ECE 232E Project 1
Random Graphs and Random Walks

Jui Chang
Wenyang Zhu
Xiaohan Wang
Yang Tang

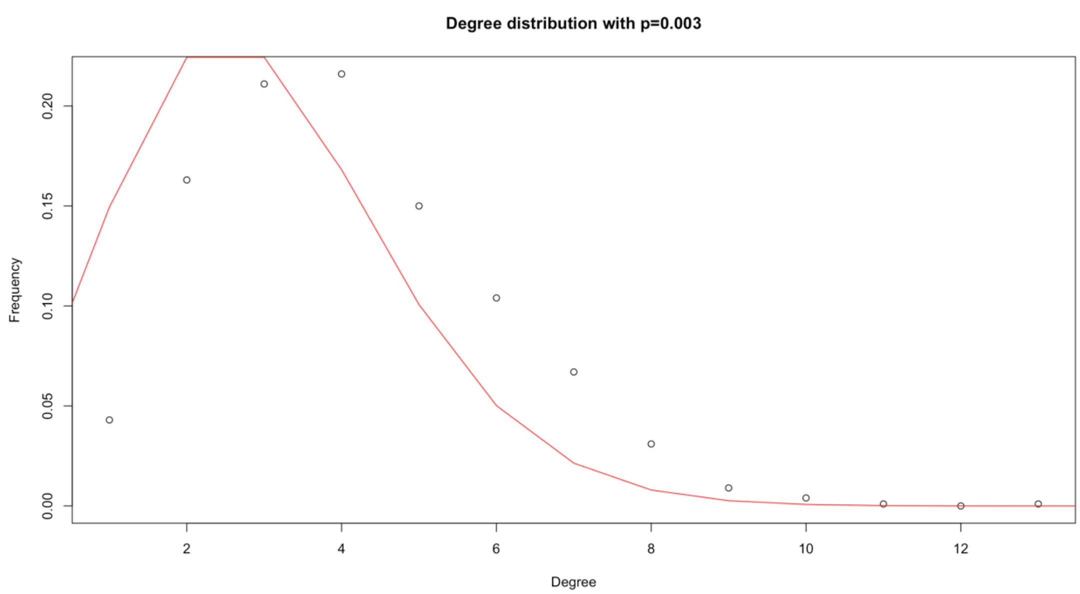
Part I. Generating Random Networks

1. Create random networks using Erdös-Rényi (ER) model

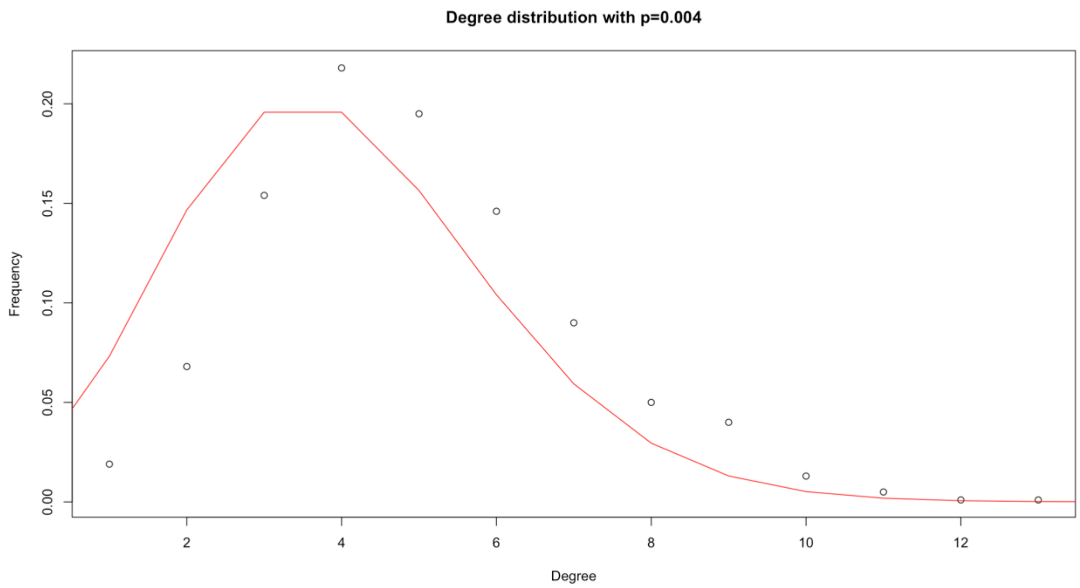
(a) We create undirected random network with n=1000 nodes, with probability of two vertices are 0.003, 0.004, 0.01, 0.05, and 0.1. Plot degree distributions.

From the below Figure 1, we can observe that the degree distribution approaches to binomial distribution, because let K to a random choose node, it can connect to any other node with probability p, just like the definition of binomial distribution. Thus, K ~ B(N − 1, p).

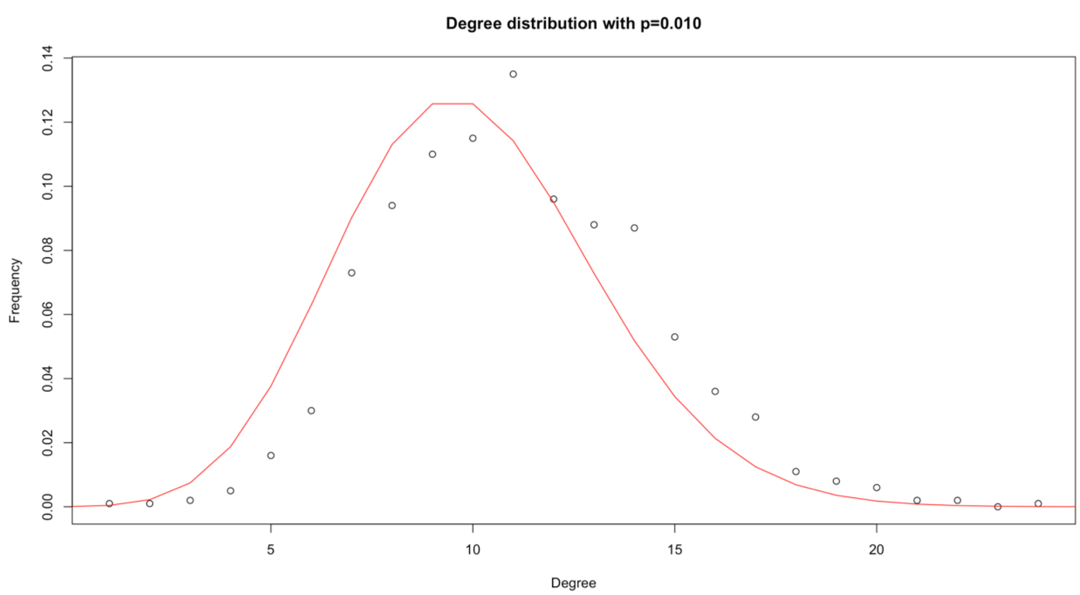
(a) P = 0.003



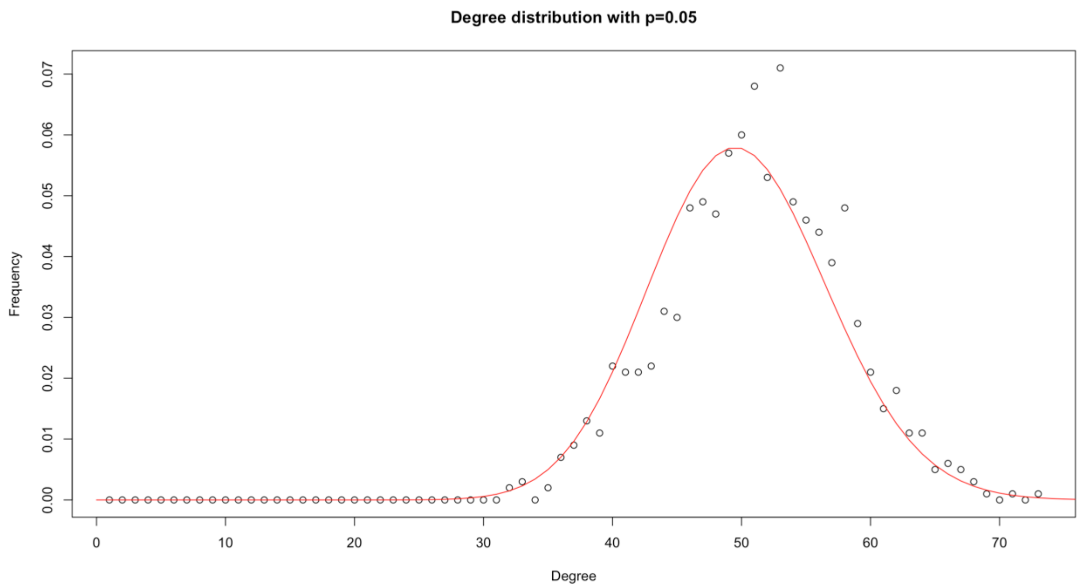
(b) P = 0.004



(c) P = 0.01



(d) P = 0.05



(e) P = 0.1

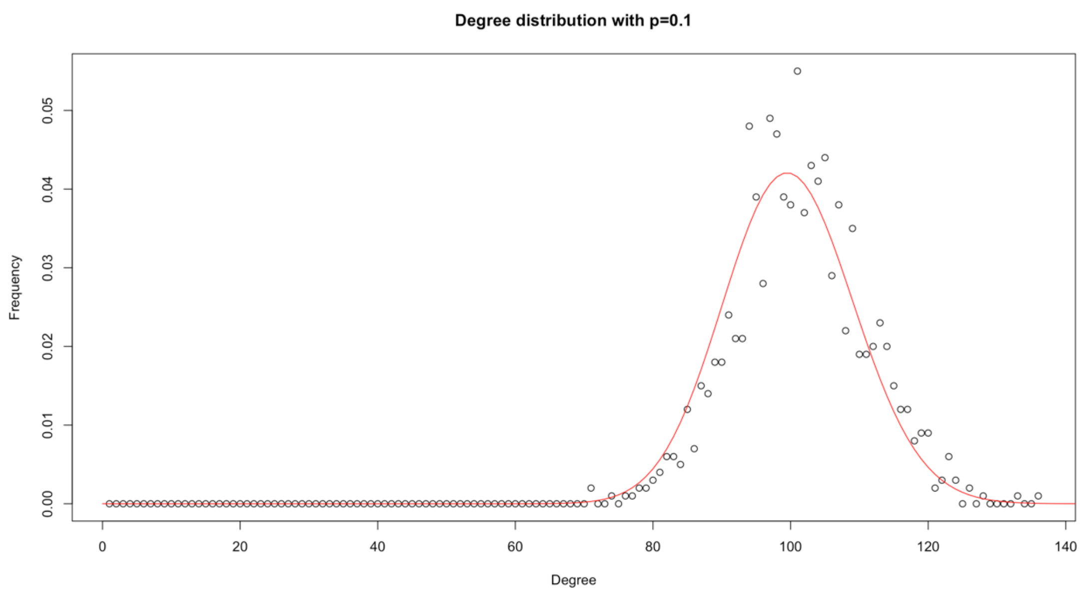


Figure1. Degree distributions of Erdös-Rényi (ER) model with n=1000 nodes and different p

From below table 1, we can observe that the mean and variance of our degree distribution is close to (within 10% error) the theoretical value of binomial distribution:

Binomial: let K be the degree of a random chosen node, $K \sim B(N - 1, p)$, the mean and variance of the distribution is in below formula:

$$E[K] = (N - 1)*p$$

$$\sigma^2 = (N - 1)*p*(1 - p).$$

Table 1. mean and variance of the degree distributions(empirical)

p	Mean	Mean(theoretical)	variance	variance(theoretical)
0.003	3.07	2.997	3.14	2.988009
0.004	4.05	3.996	4.16	3.980016
0.01	9.98	9.99	10.02	9.8901
0.05	49.49	49.95	45.38	47.4525
0.1	99.83	99.9	85.57	89.91

(b) For each p and n = 1000, answer the following questions:

From the below table 2, undirected n=1000 ER random networks is connected on p=0.05 and 0.1, not all random realizations of the ER network connected. We generate n=1000 ER random networks for each p value for 100 times and find the probability that a generated network is connected, and which is shown in below table. Also, the diameter of the GCC is show in below table.

Table 2. Properties of undirected random networks with different p (n=1000)

p	connected	Connected probability	Diameter of GCC
0.003	False	0	13
0.004	False	0	10
0.01	False	0.95	5
0.05	True	1	-
0.1	True	1	-

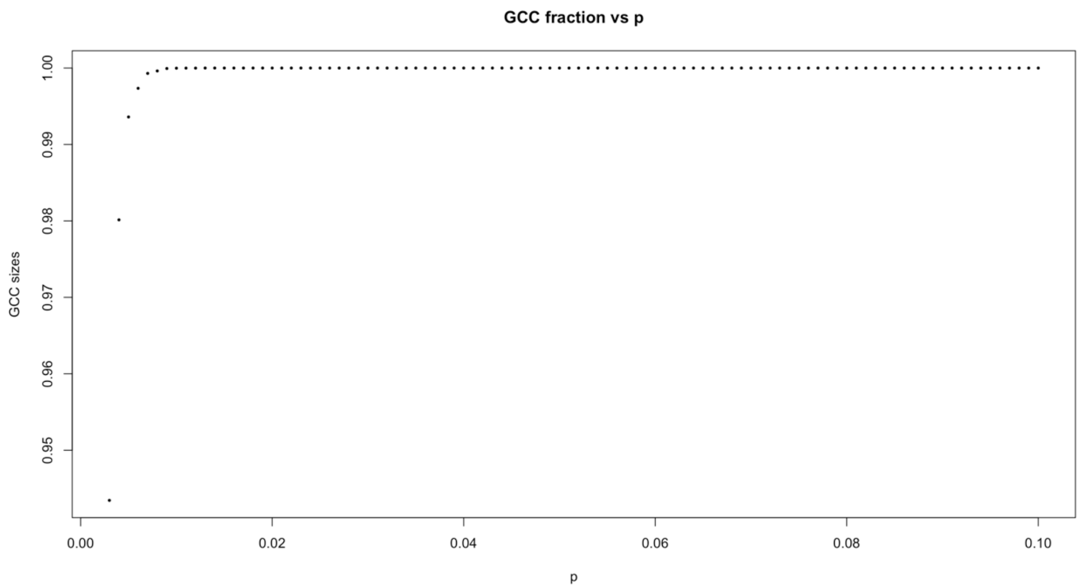
(c) GCC size is highly nonlinear of p, with properties for value $p = O(\frac{\ln n}{n})$. For n=1000, sweep over p values (create 100 random networks) and find GCC sizes. Plot GCC vs p.

$$\frac{\ln n}{n} = \frac{\ln 1000}{1000} = 0.0069$$

At first, I set p start from 0.003, and end at 0.10 with step 0.001 to plot GCC fraction vs p. The figure2(a) shows the general trend of GCC varies. We decide to put more focus on the region p = 0.0069, so I set p start from 0.003, and end at 0.015 with step 0.0001 to plot GCC fraction vs p, and the figure is shown in figure 2(b).

In figure 2(b), we can observe the threshold of p is 0.007. When p is above 0.007, GCC sizes and p are linear related, but When p is below 0.007, GCC sizes and p are logarithmic related. The threshold, p = 0.007 we find on graph is approximate identical to the theoretical value we calculate above, which is 0.0069.

(a)



(b)

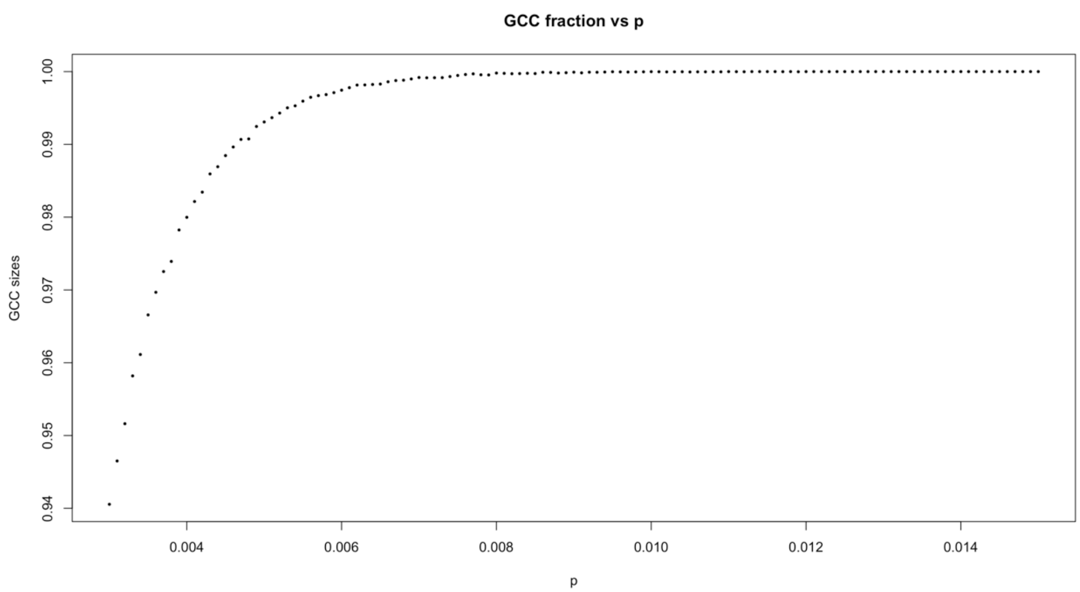


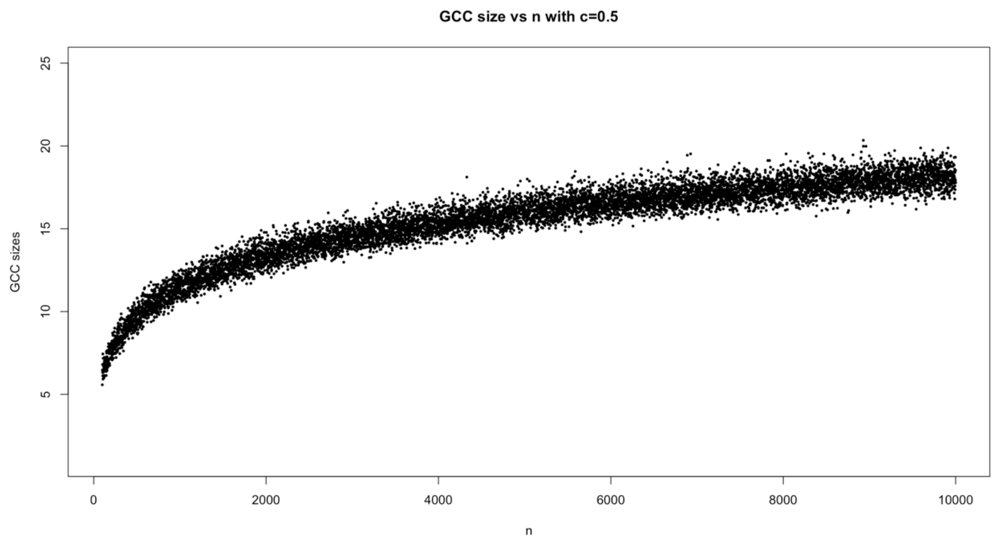
Figure 2 GCC fraction vs p with n = 1000 undirected random networks

(d)

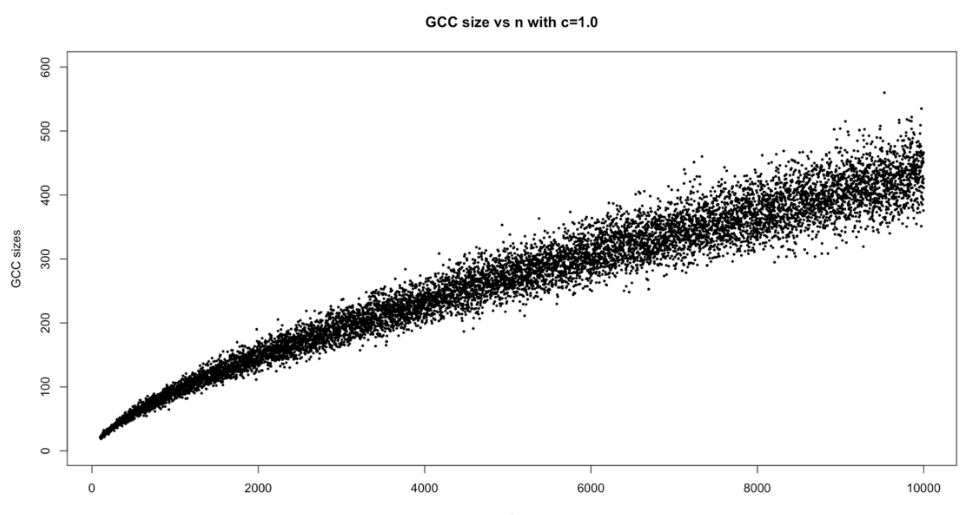
i. Define the average degree of nodes $c = n \times p = 0.5$. Sweep n from 100 to 10000, and plot GCC of ER network vs n.
ii. Repeat the same for $c = 1$.
iii. Repeat the same for $c = 1.1, 1.2, 1.3$, and show the results on the same plot

From figure 3(a), we can observe that with n increase, GCC size increases, but as n increases the slope becomes smooth, which means that the GCC size increases slower. When n large enough (more than 6000), the relation of GCC size and n is close to linear. With c larger, the relation of GCC size and n become linear with lower n threshold (figure 2b). When c = 1.1, 1.2, 1.3, the relation of GCC size and n approaches to linear for all n value. Besides, with c increases, GCC sizes become large for the same n because p = c/n, because when c larger, p become larger, so more nodes are connected, GCC size become larger.

(a) c=0.5



(b) c = 1



(c) c = 1.1, 1.2, 1.3

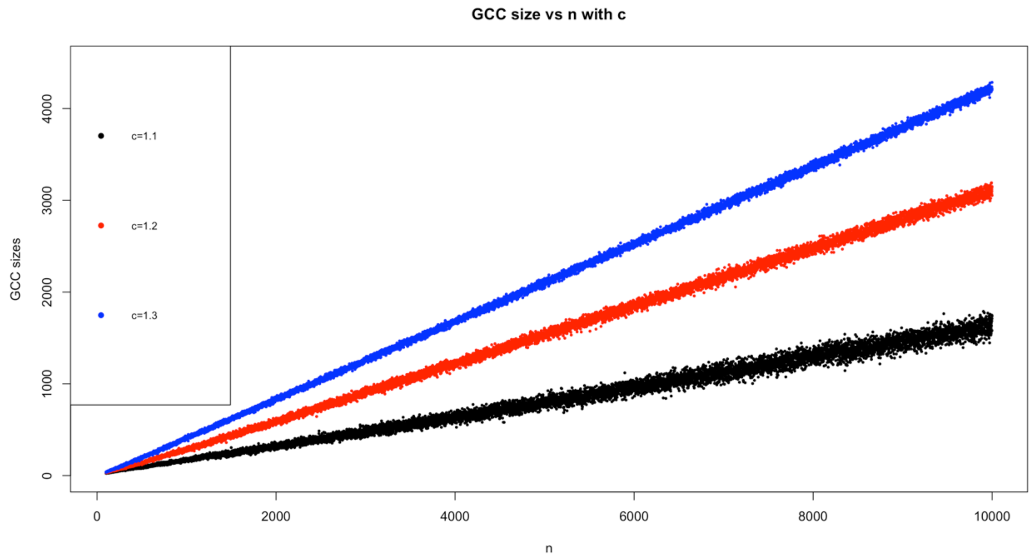


Figure 3. GCC size vs n with different c

2. Create networks using preferential attachment model

(a) From Wikipedia, "The Barabasi-Albert(BA) model is an algorithm for generating random scale-free networks using a preferential attachment mechanism." For this question, we can use the build-in function *barabasi.game* in igraph package to generate an undirected network with $n = 1000$ nodes and $m = 1$ edge attached in each time step. The generated network is as below:

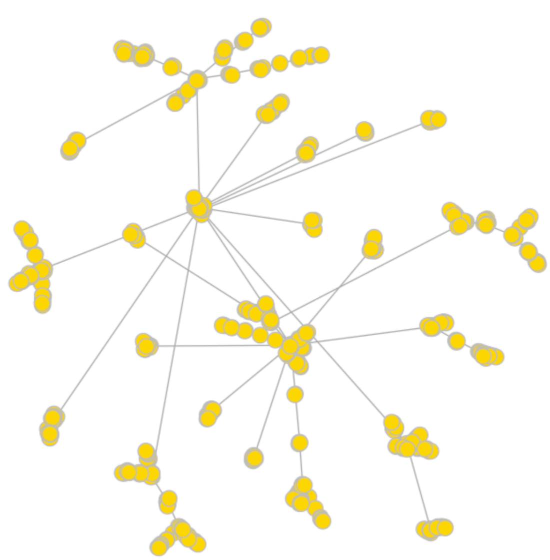


Figure 1. undirected network with 1000 nodes and 1 edge attached each time

With function *is.connected()*, after repeating the generation process for 100 times, we got that the generated network is always connected. And for all possible network generated by this model, they are always connected because preferential attachment model always creates connected network. From the theory of PA model, we know that at each time step, we will add one new node along with m edges. And the edges all need to be connected with the nodes already exist. Thus, with this mechanism, all the nodes will be connected together.

(b) "A network has community structure if it can be divided into several groups of nodes with dense connection internally and sparse connection between groups." And modularity is used to measure the effective of the community detection algorithm. The formula of modularity is

$$\mathrm{Q} = \sum_i (e_{ii} - a_i{}^2) = \sum_i \left(\frac{e_i}{m} - \left(\frac{k_{c_i}}{2m} \right)^2 \right)$$

Where $e_i - represents\ the\ number\ of\ edges\ in\ the\ community\ i;$

$m - represents\ the\ total\ number\ of\ edges\ in\ the\ network;$

$k_{c_i} = \sum_j \deg(v_j)\,, where\ v_j\ are\ in\ the\ community\ i.$

For this question, we can use the build-in function *cluster_fast_greedy()* to do community detection, and then use *modularity()* to measure the modularity. The results are as below:

Table 1. number of communities and modularity of the network (n=1000, m=1)

Number of communities	Modularity
32	0.933

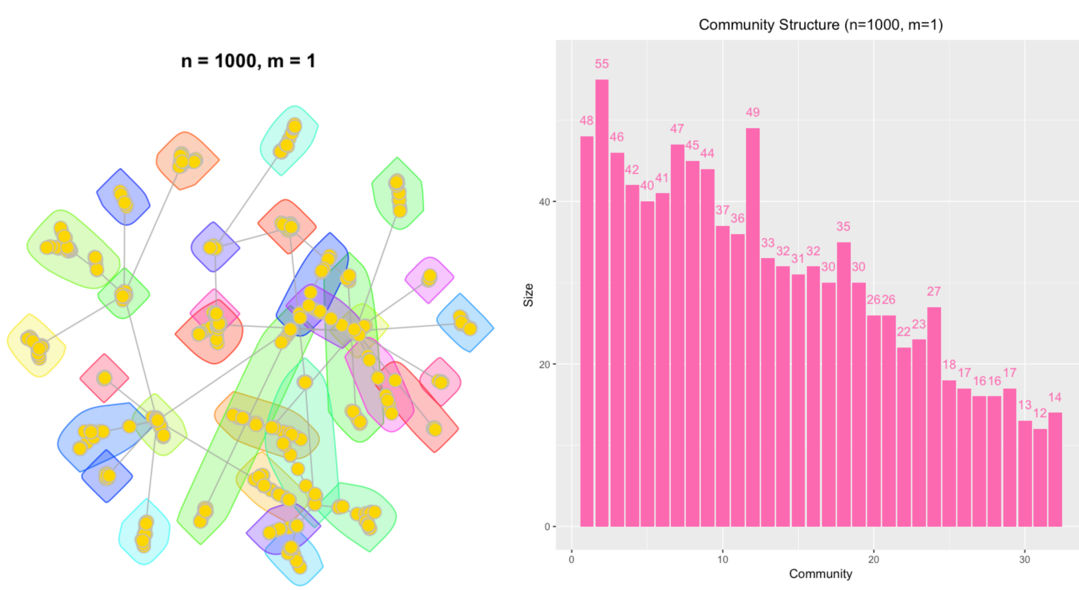


Figure 2. Community Structure (n=1000, m=1)

From the results above, we find that the network achieved a high modularity, which indicates that it's very well clustered into communities due to the properties of preferential attachment model that the newly added nodes tend to connect to the nodes with higher degree.

(c) We use the same model to generate an undirected network with 10000 nodes and $m = 1$ edge attached in each time step. The community structure and modularity are as below:

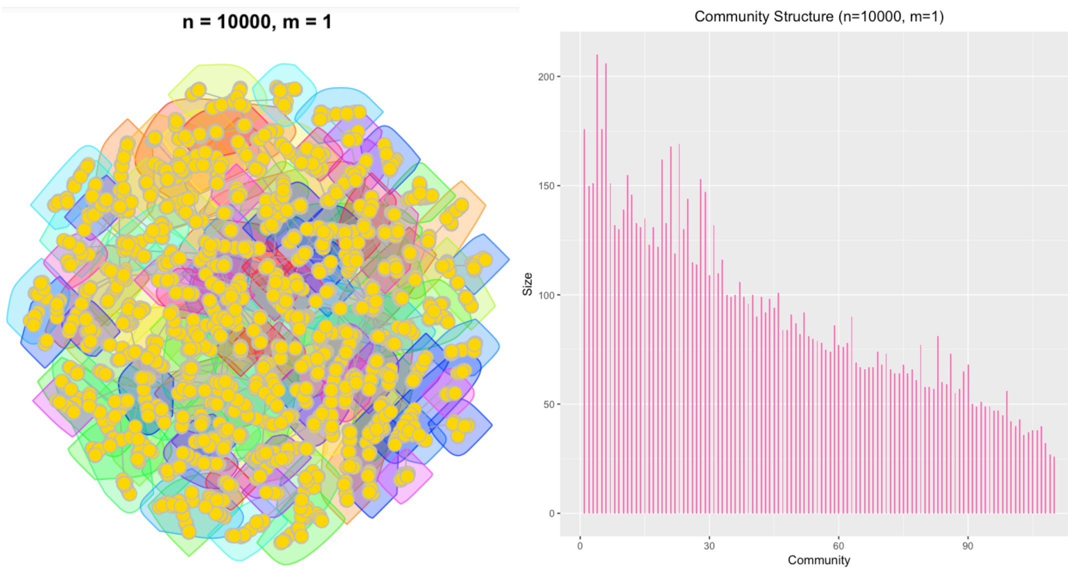


Figure 3/4. Community Structure (n = 10000, m = 1)

Table 2. number of communities and modularity of the network (n=1000&10000, m=1)

Number of nodes	Communities	Modularity
1000	32	0.933
10000	110	0.978

From the results above, we find that there are more communities clustered in the network with 10000 nodes, and the modularity is greater than that of the smaller network. It indicates that with more nodes added, it tends to amplify the number of nodes with higher degree, which intensifies the structure in each community, and sparser the connection between communities, leading to the higher modularity.

(d) Preferential Attachment model is a special case of Power Law model, so the degree distribution is

$$\lim_{k \to \infty} P_k \propto \frac{1}{k^\gamma}, \gamma > 0$$

Take log algorithm on each side, and we get

$$\log P_k = \log \frac{1}{k^\gamma} = \log 1 - \log k^\gamma = -\gamma \log k$$

We can find that $\log P_k \sim \log k$ is linear, and the slope is $-\gamma$.

1) n = 1000 nodes

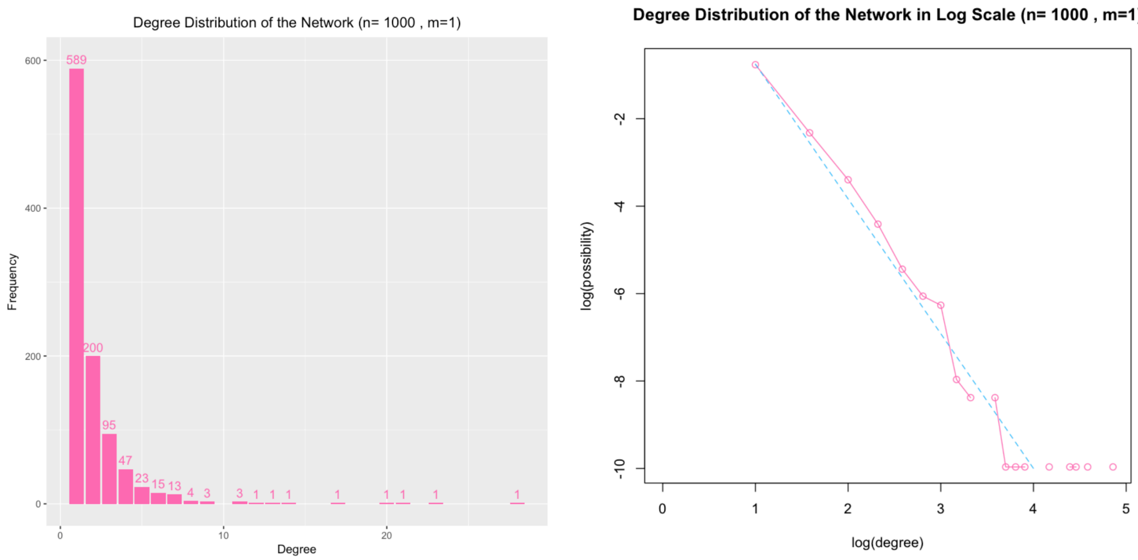


Figure 5/6. original degree distribution and in log scale (n=1000, m=1)

The slope of the plot is

$$\text{slope} = -\gamma \approx \frac{-9.97 - (-0.76)}{3.91 - 1} = -3.16$$

2) n = 10000 nodes

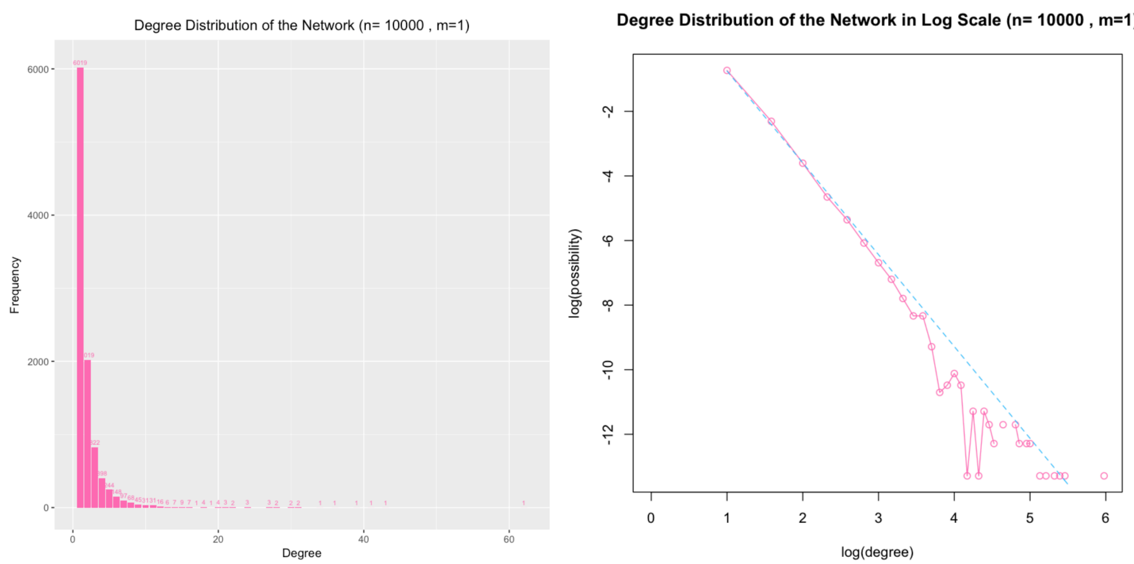


Figure 7/8. original degree distribution and in log scale (n=10000, m=1)

The slope of the plot is

$$\text{slope} = -\gamma \approx \frac{-13.29 - (-0.73)}{5.21 - 1} = -2.98$$

From the above results, the slope is around -3, and γ is about 3, which is consistent to the theory that the degree distribution of PA model is about $\frac{1}{k^3}$.

(e) For this question, we did as the following steps:

1. Randomly pick a node i
2. Randomly pick a neighbor j of node i
3. Record the degree of node j
4. Iteration 1~ 3 for *vcount()* times, then plot the degree distribution

1) n = 1000 nodes

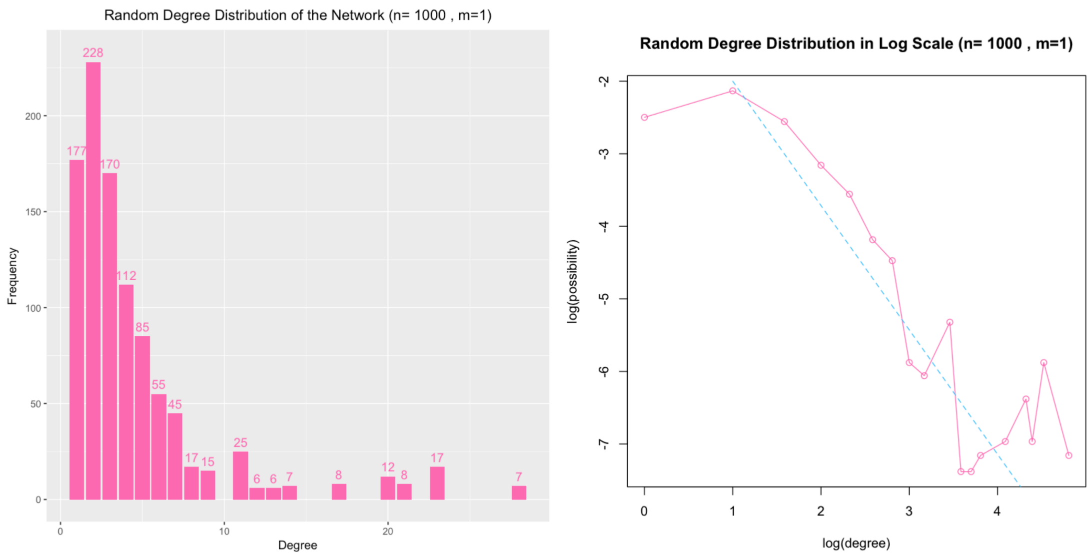


Figure 9/10. randomly original degree distribution and in log scale (n=1000, m=1)

The slope of the plot is

$$\text{slope} = -\gamma \approx \frac{-6.97 - (-2.13)}{4.09 - 1} = -1.57$$

2) n = 10000 nodes

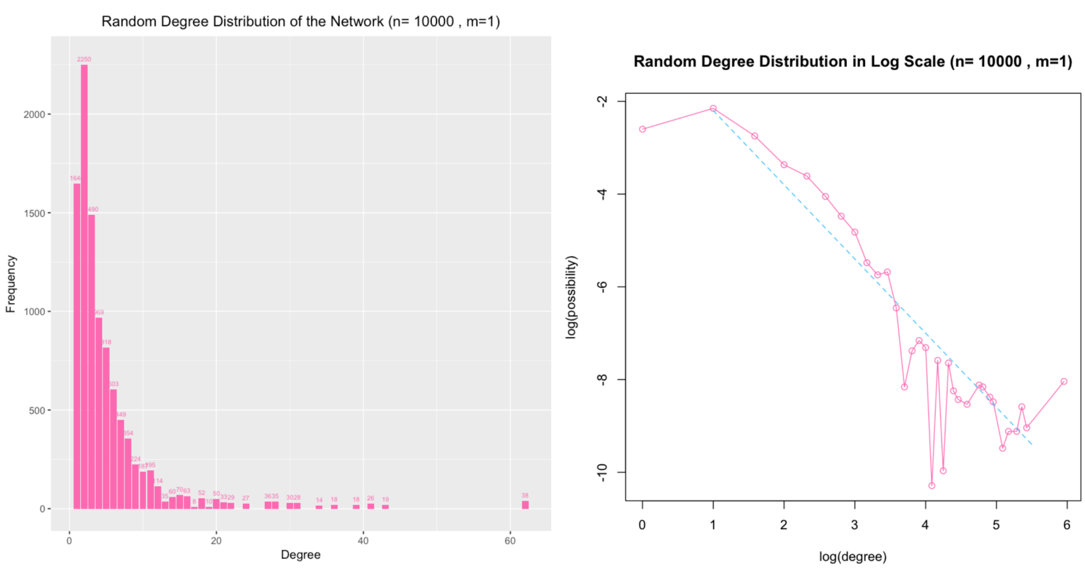


Figure 11. randomly original degree distribution and in log scale (n=10000, m=1)

The slope of the plot is

$$slope = -\gamma \approx \frac{-9.12 - (-2.15)}{5.17 - 1} = -1.67$$

3) Node degree distribution VS randomly picking degree distribution

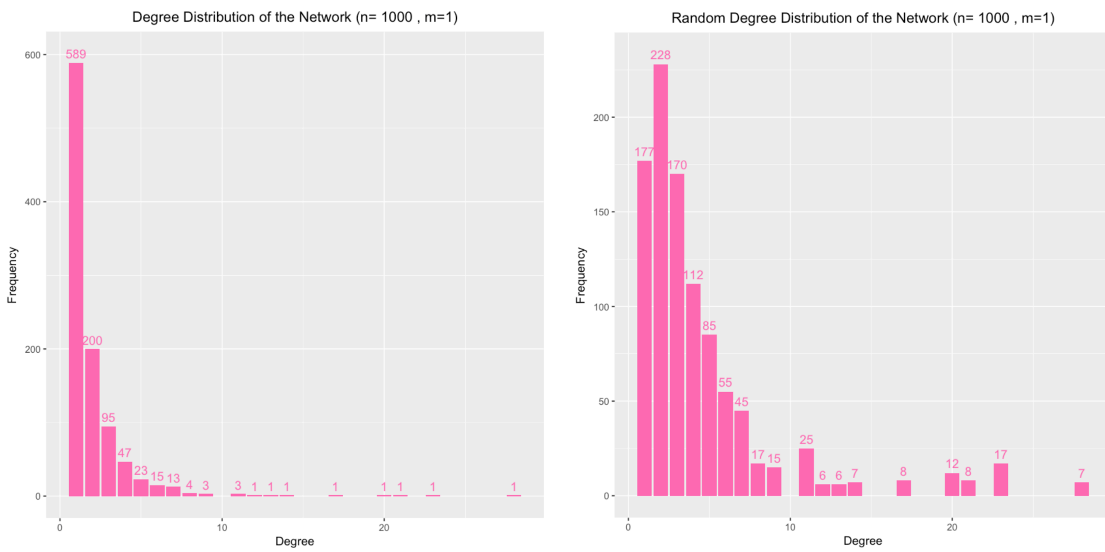


Figure 13. compare frequency~degree figure (n=1000, m=1)

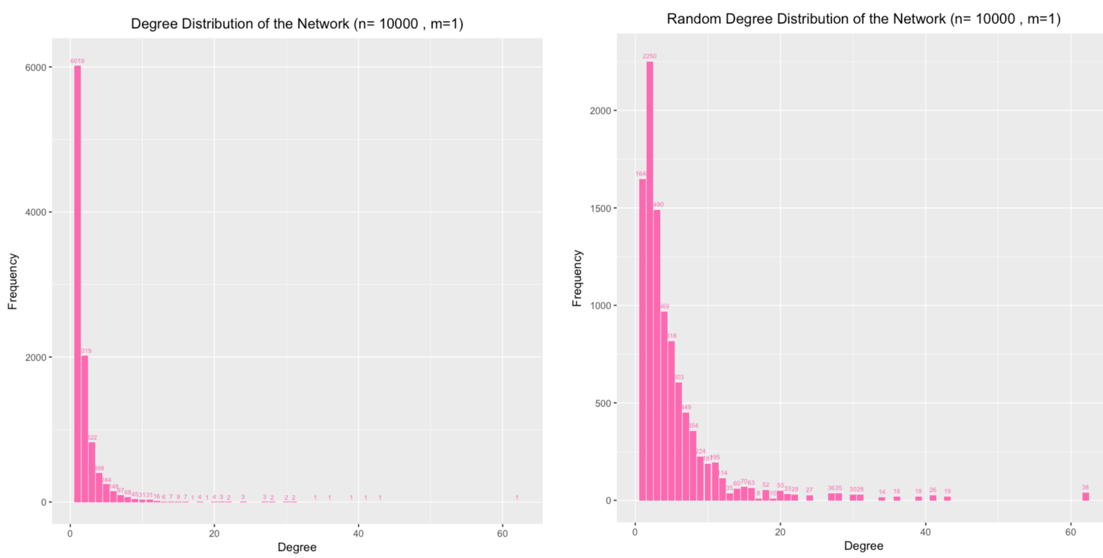


Figure 14. compare frequency~degree figure (n=10000, m=1)

For the node degree distribution, we pick and calculate the degree of each node in the network, so the probability of choosing each node is equal. For the randomly picking method, in the first randomly picking process, the probability is equal. While for the second randomly picking process, in general, the nodes with higher degree have higher probabilities to be picked up.

From the compare figures, we can find that when n = 1000, the node degree distribution has the largest frequency for about 600, while for randomly picking method, the largest number is about 230. When n = 10000, the node degree distribution has the largest number 6000, but for randomly picking method, the largest one is about 2000. Thus, in the frequency~degree figure, this randomly picking method has more flat and uniform distribution, and the node degree distribution is sharper.

Table 3. compare the power γ (m=1)

n	Node degree distribution	Randomly picking degree distribution
1000	3.16	1.57
10000	2.98	1.67

From the above results, we find that with this method, the power γ is about 1.6 which is smaller than that in the last question $\gamma = 3$.

For PA model, we have $\lim_{k\to\infty} P_k \propto \frac{1}{k^\gamma}$, for randomly picking method, when k increases, its degree P_k should increase accordingly. So the power γ should be less than that in the node degree distribution formula.

(f) For this question, we randomly generated a network based on PA model and iterated for 1000 times. For each time, we recorded the degree of each node and their corresponding age = 1000 – index, and then calculate the expectation for each node (age/timestep). The result is as below:

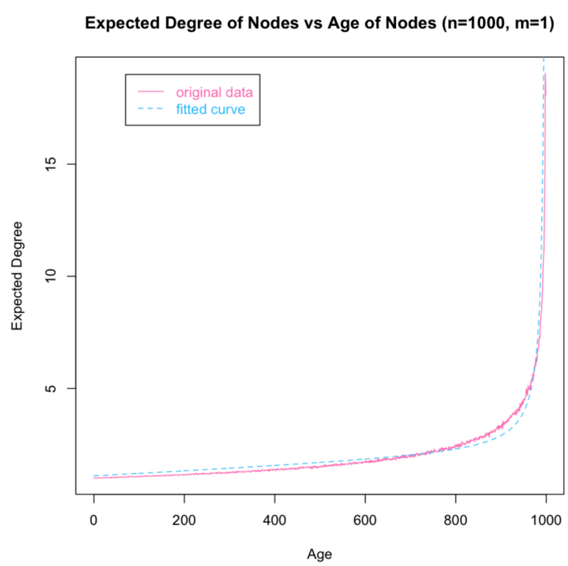


Figure 15. Expected Degree of Nodes vs Age of Nodes (n=1000, m=1)

From the above figure, we find that in general, the older the node is, the higher degree it's expected to have. The functional relationship between degree and age is approximately as

$$Expected\ Degree = \frac{100}{1000 - Age} + 0.001 * Age + 1$$

(g) For this question, we generated 4 new network and explored their properties.

1. Generate 4 networks and check their connectivity

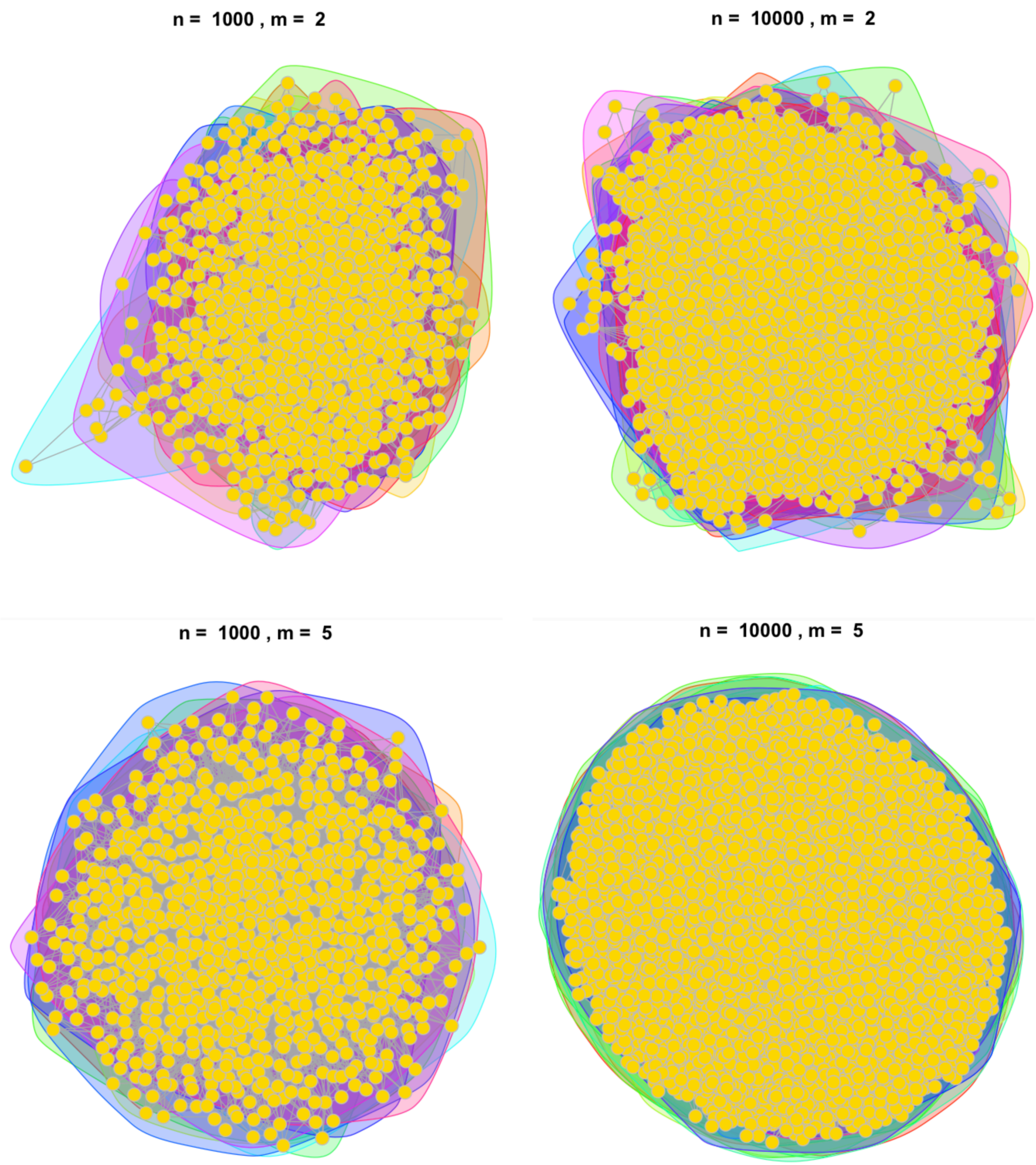


Figure 16. four newly generated networks

These 4 networks are all connected as the 2 former networks because they are all generated with preferential attachment model, and the network with this model is always connected.

2. Find community structure and measure modularity

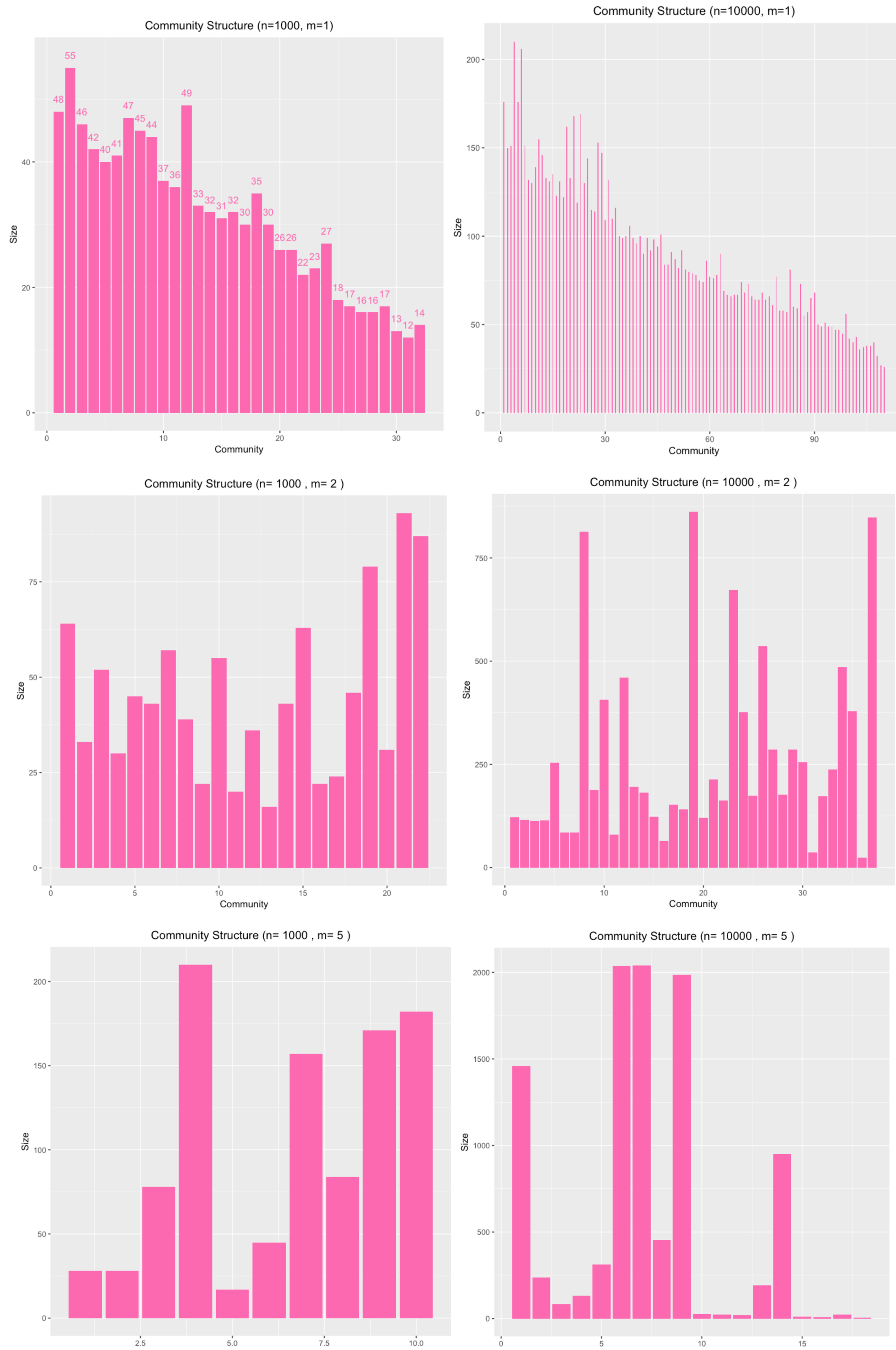


Figure 17. Community Structure Compare

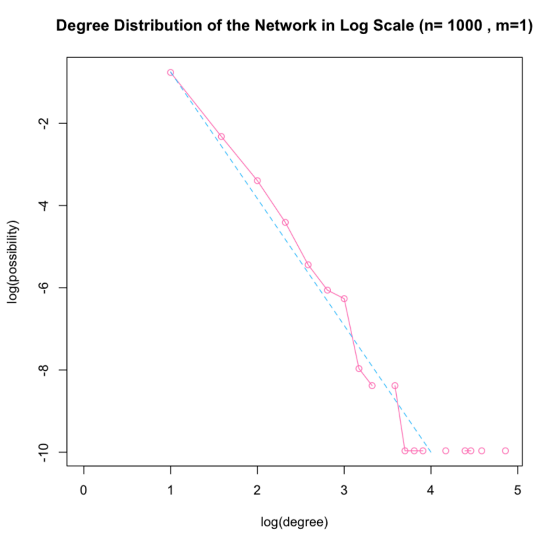
Table 4. Modularity Compare

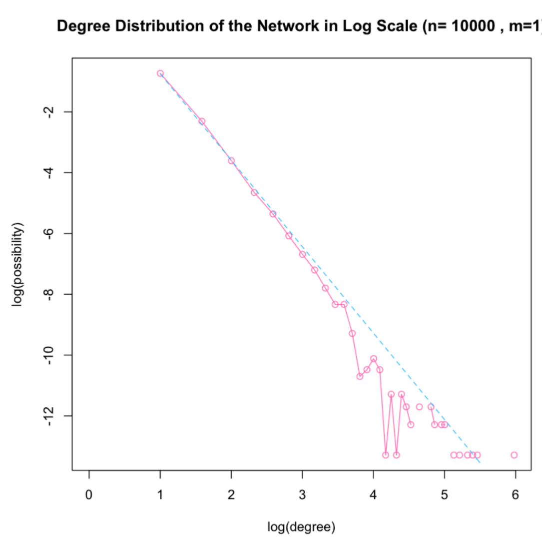
m	n	Number of communities	Modularity
1	1000	32	0.933
	10000	110	0.978
2	1000	22	0.515
	10000	37	0.530
5	1000	10	0.278
	10000	18	0.276

From the results above, we find that the more nodes to added, the higher modularity the network is. Because it tends to amplify the number of nodes with higher degree, which intensifies the structure in each community, and sparser the connection between communities, leading to the higher modularity.

While for the value m, the more edges a new node along with, the lower modularity the network is. Since a newly added node can bring more edges, it can connect with more nodes with higher degree, which will cause the degree distribution flatter and more uniform, and it may increase the connection between different communities and it's more difficult to detect communities. Thus, in this project, the networks with m = 1 have higher modularity and more communities detected.

3. Plot degree distribution and estimate the slope





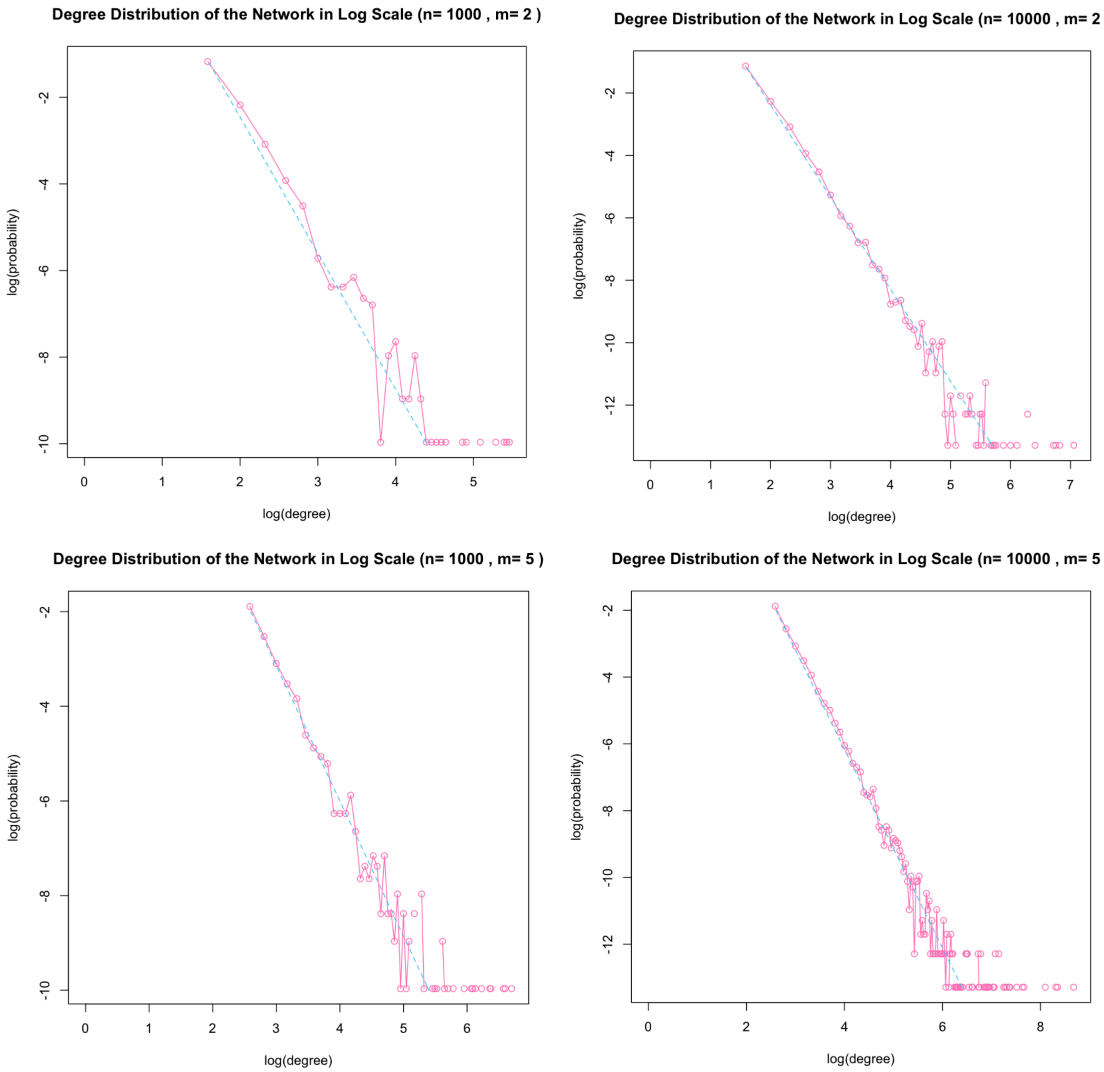


Figure 18. Degree Distribution compare

Table 5. Estimation of the slope

m	n	slope
1	1000	-3.16
	10000	-2.98
2	1000	-3.13
	10000	-2.97
5	1000	-2.95
	10000	-3.05

From results above, we find that the larger n and m are, the more degrees the network has. Besides, the slopes of the degree distribution in log scale are all approximate -3, and the power $\gamma \approx 3$ which fits the PA model.

4. Randomly picking nodes and plot the degree distribution

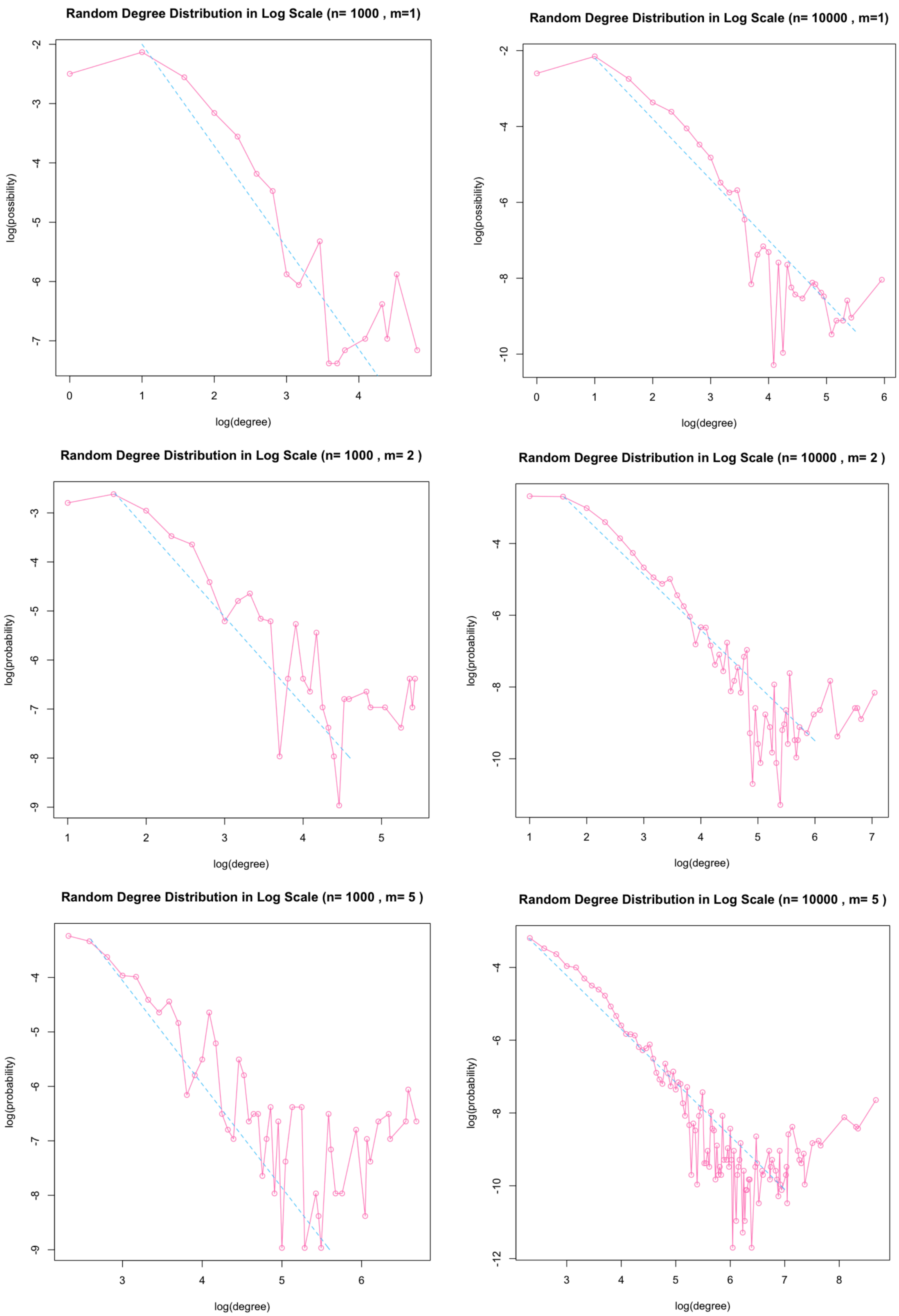


Figure 19. Degree Distribution of randomly picking method

Table 6. Estimation of slope for randomly picking method

m	n	slope
1	1000	-1.57
	10000	-1.67
2	1000	-1.80
	10000	-1.55
5	1000	-1.83
	10000	-1.51

From the results above, we find that with this randomly picking method, we can get the slope of the degree distribution curve as about -1.6, and the power $\gamma \approx 1.6$.

5. Estimate the expected degree of node at specific time step

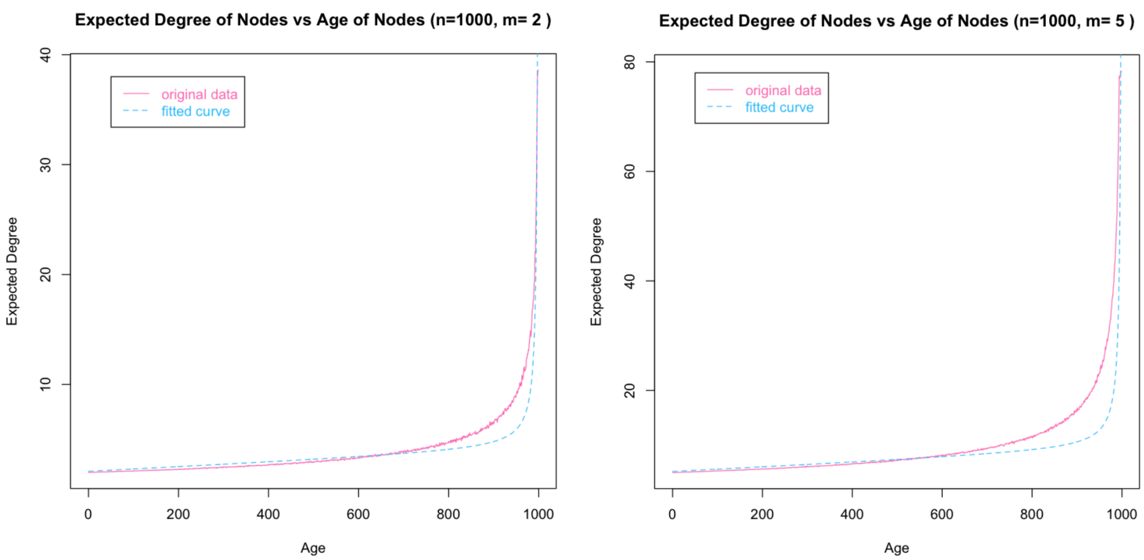


Figure 20. Expected degree vs Age

Table 7. curve fitting for expected degree vs age curve

m	Fitting function
2	$Expected\ Degree = \frac{100}{1000 - Age} + 0.002 * Age + 2$
5	$Expected\ Degree = \frac{200}{1000 - Age} + 0.004 * Age + 5$

(h) In this part, we use the build-in function *sample_degseq()* to generate a new network. To avoid multiple edges, we use "*vl*" and "*simple.no.multiple*" methods separately. Below are our results:

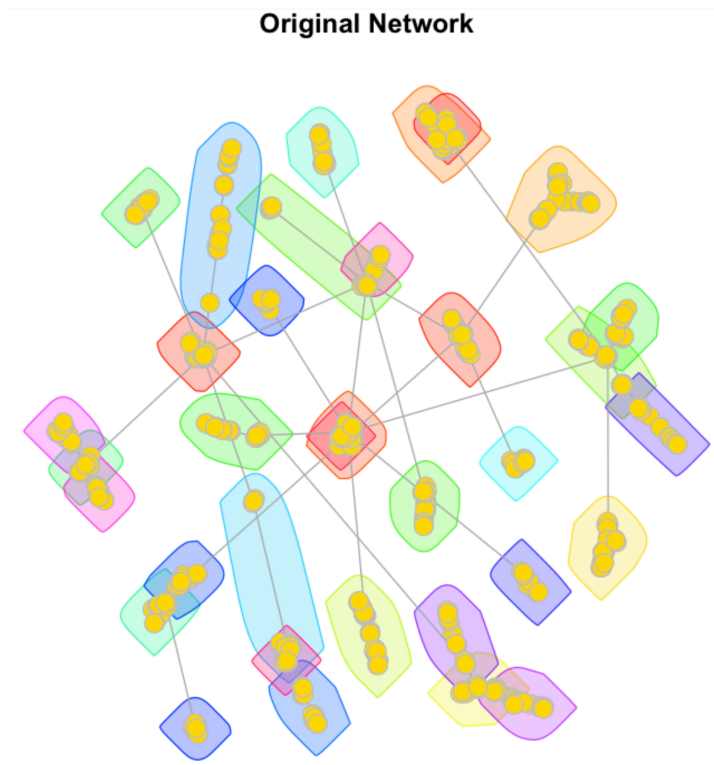


Figure 21. Original network (n=1000, m=1)

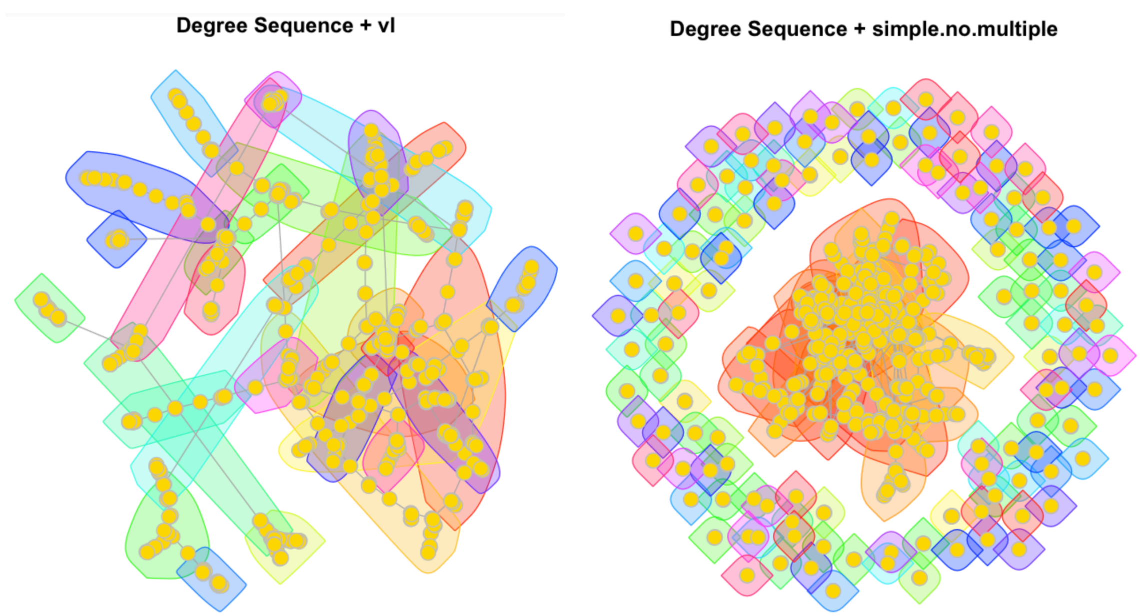


Figure 22. Degree sequence + stub-matching networks

Table 8. Properties of three networks

Model	Connectivity	Communities	Modularity
PA model	True	34	0.933
Degree sequence + vl	True	31	0.935
Degree sequence + simple.no.multiple	False	142	0.844

From the results above, we find that PA model and degree sequence model are both belong to random power-law model, so they all obey the formula

$$\lim_{k\to\infty} P_k \propto \frac{1}{k^{\gamma}}, \gamma > 0$$

While there are also some differences between PA model and degree sequence model. With the theory of PA model, the newly added nodes are always connected to the existed nodes, which results that the generated network is always connected. For degree sequence model, the nodes are all generated first, and then they will use stub-matching strategy to connect the edges. In this process, the network may not be connected because it may include self-loop. So there will be more communities detected and the modularity may be lower with those scattered nodes.

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

(a) In this part, we create a new network in which each time a new vertex is added, it creates m links to old nodes and the probability that an old node is chosen depends on its degree (PA model) and age. The mathematical formula is

$$\mathrm{P}[i] \sim (c{k_i}^{\alpha} + a)\left(d{l_i}^{\beta} + b\right)$$

Where k_i is the degree of vertex i in the current time step, and l_i is the age of vertex i. Here, we set the parameters as $\mathrm{n} = 1000, \mathrm{m} = 1, \alpha = 1, \beta = -1, \mathrm{a} = \mathrm{c} = \mathrm{d} = 1, \mathrm{b} = 0$, so the formula becomes as

$$\mathrm{P}[i] \sim (k_i + 1)\left({l_i}^{-1}\right)$$

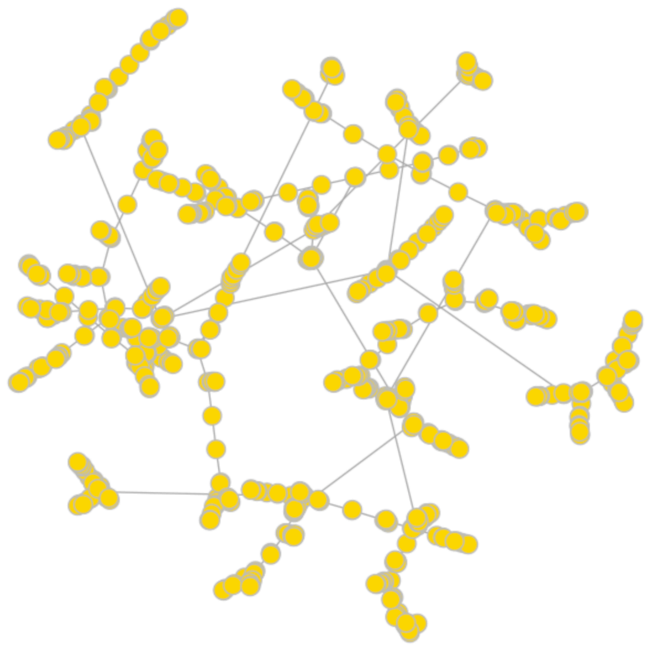


Figure 23. PA + age model-based network

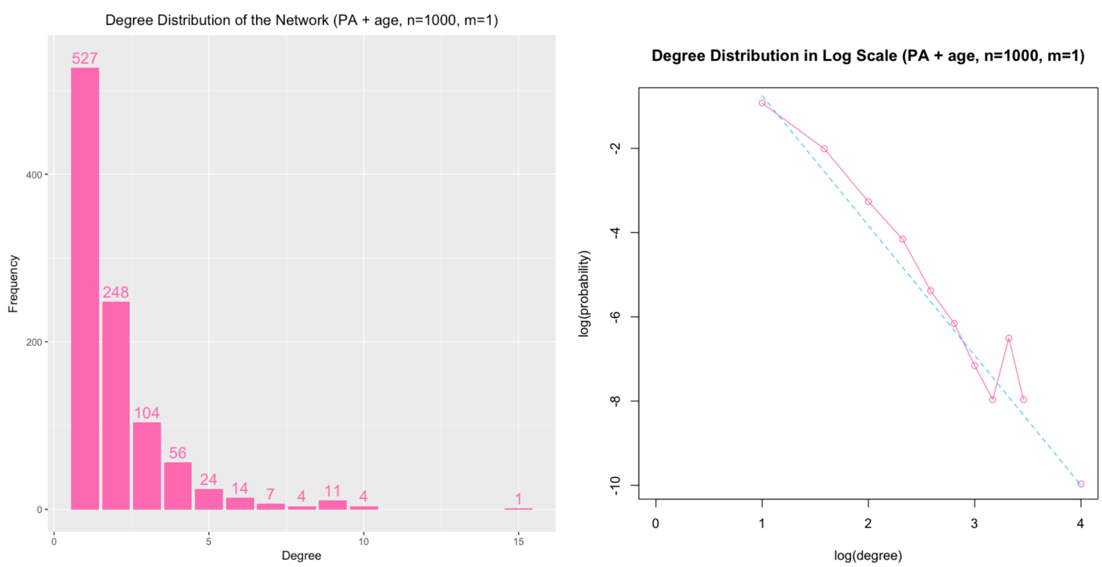


Figure 24/25. original degree distribution and in log scale (PA + age)

From the figures above, we find that the degree distribution of this model is similar to that of power law model. The estimation of the slope is

$$\text{slope} = -\gamma \approx \frac{-9.97 - (-0.92)}{4 - 1} = -3.02$$

The power law exponent is $\gamma \approx 3.02$.

(b) Using fast greedy method, we can find the community structure of the network is as below:

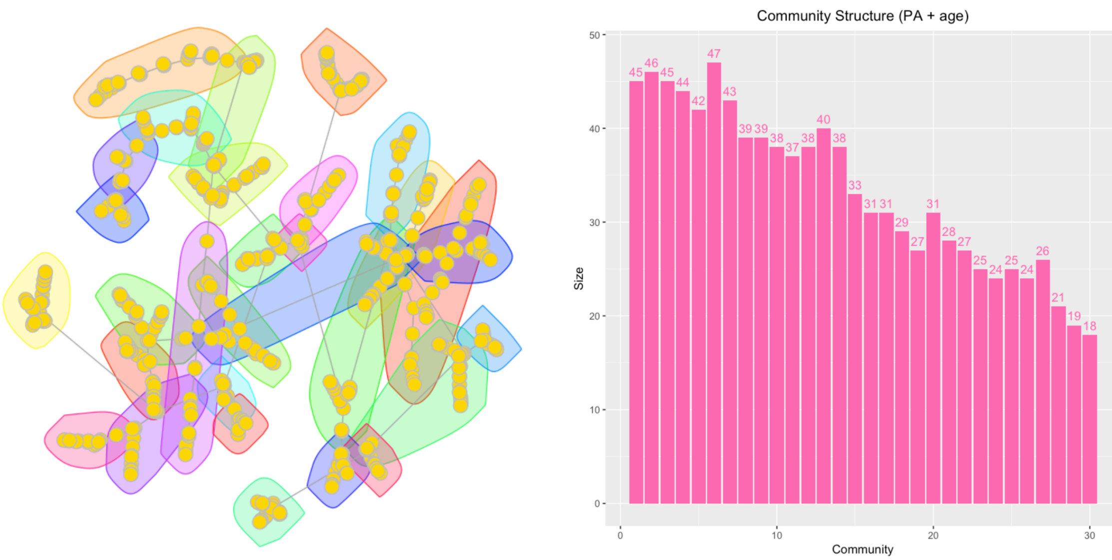


Figure 26. Community structure (PA + age)

Table 9. number of communities and modularity of the network (PA + age)

Number of communities	Modularity
30	0.935

From the results above, we can find that the community structure and modularity are similar to that of PA model. They both achieve high modularity.

Part II. Random Walk on Networks

1. Random walk on Erdös-Rényi Network

(a) We first need to create the random network with the below parameters: node number = 1000, probability for connecting any pair of nodes = 0.01, directed = False. Therefore, we used the function *random.graph.game* from *library('igraph')* to create the corresponding network.

(b) A random walker started from a randomly selected node and took t steps to reach its final destination. For each destination, the shortest path lengths s(t) are calculated. With different starting points, s(t) for each t are different. So, we plot the mean and variance for this distance with different starting nodes v.s. the number of step t.

Considering the running time, we chose **100 steps for this 1000 nodes network**. Also, for each t, we need to determine the looping time (the number of different random starting nodes). According to our observation, as the looping time goes up, the mean and variance after a fixed t-value may arrive at some relatively more stable mean and variance. For example, looping time = 1000 gets a better result than looping time = 100. However, the running time goes up as well. For this question, we chose the **looping time to be 1000**. But for larger network with node number = 10000, we will choose a smaller portion considering the running time needed.

The plots are shown below.

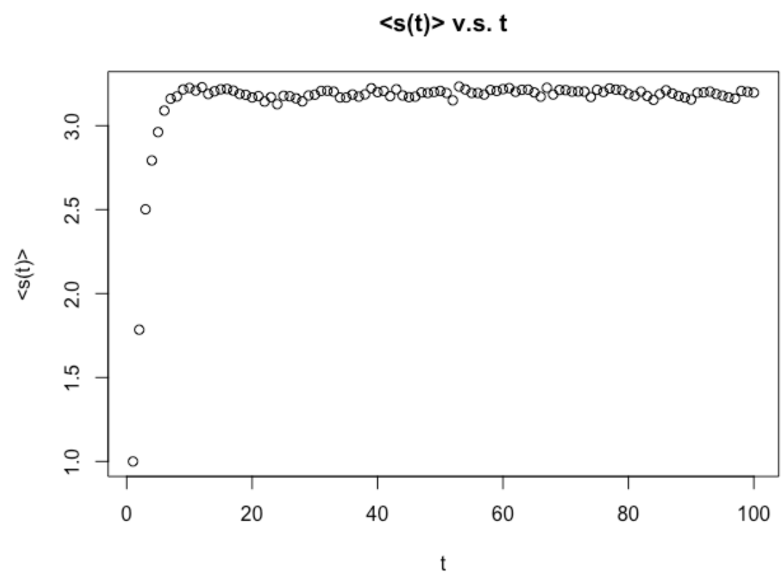


Fig 1. Average distance v.s. t steps

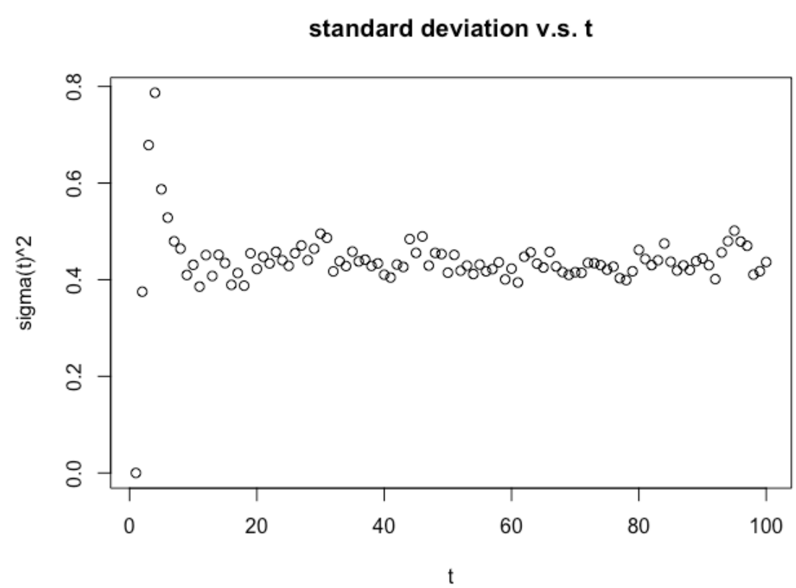


Fig 2. Standard deviation v.s. t steps

From the above two graphs, we can find that the mean value of shortest path length arrives at about 3.3, and the standard deviation arrives at about 0.42 after t = 10 steps. Therefore, 100 steps here are more than enough to get the conclusion.

(c) Continuing with the question in part(b) above, we also need to calculate the degree distribution of the destination nodes and compare that with the degree distribution of the graph. So, we still chose 100 steps and 1000 looping times here.

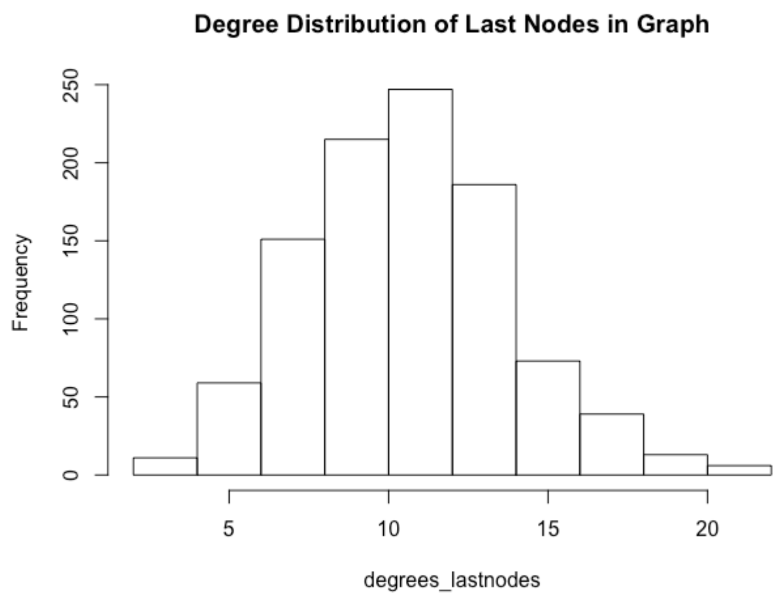


Fig 3. Degree distribution of the destination nodes

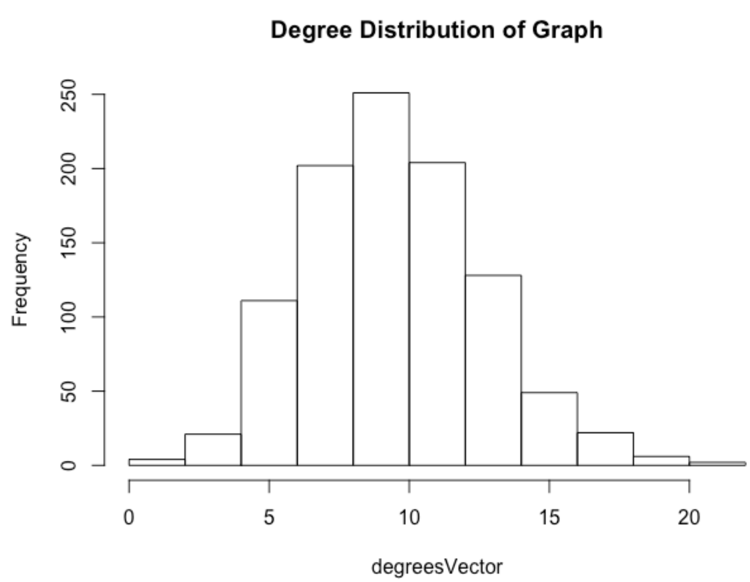


Fig 4. Degree distribution of the original graph

We also plot our results of the degree Distribution for the original graph. Comparing both histogram plots, we may conclude that the degree distribution of the nodes reached at the end of the random walk highly depends on and is also related to the degree distribution of the graph.

(d) Repeat (b) with 100 and 10000 nodes. For 100 nodes graph, we chose the step to be 500 and the looping time to be 1000. The plots are shown below:

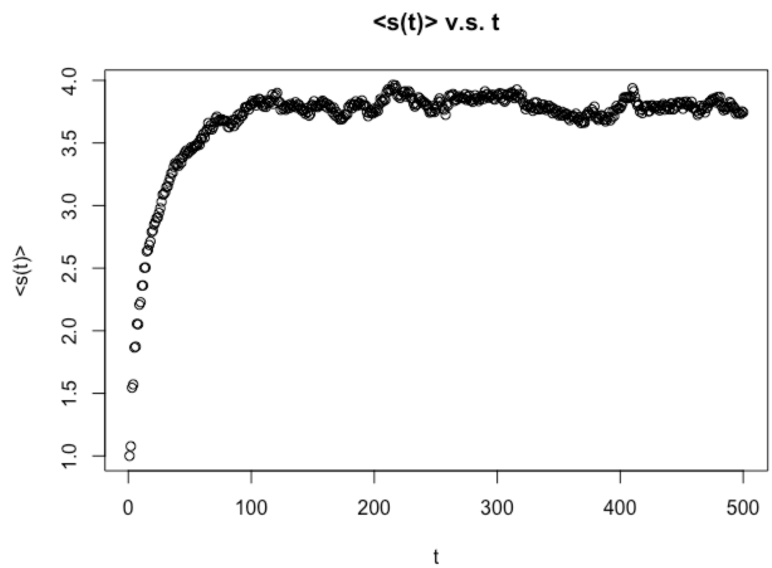


Fig 5. Average distance v.s. t steps for 100 nodes graph

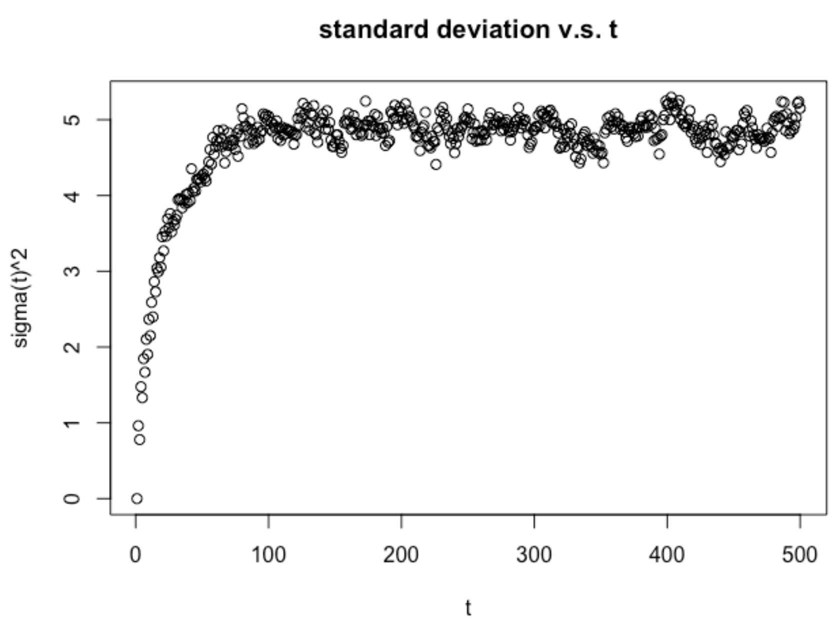


Fig 6. Standard deviation v.s. t steps for 100 nodes graph

For the 10000 nodes graph, in order to save the running time as well as get a relatively reasonable plot, we chose the step to be 1000 and the looping time to be 1000. The plots are shown below:

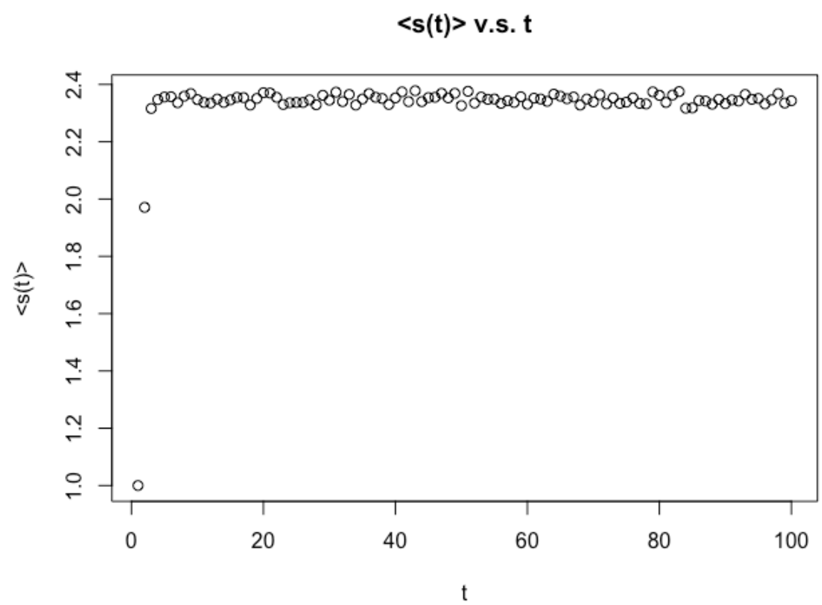


Fig 7. Average distance v.s. t steps for 10000 nodes graph

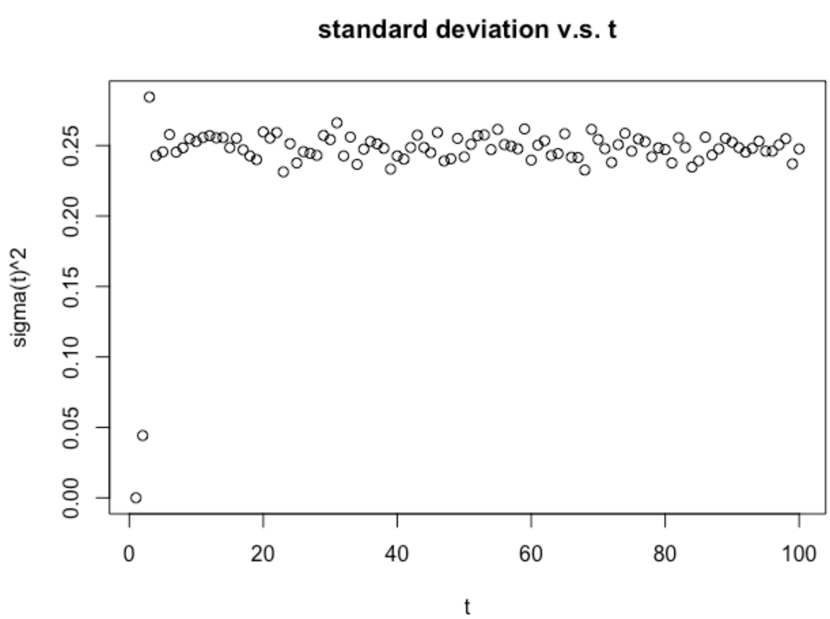


Fig 8. Standard deviation v.s. t steps for 10000 nodes graph

Comparing the results between figures 1,2,5,6,7,8, we can find: with a smaller graph (100 nodes), it takes longer steps for the mean and standard deviation to be stable. In order to show the stable state, we let the range of steps t to be 0-500, which is 5 times larger than both other graphs (graphs with 1000 nodes and 10000 nodes). And the steady state reaches after 80 steps. Moreover, the small graph also has a relatively high standard deviation, which also means that it is not as stable as the other two larger graphs.

On the contrary, graphs with 1000 nodes or 10000 nodes reach a stable mean and standard deviation value in fewer steps. For example, the graph with 1000 nodes reach a steady state after the step 10, while the graph with 10000 nodes reach the steady state after only step 4. The standard deviations are 0.42 and 0.25 respectively for both graphs, which are a lot smaller than that in the graph with 100 nodes.

Intuitively, for the same random network model, a smaller graph should have a smaller diameter. Therefore, with larger diameter, the larger graph has more information containing its degree distribution. In this way, the random walker tends to reach the destination with stable constraints in fewer steps, and the graph becomes more reliable.

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

(a) The procedures are similar to those in the 1st part. To begin with, we used the *barabasi.game* function from *igraph* library. The parameters are defined as: node number = 1000, directed = False, m = 1.

(b) We also need to plot the average and standard deviation for random v.s. the step t in this problem. Here, we chose the step range to be 500, and the looping time to be 1000. Both plots are shown below.

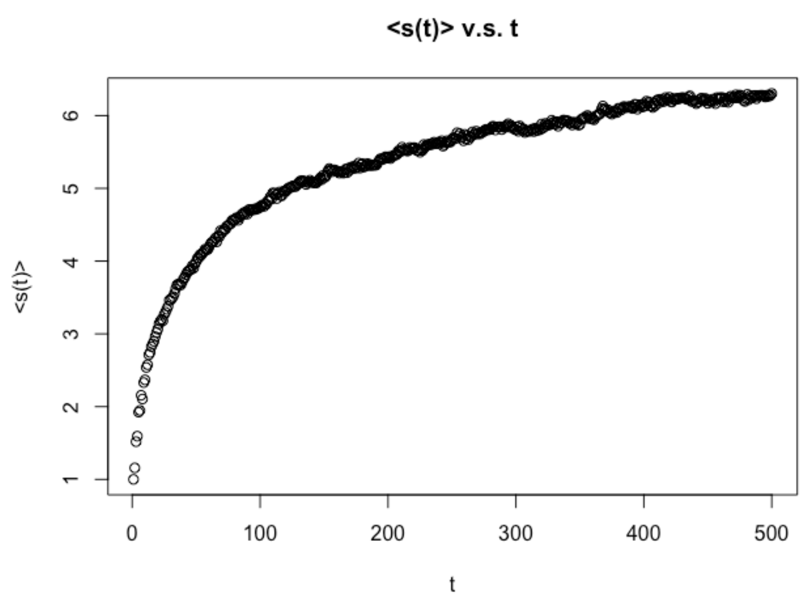


Fig 9. Average distance v.s. t steps for preferential attachment network

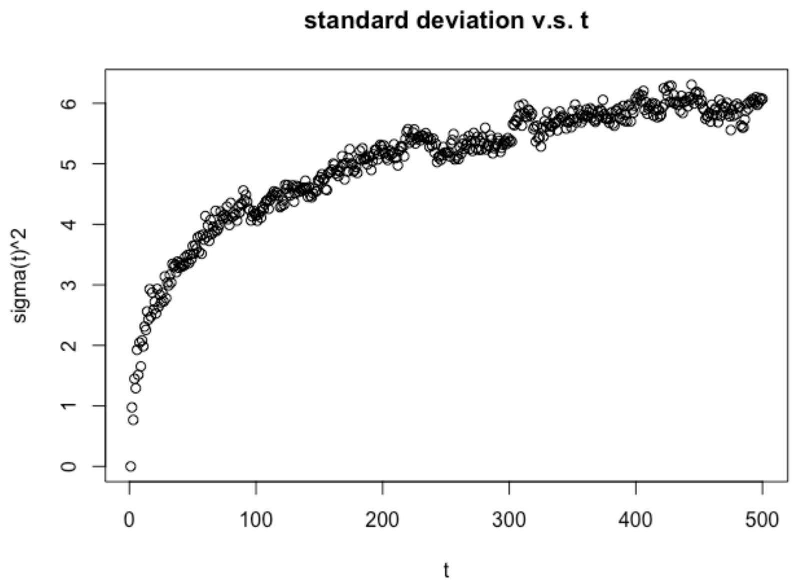


Fig 10. Standard deviation v.s. t steps for preferential attachment network

The result is different from the result in 1(b). The reason is that the preferential attachment model is different from the Erdös-Rényi Network. The indegree for each node in the preferential attachment model follows the power law distribution, and the outdegree for each node equals to m = 1. However, for Erdös-Rényi Network, the degree distribution follows the normal distribution. We can find the histogram for both distributions in part 1(c) and part 2(c).

(c) We will plot the degree distribution of the destination nodes and compare that with the degree distribution of the graph. So, we still chose 500 steps and 1000 looping times here.

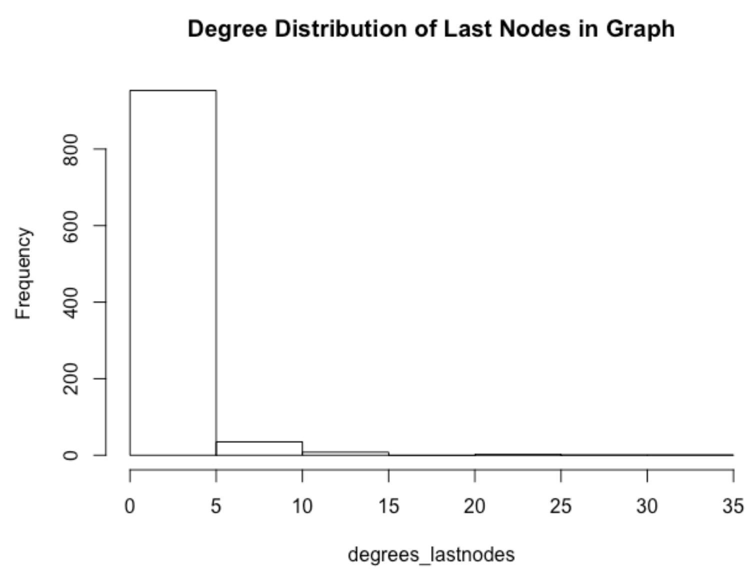


Fig 11. Degree distribution of the destination nodes for preferential attachment network

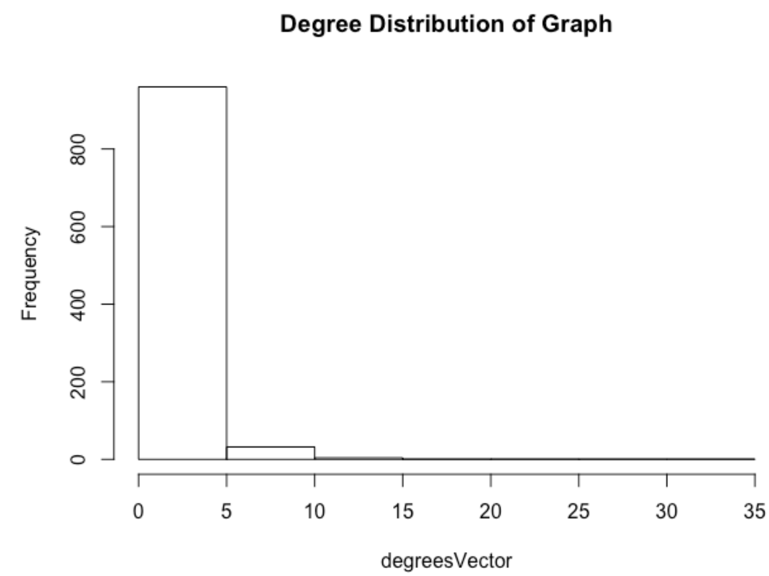


Fig 12. Degree distribution of the original graph for preferential attachment network

Comparing both histogram plots, the conclusion is the same as 1(c): the degree distribution of the nodes reached at the end of the random walk highly depends on the degree distribution of the original network. Both distributions follow the power law distribution.

(d) Repeat (b) with 100 and 10000 nodes. For 100 nodes graph, we chose the step to be 500 and the looping time to be 1000. The plots are shown below:

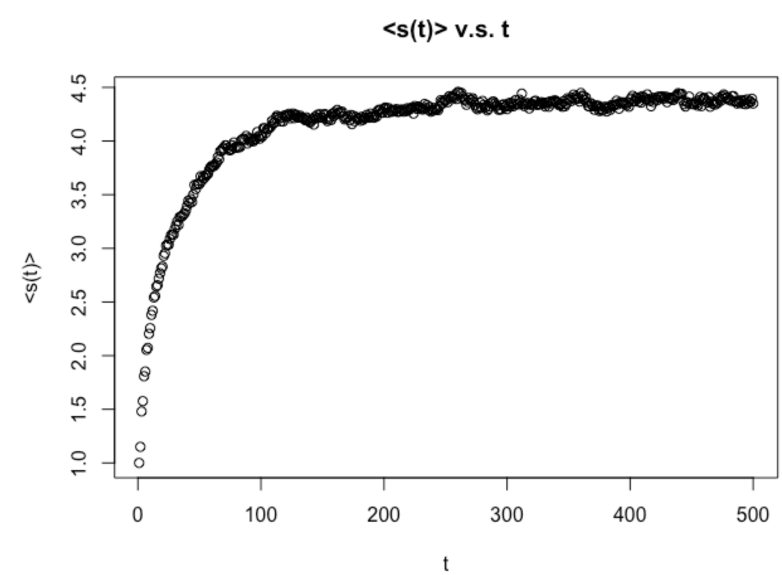


Fig 13. Average distance v.s. t steps for 100 nodes graph

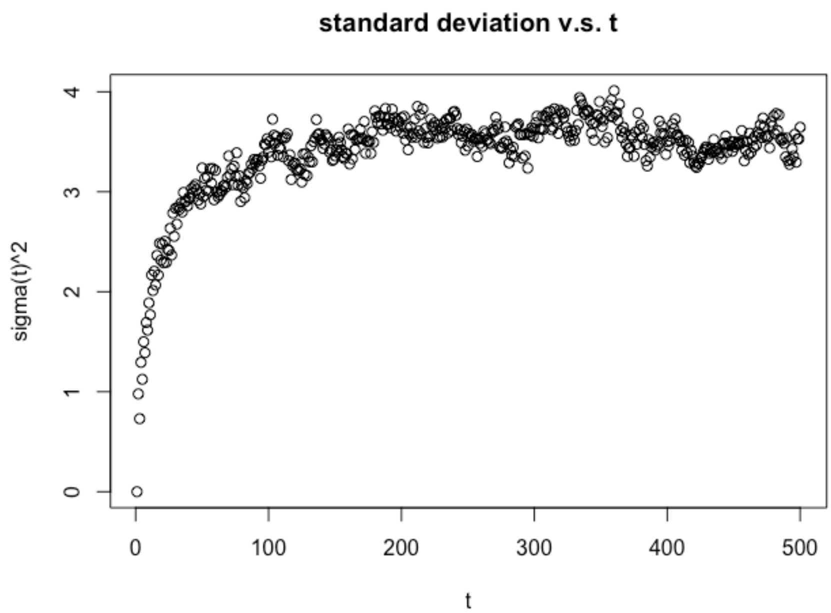


Fig 14. Standard deviation v.s. t steps for 100 nodes graph

For the 10000 nodes graph, we also chose the step to be 500 and the looping time to be 1000. The plots are shown below:

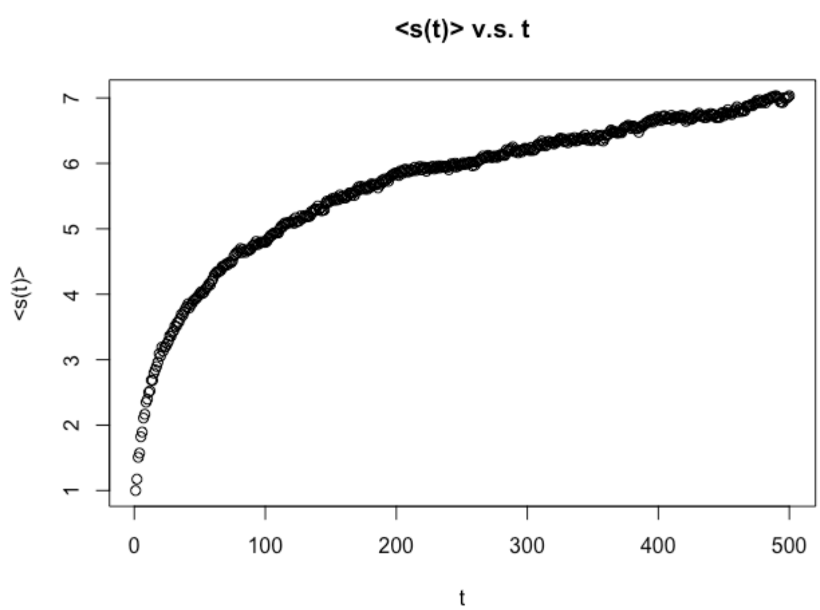


Fig 15. Average distance v.s. t steps for 10000 nodes graph

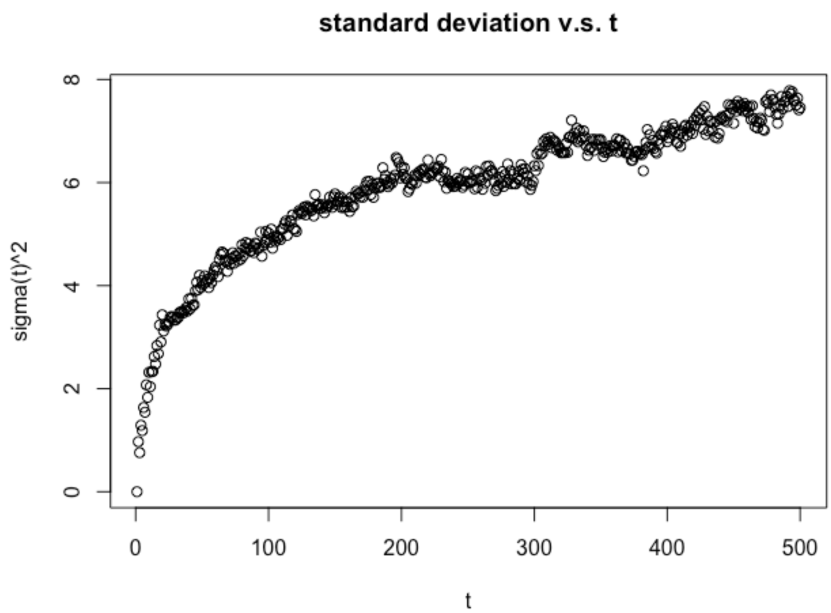


Fig 16. Standard deviation v.s. t steps for 10000 nodes graph

Comparing the results between figures 9,10,13,14,15,16, this time we can find: with a smaller graph (100 nodes), it takes smaller steps for the mean and standard deviation to be relatively stable. The steady state reaches after about 100 steps. Moreover, the small graph also has a relatively smaller standard deviation 3.6, which also means that it is more stable than the other two larger graphs.

On the contrary, graphs with 1000 nodes or 10000 nodes reach a stable mean and standard deviation value in longer steps. For example, the graph with 1000 nodes reach a steady state after about 100 steps, while the graph with 10000 nodes reach the steady state after more than 500 steps (we can get this conclusion from figure 15 and 16, but we choose not to plot more steps due to the running time). The standard deviations are around 6 and more than 8 respectively for both graphs, which are much larger than that in the graph with 100 nodes.

Therefore, for preferential attachment networks, the conclusion is contrary to that for the ER network. This is because of the different degree distribution for both networks. With larger diameter, the larger graph has less information containing its degree distribution. This means, the random walker for the larger graph will use more steps to reach the stable state. Also, after reaching the relatively stable area, the deviation is also larger than the smaller graphs.

3. PageRank

(a) In this part, a directed random network using the preferential attachment model is created with the below parameters: node number = 1000, directed = True, m = 4. Then we need to measure the probability of the walker visiting each node and compare the probability with the degree of nodes. In this question, we use steps = 500 and looping time = 1000 to ensure the walker reaches the ending nodes. Both plots of probability are show below:

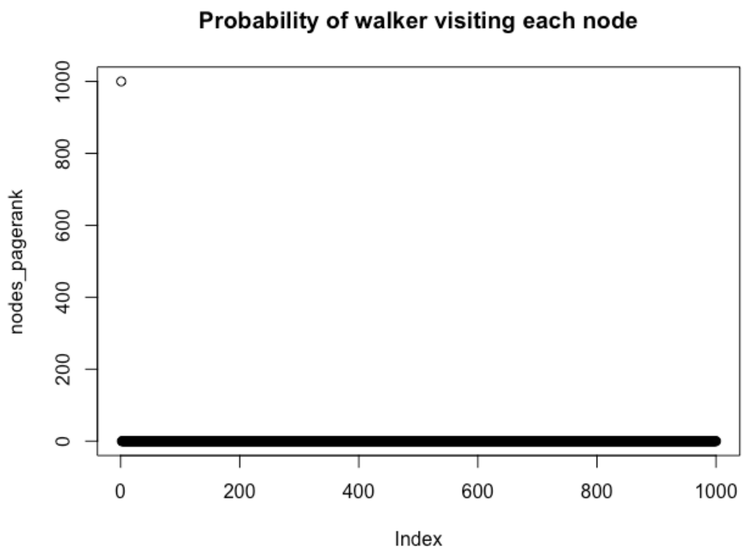


Fig 17. Probability of the walker visiting the nodes

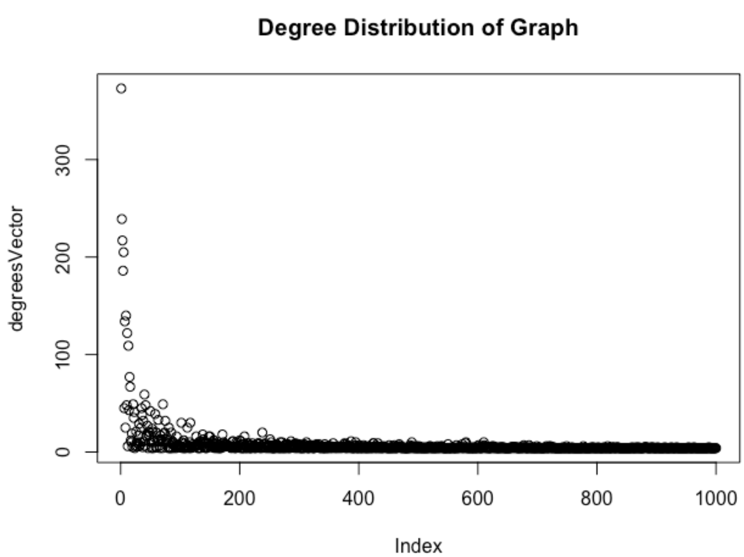


Fig 18. Degree distribution of the network

We can find from the plots that the probability distribution for the walker is: P(first node) = 1, P(rest nodes) = 0. For the whole network, however, the degree distribution follows the power law distribution. The explanation goes below. Every node in this network has out degree, except the first node. Also, the first node has the highest in degree. So, it is very likely for the random walker to walk into the first node, and then stay at the first node since it has no out degree. Therefore, after a large number of steps, the random walker final reaches its steady state, which is 100% sure to arrive at the first node. That is, the probability for the first node is 1, and the probability for the rest of the nodes is 0. So, the probability distribution for the walker is independent of the degree of the nodes in the network.

(b) This question involves teleportation. After adding the teleportation probability of 0.15, we again plot the probability of the walker visiting each node and compare the result with the degree distribution of network (Fig 18) again. The plot is shown below.

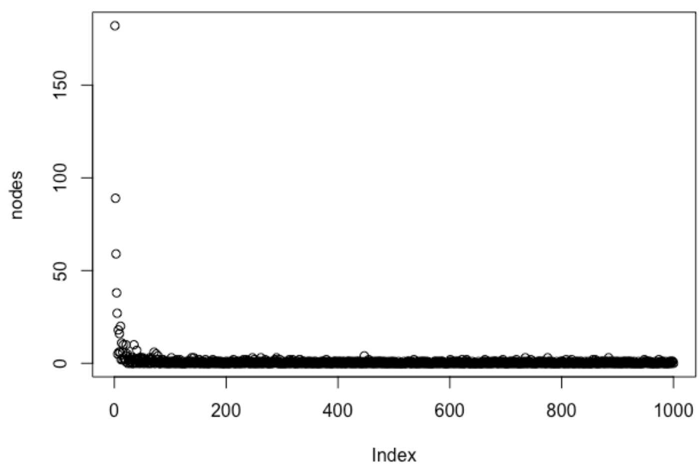


Fig 19. Probability of the walker visiting the nodes with teleportation

From Fig 18 and Fig 19, the probability distribution for the walker visiting each node is very much similar to the degree distribution for the nodes in the network. Therefore, the probability with teleportation technique is related to the degree of the node.

4. Personalized PageRank

(a) In order to define each user's own notion of importance, we don't use 1/N as the chance of visiting all the other nodes at teleportation. Instead, we let the chance to be proportional to the PageRank. The teleportation probability is still 0.15. The probability plot is shown below.

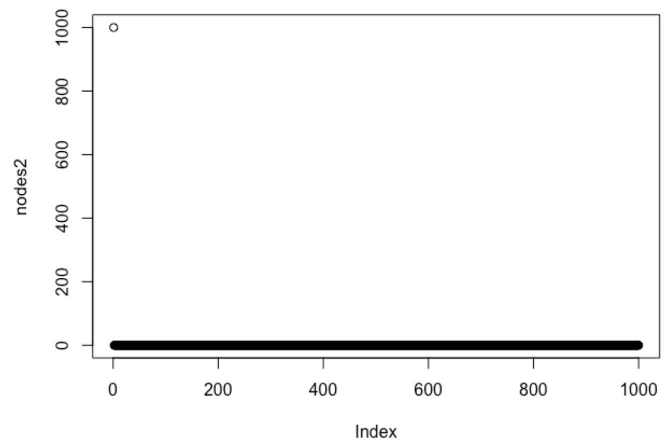


Fig 20. Probability of the walker visiting the nodes with teleportation probability proportional to PageRank

We can find from the plots that the probability distribution for the walker is: P(first node) = 1, P(rest nodes) = 0. This is because, the teleportation probability depends on the PageRank values, which follows the power law distribution. This means, no matter a node chooses to use the teleportation probability or move to its adjacent node, it has a very high probability of moving to the first node, according to the definition of power law distribution. So, the probability doesn't follow the results in 3(b).

(b) In this problem, we first need to pick out two nodes, which are defined as the median PageRanks. Then, change the probability function at teleportation to be ½ for visiting each of the two nodes, and 0 for visiting all of the rest nodes. The probability of the walker is also plot here using this teleportation probability.

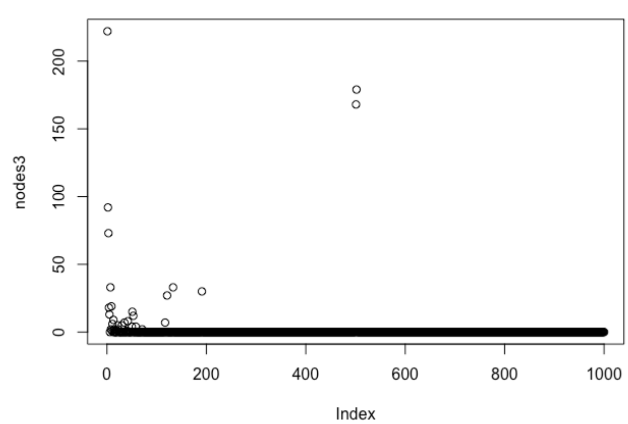


Fig 21. Probability of the walker visiting the nodes with teleportation probability related to nodes with median PageRanks

The PageRank values, in this case, tend to have high values at the median PageRanks. This is because that the probability for any node to link to the median PageRanks are high due to the teleportation method. So, generally the median PageRanks will have much higher values to indicate the relatively higher probabilities of coming into those two nodes.

(c) In this part, we combine both 4(a) and 4(b) and change the teleportation probability to cover both the PageRanks values and the influence from the trusted web pages. Therefore, we adjust the probability for the nodes with median PageRanks as ½*beta, ½*beta, while the rest of nodes follow the normal PageRanks distribution, with the weight of (1-beta).

In this way, the probability vector is changed as (1-beta) * PageRank + beta * (1/2 for nodes with median PageRanks). We defined the value beta = 0.2 in our design, and the corresponding probability plot is shown below. From the graph, we can see that the results now have the influence from both the PageRank values and the median PageRanks.

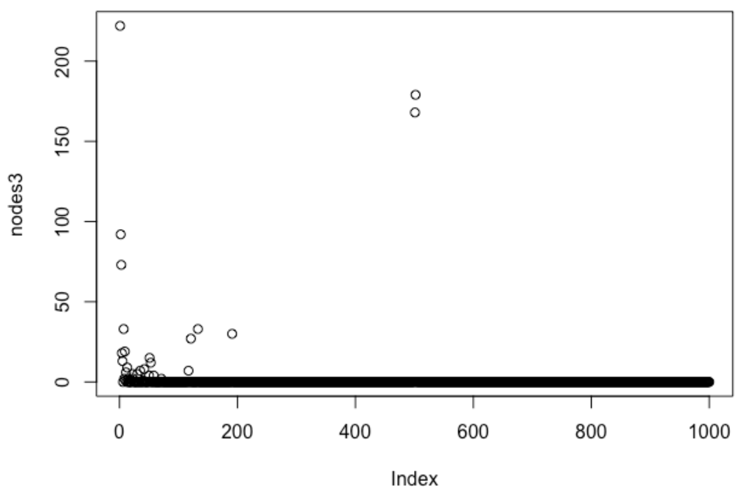


Fig 22. Probability of the walker visiting the nodes with teleportation probability related to both PageRank values and nodes with median PageRanks