Kevin Wang (604036846)

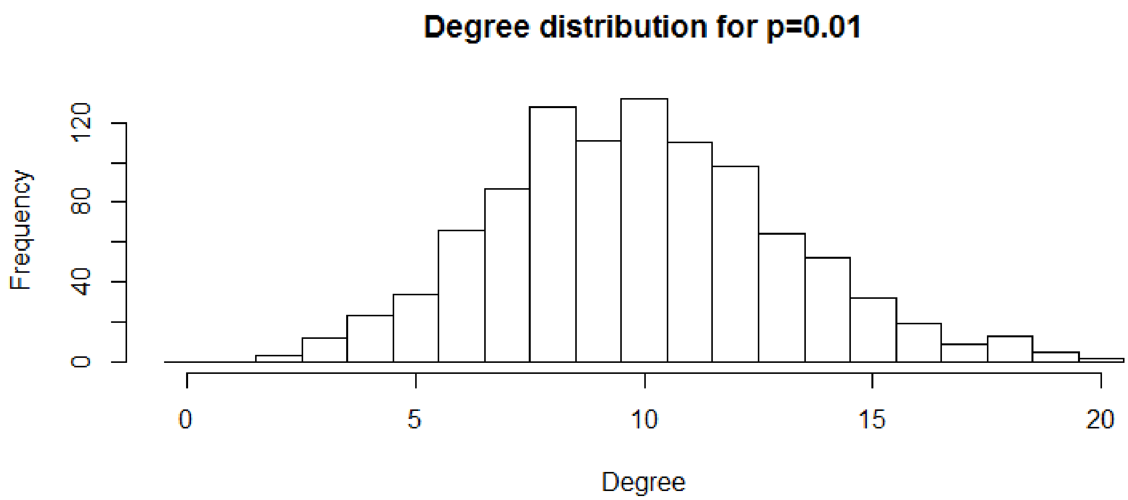
EE 232E Homework 1

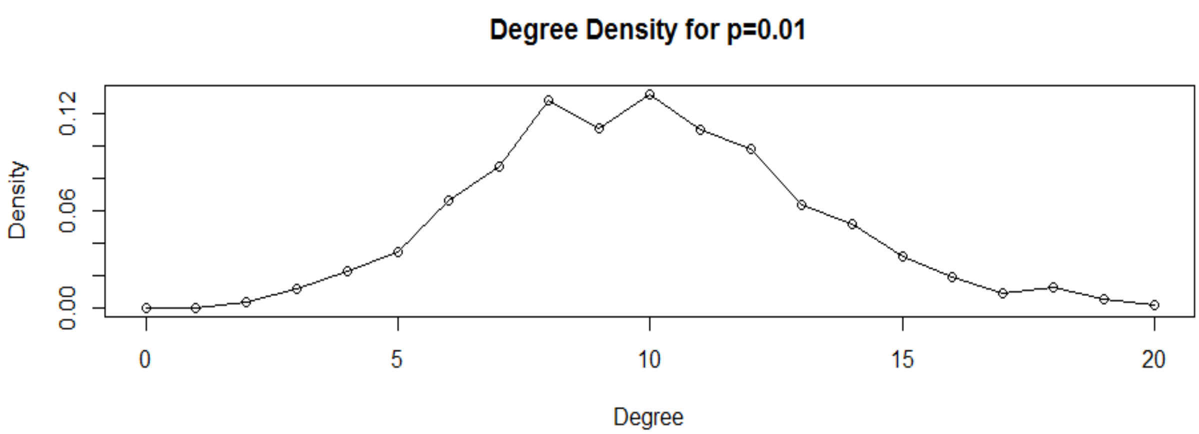
The coding for this assignment was done in R. There is a separate .R file for each problem, named in the fashion hw1-x, where x is the problem number.

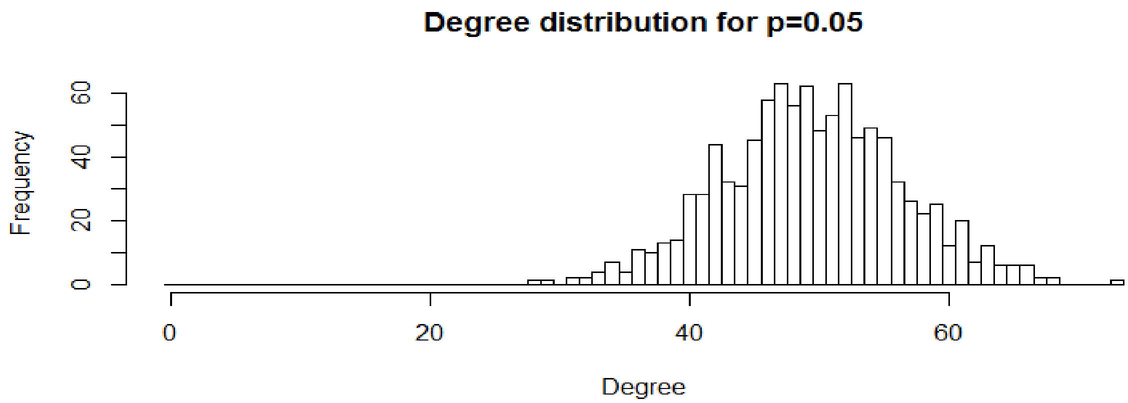
Problem 1

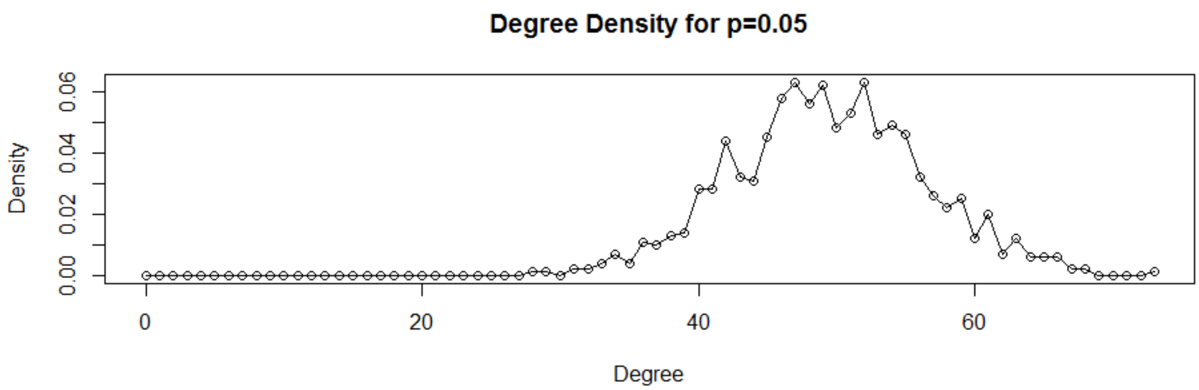
Part 1a

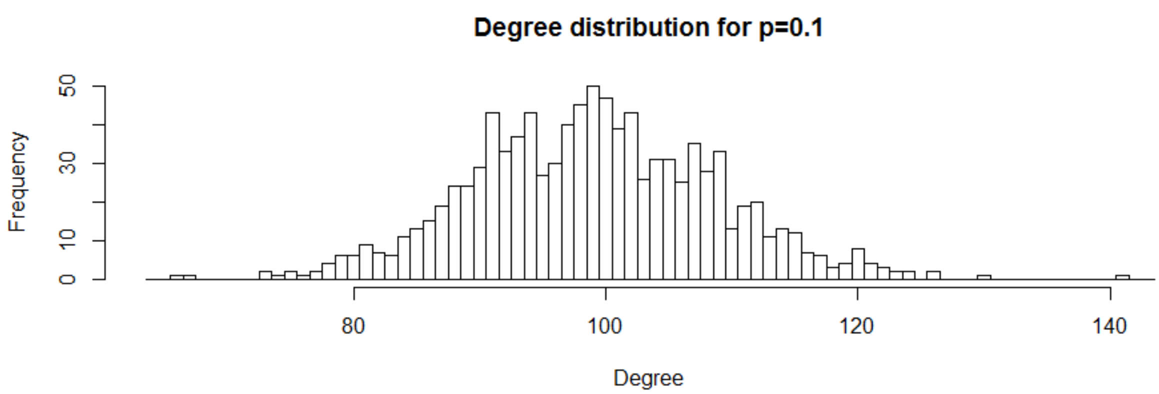
The three undirected networks are generated using R and the igraph package. The following plots show the degree distributions for each p value followed by another plot in terms of degree density.

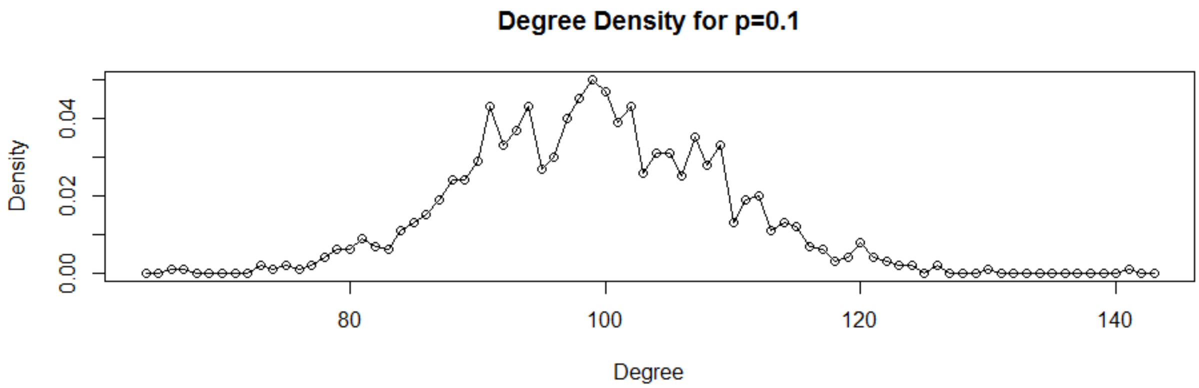





Degree distribution for p=0.05
Frequency
0
20
40
60
0
20
40
60
Degree


Degree Density for p=0.05
Density
0.00
0.02
0.04
0.06
0
20
40
60
Degree


Degree distribution for p=0.1
Frequency
0
10
30
50
80
100
120
140
Degree



Part 1b

According to the `is.connected` function, all 3 generated networks are connected.

For this set of generated graphs, the following diameters are obtained from the `diameter` function:

p	0.01	0.05	0.1
Diameter	6	3	3

Part 1c

The number of nodes is set to 1000 as before. Starting from p=0, p is incremented by 0.00005 until the generated graphs become connected. For 1 particular trial, the estimate for p_c is found to be about 0.00625.

Part 1d

Part is dropped

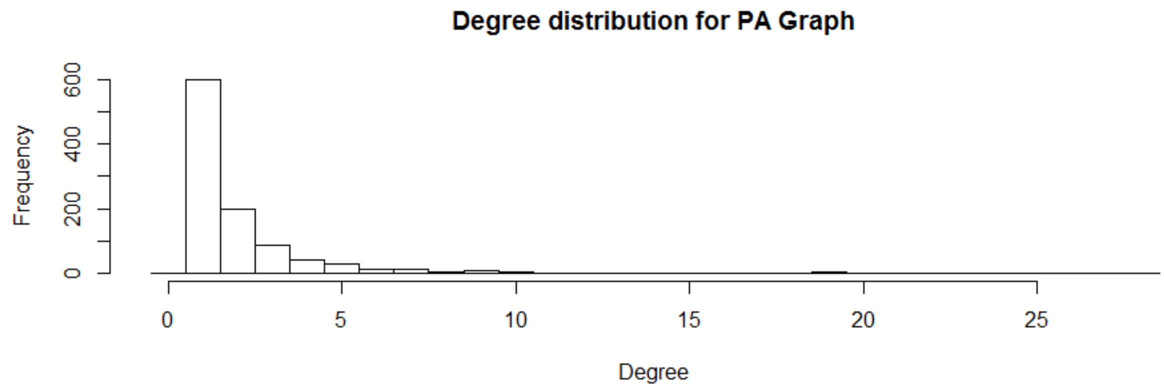
Problem 2

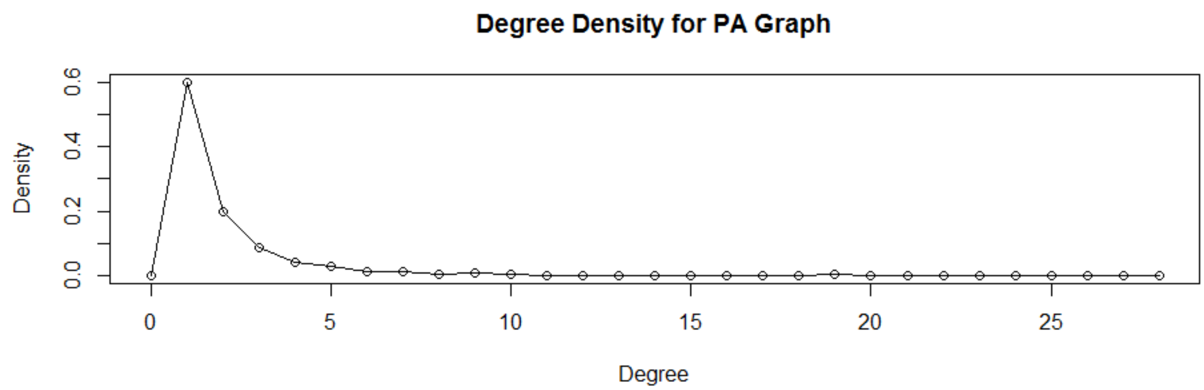
Part 2a

To create the networks with fat-tailed degree distributions, the `barabasi.game` function is used to generate graphs with preferential attachment.

In this case, the diameter of the generated graph is found to be 18.

The degree distributions are shown:





Part 2b

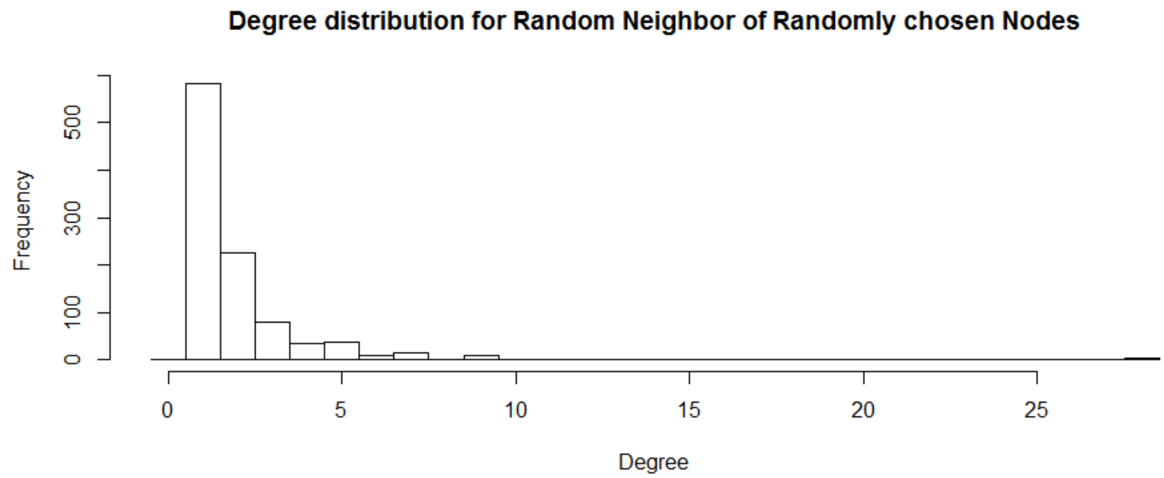
According to the `is.connected` function, the generated network is connected.
The modularity of the GCC is found to be 0.9322962. The clusters function returned all of the nodes in 1 cluster only, so the GCC was just the whole network, implying that the probability of there being an edge exceeded the critical probability. This is probably why the modularity ended up being so large, since there was only 1 cluster.

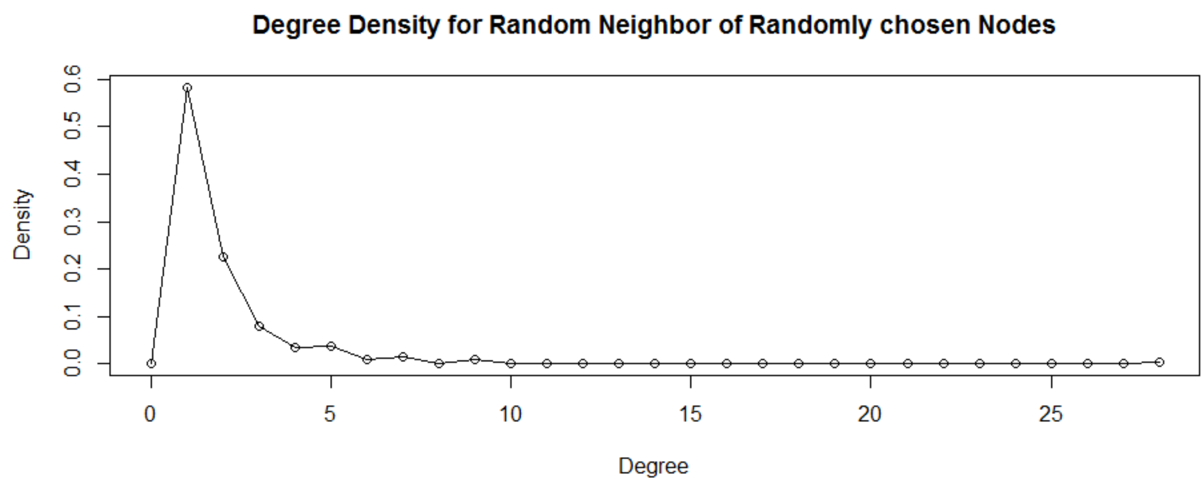
Part 2c

The new modularity is 0.9787266 with 10,000 nodes. This number is larger than that of the smaller network, but they are still pretty close in number. Perhaps this is the result of the random generation of the network.

Part 2d

In this part, the number of nodes was reset to 1000. A node is selected at random, and then one of its neighbors is randomly selected and its degree recorded; this process is repeated 1000 times to obtain the following degree distributions:





The degree distribution of the neighboring nodes is similar to that of the regular network, which is not that surprising.

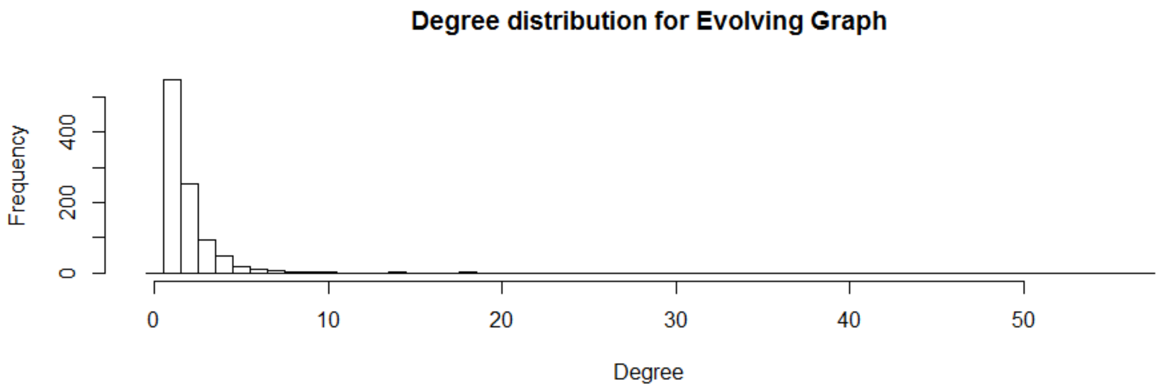
<u>Problem 3</u>

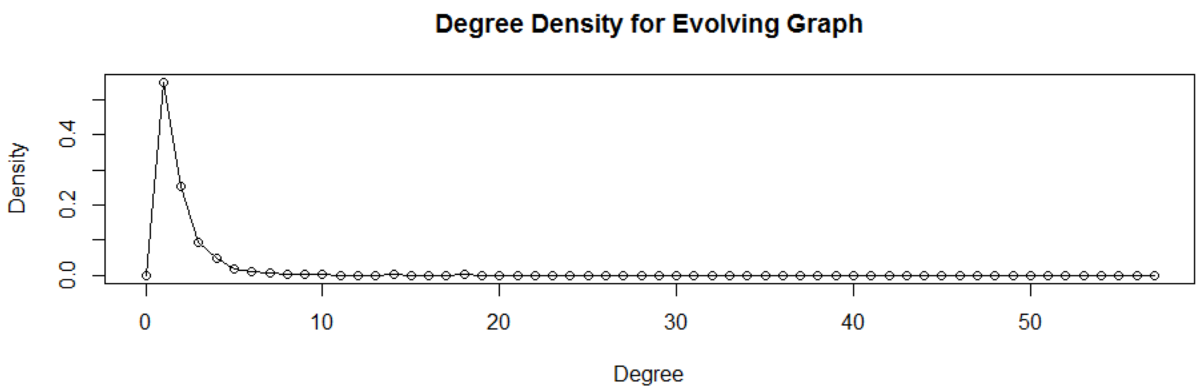
<u>Part 3a</u>

In this part, an evolving/aging graph must be made, so the `aging.barabasi.game (sample_pa_age)` function is used. The exponents for the preferential attachment and aging need to be specified, so for this particular trial instance the following values are chosen:

pa.exp = 2, aging.exp = -2

The following degree distribution plots can then be made:





<u>Part b</u>

The `modularity` function is used on the output of the function `cluster_fast_greedy`. The result is:

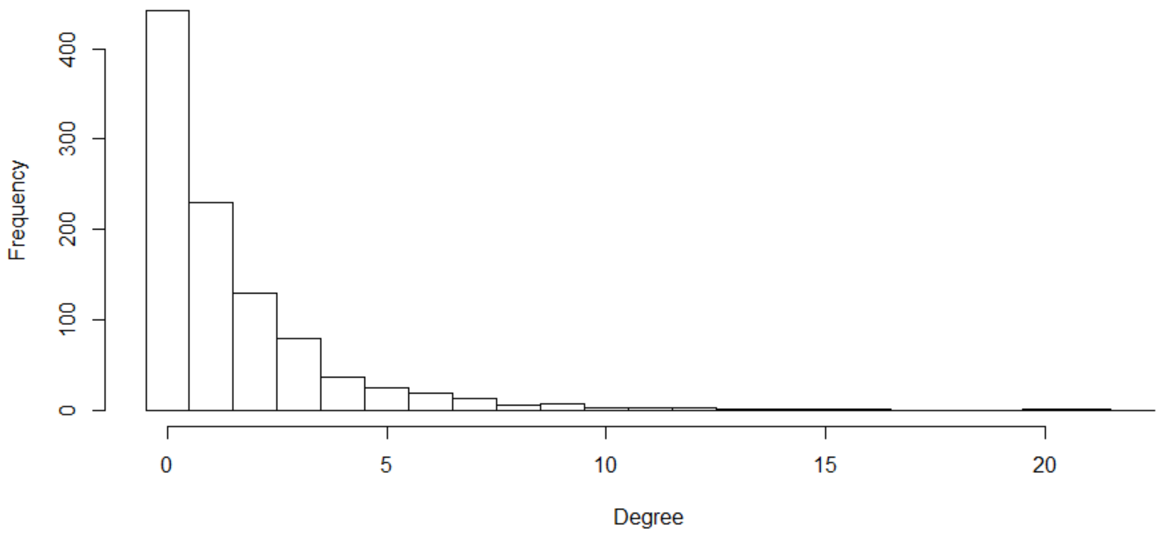
Modularity = 0.9281925

<u>Problem 4</u>

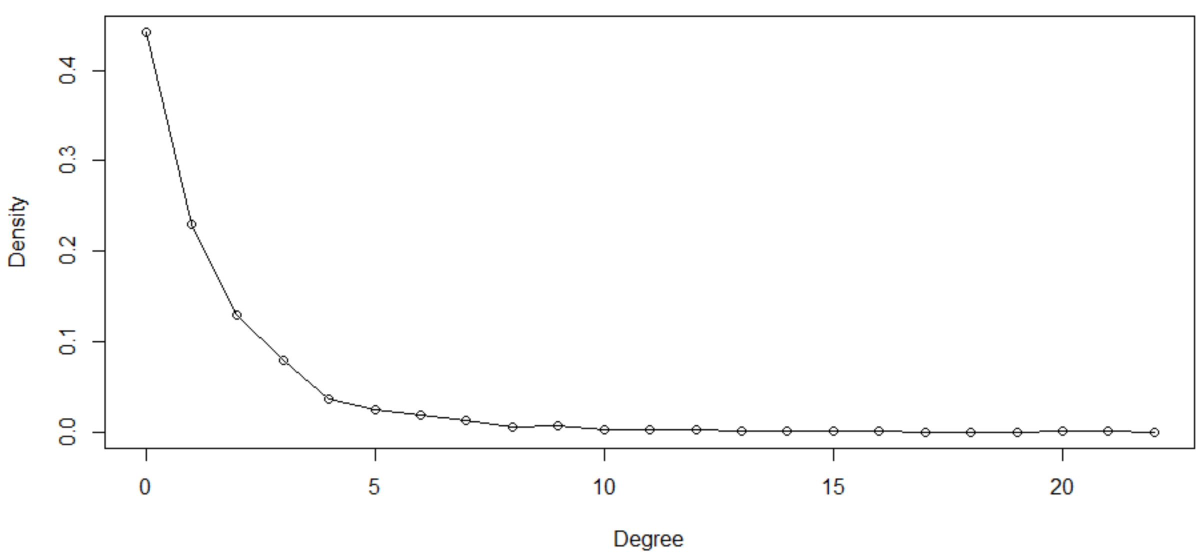
<u>Part 4a</u>

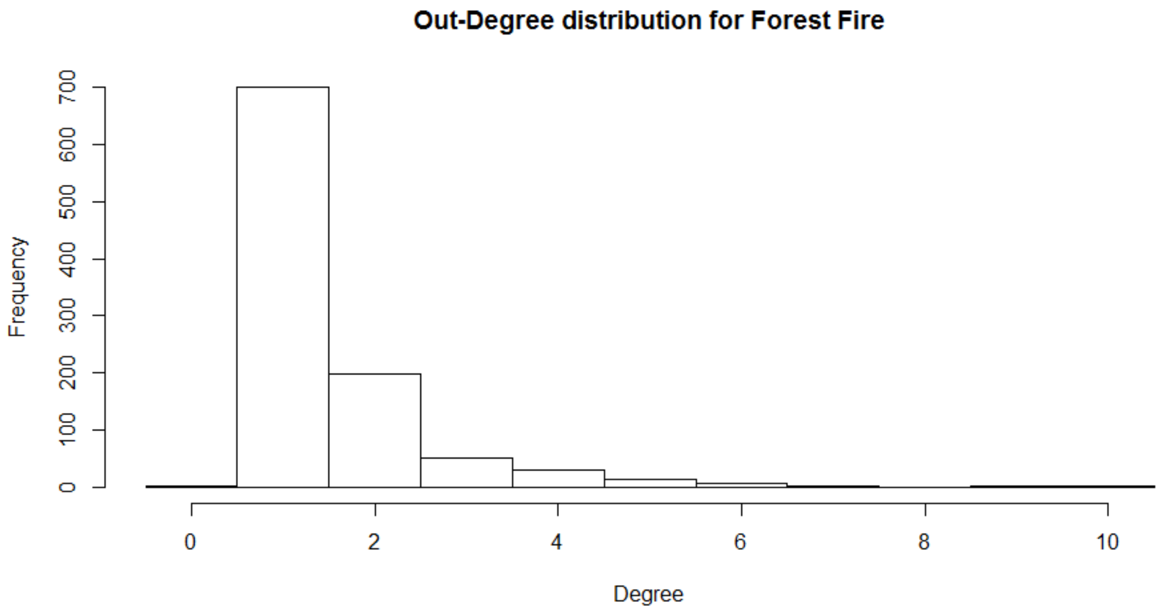
The `forest.fire.game` function is used to generate the network model. Various values for the forward burning probability fw.prob were tested. The following analysis shows the case when fw.prob is set to 0.2.

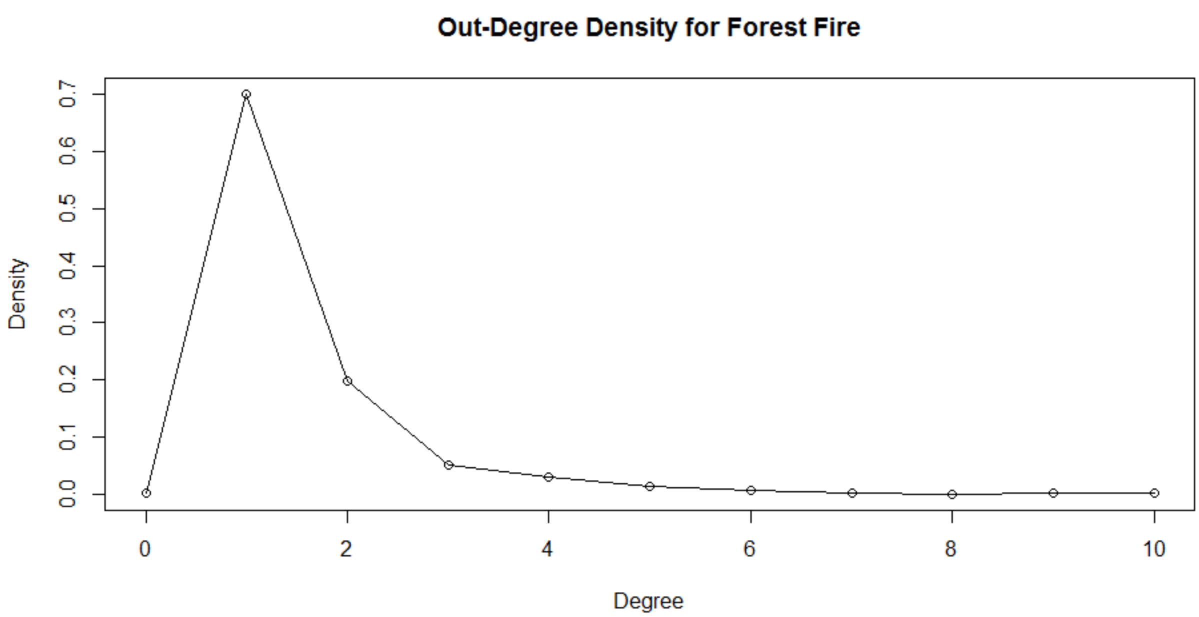
In-Degree distribution for Forest Fire



In-Degree Density for Forest Fire







<u>Part b</u>

The diameter of the graph is found to be 12.

<u>Part c</u>

The community structure is found using the `infomap` method through `cluster_infomap` since it is a directed graph and cluster_optimal made my computer freeze up. The resulting modularity is found to be 0.8228.