ECE 232 Project 1

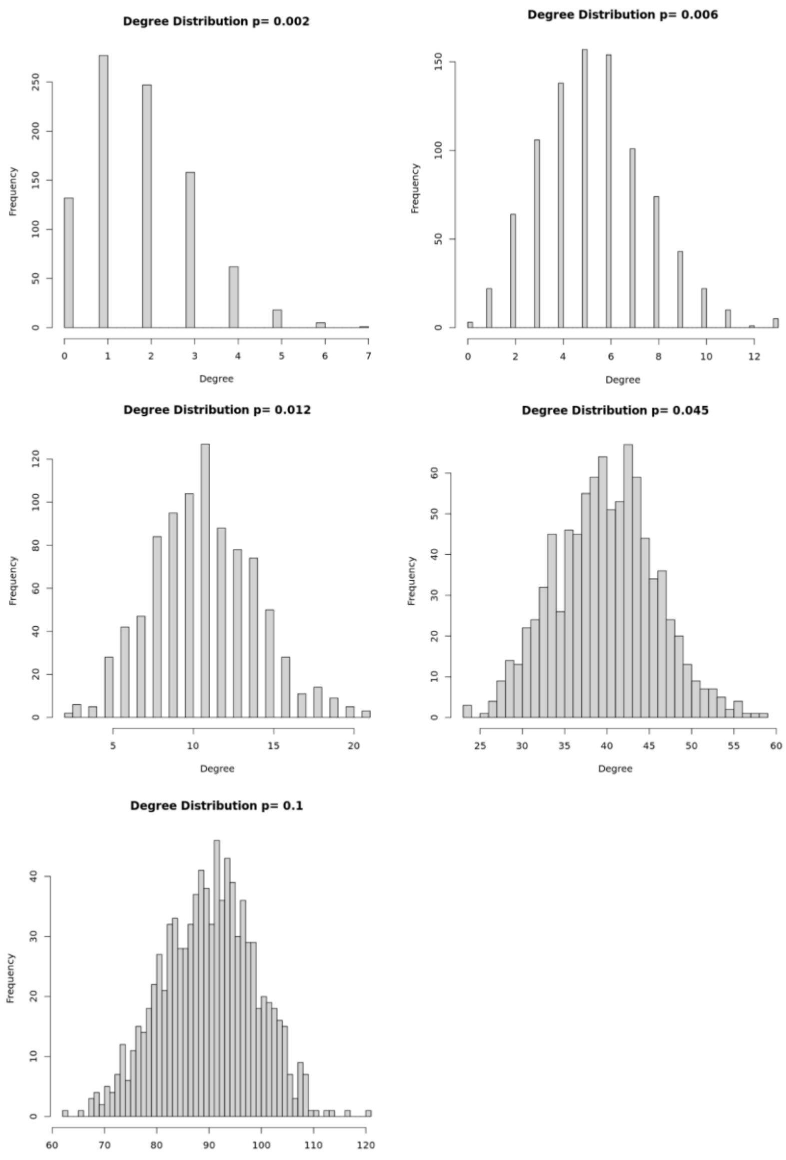
Angela Fan

Joo Wan Lim

Part 1. Generating Random Numbers

1. **Create random network using Erdös-Rényi Network**
    a. **We create undirected random network with n=900 nodes, with probability of drawing edge between two arbitrary vertices being 0.002, 0.006, 0.012, 0.045, and 0.1.**

    The plots are provided below:



From the table below, we can observe the mean and variance of our degree distribution is very close to its theoretical values.

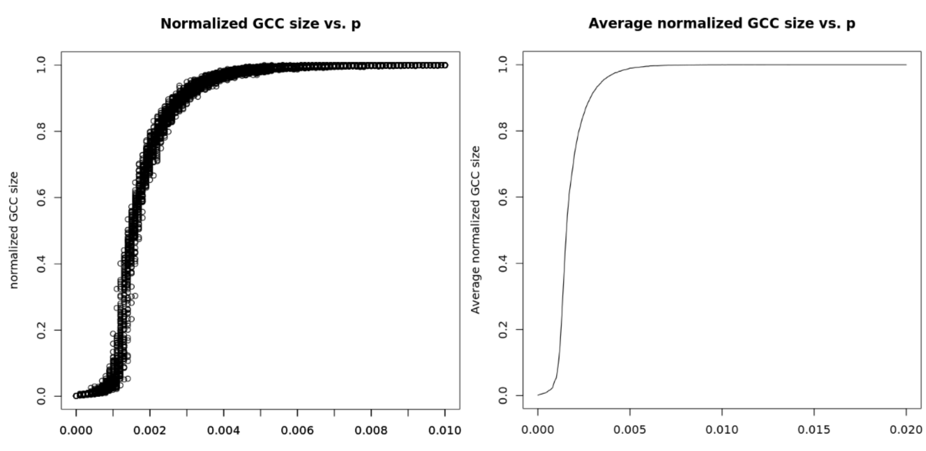
p	Mean	Theoretical Mean	Variance	Theoretical Variance
0.002	1.800	1.798	1.604	1.794
0.006	5.358	5.394	5.162	5.362
0.012	10.893	10.788	10.856	10.659
0.045	40.282	40.455	35.108	38.634
0.1	90.442	89.900	79.246	80.910

b. For each p and n = 900, answer the following questions: Are all random realizations of the ER network connected? Numerically estimate th probability that a generated network is connected. For one instance of the networ s with th$a$t p, find the giant connected component (GCC) if not connected. What is the diameter of the GCC?

p	Connected	Probability that the network is connected	Diameter of GCC
0.002	False	0%	23
0.006	False	1.6%	8
0.012	True	98.3%	5
0.045	True	100%	3
0.100	True	100%	3

c. Normalized GCC is highly nonlinear function of p, with interesting properties occurring for values where, $\boldsymbol{p = O(\frac{\ln n}{n})}$. For n = 900, weep over values of p and find GCC sizes.

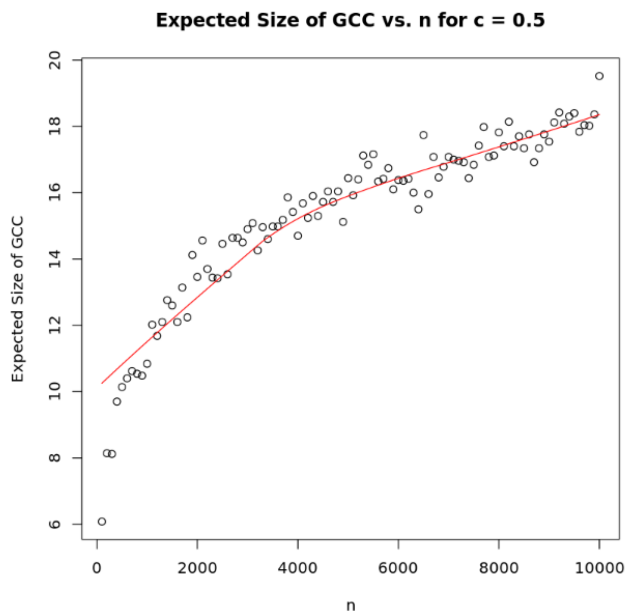
The estimated value of p is: $\frac{\ln(900)}{900} = 0.0075$.



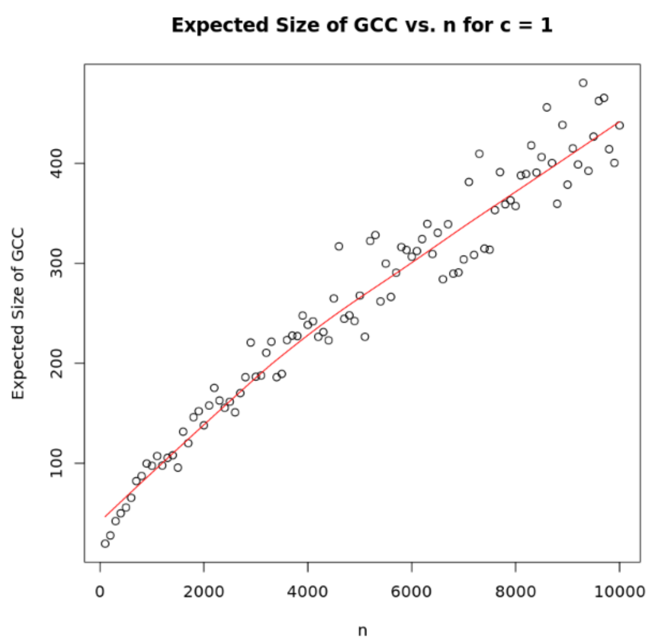
In observing both graph, the threshold for p is around 0.004, which is quite close to the theoretical value we calculated which was 0.0075.

d.

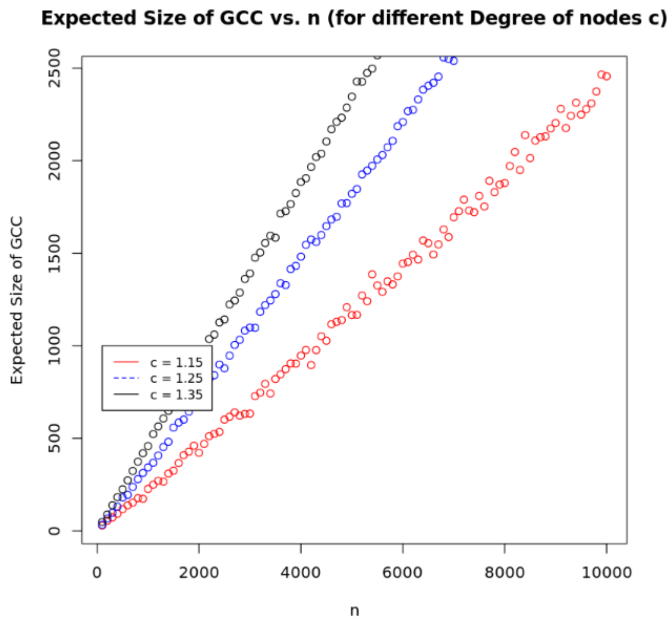
i. **Define the average degree of nodes c = n x p = 0.5. Sweep over the number of nodes n, raning from 100 to 1000. Plot the expected size of the GCC of ER networks with n nodes and edge formation probabilities p = c/n, as function of n? What trends is observed?**



ii. **Repeat the came for c = 1.**



iii. **Repeat the same for values of c = 1.15, 1.25, 1.35 and show the results for these three values in a single plot**

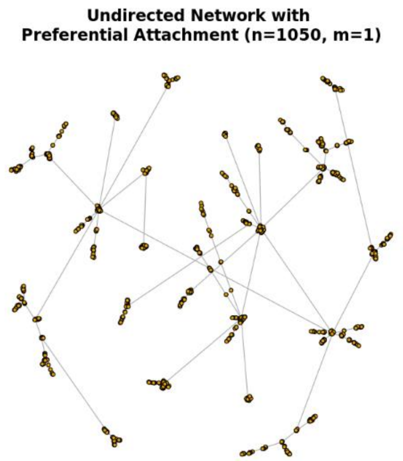


iv. **What is the relation between the expected GCC size and n in each case?**

From the graphs presented, it is observed that the size of the GCC (Giant Connected Component) increases as the value of n increases. Additionally, the rate at which the GCC size increases becomes more pronounced with higher values of c for a given n. This relationship aligns with the equation p=c/n, suggesting that a larger c leads to a greater p, thereby increasing the likelihood that more nodes are connected, and consequently enlarging the size of the GCC.

2. **Create networks using preferential attachment model**
   a. **Create an undirected network with *n* = 1050 nodes, with preferential attachment model, where each new node attaches to *m* = 1 old nodes. Is such network always connected?**

In this section, we discuss the creation of networks using the Barabási-Albert (BA) method. The BA model is a process used to create random networks that don't have a typical scale, due to their reliance on a method called preferential attachment. This method dictates that new nodes joining the network are more likely to connect to nodes that already have a larger number of connections, in comparison to those with fewer connections. In other words, new nodes tend to favor linking with those that are already well-connected. The resulting network's degree distribution—the way connections are distributed among the nodes—does not have a typical scale and follows a specific mathematical power law:$P(k) \sim ck^{-\gamma}$, where $\gamma$is a constant that usually falls between -2 and -3. We put this model into action by constructing an undirected network made up of 1050 nodes using the preferential attachment rule where each new node connects to exactly one existing node, ensuring that all parts of the network are reachable from any other part.



b. Use fast greedy method to find the community structure. Measure modularity. Define Assortativity. Compute Assortativity.

Assortativity is defined as a preference for network's node to attach to others that are similar in some way.
The modularity of the network is: 0.9351
The degree assortativity of the network is -0.0911

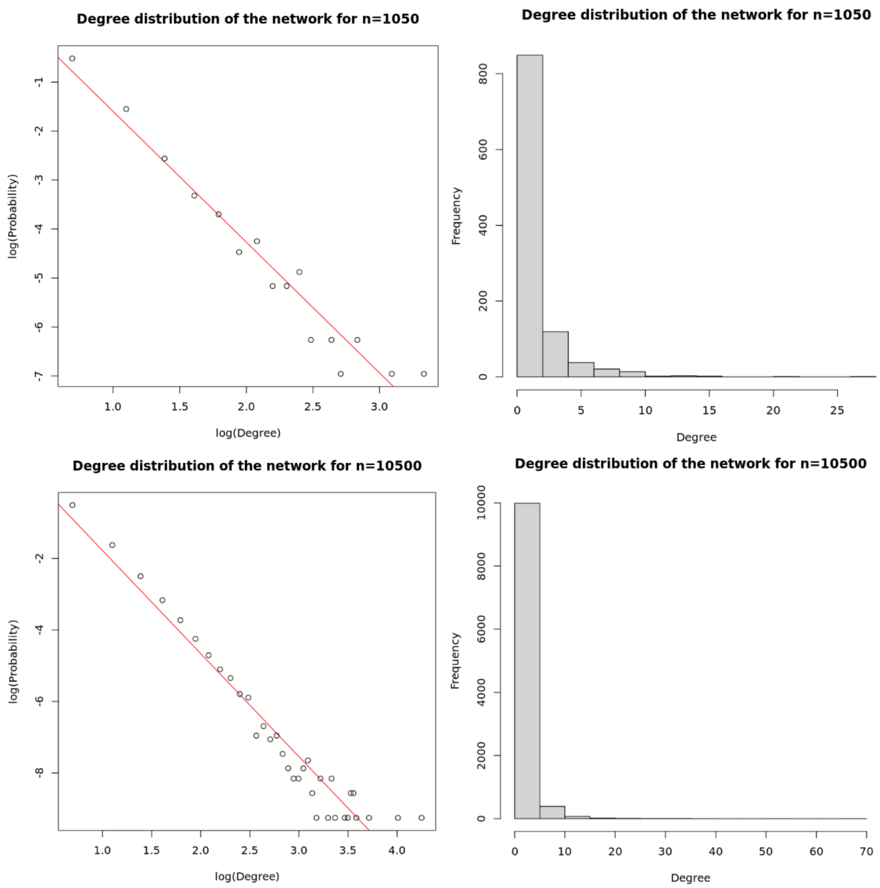
c. Try to generate a larger network with 10500 nodes using the same model. Compute modularity and assortativity. How is it compared to the smaller network's modularity?

We use the same model to generate an undirected network with 10500 nodes. The modularity and assortativity is show below:

Number of nodes	Modularity	Assortativity
1050	0.9351	-0.0911
10500	0.9785	-0.0455

The outcomes indicate that networks with a larger number of nodes not only exhibit higher modularity but also show increased assortativity.

d. Plot the degree distribution in a log-log scale for both n = 1050, 10500, then estimate the slope of plot using linear regression.

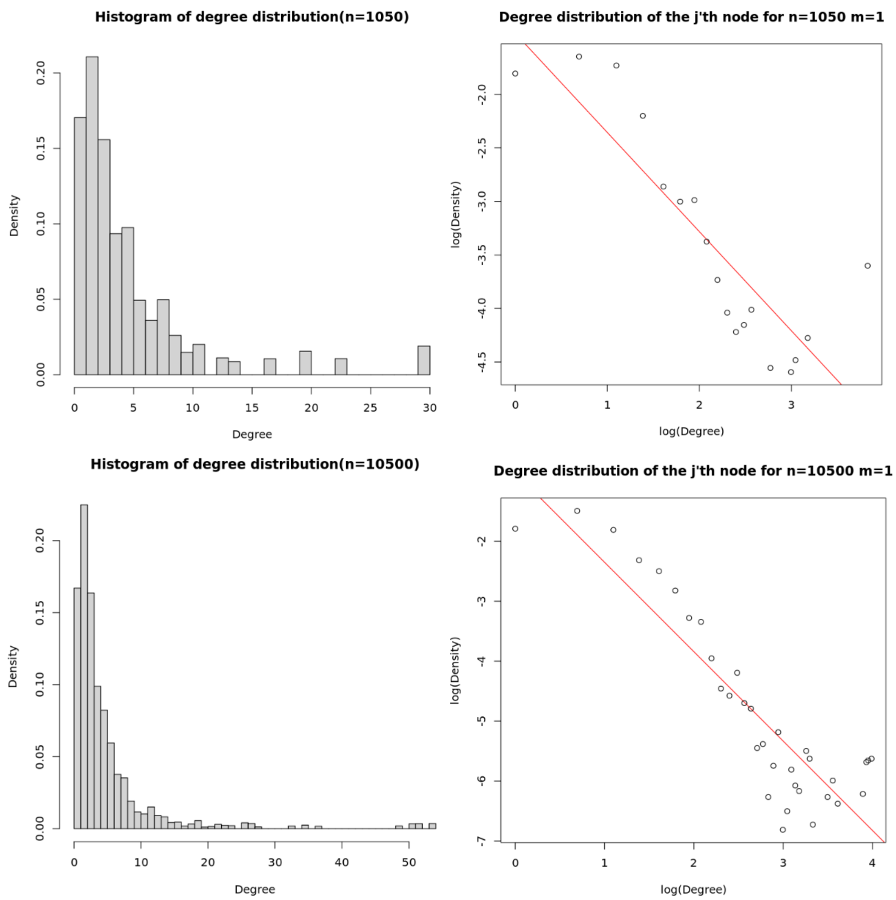


Number of node	Slope	Intercept
1050	-2.671	1.072
10500	-2.886	1.109

From the above figures, the estimated power law exponent is 2.67 for n = 1050 n=1050 and 2.89 for n=10500. These values are quite close to the theoretical power law exponent of 3 predicted by the Barabási model, indicating that our estimates are reasonable.

e. In the two networks generated in 2(a) and 2(c), perform the following: Randomly pick a node *I*, and then randomly pick a neighbor *j* of that node. Plot the degree distribution of nodes *j* that are picked with this process, in the log-log scale. Is the distribution linear in the log-log scale? If so, what is the slope? How does this differ from the node degree distribution?

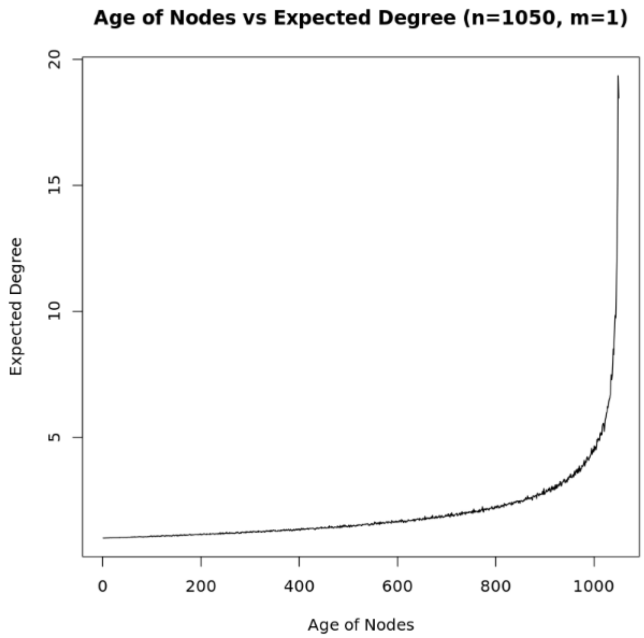
For networks with n=1050 and n=10500, we select a node i at random and then choose one of its neighbors, node j, at random. We then plot the degree distribution of the nodes j selected through this method on a log-log scale.



Number of nodes	Slope	Intercept
1050	-0.9265	-1.4292
10500	-0.8592	-1.4914

From examining the graph, the trend appears to be linear. The table provided lists the slope of this line. Although the slope is negative and diminishes as � n decreases, it's evident that the degree distribution of randomly selected nodes does not align with the power law distribution described by the Barabási model, which has an exponent of 3.

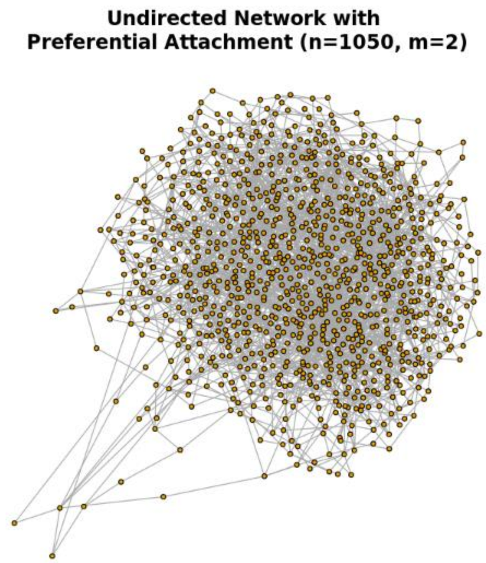
f. **Estimate the expected degree of a node that is added at time step i for $1 \leq i \leq 1050$. Show the relationship between the age of nodes and their expected degree through an appropriate plot. Note that the newest added node is the youngest.**

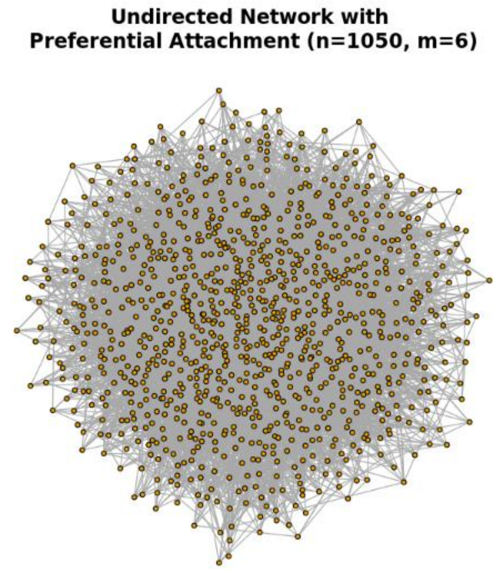


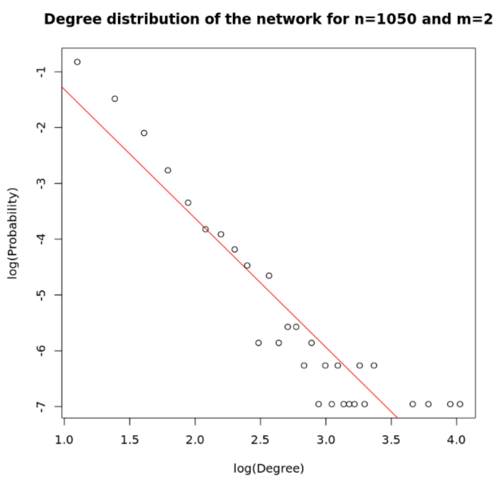
The plot illustrates the relationship between the age of a node and its expected degree. It shows that as nodes age, their expected degree tends to increase. This observation is consistent with the mathematical framework of Barabási networks. According to the preferential attachment model, a new node added to the network is more likely to connect with an already well-connected node (a node with a higher degree). Consequently, nodes established earlier in the network's formation gain more connections compared to those added more recently.

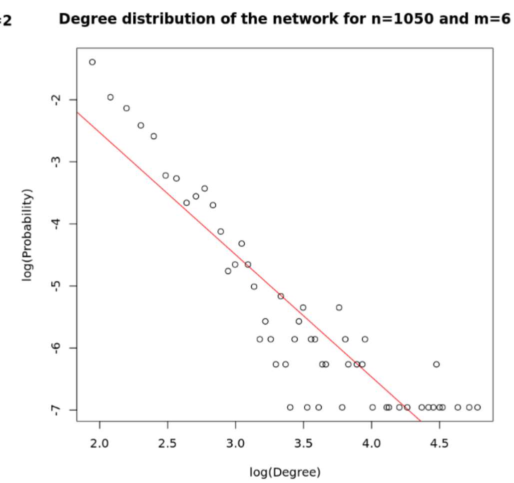
g. **Repeat the previous part (a-f) for *m*=2, and *m*=6. Compare the results of each part for different values of *m*.**

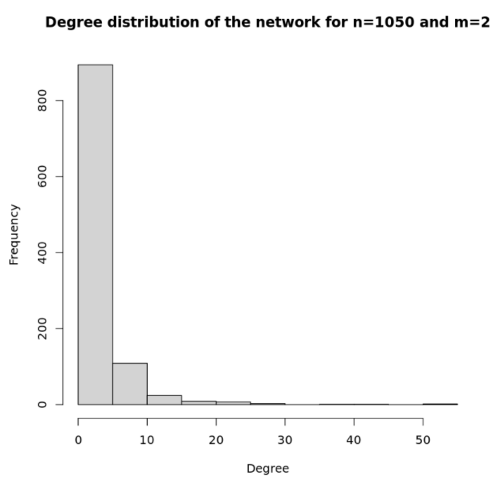
In a Barabási network, the parameter *m* denotes the number of edges that a new node forms with existing nodes. Increasing *m* leads to a higher number of connections between the new node and the rest of the network, decreasing the network's sparsity, particularly among inter-community links. This in turn reduces the network's modularity.
The experiments from 2a to 2f were repeated for *m*=2 and *m*=5, with the resulting plots and figures presented below.

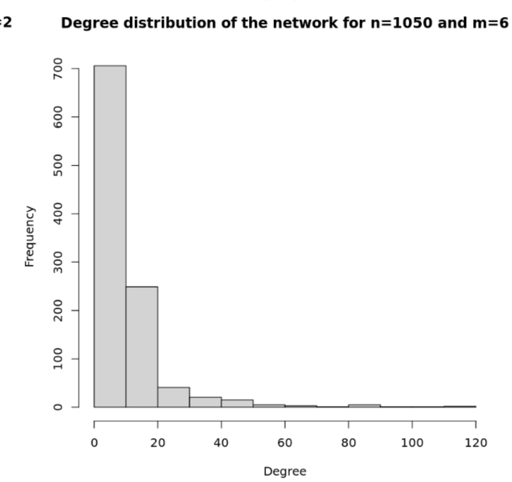

Undirected Network with
Preferential Attachment (n=1050, m=2)

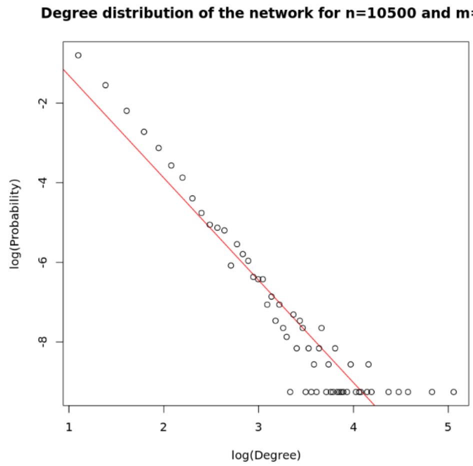

Undirected Network with
Preferential Attachment (n=1050, m=6)

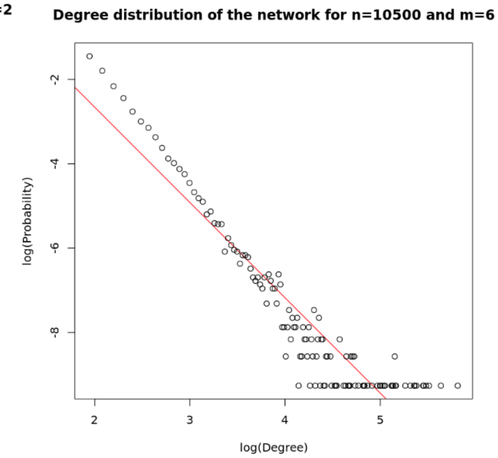

Degree distribution of the network for n=1050 and m=2
log(Probability)
log(Degree)

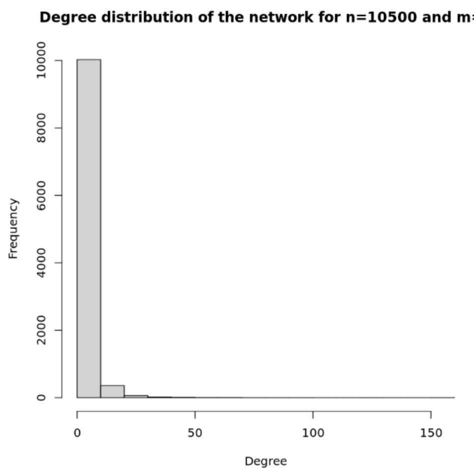

Degree distribution of the network for n=1050 and m=6
log(Probability)
log(Degree)

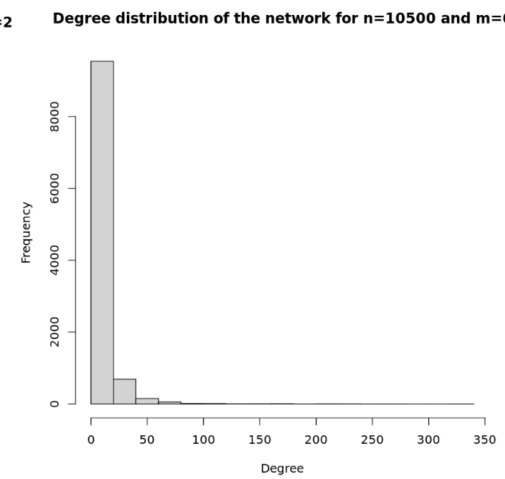

Degree distribution of the network for n=1050 and m=2
Frequency
Degree


Degree distribution of the network for n=1050 and m=6
Frequency
Degree







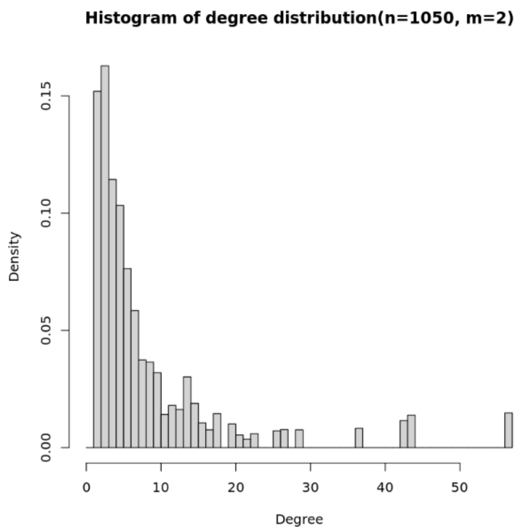


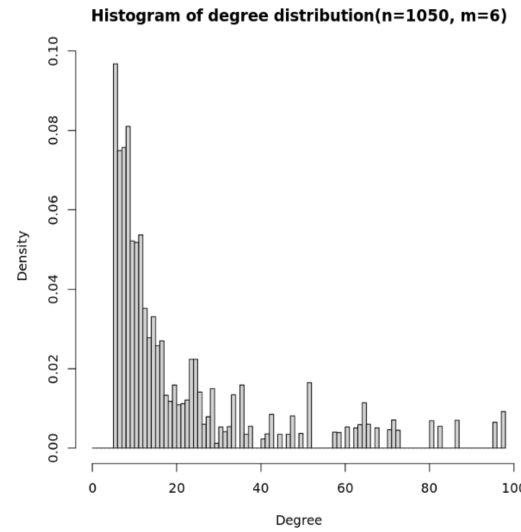
We aim to determine the slope of the log-log distribution plot, which is instrumental in estimating the exponent of the power law distribution. Across each scenario, the relationship is linear and exhibits a negative slope. The calculated slopes for the respective experiments are detailed below:

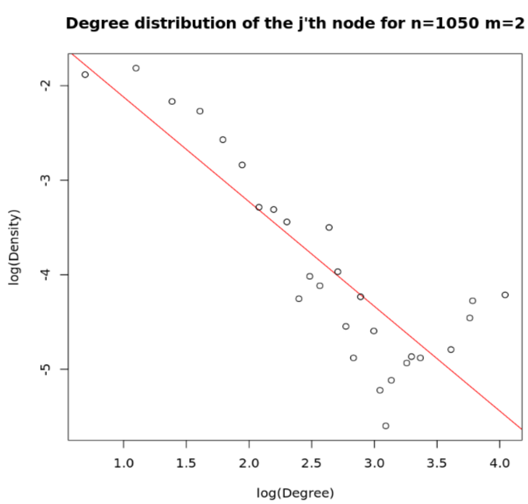
Number of nodes	M	Slope	Intercept
1050	2	-2.310	0.993
1050	6	-1.969	1.411
10500	2	-2.573	1.269
10500	6	-2.257	1.854

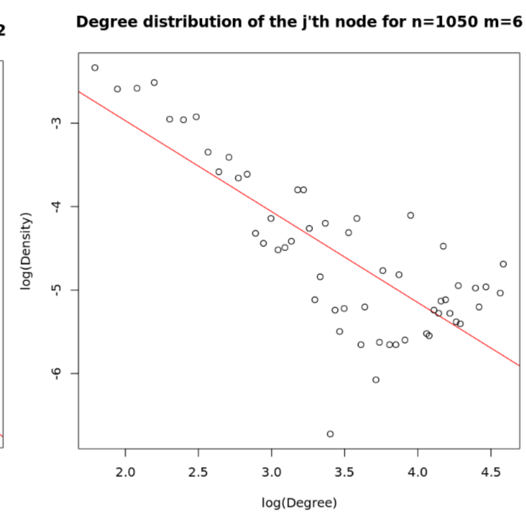
We note that with each fixed *n*, increasing *m* from 1 to 6 and then to 6 causes a gradual rise in the power law exponent away from the -3 benchmark typical for Barabasi networks. Concurrently, the likelihood of given degree also ascends, with this increment being more pronounced at higher degrees of *k*.

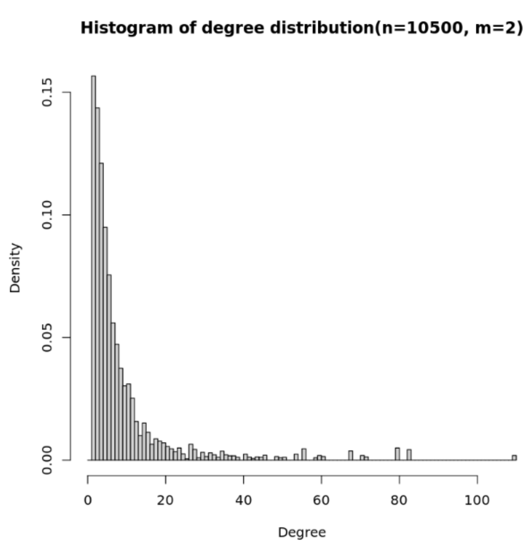
In the graph below, we pick a random neighbour *j* of a random node *i* and plot the degree distribution on a log-log plot of that node. The slope for each case is also mentioned in the table below.

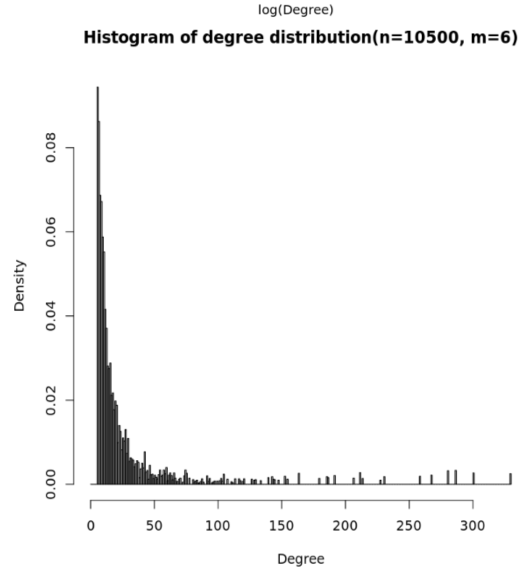


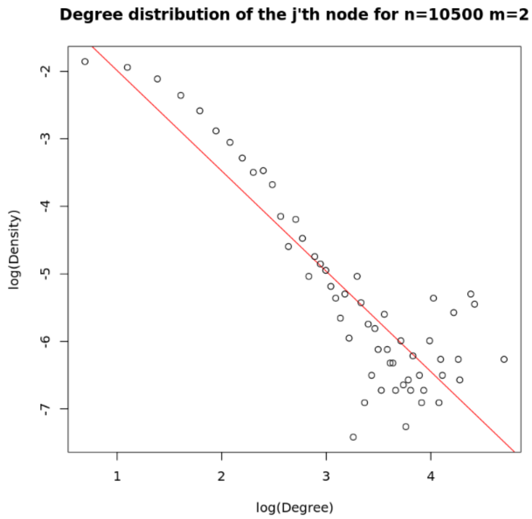


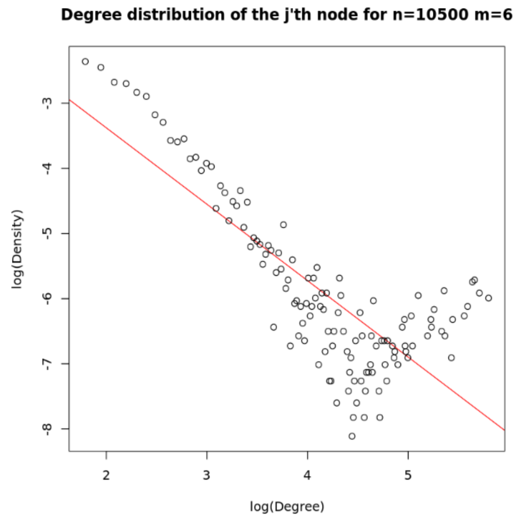








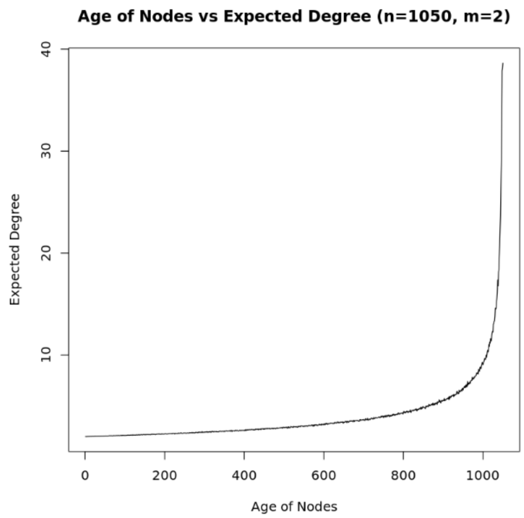


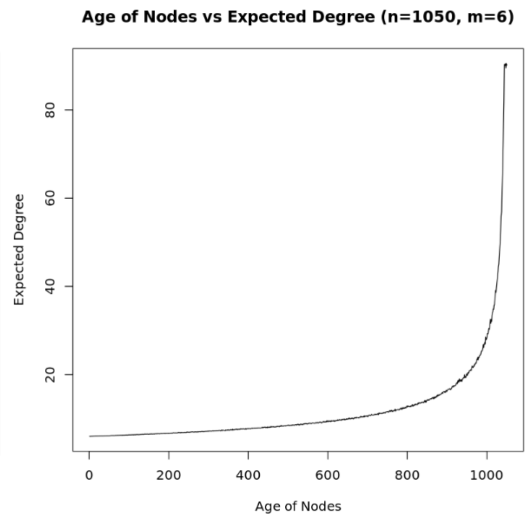


Number of nodes	M	Slope	Intercept
1050	2	-1.107	-1.014
1050	6	-1.0897	-0.7911
10500	2	-1.487	-0.500
10500	6	-1.173	-1.032

The degree distribution's slope for the *j* th node shows a considerable deviation from the typical power law exponent close to -3, as observed in Barabási networks. Despite this, the trend maintains linearity with a negative slope. With an increase in *m* for a constant number of nodes *n* within the network, there's an increase in the power law exponent for the *j* th node, which reflects a similar increase in the overall power law exponent of the network's graph.

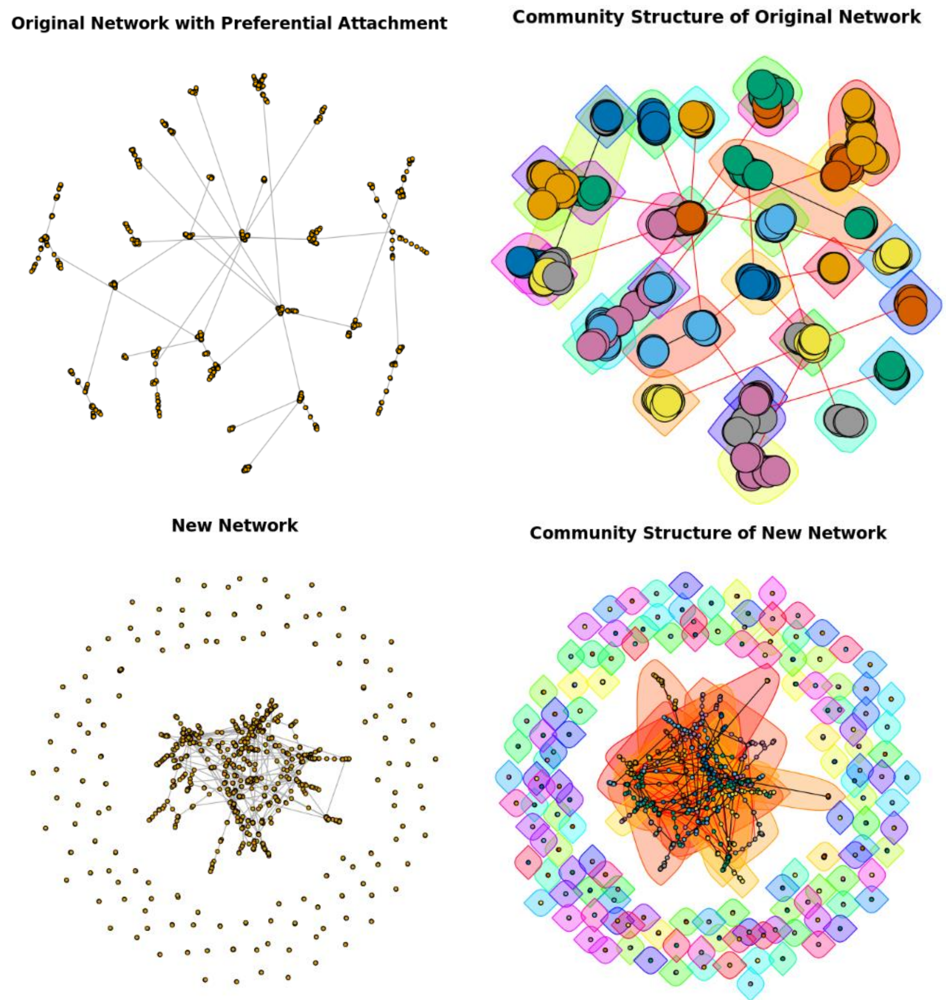
In the graph below, we depict the age of node vs expected degree for networks with $n = 1050$ and m=2,6. This graph suggests that an increase in a node's age correlates with a rise in its expected degree, a pattern that holds for both tested values of \(m \) and aligns well with the preferential attachment model characteristic of the Barabási network.





h. Again, generate a preferential attachment network with *n*=1050, *m*=1. Take its degree sequence and create a new network with the same degree sequence, through stub-matching procedure. Plot both networks, mark communities on their plots, and measure their modularity. Compare the two procedures for creating a random power law networks.

During the stub matching procedure, each vertex has a pre-set degree instead of relying on a probability distribution to determine this degree. Unlike the Erdős-Rényi (ER) model, where the degree sequence typically follows a Binomial or Poisson distribution, the configuration model allows for more flexibility by permitting any desired degree distribution. The fundamental principle of the stub matching method for generating random networks is to start with a given degree sequence and reduce the network to a series of "stubs," which are essentially nodes with unconnected edges. The construction of the random network then proceeds by randomly pairing these stubs to form connections. For both types of networks, the fast greedy algorithm is employed to identify the community structure.



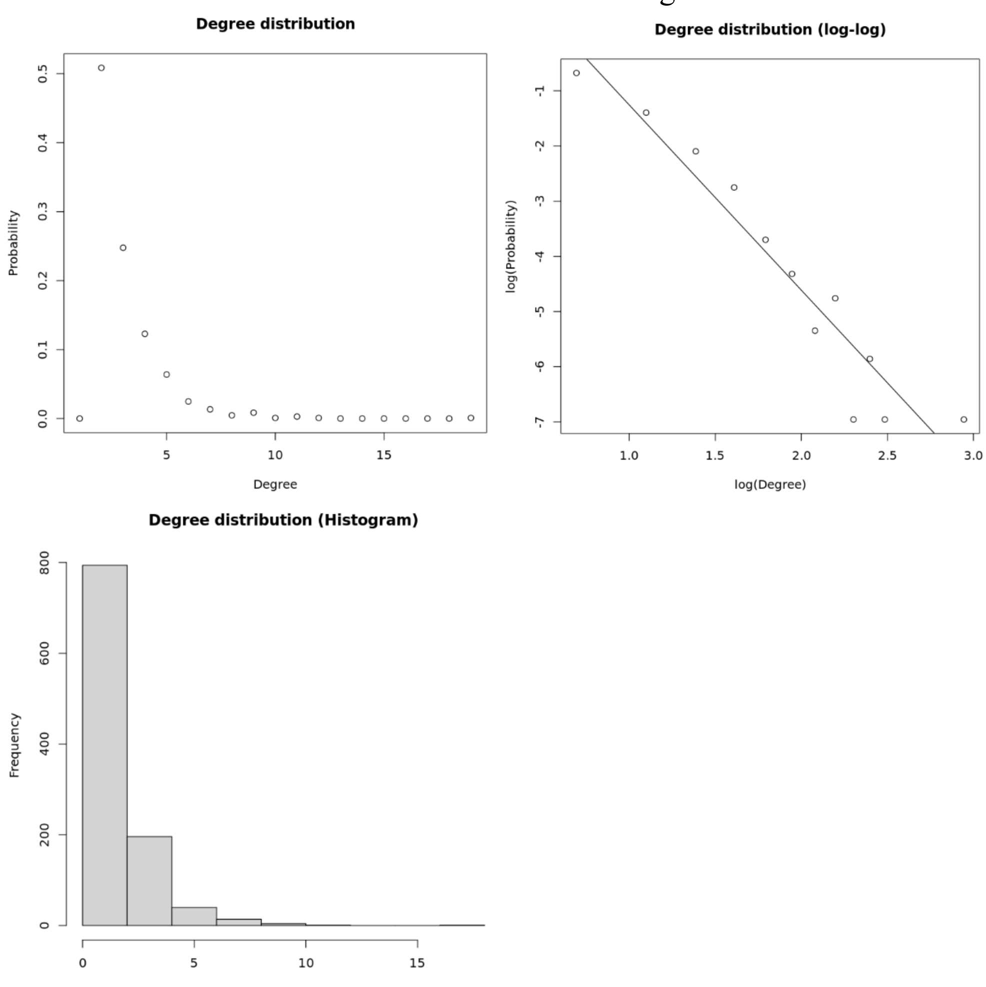
The modularity of the old network is 0.9336546 and the modularity of the new network is 0.8397398. In networks formed via the stub matching process, numerous dangling nodes remain unconnected, leading to fewer intra-community connections and hence a less dense community structure. This stems from the inherent randomness of stub matching, which does not guarantee the full connectivity that is a characteristic feature of the preferential attachment model, where each new node connects to an existing member, maintaining network integrity throughout its growth. Consequently, networks developed through stub matching tend to exhibit lower modularity in contrast to those formed by preferential attachment, likely due to the decreased connectivity between different modules inherent to the stub matching approach.

3. **Create a modified preferential attachment model that penalizes the age of a node**
    a. **Each time a new vertex is added, it creates *m* links to old vertices and the probability that an old vertex is cited depends on its degree, preferential attachment, and age. Produce such an undirected network with 1050 nodes and parameters *m*=1, $\boldsymbol{\alpha = 1}$, $\boldsymbol{\beta = -1}$and a=*c*=*d*=1,*b*=0. Plot the degree distribution. What is power law exponent?**

    In this section, the construction of a network includes a penalty based on node age. The probability distribution for this tpe of network is described as follows:

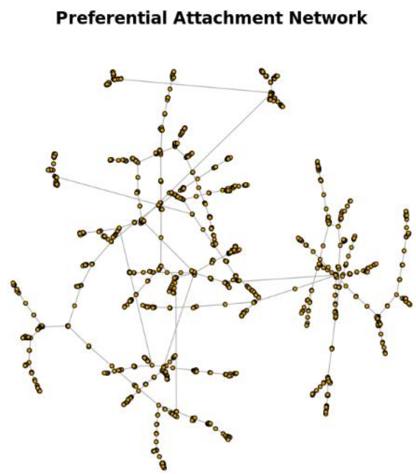
    $$P(i) \sim (c * k_i^{-\alpha} + a) + (d - l_i^{-\beta} + b)$$

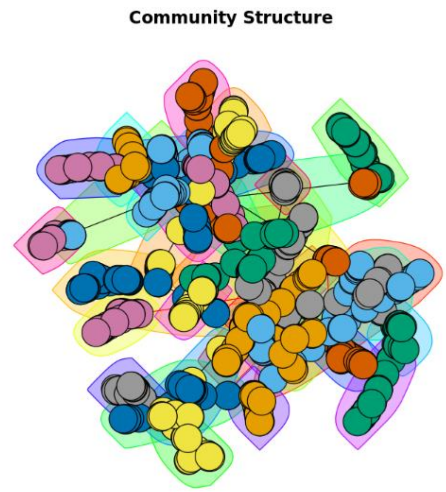
    where k_i represents the node's degree and l_i its age. The parameter α stand for the exponent related to preferential attachment, while β denotes the aging exponent. The term α relates to the degree of attraction for nodes that have no immediate neighbors, and *b* to the initial attractiveness for nodes at the beginning of their lifespan. *c* and *d* are coefficients that correspond to degree and age, respectively. With $\beta = -1$ and $\alpha = 1$ as specified parameters, the likelihood of forming a connection with an older node increases with its degree but decreases with its age.



b. **Use fast greedy method to find the community structure. What is the modularity?**

The network formed by incorporating age penalties is depicted in the figures below. It exhibits a modularity of 0.9361651. Given that modularity quantifies the strength of community structure within a network, it can be inferred that introducing age penalties during network construction fosters the development of robust communities or modules. Within these communities, connections are dense, while connections between nodes from different communities remain sparse.

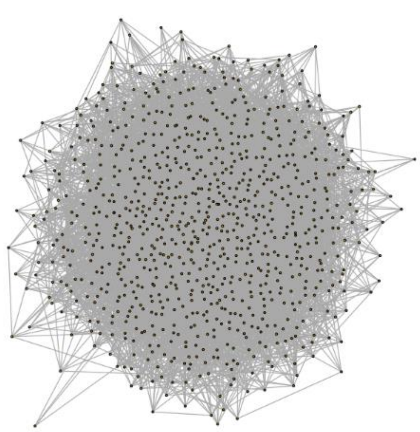




Part 2. Random Walk on Networks

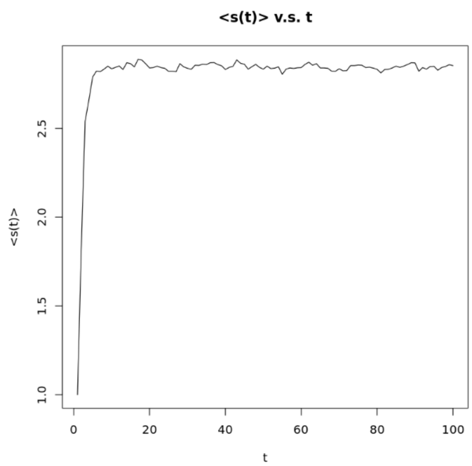
1. Random walk on Erdös-Rényi Network

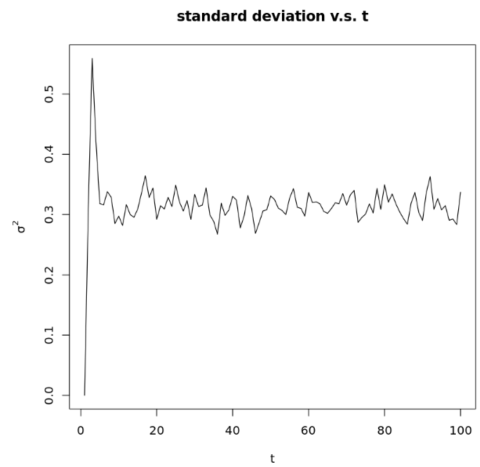
a. We created an undirected random networks with 900 nodes and the probability of edge creation equaling to 0.015, using *erdos.renyi.game()* function. The created graph looks like the following:



b. A random walker began at a node chosen at random and took t steps to reach its final destination. The shortest path lengths, denoted as s(t), were determined for each destination. These lengths varied with different starting points, leading us to plot the average and variance of these distances for varying starting nodes against the number of steps, t. Given the consideration for computational efficiency, we limited our analysis to 100 steps within a network of 900 nodes. Additionally, we evaluated the impact of the number of trials (different random starting points) on these metrics. We observed that increasing the number of trials tends to stabilize the mean and variance after a certain number of steps. For instance, conducting 1000 trials yielded more consistent results than just 100 trials, albeit at the cost of increased computational time. For this study, we settled on 1000 trials.

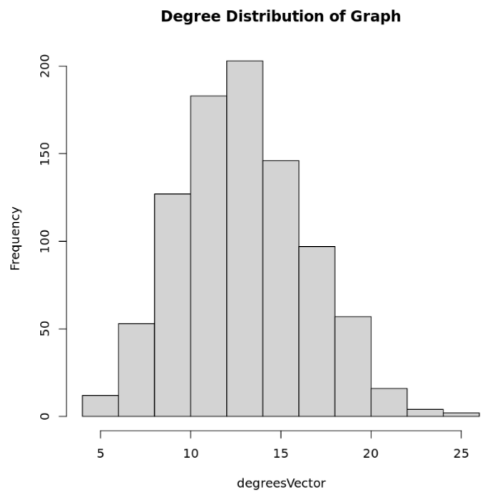
The plots are shown below:

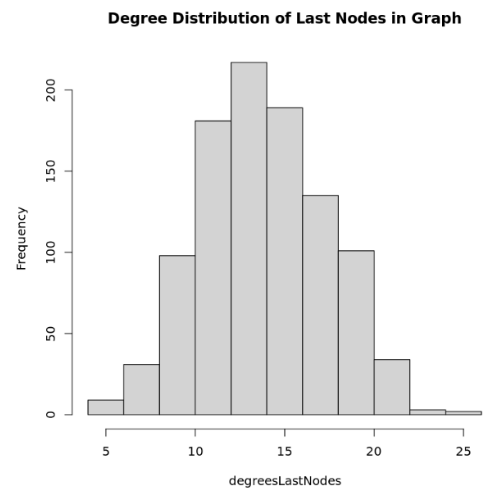




From the provided graph, we can clearly see that the mean value of the shortest path length converges to 2.9, and the standard deviation converges to 0.3 after 10 steps.

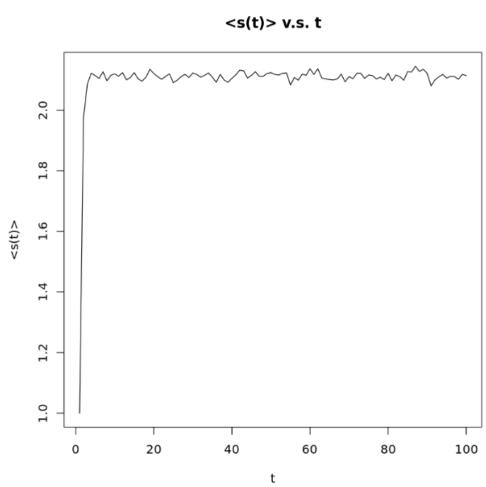
c. We also calculate the degree distribution of nodes reacted at the end of the random walk.
The plots are shown below:

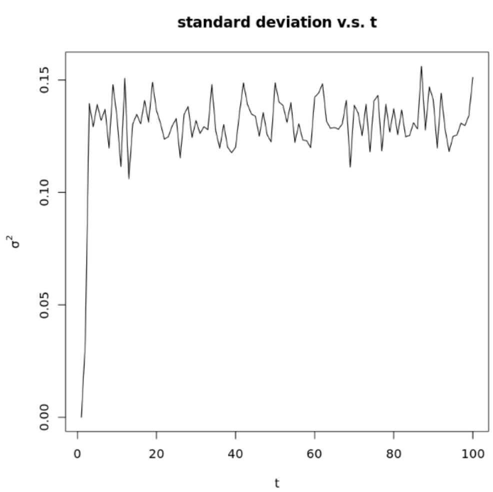




From observing the two plot we can see that they both have bell shaped curve, indicating that the degree distribution of the nodes reached at the end of the random walk highly correlates to degree distribution of the graph.

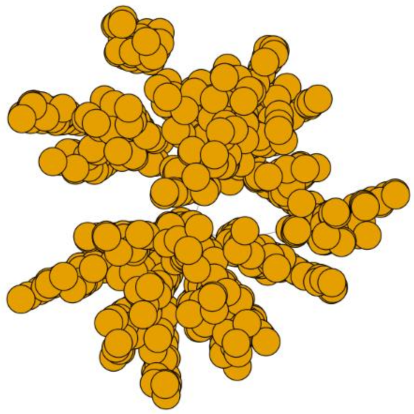
d. Now we repeat what we did in 1b for undirected random networks with 9000 nodes.
The plot are provided below:



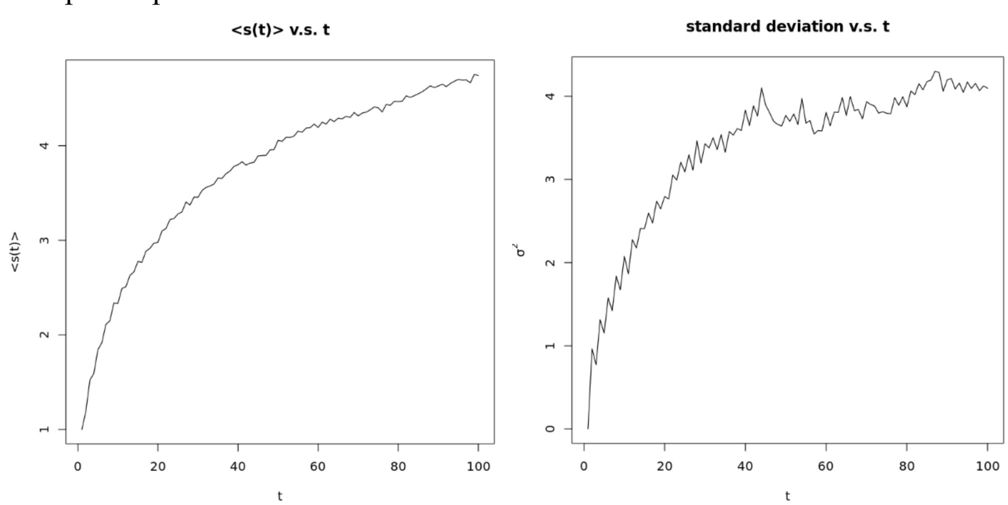


Comparing the result with a graph of 9000 nodes to the graph with 900 nodes, there is no clear difference between the two as the two graph show similar behavior. The only difference is that for the graph with 9000 nodes the mean value of the shortest path length converges to 2.2 and the standard deviation converges to around 0.14.

2. **Random Walk on Networks with Fat-Tailed Degree Distribution**
    a. We generate an undirected preferential attachment networks with 900 nodes, where each new node attaches to 1 old nodes, using *barabasi.game()* function. The plot is provided below:

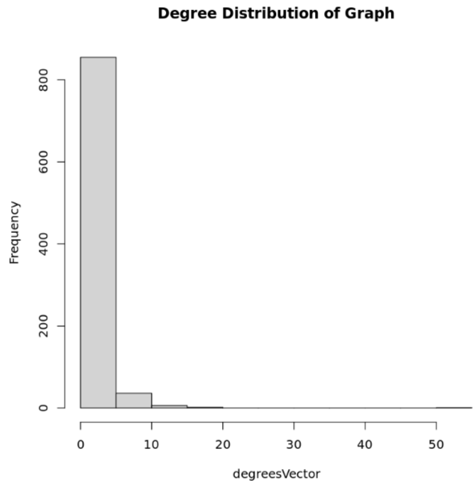


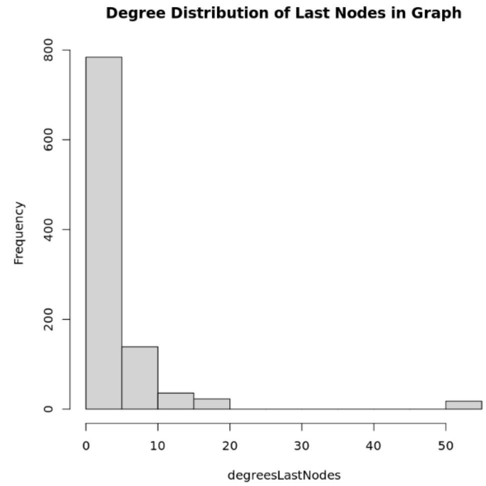
    b. We also plot the average and standard deviation of the random walk v.s. the time step *t*.
    The plot is provided below:



    The result is different from that of 1(b), as preferential attachment network is different from Erdös-Rényi Network. In the preferential attachment model, the indegree of each node adheres to a power law distribution, while the outdegree for each node is fixed at m = 1. Conversely, in the Erdös-Rényi Network, the degree distribution of nodes conforms to a normal distribution.

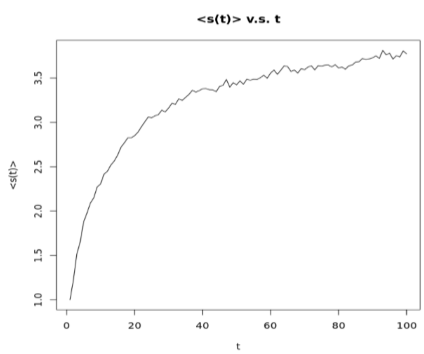
c. Similar to part 1, we will plot the degree distribution of the last nodes and compare them with the degree distribution of the graph.
The plot is provided below:

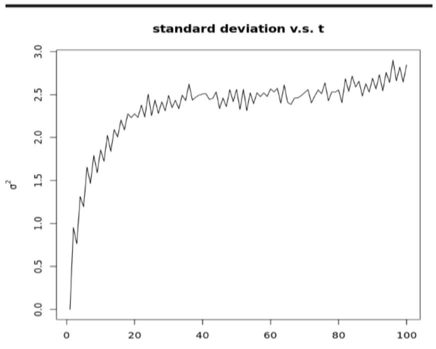




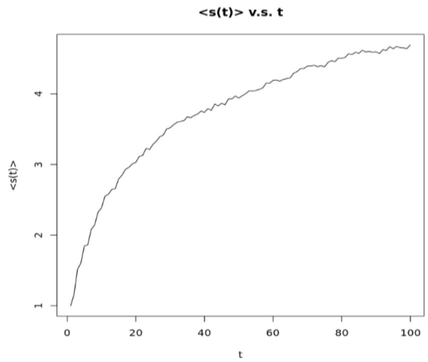
From observing the two plot, we come to similar conclusion made at 1(c), the degree distribution of the last node is related to the degree distribution of the graph. It is also important to point out that both distributions follow power law distributions.

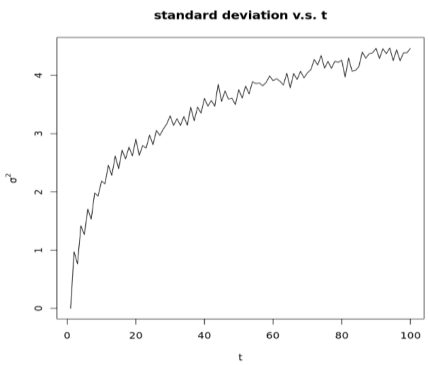
d. Now we repeat 2(b) with 90 and 9000 nodes.
The plot is provided below for a graph with 90 nodes:





The plot is provided below for a graph with 9000 nodes:

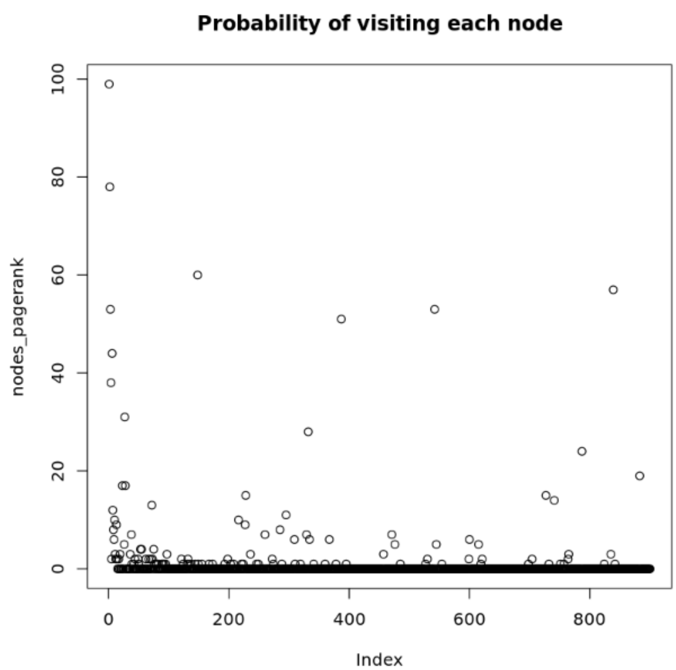


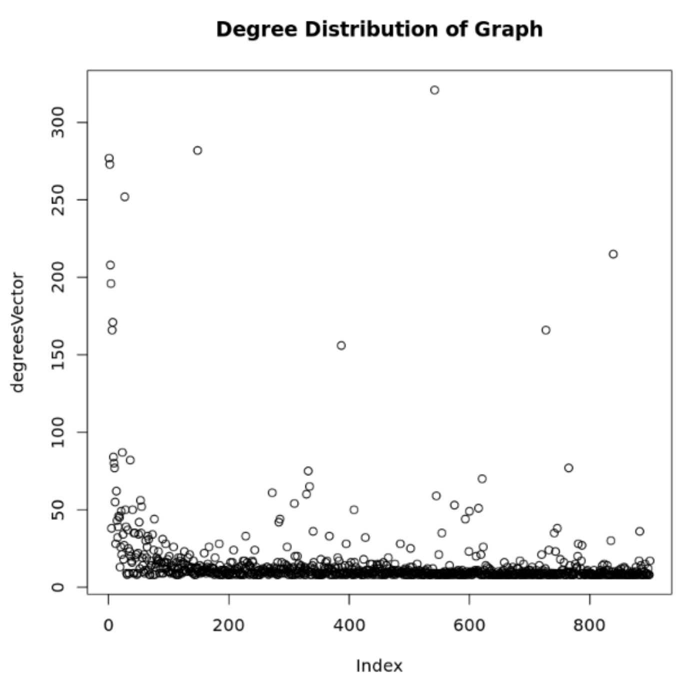


Comparing the graph with 90, 900, and finally 9000 nodes, we see the same trend and the only difference between these graphs are that the mean and the standard deviation varies slightly.

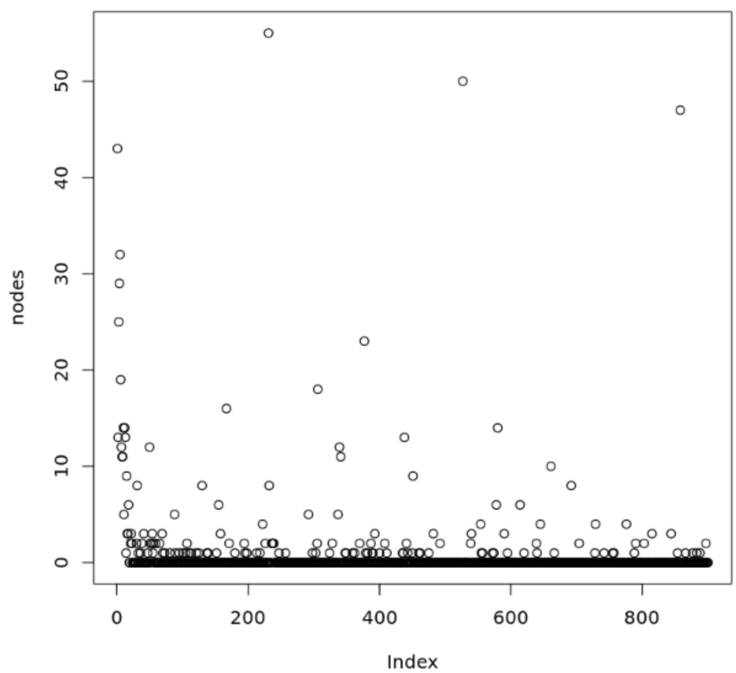
3. PageRank

a. In this section, we construct a directed random network based on the preferential attachment model, characterized by the following parameters: number of nodes = 900, and m = 4. Subsequently, we aim to determine the likelihood of a walker visiting each node and analyze how this likelihood correlates with the nodes' degrees.
The plot is provided below:



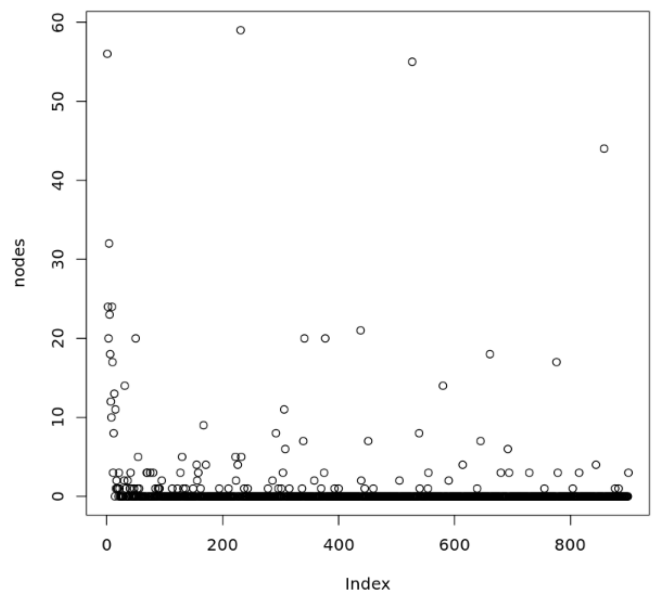


b. This part of the exercise introduces teleportation. By incorporating a teleportation probability of 0.2, we plot the probability of a walker landing on each node once more and then compare these findings with the degree distribution presented in 3(a). The resulting plot is as follows:

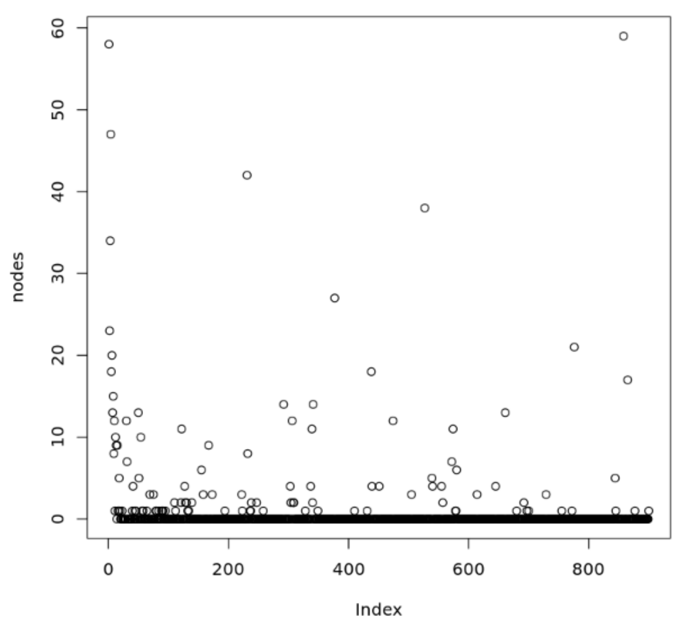


Upon examining Figures 3(a) and 3(b), it's evident that the probability distribution of a walker visiting each node closely resembles the degree distribution of the nodes within the network. Hence, the probability when employing the teleportation technique appears to be associated with the node's degree.

4. Personalized PageRank
    a. To tailor the concept of significance to each user, the probability of teleportation to any other node isn’t set uniformly at 1/N. Rather, it is determined in proportion to the PageRank values. We maintain the teleportation probability at 0.2, and the corresponding probability plot is displayed below.



    b. For this task, we initially select two nodes characterized by the median PageRank values. Subsequently, we adjust the teleportation probability function such that there's a 50% chance of visiting each of these two nodes, and zero probability for visiting any of the other nodes. Here, we also graph the walker's probability using this adjusted teleportation probability.



c. In this section, we integrate the approaches from 4(a) and 4(b) by modifying the teleportation probabilities to account for both the PageRank values and the input from trusted web pages. This involves setting the teleportation probabilities for the nodes with median PageRanks to $\frac{1}{2} * \beta$ each, while the other nodes are assigned probabilities based on the standard PageRank distribution, multiplied by the factor $1 - \beta$.