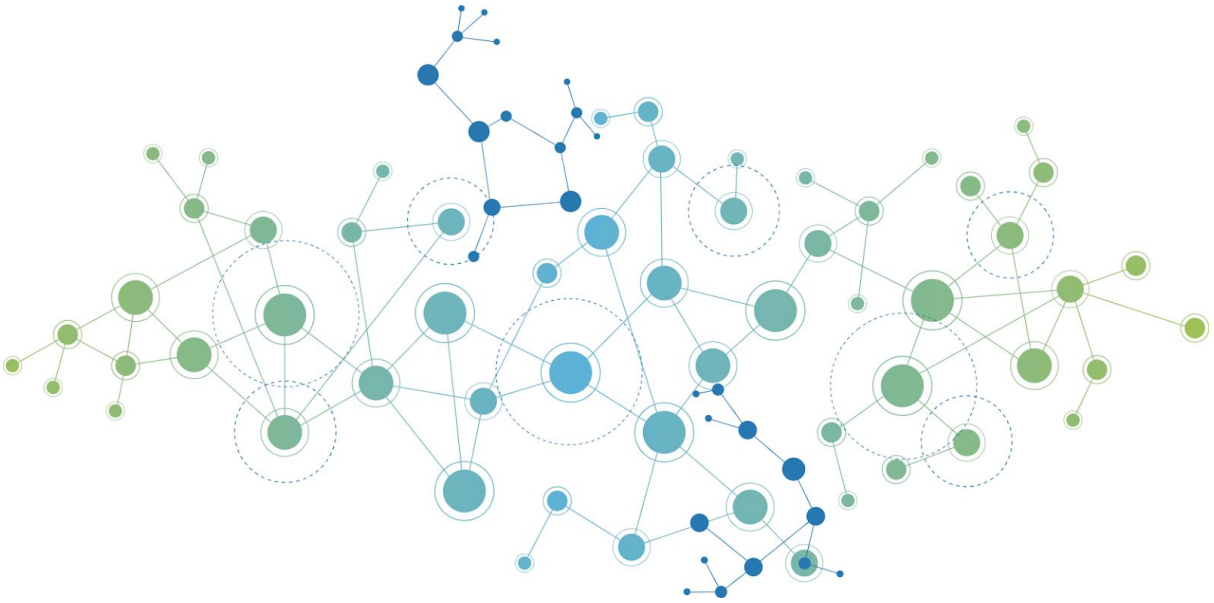
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO



Homework #3

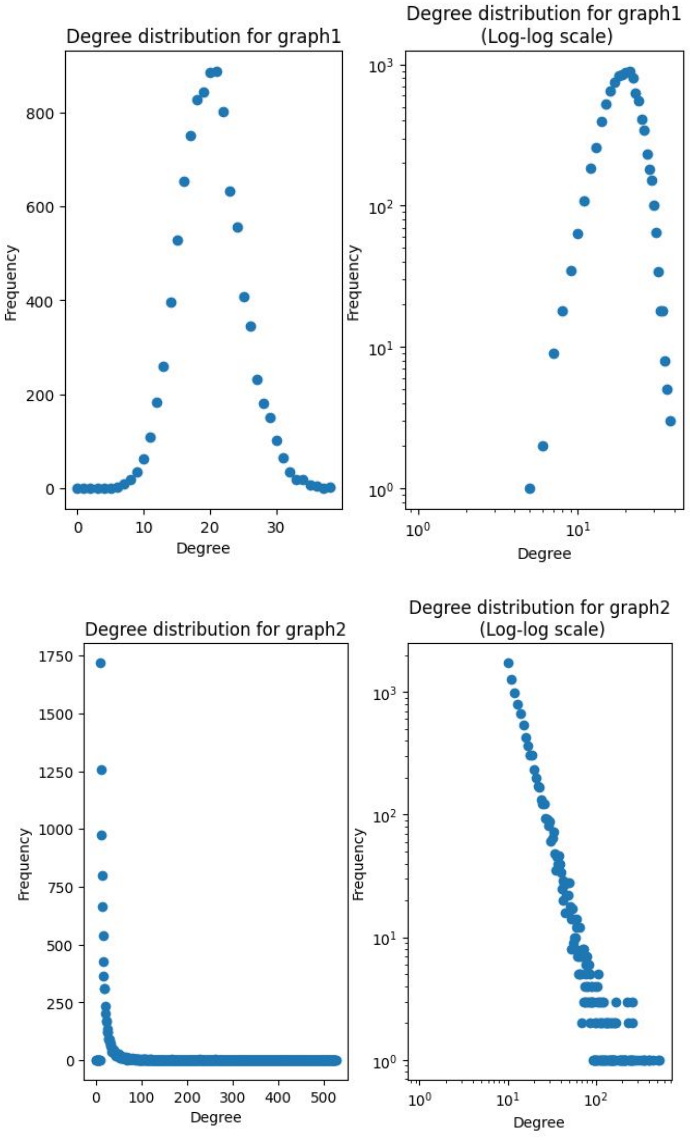
Network Science

Henrique Branquinho **201804341**

May, 2020

1. Diffusion and Cascading Behavior

a) Read the two graphs and plot their degree distributions in linear scale and in log-log scale. Make a brief comment on what distributions you are observing and what graph model might have been used to create each of the graphs.



graph1 appears to have a **normal** degree distribution. I would say this was generated by a Erdos-Renyi model.

graph2 appears to follow a **power law** degree distribution. I would say this was generated by a Barabási-Albert model.

In the folder ex1_2 there is a python script for plotting the degree distribution of a network called **degreePlot.py**. degreePlot.py plots the degree distribution of a graph passed as console parameter:

python3 degreePlot.py graphfile.txt

Stores this plot as graphfile_degree_plot.png.

2. Decision Based Models

a) Basic forecast

Graph 1

A : 4952
B : 5018
Undecided : 30
B wins by 66 votes
Took 5 iterations to converge

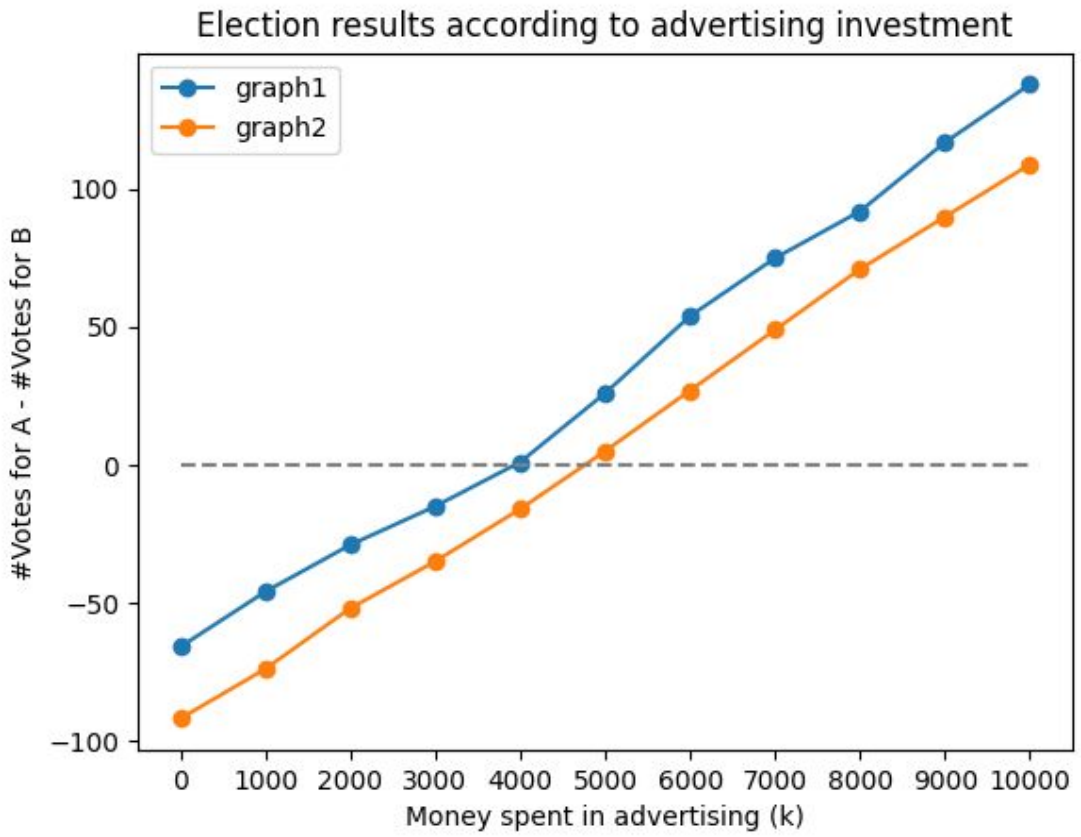
Graph 2

A : 4925
B : 5017
Undecided : 58
B wins by 92 votes
Took 5 iterations to converge

This results were obtained by running the python script **basicForecast.py** in folder ex1_2. Usage:

python3 basicForecast.py network.txt

b) TV advertising



Graph 1

In graph1, A needs to spend 4000 on advertising to win, and wins by 1 vote.

Graph 2

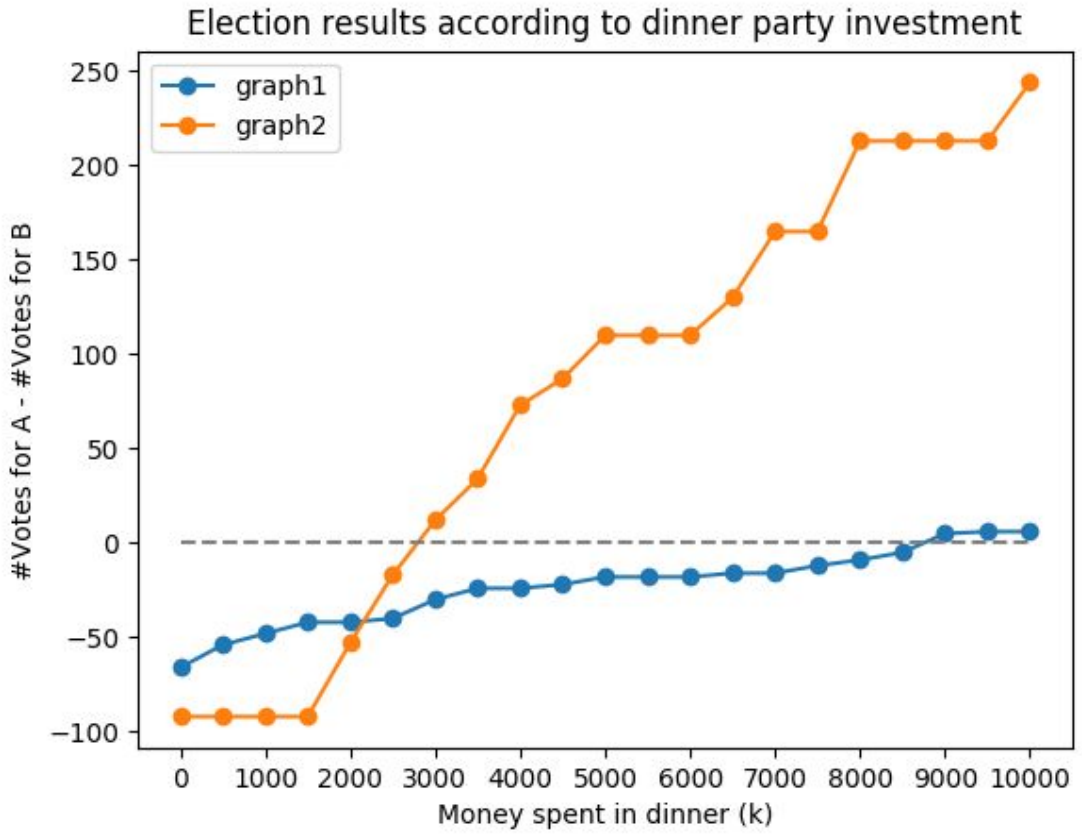
In graph2, A needs to spend 5000 on advertising to win, and wins by 5 votes.

There results were obtained by running the script **tvAdvertising.py** on folder ex1_2. Usage:

python3 tvAdvertising.py graph1.txt graph2.txt

The script takes all networks passed as console parameters, and runs the TV advertising method with values of K from 0 to 10000. After applying the TV advertising effect, runs the rest of the simulation with the **basicForecast** method from the previous exercise script. Outputs results in terminal and saves a plot like the one presented as “tvAdvertising_plot.png”.

c) A fancy dinner



Graph 1

In graph1, A needs to spend 9000 on dinner invitations (inviting 18 voters) to win, and wins by 5 votes.

Graph 2

In graph2, A needs to spend 3000 on dinner invitations to win (inviting 6 voters), and wins by 12 votes.

In graph1 (normal degree distribution), high degree nodes tend to not have that high of a degree (max degree ~= 40, 10000 nodes). In practical terms, this means they don't have many friends, so they won't be able to influence many people. That is why A needs to invite a lot of voters (18) to win the election. This is noticeable in the plot as A's advantage increases very slowly with increasing dinner invitations.

In graph2 (power law degree distribution), we have few nodes with high degree (max degree ~= 500, 10000 nodes), but their degree is considerably high. In practical terms, this means that there are few but heavily influential people in this network, so A only needs to invite a few voters (6) to win the election. Since these voters have many friends, they will be quick to spread the word of A's greatness. This is noticeable in the plot as A's advantage increases very rapidly with increasing dinner invitations.

There results were obtained by running the script **fancyDinner.py** on the folder ex1_2. Usage:

python3 fancyDinner.py graph1.txt graph2.txt

The script maintains a list of sorted nodes by decreasing degree, so that it can pick the K-highest degree nodes when running simulations for each value of k€. After applying the effect of the dinner, runs basicForecast from the previous script to complete the simulation.

3. Probabilistic Models

The simulations ran to answer the following questions were ran by the script **covidSimulation.py** in the folder ex3. Usage:

python3 covidSimulation.py graphfile probInfection probRecovery n0Infected kRepetitions gammaTesting(Y/N)

Where graphfile is the network file, probInfection is β, probRecovery is δ, n0Infected is the number of random infected nodes at the start of the simulation, kRepetitions is the number of repetitions to be ran to get mean values, and gammaTesting sould be "Y" if we are running the vaccination simulation for exercise 3.e).

Graph nodes can have 3 states: “S” for susceptible, “I” for infected and “R” for recovered. Initially, n0Infected nodes are chosen and set to “I” state. Then, for each day of the simulation, there is virus propagation as described in the problem description.

Each simulation returns 3 lists of 101 elements each, corresponding to the number of susceptible, infected and recovered nodes at each day of the simulation, and two values corresponding to R0 at day 6 and estimated R at peak infection day.

Then, after K simulations, 3 new lists similar to these are calculated by averaging the K values on each simulation day, from each simulation.

These final values are then plotted, with standard deviation on each point.

The script prints a log for each simulation day with statistics, and prints the mean values at the end, along with some information used to answer the following questions.

Examples of logs (small prints):

```
---------------------- K = 1 ---------------

Day 0   #Susceptible = 9995     #Infected = 5 (+0)      #Recovered = 0 (+0)
Day 1   #Susceptible = 9993     #Infected = 7 (+2)      #Recovered = 0 (+0)
Day 2   #Susceptible = 9977     #Infected = 22 (+16)    #Recovered = 1 (+1)
Day 3   #Susceptible = 9928     #Infected = 68 (+49)    #Recovered = 4 (+3)
Day 4   #Susceptible = 9787     #Infected = 203 (+141)  #Recovered = 10 (+6)
Day 5   #Susceptible = 9472     #Infected = 506 (+315)  #Recovered = 22 (+12)
Day 6   #Susceptible = 8756     #Infected = 1191 (+716)         #Recovered = 53 (+31)
Day 7   #Susceptible = 7626     #Infected = 2242 (+1130)        #Recovered = 132 (+79)
Day 8   #Susceptible = 6181     #Infected = 3520 (+1445)        #Recovered = 299 (+167)
Day 9   #Susceptible = 4727     #Infected = 4744 (+1454)        #Recovered = 529 (+230)
Day 10  #Susceptible = 3432     #Infected = 5684 (+1295)        #Recovered = 884 (+355)
Day 11  #Susceptible = 2481     #Infected = 6192 (+951)         #Recovered = 1327 (+443)
```

```
----------------------MEAN VALUES---------------

Day 0   #Susceptible = 9995     #Infected = 5 (+0)      #Recovered = 0
Day 1   #Susceptible = 9990     #Infected = 8 (+4)      #Recovered = 0
Day 2   #Susceptible = 9978     #Infected = 20 (+12)    #Recovered = 1
Day 3   #Susceptible = 9942     #Infected = 54 (+35)    #Recovered = 2
Day 4   #Susceptible = 9853     #Infected = 139 (+89)   #Recovered = 6
Day 5   #Susceptible = 9634     #Infected = 350 (+219)  #Recovered = 15
Day 6   #Susceptible = 9178     #Infected = 783 (+456)  #Recovered = 38
Day 7   #Susceptible = 8340     #Infected = 1564 (+838)         #Recovered = 95
Day 8   #Susceptible = 7146     #Infected = 2649 (+1193)        #Recovered = 204
Day 9   #Susceptible = 5738     #Infected = 3873 (+1407)        #Recovered = 388
Day 10  #Susceptible = 4339     #Infected = 4995 (+1399)        #Recovered = 665
Day 11  #Susceptible = 3155     #Infected = 5812 (+1183)        #Recovered = 1031
```

```
Peak number of infected nodes = 6531.8 at day 13
Peak number of NEW infected nodes = 1407.3 at day 9
R0 at day 6 = 4.77534266939597
Estimated R at day of peak infected nodes = 0.1878469978213365
```

To answer the following questions, I ran 10 simulations to get my results.

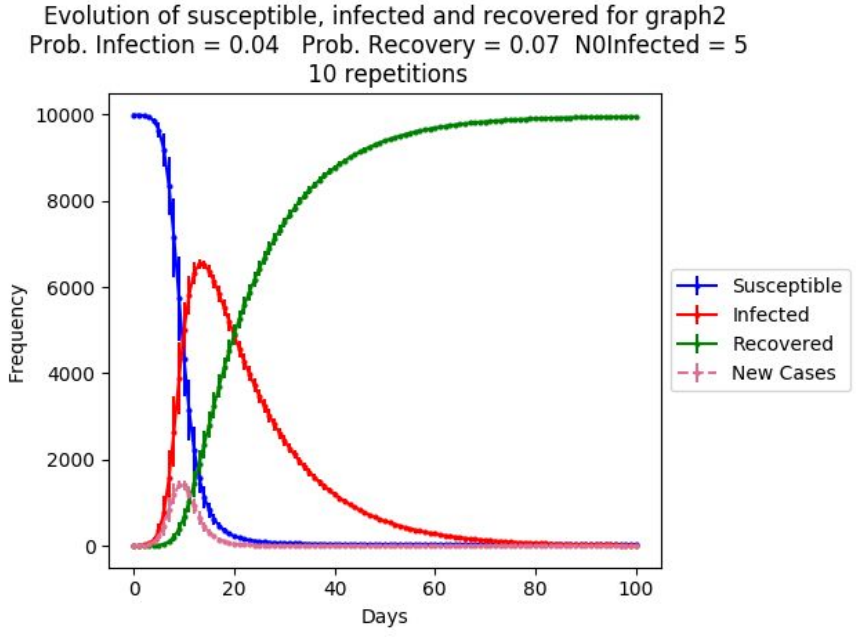
The log files corresponding to these simulations are present in the folder ex3.

The script also creates two plots and saves them as “network_xxx_plot.png” and “network_xxx_newcases_plot.png” where “xxx” is the current time to differentiate between different plots for different simulations.

If the vaccinated option is selected when running the script, the only values that are used correspond to the number of total infected nodes in each simulation, as it would be hard for me to get mean values to present since each simulation could take a variable number of days to converge. In this case, only one plot is created, and it is saved as “network_xxx_vaccination_plot.png”.

To simulate the vaccination, the script runs K simulations for each value of *gamma* (from 0% to 100% with jumps of 5%).

a) Basic plot



The graph represents what I was expecting.

The graph displays a traditional evolution of a SIR model. By looking at the class material:

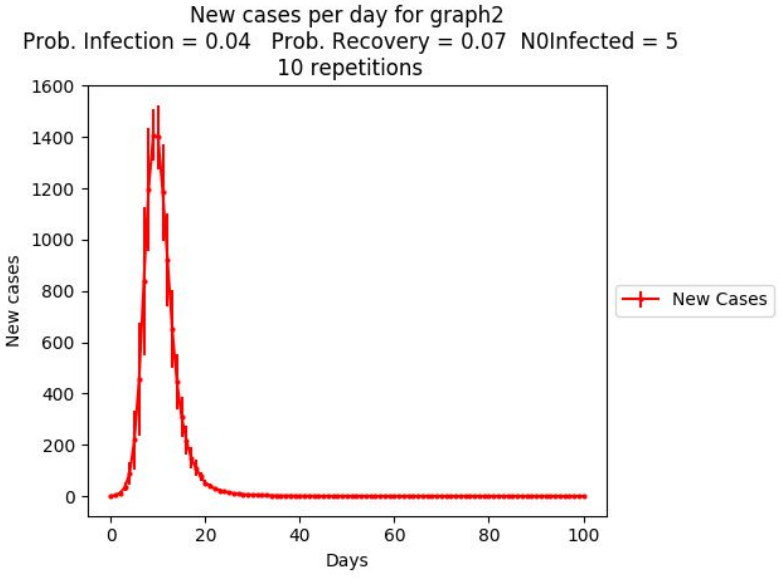
$$\frac{dS}{dt} = -\beta SI \qquad \frac{dR}{dt} = \delta I$$

$$\frac{dI}{dt} = \beta SI - \delta I$$

Susceptible numbers initially decrease very slowly (as I is still small), but the infection growth will rapidly increase (as it is positive and depends on I, which is small at first but starts to grow and is not initially reduced by -δI because δ is relatively small and I is also small), which means the Susceptible numbers will start to rapidly decrease (which is noticeable on the graph). Only after 14 days do we begin to observe a decrease in Infected nodes (due to increasing recovery cases and almost no susceptible nodes left, makes sense because the expected number of days with the infection is 14). Recovered nodes begin to rapidly increase with some delay regarding the infected nodes due to the recovery time, and then increase slowly as there are fewer and fewer infected nodes.

Peak number of infected nodes : 6531.8 at day 13.

b) Basic statistics



Peak number of NEW infected nodes = 1407.3 at day 9

This peak is before the peak of infected nodes (around 4 days earlier). It is plausible that the peak of new infected is before the peak of infected nodes because this peak will help increase the growth of infected nodes, that along with the already infected nodes that are not yet recovered, will only peak a bit after, with the accumulation of more cases.

The number of new infected nodes (and infected nodes) only peak once because of the non reversible transition states of this model. Nodes can only go from Susceptible to Infected to Recovered. Initially, the infection spreads quick because most nodes are susceptible, and as more nodes get infected, the more nodes they infect. However, the spread reaches a point where most nodes were already infected or are starting to recover, so the virus has “nowhere to go”, and the numbers start to decrease after that. Also, the topology of the network can also be taken into account, as a power law model, because most nodes will not have many neighbour nodes to infect and it is likely that the high degree nodes will get infected early, meaning they will be responsible for the bigger part of the infection spread early on, and after that the remaining nodes will not be able to infect many people due to their low degree.

c) Estimating R_0

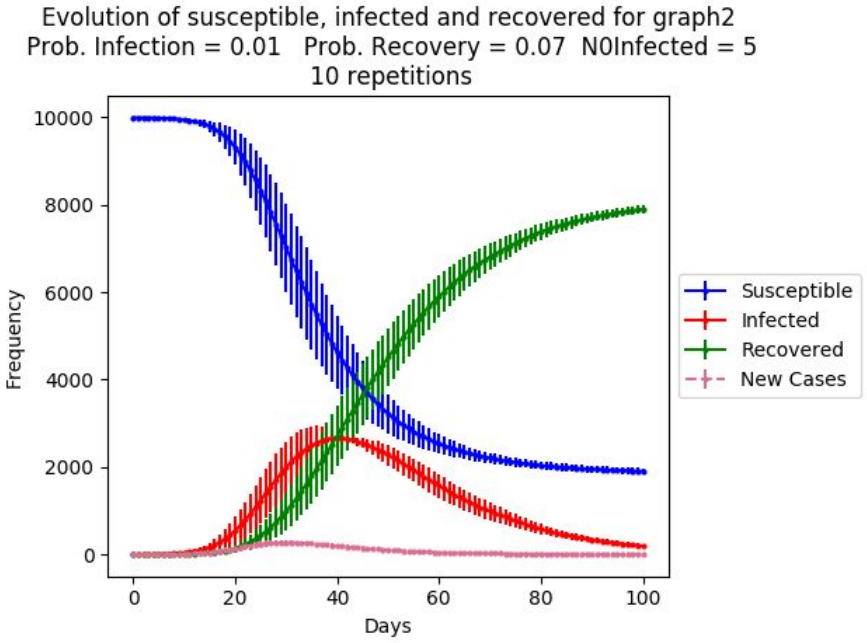
R_0 at day 6 = 4.775

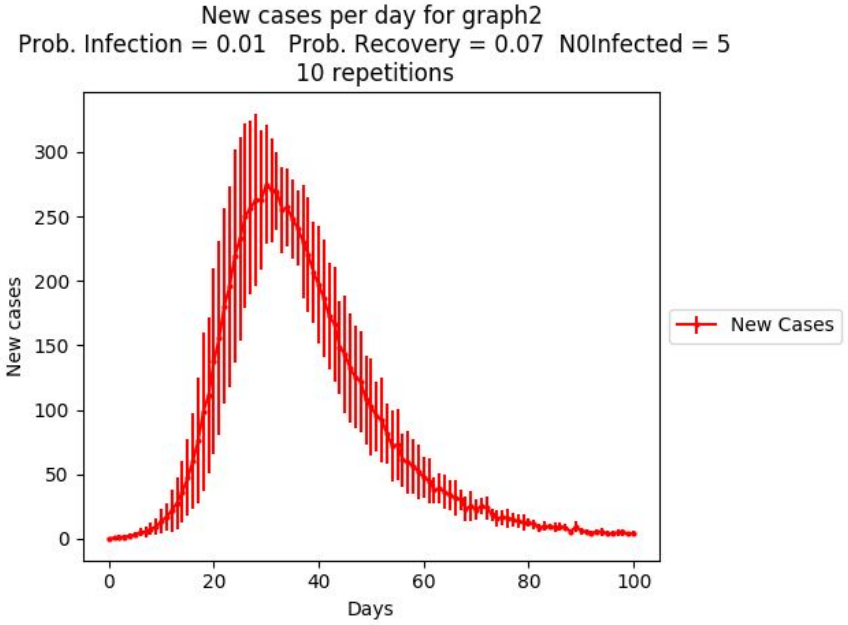
On day 6, R_0 is greater than 4, which is in agreement with the model’s outcome. This means that every person infects more than four people on average, which helps explaining the rapid growth of infected nodes.

Estimated R at day of peak infected nodes = 0.1878

After the infection peak R is smaller than 1, which means that there are less people getting infected than those already infected, meaning the virus is dying out, because the infected people will recover and they are infecting less and less people over time.

d) Lowering the curve





This new plot has some interesting differences from the previous plot.

First of all, the total number of infected nodes is smaller than before (high number of “susceptible” nodes at day 100, meaning they didn’t get infected).

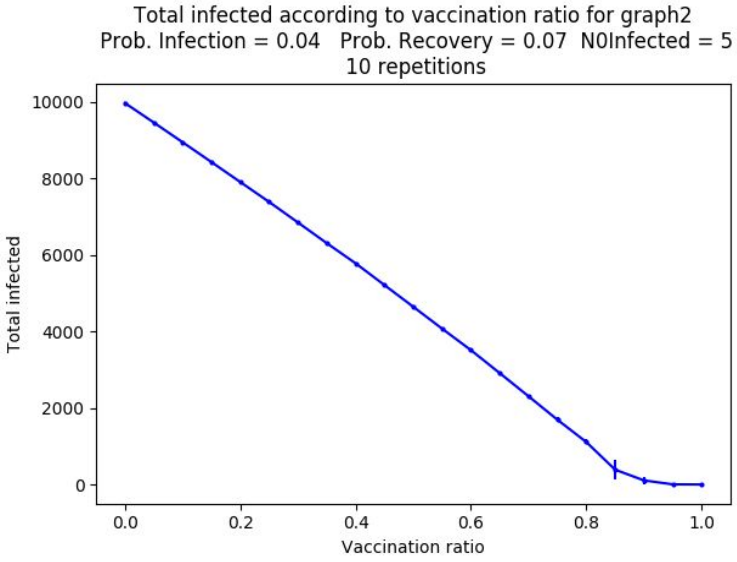
Also, the growth of infected nodes was more gradual,with a later peak day, corresponding to fewer infected nodes than before (2663.2 at day 40 instead of 6531.8 at day 13). We can also notice that the number of new infected nodes per day is pretty much stable throughout the simulation.

R0 at day 6 = 2.83599

The new R_0 at day 6 is smaller than before (almost half) meaning that each person will infect on average 2 to 3 people , instead of 4 or more.

This graphic is indicative of a much more controlled spread of the virus. Infection spreads slowly and ends up not spreading to everyone (~ 20% of nodes don't get infected at all). This behaviour shows the meaning of the expression "flattening the curve" that we hear about so much. With this kind of behaviour, healthcare systems have a smaller influx of people over time, meaning they can probably attend to every infection case, unlike in the previous case, where infection grows rapidly in the first days and with a very high peak of infected nodes, which could translate into not enough medical resources to treat so many people (like we saw in some countries with the COVID-19).

e) Effect of vaccines



This plot shows that vaccination is very important to contain the spread of infectious diseases. The number of infected nodes seems to be proportional to the vaccination ratio, which means that to achieve **herd immunit**, most nodes need to be vaccinated (total infected stabilize near 0 after 95% vaccination ratio).

4. Network Construction

a) A first network

i) Basic statistics

Number of nodes: 919
Number of edges: 1353
Avg. degree: 2.945
Avg. weighted degree: 0.696
Number of connected components: 3
Size of the giant component: 911 (99.1% of nodes)

ii) Giant component statistics

Diameter: 8
Avg. path length: 4.196
Top 3 nodes with highest degree:

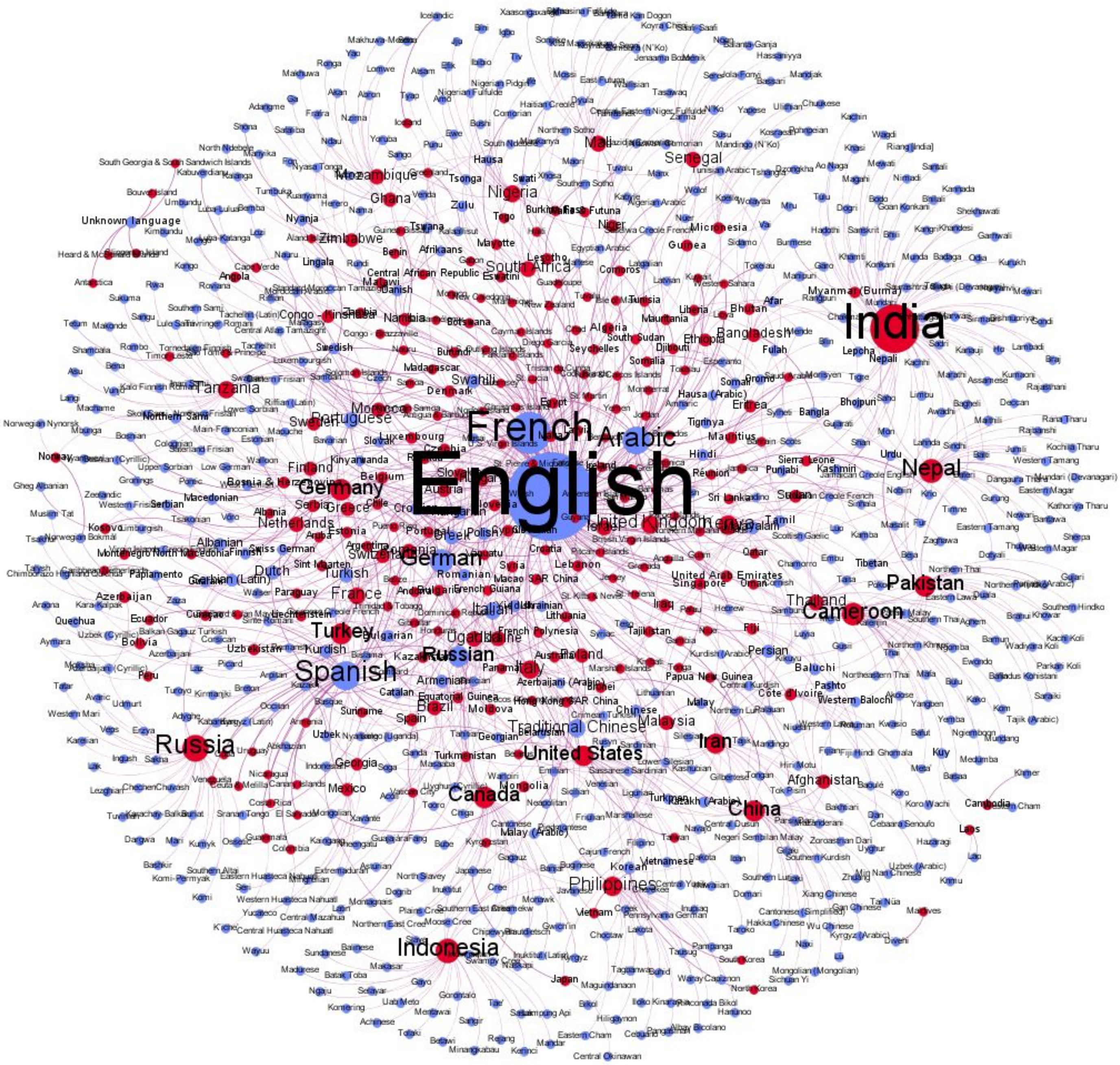
- English (language) -> 147
- India (country) -> 74
- French (language) -> 61

Top 3 nodes with highest pagerank (beta = 0.85, epsilon = 0.001 , weighted):

- English (language) -> 0.09664
- French (language) -> 0.031
- Spanish (language) -> 0.0212

iii) Image depicting the network

In blue, languages.
In red, countries.

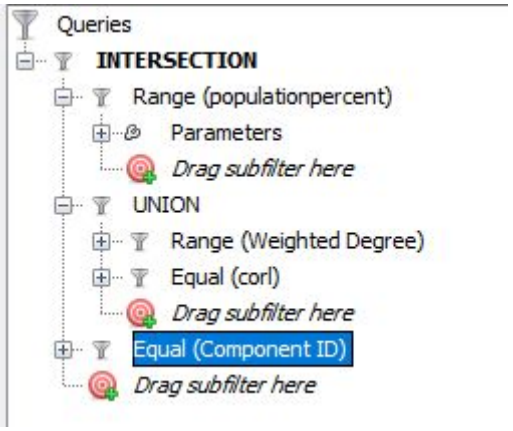


b) Filtering the network

Nodes: 351

Edges: 443

To apply this filter, I needed to copy the “population amount” column to the “Weight” column in order to use weighted degree as a filter. I first filtered the edges that corresponded to less than 10% of the country population. Then I filtered the languages spoken by less than 10 million people (I did this after because this is influenced by the edges; Equal (corl) means that the nodes are countries, I perform a union with this filter so that I don’t end up removing nodes that are countries with less than 10 million people), then I calculated the number of components, and then I added the condition for nodes in the giant component.



c) Projecting on one mode

Nodes: 253
Edges: 13624
No. of communities: 8

I performed the projection having only in account the number of common languages and discarding the information about percentage of speakers.

Modularity Class 0:

- Germany (0.011)
- United States (0.010)
- Romania (0.009)
- France (0.009)
- United Kingdom (0.008)

Important languages (English, French and German) are present here in all of these countries

Modularity Class 1:

- Antarctica (0.004)
- Bouvet Island (0.004)
- Clipperton Island (0.004)
- Heard & McDonald Islands
- South Georgia & South Sandwich Islands (0.004)

Modularity Class 2:

- Greenland (0.0009)
- Iceland (0.0008)

They share a language (Danish)

Modularity Class 3:

- Maldives (0.0006)

Modularity Class 4:

- Portugal (0.0077)
- Philippines (0.0059)
- Panama (0.0059)
- Sint Maarten (0.0057)
- Spain (0.0055)

Modularity Class 5:

- Angola (0.0012)
- Guinea-Bissau (0.0012)
- Cape Verde (0.0011)
- São Tomé & Príncipe (0.0011)
- Timor-Leste (0.0011)

Countries with Portuguese influence

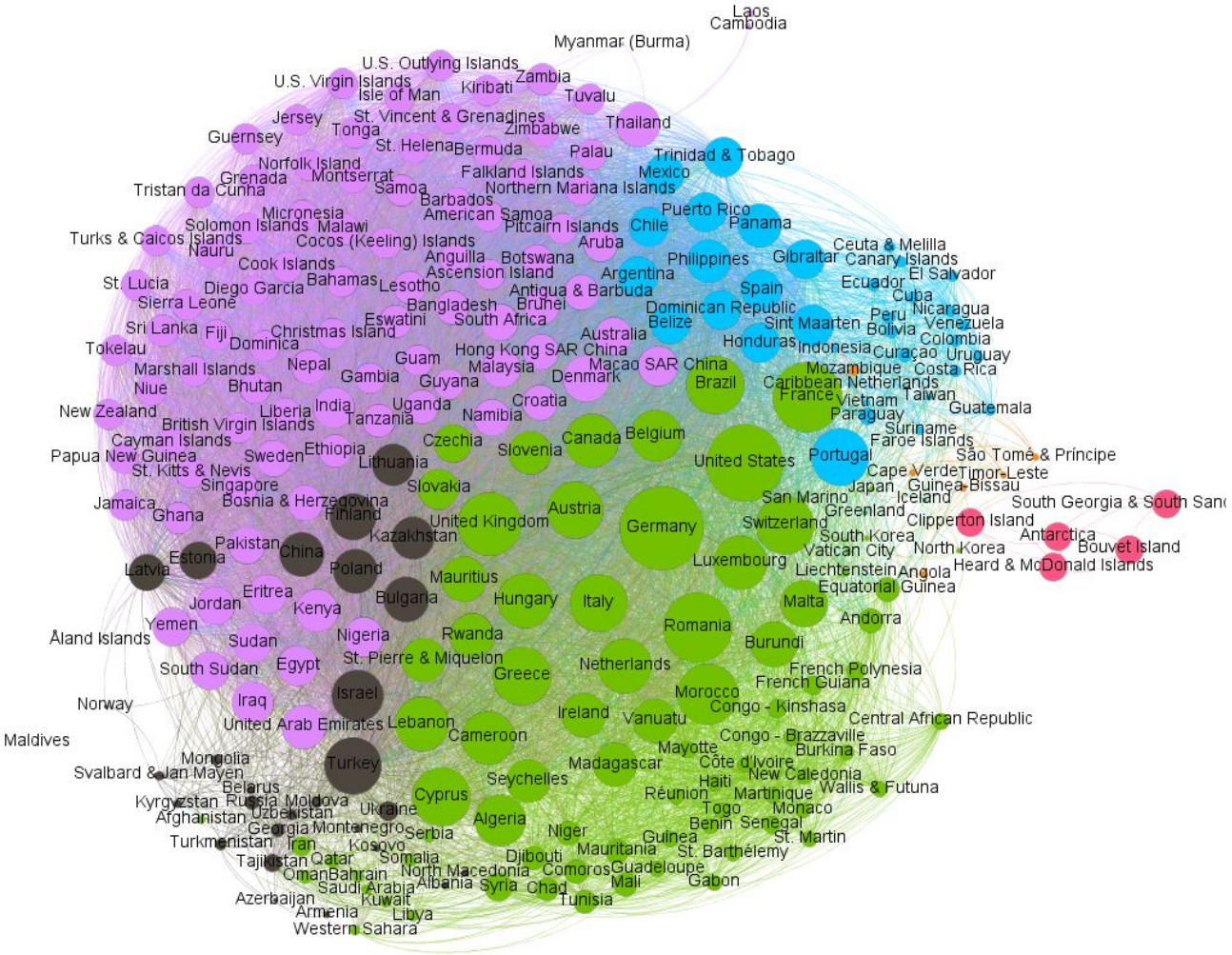
Modularity Class 6:

- Denmark (0.0064)
- Thailand (0.0060)
- United Arab Emirates (0.0060)
- Kenya (0.0058)
- Iraq (0.0057)

Modularity Class 7:

- Turkey (0.0077)
- Israel (0.0070)
- Finland (0.0063)
- Bulgaria (0.0061)
- Poland (0.0061)

Eastern european countries except for Israel, may have some Russian influences


Laos
Cambodia
Myanmar (Burma)
U.S. Outlying Islands
Zambia
U.S. Virgin Islands
Kiribati
Isle of Man
Tuvalu
Jersey
St. Vincent & Grenadines
Thailand
Tonga
Zimbabwe
Guernsey
St. Helena
Bermuda
Palau
Trinidad & Tobago
Norfolk Island
Montserrat
Falkland Islands
Mexico
Grenada
Tristan da Cunha
Samoa
Northern Mariana Islands
Micronesia
Barbados
American Samoa
Puerto Rico
Solomon Islands
Malawi
Pitcairn Islands
Chile
Panama
Turks & Caicos Islands
Cocos (Keeling) Islands
Aruba
Ceuta & Melilla
Nauru
Anguilla
Botswana
Philippines
Gibraltar
Canary Islands
Cook Islands
Ascension Island
Argentina
El Salvador
St. Lucia
Bahamas
Lesotho
Ecuador
Diego Garcia
Antigua & Barbuda
Spain
Cuba
Sierra Leone
Bangladesh
Brunei
Dominican Republic
Nicaragua
Sri Lanka
Fiji
Eswatini
South Africa
Belize
Sint Maarten
Peru
Venezuela
Christmas Island
Australia
Bolivia
Tokelau
Dominica
Honduras
Colombia
Hong Kong SAR China
Indonesia
Curaçao
Uruguay
Marshall Islands
Nepal
Guam
Malaysia
Macao SAR China
Mozambique
Costa Rica
Gambia
Denmark
Niue
Bhutan
Guyana
Brazil
Caribbean Netherlands
France
Taiwan
New Zealand
India
Liberia
Uganda
Croatia
Vietnam
Guatemala
Tanzania
Namibia
Paraguay
British Virgin Islands
Suriname
Cayman Islands
Czechia
Canada
Belgium
Faroe Islands
Papua New Guinea
Ethiopia
Slovenia
Portugal
São Tomé & Príncipe
Sweden
United States
St. Kitts & Nevis
Lithuania
Cape Verde
Timor-Leste
Singapore
Slovakia
Japan
Guinea-Bissau
Jamaica
Bosnia & Herzegovina
San Marino
Iceland
South Georgia & South Sand
Austria
Ghana
Finland
Germany
Greenland
Clipperton Island
United Kingdom
Switzerland
Antarctica
Pakistan
Kazakhstan
South Korea
Estonia
China
Vatican City
North Korea
Bouvet Island
Luxembourg
Latvia
Poland
Liechtenstein
Angola
Heard & McDonald Islands
Mauritius
Equatorial Guinea
Eritrea
Bulgaria
Hungary
Italy
Malta
Jordan
Kenya
Andorra
Yemen
Romania
Åland Islands
Nigeria
Rwanda
Burundi
Sudan
St. Pierre & Miquelon
French Polynesia
South Sudan
Egypt
Greece
Netherlands
French Guiana
Morocco
Israel
Norway
Iraq
Congo - Kinshasa
Ireland
Vanuatu
Central African Republic
United Arab Emirates
Lebanon
Cameroon
Congo - Brazzaville
Maldives
Mayotte
Burkina Faso
Mongolia
Turkey
Madagascar
Côte d'Ivoire
Svalbard & Jan Mayen
Seychelles
New Caledonia
Haiti
Belarus
Réunion
Martinique
Wallis & Futuna
Kyrgyzstan
Russia
Moldova
Cyprus
Ukraine
Togo
Monaco
Afghanistan
Uzbekistan
Algeria
Benin
Senegal
Georgia
Montenegro
Serbia
Niger
St. Martin
Turkmenistan
Iran
Kosovo
Guinea
Mauritania
St. Barthélemy
Tajikistan
Qatar
Somalia
Djibouti
Oman
Bahrain
North Macedonia
Comoros
Guadeloupe
Saudi Arabia
Albania
Syria
Chad
Mali
Gabon
Azerbaijan
Kuwait
Tunisia
Armenia
Libya
Western Sahara