EE 232E Project 1

Random Graphs and Random Walks

Wei DU
UID: 005024944
Email: ericdw@g.ucla.edu

Xiao Yang
UID: 104946787
Email: avadayang@icloud.com

Fangyao Liu
UID:204945018
Email:fangyaoliu@g.ucla.edu

Ruchen Zhen
UID:205036408
Email:rzhen@ucla.edu

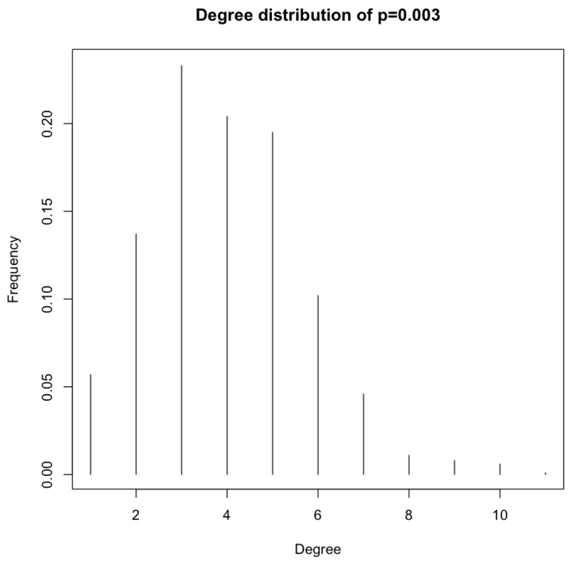
Part1 Generating Random Networks

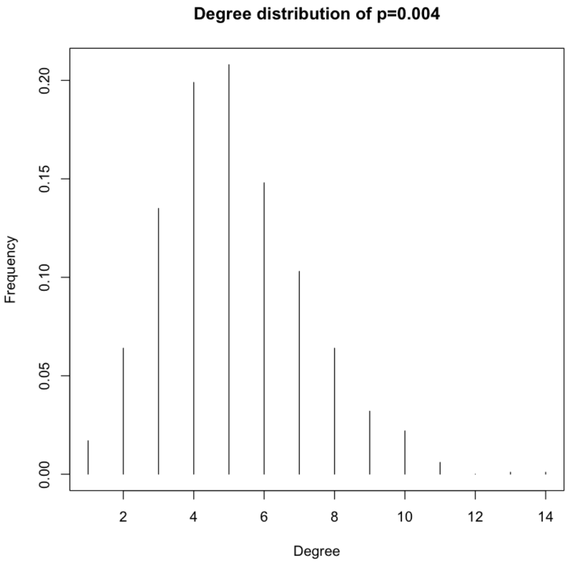
1. Create random networks using Erdös-Rényi model

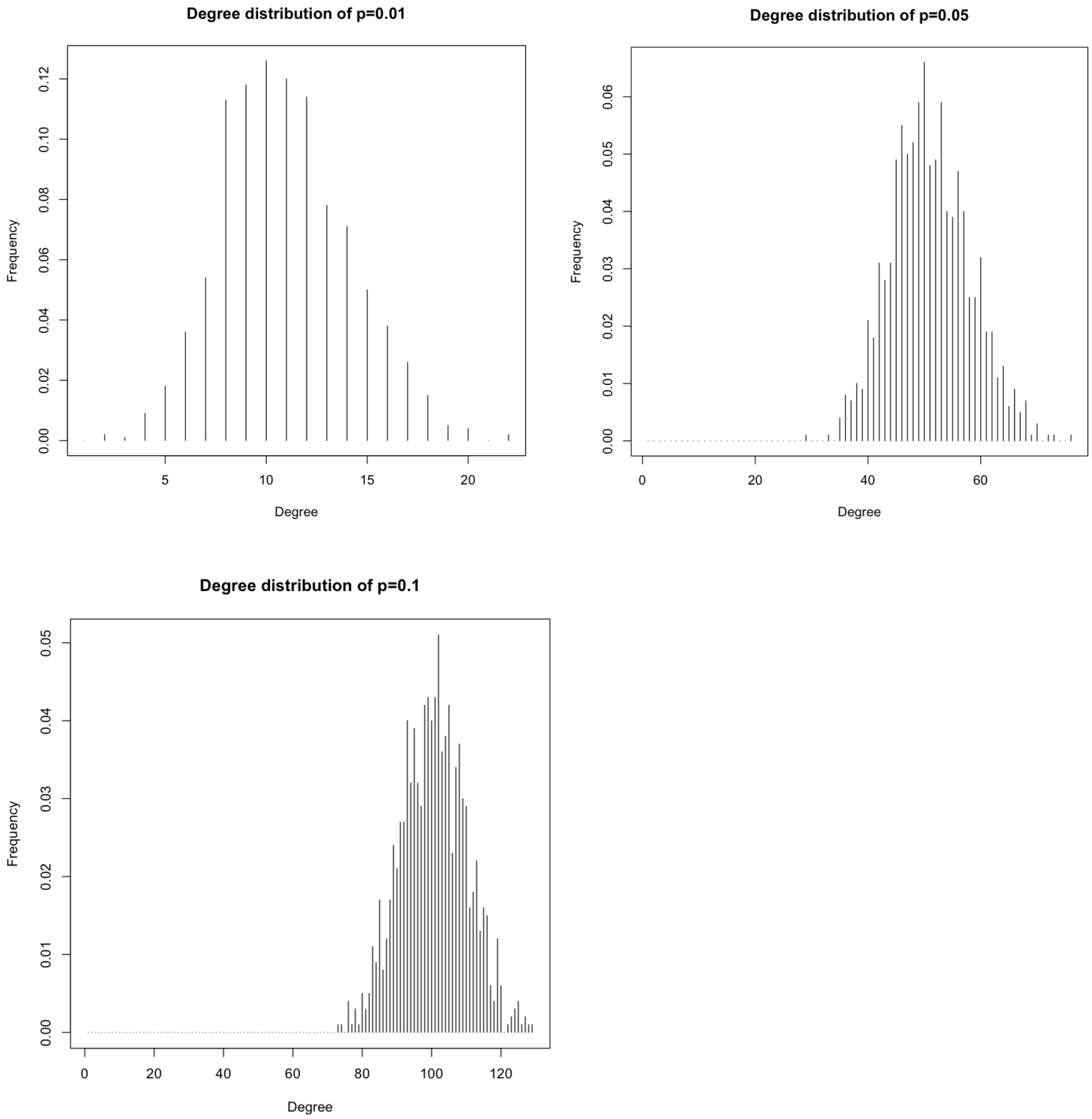
In this part of the project, we are asked to first create undirected random networks under different situations like same node and different probability. Then we will discover the characteristics of the Erdös-Rényi networks like GCC size etc.

1.1.a Create undirected random networks under different probabilities

As required, we will create an undirected random networks with 1000 nodes and plot its degree distributions, under probabilities of 0.003, 0.004, 0.01, 0.05 and 0.1. Graphs are presented below:







Degree distribution of p=0.01

Degree distribution of p=0.05

Degree distribution of p=0.1

Figure 1 Undirected random networks with 1000 nodes in different prob

It can be observed that all of them conforms to binomial distributions. The reason is that for each vertex, the probability of d nodes connecting to it (which means this vertex has d degrees) is

$$\binom{n}{d} * p^d * (1-p)^{n-d}$$

This is obviously the definition of binomial distribution.

Also empirical means, empirical variances, theoretical means and theoretical variances are presented in the following table.

	Expected mean	Real mean	Expected var	Real var
P=0.003	3	2.834	2.991	2.767
P=0.004	4	4.17	3.984	3.871
P=0.01	10	10.234	9.9	10.155
P=0.05	50	49.446	47.5	45.497
P=0.1	100	100.4	90	96.238

From the table, we could find out that expected(theoretical) values are pretty close to the real(empirical) values.

1.1.b Connectivity of random networks

When we are considering about the problem that whether all these random realizations of the ER network above connected, it might be good to test it then reach a conclusion.

First test with the original graph, the result is presented below:

	P=0.003	P=0.004	P=0.01	P=0.05	P=0.1
connected	no	no	yes	yes	yes

Run 10000 loops and count the times that random graph with certain probability is connected. Result is presented below:

	P=0.003	P=0.004	P=0.01	P=0.05	P=0.1
Probability of connected	0	0	95.99%	100%	100%

As required, find the diameter of giant connect component if not connected. So only the GCC diameter of p=0.003, p=0.004 and p=0.01 will be presented, since under the other two probabilities, there are not only disconnected network appear.

	P=0.003	P=0.004	P=0.01
Diameter of GCC	14.22	10.85	5.39

1.1.c Normalized GCC size v.s. probability

In this part, we try to check the nonlinear relationship between normalized GCC size and probabilities. Given n equals to 1000, sweep over values where are around p = O(ln (n)/n). Here, we select a region of (0.0001, 0.02) with step size 0.0001. Then we get a scatter plot of normalized GCC sizes vs probability.

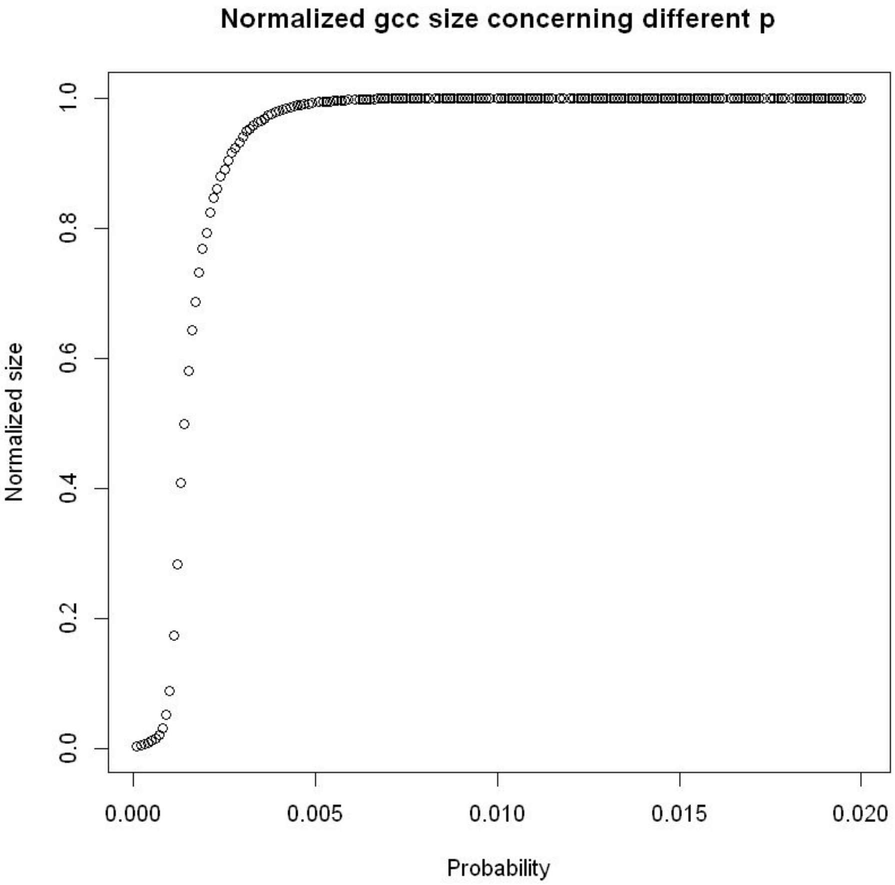


Figure 2 Normalized GCC size and probabilities

Our group gives the definition of “GCC emergence” as follow: under certain p, whenever how many times we randomly generate this graph, a GCC with fixed size(which means connecting all the nodes) will definitely appear every time. Another interpretation of this definition is that normalized GCC size is stable and equals to 1. Theoretically, p=ln(n)/n is a sharp threshold for connectedness of G(n, p). From the graph, we can observe that when p is around 0.007, normalized GCC size becomes stable and starts to equal to one. In theoretical way, GCC will emerge when p=ln(n)/n= 0.0069. So results match in both theoretical and empirical ways.

1.1.d Characteristics of random networks under different np relationships

In this part we discover GCC size under different np relationships using empirical methods. For each n, we generate 50 random networks and take the average of 20 GCCs' sizes and get the graphs below:

I. c = 0.5

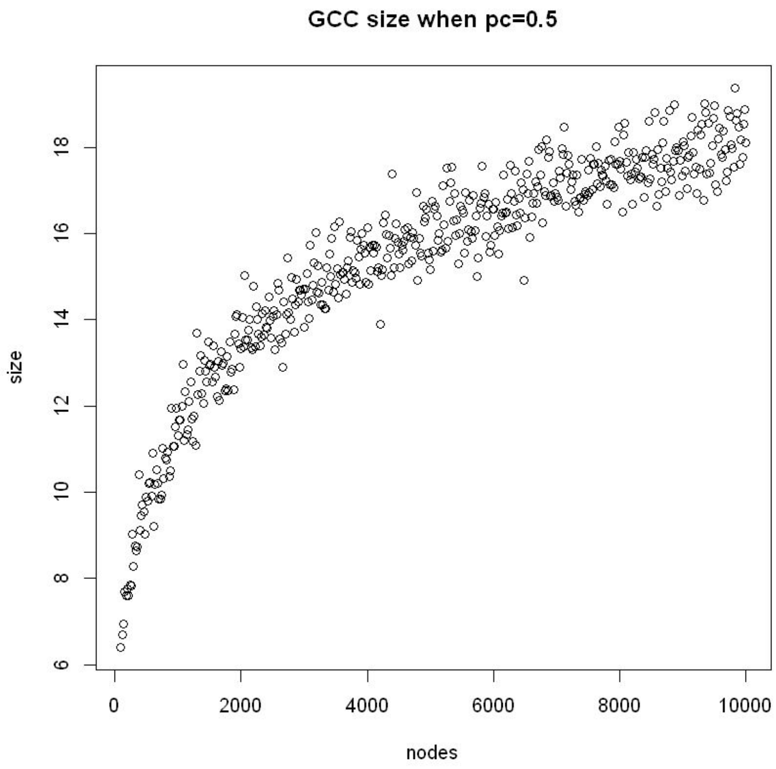


Figure 3 GCC size when pc = 0.5

Here we can observe a log trend between nodes and sizes of GCC when np<1

II. c=1

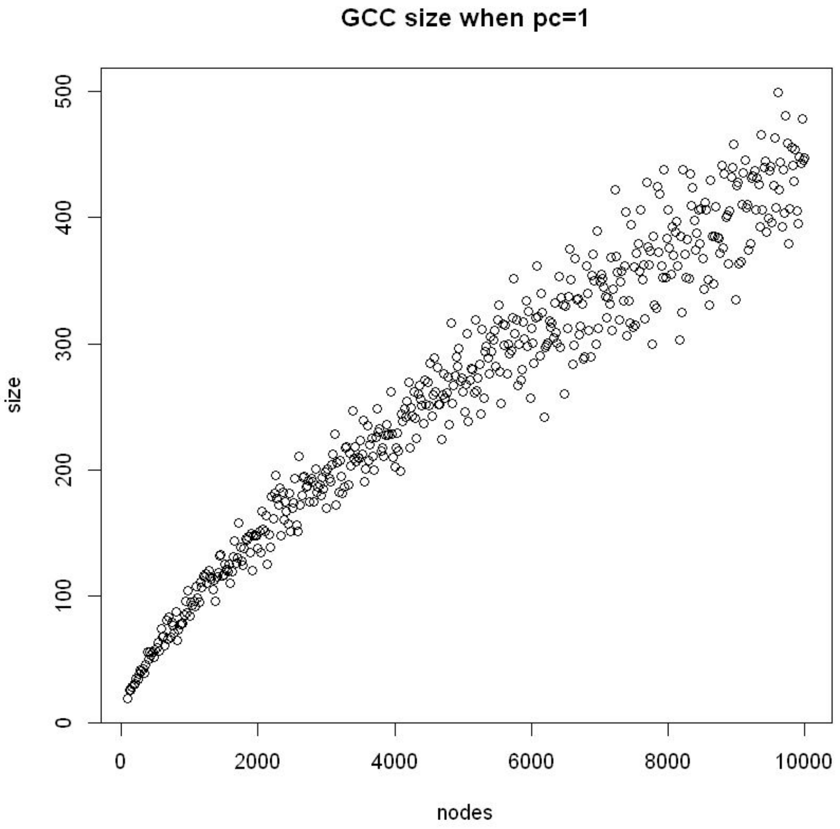


Figure 4 GCC size when pc = 1

Here we can observe a power trend between nodes and sizes of GCC when np=1

III. c=1.1, 1.2, 1.3

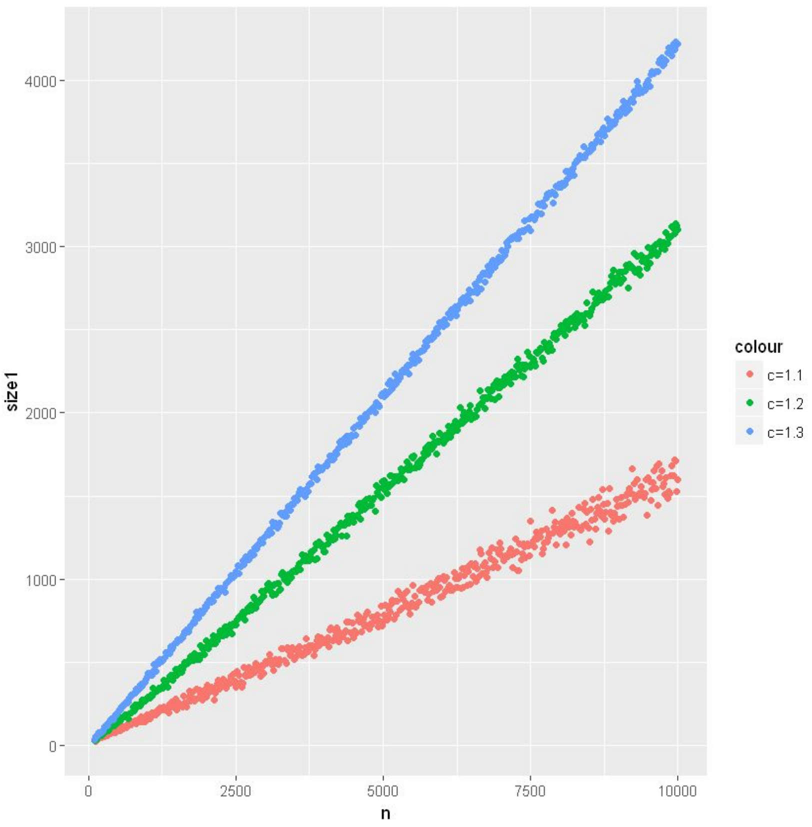


Figure 5 Nodes and sizes of GCC when np>1

Here we can observe a linear trend between nodes and sizes of GCC when np>1

2. Create networks using preferential attachment model

2.a Connectedness of network

We build a preferential attachment model with n=1000 nodes and m=1 for 100 times. All the generated models are connected. So we can conclude that such a network is always connected.

2.b Modularity measurement

This part we use fast greedy algorithm to find the community structure and measure the modularity. The modularity of the network is: 0.9337516.

Community sizes:

community	1	2	3	4	5	6	7	8
Number of vertices	49	45	44	49	42	39	49	38
community	9	10	11	12	13	14	15	16
Number of vertices	40	41	40	37	38	33	35	33
community	17	18	19	20	21	22	23	24
Number of vertices	29	30	31	27	28	23	22	21
community	25	26	27	28	29	30	31	32
Number of vertices	18	19	21	18	18	16	14	13

Graph representation:

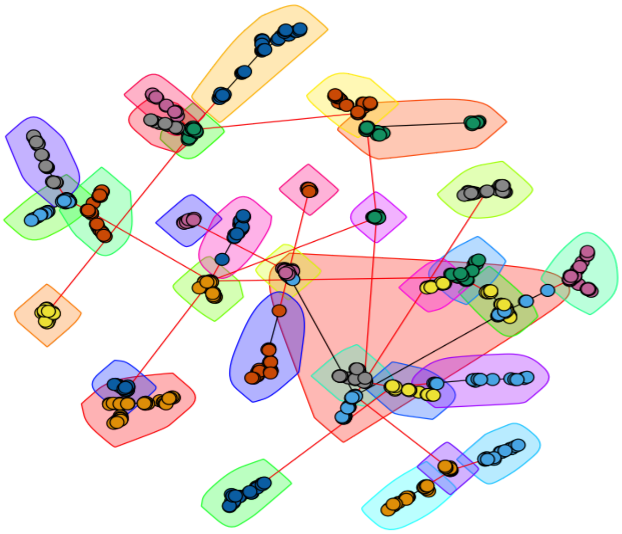


Figure 6 Barabasi model with n=1000 nodes and m=1

2.c Larger network properties

According to the requirement, we repeat the same procedures for a larger network. The modularity of the larger Barabasi network (n=10000) is: 0.9780523. So we can conclude that the modularity of a larger Barabasi network is larger than the smaller one.

2.d Slope of the plot

First we plot the degree distribution in a log-log scale for n=1000 and n=10000. Then we will estimate the slope of the plot.
n=1000

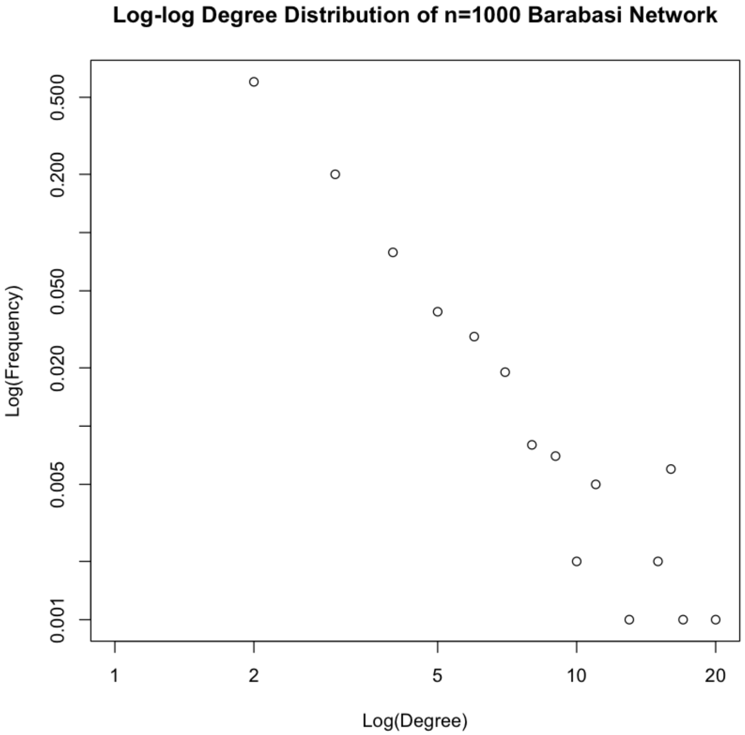


Figure 7 Degree distribution in a log-log scale with n = 1000

Slope is approximately 2.26

n=10000

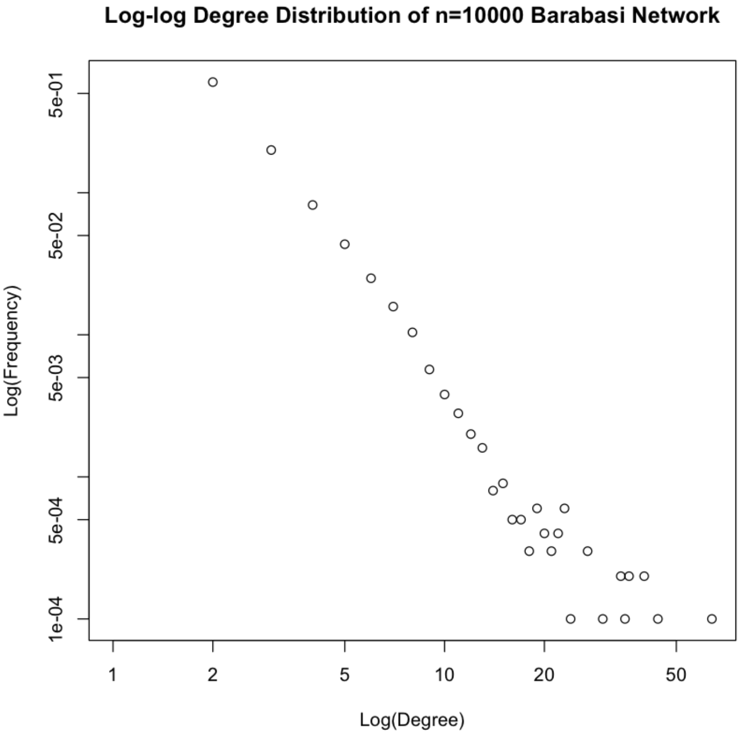


Figure 8 Degree distribution in a log-log scale with n = 10000

Slope is approximately 2.8

2.e Degree Distribution of Random Neighbours of Random Picked Nodes

First we plot the degree distribution of this random pick process.

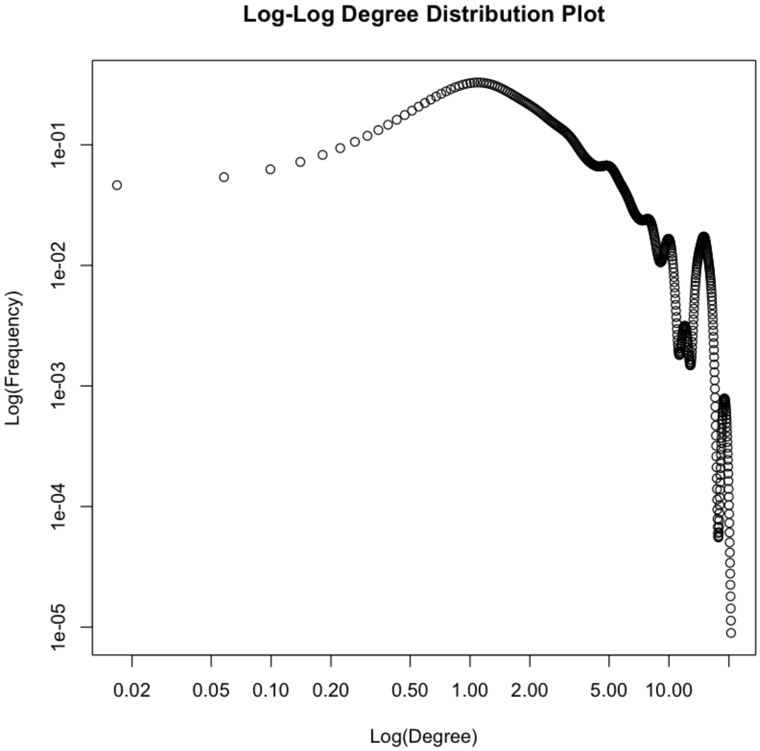


Figure 9 Degree distribution of this random pick

Different from the the node degree distribution, the linearity of the log-log degree distribution plot only remains in a certain degree domain. At other regions, the degree distribution changes with no significant rule.

2.f Expected Degree vs. Age

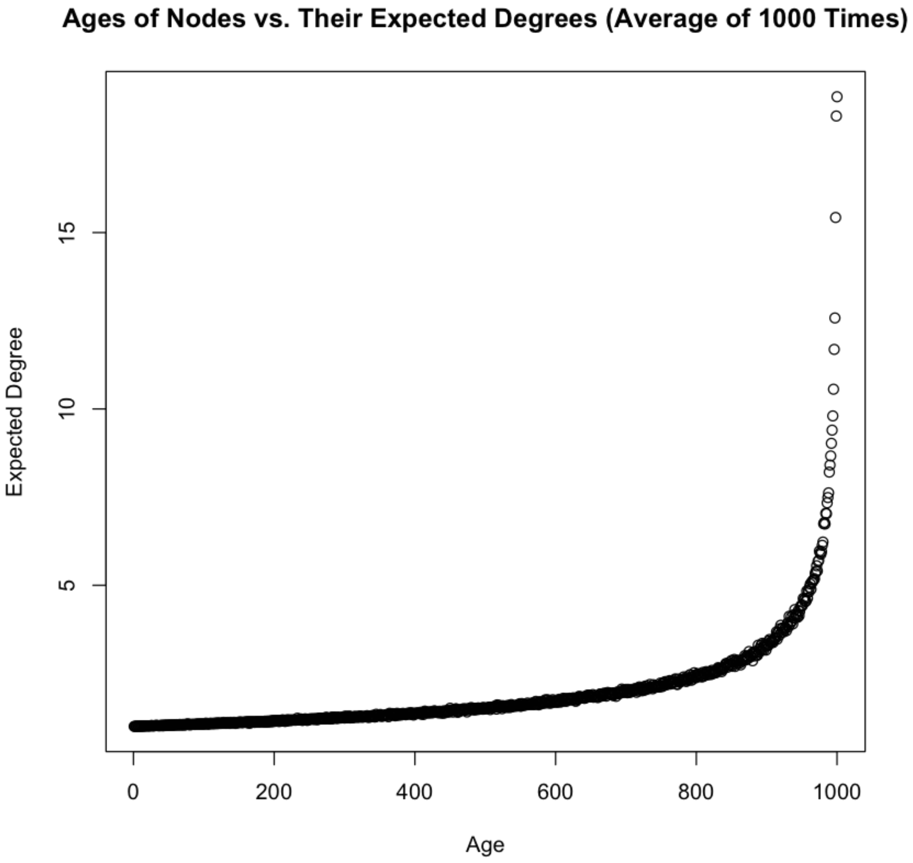


Figure 10 Expected degrees and the age of nodes

As we can see from the figure, the expected degree increases exponentially with the age of a specific node.

2.g Relationship between Modularity and m

The plot of Barabasi network with m=2 is shown in the following figure:

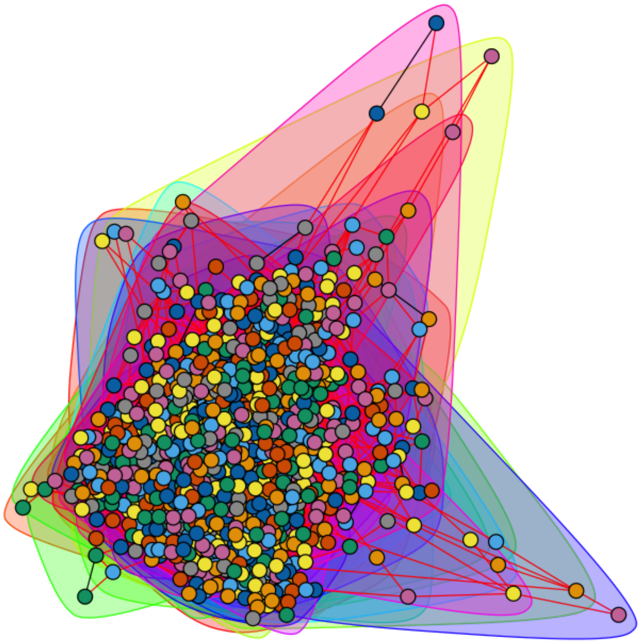


Figure 11: Barabasi network with m=2

The plot of Barabasi network with m=5 is shown in the following figure:

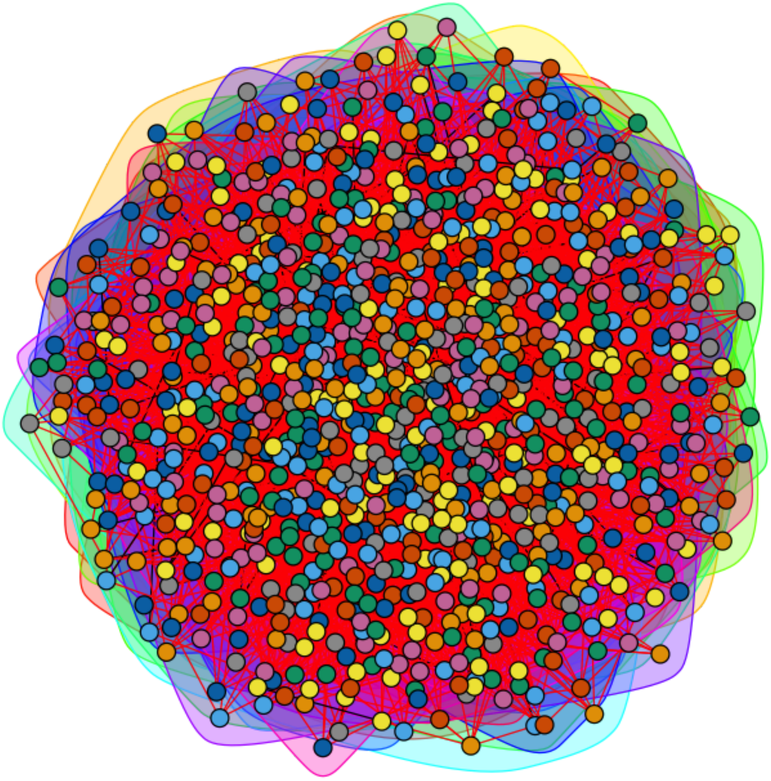


Figure 12: Barabasi network with m=5

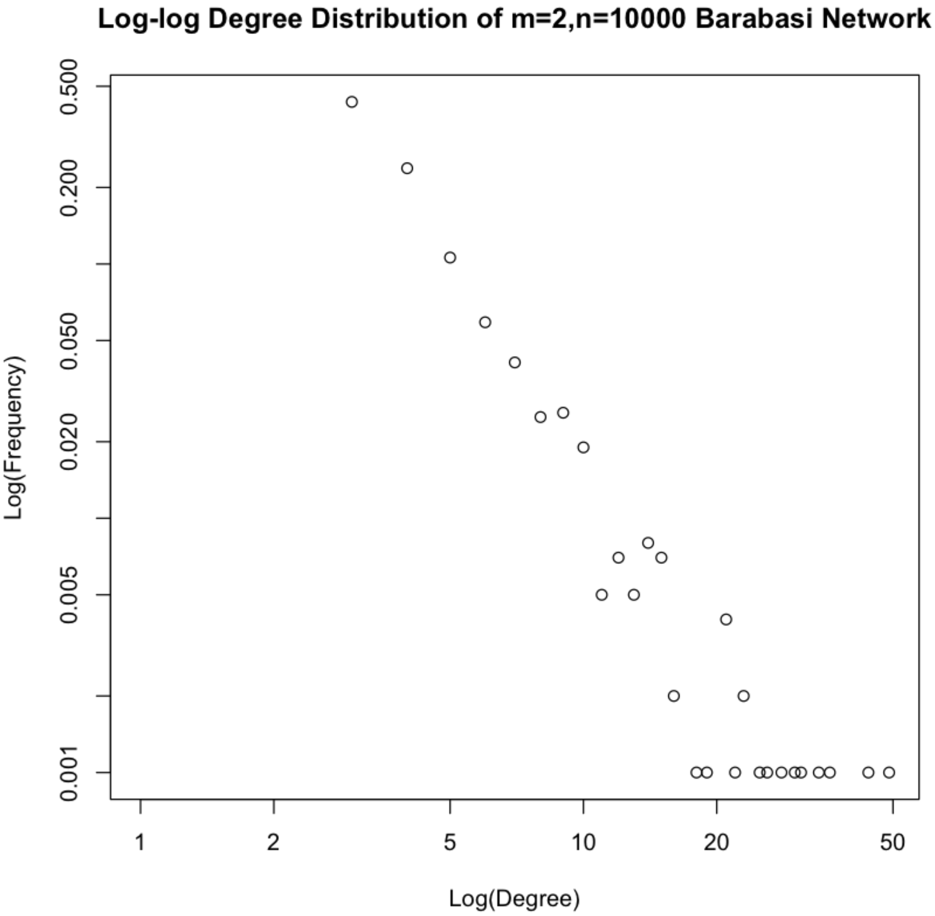


Figure 14 Degree Distribution of m =2 and n =10000

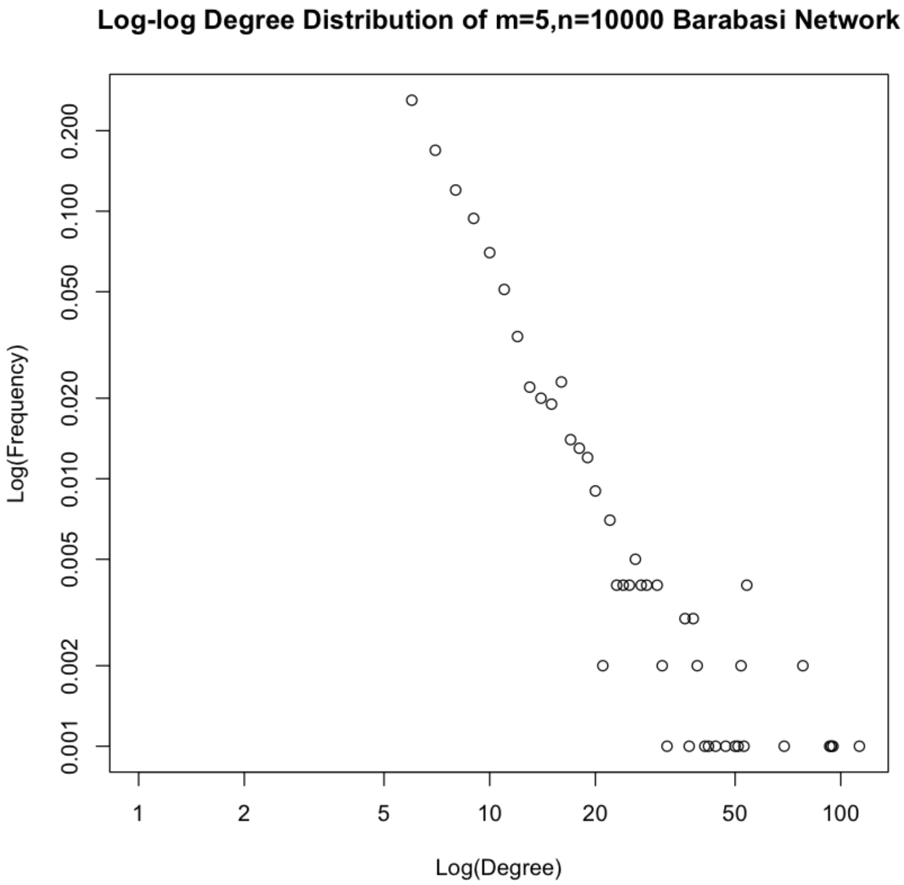


Figure 14 Degree Distribution of m =5 and n =10000

Modularity Table:

modularity	m=1	m=2	m=5
n=1000	0.9337516	0.5227439	0.2792011
n=10000	0.9780523	0.5267528	0.2766691

The modularity is defined as:

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j,$$

When m increase, a new added vertice will have to connect to more existed vertices. Therefore, it is harder to have a vertice which degree is significant higher than the others, which will be more attracted to new vertices. Thus, it is more difficult for the graph to modularize.

2.h Network Generation: Barabasi vs. Stub-matching

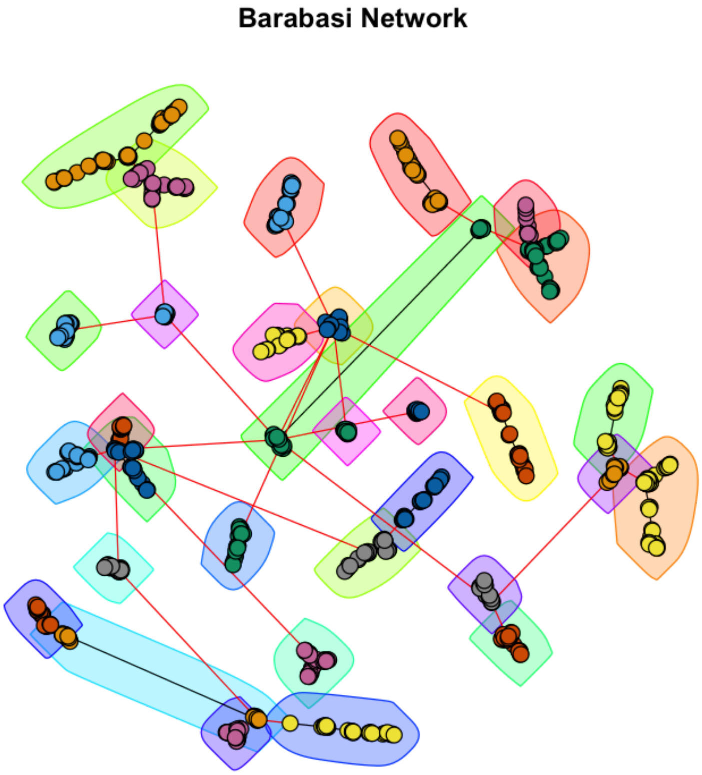


Figure 15 Network generated through barabasi

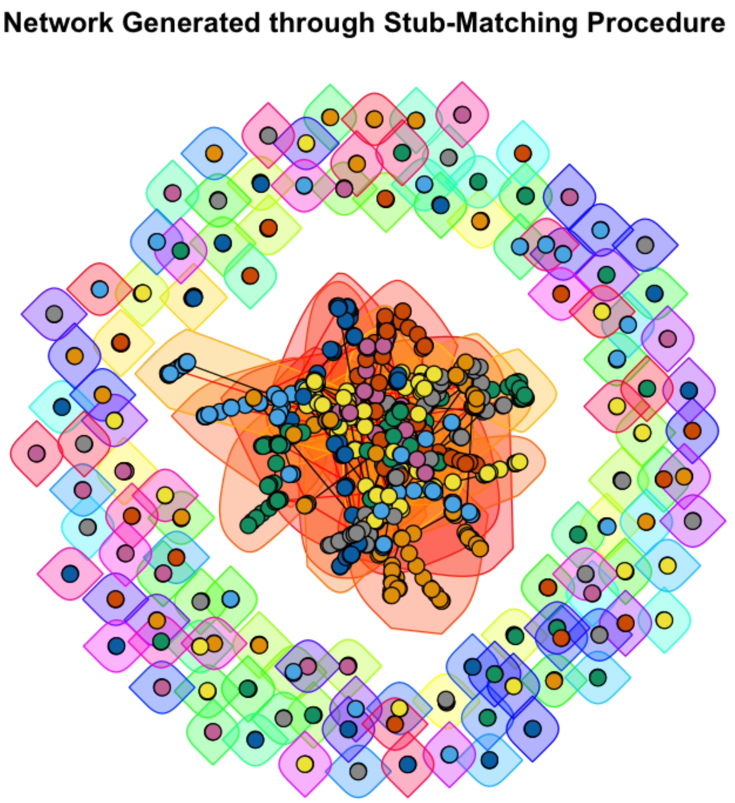


Figure 16 Network generated through stub-matching procedure

The modularity of the Barabasi network is: 0.9341884.
The modularity of the Network Generated through Stub-Matching Procedure is: 0.8422752.

Compare the two procedures for creating random power-law networks:
In the Barabasi network, we add one vertice and corresponding edges at a time. The vertices that the new vertice choose to connect is based on their degree. In the Stub-Matching Procedure, however, generating the vertices of the network along with their degrees first, then connecting the vertices of by edges.

For the network plot, the Barabasi network always connected, however, the Stub-Matching generated network has many isolated vertices.

3. Create modified preferential attachment model

3.a Degree Distribution

According to problem description, we plot the degree distribution of modified preferential attachment model.

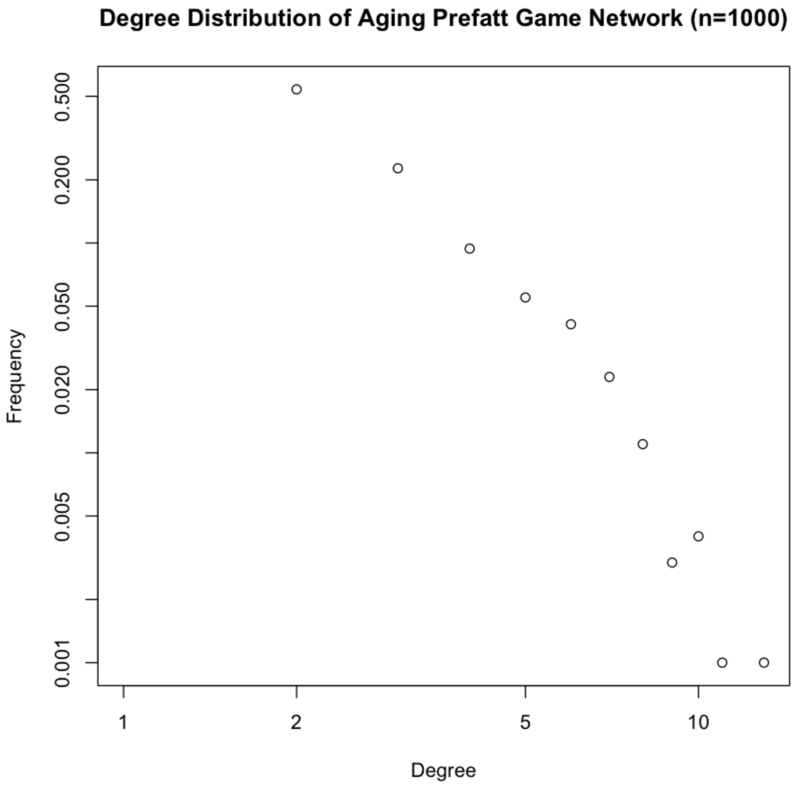


Figure 17 Degree distribution of aging prefatt game network with n = 1000

Power law exponent is 2.513

3.b Modularity

Using fast greedy method, we could find the modularity as 0.9359.

Community sizes:

community	1	2	3	4	5	6	7	8
Number of vertices	45	42	35	37	36	36	37	36
community	9	10	11	12	13	14	15	16
Number of vertices	37	31	33	32	32	33	31	31
community	17	18	19	20	21	22	23	24
Number of vertices	34	32	29	26	25	25	24	25
community	25	26	27	28	29	30	31	32
Number of vertices	25	22	23	22	23	24	23	22
community	33	34						
Number of vertices	18	14						

Graph Representation:

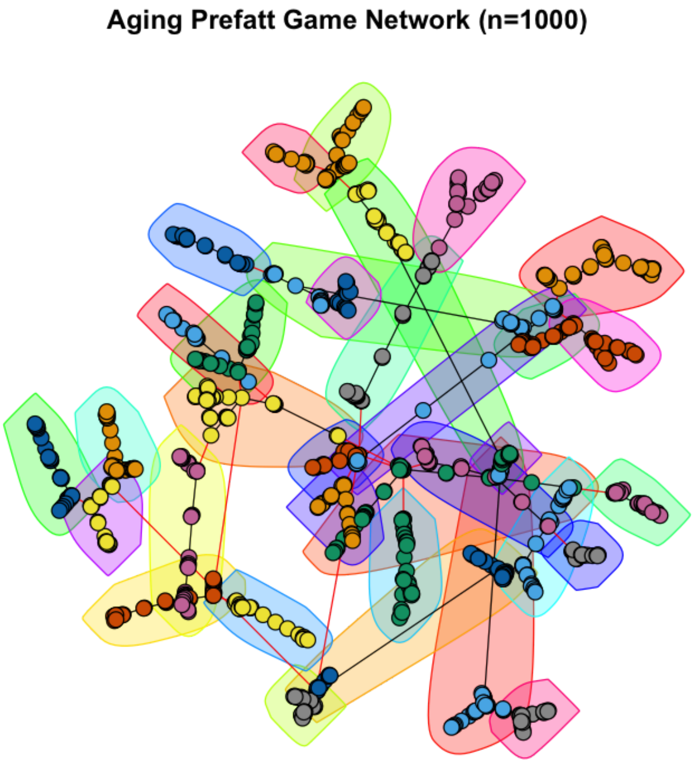


Figure 18 Aging Prefatt Game Network with nodes = 1000

Part2 Random Walk on Networks

1. Random walk on Erdös-Rényi networks

In this question, we will first create an undirected random network with 1000 nodes, and the probability p = 0.01 to draw an edge between any pair of nodes.(And we will also create 100 nodes and 10000 nodes graphs). Then let a random walker start from a randomly selected node. Measure the average shortest path length and the standard deviation of the walker from his starting point at certain step.

1.1 Network with 1000 nodes

1.1.1 mean and std

We generated a random network for 1000 nodes with probability of 0.01 for drawing an edge between any pair. The network is depicted as below in figure 1.

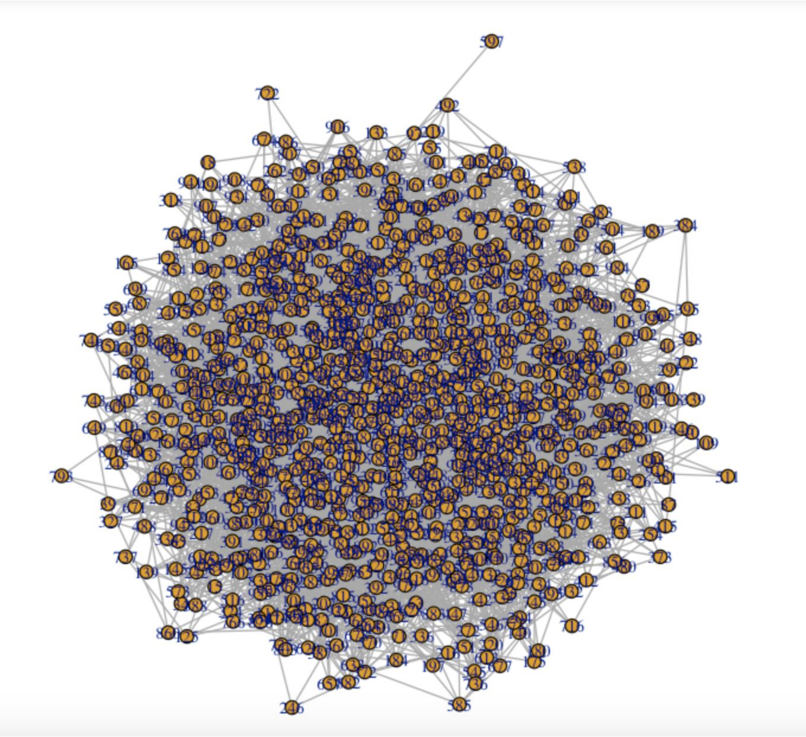


Figure 19 Erdos-Renyi model network with nodes = 1000

The network has the property of:

Diameter = 6
Connected = TRUE

The pictures below depicts the calculated mean and standard deviation of s(t):

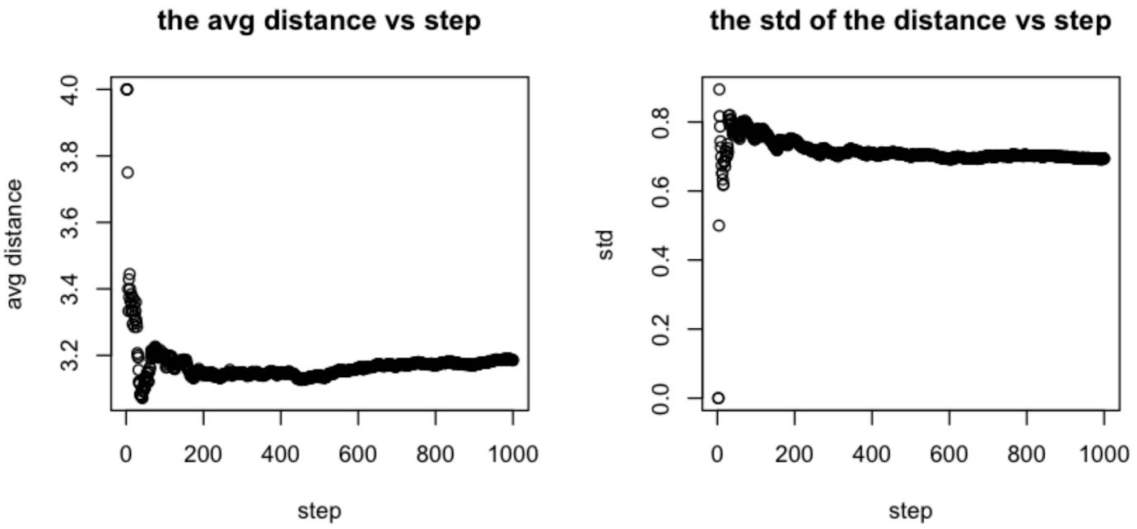


Figure 20 mean and std of the 1000 nodes network

The steady state of average path length has the equation of:

$$Average\ Path\ length\ (Erdos\ Renyi) = ln(N)/ln(Np)$$

From this equation we can calculate that ln1000 = 6.9 and ln10 = 2.3, which the avg path length is 3.
From figure 2, we can infer that the average path length reaches a steady state value of 3.2 approximately and the standard deviation is 0.71 approximately. And the result is really close to the value which is calculated by the equation.

1.1.2 the degree distribution

Then we draw the degree distribution of the nodes reached at the end of the random walk and the degree distribution of the graph.
The picture are shown below:

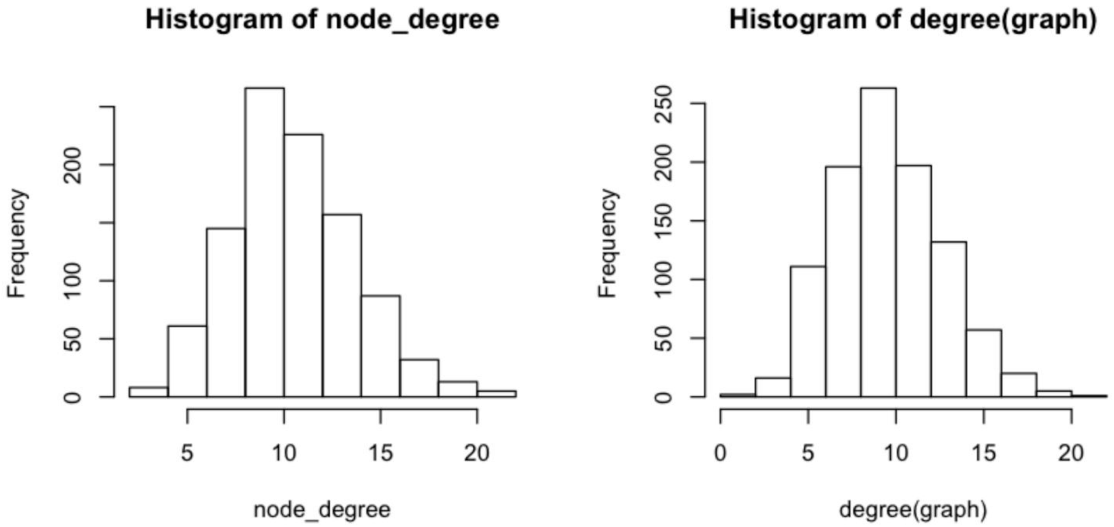


Figure 21 degree distribution of final nodes and whole graph

1.2 Network with 100 nodes

1.2.1 mean and std

We generated a random network for 100 nodes with probability of 0.01 for drawing an edge between any pair. The network is depicted as below in figure 4.

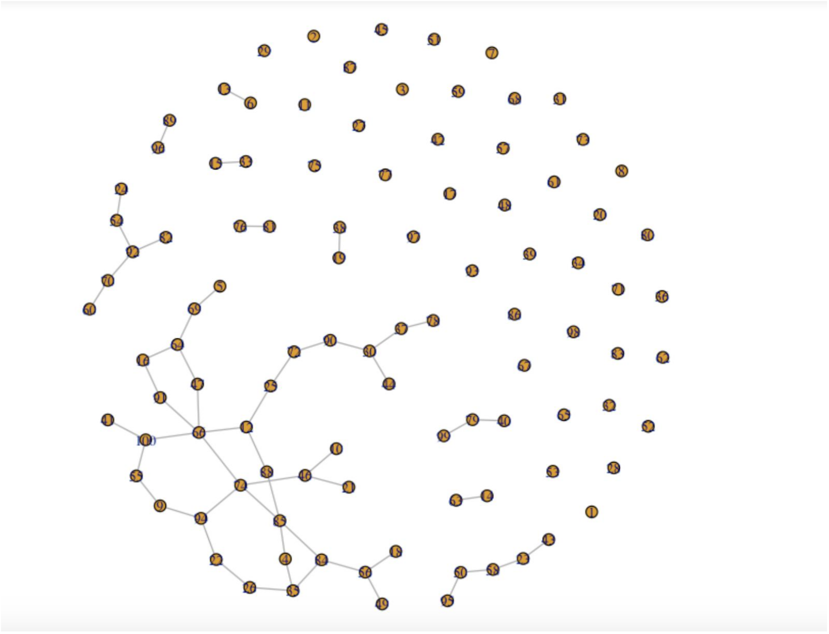


Figure 22 Erdos-Renyi model network with nodes = 100

We will calculate transition matrix, so the graph should be connected. In this case, we find the GCC of this unconnected graph which depicted below:

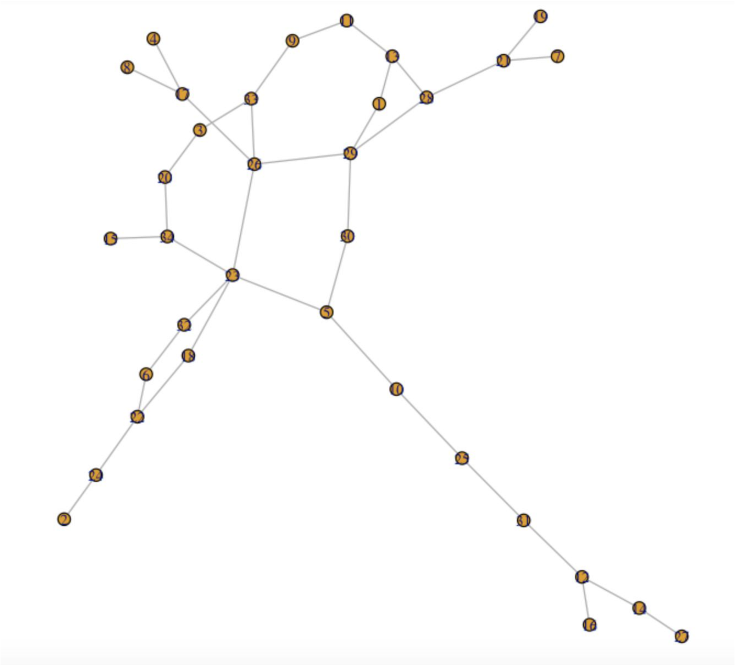


Figure 23 GCC of Erdos-Renyi model network with nodes = 100

The network has the property of:

Diameter = 11
Connected = FALSE

The pictures below depicts the calculated mean and standard deviation of s(t):

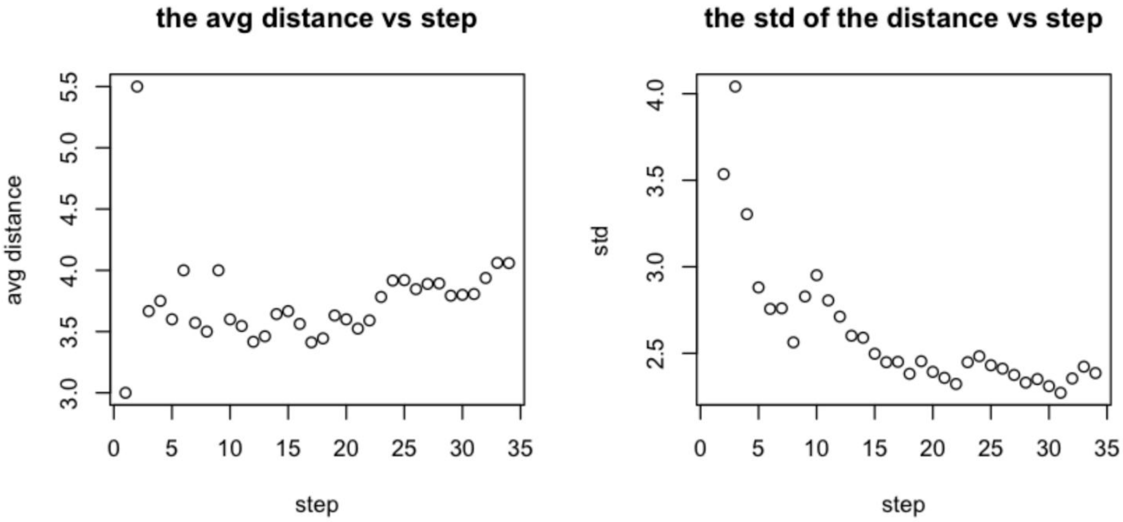


Figure 24 mean and std of the 100 nodes network

1.2.2 the degree distribution

Then we draw the degree distribution of the nodes reached at the end of the random walk and the degree distribution of the graph.
The picture are shown below:

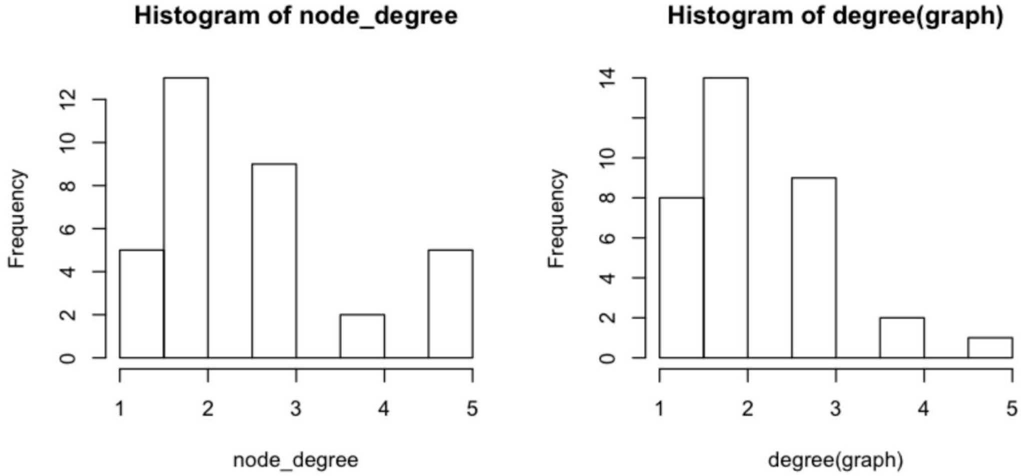


Figure 25 degree distribution of final nodes and whole graph

1.3 Network with 10000 Nodes

1.3.1 mean and std

We generated a random network for 10000 nodes with probability of 0.01 for drawing an edge between any pair. In this case, we modified the algorithm to perform only for 1000 iterations instead of 10,000. This can accelerate the speed of the program.
The network has the property of:

Diameter = 3
Connected = TRUE

The pictures below depicts the calculated mean and standard deviation of s(t):

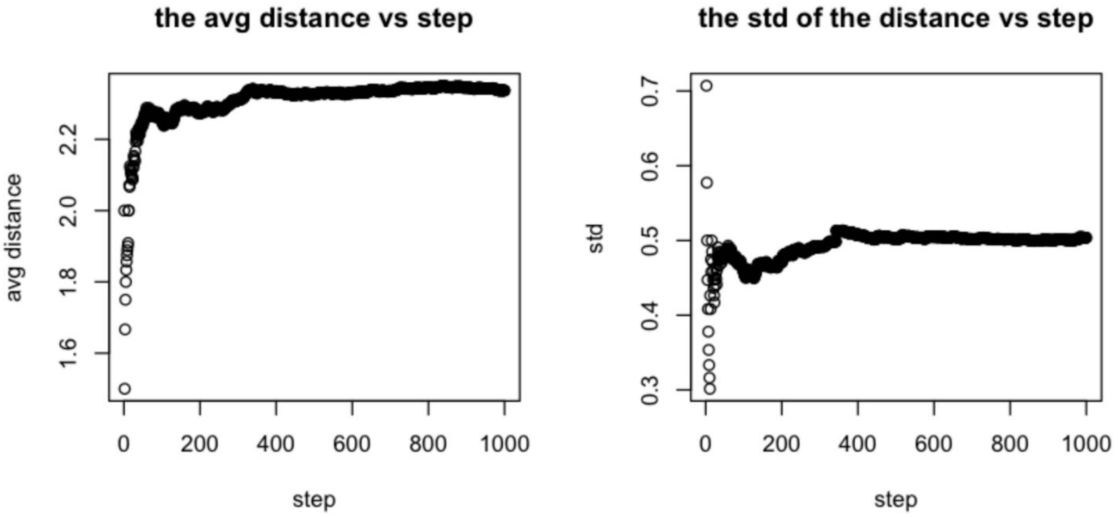


Figure 26 mean and std of the 10000 nodes network

From the equation listed above we can calculate that ln10000 = 9.21 and ln10 = 4.6, which the avg path length is 1.99.
From figure 8, we can infer that the average path length reaches a steady state value of 2.3 approximately and the standard deviation is 0.5 approximately. And the result is really close to the value which is calculated by the equation.

1.3.2 the degree distribution

Then we draw the degree distribution of the nodes reached at the end of the random walk and the degree distribution of the graph.
The picture are shown below:

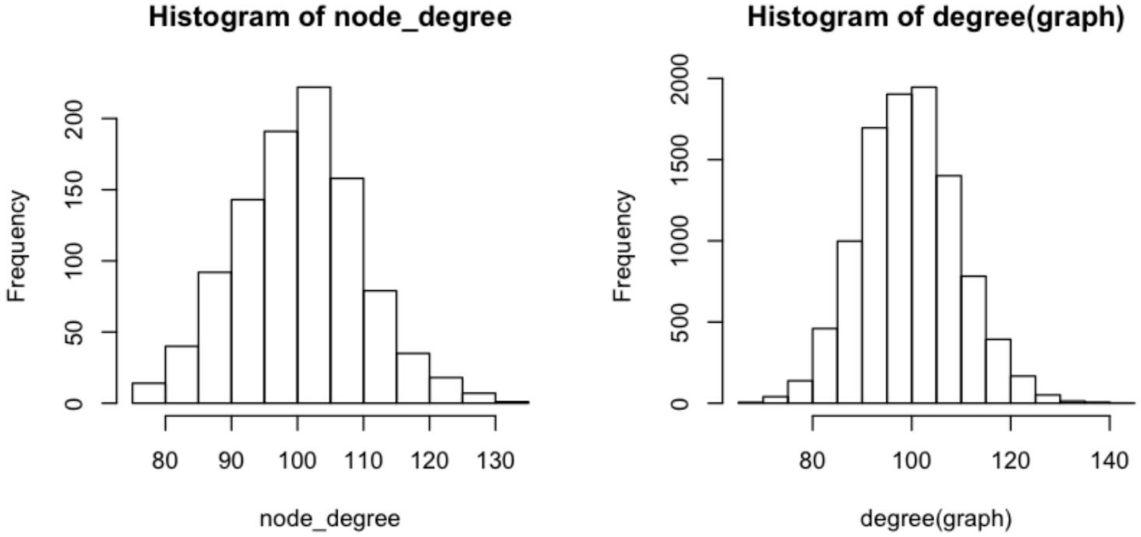


Figure 27 degree distribution of final nodes and whole graph

1.4 Conclusion

1.4.1 The distribution of the degree

From the above plots such as Figure 3, 7 ,9 , we can observe that the degree distribution of the nodes reached at the end of the random walk closely resembles the degree distribution of whole network. Because the random walk almost pass the whole network in this case in each iteration. This is the reason why the distribution of the final node is almost the same as the network.

1.4.2 The effect of Diameter

In our algorithm we used shortest path to find the distance between two nodes. We also inferred that the average distance of random walk on networks is proportional to the diameter of the network. The average path length for 1000 and 10000 nodes are 3.2 and 2.3 respectively. And the diameter is 6 and 3 respectively. In conclusion, if the nodes increases, the diameter decreases and the average path length decreases.

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

We utilized The Barabási-Albert Model to generate these undirected preferential attachment networks with N nodes, where each new node attaches to m=1 old nodes.

Hyperparameter: sample size = 200 ; steps = 300

Therefore, we here present the four measurements about each graph: the average distance (defined as the shortest path length) and relevant standard deviation of the walker from his starting point at step t, the degree distribution of the nodes reached at the end of the random walk, and finally the degree distribution of the graph.

2.1 Network with 1000 Nodes

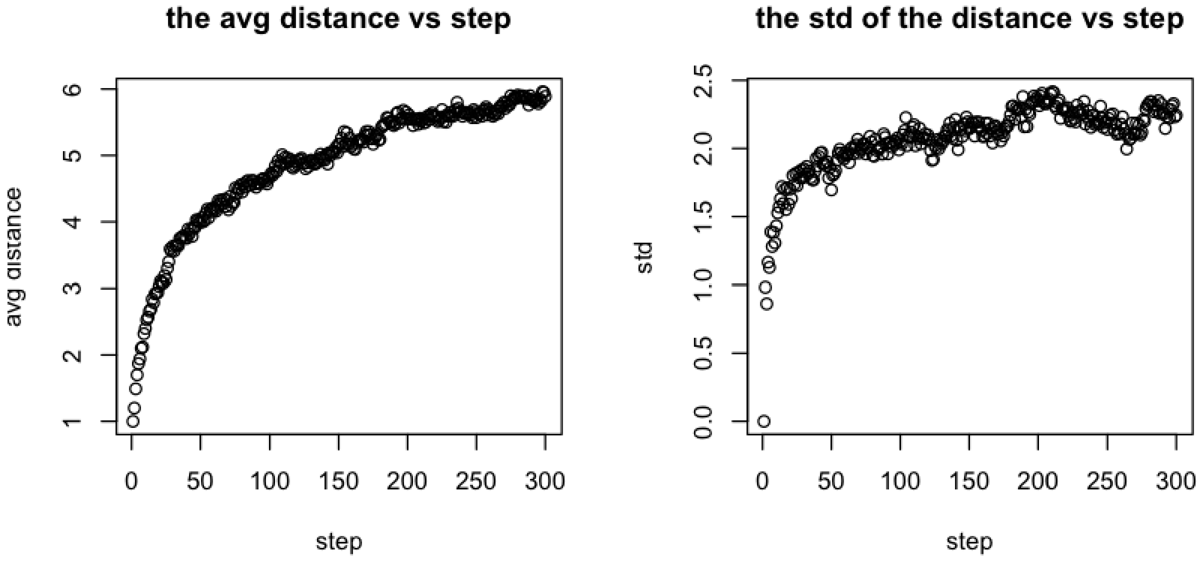


Figure 28 avg distance vs step and std of distance vs step for graph with 1000 nodes

From graph we can tell the mean distance grows with number of steps taken, and doesn't seem to converge at a fixed value; meanwhile, the standard deviation of distance gradually converges to 2.3

According to result, we can also have observation that the degree distribution of the nodes reached at the end of the random walk aligns with the degree distribution of the graph, which indicates that, with sufficient steps random walk can be a good simulation of preferential attachment.

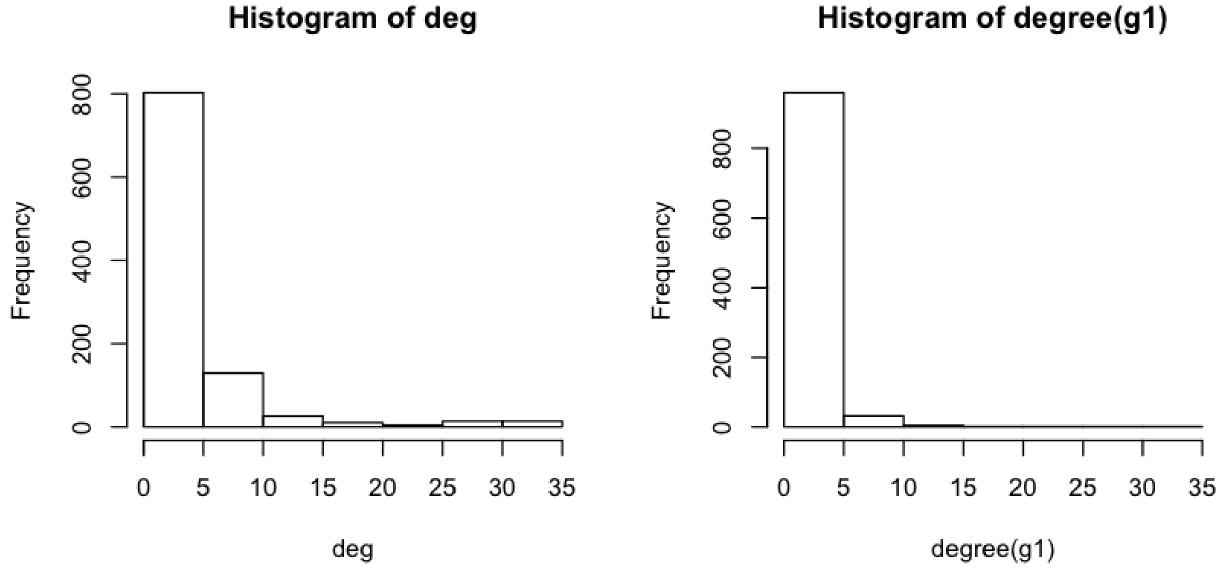


Figure 29 degree distribution of the nodes reached at the end of the random walk, and the degree distribution for graph with 1000 nodes

2.2 Network with 100 Nodes and 1000 Nodes

Here we give results of difference size of graphs generated by Barabási-Albert Model.

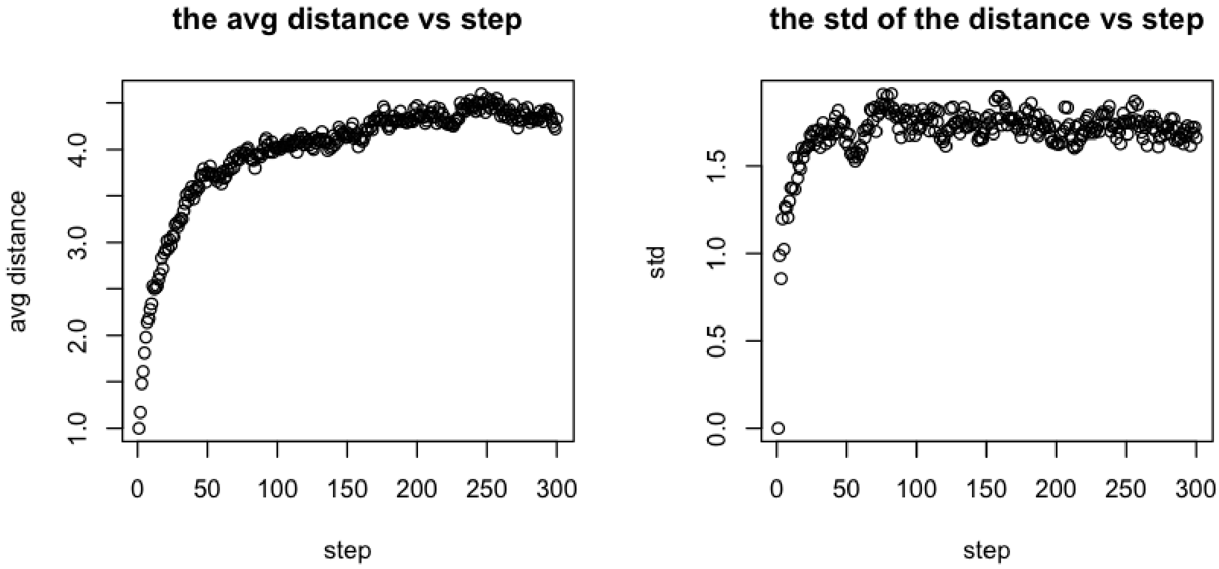


Figure 30 avg distance vs step and std of distance vs step for graph with 100 nodes

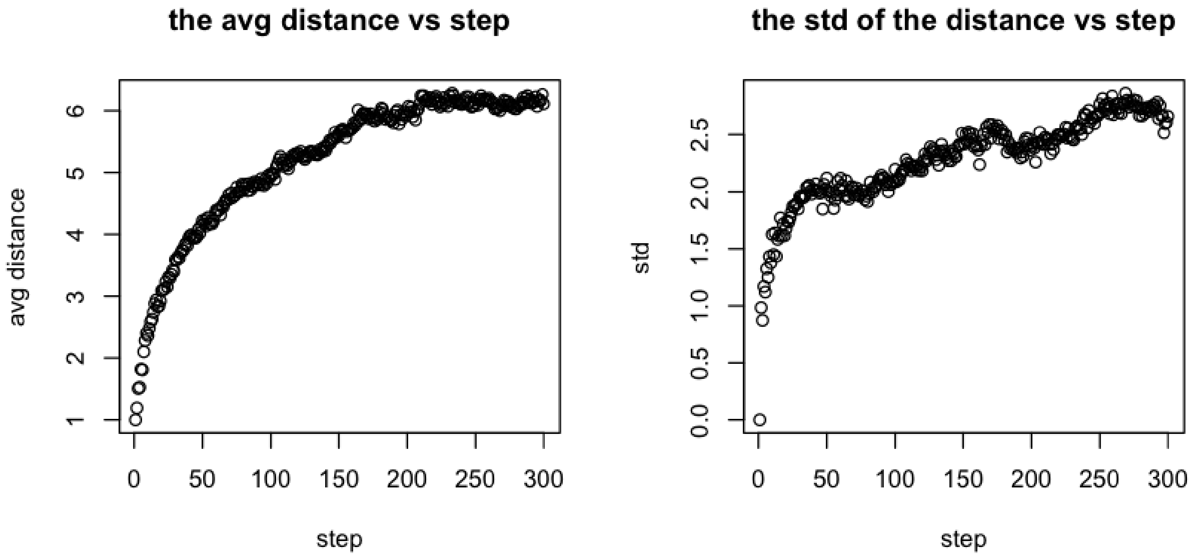


Figure 31 avg distance vs step and std of distance vs step for graph with 10000 nodes

Graph size	connected	diameter
100	True	4.9
1000	True	7.9
10000	True	11.9

Table 1 connectivity and diameter of graphs

From the results we can conclude that the diameter does play a role in affecting the average distance and std of distance of graphs generated by Barabási-Albert Model with all other being equal. The positive correlation between diameter and the average distance and std of distance of the walker from his starting point at step t is seen. One plausible explanation can be: the final location of random walker has a great chance to be the nodes with max degree, while larger diameter adds outliers that are far from nodes mentioned.

3. PageRank

The PageRank algorithm can be used to influence the ranking of search results which exploits the “importance” scores of pages. Here, we use random walk to simulate PageRank.

3.1 Pagerank without teleport

In this problem, we first create a directed random network with 1000 nodes, using the preferential attachment model, where m = 4, in which the in-degrees follow a power law distribution. We then measured the probability of the walker visiting each node. We also computed the degree of the nodes. In order to get a sense of the relationship between the visit probability and the node degree, we plotted them. The plot is given below:

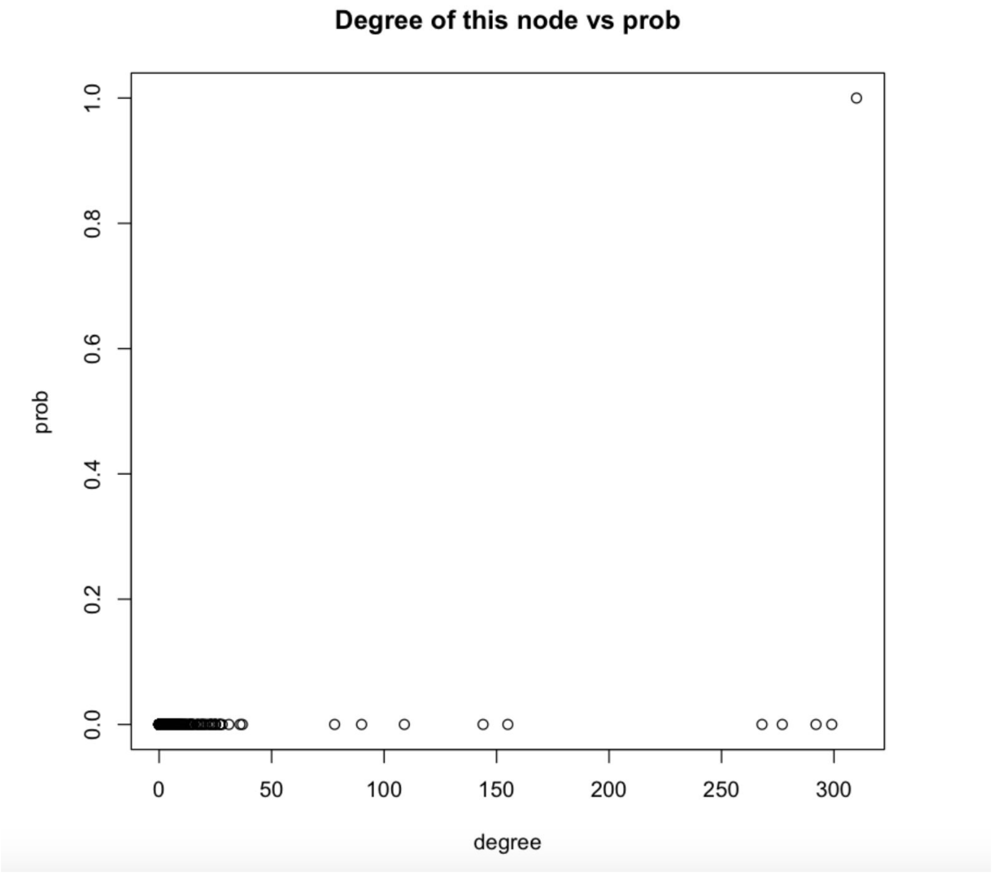


Figure 32 Visit probability and Node degree without teleport

From the above plot it can be seen that the visit probability is 100% of the biggest degree, which is the node number 1. Because we create a directed random network with 1000 nodes, using the preferential attachment model. And the construction process of this graph is like this:

1. First generate node 1 without in-degree and out-degree
2. Generate directed node 2 connect to node 1

3. Generate directed node 3 connect to node 1 and node 2
4. After generate all the nodes with 4 out-degree, the path will finally lead to node 1

This is the reason why when we take random walk in this directed graph, we will finally walk to the node 1 and hence the node 1 has the 100% probability with highest in-degree.

3.2 Pagerank with teleport

In all previous questions, we didn't have any teleportation. Now, we use a teleportation probability of 0.15. And use the same graph in the previous question. We then measured the probability of the walker visiting each node. The plot is given below:

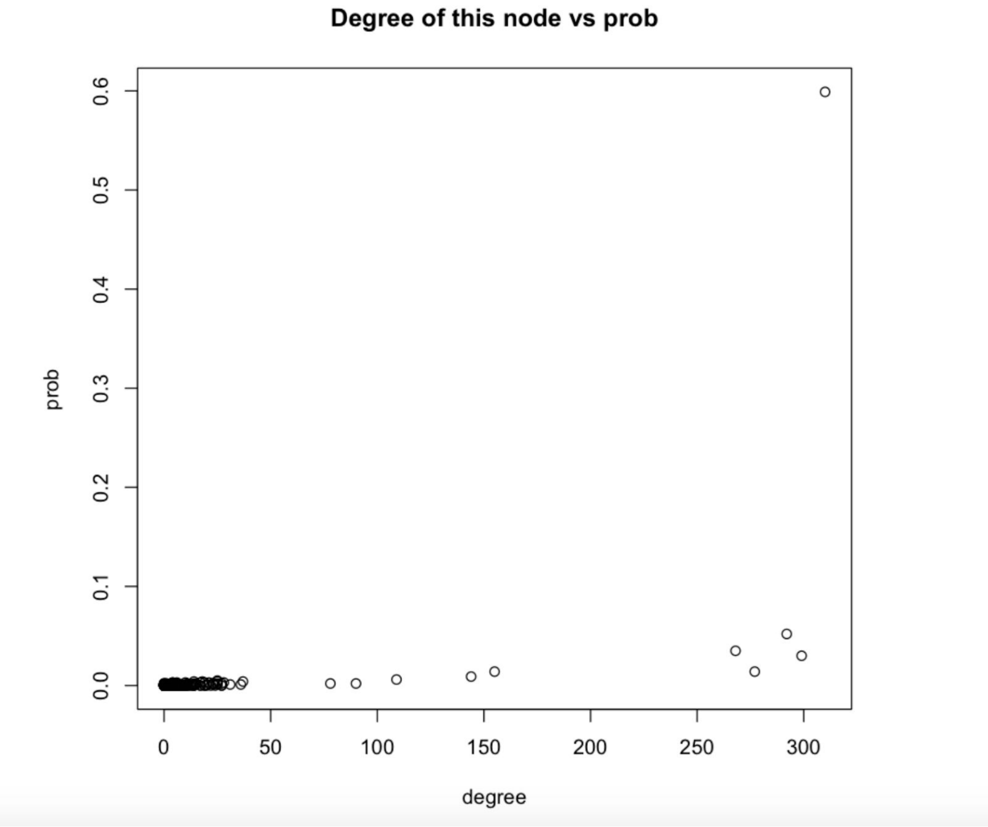


Figure 33 Visit probability and Node degree with teleport

From the above plot it can be seen that the visit probability is 59.9% of the biggest degree, which is the node number 1. And 5.21% of the second biggest degree, which is the node 2. And 3.1%, 3% for the following node 3 and 4.

We can know from the result that with teleportation, even though we also have the highest probability to reach the node 1(the highest in-degree), we also have a little increase in the probability of the second highest in-degree node and third one. This is because in every step, we have the probability to teleport to any random nodes, if this happens in the last few steps, we may not reach the node 1. However, highest in-degree still has more chance to reach.

4. Personalized PageRank

4.1 General Personalized PageRank

In last problem, each node has 1/N change to be destination of teleportation, while in this problem, we are about to change that by replacing 1/N with the node’s PageRank.

Original PageRank Formula：

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

Which is equal to this form: R = (1-d) * 1/N + d A R
However, when we are done replacing, it becomes R = (1-d) * R + d A R

When we compare the two graphs, we can tell that the nodes with higher degree now has better chance to be visited by random walker. Since the PageRank, compared with uniform distribution of 1/N favors nodes with high degree more.

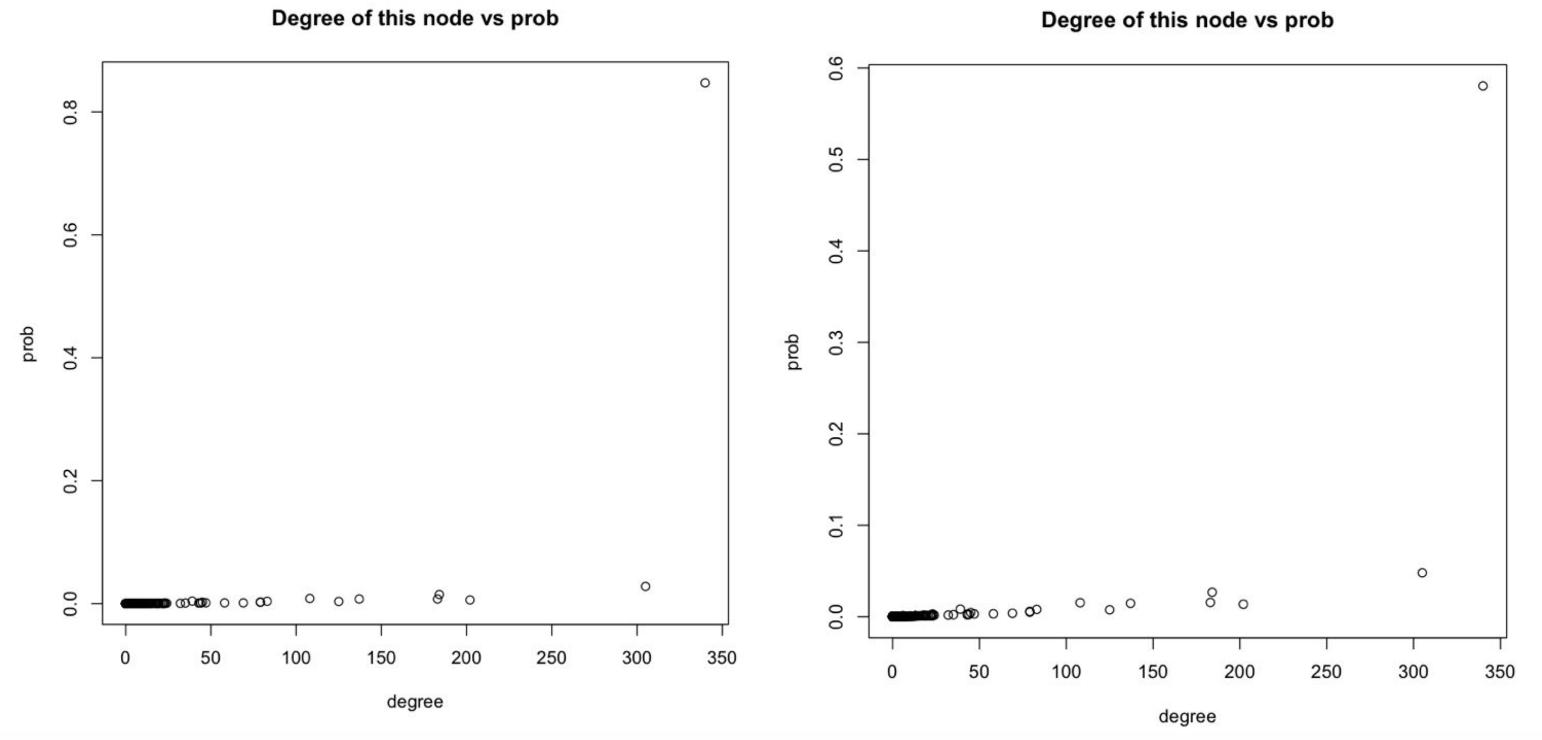


Figure 34 after (left) and before (right) replacing

4.2 General Personalized PageRank

In this section, we will then make modification to the previous PageRank, and only make nodes with median original PageRank to be source of teleportation with equal chance.

In this case, we foresee having these two nodes increase their probability of being visited by the random walker. We then performed experiment and had the following result as figure 3.

Nodes with median original PageRank are node 503 and node 529, both has degree 4. From the graph we can tell, probability of being visited peaked at nodes with roughly degree equal to 4, which is consistent with our guess.

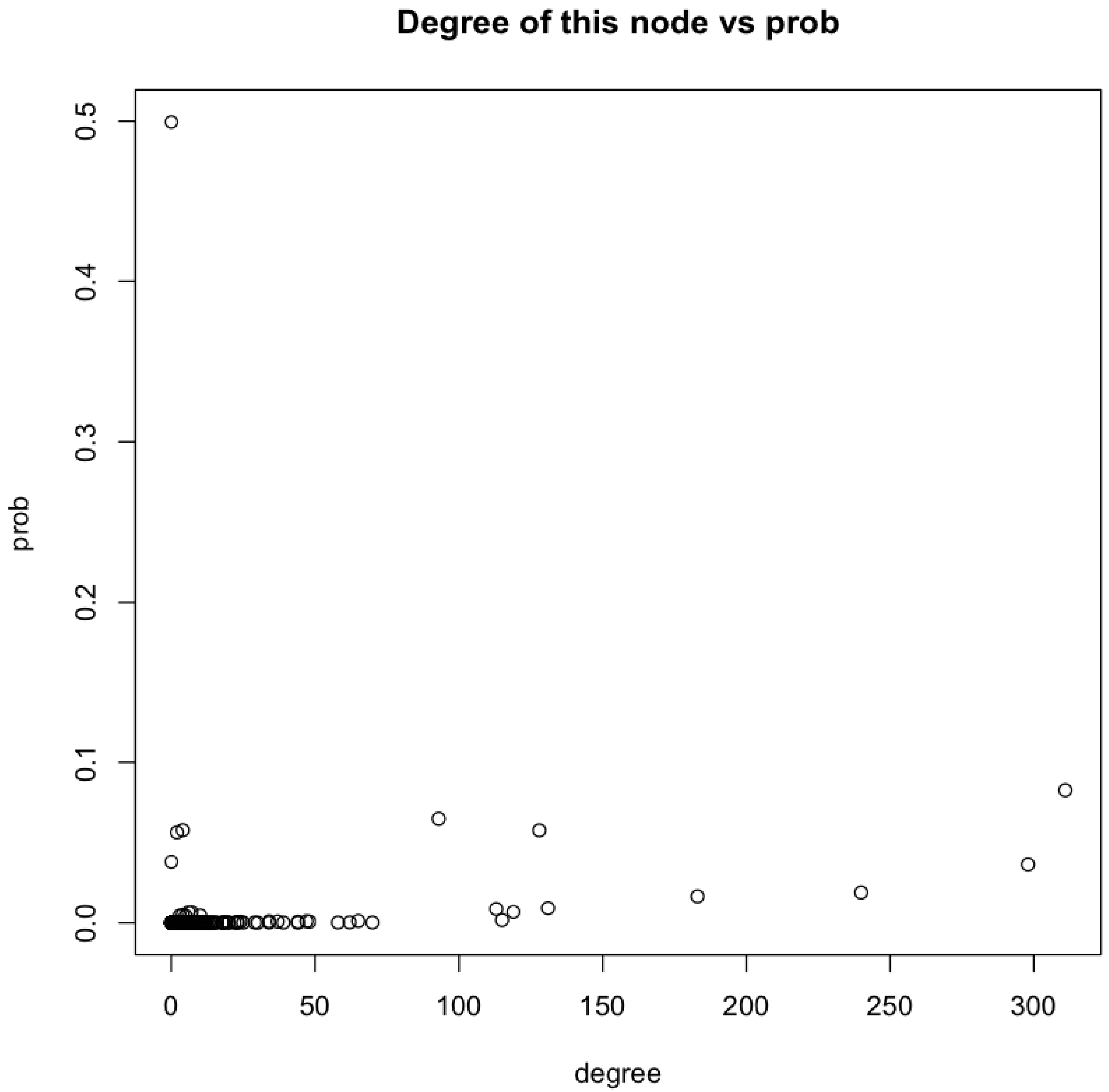


Figure 35 results given modified teleportation policy

4.3 Realistic Personalized PageRank

In this section, we provide a more realistic personalized PageRank given that web user normally only teleports to a set of trusted web pages. Therefore, we modified teleportation policy by teleporting to 10 nodes with highest original PageRank and with the probability proportional to its original PageRank. Then we get result as following:

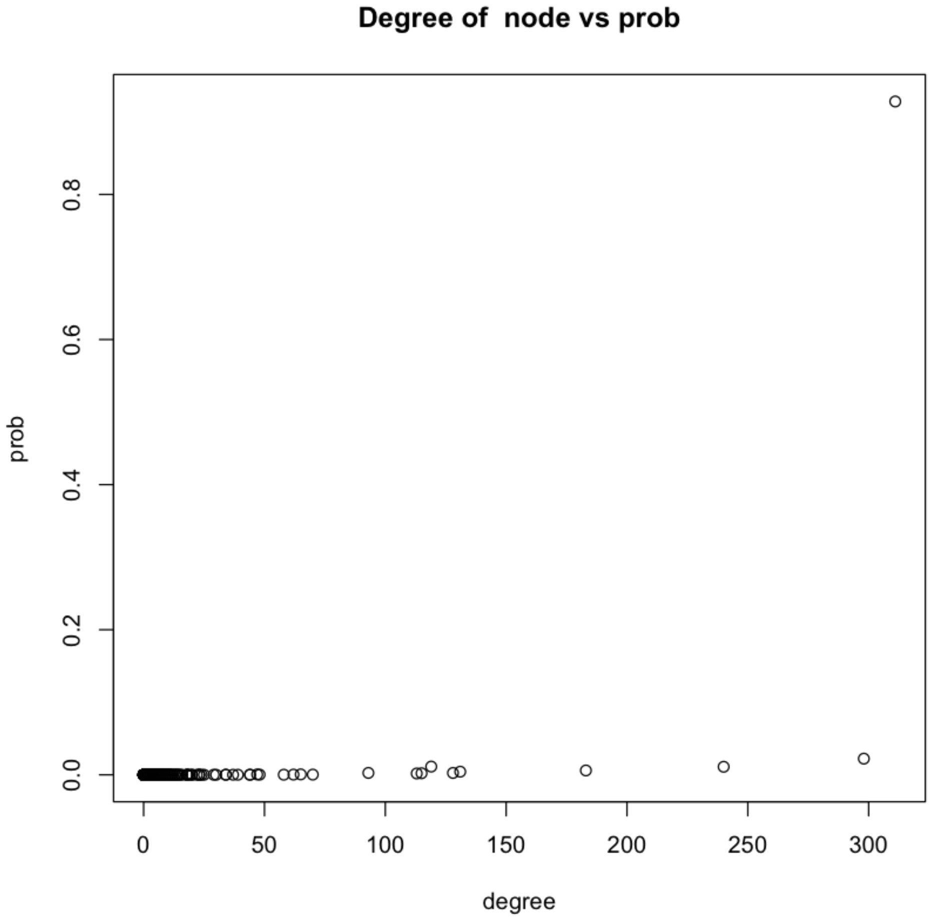


Figure 36 results given realistic teleportation policy

Top	1	2	3	4	5	6	7	8	9	10
Degree	311	299	243	121	187	135	97	132	119	117
Normalized PageRank	0.77	0.06	0.04	0.03	0.02	0.02	0.01	0.01	0.01	0.01

Table 2 top 10 nodes detail

From both figure and table we can tell the nodes with higher original PageRank has more chance to be visited, which is consistent with our intuitive. However, we can tell that popular pages become even more popular, with indication shown in the graph that their probability of being visited are further boosted. And the nodes with the highest original PageRank receives the most chance for being visited.