ECE 232E Project 1

Yifan Shu, Chengshun Zhang, Xuan Yang, Yifan Zhang

Part 1: Generating Random Networks

1. Create random networks using Erdos-Renyi (ER) model

(a)

For this problem, we created undirected random networks with n = 1000 nodes, and the probability p for drawing an edge between two arbitrary vertices 0.003, 0.004, 0.01, 0.05, and 0.1. The degree distribution of those random networks were plotted as below. Each of those distribution plots (shown as below) has one peak frequency at around the value of n nodes times the probability p, which is degree. The degree of a node in the network is the number of connections it has to other nodes and the degree distribution is the probability distribution of these degrees over the whole network, so the highest frequency of the degree occurs at the degree of n nodes times probability p.

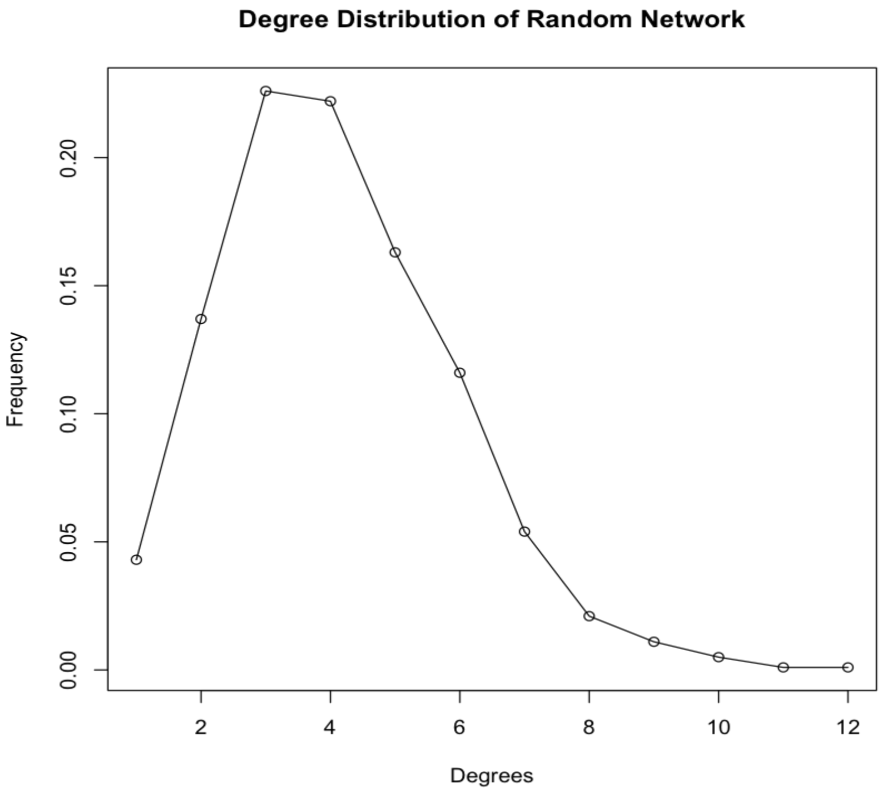

Figure 1.1.1 Degree distribution of random network with p = 0.003

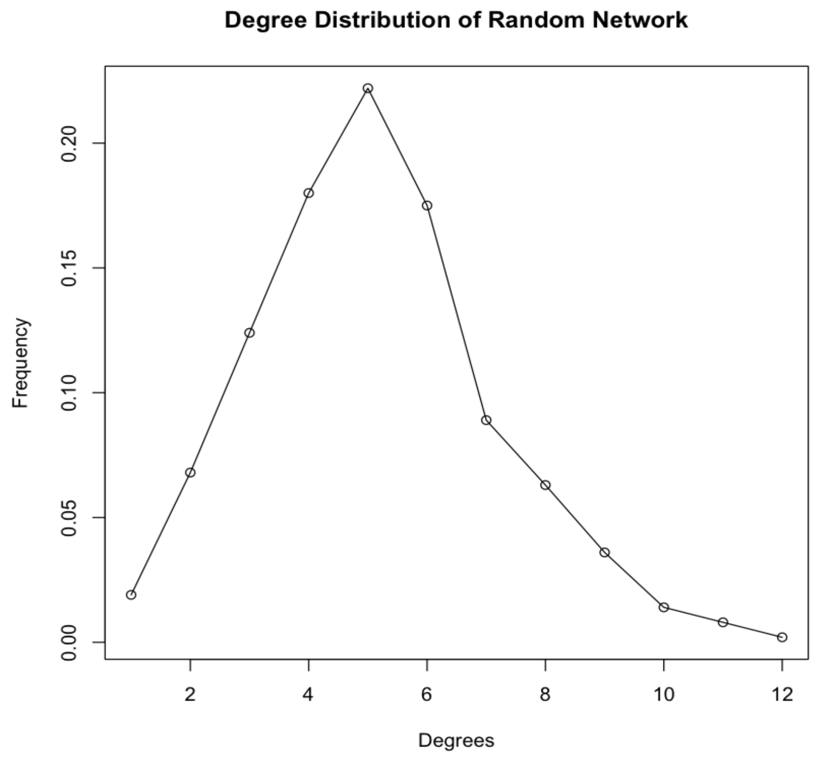

Figure 1.1.2 Degree distribution of random network with p = 0.004

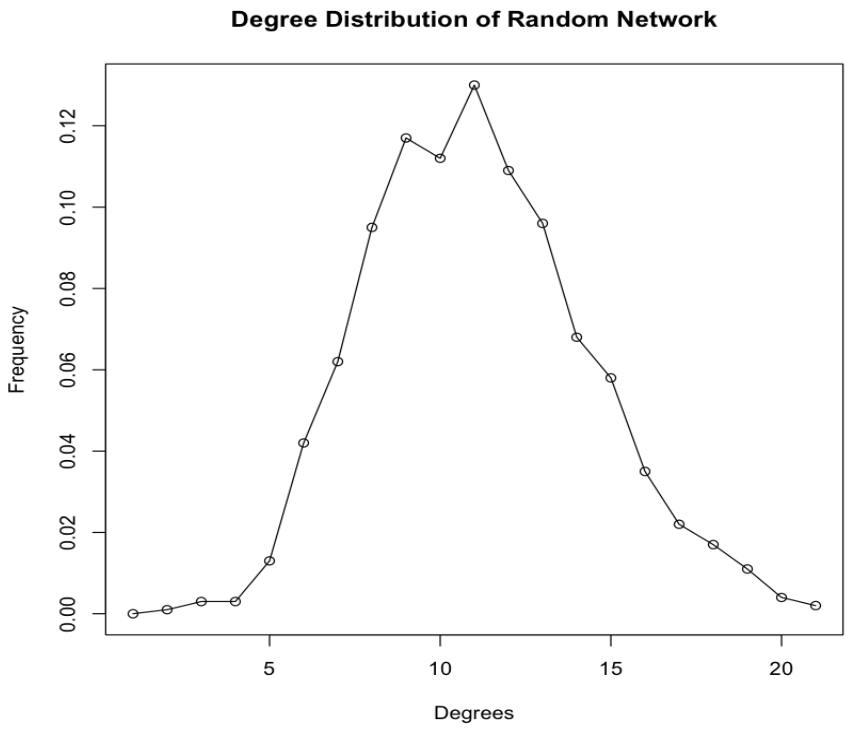

Figure 1.1.3 Degree distribution of random network with p = 0.01

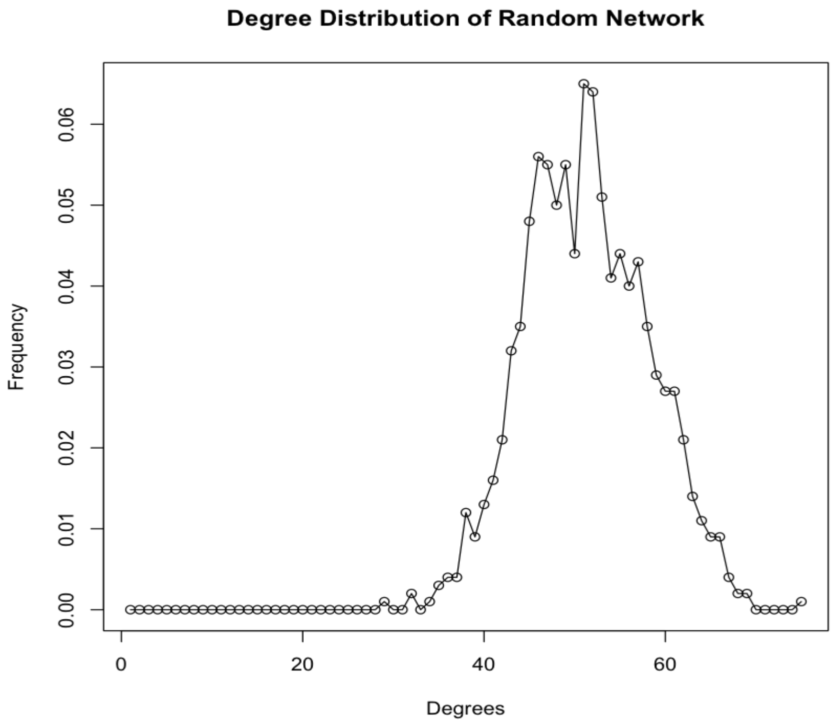

Figure 1.1.4 Degree distribution of random network with p = 0.05

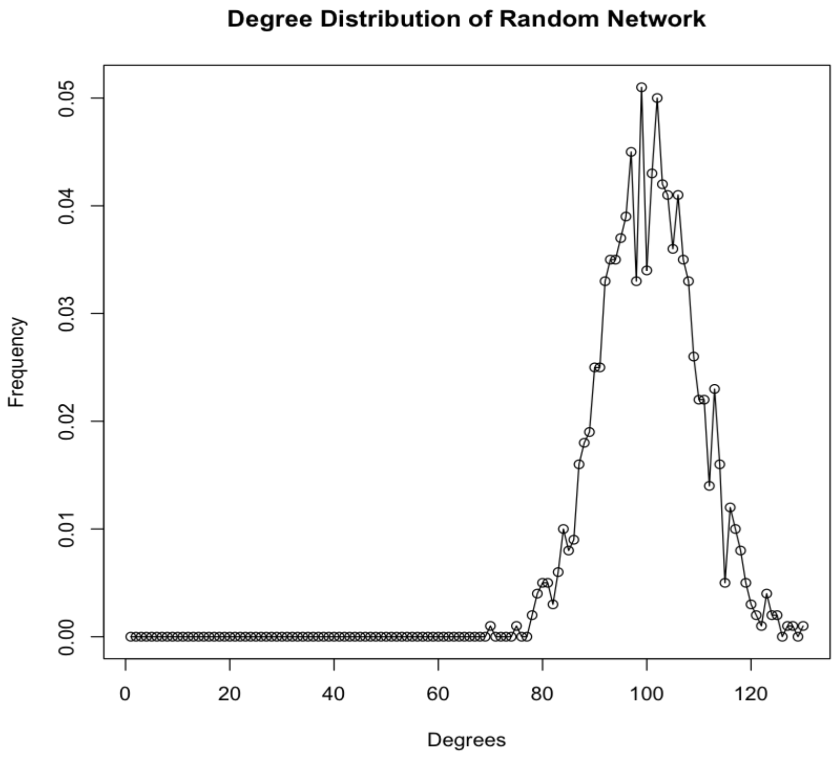

Figure 1.1.5 Degree distribution of random network with p = 0.1

	P = 0.003	P = 0.004	P = 0.01	P = 0.05	P = 0.1
Mean	3.112	4.11	10.086	50.284	99.47
Variance	3.168625	3.993894	10.12873	45.97332	80.56366

Table 1.1.1 Mean & Variance of each random network corresponding to certain p.

(b)

For each p and n = 1000 nodes, we checked the ER networks' connection condition. We can find that ER networks are not connected until probability p reaching 0.01, so we can say that not all the ER networks are connected. The estimated probability is 0.01 which generate connected network.

	P = 0.003	P = 0.004	P = 0.01	P = 0.05	P = 0.1
Is connected?	False	False	True	True	True
d of GCC	15	11	-	-	-
Total d	15	11	6	3	3

Table 1.1.2 Connection condition & size of GCC for each random network with certain p.

(c)

For n = 1000, we swept over values of p from 0 to a p_{max} that makes the network almost surely connected and create 100 random networks for each p. And scattered the plot of size of GCC vs p. The estimated maximum p value is 0.01 from part b. If we define the emergence as the normalized GCC size going to 1, the experimental result of p value, where a giant connected component starts to merge, is 0.005. When we try p value around 0.007, the result that networks start to be connected becomes stable. This matches to theoretical value. So the empirically estimated value of p, where the giant connected component takes, up over 99% of the nodes, is 0.007.

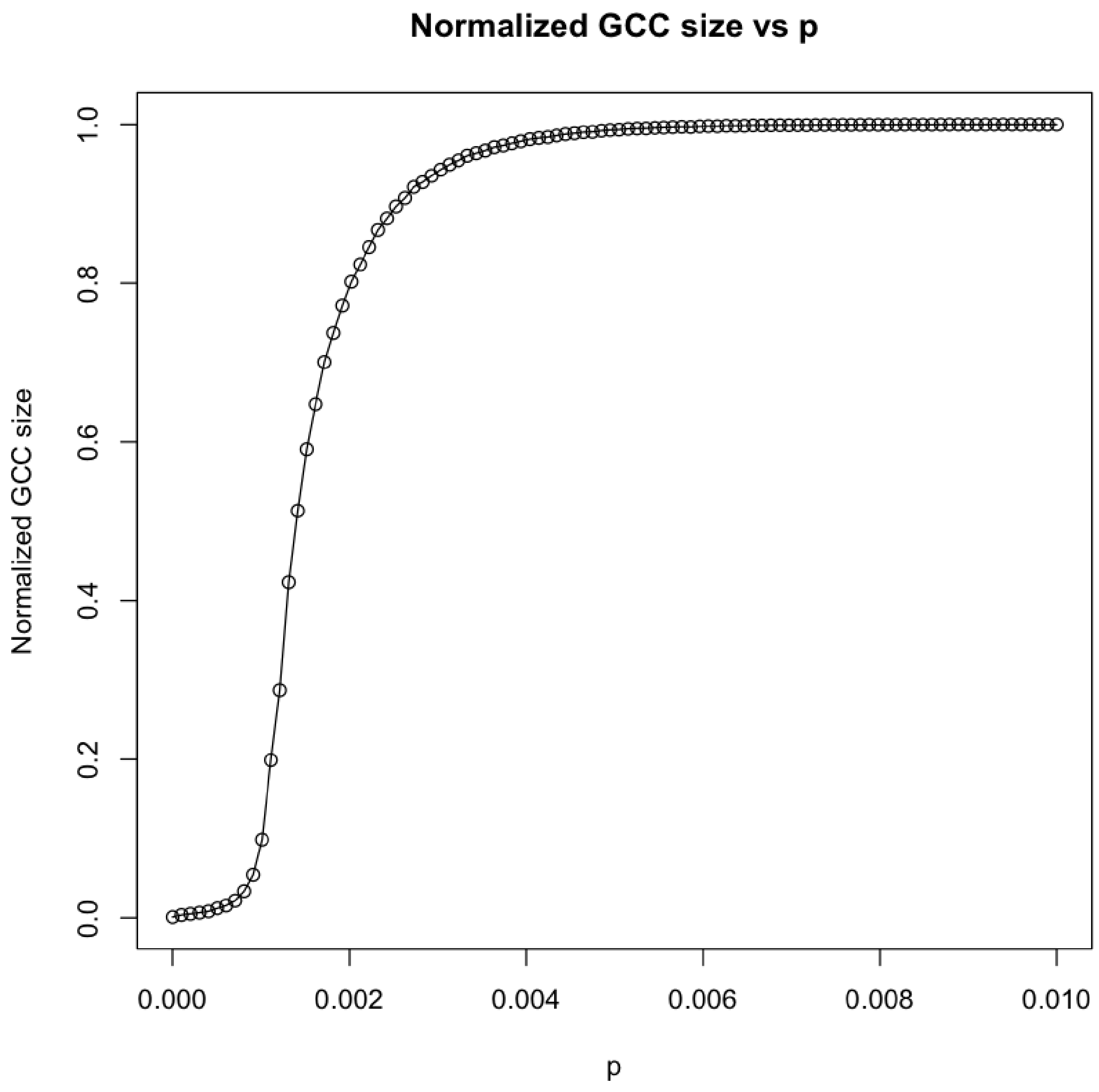

Figure 1.1.6 Normalized GCC size vs p varying from 0 to 0.01

(d)

Define the average degree of nodes c = n * p = 0.5. We swept over the number of nodes, n, ranging from 100 to 10000 and plotted the expected size of the GCC of ER networks with n nodes and edge-formation probabilities p = c/n, as a function of n. We also repeated the same step for c = 1, 1.1, 1.2, 1.3. When the average degree of nodes is defined as a constant 0.5, we find that the size of GCC is increased with the increased n value, but the increasing speed is decreasing when n value increases. When the average degree of nodes is defined as a constant 1, we find that the increasing speed is almost constant when n value increases. When the average degree of nodes is defined as a constant 1.1, 1.2, and 1.3, we find that the increasing speed is almost constant when n value increases, but the slope of those plots increases with the increasing c value. The expected GCC size increases when n increases, but the increasing speed differentiates by different degrees.

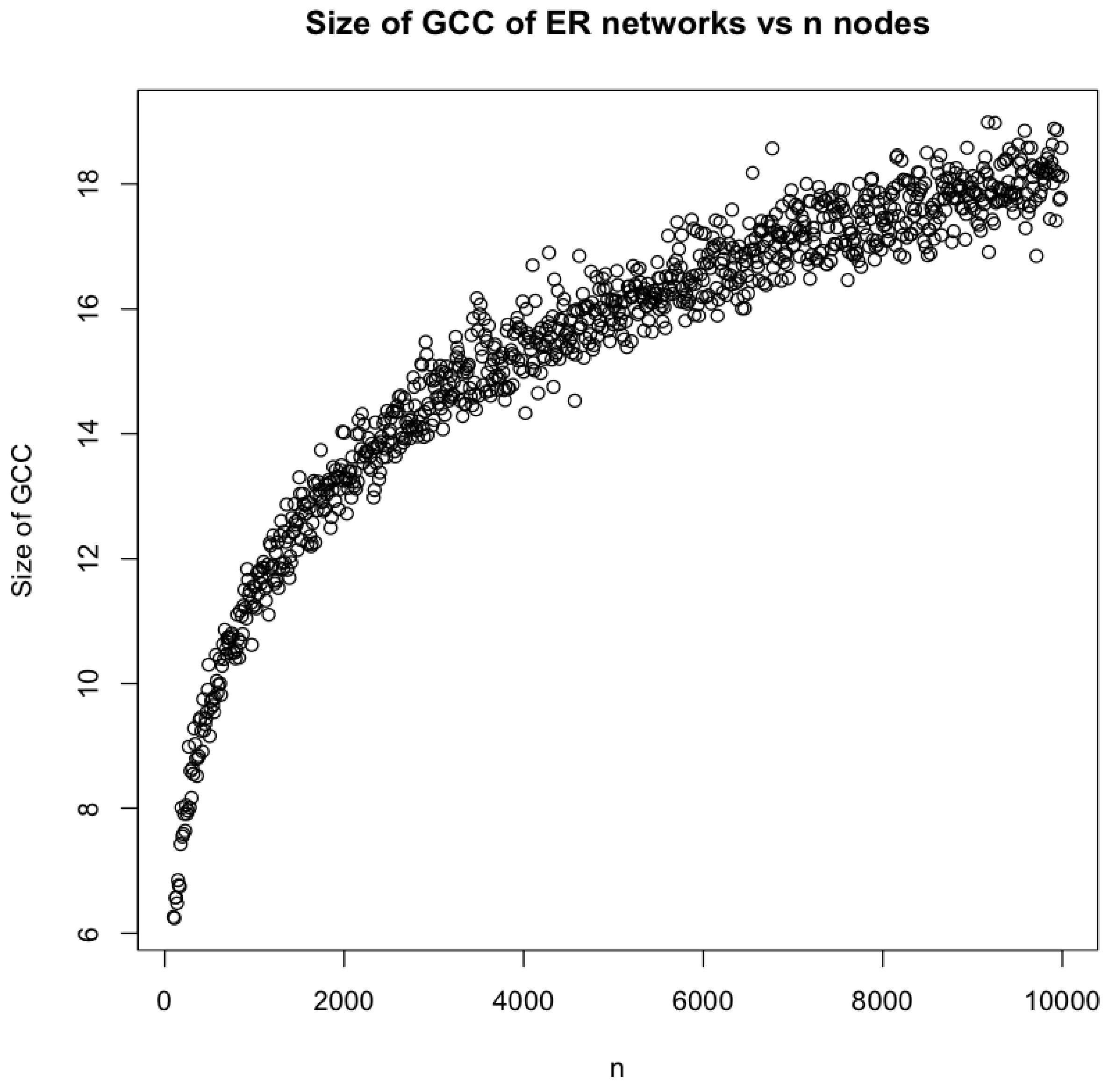

Figure 1.1.7 Size of GCC of ER networks vs n nodes when c = 0.5

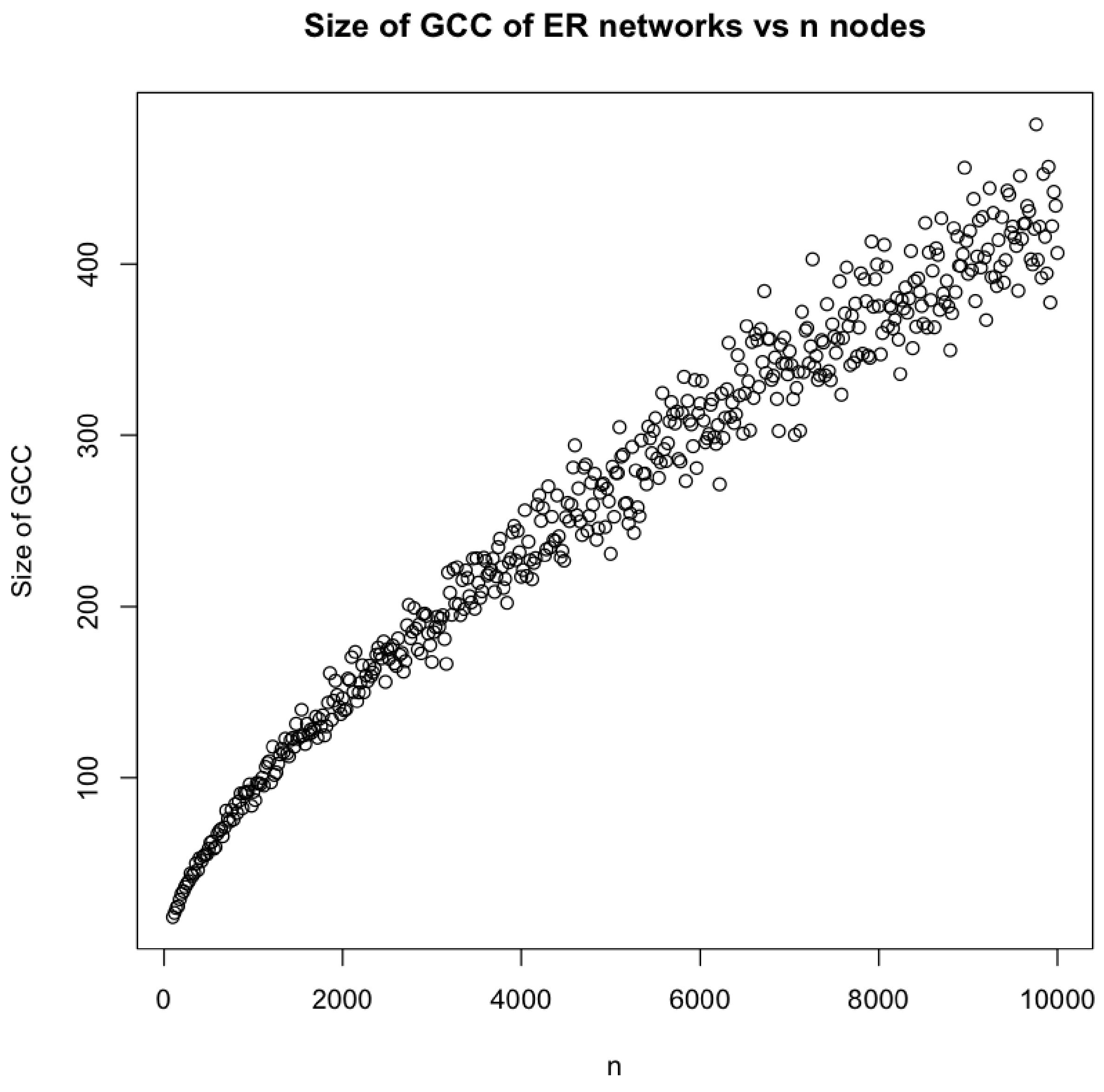

Figure 1.1.8 Size of GCC of ER networks vs n nodes when c = 1

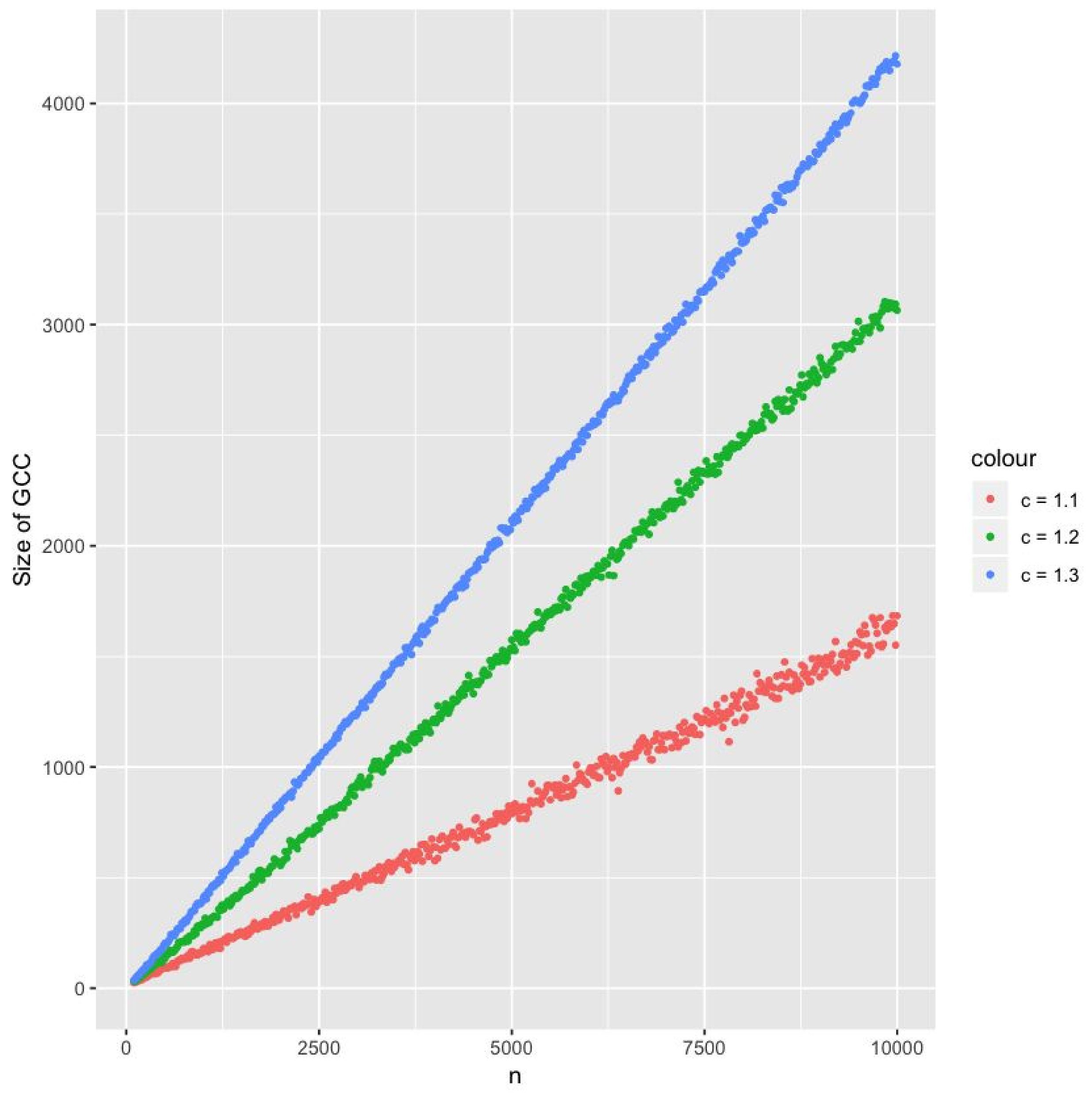

Figure 1.1.9 Size of GCC of ER networks vs n nodes when c = 1.1, 1.2, 1.3.

2. Create networks using preferential attachment model

(a)

We create an undirected network with n = 1000 nodes, with preferential attachment model, where each new node attaches to m = 1 old nodes. From the experiment, we know that this network is always connected.

(b)

We use fast greedy method with n = 1000 nodes to plot the community structure below. The modularity is 0.9326844.

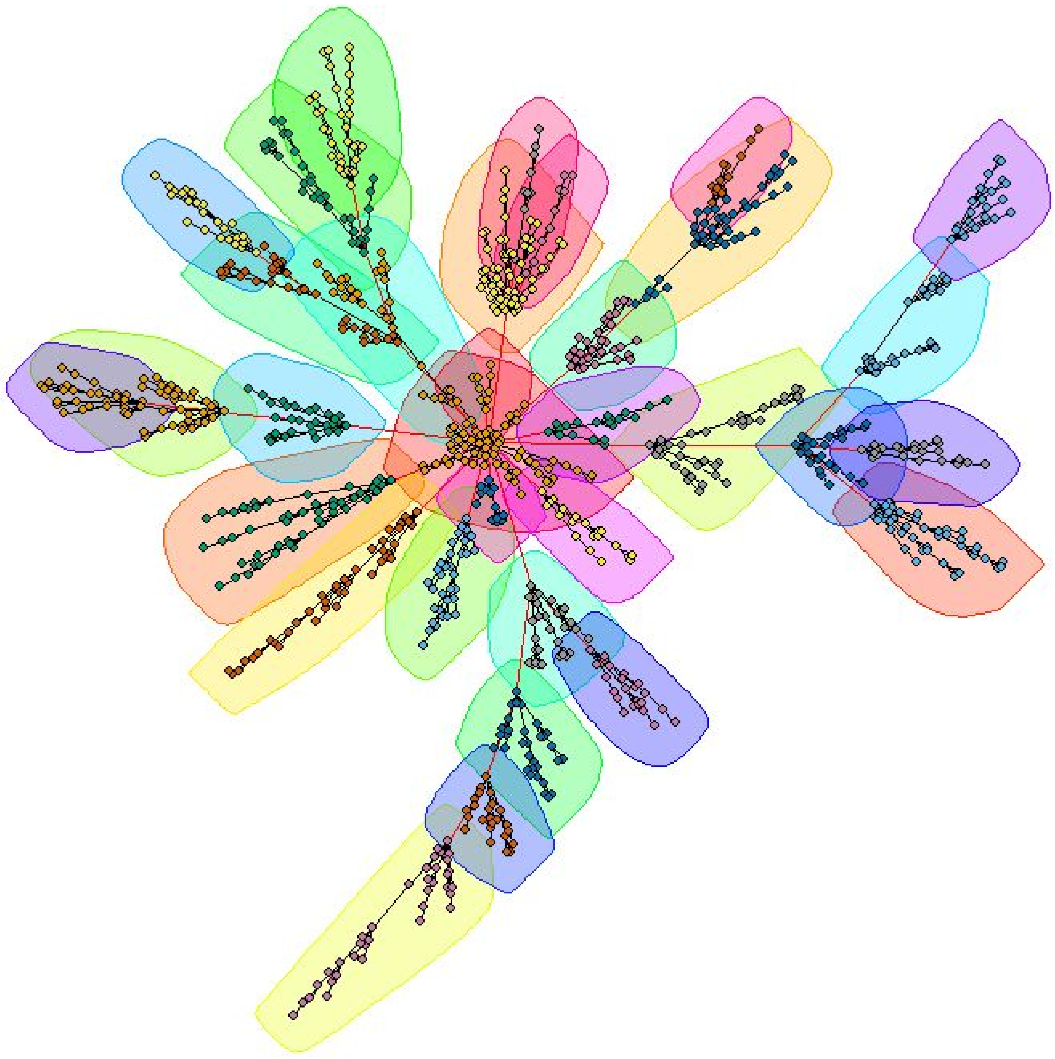

Figure 1.2.1 community structure using fast greedy method

(c)

We use fast greedy method with n = 10000 nodes to plot the community structure below. The modularity is 0.9783202. Its modularity is larger than the smaller network's modularity.

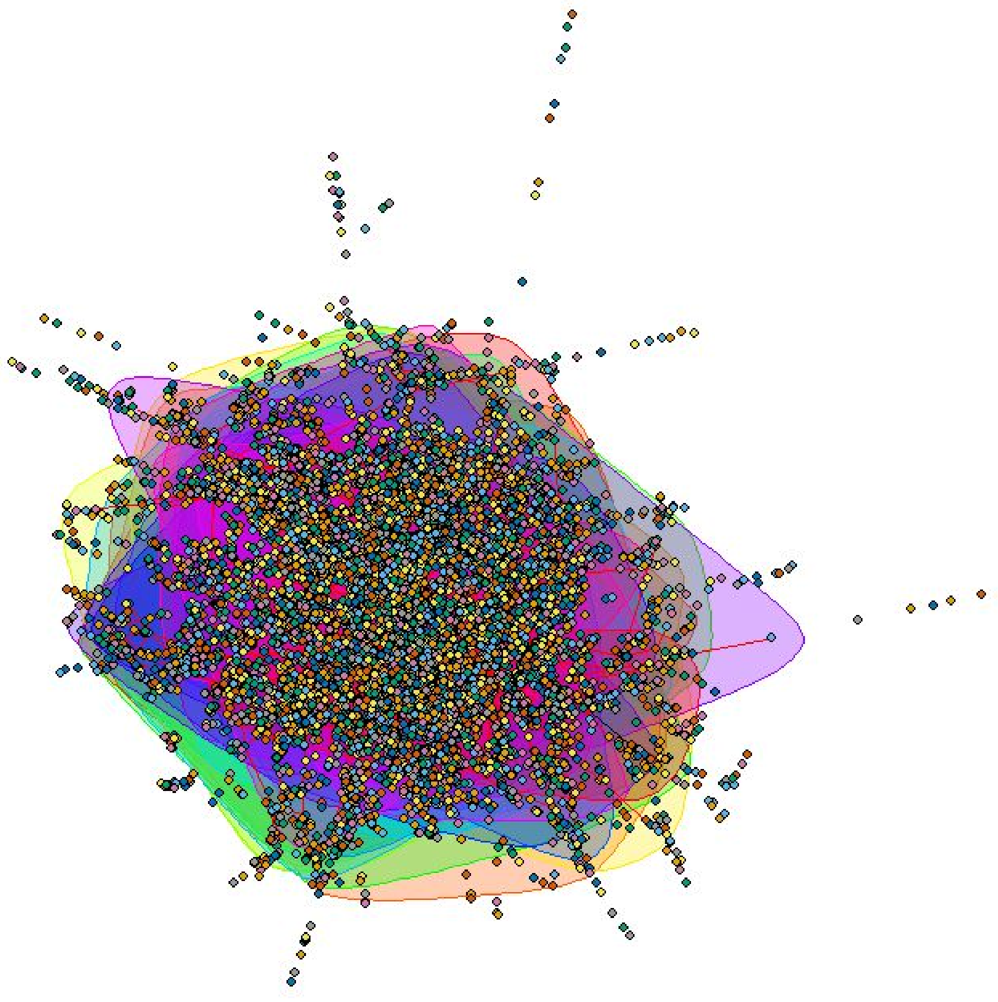

Figure 1.2.2 community structure using fast greedy method

(d)

We plot the degree distribution in a log-log scale for both n = 1000, and n = 10000, and then estimate the slope of the plot using linear regression. From the plot for n = 1000 nodes, we pick two points (2,0.500) and (10,0.005) to calculate the slope for n = 1000, and find the slope is -2.86. Similarly, we pick (2,5*10^-1) and (20,5*10^-4) for n = 10000 and the slope is -3.

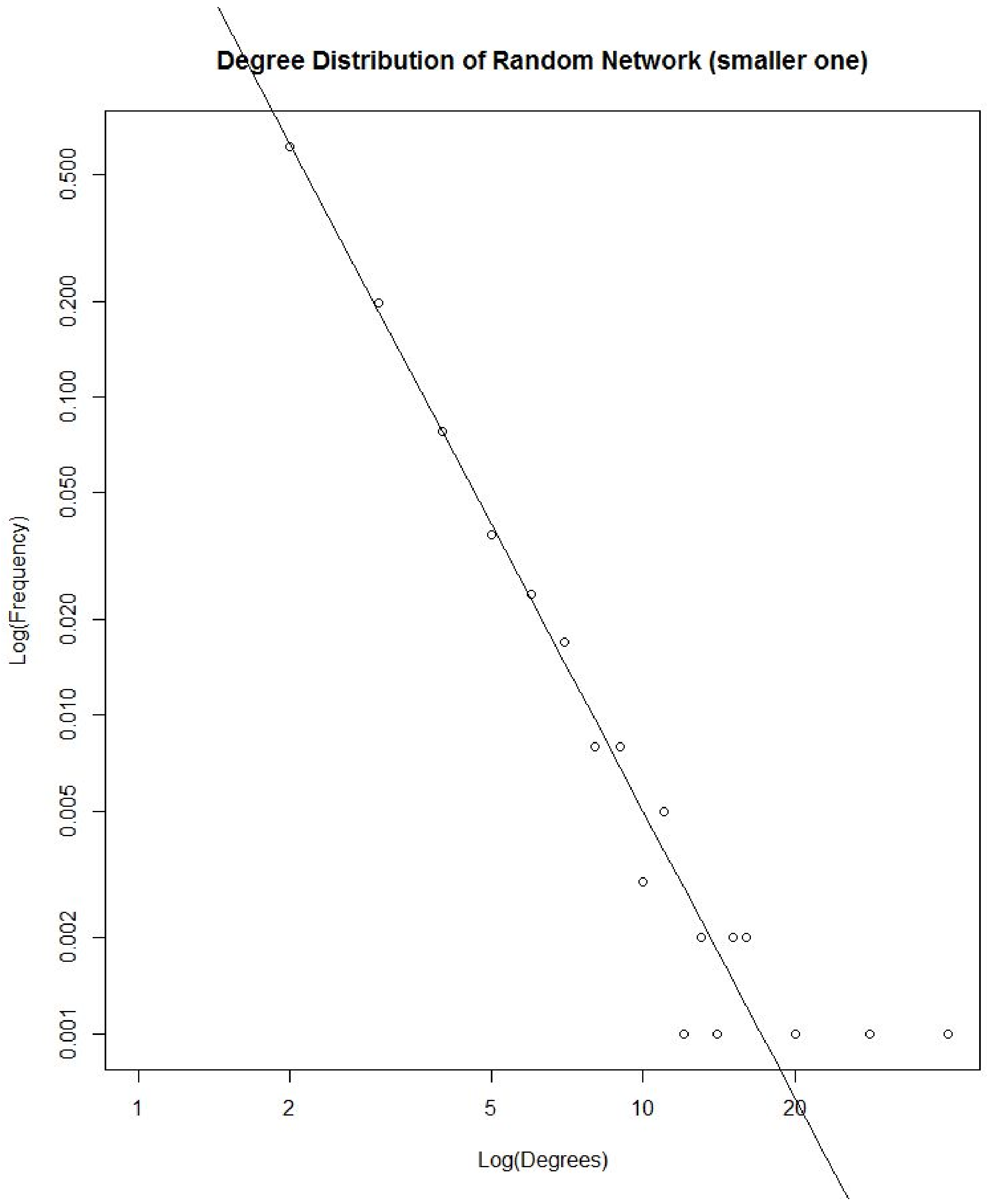

Figure 1.2.3 Degree distribution of random network with n = 1000

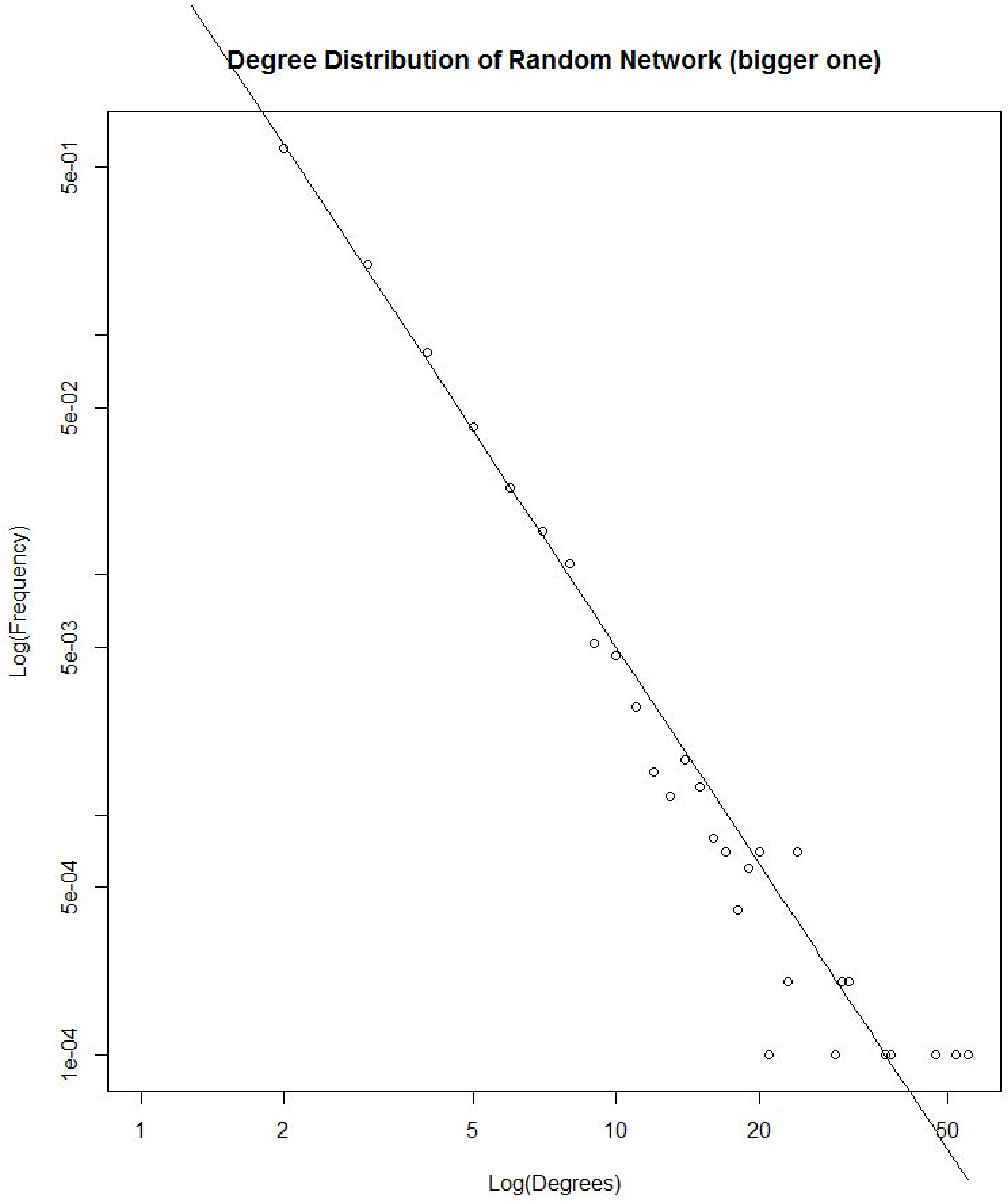

Figure 1.2.4 Degree distribution of random network with n = 10000

(e)

From the networks plots above in 2(d), we pick the first random node, and plot the degree distribution for another random node to the first node picked in log-log scale. The degree distribution plot is below. It's not linear for the log-log scale. The previous log-log plots have a constant slope when we use linear regression method, but the slope of this plot is different, which keeps constant at first and then sharply decreases and increases back and forth.

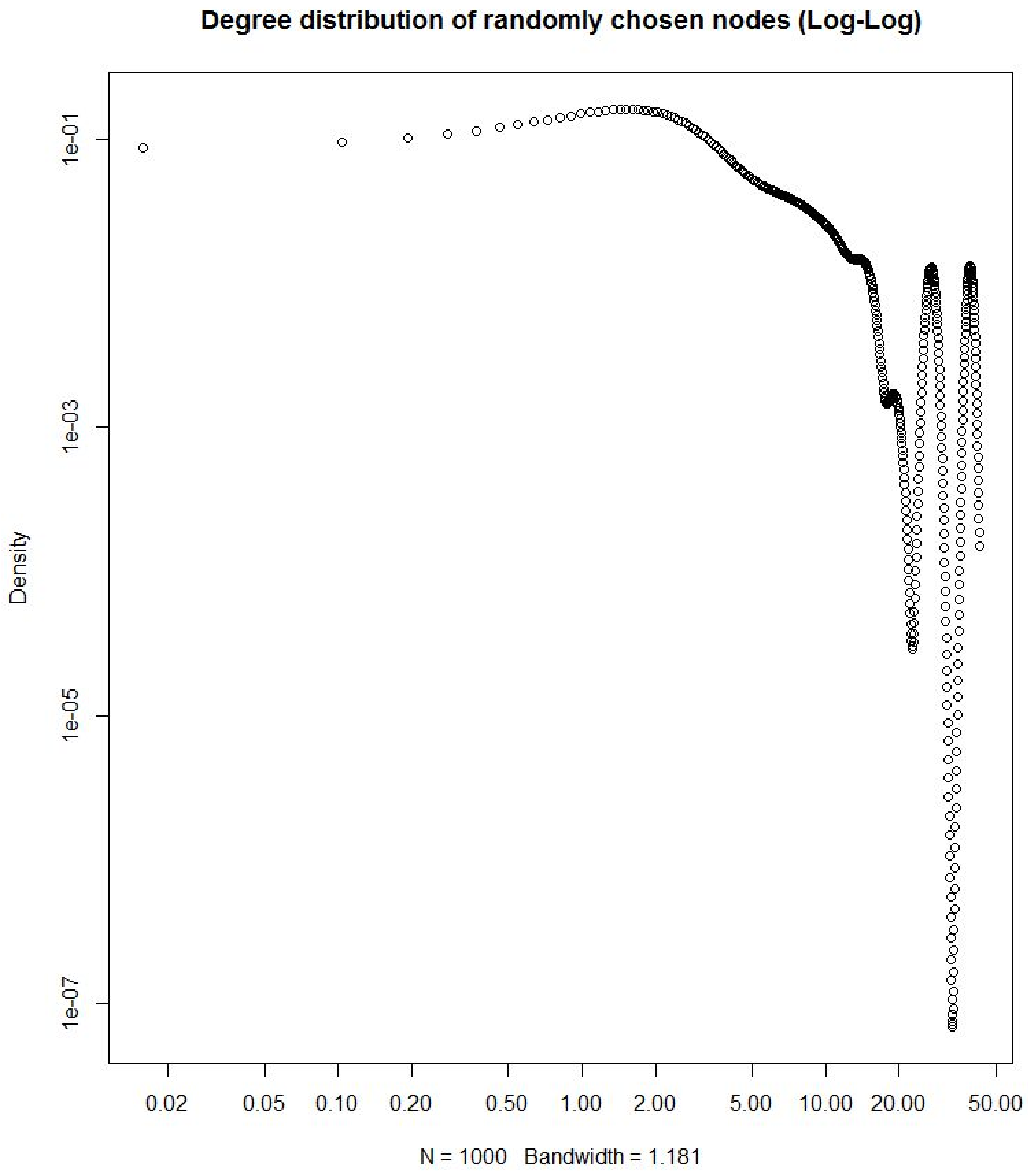

Figure 1.2.5 Log-log scaled degree distribution of randomly chosen nodes

(f)

For a time step which is larger than 0 and less than 1001, we estimate the expected degree of a random node that is added. Then we plot the relationship between the age of nodes and their expected degree below.

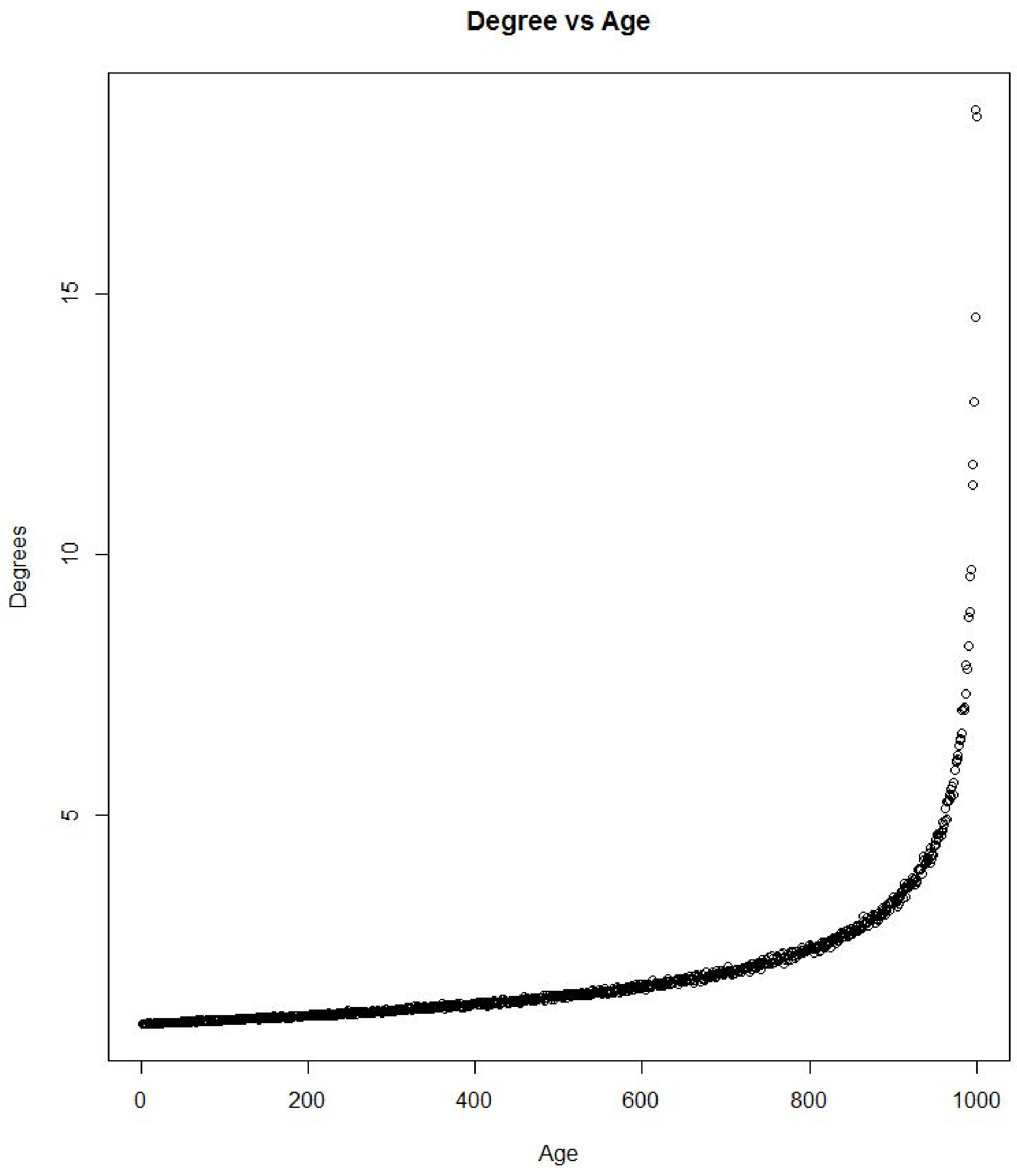

Figure 1.2.6 the expected degree of a node V.S. age of the node

(g)

The m is a numeric constant, the number of edges to add in each time step in BA model, and the modularity is a measurement of the strength of division of a network. So when m increases, the density of the network reduces and thus decreases the modularity.

We repeat part (a), (b), and (d), because for part (c), (e), and (f), they are redundant and have a similar result to the result when m = 1. But the modulation in (a), the visualization of network in (b), and degree distribution log-log plot are different for different m values.

	M = 1	M = 2	M = 5
Is connected?	True	True	True
Modularity	0.9323112	0.5244973	0.2755903

Table 1.2.1 Connection condition & modularity for network with different m values

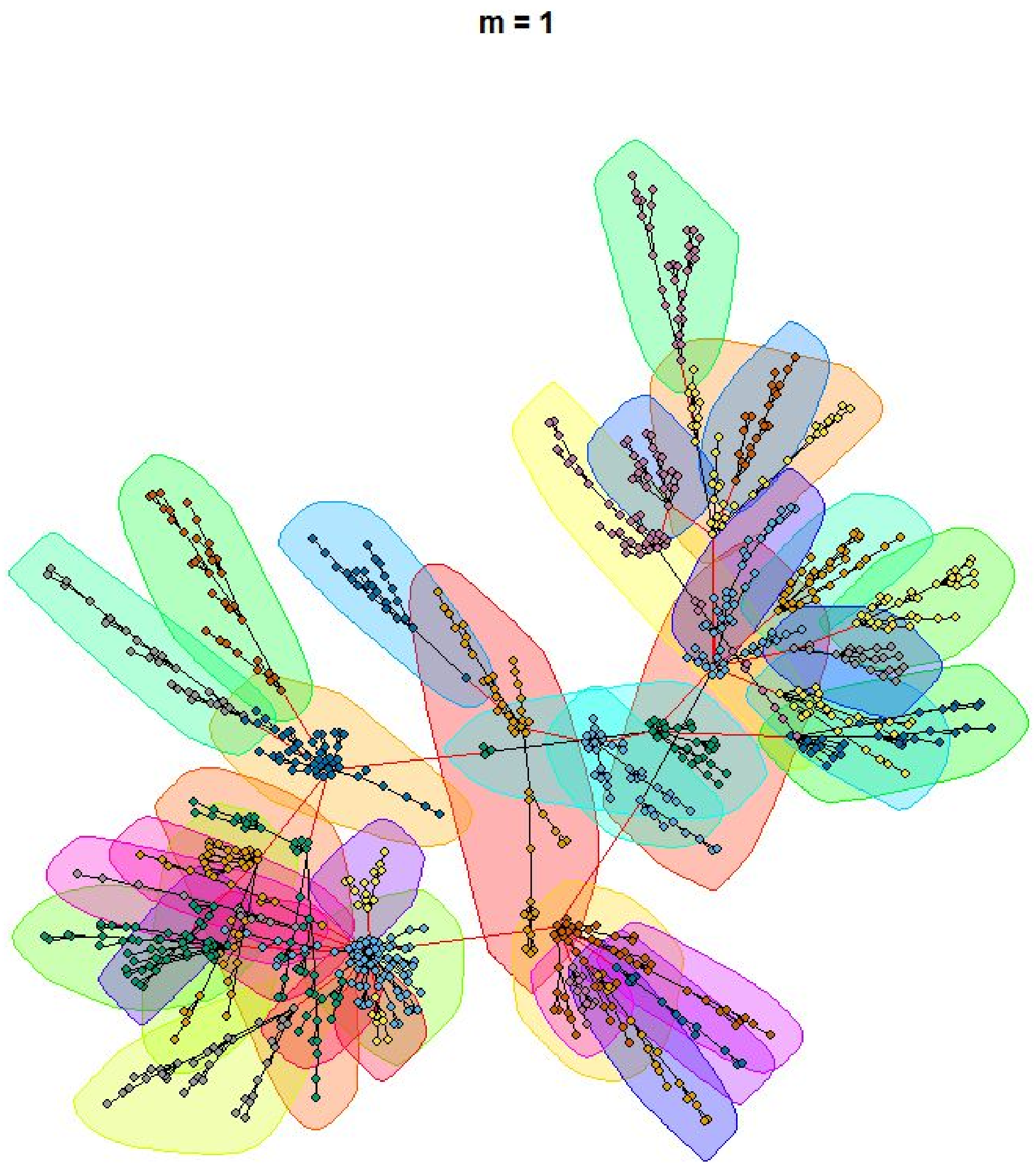

Figure 1.2.7 community structure using fast greedy method with m = 1

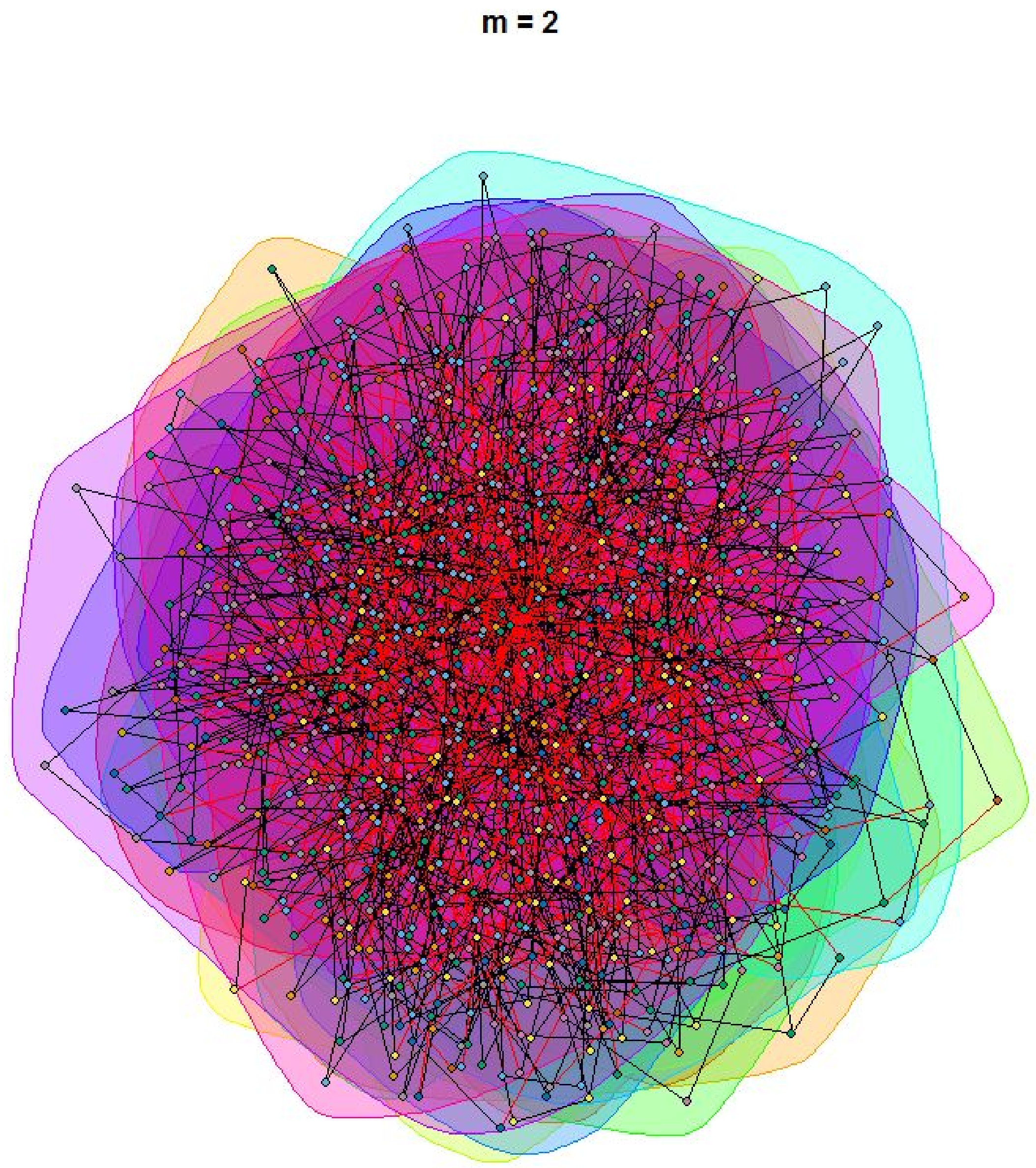

Figure 1.2.8 community structure using fast greedy method with m = 2

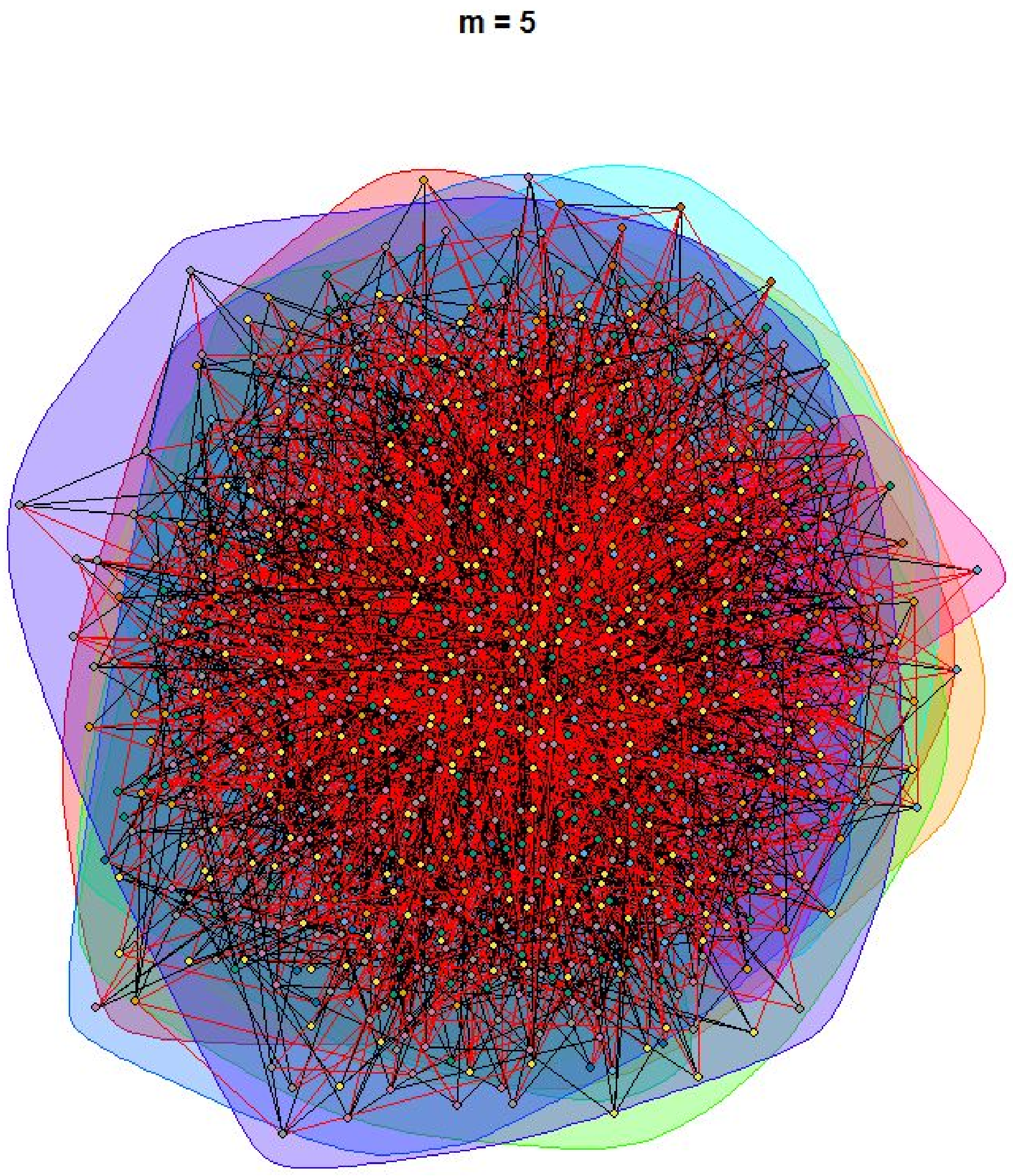

Figure 1.2.9 community structure using fast greedy method with m = 5

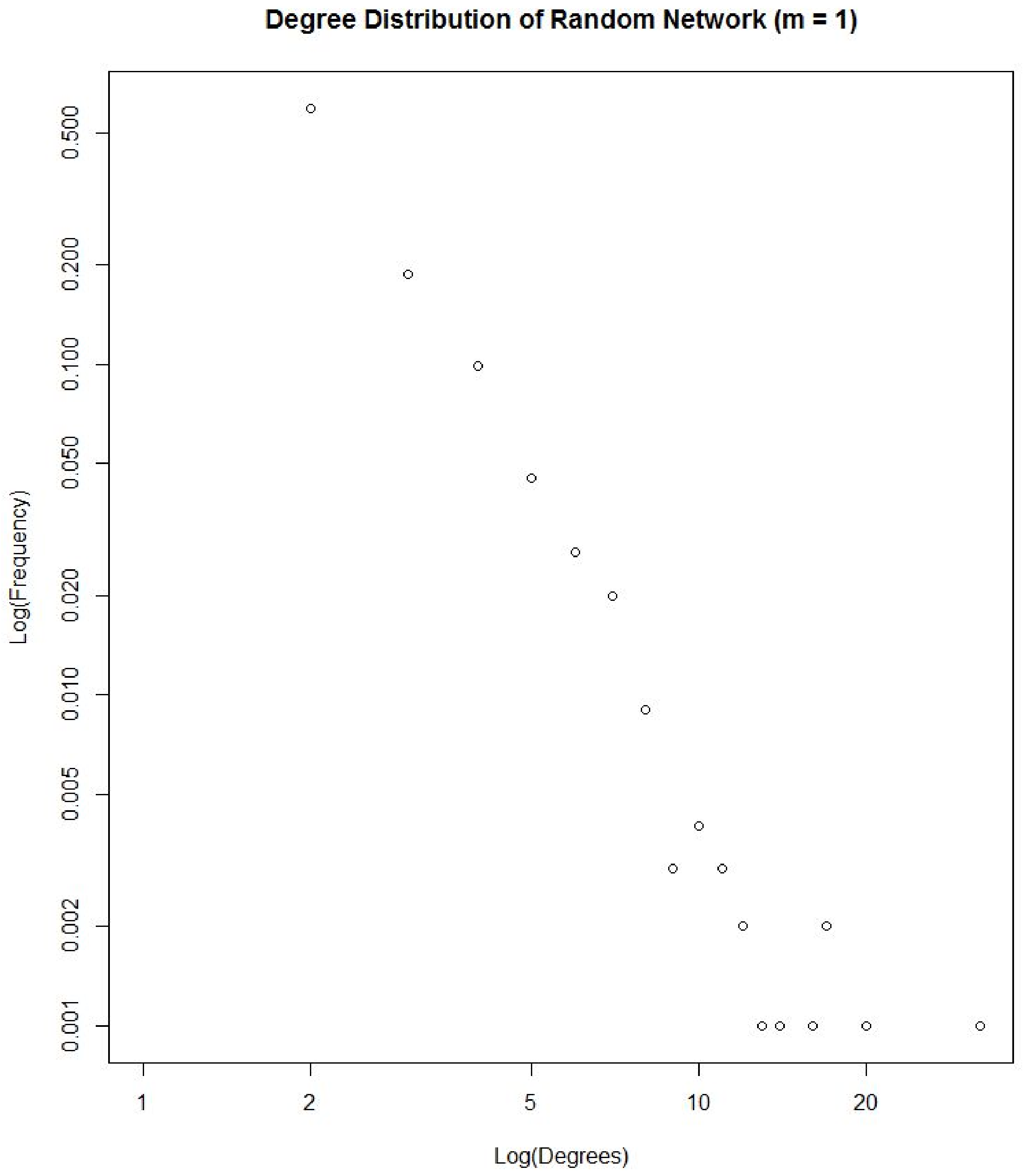

Figure 1.2.10 degree distribution of random network with n = 1000 with m = 1

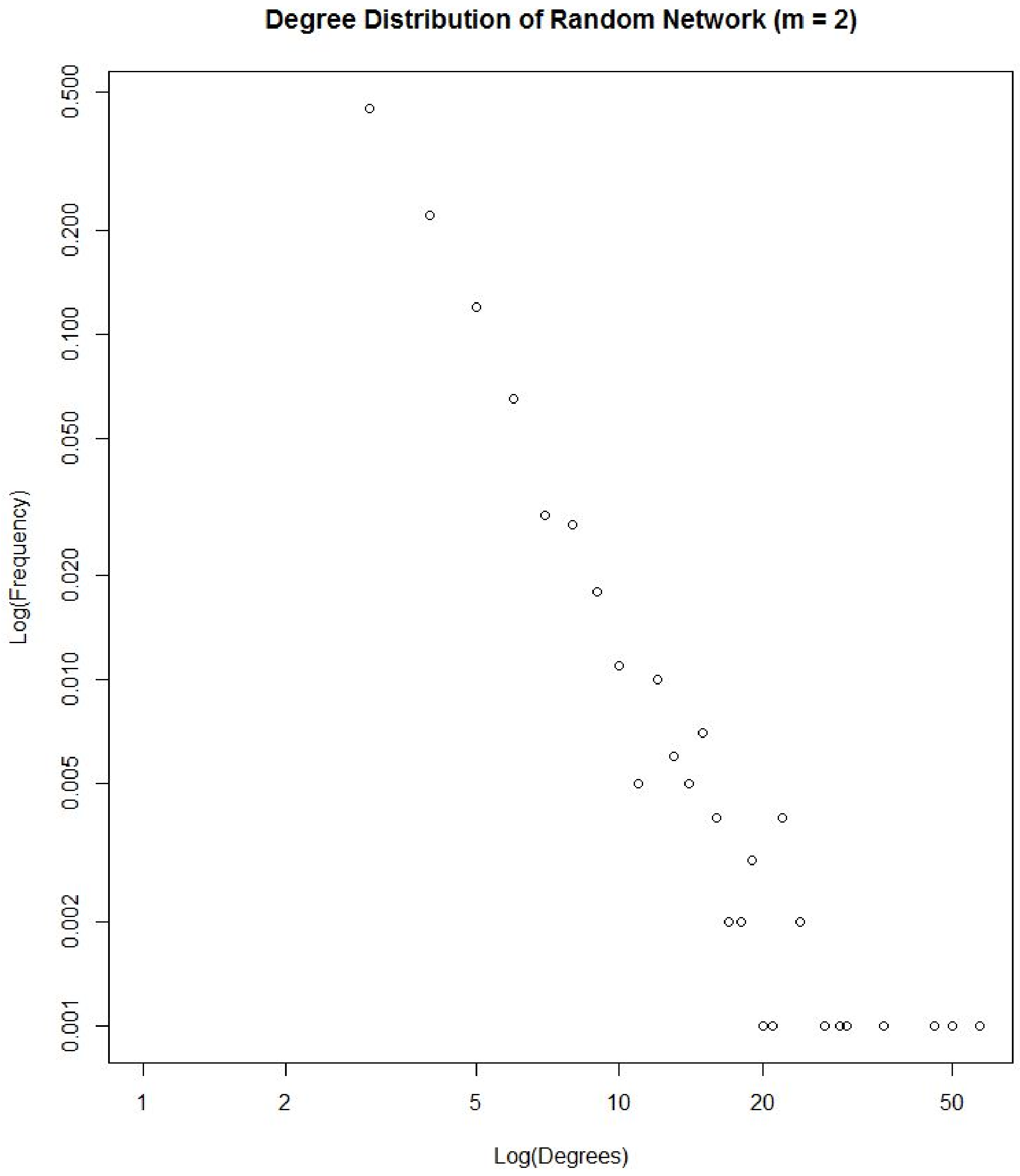

Figure 1.2.11 degree distribution of random network with n = 1000 with m = 2

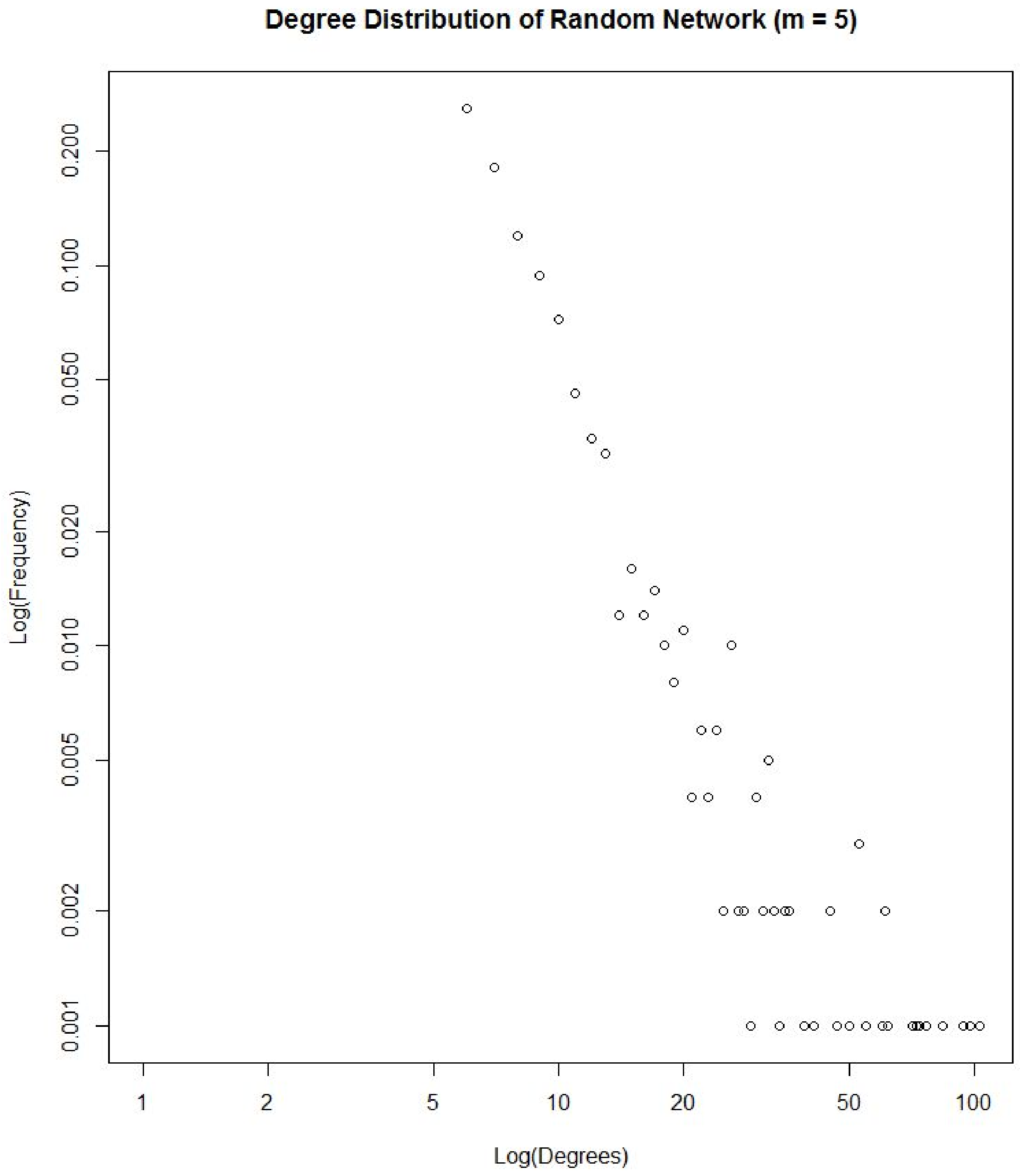

Figure 1.2.12 degree distribution of random network with n = 1000 with m = 5

(h)
We plot a preferential attachment network with n = 1000, m = 1 below. Then take its degree sequence and plot this new network with the same degree sequence below, through stub-matching procedure. The modularity of the original one is 0.9308032. The modularity of the

new network created by stub-matching is 0.847405. The original one has larger modularity which means denser networks between nodes. The new network created by stub-matching procedure has more overlapping area and large GCC size.

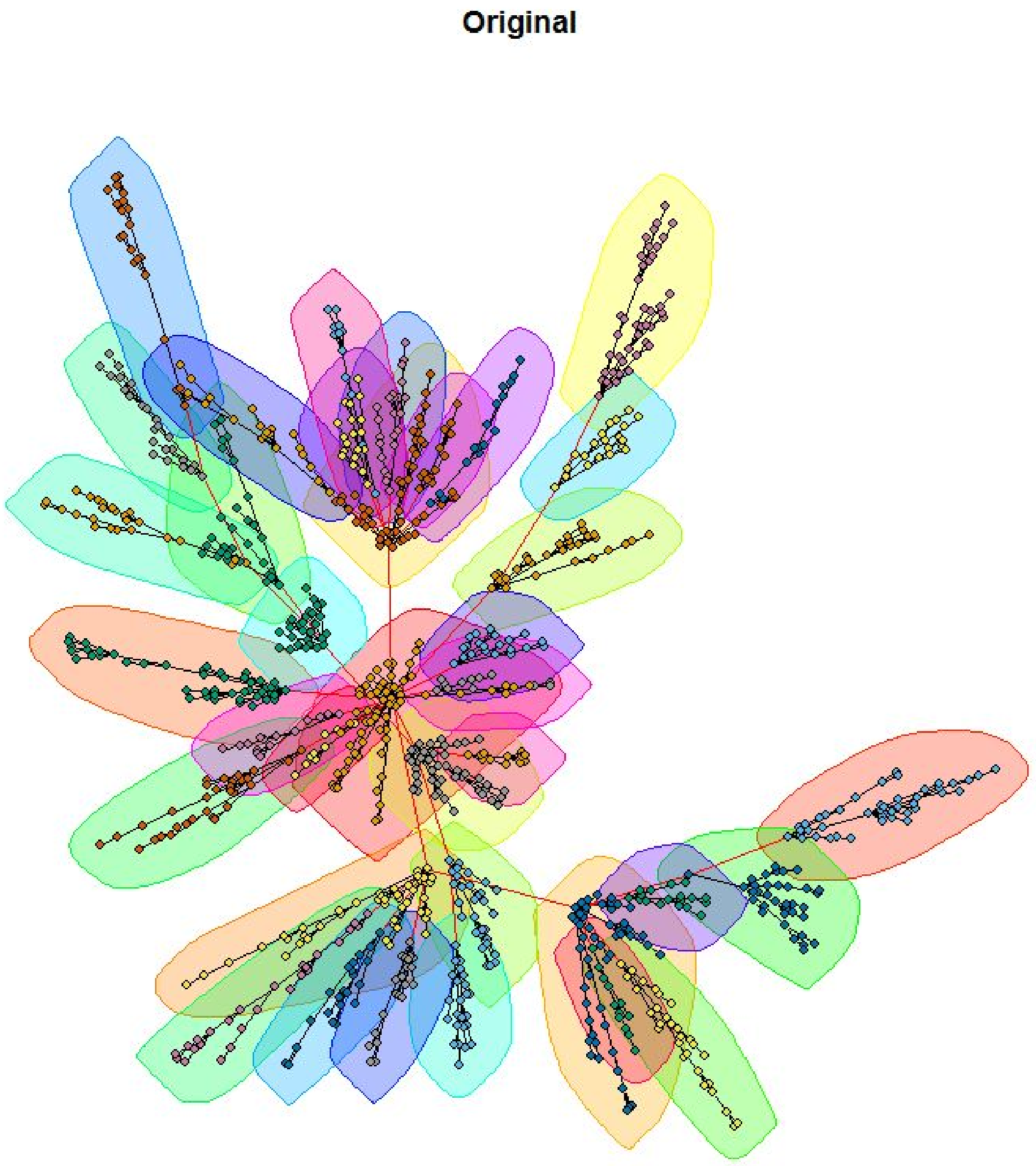

Figure 1.2.13 preferential attachment network with n = 1000, m = 1

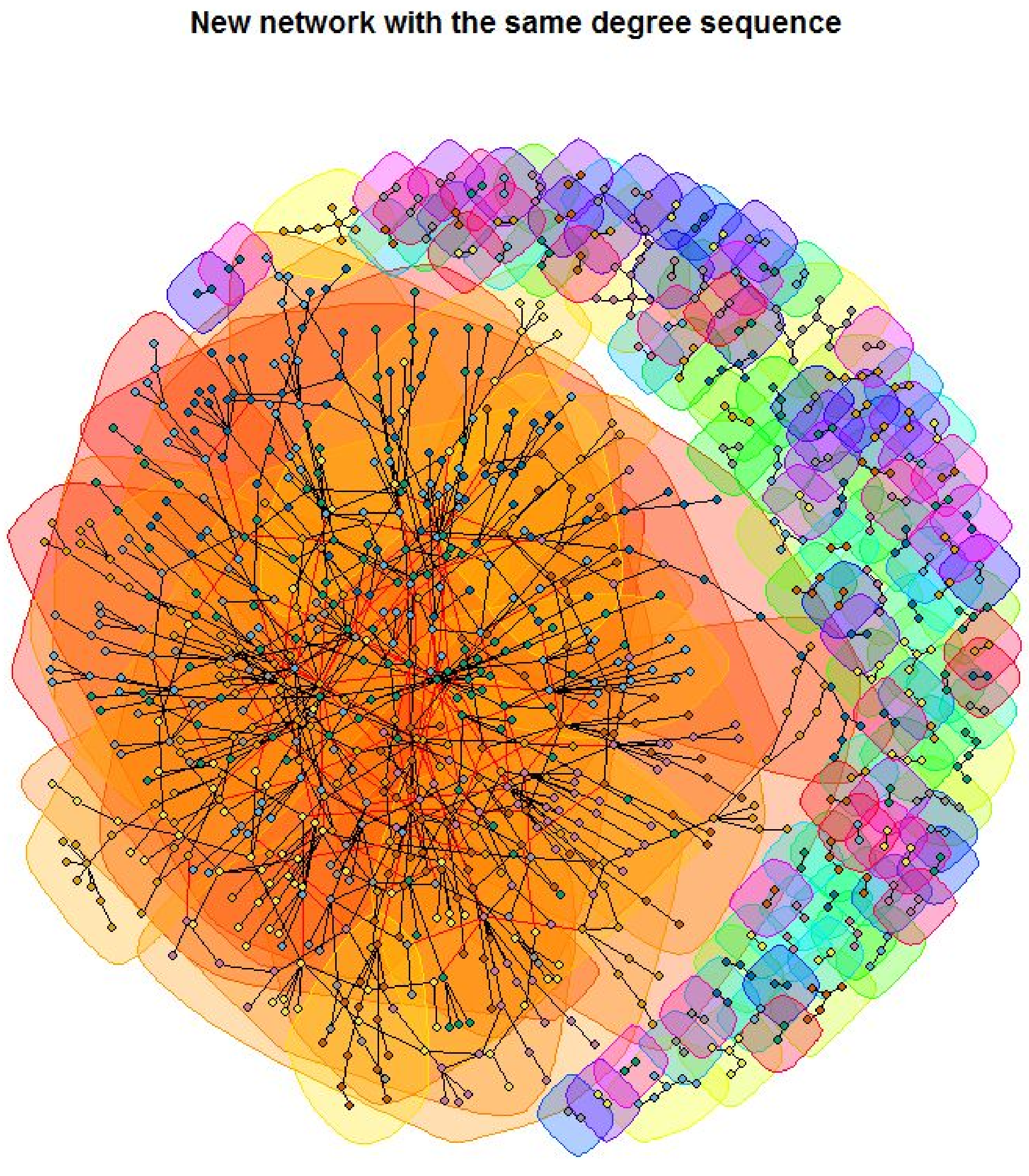

Figure 1.2.14 new network with the same degree sequence through stub-matching procedure

3. Create a modified preferential attachment model that penalizes the age of a node

(a)

We create a modified preferential attachment model, which creates m links to old vertices and the probability that an old vertex is cited depends on its degree (preferential attachment) and age. The probability that a newly added vertex connects to an old vertex is proportional to: P[i]~(c*ki ^A + a)(d*li^B+b) where ki is the degree of vertex i in the current time step, and li is the age of vertex. We plot an undirected network using the above model with 1000 nodes and parameters m = 1, A = 1; B = -1, and a = c = d = 1, b = 0. The power law exponent is 3.

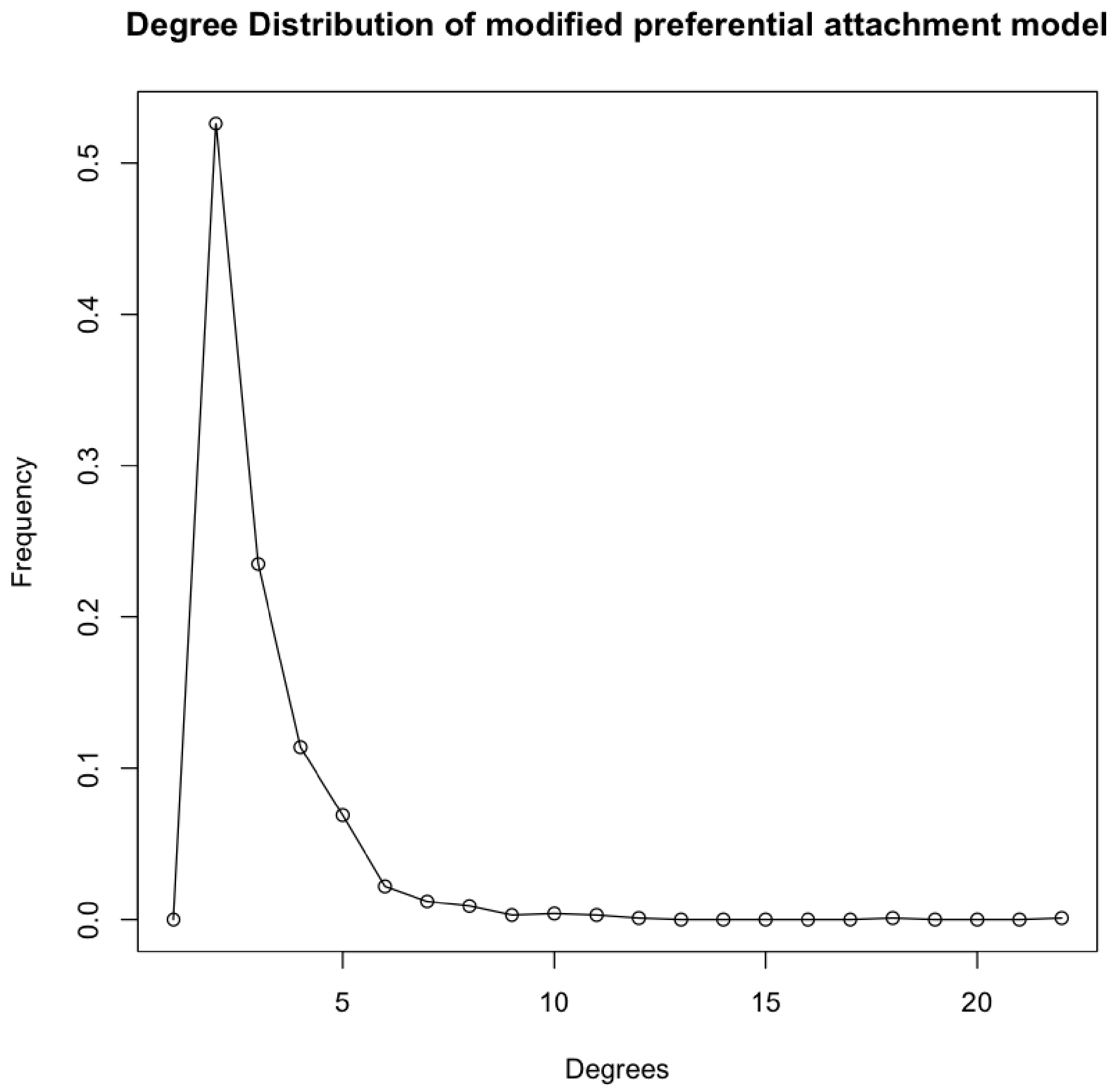

Figure 1.3.1 Degree distribution of modified preferential attachment network

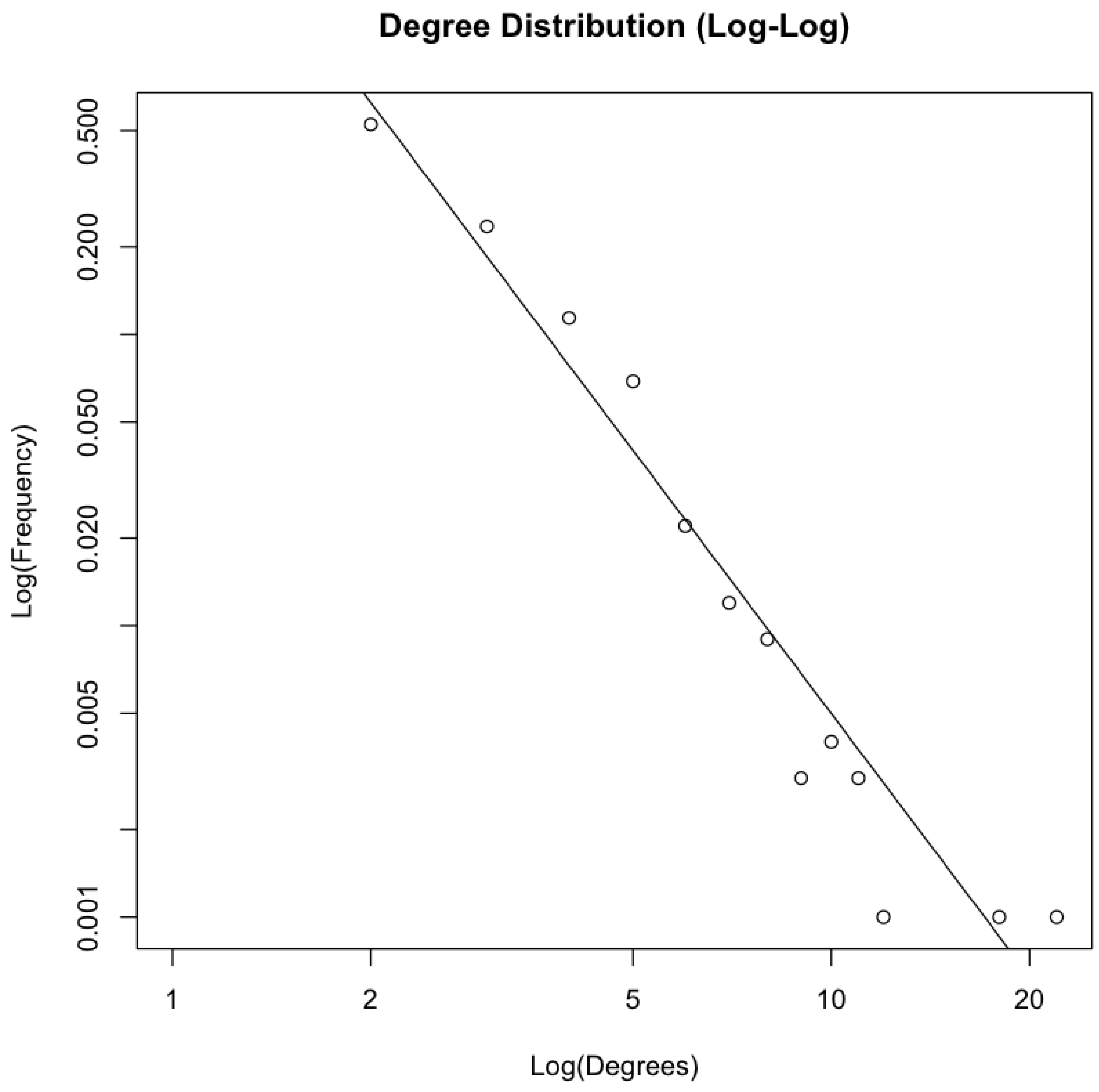

Figure 1.3.2 Log-log scale degree distribution of modified preferential attachment network

(b)

We use fast greedy method to find the community structure, and plot the community below. The modularity is 0.9356679.

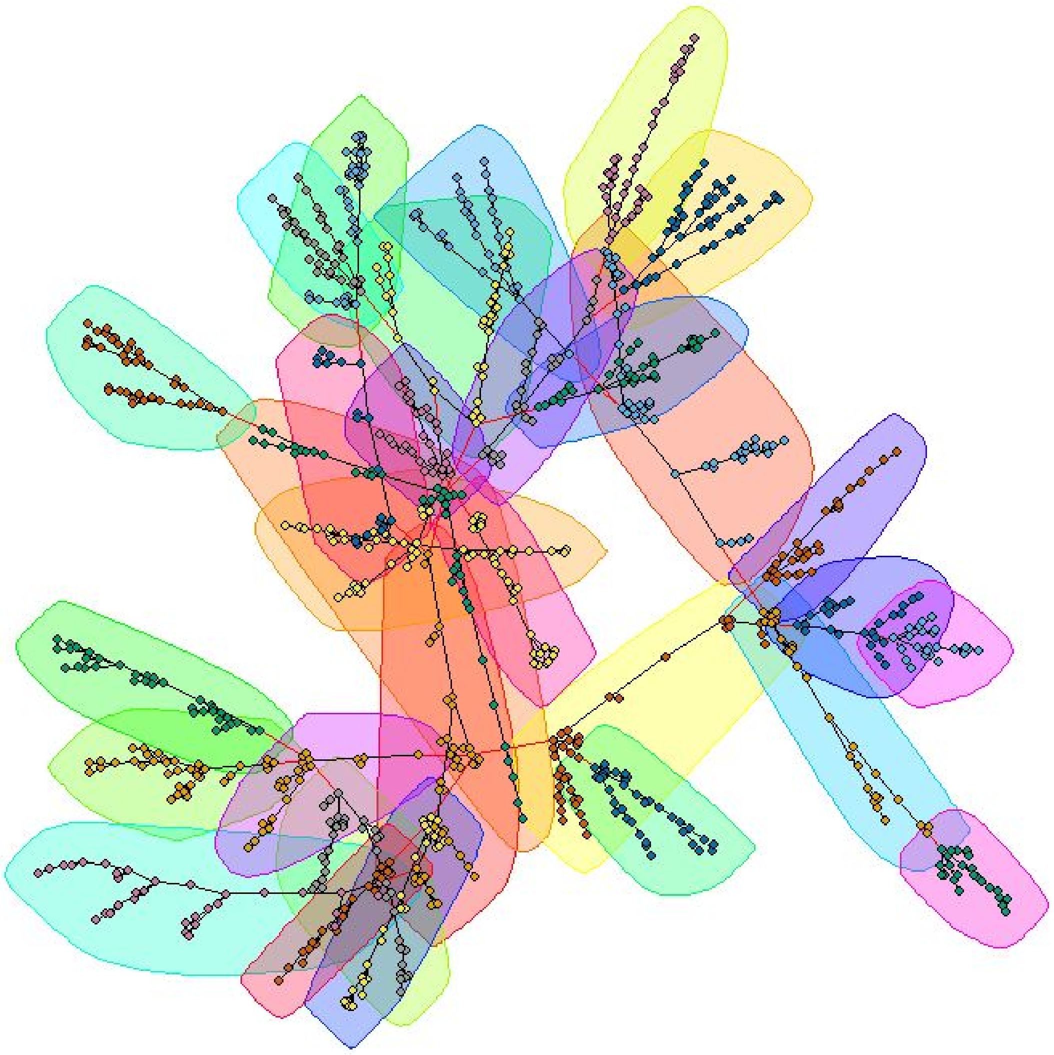

Figure 1.3.3 community structure through fast greedy method

Part 2: Random Walk on Networks

1. Random walk on Erdos-Renyi networks

(a) In this problem, we used the following code to generate the undirected random network with 1000 nodes.

```
g = sample_gnp(n=1000, p=0.01, directed=F)
```

(b)

For this problem, we checked the relationship between the distance (the shortest path length from the start vertex to the end vertex) after the random walk and number of steps that the walker has taken. We iterated the number of steps from 0 to 200, and for each number of step, we tried 200 times. Then, for each step size t, we can plot the mean (Fig. 2.1.1) and variance (Fig. 2.1.2) for each steps sizes.

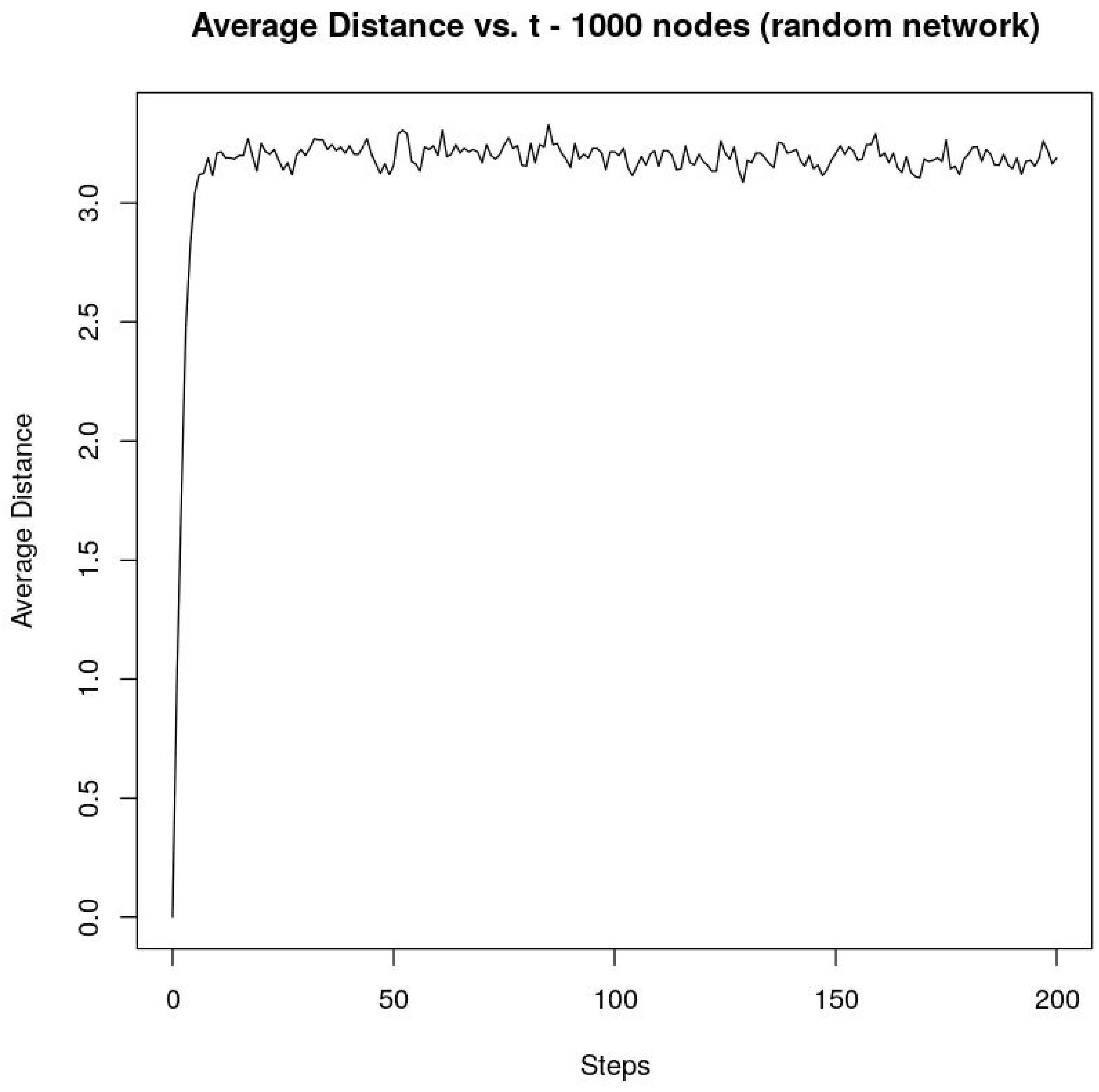

Figure 2.1.1 Mean Value of Distance <s(t)> vs. Number of Steps t (Random Network with 1000 Nodes and p = 0.01)

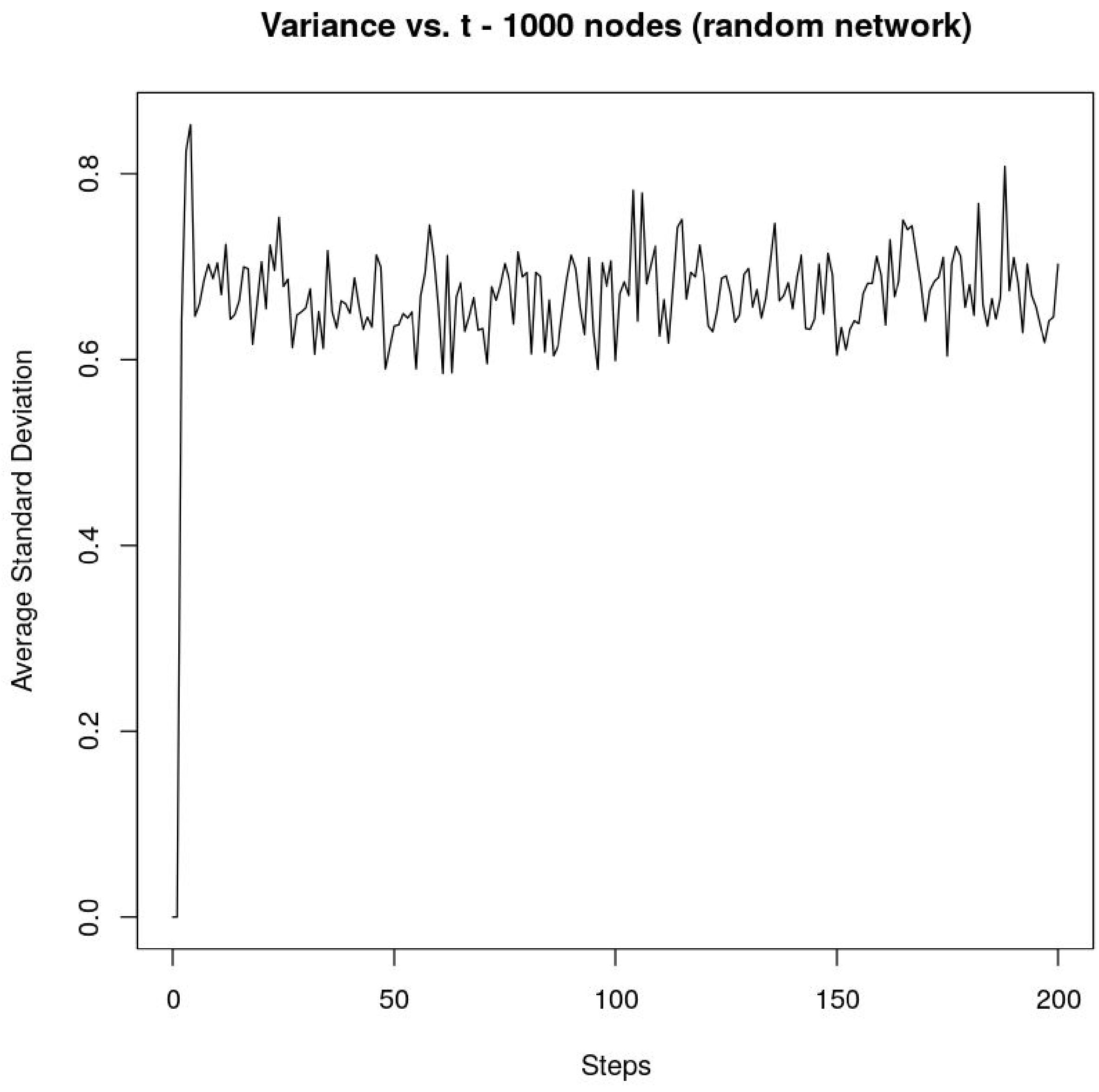

Figure 2.1.2 Variance of Distance vs. Number of Steps t (Random Network with 1000 Nodes and p = 0.01)

(c)

For this problem, we tried 200 times the degree of the last node after a random walk with time step = 200. We found that the degree distribution of graph (Fig. 2.1.3) is similar to the degree distribution of the nodes reached at the end of the random walk (Fig. 2.1.4). The result can be shown as follows”

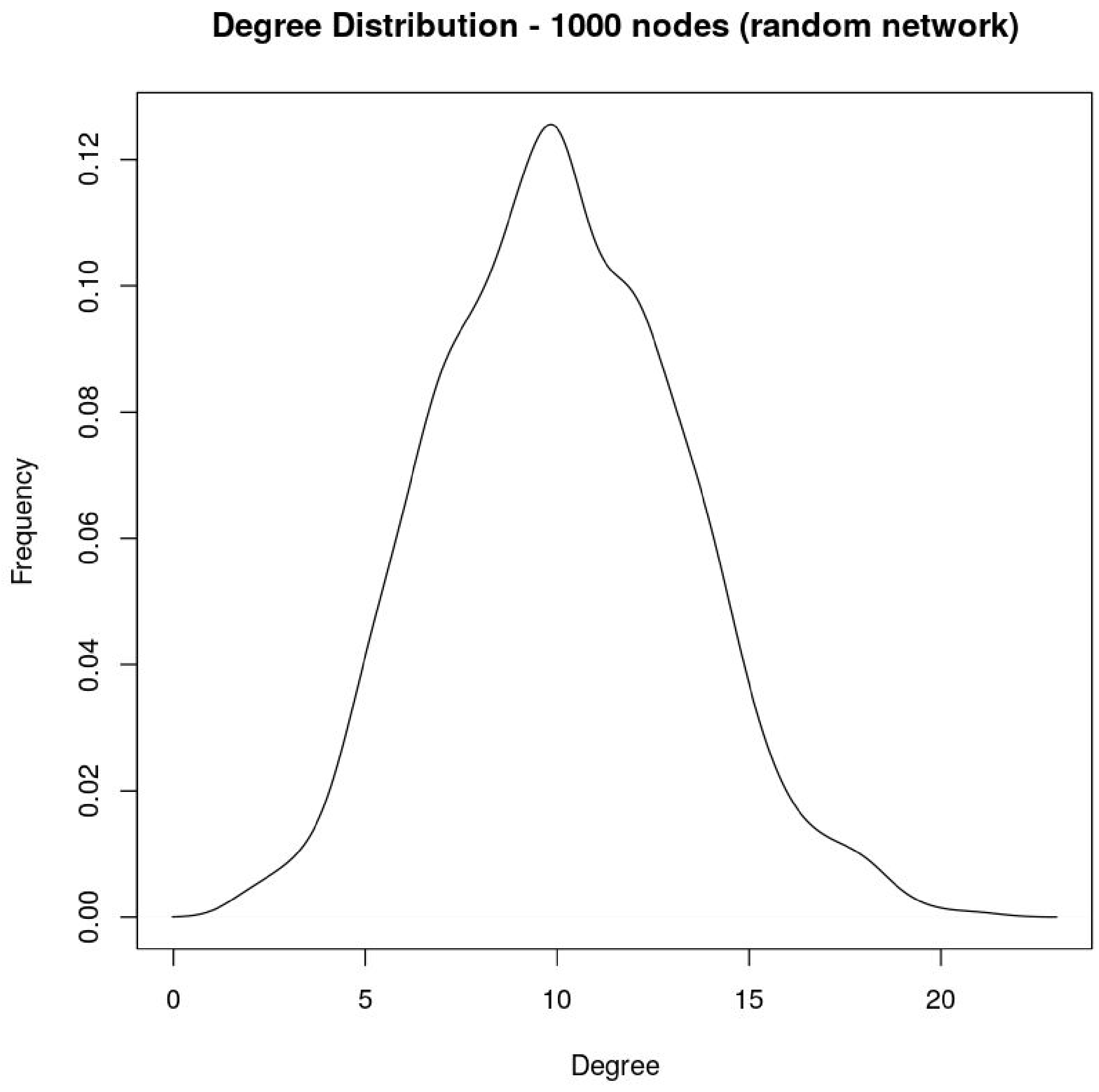

Figure 2.1.3 Degree Distribution An Undirected Random Network with 1000 Nodes (p = 0.01)

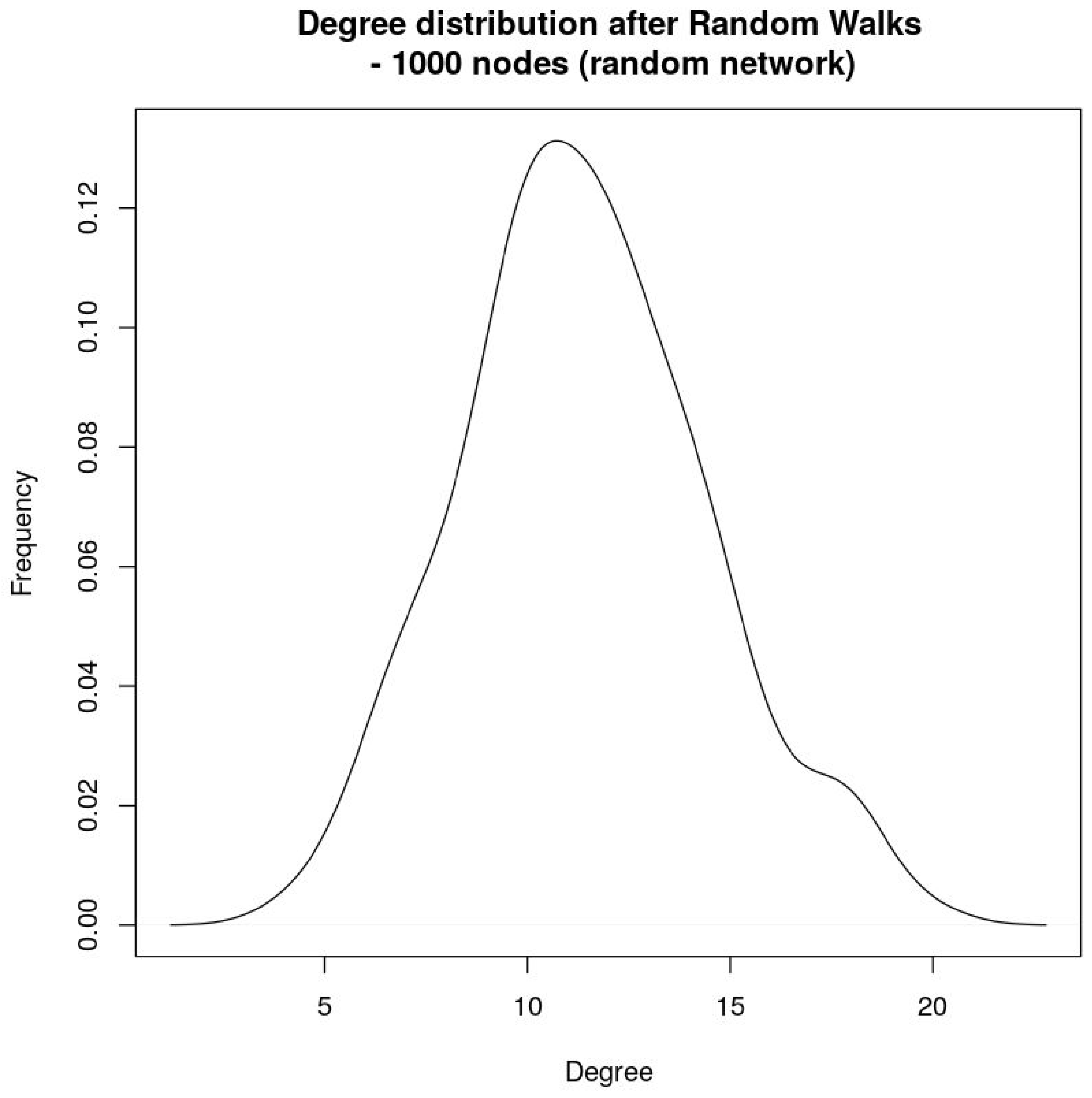

Figure 2.1.4 Degree Distribution of the End Nodes After the Random Walk (Random Network with 1000 Nodes and $p = 0.01$)

It can be shown that the distribution is similar, where they have similar mean and similar variance. This is reasonable because it can be shown that O(ln(n)) steps are enough for the random walk to achieve a state, where the node occupancy probability is the same as the node-picked probability.

(d)

For this problem, we change the node from 1000 to 10000, and the average distances (Fig. 2.1.5) and variance (Fig. 2.1.6) versus time steps can be shown as follows:

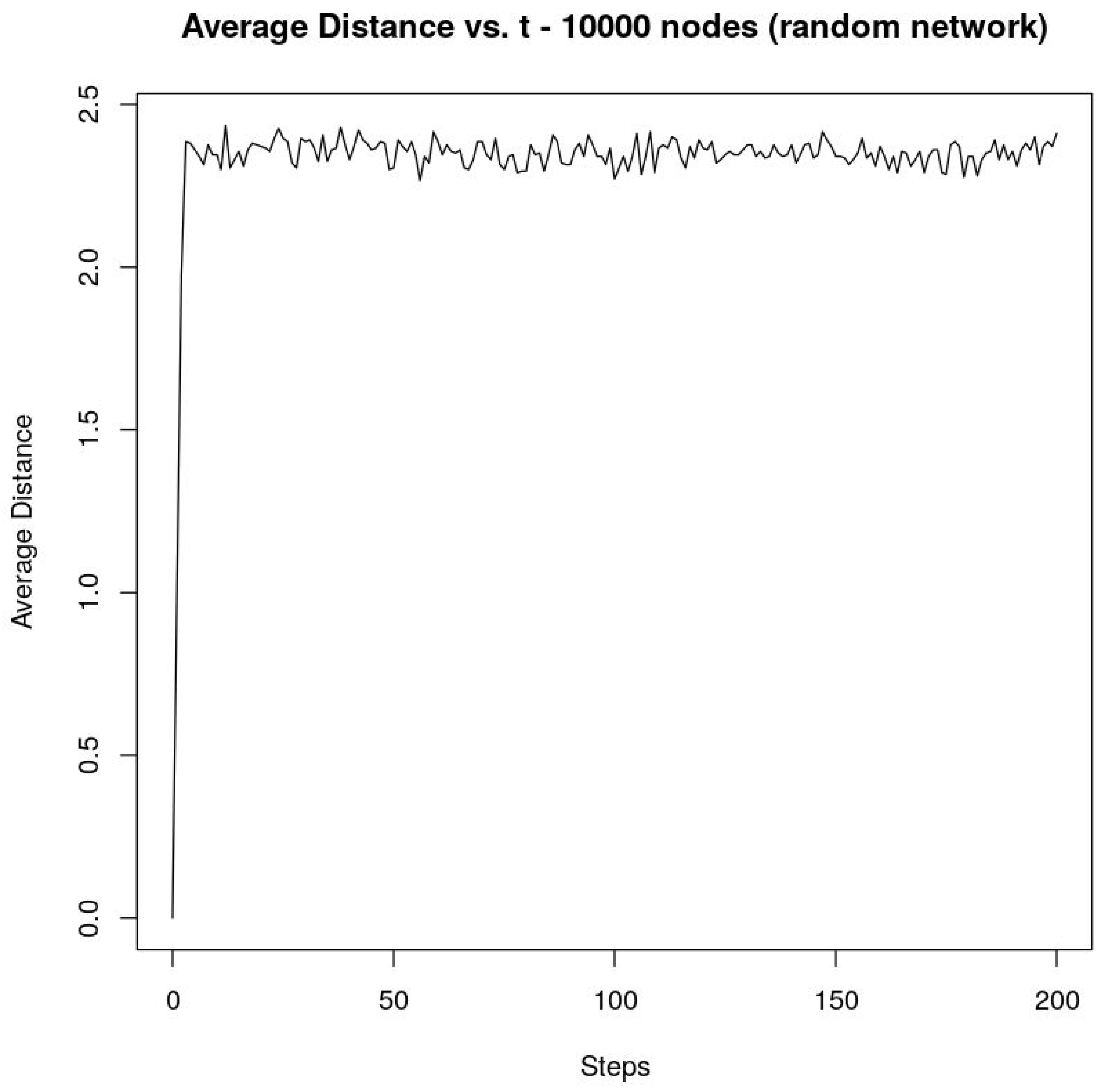

Figure 2.1.5 Mean Value of Distance <s(t)> vs. Number of Steps t (Random Network with 10000 Nodes and p = 0.01)

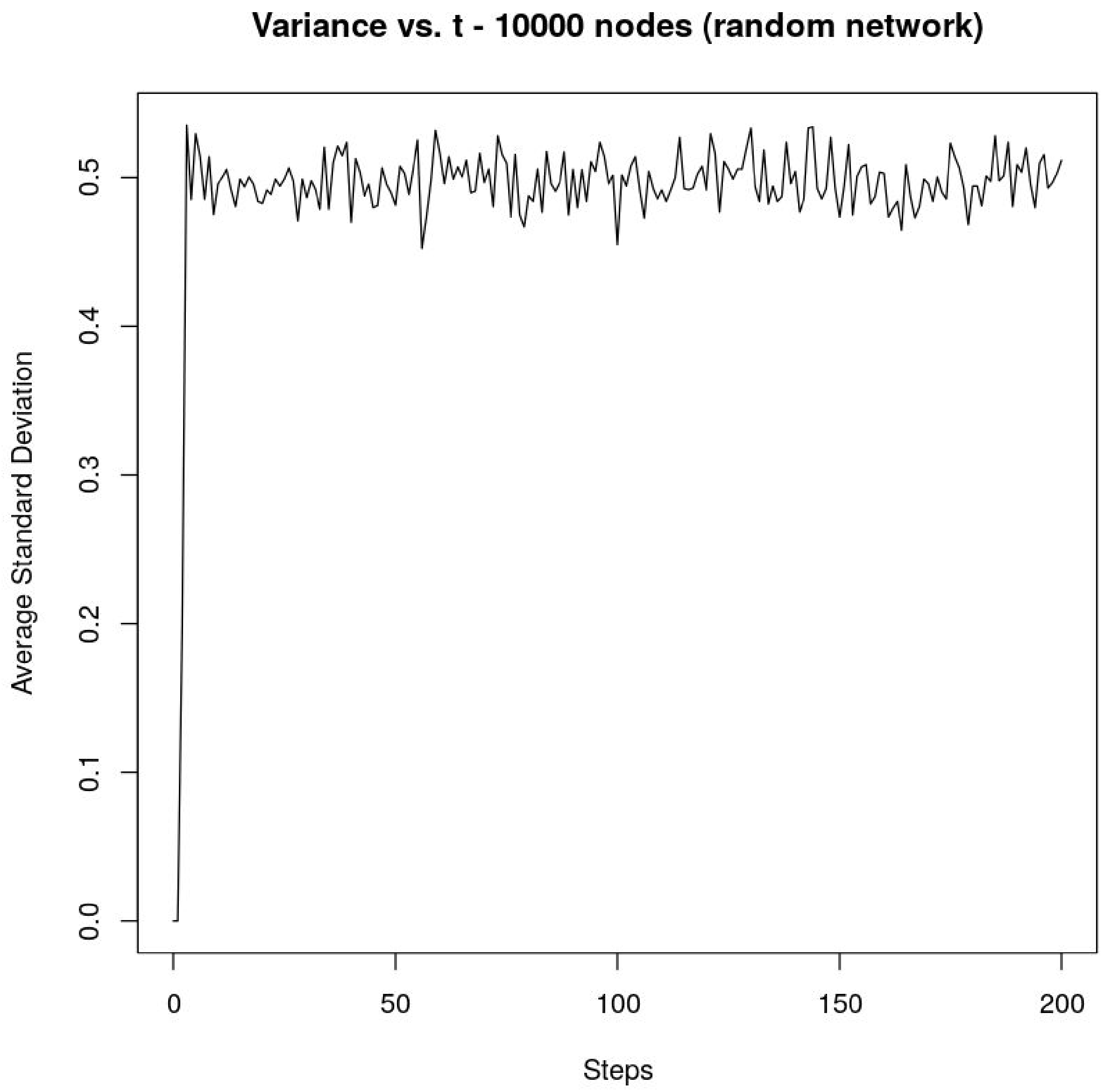

Figure 2.1.6 Variance of Distance vs. Number of Steps t (Random Network with 10000 Nodes and p = 0.01)

By comparing the result with graph with 1000 nodes, we found that the average distances and variance are both decreasing. We then checked the diameter of two graphs, and found that the diameter for graph with 1000 nodes is 5 and the diameter for graph with 10000 nodes is 3.

Thus, the diameter of the network does play a role for random network. The larger the diameter is, the larger the average distance and variance after the random walk are.

2. Random walk on networks with fat-tailed degree distribution

(a) In this problem, we used the following code to generate an undirected preferential attachment network with 1000 nodes, where each new node attaches to m = 1 old nodes.

```
pa_grapgh1000 = barabasi.game(1000, 1, directed = F)
```

(b)

For this problem, we also checked the relationship between the distance (the shortest path length from the start vertex to the end vertex) after the random walk and number of steps that the walker has taken. We iterated the number of steps from 0 to 200, and for each number of step, we tried 200 times. Then, for each step size t, we can plot the mean (Fig. 2.2.1) and variance (Fig. 2.2.2) for each steps sizes.

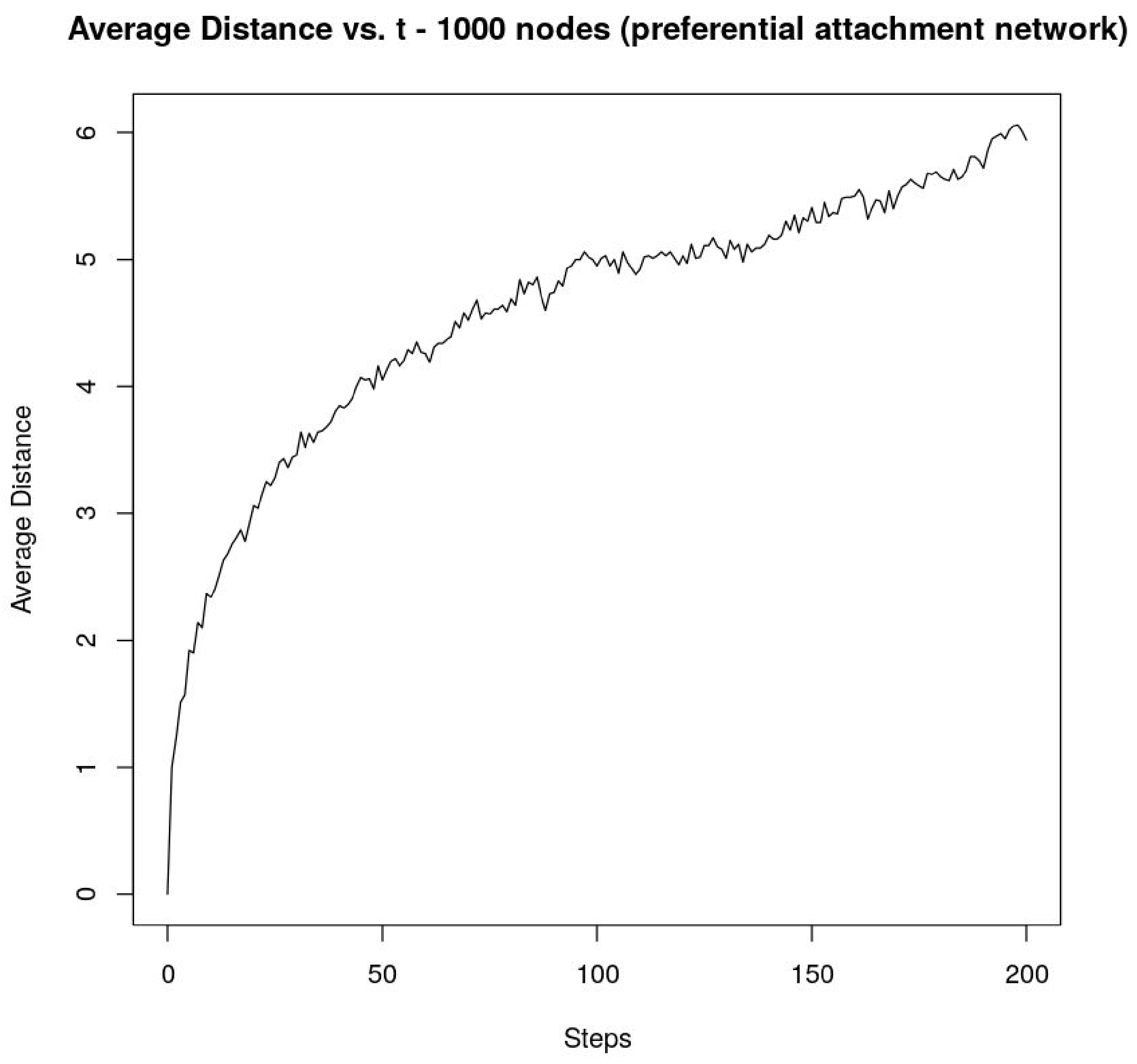

Figure 2.2.1 Mean Value of Distance <s(t)> vs. Number of Steps t (Preferential Attachment Network with 1000 Nodes and m = 1)

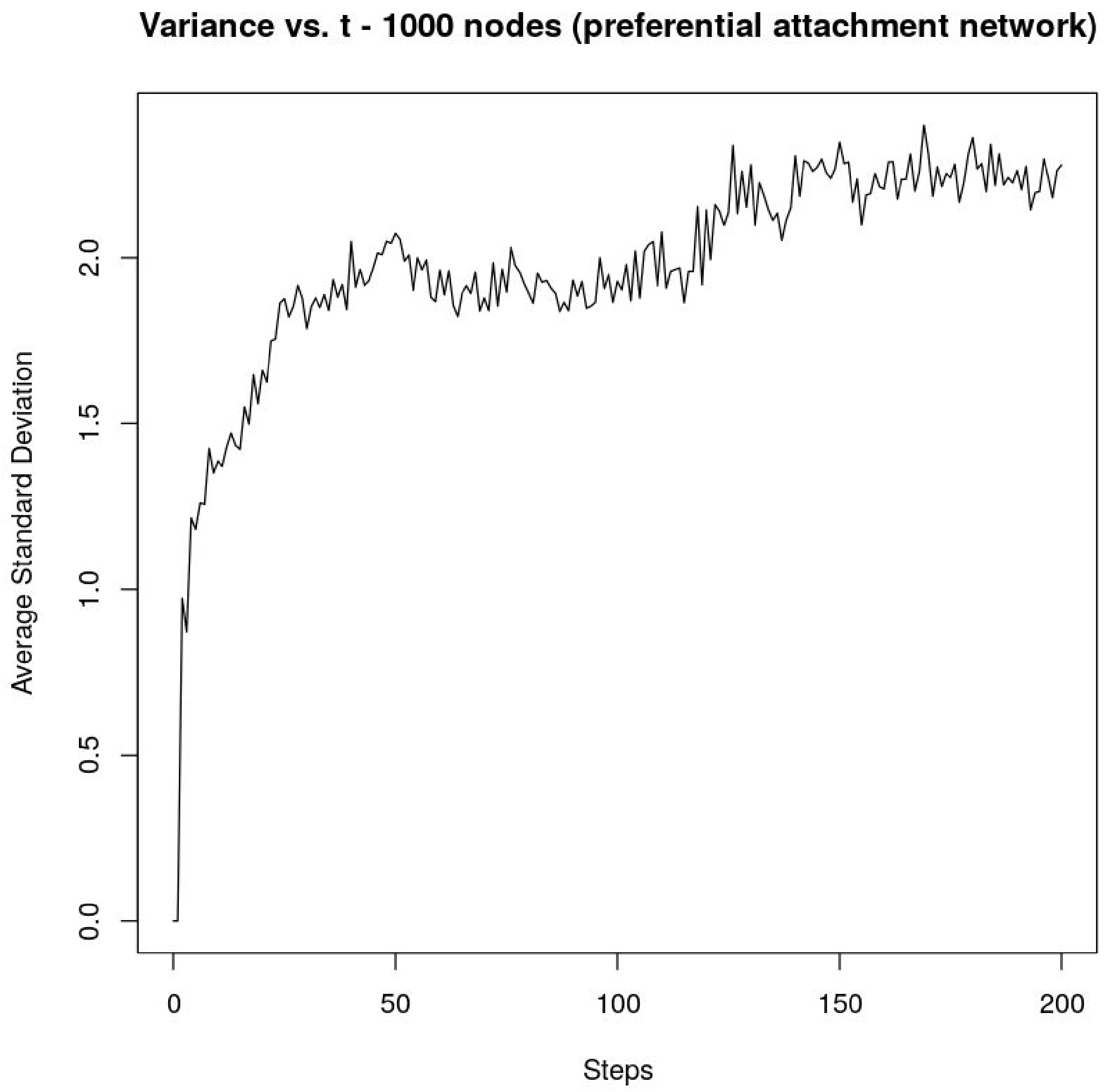

Figure 2.2.2 Variance of Distance vs. Number of Steps t (Preferential Attachment Network with 1000 Nodes and m = 1)

(c)

For this problem, we tried 200 times the degree of the last node after a random walk with time step = 200. We found that the degree distribution of graph (Fig. 2.2.3) is similar to the degree distribution of the nodes reached at the end of the random walk (Fig. 2.2.4). The result can be shown as follows"

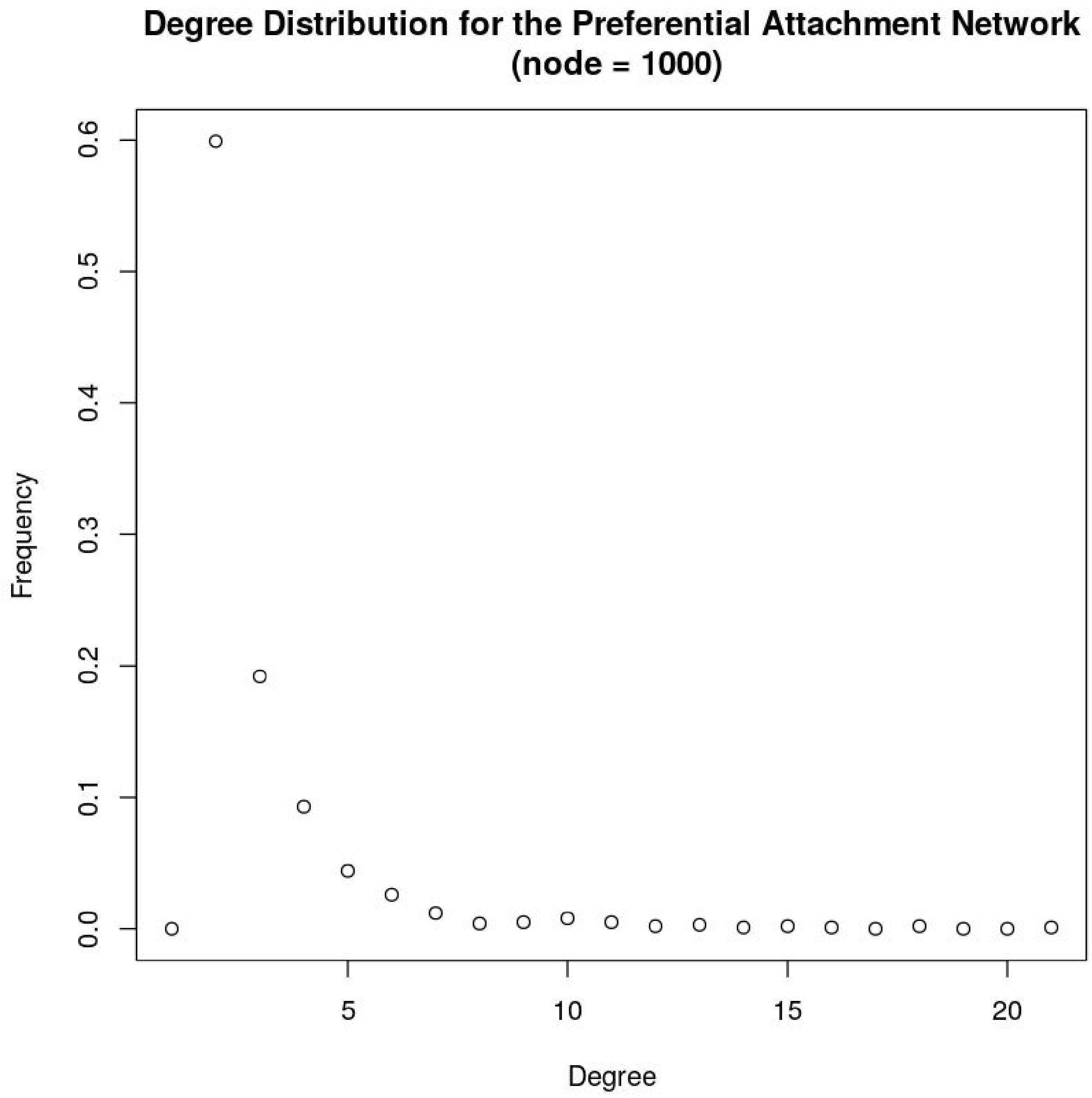

Figure 2.2.3 Degree Distribution a Preferential Attachment Network with 1000 Nodes (m = 1)

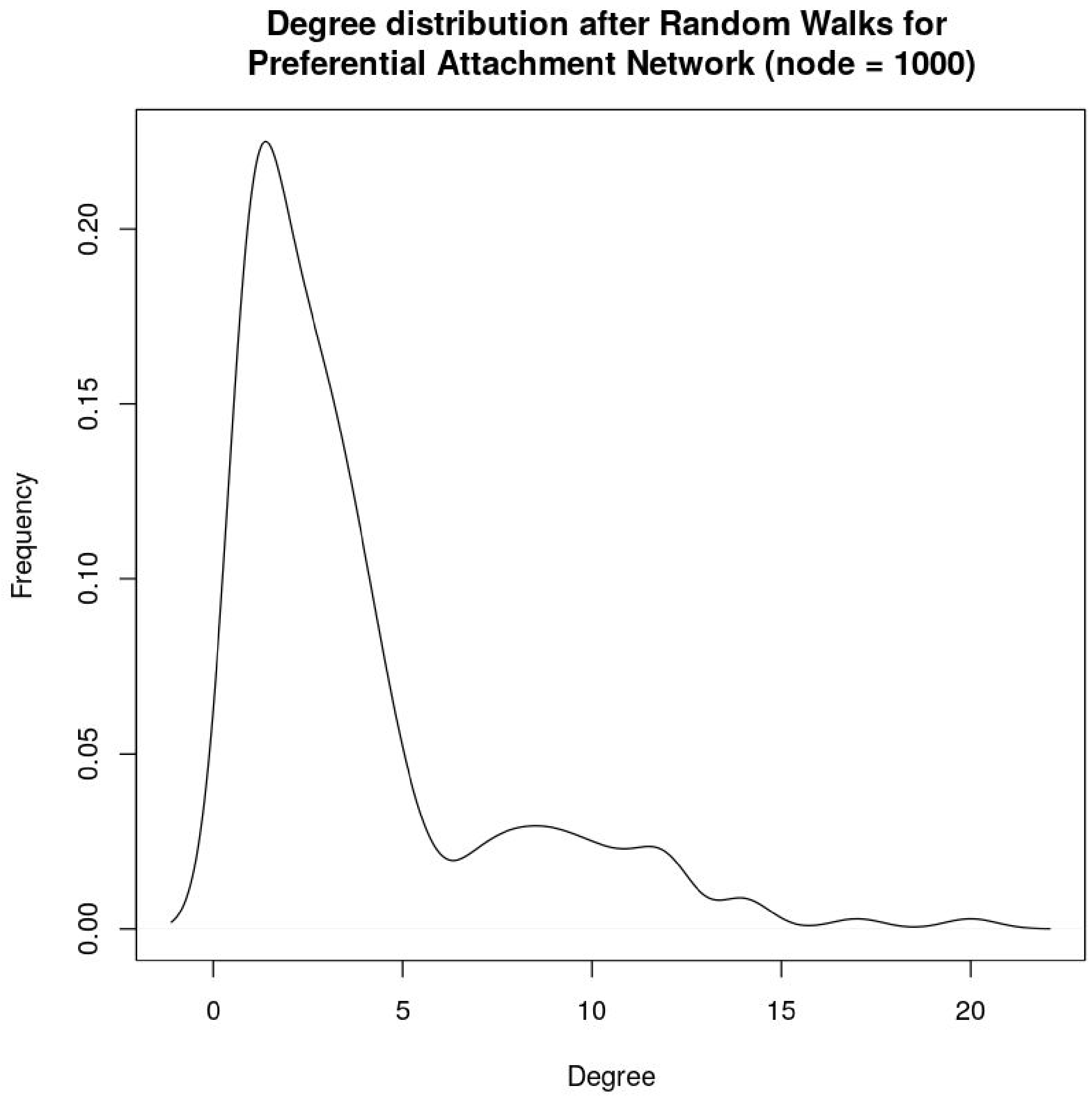

Figure 2.2.4 Degree Distribution of the End Nodes After the Random Walk (Preferential Attachment Network with 1000 Nodes and m = 1)

It can be shown that for preferential attachment network, the distribution is also similar, where they have similar mean and shape.

(d)

For this problem, we change the node from 1000 to 100 and 10000, and the average distances (Fig. 2.2.5) and variance (Fig. 2.2.6) versus time steps can be shown as follows:

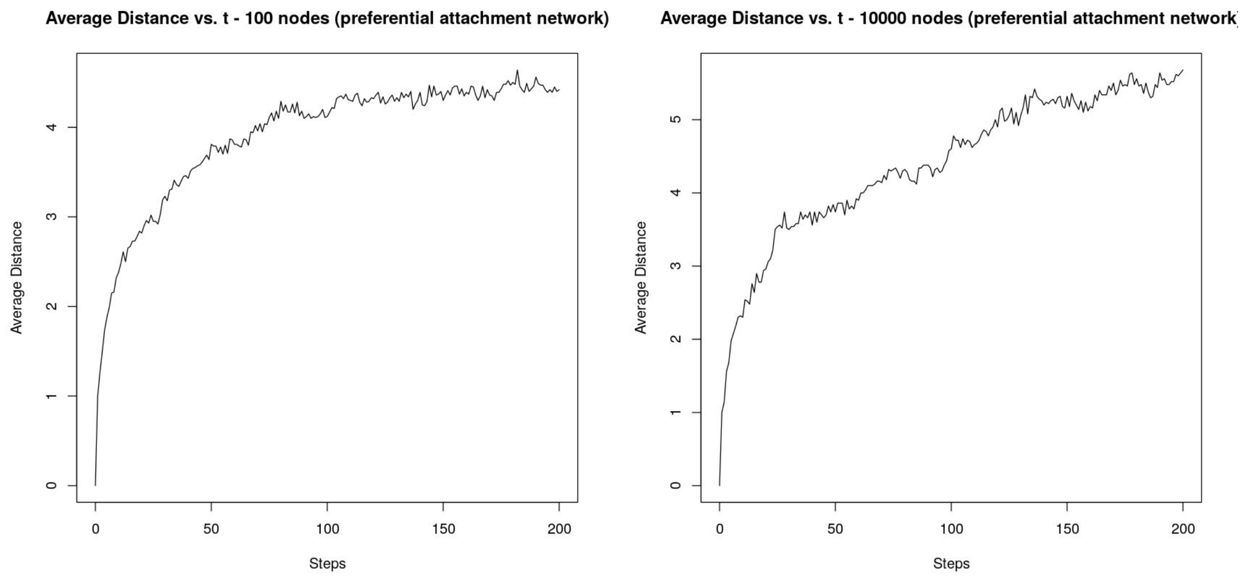

Figure 2.2.5 Mean Value of Distance <s(t)> vs. Number of Steps t (Preferential Attachment Network with 100 Nodes (left) and 10000 Nodes (right))

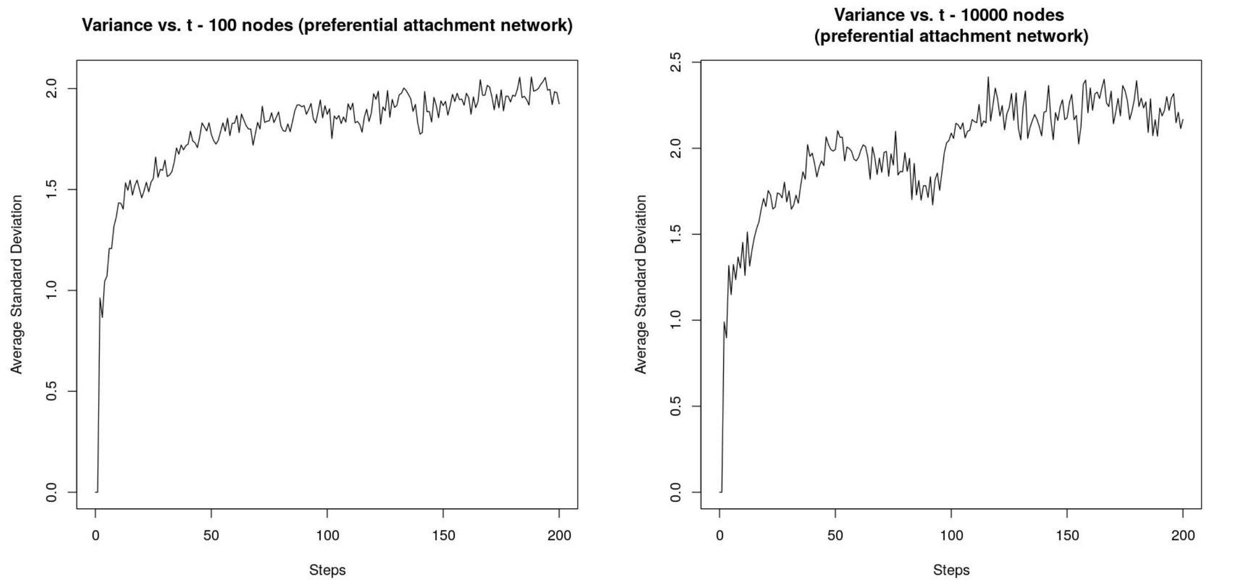

Figure 2.2.6 Variance of Distance vs. Number of Steps t (Preferential Attachment Network with 100 Nodes (left) and 10000 Nodes (right))

We first shows that diameter of each graph in Table 2.2.1 as follows:

Number of Nodes	Diameter
100	15

1000	21
10000	30

Table 2.2.1 The Relationship Between the Number of Nodes of Preferential Attachment Network and Its Corresponding Diameter

From the Table 2.2.1 and the previous graph we showed, it can be shown that the diameter does affect the average distances and the variance of distances, which is the same as the conclusion in Question 2.1.(d). The larger the diameter is, the larger the average distance and variance after the random walk are.

3. PageRank

(a)

For this problem, we used *barabasi.game* function provided by igraph library to create two directed preferential attachment graph, each with 1000 nodes and out-degree 4. Then we permuted the indices of the nodes of second graph and get an edge list. We merged the two graphs by adding the edges got from the second graph into the first graph. In this way, we can avoid the “black hole” problem. After we got the desired network, we constructed the transition matrix for each node and calculated the probability of stopping at each node, which is stored in pr_end variable. We started from each node once and set the number of steps taken starting from each node to be 100. The result is shown below in Fig 2.3.1.

We can see that only a few nodes have large probability to be visited and they are the older nodes. Some nodes that have large node number but still have large probability is because we shuffle the indices of nodes of second graph and merge them into our network.

The probability of each node to be visited versus its in-degree is shown in Fig 2.3.2. From the figure, it is clear that the probability is highly related to in-degree of each node. Typically, the larger one node’s in-degree, the more likely it is visited. Since we only walk 100 steps, some nodes with smaller in-degree may also have higher possibility. When the number of steps get larger, the monotonous relation is more evident.

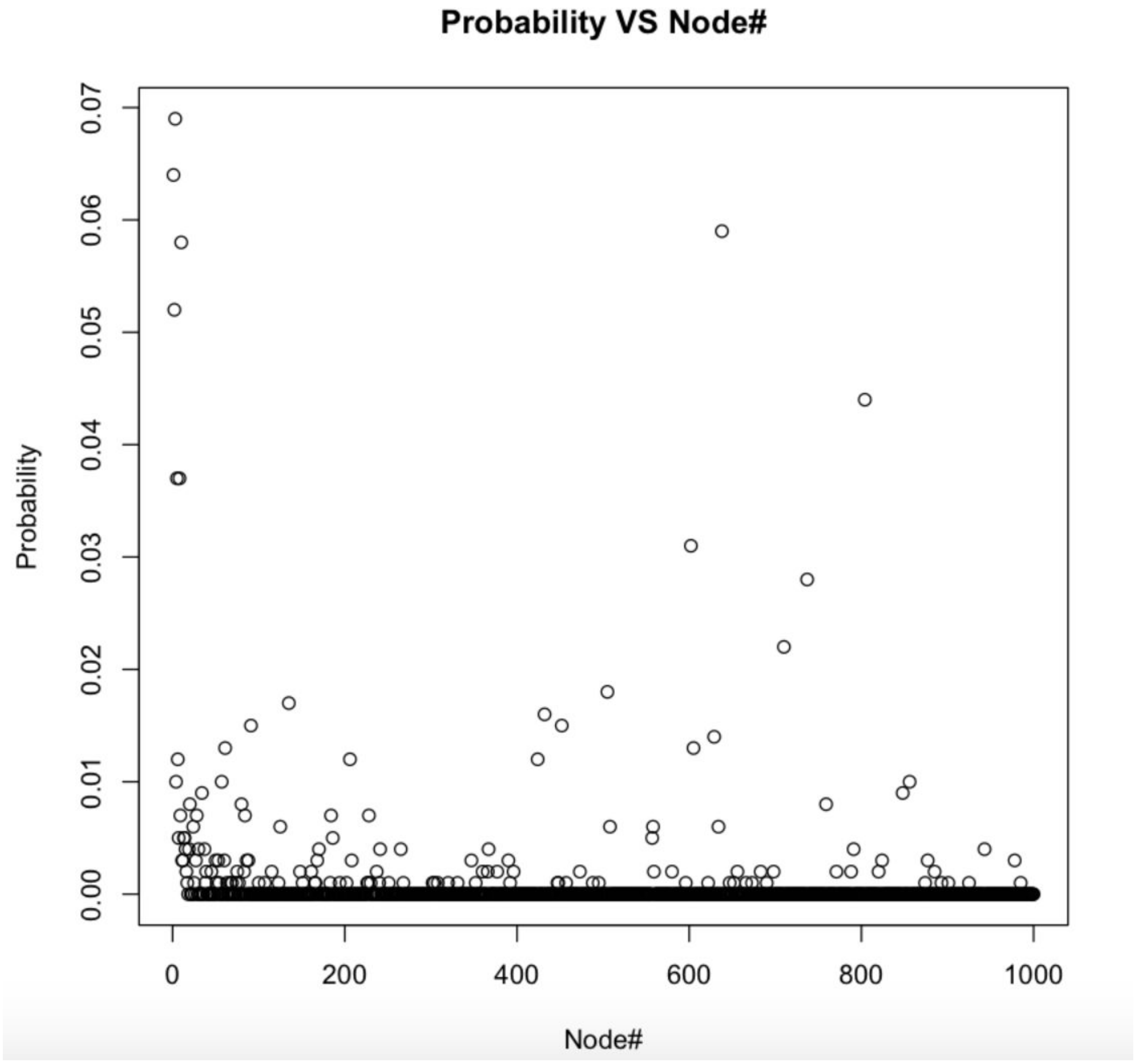

Figure 2.3.1 Probability Versus Node Index Without teleportation

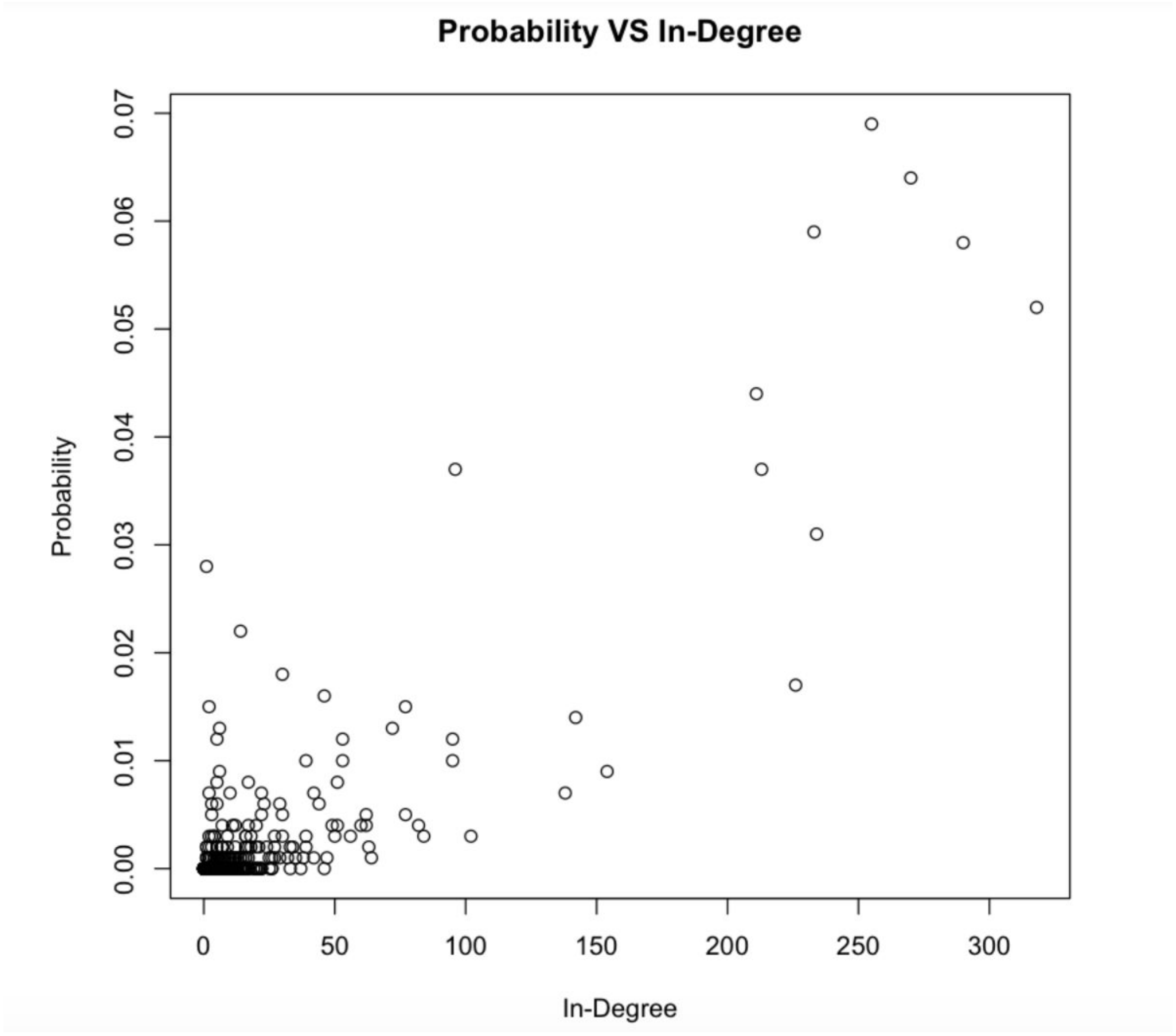

Figure 2.3.2 Probability Versus In-degree Without Teleportation

(b)

In all previous questions, we didn't have any teleportation. Now, we use a teleportation probability of α = 0.15. By performing random walks on the network created in 3(a), measure the probability that the walker visits each node. Is this probability related to the degree of the node?

We set the probability of teleportation to be 0.15 and leave the other settings same as 3(a). In each step of our random walk, we generate a random number between 0 and 1. If it's smaller than 0.15, teleportation happens, with the possibility of transporting to each node to be the same (1/N). Otherwise, it decides to go to which next node the same as 3(a). The result is shown in Fig 2.3.3.

The shape of the figure is almost the same as that in 3(a) because the destination node of every teleportation is of same possibility. The difference is that the probability of visiting the nodes with high probability decreases a bit because of the teleportation.

The probability of visiting each node versus in-degree of nodes is shown in Fig 2.3.4. Still we can see that the probability is highly related to the in-degree and the larger the in-degree of one node, the more likely the node is visited.

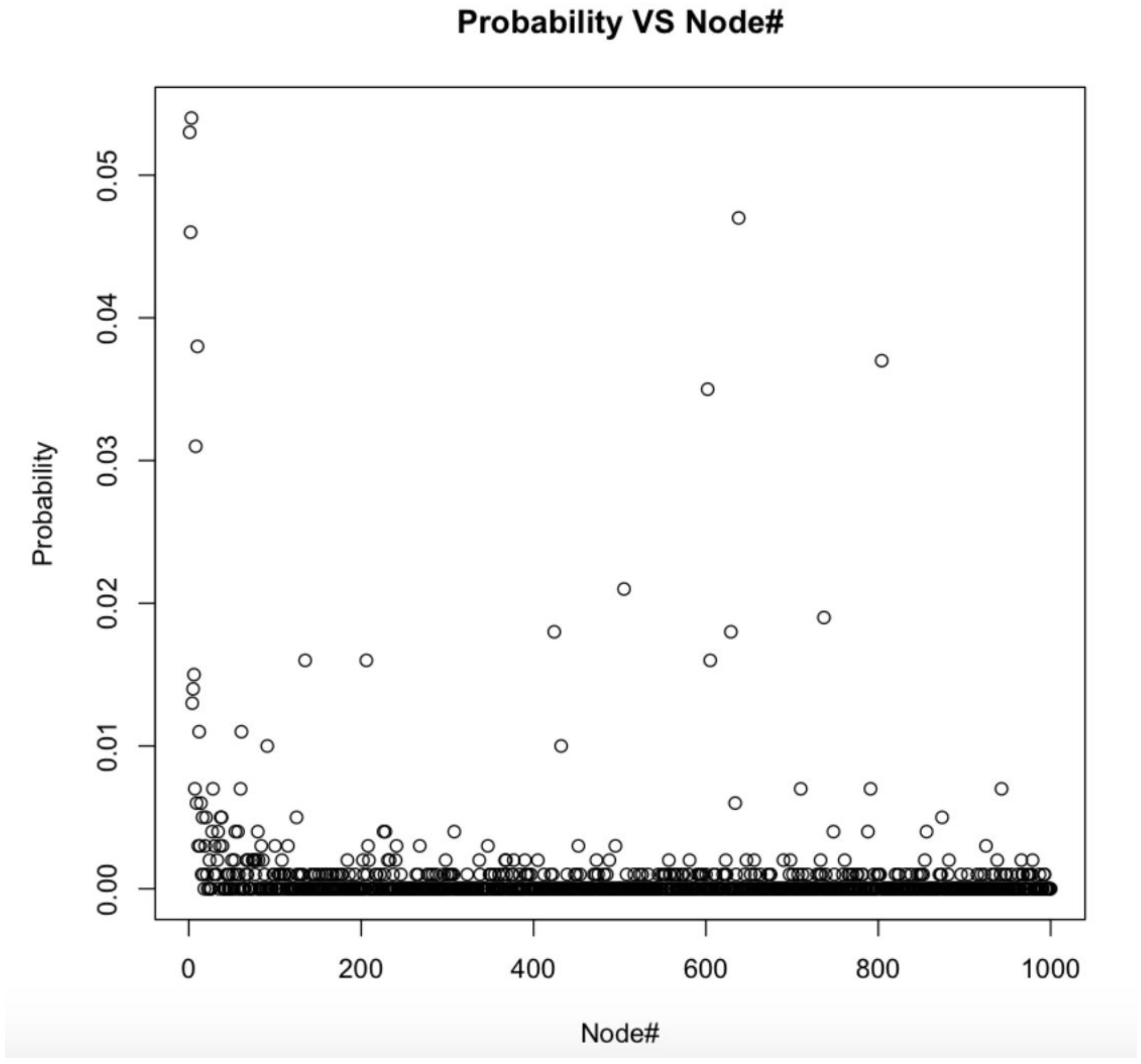

Figure 2.3.3 Probability Versus Node Index With Teleportation - Equal Probability

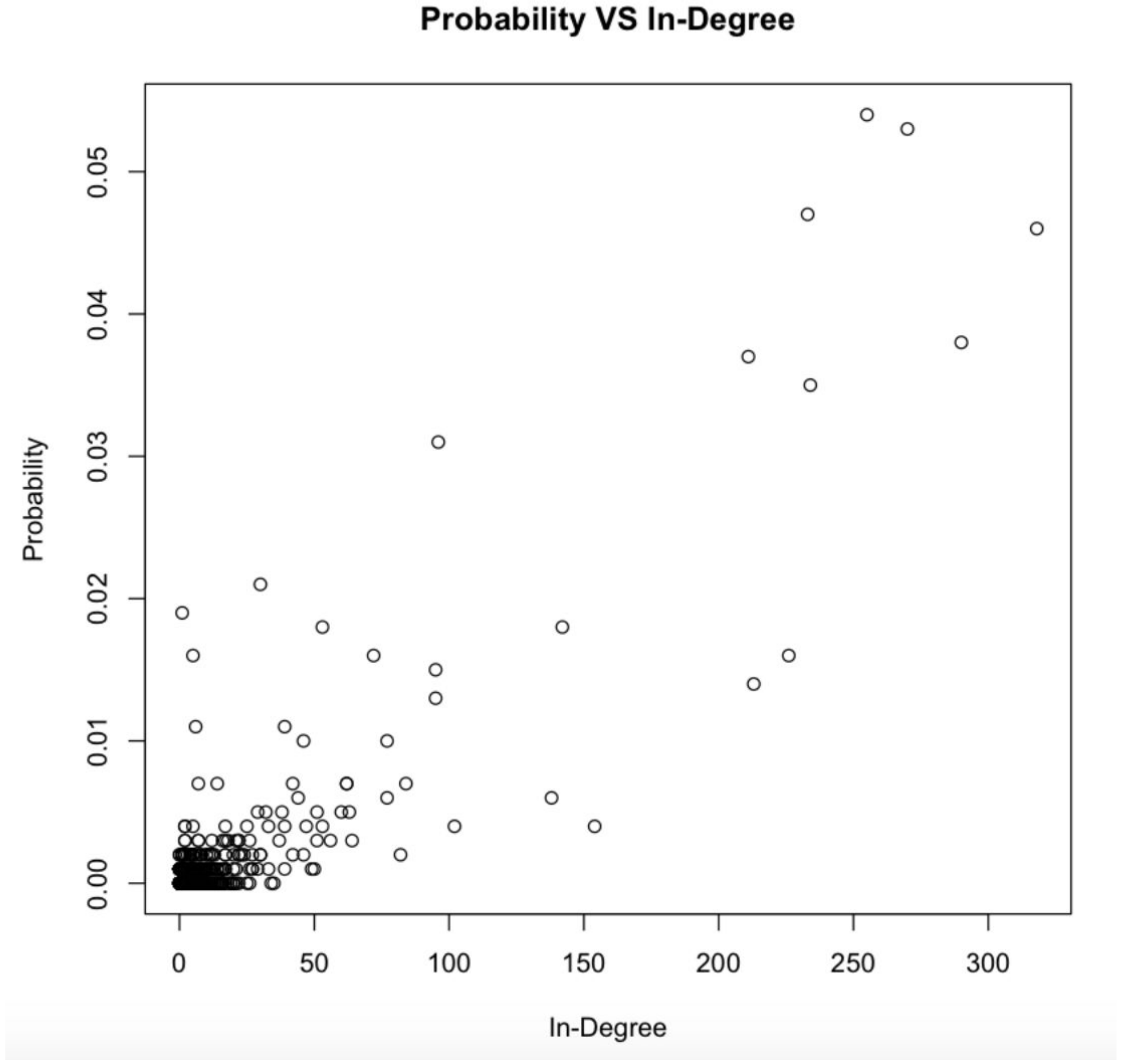

Figure 2.3.4 Probability Versus In-degree With Teleportation - Equal Probability

4. Personalized PageRank

(a)

For this question, we remain all the settings the same as 3(b) except that in case of teleportation, the probability to each node is not uniform (which is 1/N in 3(b)). Instead, we use the PageRank of each node as the probability to teleport to that node. In implementation, we just set the probability to be the PageRank value we get in 3(b).

The probability that the walker visits each node is shown in Fig 2.4.1. There is no large difference between the result in 3(a), with older nodes typically having higher probability.

However, we can see that the value becomes a bit larger for nodes with high probability and the vale becomes smaller for nodes with low probability. Put it another way, interesting nodes get visited more often and less interesting nodes less frequently.

The probability of visiting each node versus the in-degree of that node is shown in Fig 2.4.2. Again, the nodes with larger in-degree usually have higher probability as in 3(a) and similarly, the nodes visited are more polarized as discussed in the previous paragraph.

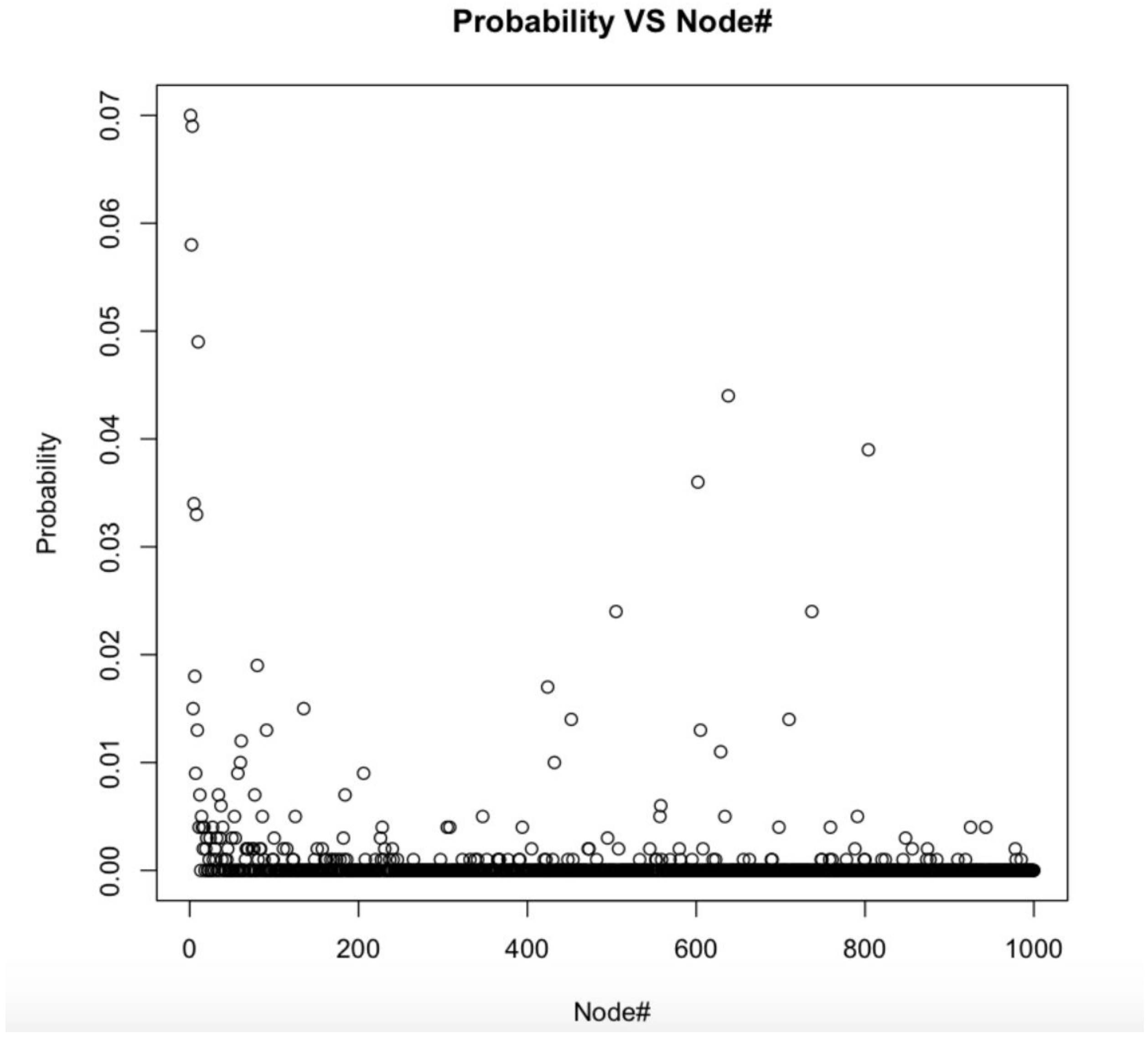

Figure 2.4.1 Probability Versus Node Index With Teleportation - Probability Proportional to PageRank

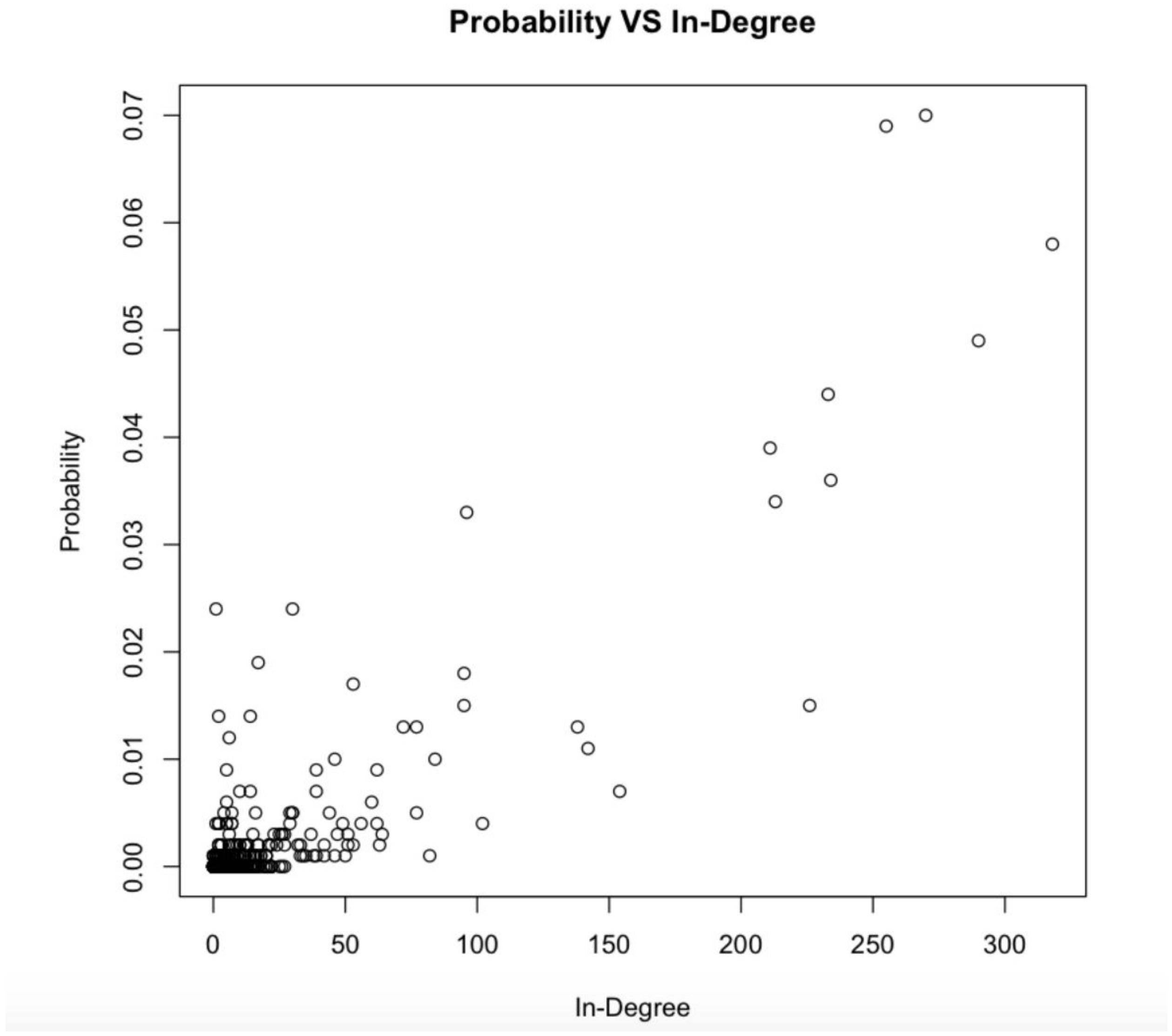

Figure 2.4.2 Probability Versus In-degree With Teleportation - Probability Proportional to PageRank

(b)

We sort the PageRank array we get in 3(b) and find the two nodes with median PageRank value. They are node 723 and node 724. We then keep other settings the same as before, but set the probability of teleportation to each node to be 0.5 and 0.5 for node 723 and node 724, while other nodes 0. The result is shown in Fig 2.4.3.

We can see that node 723 and node 724 have very high probability of being visited. It is more clear in probability versus in-degree graph, as is shown in Fig 2.4.4. We can see that two nodes in the upper left corner of the graph, with almost 0 in-degree, but have extremely high probability, 0.062 and 0.084 respectively. This is apparently due to the effect of teleportation.

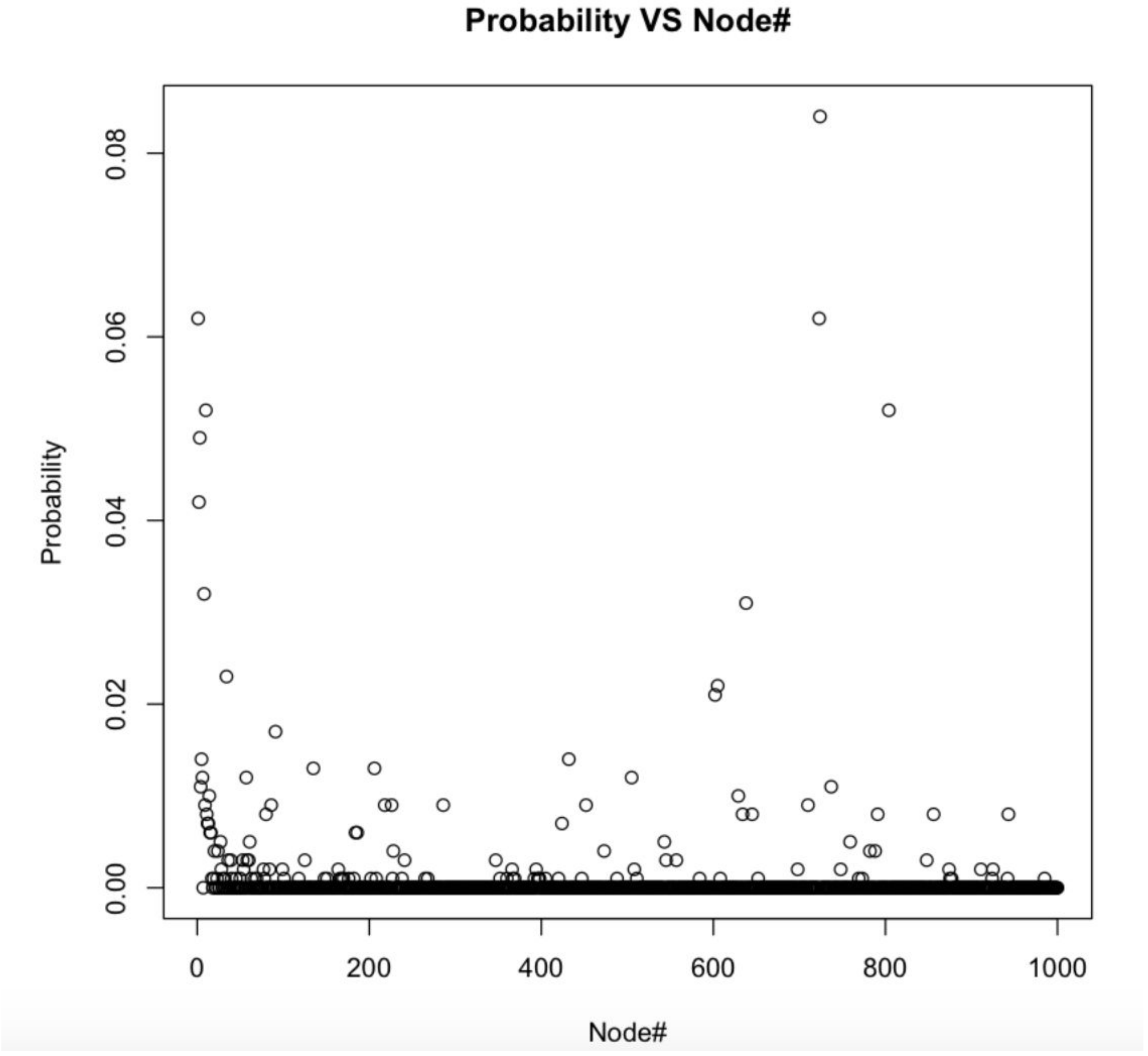

Figure 2.4.3 Probability Versus Node Index With Teleportation - Two Median Nodes

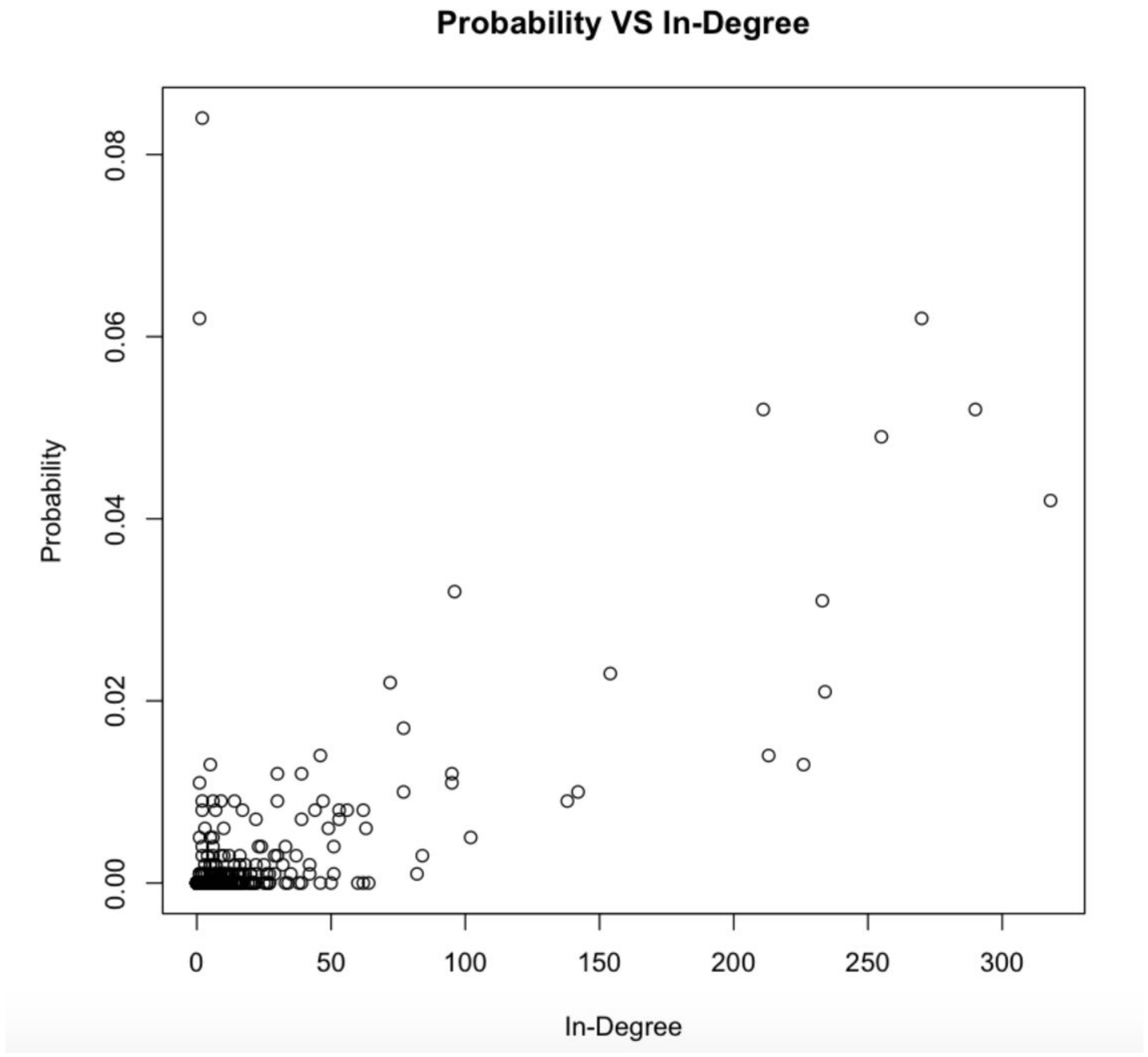

Figure 2.4.4 Probability Versus In-degree With Teleportation - Two Median Nodes

(c)

The original PageRank equation is

$$PR_i = \frac{1-d}{N} + d \sum_{j \in \{incoming\ nodes\ of\ i\}} \frac{PR_j}{L_j}$$

The modified equation should take into account the PageRank of each node, which is

$$PR_i' = (1-d)PR_i + d \sum_{j \in \{incoming\ nodes\ of\ i\}} \frac{PR_j}{L_j}$$