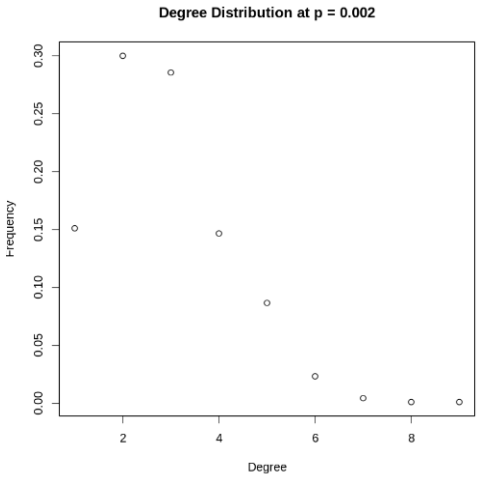
Report for Project 1

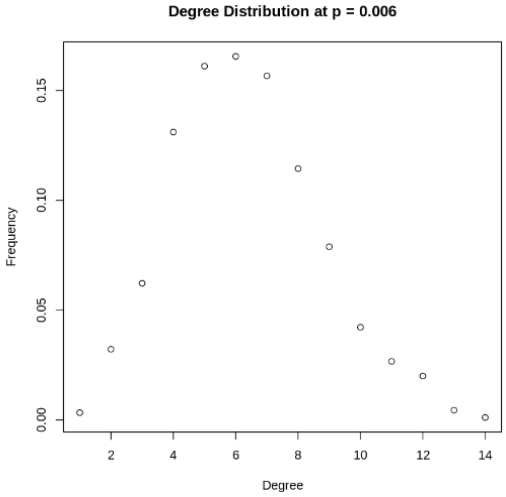
1.Generating Random Networks

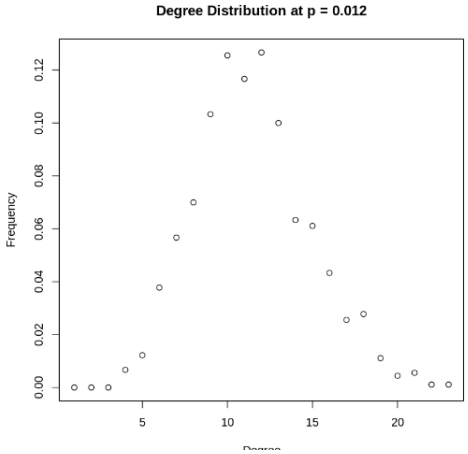
1.

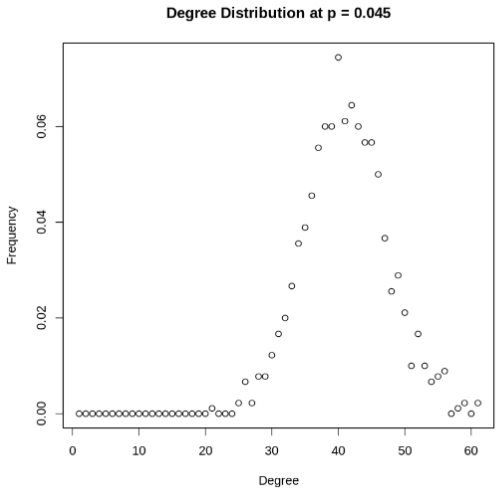
(a)

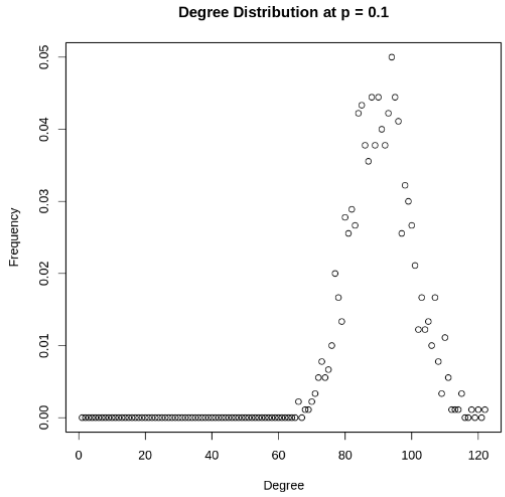
We create 5 undirected random networks using the Erdos-Renyi model with n = 900 nodes, the probability p for drawing an edge between two arbitrary vertices 0.002, 0.006, 0.012, 0.045 and 0.1 and get the following degree distributions.











From the graph, we observe a binomial distribution for the degree distributions. The reason is that for the nodes in the Erdos-Renyi model, they are independently connected with probability p. Therefore, the probability that a node has degree k is $P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$. Thus, the degree distributions are binomial distributions.

Probability	Mean	Variance	Theoretical Mean	Theoretical Variance
0.002	1.84	1.796396	1.798	1.794404
0.006	5.411111	5.216784	5.394	5.361636
0.012	11.033333	11.084538	10.788	10.658544
0.045	40.42	38.237197	40.455	38.634525
0.1	89.713333	80.740868	89.9	80.91

The table above shows the means and variance of the degree distributions and their theoretical values where mean is np and variance is $np(1 - p)$. We can find that the means and variance of the degree distributions are close to their theoretical value because n is big enough.

(b)

Not all random realizations of the ER network are connected.

```
## 0.002 0.006 0.012 0.045   0.1
## FALSE FALSE  TRUE  TRUE  TRUE
```

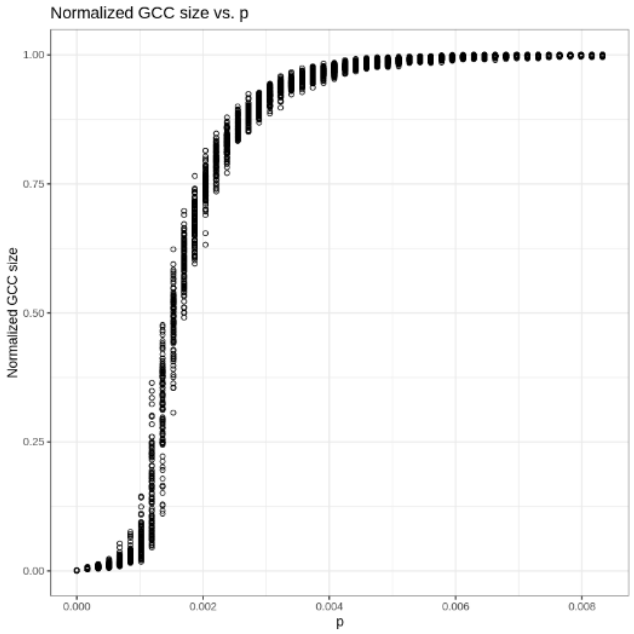
As the result shows above, we know that for p = 0.002 and 0.006, the ER network is not connected.

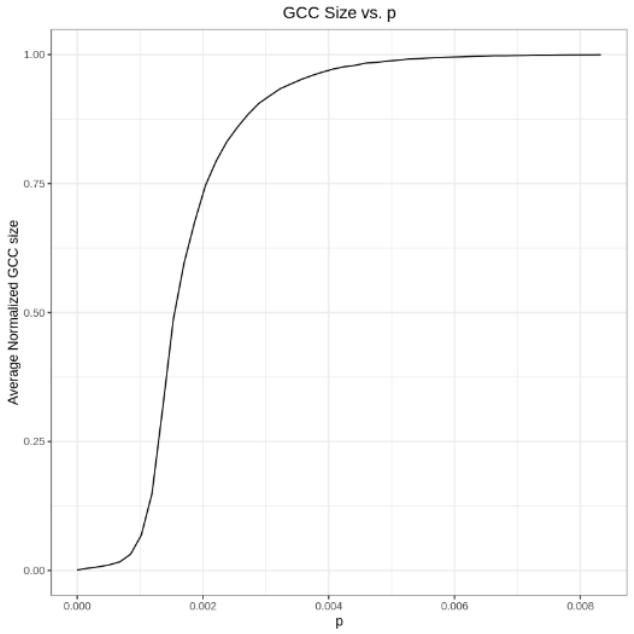
And we estimate the probability that a generated network is connected for the given probabilities by creating the network 1000 times and calculate the probability. And the estimated probabilities that the network is connected for each p is:

p	probability
0.002	0
0.006	0.024
0.012	0.989
0.045	1
0.1	1

For p=0.002, the giant connected component is 646, and the diameter of the GCC is 22.
For p=0.006, the GCC size is 896 and the diameter of the GCC is 8.

(c)


Normalized GCC size vs. p
Normalized GCC size
1.00
0.75
0.50
0.25
0.00
0.000
0.002
0.004
0.006
0.008
p


GCC Size vs. p
Average Normalized GCC size
1.00
0.75
0.50
0.25
0.00
0.000
0.002
0.004
0.006
0.008
p

i.

The criterion of “emerge” in this case means that the GCC size starts to increase. The GCC starts to emerge at p = 0.001, which is close to the theoretical value, which is $\frac{1}{n}$ = 0.0011

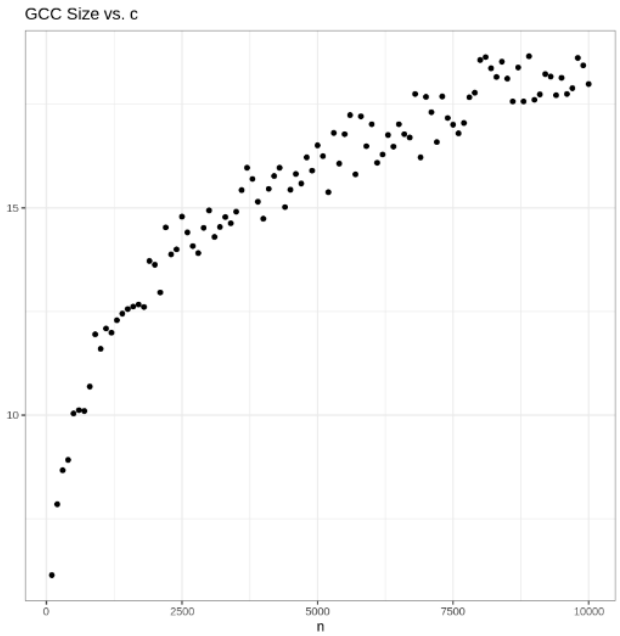
ii.

From the graph, at p = 0.007, the GCC takes up over 99% of the nodes in almost every experiment. Which is close to the theoretical value $\frac{ln(n)}{n}$ = 0.0075.

(d)

i.

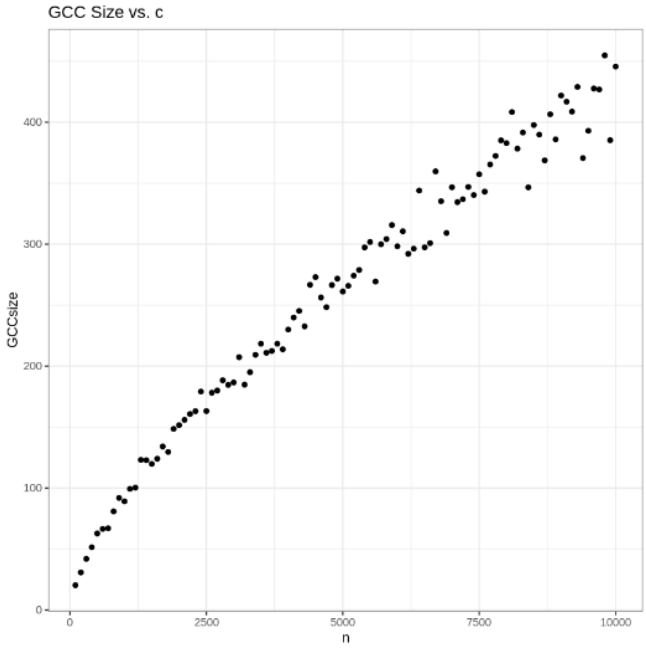
For c = np = 0.5, the expected size of the GCC of ER networks is as follows:



We can observe that as the number of the nodes n increases, the GCC size tends to increase. The trend slows down at some point near n = 2500 nodes.

ii.

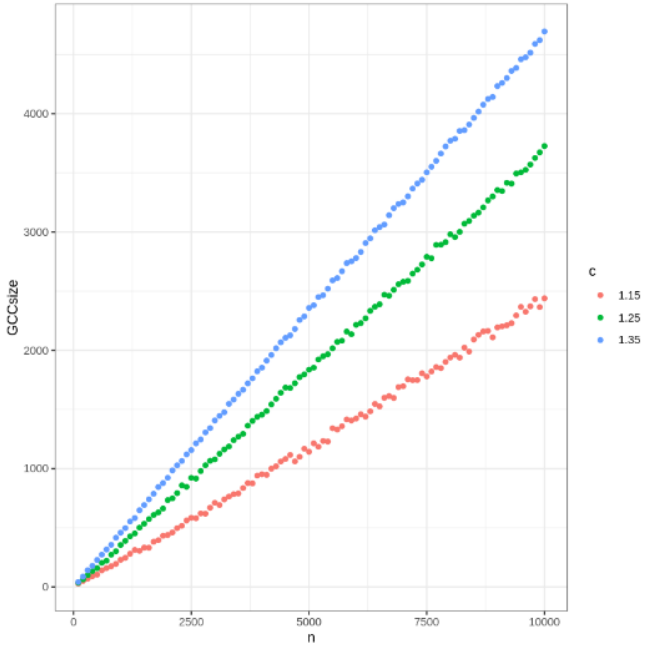
For c = np = 1, the expected size of the GCC of ER networks is:



The trend remains the same. As n increases, the GCC size is also increasing. In addition, we observe that the trend becomes more linear. Compared to the figure for c=0.5, the GCC size becomes larger because the probability of the edges connect to each other increases.

iii.

For c = np = 1.15, 1.25 and 1.35, the expected sizes of the GCC of ER networks are:



And the observation remains the same as (a) and (b).

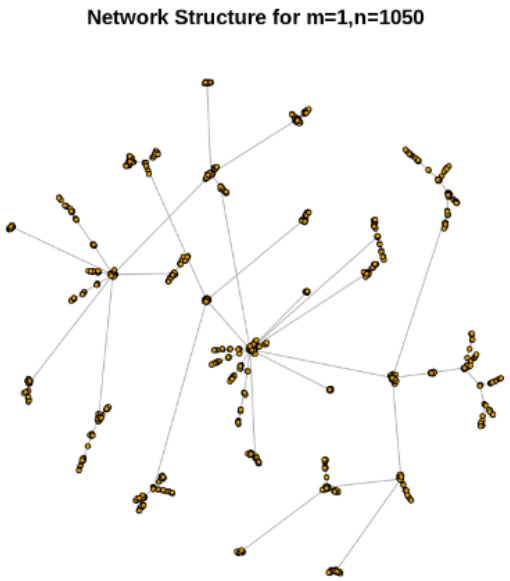
iv.

The expected GCC size has a positive linear relationship with the number of nodes n in all cases since the number of nodes increases, the probability that each edge connects to each other will be larger . From each figure, we can find that the slope gets larger as c increases and therefore increases the GCC size. This is because as c increases, the probability that the nodes connect to others increases, thus the GCC size increases.

2.

(a)

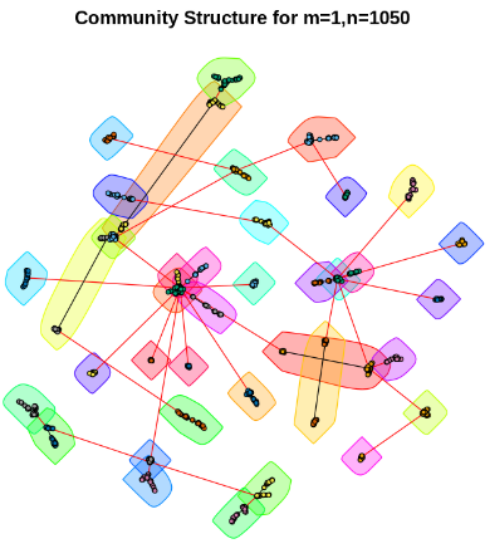
The undirected network using preferential attachment model with m=1, n=1050 is:



The network is always connected because each new node connects to an older one, therefore, we can always find a path between any pair of nodes.

(b)

The community structure using fast greedy method is:

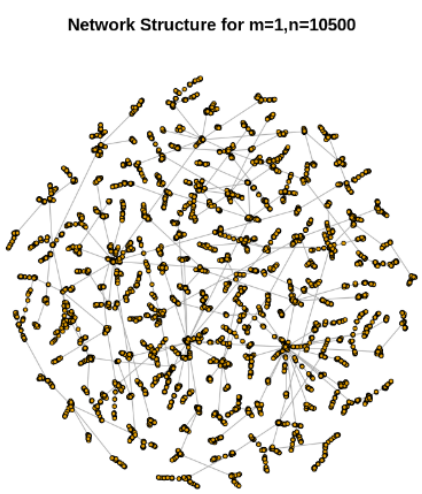


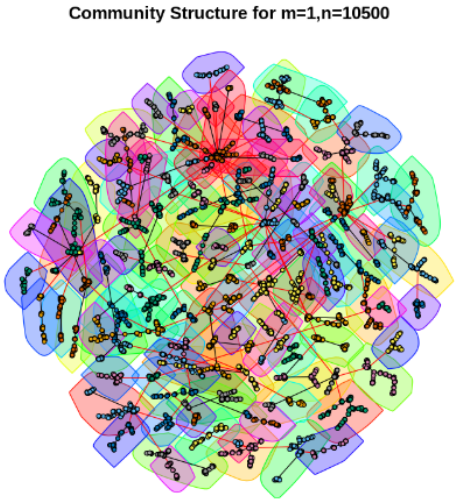
The modularity is 0.93272952314656.

Assortativity is the coefficient that measures the tendency of the vertices that have similar properties to connect to each other. In this case, we use the property degree to calculate the assortativity. The assortativity of the network is -0.0732404296651051, which means that the vertices that have similar properties tend not to connect with each other.

(c)

The undirected network structure and community structure with m=1, n=10500 is:





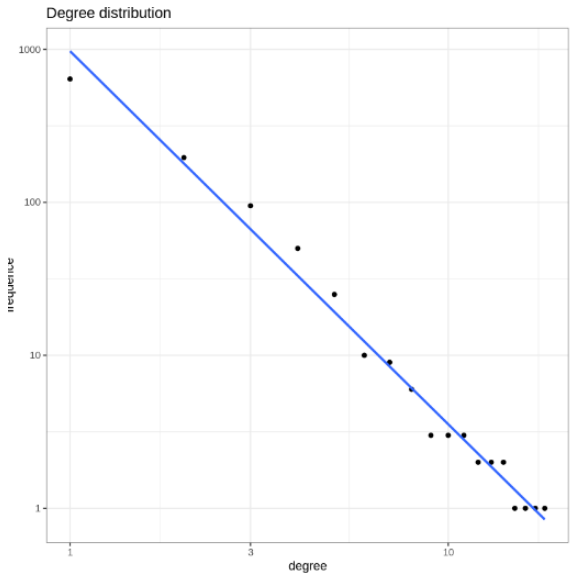
The modularity is 0.978818201391497.

The assortativity is -0.026807513523105.

Compared to the smaller network's modularity, the modularity increases. This is because as n increases, the number of communities increases.

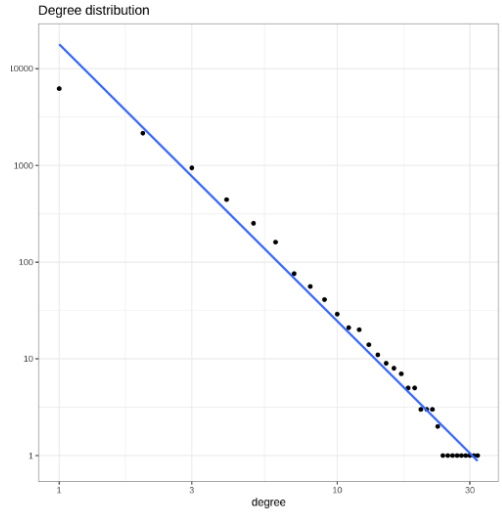
(d)

The degree distribution in a log-log scale for n=1050 is:



The slope is -2.43852784412263.

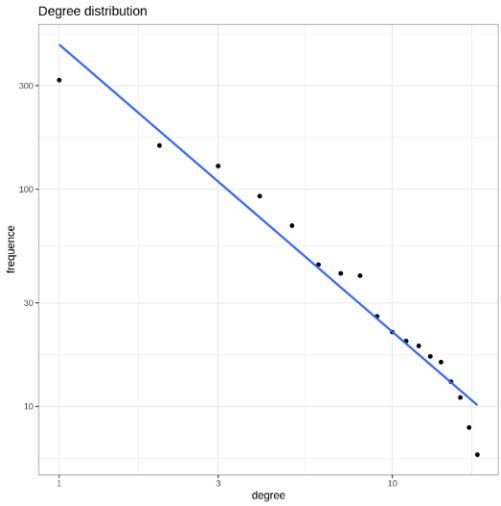
The degree distribution in a log-log scale for n = 10500 is:



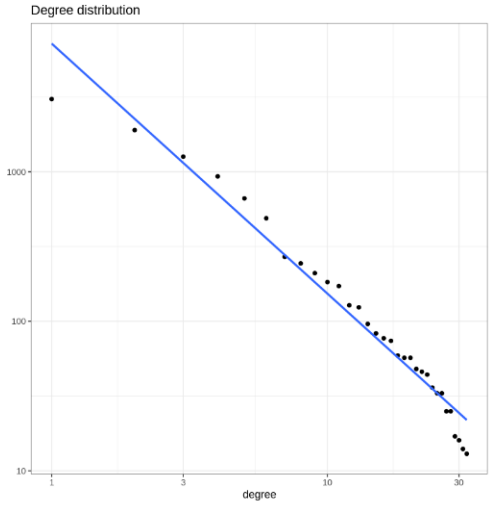
The slope is -2.86217969802433.

(e)

The degree distribution of nodes j that are picked with this process for n = 1050 in log-log scale is:



The degree distribution of nodes j that are picked with this process for n = 10500 in log-log scale is:

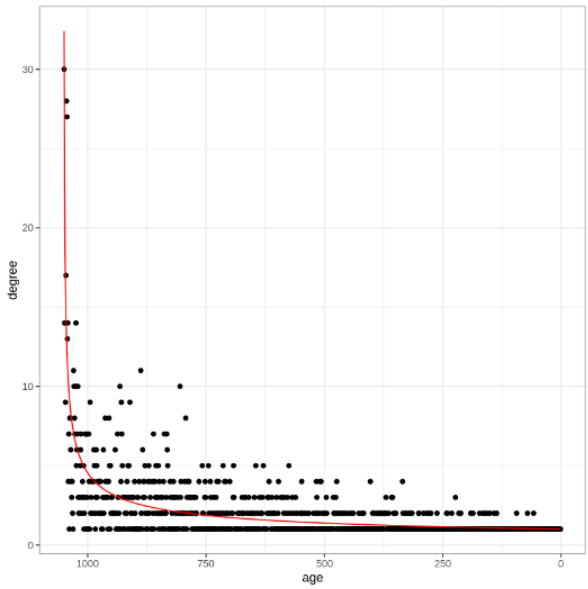


From the graphs, we can observe that the distribution is linear in a log-log scale. The slope of degree distribution for n = 1050 is -1.32255039889333. And the slope of degree distribution for n = 10500 is -1.67257229988512.

Compared to the node distribution, the slope is much smaller. It is because for the node with high degree, the probability that the neighbor also has high degree. Thus, there are more nodes in high degree areas and thus the slope becomes smaller.

(f)

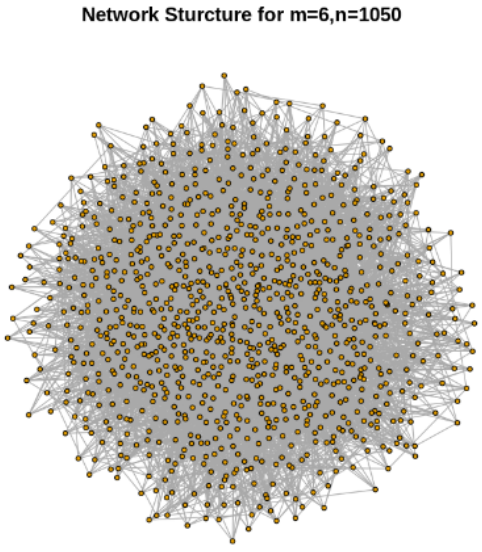
The plot for relationship between the node age and their expected degree is:



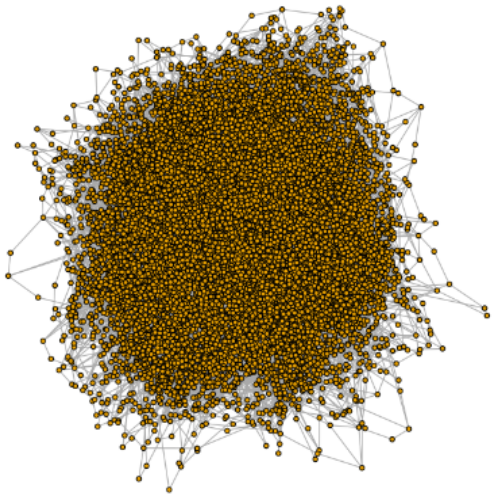
We can find that for the expected degree of nodes increases as the age increases, and the distribution follows the theoretical value: $k(i;t) = m(\frac{t}{i})^{\frac{1}{2}}$.

(g)

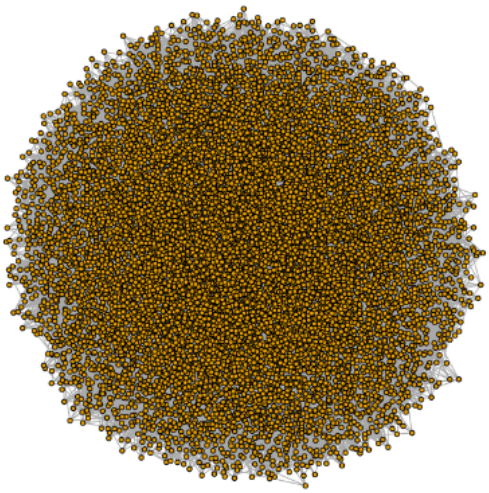
The undirected network structure and community structure for m=2, 6, n=1050, 10500 is:



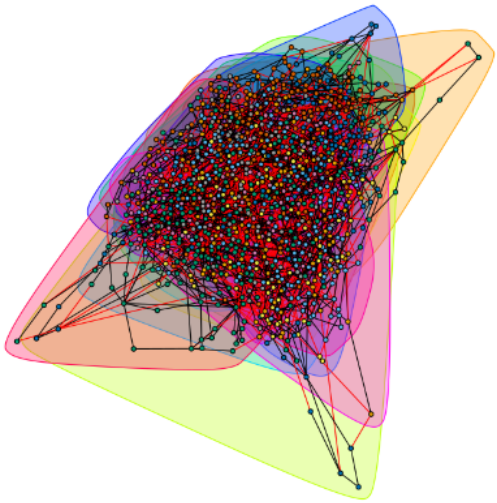
Network Structure for m=2,n=10500



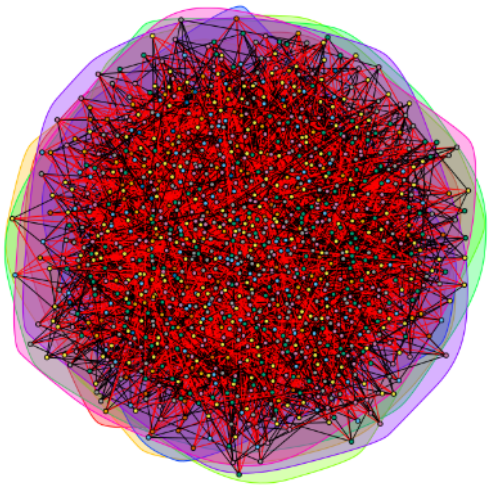
Network Structure for m=6,n=10500

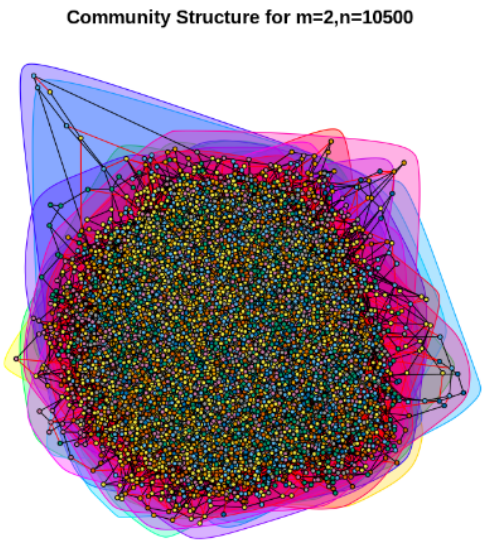


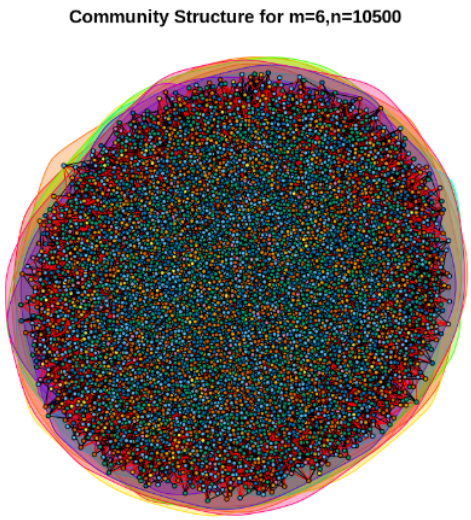
Commnunity Structure for m=2,n=1050



Community Structure for m=6,n=1050







Since m = 2, 5 > 1, all the networks are connected.

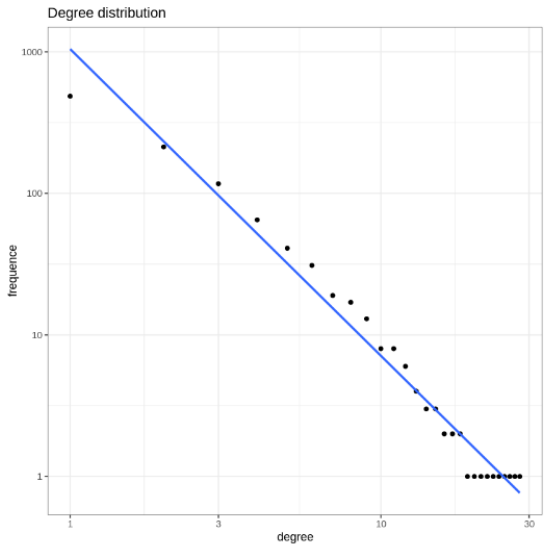
The modularities and assortativities for each network are shown in the table below.

Modularity	m=2	m=6
n=1050	0.5225247413	0.2412317596
n=10500	0.5312215808	0.2513717967

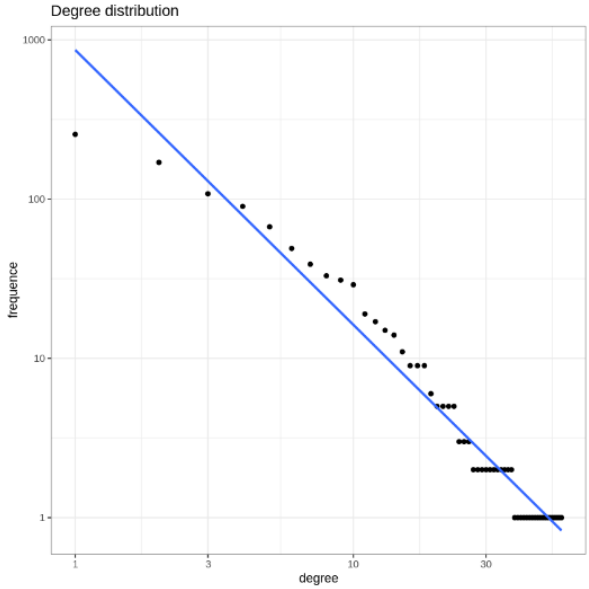
Assortativity	m=2	m=6
n=1050	-0.04379573344	-0.00861221245
n=10500	-0.01076866439	-0.002247741418

As we can see from the table, the modularity and assortativity decreases as m increases. Since m increases, the edges for a node increases, therefore reducing the density of connections in the community, thus modularity decreases.

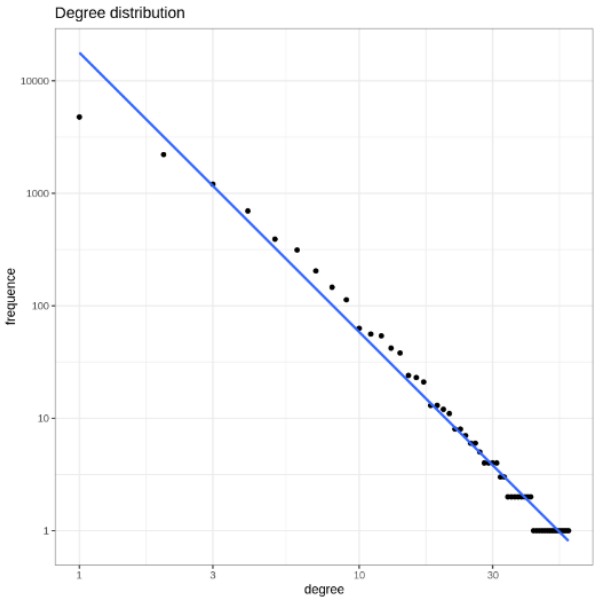
The degree distribution in a log-log scale for m=2, n=1050 is:



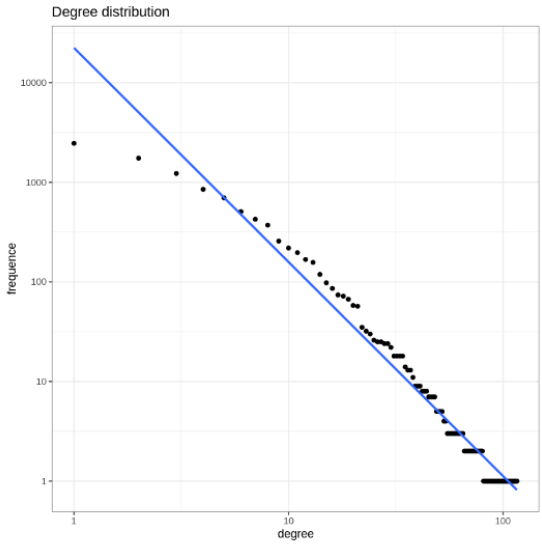
The degree distribution in a log-log scale for m=6, n=1050 is:



The degree distribution in a log-log scale for m=2, n=10500 is:



The degree distribution in a log-log scale for m=6, n=10500 is:

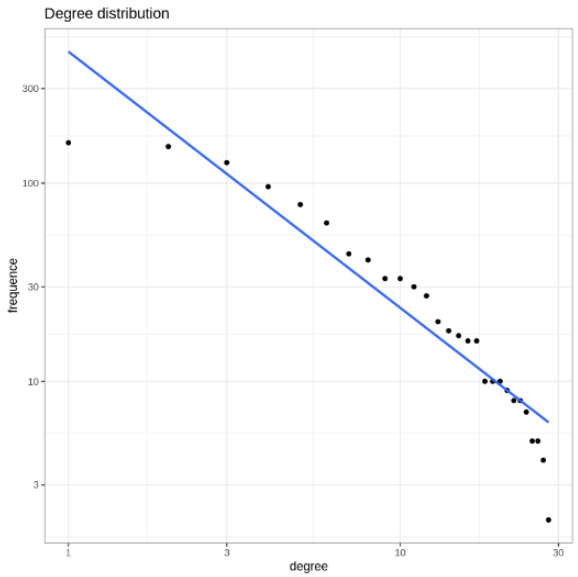


The slope of the plot using linear regression is shown in the following table:

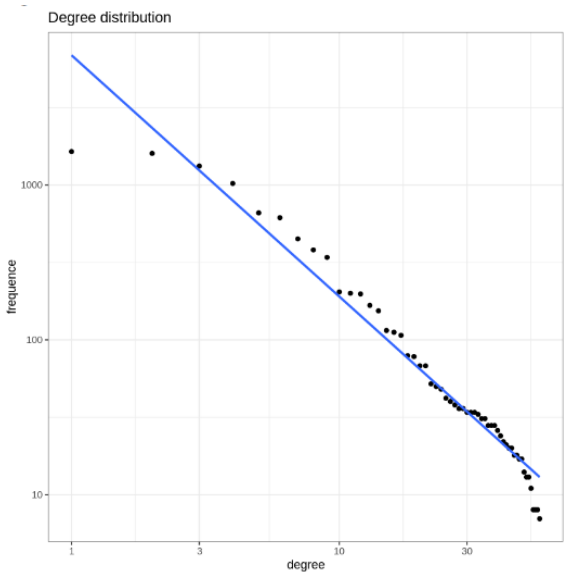
Slope	m=2	m=6
n=1050	-2.166540492	-1.725703002
n=10500	-2.483224576	-2.150132309

As m increases, the slope for the plot increases since the degrees for each node increases.

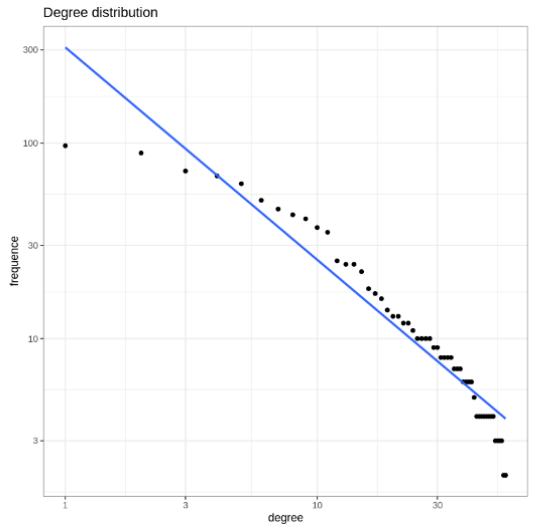
The degree distribution of nodes j that are picked with this process, in the log-log scale for m=2, n=1050 is:



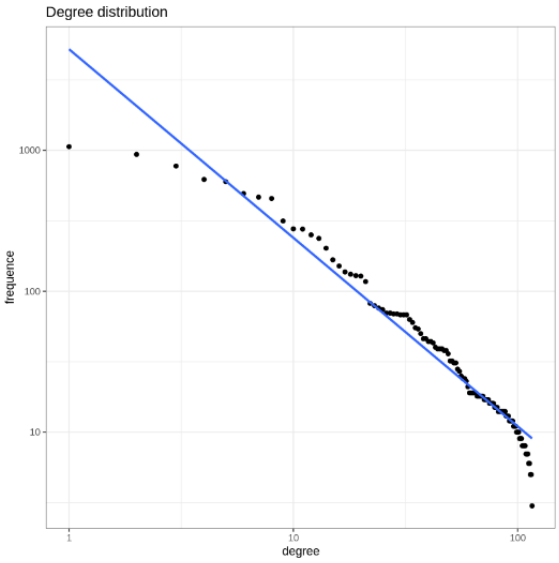
The degree distribution of nodes j that are picked with this process, in the log-log scale for m=2, n=10500 is:



The degree distribution of nodes j that are picked with this process, in the log-log scale for m=6, n= 1050 is:



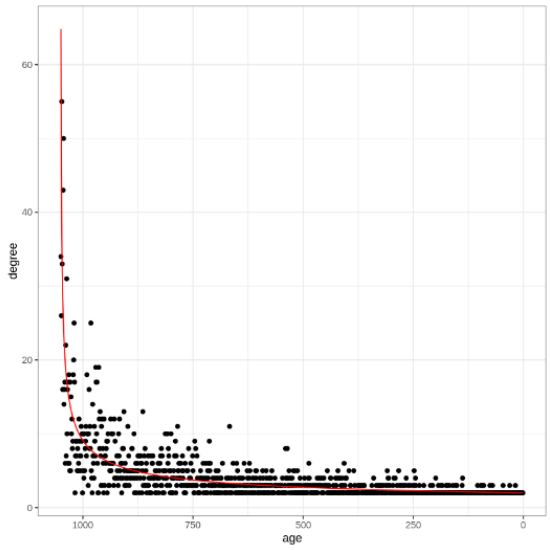
The degree distribution of nodes j that are picked with this process, in the log-log scale for m=6, n= 1050 is:



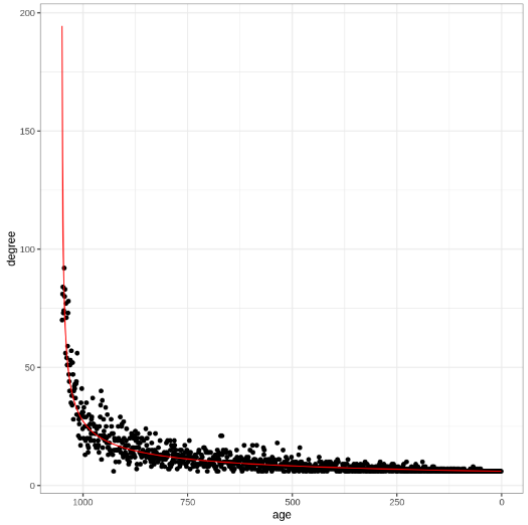
From the figures, we can observe that the relationship in log-log scale is linear.
The slope for m=2, n=1050 is -1.29248661020041.
The slope for m=2, n=10500 is -1.55815237648658.
The slope for m=6, n=1050 is -1.0865108507428.
The slope for m=6, n=10500 is -1.33776257449934.

The slope also increases as m increases. the same reason for the degree distribution.

The plot for expected degree vs. age for m=2 is:



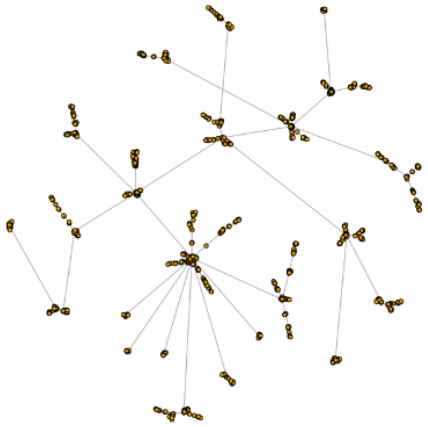
The plot for expected degree vs. age for m=6 is:



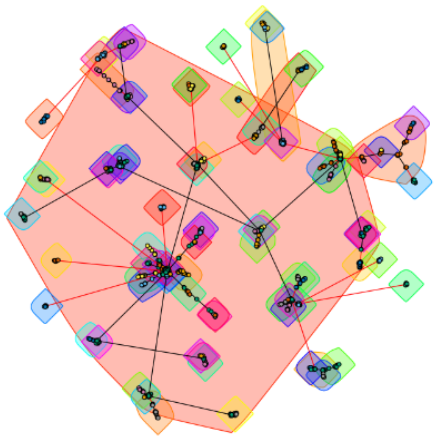
From the graphs, we can see that as m increases, the expected degree of each node attaches more to the theoretical curve.

(h)

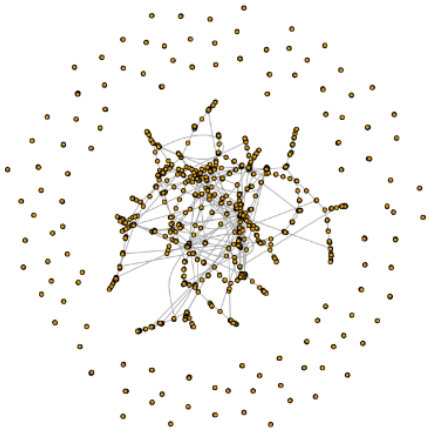
Network Structure using preferential attachment with m=1, n=1050



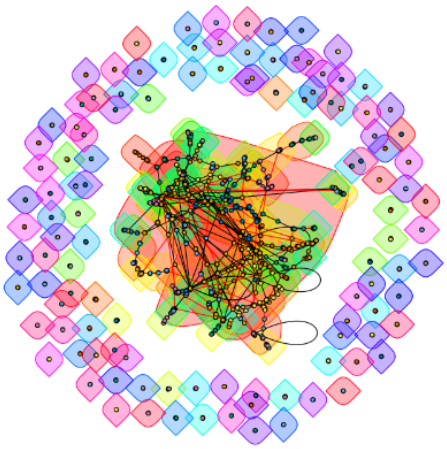
Community Structure using preferential attachment with m=1, n=1050



Network Structure with Same Degree Sequence



Community Structure with Same Degree Sequence

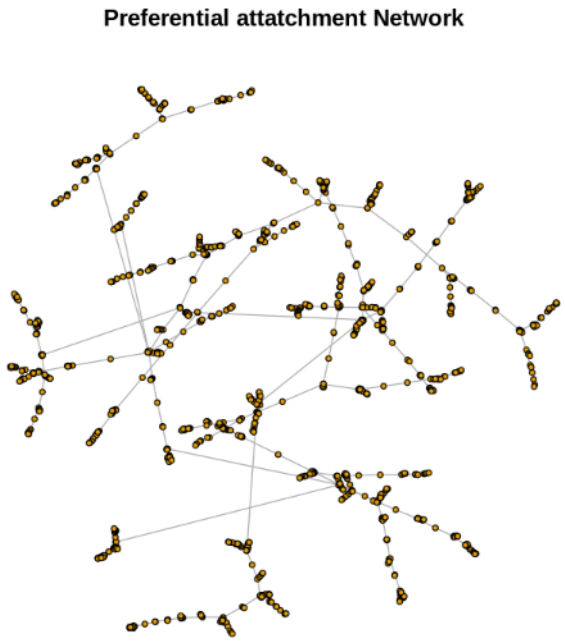


The modularity for preferential attachment is 0.840753507130582.
The modularity for the network with the same degree sequence is 0.758659343275769.

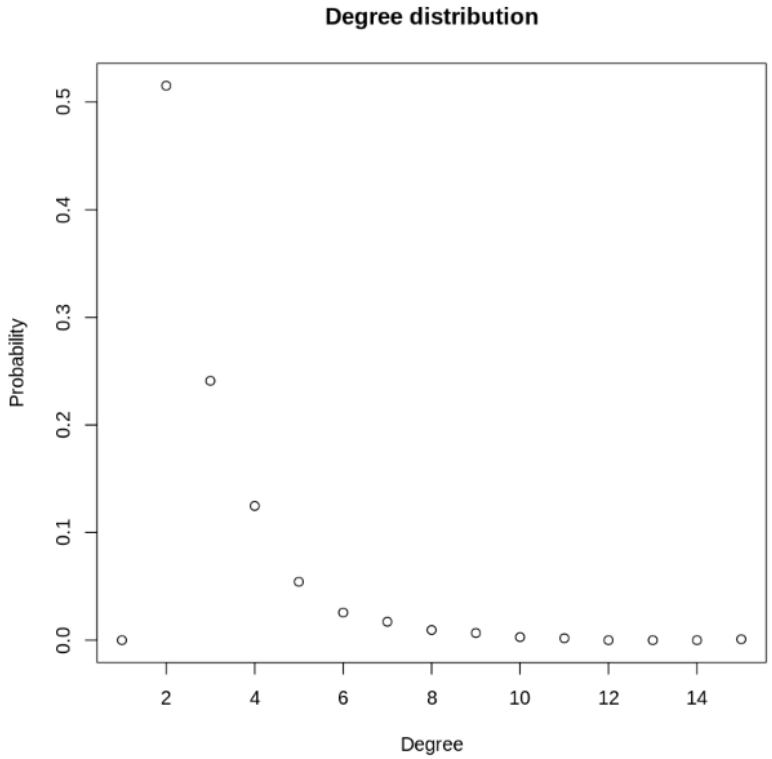
We can observe that the modularity of the network created by preferential attachment is much higher than the network created with the same degree sequence. It is because the stub-matching procedure is random. The network is not connected and therefore reduces the connections in the communities.

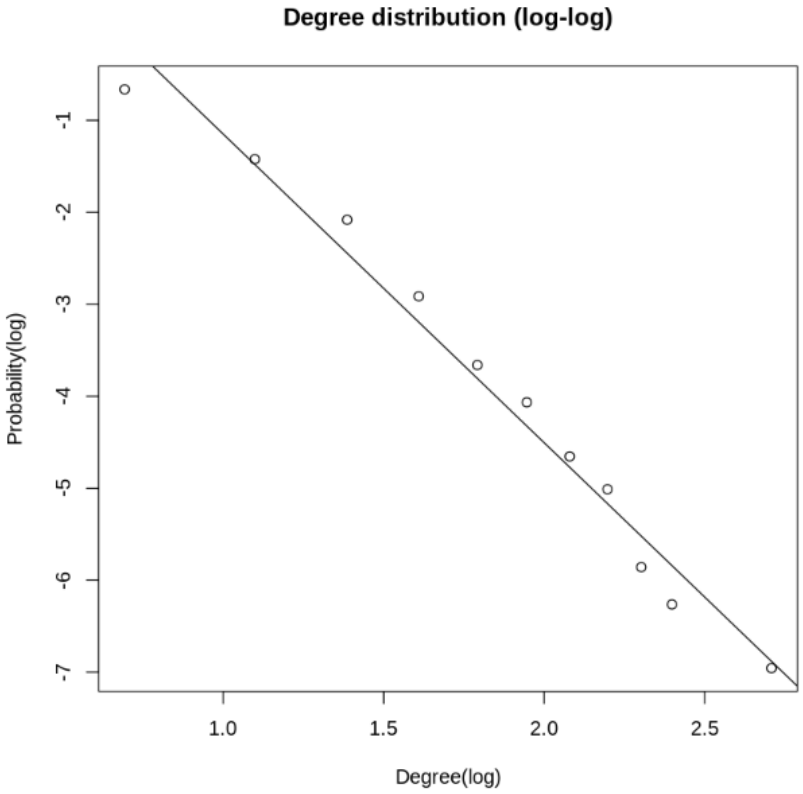
3. Create a modified preferential attachment model that penalizes the age of a node

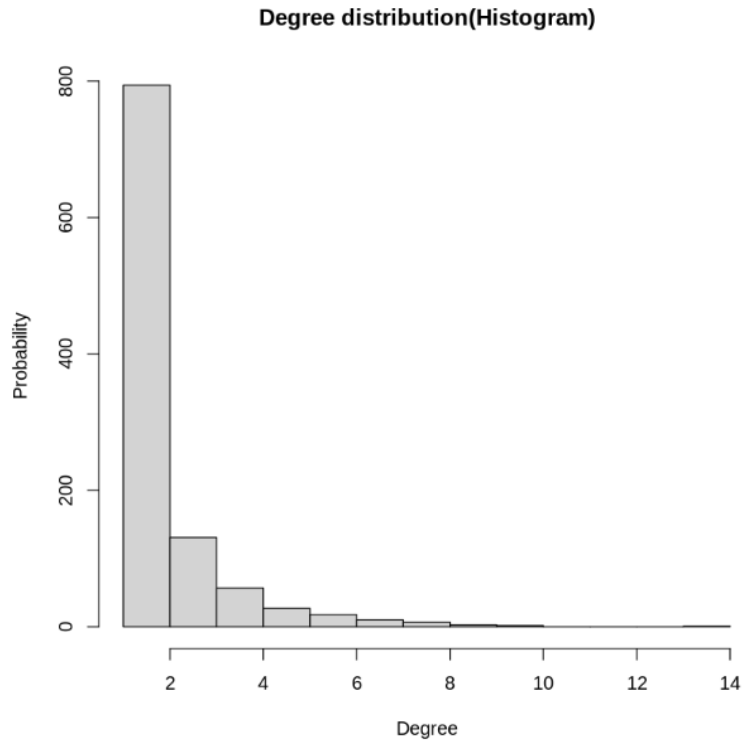
(a) The graph shown below is the undirected network with 1050 nodes and parameters m = 1, α = 1, β = −1, and a = c = d = 1, b = 0.



The graphs shown below are the degree distribution plots (in different forms, specified below)




Degree distribution (log-log)
Probability(log)
-1
-2
-3
-4
-5
-6
-7
1.0
1.5
2.0
2.5
Degree(log)


Degree distribution(Histogram)
Probability
0
200
400
600
800
2
4
6
8
10
12
14
Degree

Power law exponent:

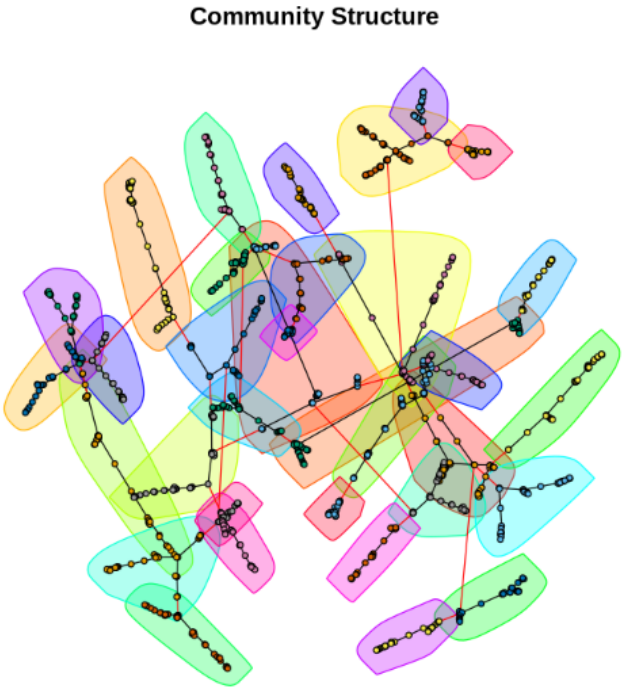
```
print(lm(y~x))
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      2.208       -3.356
```

The power law exponent is -3.356

(b)
The community structure plot is shown below:

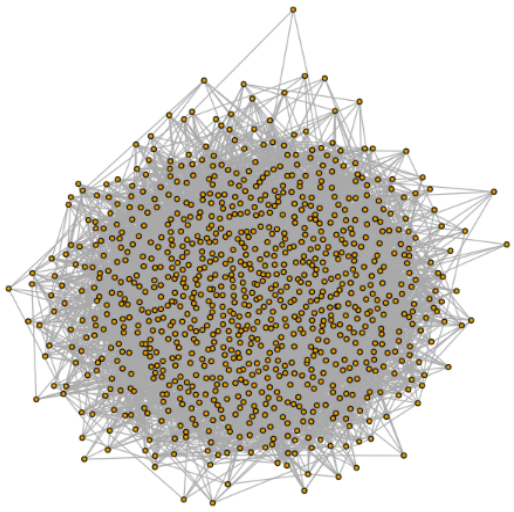



```
Modularity is  0.936709435923814
```

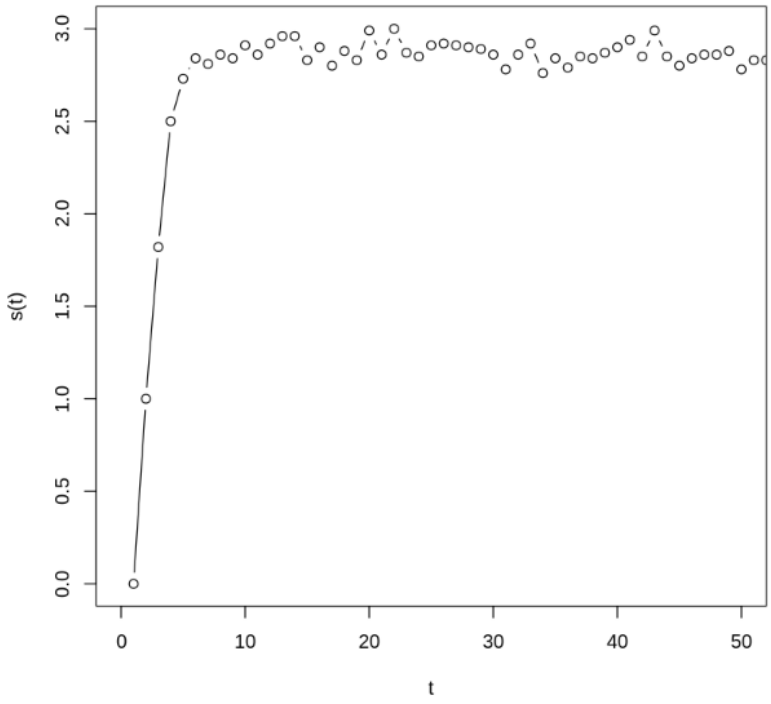
2. Random Walk on Networks

1. Random walk on Erd¨os-R´enyi networks

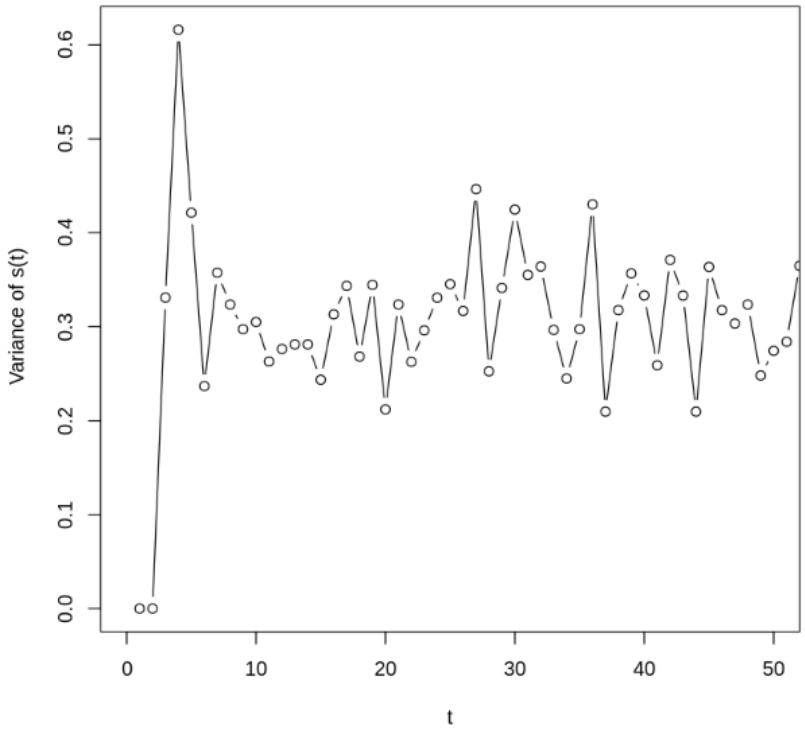
(a) The graph below is an undirected random network with 900 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.015



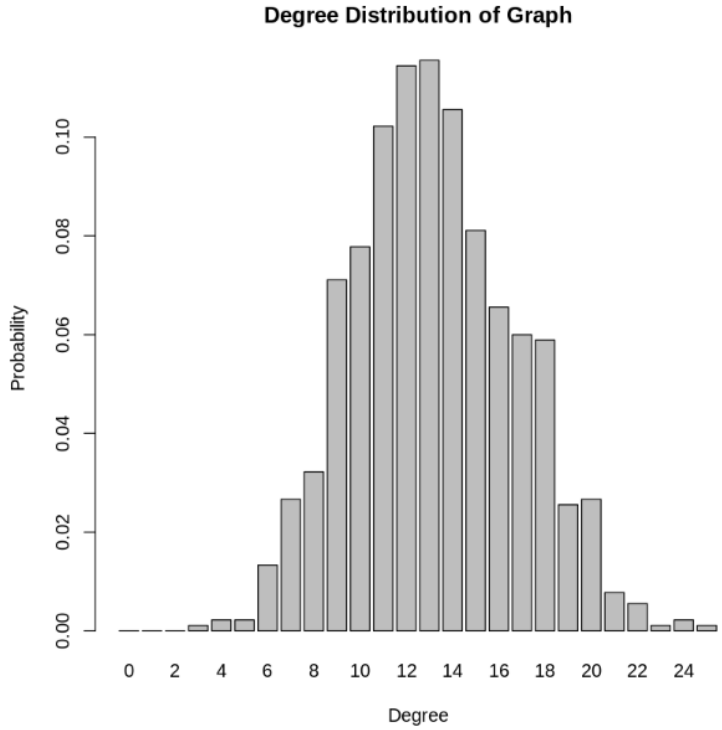
(b) Plot of ⟨s(t)⟩ v.s. t for 900 nodes

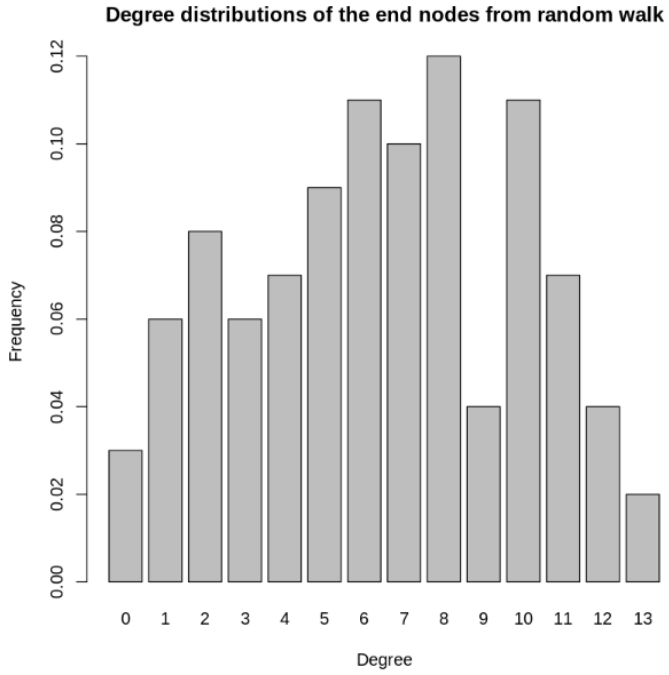


Plot of σ^2 (t) v.s. t. for 900 nodes


Variance of s(t)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
50
t

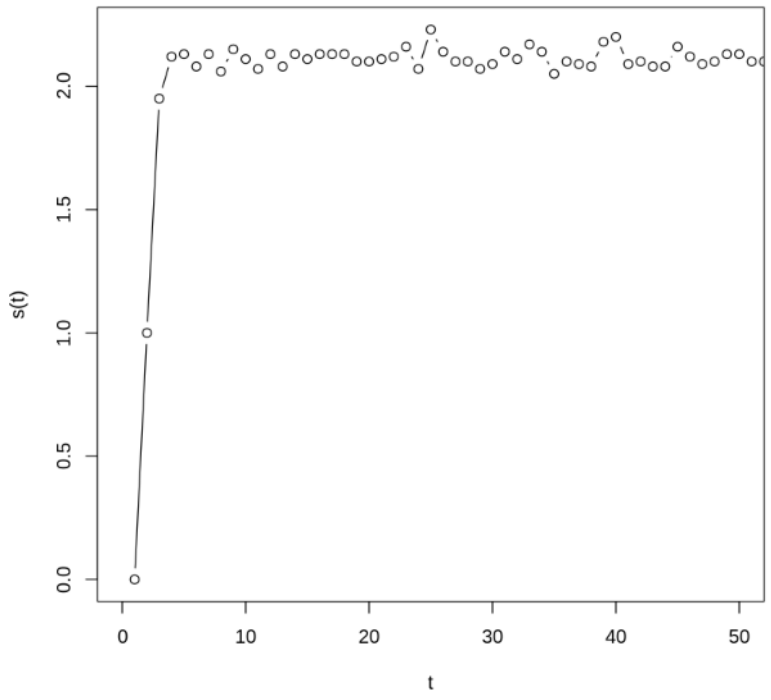
(c)


Degree Distribution of Graph
Probability
0.00
0.02
0.04
0.06
0.08
0.10
0
2
4
6
8
10
12
14
16
18
20
22
24
Degree

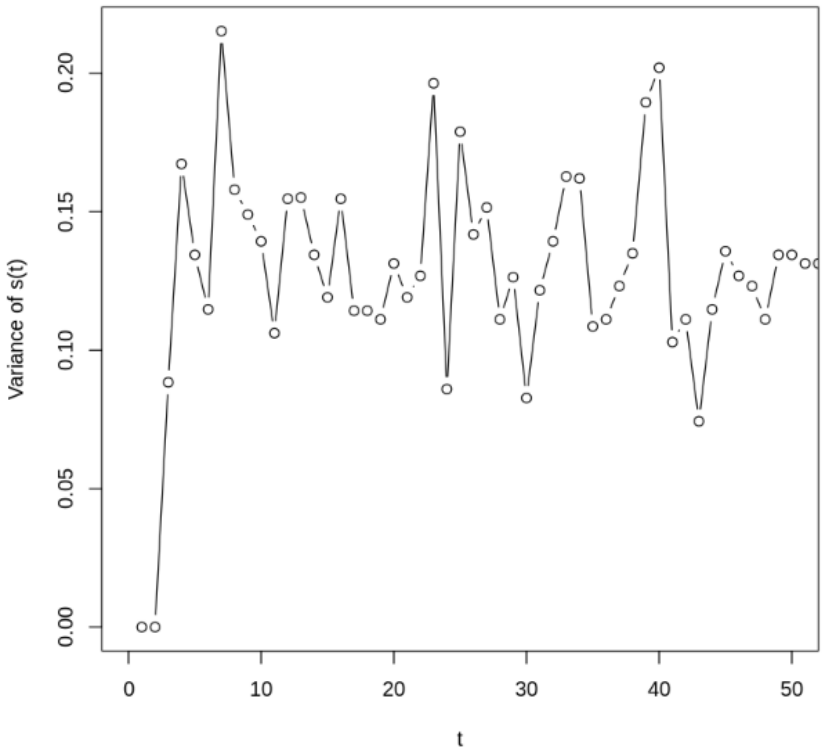


The two graphs have different peaks but they look similar in shapes. (kind of like normal distribution) Therefore we can say that the degree distribution of the nodes reached at the end of the random walk highly relates to the degree distribution of the graph.

(d) Plot of ⟨s(t)⟩ v.s. t for 9000 nodes



Plot of σ^2 (t) v.s. T. for 9000 nodes



The graphs for 900 nodes are above this question.
For the 900 nodes network, the mean is centered between 2.5-3.0, the variance is centered between 0.3-0.4.
For the 9000 nodes network, the mean is centered between 2.0-2.5, the variance is centered between 0.10-0.15.

```
diameter(g)
diameter(g2)
```

```
4
3
```

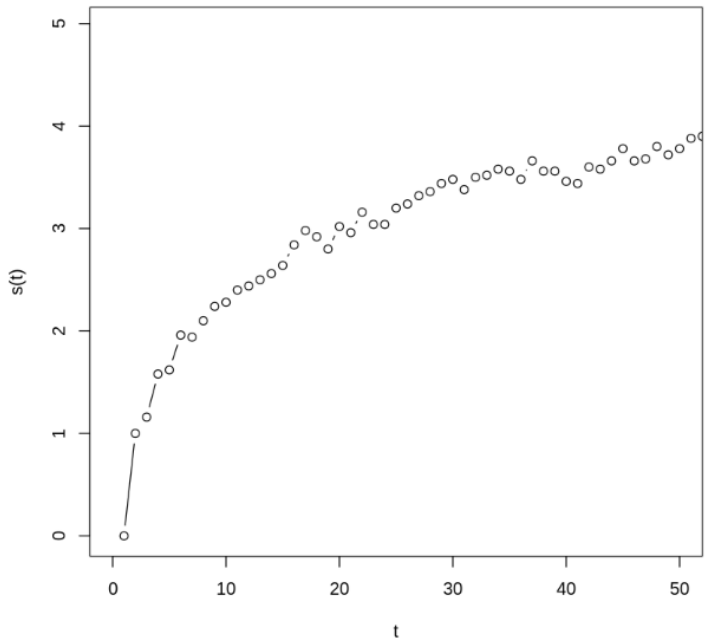
The diameter of the network **does play a role.** The network with smaller diameter has a lower mean distance and lower variance of distance.

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

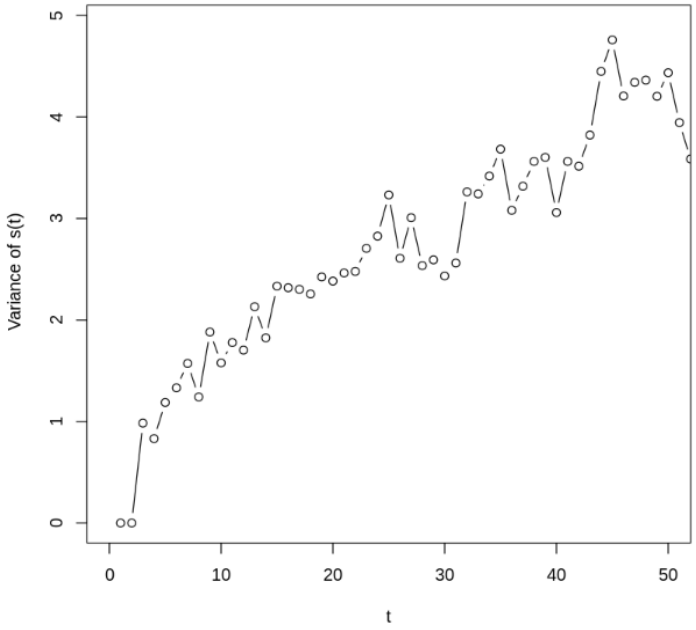
(a) The graph shown below is the undirected preferential attachment network with 900 nodes, where each new node attaches to m = 1 old nodes

(b)

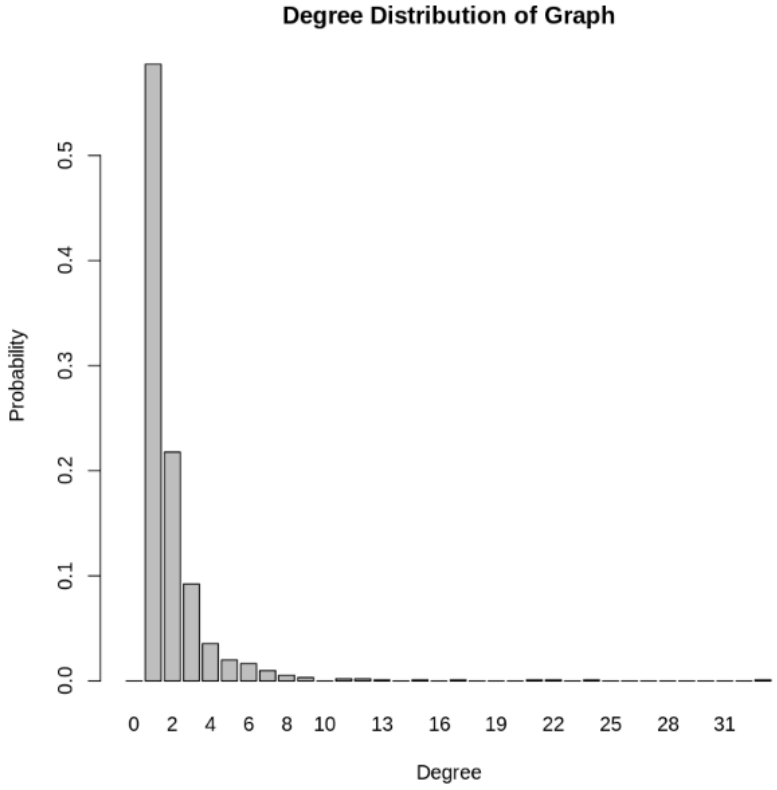
Plot of ⟨s(t)⟩ v.s. t for 900 nodes

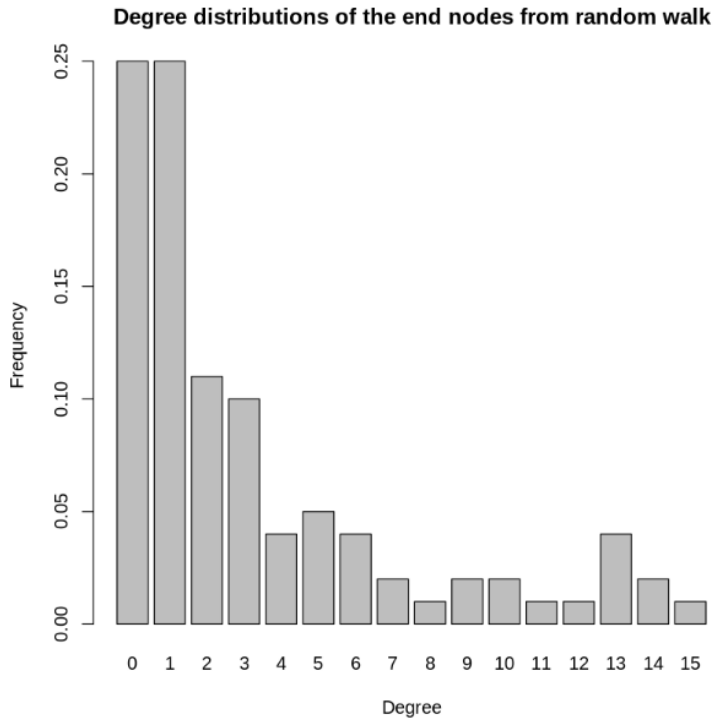


Plot of σ^2 (t) v.s. t. for 900 nodes


Variance of s(t)
0
1
2
3
4
5
0
10
20
30
40
50
t

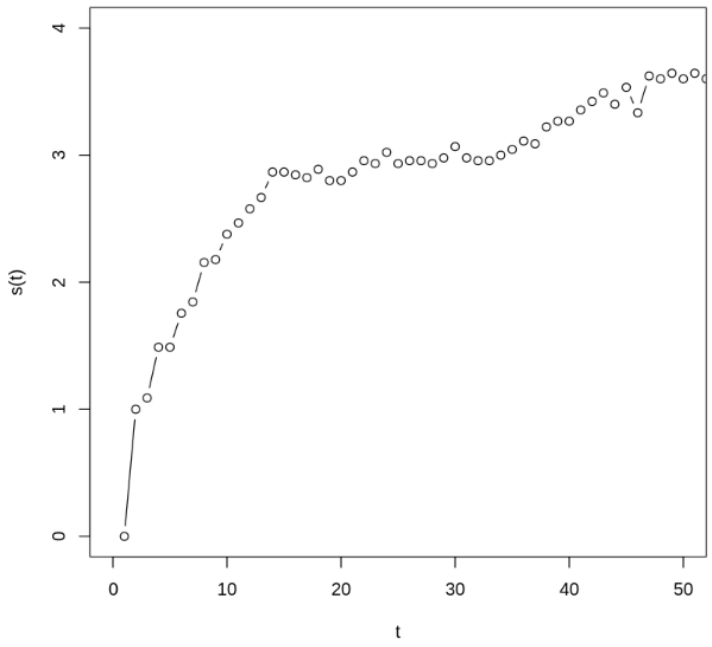
(c)


Degree Distribution of Graph
Probability
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
13
16
19
22
25
28
31
Degree

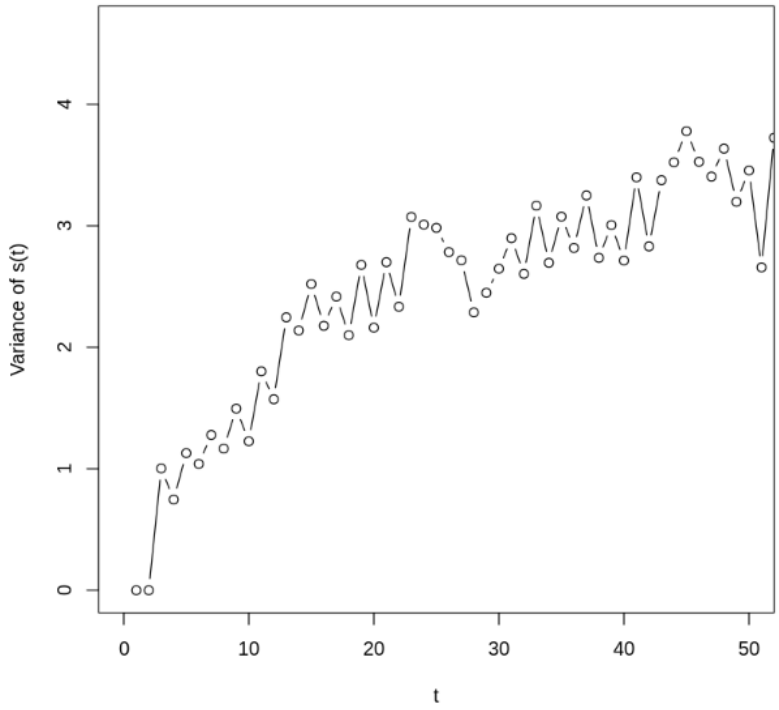


Both graphs look similar in shape (both look like a log normal or right-skewed distribution). Therefore we can say that the degree distribution of the nodes reached at the end of the random walk highly relates to the degree distribution of the original graph.

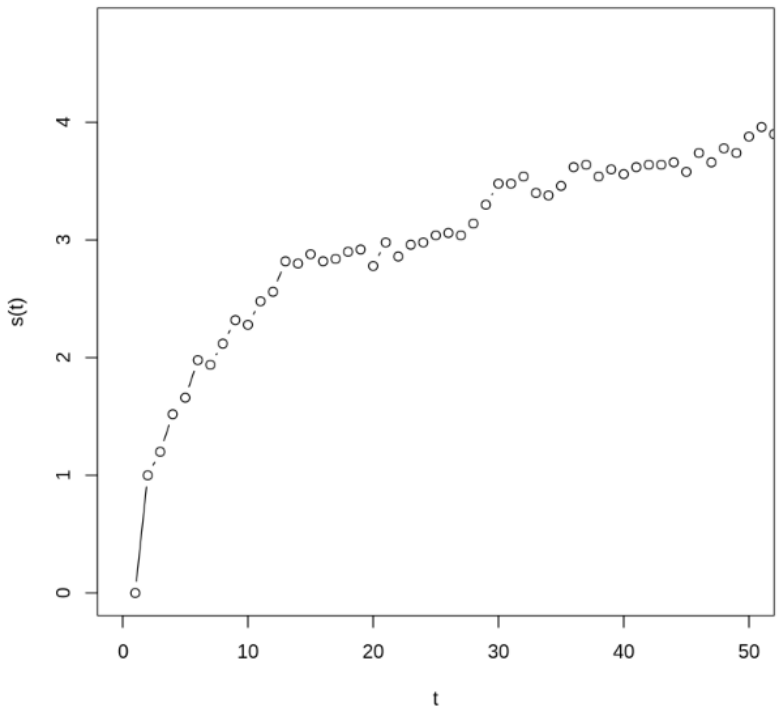
(d) Plot of ⟨s(t)⟩ v.s. t for 90 nodes, m=1



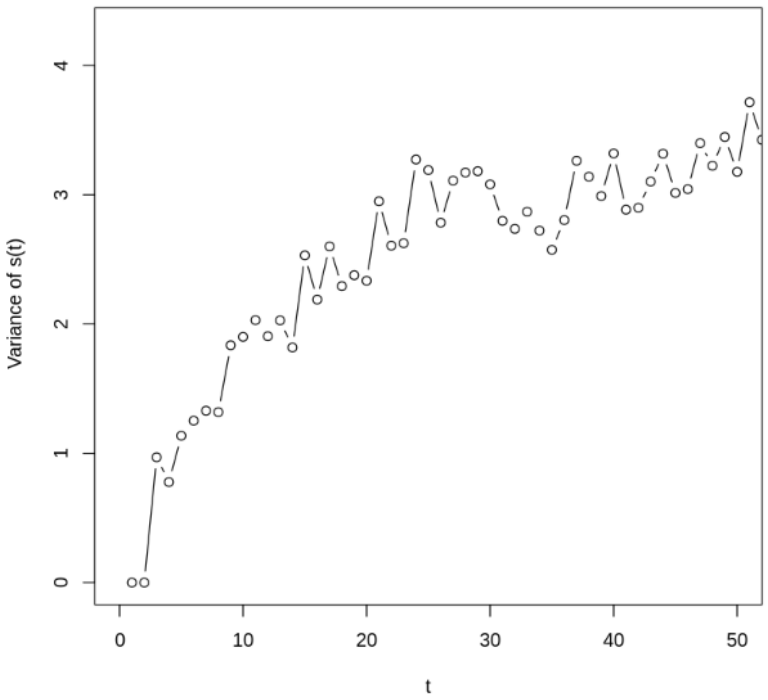
Plot of σ^2 (t) v.s. t. for 90 nodes, m=1



Plot of ⟨s(t)⟩ v.s. t for 9000 nodes, m=1



Plot of σ^2 (t) v.s. t. for 9000 nodes, m=1



For the 90 nodes network, the mean is centered between 3.0-4.0, the variance is centered around 3.0.
For the 9000 nodes network, the mean is centered between 3.0-4.0, the variance is centered around 3.0.

```
diameter(g4)
diameter(g5)
```

```
11
32
```

In this case, the diameter **doesn't play a role**, the networks with different diameters have similar mean distance and similar variance of distance.

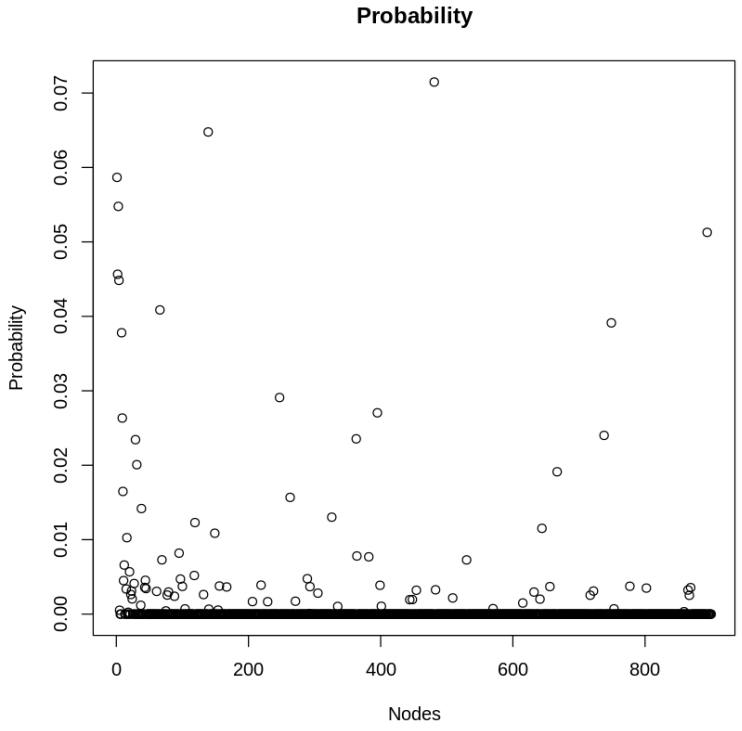
3. PageRank

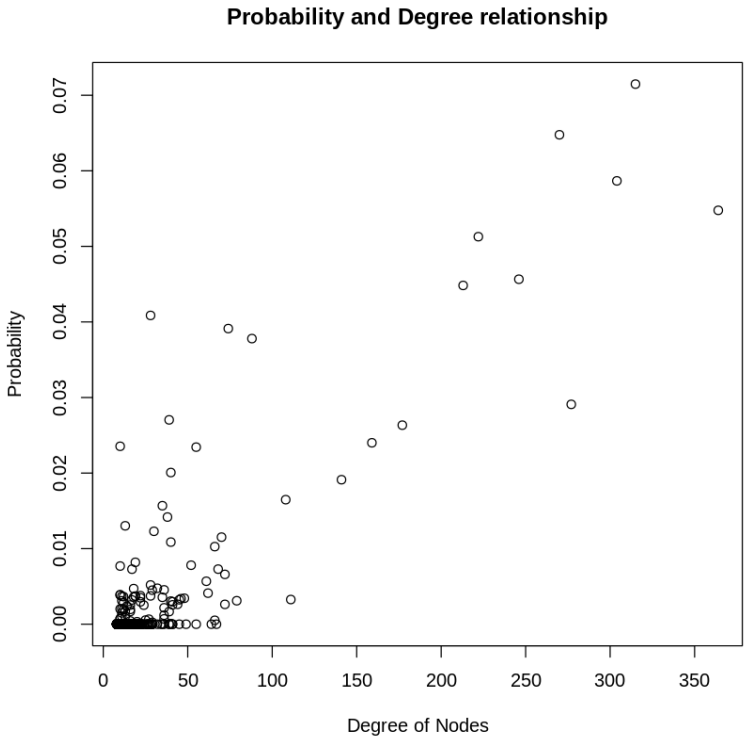
(a) We used the Transition Matrix and Random Walk function. Using barabasi.game to generate 2 graphs with parameters setted:n = 900, m = 4, directed = TRUE. In order to merge the two networks by adding the edges of the second graph to the first graph with

a shuffling of the indices of the nodes, we used add_edges between the first graph and the transposed edge list which was generated by the second graph.

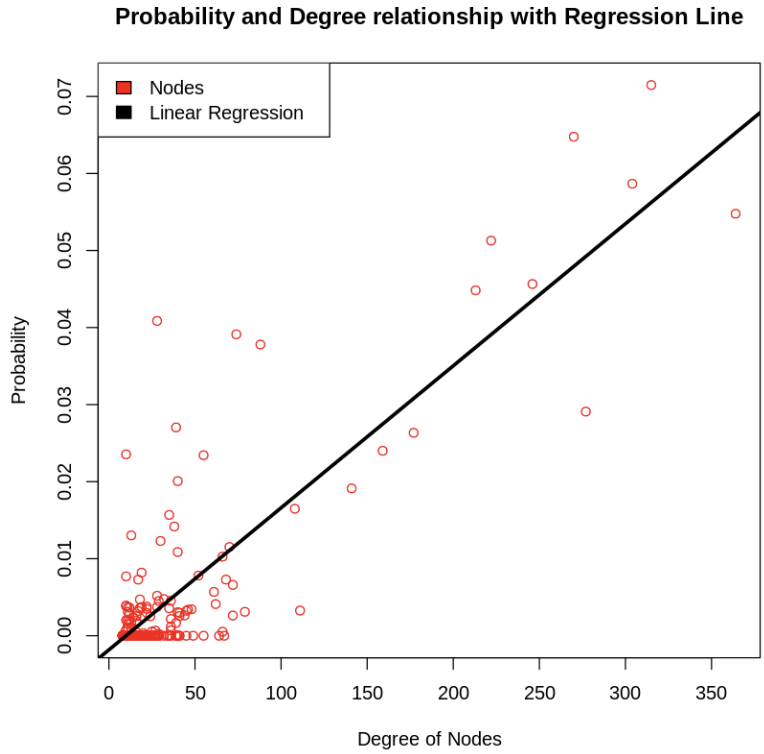
The random walk is created by setting parameters as: num_steps = 900 and looped 500 times to ensure that the random walk can get to the end nodes. As the walk passed the node one time, the frequency would plus 1. So the probability that the walker visits each node equals the overall frequency / sum(overall frequency).

The plots below show the the distribution of probability and the relationship between visiting probability and the degree of nodes:





As we can see most probability is 0, and most degree of nodes is less than 50, to view the relationship more clearly we also created linear regression between them:



```
Call:
lm(formula = probability_3a ~ degree3a)

Coefficients:
(Intercept)     degree3a
 -0.0018301    0.0001843
```

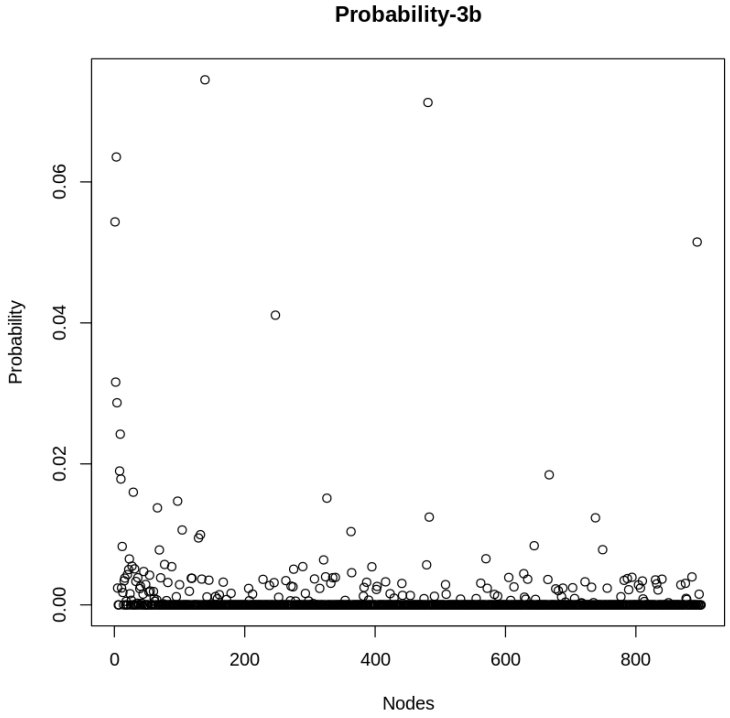
The result of linear regression: $y = 0.001843 * x - 0.0018301$

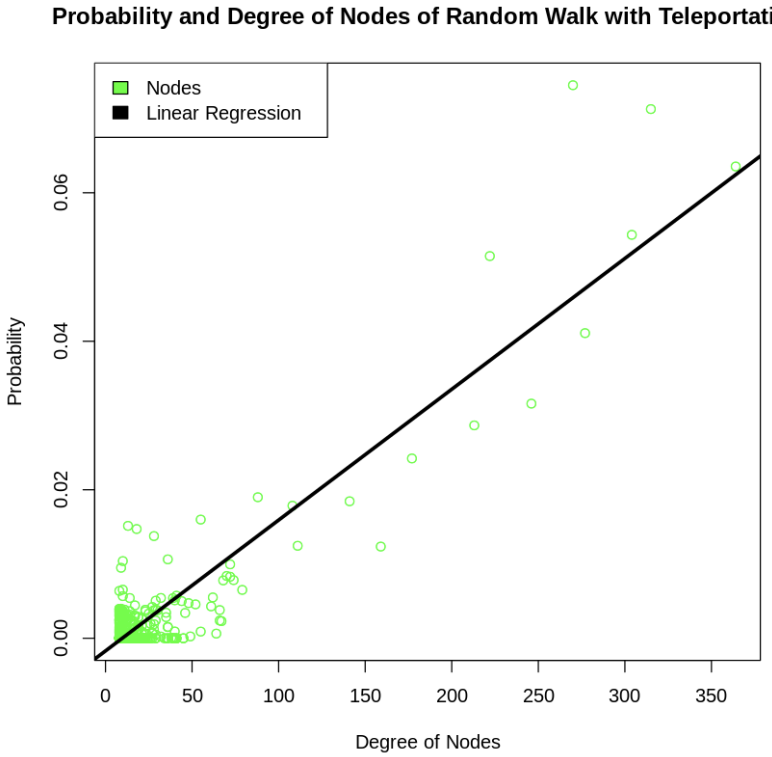
We also calculated the correlation between them: The correlation between Degree of nodes and Probability is **0.871192338152671**.

From both the correlation and the linear regression plot, we can conclude that the probability is strongly correlated with the degree of nodes. Because the possibility of nodes with higher degrees can be visited are larger than that of lower-degree nodes.

(b) Based on the random walk function in 3a, we combined the teleportation probability into it by change: generating a random number from 1 to 100, if it is smaller than α = 0.2 * 100, the probability of start node = (1/vcount(merged graph), vcount(merged graph)), merged graph is what we got in 3a.

The plot of the distribution of probability and plot of the relationship between visiting probability and the degree of nodes are as follows:






```
Call:
lm(formula = probability_3b ~ degree3b)

Coefficients:
(Intercept)     degree3b
 -0.0017001    0.0001762
```

The result of linear regression: $y = 0.0011762 * x - 0.0017001$

The correlation between Degree of nodes and Probability **0.91421664817641.**

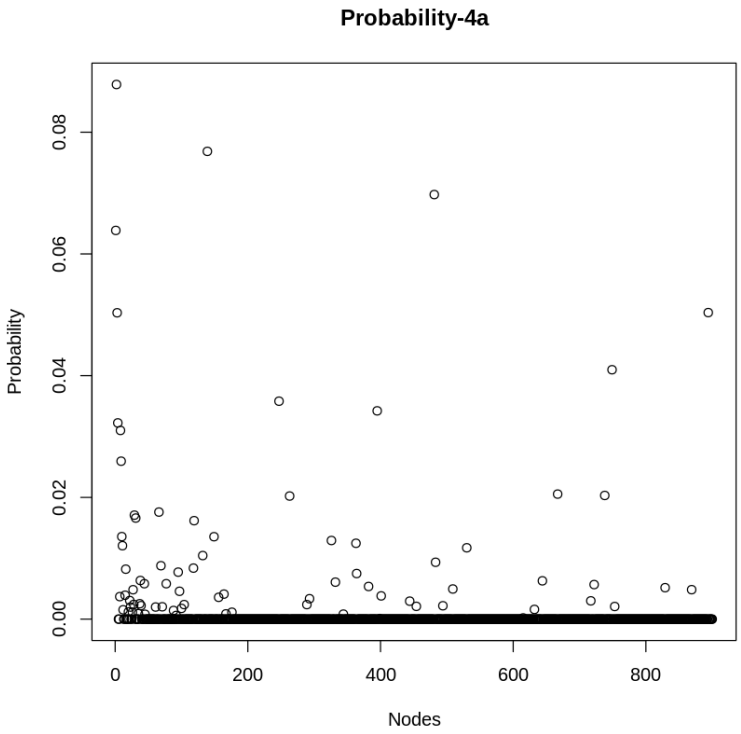
Compared with 3a, there are more nodes whose visiting probability is larger than 0. What's more, the correlation coefficient is larger which means that probability that the walker visits each node is more strongly correlated with the degree of nodes.

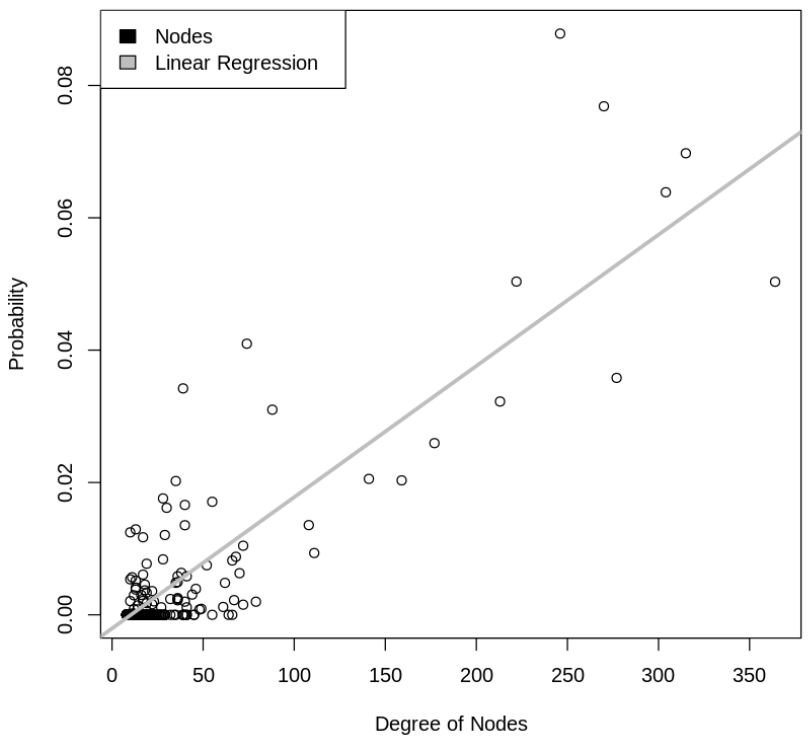
Compared with linear equations in 3a, 3b provides a lower slope. We can observe in 3b that the probabilities of nodes who have higher degrees decrease, compared with 3a. In other words, teleportation probability α can cause a lower slope, reducing the probability of nodes with higher degrees.

4. Personalized PageRank

(a) In this question, we still used the random walk function we defined in question 3b. In order to realize the teleportation probability to each node is proportional to its PageRank, we changed the probability of start node = probability we got in question 3a, when the random variable from 0 to 1 is less than α.

The plot of the distribution of probability and plot of the relationship between visiting probability and the degree of nodes are as follows:


Probability-4a
Probability
0.00
0.02
0.04
0.06
0.08
0
200
400
600
800
Nodes


Probability and Degree of Nodes with Proportional Probability
Nodes
Linear Regression
Probability
0.00
0.02
0.04
0.06
0.08
0
50
100
150
200
250
300
350
Degree of Nodes

```
Call:
lm(formula = probability_4a ~ degree4a)

Coefficients:
(Intercept)     degree4a
 -0.0020537    0.0001984
```

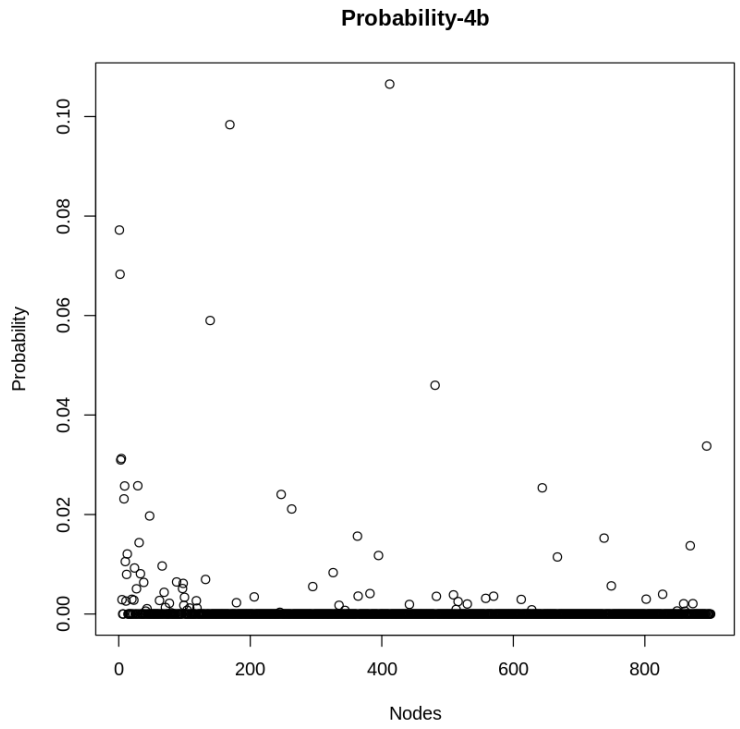
The result of linear regression: $y = 0.001984 * x - 0.0020537$

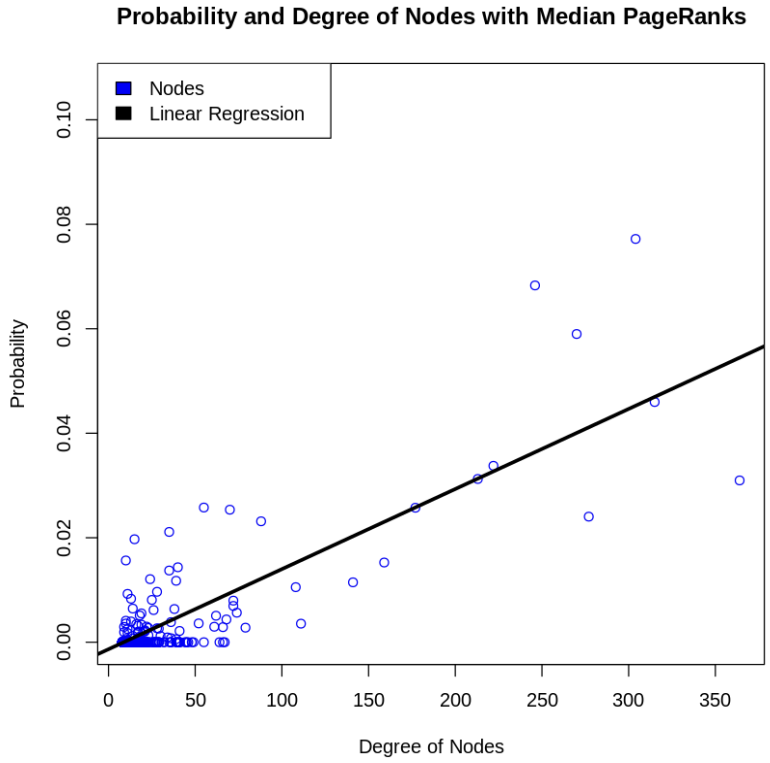
The correlation between Degree of nodes and Probability **0.875481641411298**.

Compared with 3a, the correlation coefficient is larger, but smaller than 3b. The probability is more strongly correlated with the degree of nodes.What's more, the slope is higher, and the probabilities of nodes with higher degree are increasing. Because the probability of teleportation is proportional to pagerank, so random walker is more possible to visit nodes with higher PageRank values, as a result, they have higher degrees.

(b)To get the median Pagerank value, using the page_rank function in the merged graph we have already got. After ordering the page rank values, the medians' indexes are half of node numbers and plus 1. Therefore, we can get the 2 medians. According to the random walk function, when random value is less than α, we set the probability of start node = probability of medians, which is $\frac{1}{2}$.

The plot of the distribution of probability and plot of the relationship between visiting probability and the degree of nodes are as follows:



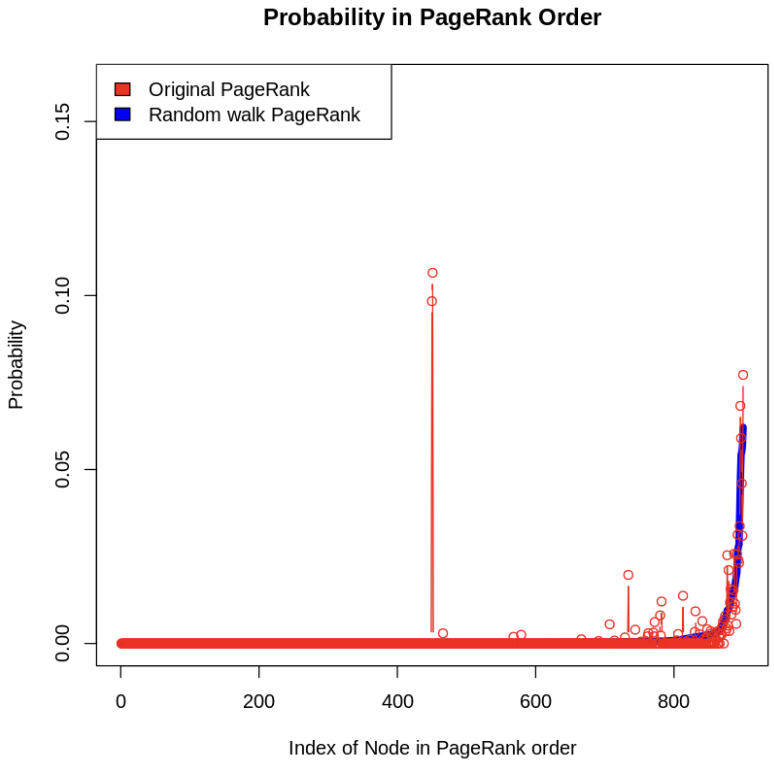


The result of linear regression: $y = 0.001533 * x - 0.0013352$

The correlation between Degree of nodes and Probability **0.616974312852263**.

From the correlation coefficient, the result is smaller than the previous. The probability is not very correlated with the degree of nodes. Because we can observe from previous plots in 4b that two medians have dramatically higher probability to be visited, however the rest of nodes will not be affected by teleportation

As we need to know the influence of PageRank values, we plotted the comparison between original Pagerank and random walk:

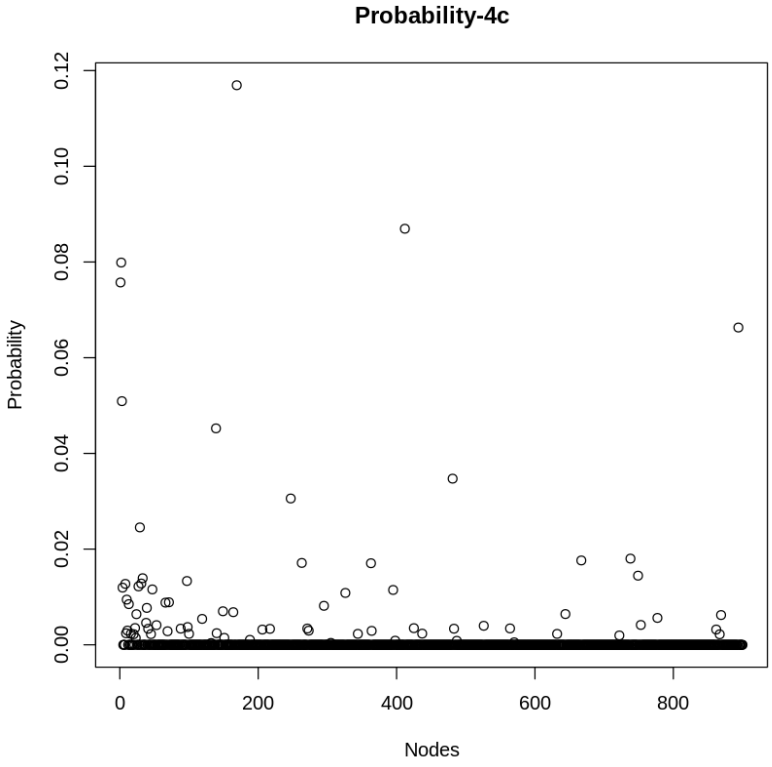


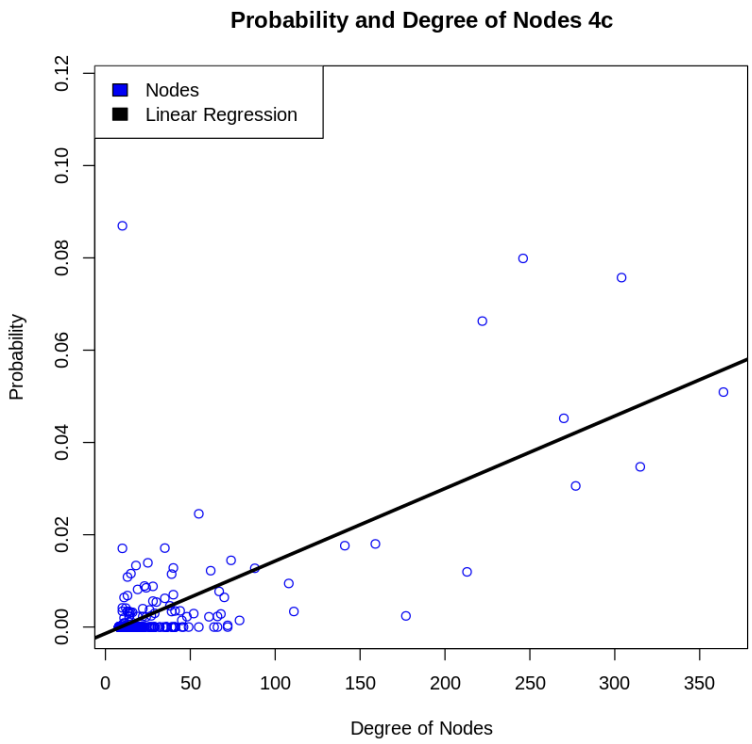
As for the index of nodes following the PageRank order, from the plot, there are a small number of nodes that have a higher probability than the original. When the medians get visited, the probability of the nodes in the range near medians will increase, the PageRank values will increase as well.

(c) To assume the effect of this self-reinforcement, we decided to combine question 4a and 4b, which means the teleportation probability is defined by both PageRank and the trusted web pages. We changed the probability of nodes who have median PageRank values into $\frac{1}{2}\beta$, and the other nodes can be the normal distribution, but are added a weight of $1 - \beta$. As a result, the changed probability vector is:

$$P = \frac{1}{2} * \beta * (nodes\ with\ median\ PageRanks) + (1 - \beta) * PageRank$$

If we set β = 0.8, other steps are all the same with previous questions.The plot of the distribution of probability and plot of the relationship between visiting probability and the degree of nodes are as follows:





The correlation between Degree of nodes and Probability **0.616168062160833**.