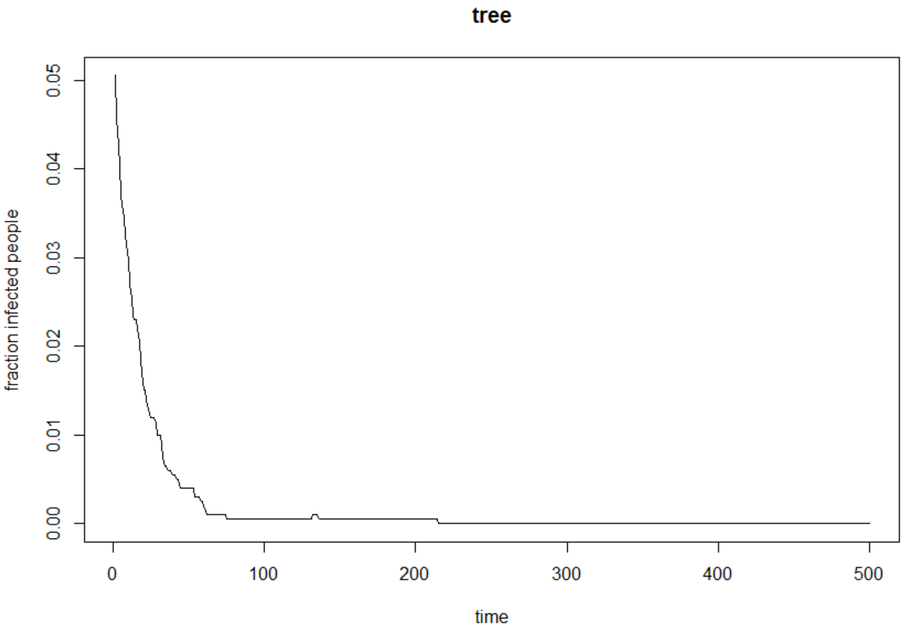


Figure 1: Task 1. Fixed parameters $\gamma = 0.05$ and $\beta = 0.005$

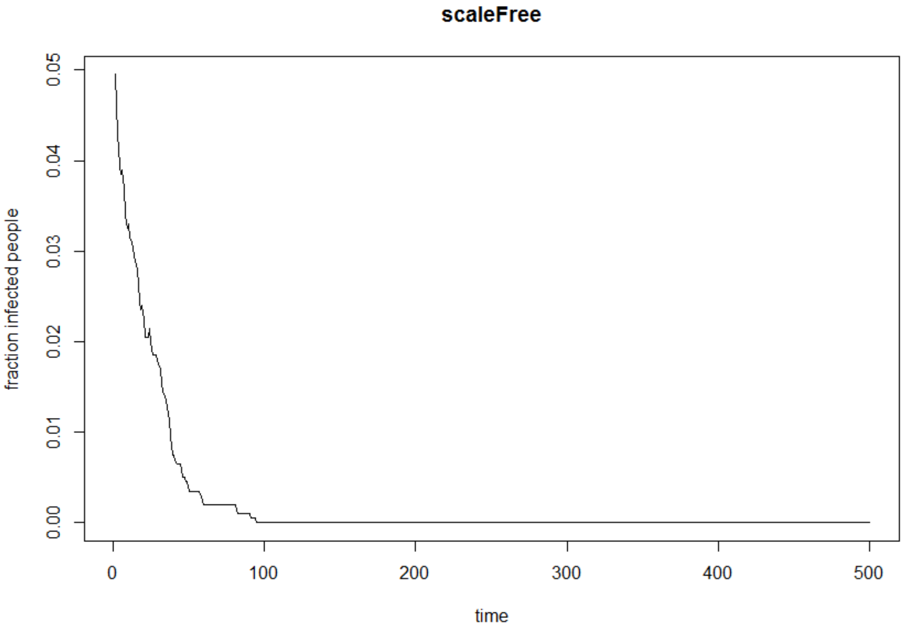


Figure 2: Task 1. Fixed parameters $\gamma = 0.05$ and $\beta = 0.005$

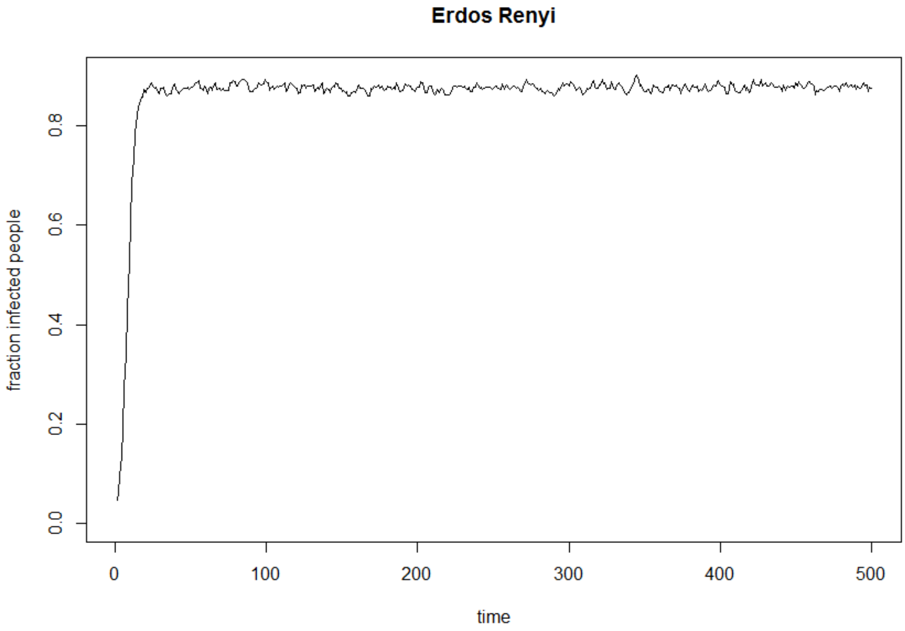


Figure 3: Task 1. Fixed parameters $\gamma = 0.05$ and $\beta = 0.005$

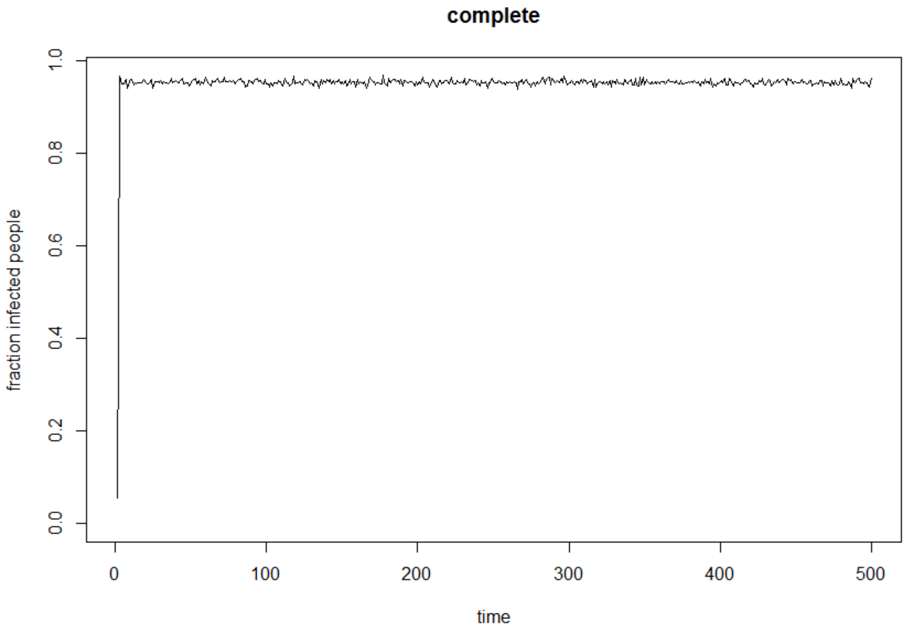


Figure 4: Task 1. Fixed parameters $\gamma = 0.05$ and $\beta = 0.005$

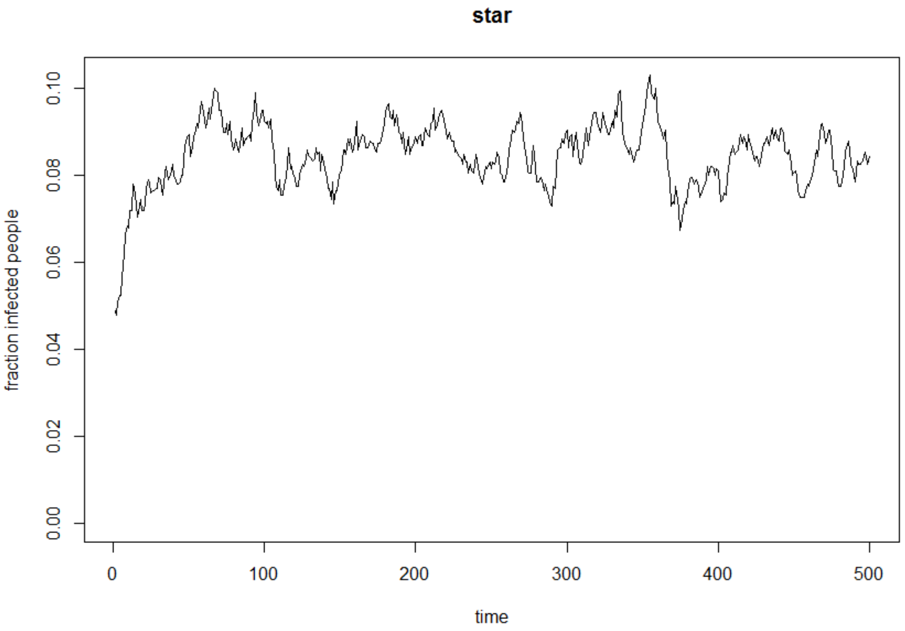


Figure 5: Task 1. Fixed parameters $\gamma = 0.05$ and $\beta = 0.005$

3 Discussion

3.1 Task 1

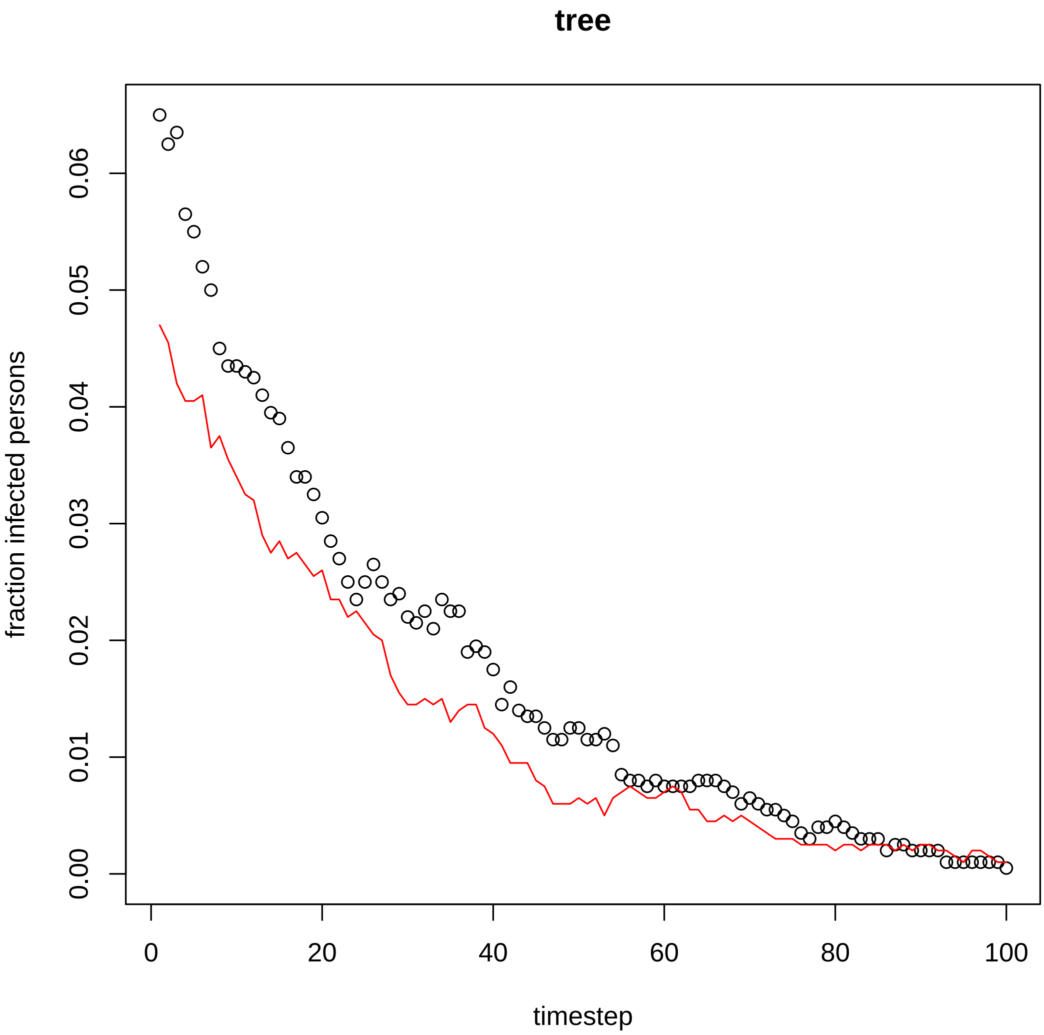
In the simulations with fixed parameters of task 1 there is a big difference between the networks. In the densest ones (Erdös-Renyi, complete graph) the infection spreads very quickly and reaches all the nodes (at every time step we will only have a small fraction of non-infected nodes: those which recovered in the previous step), while in the tree and the scale-free networks the infection decreases until it goes extinct. In the case of the star graph, the number of infected nodes stays at slightly below 10% of the nodes.

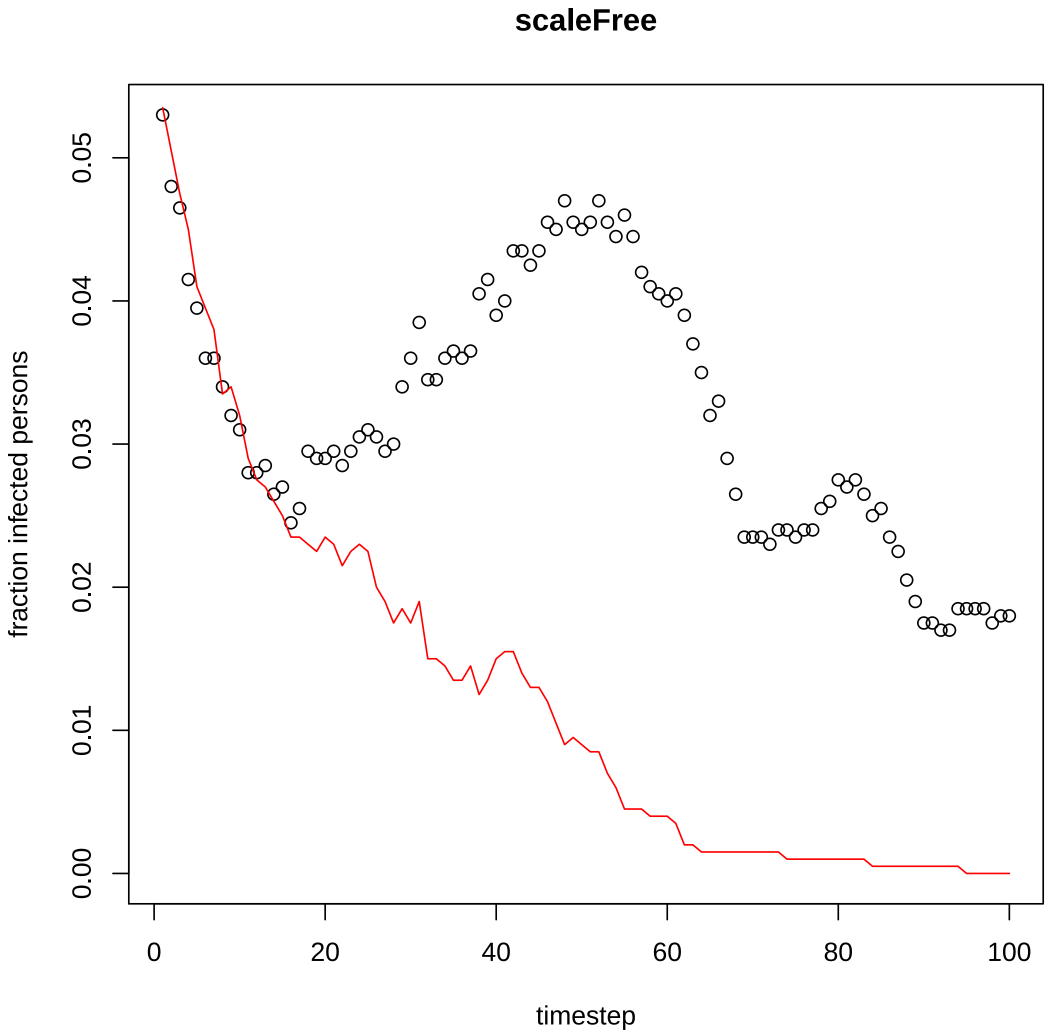
While it is obvious that edge density has an effect on determining whether an infection spreads or goes extinct, notice that the tree and star graphs have the same edge density completely different behaviours. From this, we can conclude that the shape of the network also plays an important role.

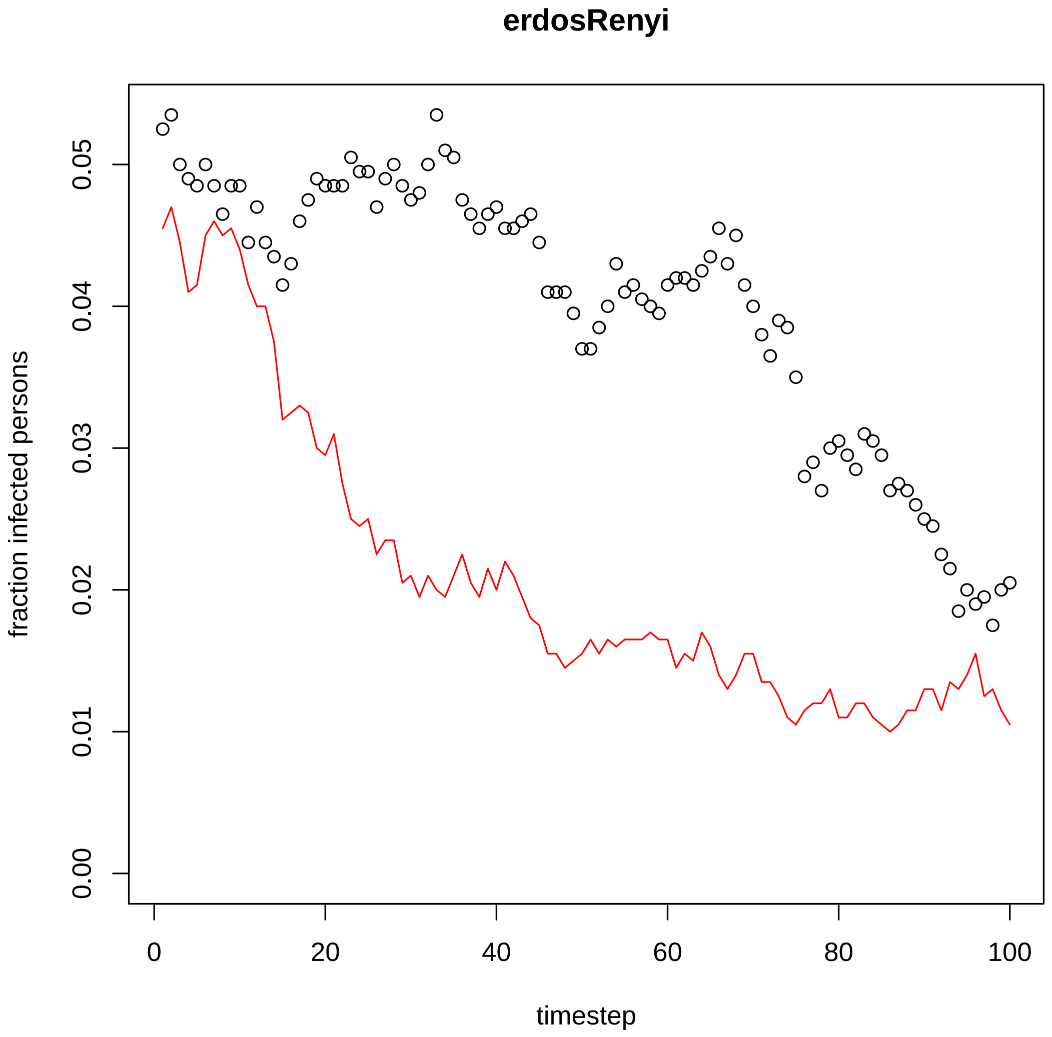
3.2 Task 2

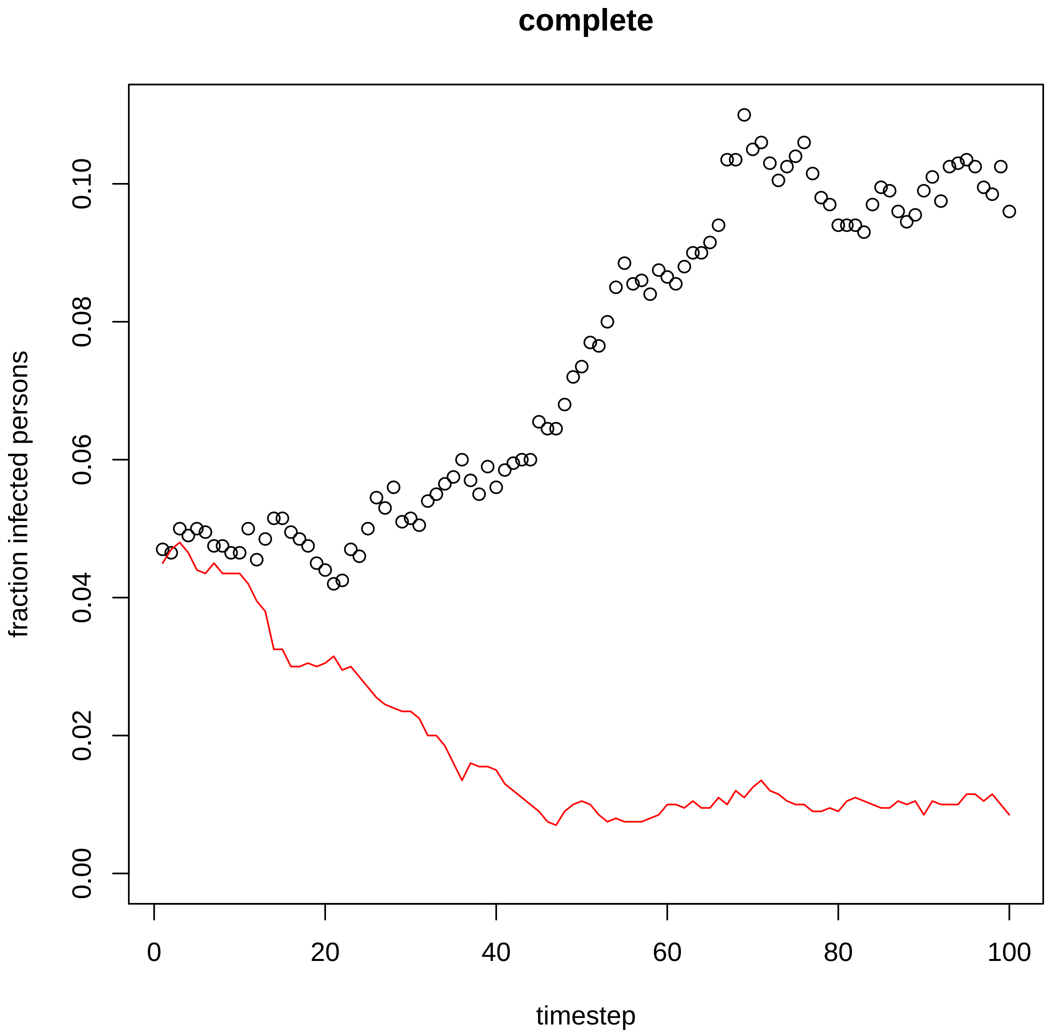
We notice that we see we crossed a threshold value for the complete graph, the Erdos-Renyi graph and the scalefree graph. The star graph, tree, however, do not seem to have crossed any threshold. If we would

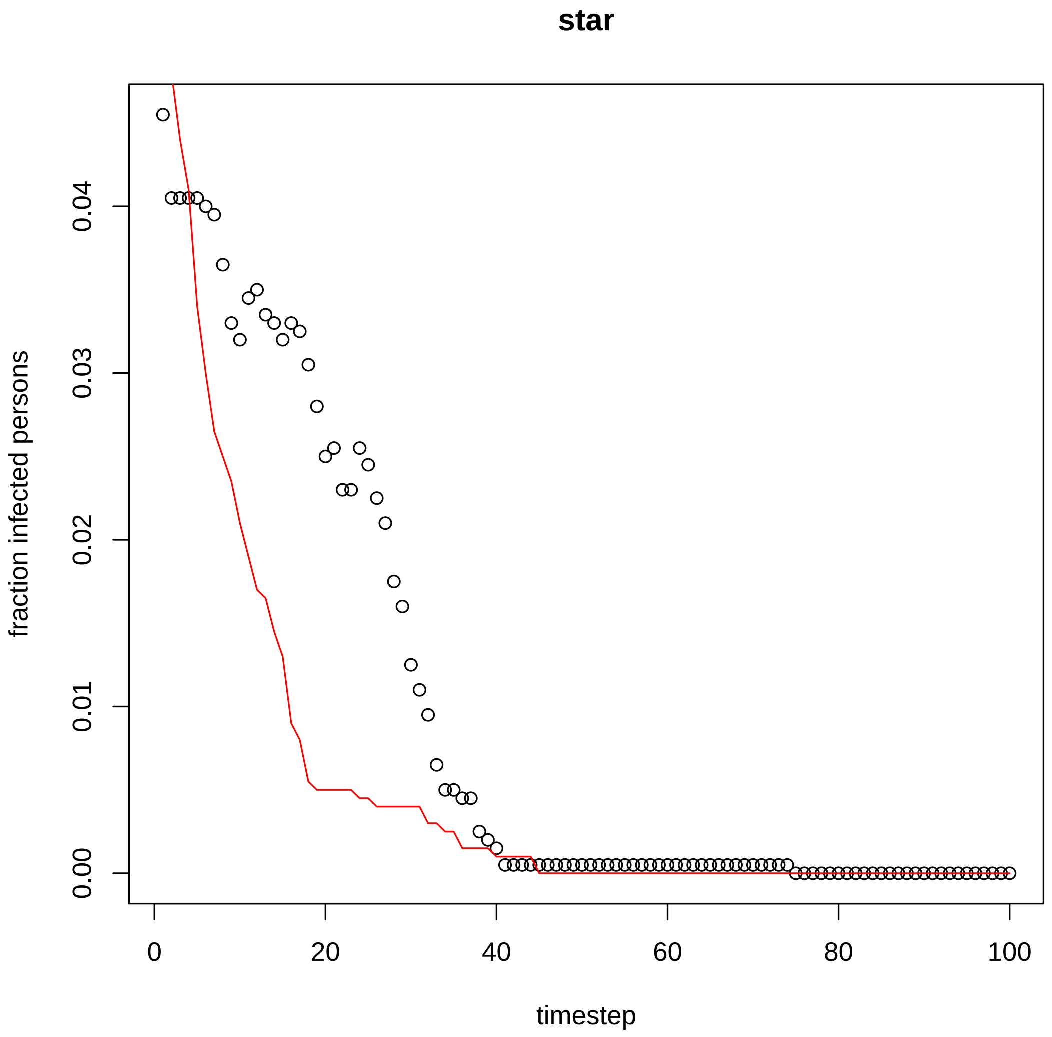
From this, we can conclude that the theory doesn't hold for some graph, in particular it doesn't hold for the trees we tested it on. To be sure it does hold for the complete, Erdos-Renyi and scalefree graph, we would have to do some more tests (values closer


tree
fraction infected persons
0.00
0.01
0.02
0.03
0.04
0.05
0.06
timestep
0
20
40
60
80
100


scaleFree
fraction infected persons
0.00
0.01
0.02
0.03
0.04
0.05
0
20
40
60
80
100
timestep


erdosRenyi
fraction infected persons
0.00
0.01
0.02
0.03
0.04
0.05
timestep
0
20
40
60
80
100


complete
fraction infected persons
0.00
0.02
0.04
0.06
0.08
0.10
0
20
40
60
80
100
timestep


star
fraction infected persons
0.00
0.01
0.02
0.03
0.04
0
20
40
60
80
100
timestep

to the threshold, different numbers of vertices, run the simulation longer), but we are confident there is a threshold here.

4 Methods

To find the thresholds we used ARPACK, a package partially built in in igraph. It has fast methods for calculating the largest eigenvalue of a graph. Calculating the 5 eigenvalues of the 5 graphs with 2000 vertices only takes seconds.

For generating the graphs and plots we set a seed so it can be easily replicated.

We chose the parameters for the Erdos-Renyi and scalefree graph so that they both had about 20000 edges.

4.1 SIS infection simulation

The SIS infection simulation is done by the function `spread`, in `Lab7.R`. This function receives as input the adjacency matrix M of the network, a vector v indicating which nodes are infected at the initial time, the parameters β and γ, and $tmax$, the maximum time for which we will run the simulation.

First, it computes $n_{inf} = Mv$ the vector counting how many infected neighbours each node has. Then we consider a node i to get infected if $n_{inf}[i]$ is strictly bigger than a random sample from a geometric distribution with probability β (the geometric distribution counts how many failures happen before the first "successful" infection. Therefore, if a node has less infected neighbours than this number, it will not get infected). The implementation of this step is very straightforward using the R function `rbinom`.

To simulate which infected nodes get cured, a sample of the binomial distribution with size 1 and probability γ is generated with `rbinom`. Then, we update the states of the nodes as follows: a node is infected if it already was and has not been cured, or if it has been infected by a neighbour.