Problem 1

Modeling an SIR epidemic on a k-regular symmetric undirected graph:

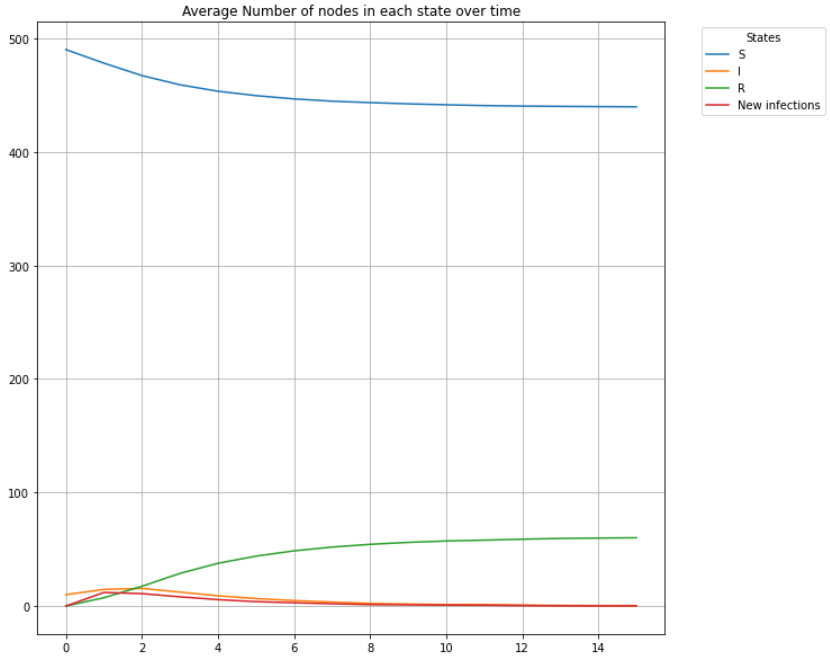
Given configuration parameters:

- each node is connected to the k nodes whose index is closest to their own modulo *n*.
- a susceptible node *i* has *m* infected neighbor
- parameters: $|V| = 500$, $k = 4$, $\beta = 0.3$, $\rho = 0.7$
- transition probabilities:
    - $P(X_i(t + 1) = I \mid X_i(t) = S) = 1 - (1 - \beta)^m$
    - $P(X_i(t + 1) = R \mid X_i(t) = I) = \rho$

we start by defining our graph. Since there is a constrain on the connection of neighbors, we cannot simply draw the graph using automated tool provided by **`networkx`**. However, since what we need to continue our calculations is the adjacency matrix of the graph, we can construct it by generating each row in a fashion that secures connection between a node and all 4 closest one to make the lowest modulo. Then we shift the elements for the proceeding rows.

Next is to insert an initial configuration where *i* random nodes are **Infected (*I*)** and the other *n* - *i* are **Susceptible (*S*)**, we simulate the epidemic model for *t* time units. This means that, at each time an *S* node can be infected with a probability *P*(*I*) where *m* is the number of infected neighbors of the node. We directly use this to define our infection spread module where this enables us to proceed with simulation without the need to compute the Lambda function. The initial infected nodes will be set to 10 random nodes which would have a probability [0,1] of infecting their susceptible neighbors. Since our assumption is to employ an SIR model there is a chance of recovery at each time frame assigned with ρ.

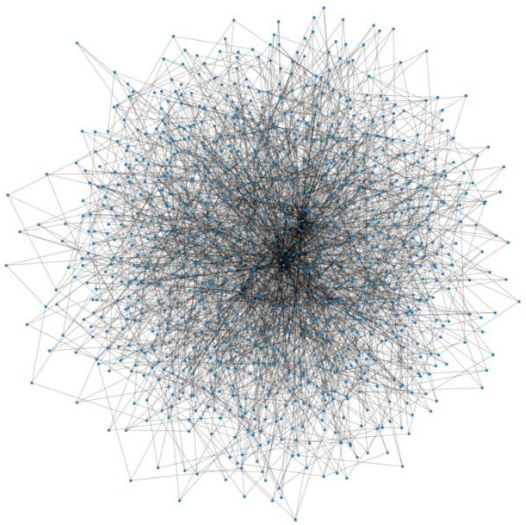
We simulate the epidemic graph with above mentioned parameters. We then perform this simulation for $N = 100$ times, recording data during each simulation, focusing number of nodes per state and the number of *susceptible* nodes that become *infected* at each time step. Average behavior over the 100 simulations is as follows:



For the second part of the exercise, we are going to generate a random graph using preferential attachment model. We start building by generating a complete graph with k+1 node and we add nodes with degree $c = k/2$ (alternating between floor(k/2) and *ceil(k/2)* in case *k* is odd) until we reach the number of desired nodes *n*. For each node we add to the graph we link it to *c* already present nodes, with probability *P*, if we denote the new node *nt*, the probability that there will be a link between node *nt* and node *i is*:

$$\mathbb{P}(W_{n_t,i}(t) = W_{i,n_t}(t) = 1 \mid \mathcal{G}_{t-1} = (\mathcal{V}_{t-1}, \mathcal{E}_{t-1})) = \frac{w_i(t-1)}{\sum_{j \in \mathcal{V}_{t-1}} w_j(t-1)}, \qquad i \in \mathcal{V}_{t-1},$$

By varying the degree of the graph (k) we generate different random graph. In our case we generated it with k=6 and n=1000.



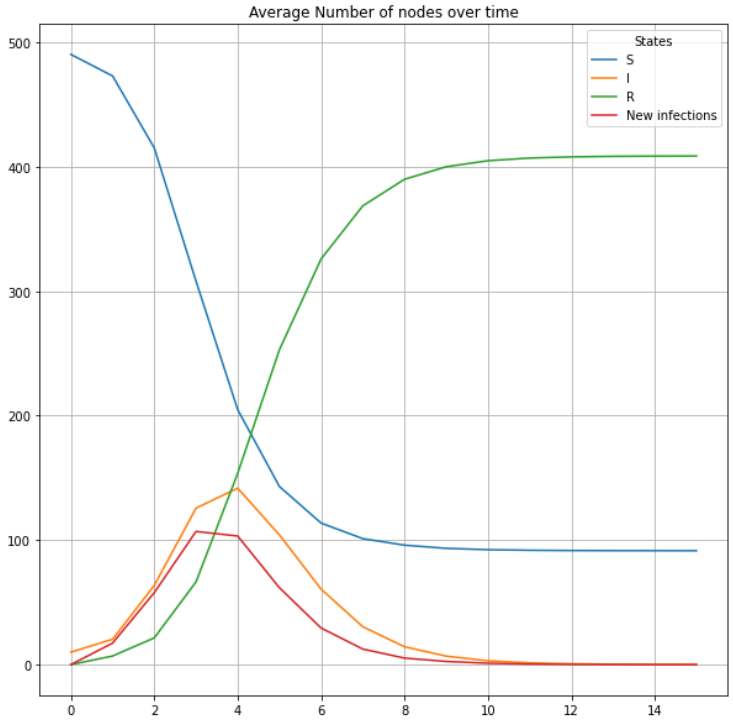
Problem 2

Modeling an SIR epidemic on a random graph Built with <u>preferential attachment</u>*:*

Given configuration parameters:

- Graph generated with previous model with k=6 n=500
- a susceptible node *i* has *m* infected neighbor
- parameters: $\beta = 0.3$, $\rho = 0.7$
- transition probabilities:
    - $P(X_i(t+1) = I \mid X_i(t) = S) = 1 - (1 - \beta)^m$
    - $P(X_i(t+1) = R \mid X_i(t) = I) = \rho$

The procedure is exactly as Problem 1. starting with 10 random infected nodes, and simulate 15 steps for $N = 100$ simulations, yet this time we perform it on a random generated graph with PA.



Problem 3

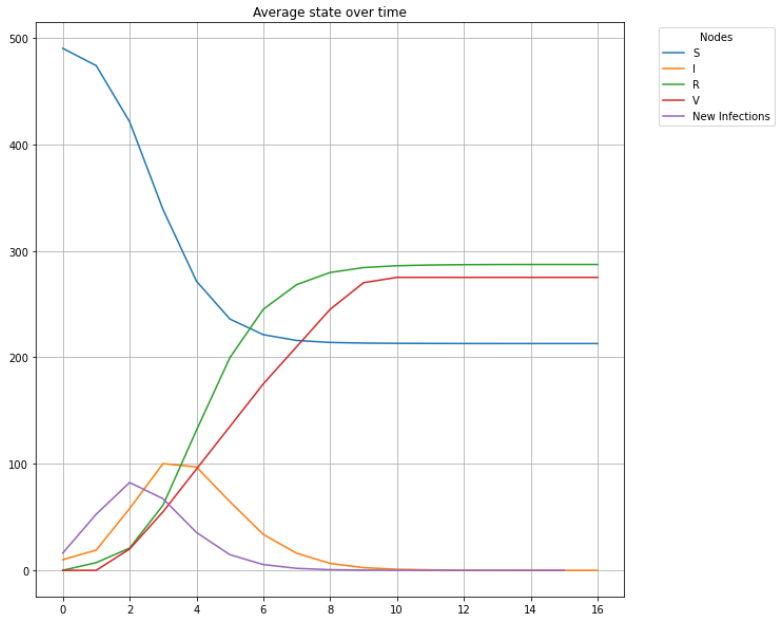
Modeling an SIR epidemic on a random graph PA (with Vaccination):

Given configuration parameters:

- Graph generated with previous model with k=6 n=500
- a susceptible node *i* has *m* infected neighbor
- parameters: $\beta = 0.3$, $\rho = 0.7$
- transition probabilities:
  - $P(X_i(t+1) = I \mid X_i(t) = S) = 1 - (1-\beta)^m$
  - $P(X_i(t+1) = R \mid X_i(t) = I) = \rho$

before vaccination	after vaccination	Description
S	S	Vaccinated nodes unable to be infected by their neighbors.
I	R	Vaccinated nodes unable infect other neighbors
R	R	Vaccinated nodes have no difference to not-vaccinated ones

In this exercise we simulate an epidemic including a vaccinated state *V*. Any node can be vaccinated, independently of its current state. Vaccinations will be managed by following a policy determining the percentage of total population to be vaccinated each week, and vaccinations will have immediate effects which are described in the configuration parameters above. To implement this, we will modify our original Epidemic module and apply a modification to stats w.r.t the effect of vaccination, where it prevents some nodes to take part to infection interactions during the simulations**.** We apply this in each step (week) and update the nodes accordingly. The given vaccination array (Vacc(*t*) = [0, 5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60, 60]) is describing percentage of total number of nodes (population) vaccinated in each time frame. So, it starts with 0% infected at in the first week, and reaching a maximum of 60% from week 8 and proceeding weeks)



Problem 4

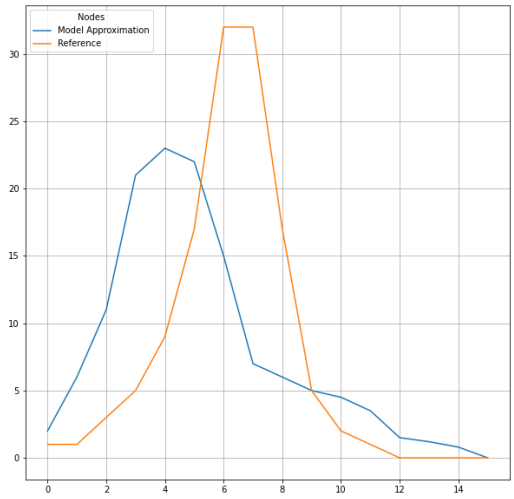
Modeling H1N1 pandemic:

In this exercise we simulate a model to be able to approximate the situation of H1N1 pandemic in Sweden, on a scaled down model by a factor of 104 from actual population data. Our model to generate the starting graph and infection spread is as applied in previous exercise. Holding true the vaccination schedule Vacc(t) = [5, 9, 16, 24, 32, 40, 47, 54, 59, 60, 60, 60, 60, 60, 60, 60]. and the behavior of vaccine on different node states.

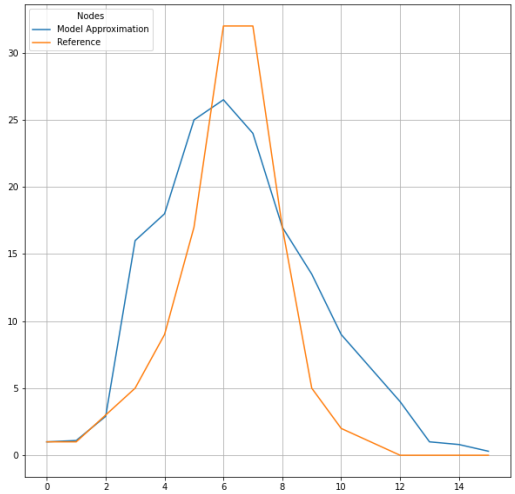
Here our goal is to make the experiment environment close to what happened in real life. For this we use RMSE as our objective function. Although there are many Gradient-based Optimization algorithms (e.g., SLSQP) since we are using our own custom module to reconstruct the Epidemic states in each simulation. It is no possible to use the pre-written optimization tools like (**`scipy.optimize`**). So, we will go ahead and define our own module by searching for the parameters of k, β, ρ which will for reaching the lowest RMSE with performing a gradient descent of each parameter set in their parameter-space $\{k0 - \Delta k, k0, k0 + \Delta k\}, \{\beta 0 - \Delta \beta, \beta 0, \beta 0 + \Delta \beta\}, \{\rho 0 - \Delta \rho, \rho 0, \rho 0 + \Delta \rho\}$.
The simulation will be repeated for each combination of parameters, and since we are trying to find the optimum point for 3 independent parameters we create a check point for each local optima achieved based on current variable manipulation; Then we reduce our deltas, up to half, to examine if we can move further toward lower values of RMSE. This would help to avoid being stuck in local minimum. Yet, if we didn't locate such point the algorithm will start from beginning with original deltas to start a simulation with new values of next variable.

In the end we must define our stopping criterion by assigning a "threshold" for the variation and a "patience" for number of repetitions of simulation allowed to try to overcome the local optima. After reaching the limit determined by patience parameter, if there were no further changes, we can accept the final result of the objective function as the best minimum possible for that set of parameters. And plot the final achieved model to compare.

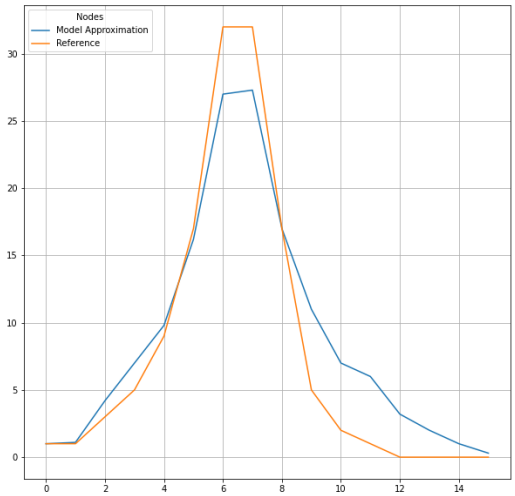
Our experiments report that, while it is possible to approximate the new infections reference plot with our concluded set of parameters ($k = 5, \beta = 0.45, \rho = 0.25$); The model could not simulate the exact environment of H1N1 pandemic which is acceptable since there's definitely more aspect to the spread of infection than simply the probability of infection as a fixed numerical value. Chance of recovery and the effect of vaccines also follow the same complex behavior. Our experiments are represented below:



Best RMSE=9.88 at: k=4 beta=0.2 ro=0.8



Best RMSE=9.82 at: k=4 beta=0.45 ro=0.5



Best RMSE=9.7 at: k=5 beta=0.45 ro=0.25