Project 1 - Random Graphs and Random Walks

Aryaman Gokarn **UID:506303588**

Mugdha Bhagwat **UID: 606297799**

Tania Rajabally **UID: 806153219**

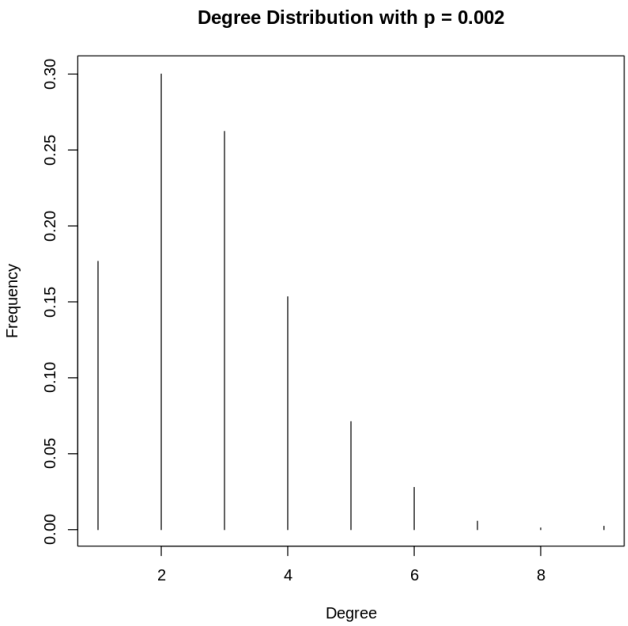
GENERATING RANDOM NETWORKS

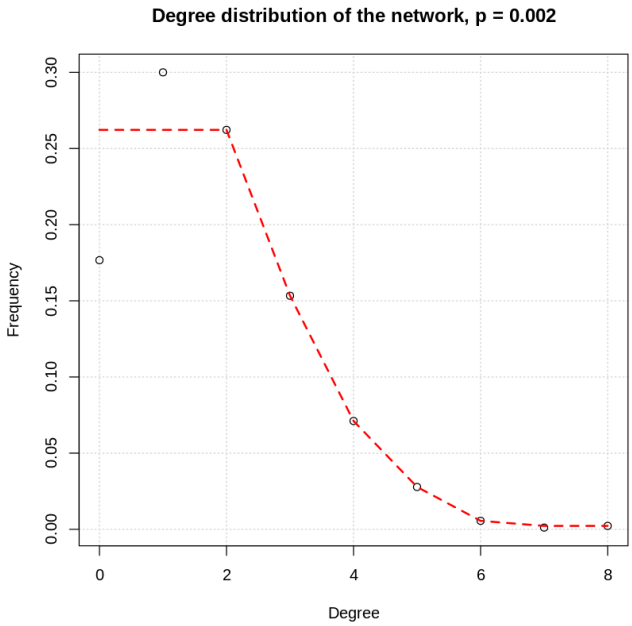
QUESTION 1:

Create random networks using Erd¨os-R´enyi (ER) model

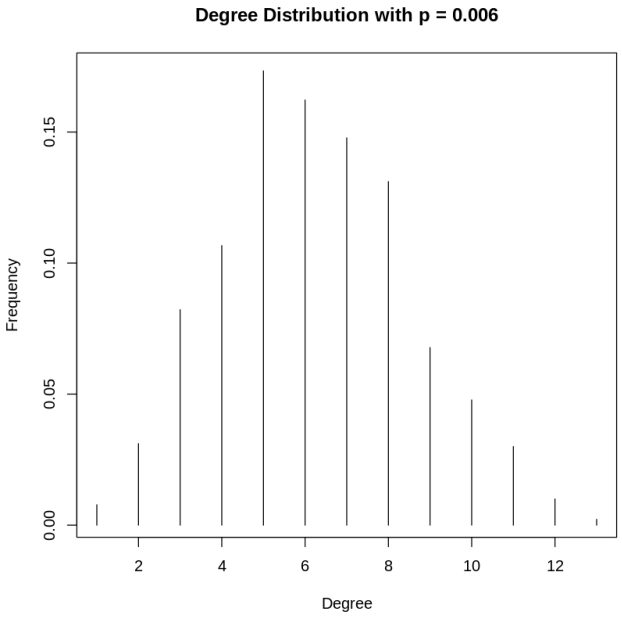
(a) Create undirected random networks with n = 900 nodes, and the probability p for drawing an edge between two arbitrary vertices 0.002, 0.006, 0.012, 0.045, and 0.1. Plot the degree distributions. What distribution (linear/exponential/gaussian/binomial or something else) is observed? Explain why. Also, report the mean and variance of the degree distributions and compare them to the theoretical values. Hint Useful function(s): samplegnp (erdos.renyi.game) , degree , degreedistribution , plot

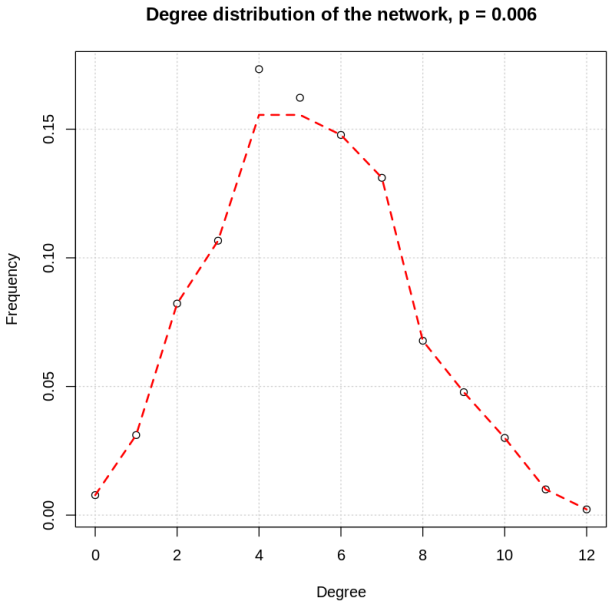
For p = 0.002 :



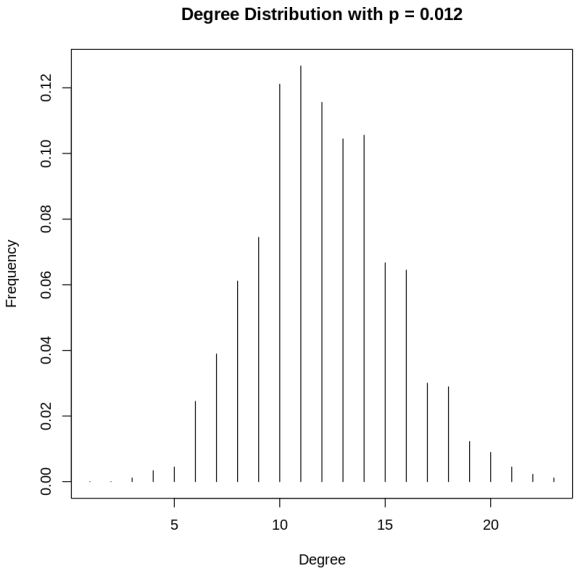


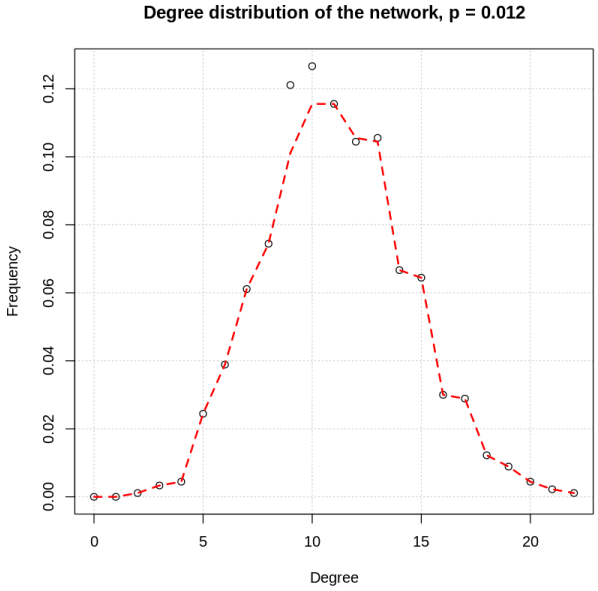
For p = 0.006 :



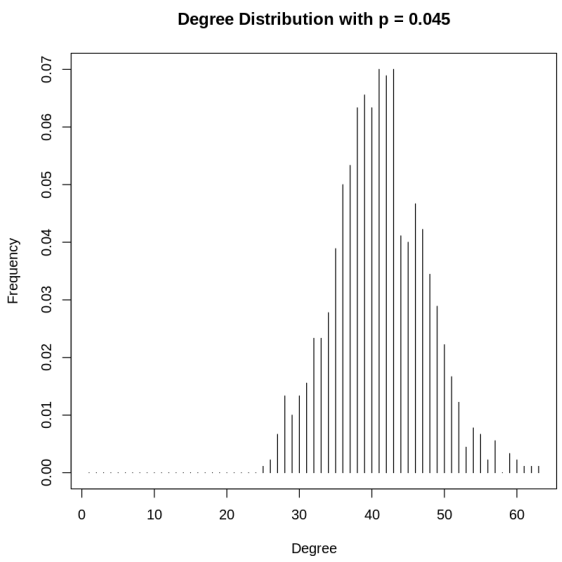


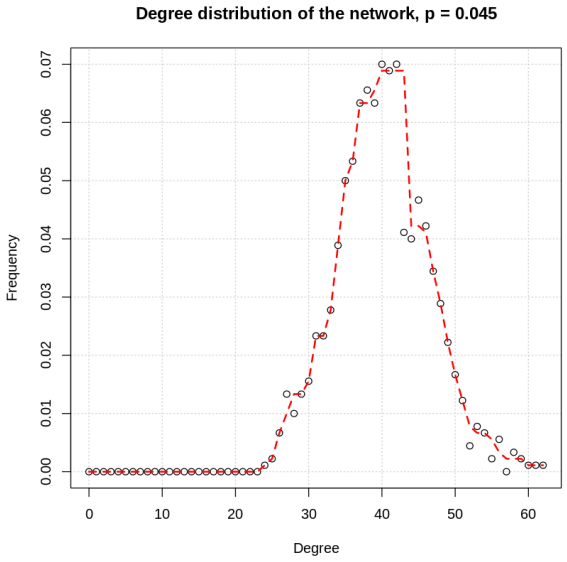
For p = 0.012 :



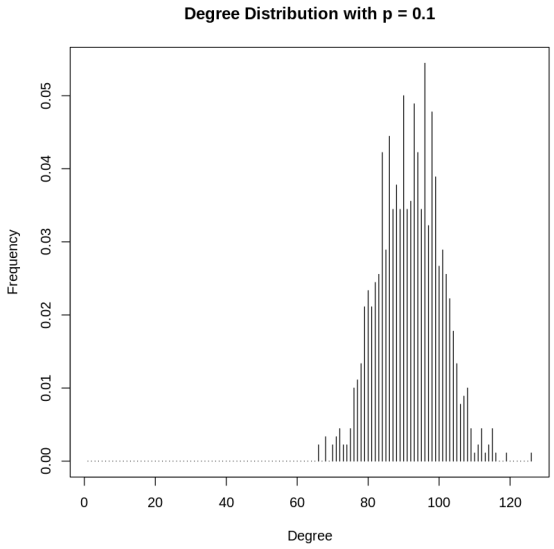


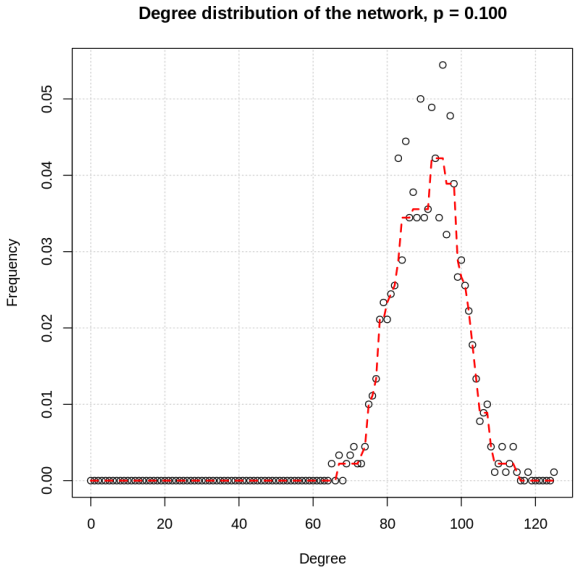
For p = 0.045 :





For p = 0.1 :





The figures above show the degree distributions. The degree distribution follows a Poisson distribution. The Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space. In the context of random networks, it describes the probability distribution of node degrees. In Erdős-Rényi random graphs, where edges are added independently with a fixed probability p, the degree distribution tends to follow a Poisson distribution when the number of nodes n is large and p is not too small. This is because each node independently forms connections with other nodes, leading to a distribution where the probability of having k connections (degree) follows a Poisson distribution. The Poisson distribution is characterized by its mean λ, which represents the average number of connections per node in the network. As n becomes large and p is not too small, the mean degree λ of the Poisson distribution is approximately equal to np. The observed degree distribution in Erdős-Rényi random networks with sufficiently large n and moderate p is Poisson, where the mean degree λ is proportional to np.

For p = 0.002 :
Mean degree = 1.766667 , Variance = 1.83871
Theoretical: Mean degree = 1.798 , Variance = 1.794404

For p = 0.006 :
Mean degree = 5.233333 , Variance = 5.249166
Theoretical: Mean degree = 5.394 , Variance = 5.361636

For p = 0.012 :
Mean degree = 11.08222 , Variance = 10.46932
Theoretical: Mean degree = 10.788 , Variance = 10.65854

For p = 0.045 :
Mean degree = 40.08222 , Variance = 39.19902
Theoretical: Mean degree = 40.455 , Variance = 38.63452

For p = 0.1 :
Mean degree = 90.78667 , Variance = 76.88436
Theoretical: Mean degree = 89.9 , Variance = 80.91

We can see that the theoretical and experimental values of the mean and variance are in close agreement with each other.

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

We obtained the values as below:

For p = 0.002 :
Probability that a network is connected = 0
Is the graph connected : FALSE
The number of nodes of the GCC are 696
The number of edges of the GCC are 798
The diameter of the GCC is 25.000000

For p = 0.006 :
Probability that a network is connected = 0.03
Is the graph connected : FALSE
The number of nodes of the GCC are 894
The number of edges of the GCC are 2417
The diameter of the GCC is 9.000000

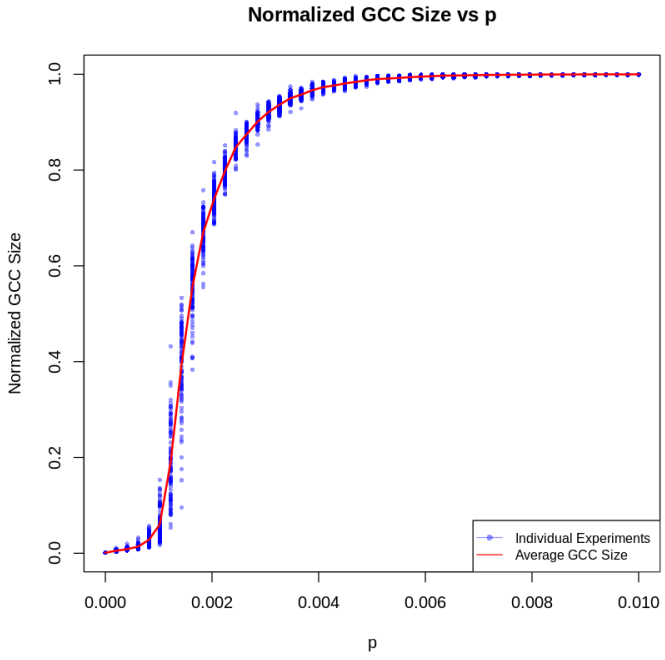
For p = 0.012 :
Probability that a network is connected = 1

Is the graph connected : TRUE
The number of nodes of the GCC are 900
The number of edges of the GCC are 4861
The diameter of the GCC is 5.000000

For p = 0.045 :
Probability that a network is connected = 1
Is the graph connected : TRUE
The number of nodes of the GCC are 900
The number of edges of the GCC are 18276
The diameter of the GCC is 3.000000

For p = 0.1 :
Probability that a network is connected = 1
Is the graph connected : TRUE
The number of nodes of the GCC are 900
The number of edges of the GCC are 40585
The diameter of the GCC is 3.000000

(c) It turns out that the normalized GCC size (i.e., the size of the GCC as a fraction of the total network size) is a highly nonlinear function of p, with interesting properties occurring for values where p = O(1/n) and p = O(ln n/n). For n = 900, sweep over values of p from 0 to a pmax that makes the network almost surely connected and create 100 random networks for each p. pmax should be roughly determined by yourself. Then scatter plot the normalized GCC sizes vs p. Plot a line of the average normalized GCC sizes for each p along with the scatter plot.



Refer to the plot above.

i. Empirically estimate the value of p where a giant connected component starts to emerge (define your criterion of "emergence")? Do they match with theoretical values mentioned or derived in lectures?

The value of p where the giant connected component starts to emerge is 0.001224.
Theoretically the value of p at which GCC starts to appear is given by

$$p_{T1} = \frac{1}{n} = \mathcal{O}\left(\frac{1}{n}\right)$$

From the above equation, we can see that for the value n=900, the value of p obtained is 0.001111. Thus we notice that the empirical value is very close to the theoretical value.

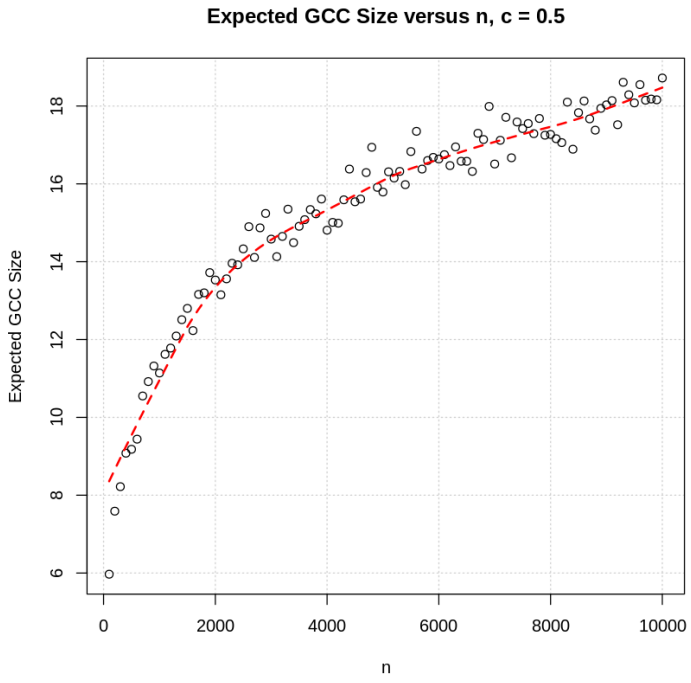
ii. Empirically estimate the value of p where the giant connected component takes up over 99% of the nodes in almost every experiment.

The value of p at which 99 percent nodes belong to GCC is 0.007551.
Theoretically, the value of p where the expected GCC is almost the same as the number of vertices is given by:

$$p_{T2} = c\frac{\ln(n)}{n} = \mathcal{O}\left(\frac{\ln(n)}{n}\right)$$

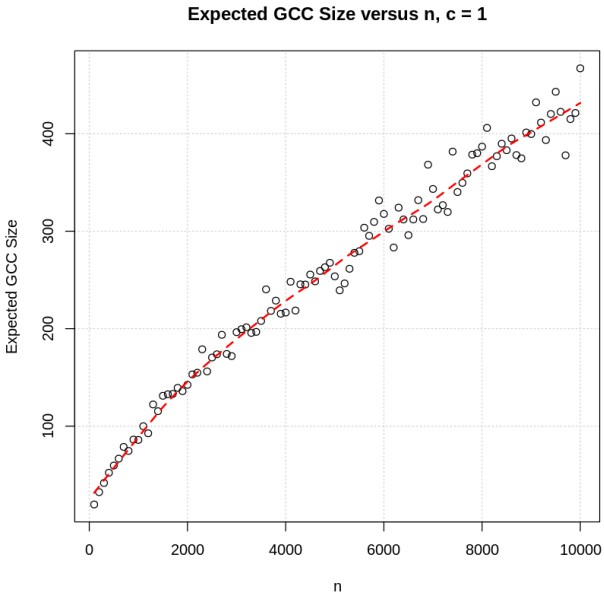
From the above equation, we obtain the theoretical value of p as 0.00755829 when c=1.00001 and n=900 as mentioned in the question.
We notice that the experimental value is very close to the theoretical value obtained.

(d) i. Define the average degree of nodes c = n × p = 0.5. Sweep over the number of nodes, n, ranging from 100 to 10000. Plot the expected size of the GCC of ER networks with n nodes and edge-formation probabilities p = c/n, as a function of n. What trend is observed?



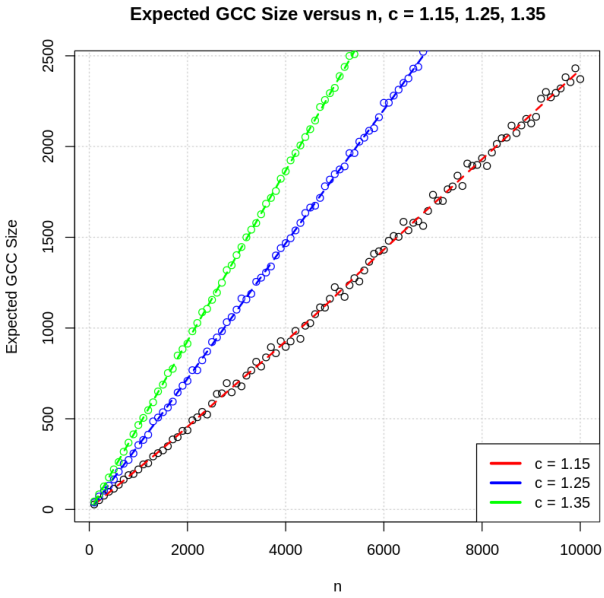
Please refer to the figure above. We see that as n increases, it converges to a constant value, following a natural logarithmic trend. As n increases, since c is constant, p decreases. If np=c<1 then there will be no connected components of size greater than O(ln(n)). For c<1, every connected component in the network has at most one cycle with the number of vertices growing in a logarithmic function.

ii. Repeat the same for c = 1.



Refer to the graph above. As expected, as the value of n increases, the expected GCC size increases. We can also see that the range of expected GCC is higher for c=1 as compared to c=0.5. This is because the expected GCC size is dependent on np. Thus, as the average degree of the network increases, the expected GCC size increases.

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.



Please refer to the figure above. We can see that as n increases, the expected GCC size increases linearly. Also, as the value of c increases, the expected GCC size increases.

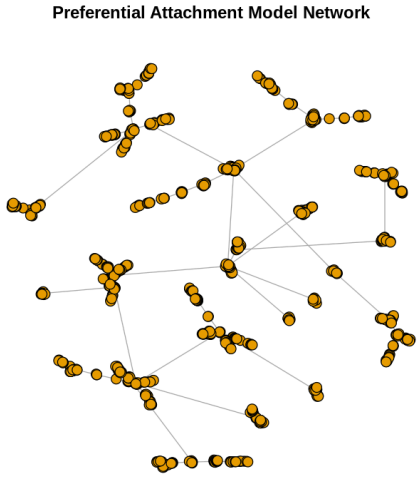
iv. What is the relation between the expected GCC size and n in each case?

We can see that as n increases, the expected GCC size increases linearly. The slope of these linear relationships increases as c increases i.e as c increases (c=np which is the average degree of nodes), it becomes more linear. This is expected because for a given constant n, if c increases, then the probability of edge formation p increases. This means that as c increases, more nodes have higher probability to connect and GCC size becomes larger.

QUESTION 2:

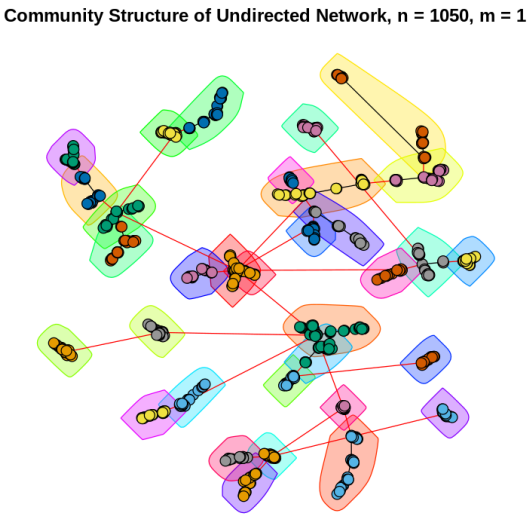
Create networks using preferential attachment model

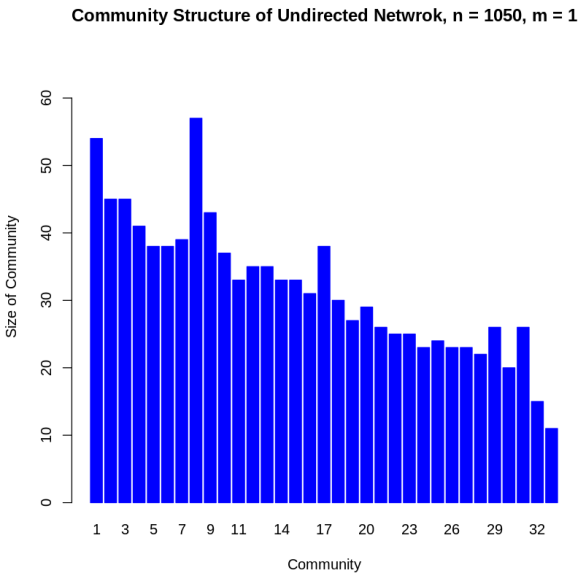
(a) Create an undirected network with n = 1050 nodes, with preferential attachment model, where each new node attaches to m = 1 old nodes. Is such a network always connected?
Hint Useful function(s): samplepa (barabasi.game)



Please refer to the figure above. Yes, such a network is always connected. We tried on 100 randomly generated undirected networks and observed that it is always connected.

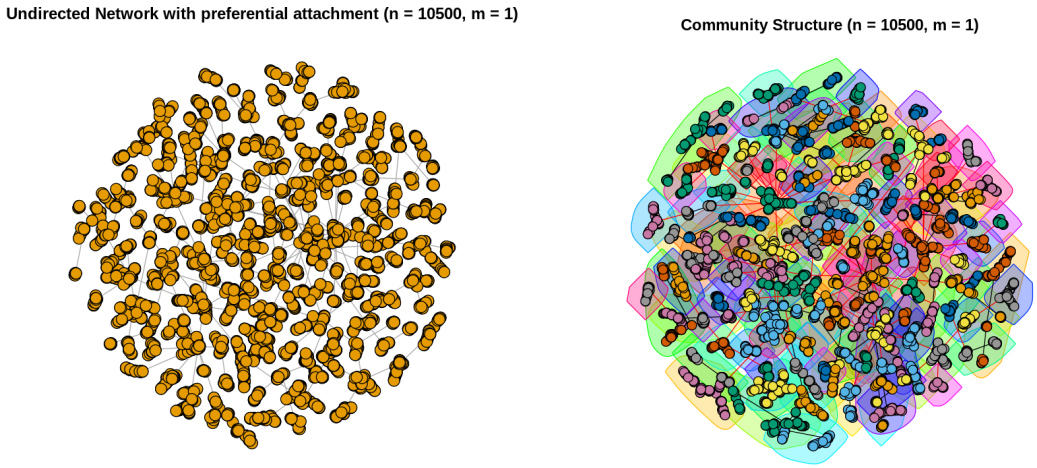
(b) Use fast greedy method to find the community structure. Measure modularity. Define Assortativity. Compute Assortativity.
Hint Useful function(s): clusterfastgreedy , modularity





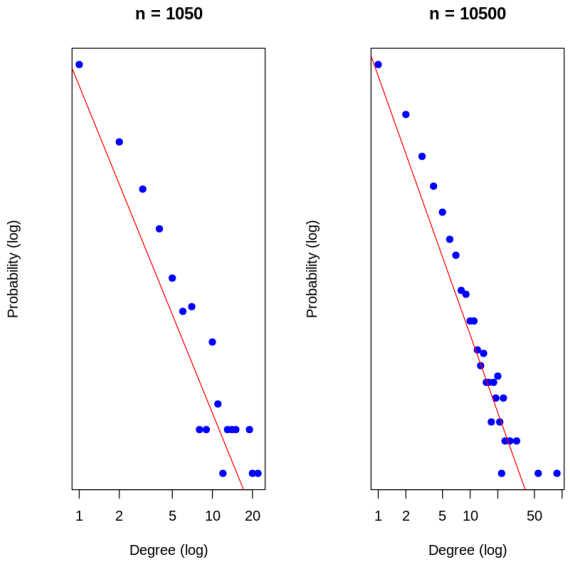
Modularity measures the strength of division of a network into communities. It quantifies the quality of the community structure by comparing the number of edges within communities to the expected number of edges in a random network with the same node degrees.
Assortativity measures the preference of a network's nodes to attach to others that are similar or dissimilar in some way. In an assortative network, nodes tend to connect to others with similar properties, while in a disassortative network, nodes tend to connect to others with dissimilar properties.
For the above network, we find the below values:
Modularity: 0.9359115
Degree Assortativity: -0.1136649

(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?



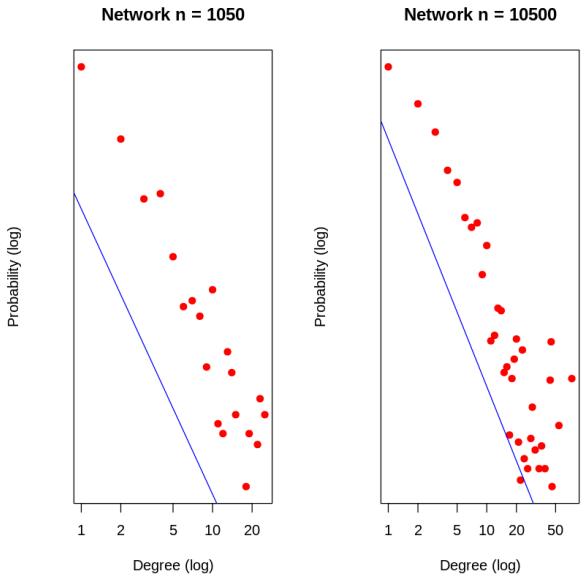
Refer to the network above. For this network, we obtain the values as below:
Modularity of Larger Network: 0.9789827
Assortativity of Larger Network (Degree): -0.03337665
We can see that there are more communities in a network with a larger number of nodes as compared to the network with a smaller number of nodes. The number of vertices are also higher in the larger network. Apart from this, the modularity index is also higher indicating that the larger network is better capable of being partitioned into communities with strong intra community connectedness and sparse intercommunity connectedness. The probability of strong intra community is higher for larger networks and the likeness of sparsity is also higher for larger networks.

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



Please refer to the graphs above.
Estimated slope for n = 1050 : -2.249561
Estimated slope for n = 10500 : -2.40196
We can see that larger networks follow power law distribution more closely than smaller ones.

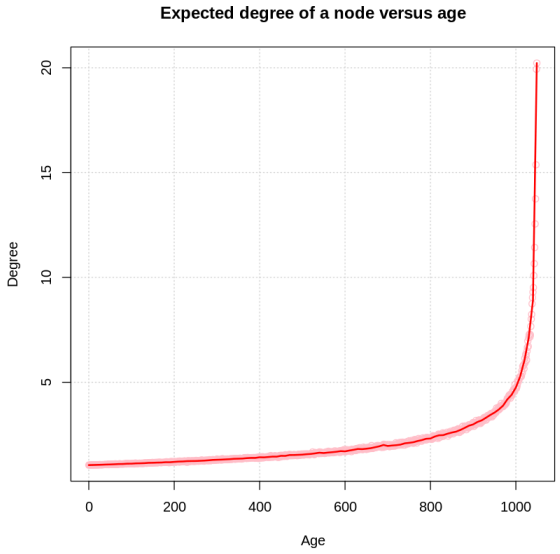
(e) In the two networks generated in 2(a) and 2(c), perform the following:
Randomly pick a node i, and then randomly pick a neighbor j of that node. Plot the degree distribution of nodes j that are picked with this process, in the log-log scale. Is the distribution linear in the log-log scale? If so,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



Refer to the plots above. We can see that both the distributions are linear. The slopes are:
Estimated slope for neighbors in Network n = 1050 : -1.194989
Estimated slope for neighbors in Network n = 10500 : -1.442518

Random sampling of only one neighbor does not follow the node degree distribution. It can be observed that although the slope is negative and decreases and n decreases, the degree distribution of the random node selection does not follow the power law distribution of the Barabasi Model having an exponent of 3. Since we are sampling one node at random, both the expected degree and variance are unbounded. Thus, the degree distribution will change more slowly compared to node degree distribution where the gradient is in the range of 2 to 3 and the expected degree is bounded.

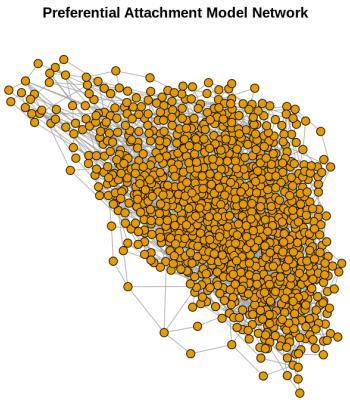
(f) Estimate the expected degree of a node that is added at time step i for 1 ≤ i ≤ 1050. Show the relationship between the age of nodes and their expected degree through an appropriate plot. Note that the newest added node is the youngest.



Refer to the figure above. We can see that the expected degree of a node increases monotonically as its age increases. This is expected because in preferential ttachment models, nodes with higher degrees or connectedness are more likely to receive edges with newer nodes. Since older nodes are more likely to be connected to newer nodes with time, the degree of older nodes increases with time.

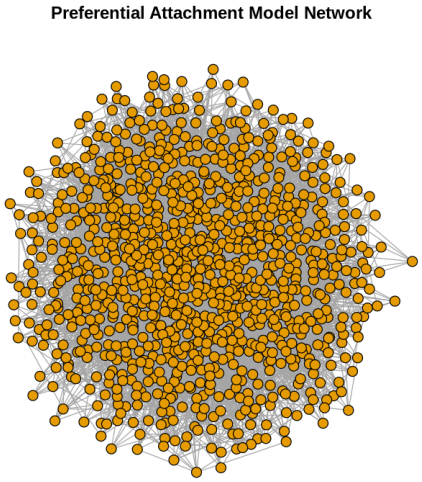
(g) Repeat the previous parts (a-f) for m = 2, and m = 6. Compare the results of each part for different values of m.

For m=2:



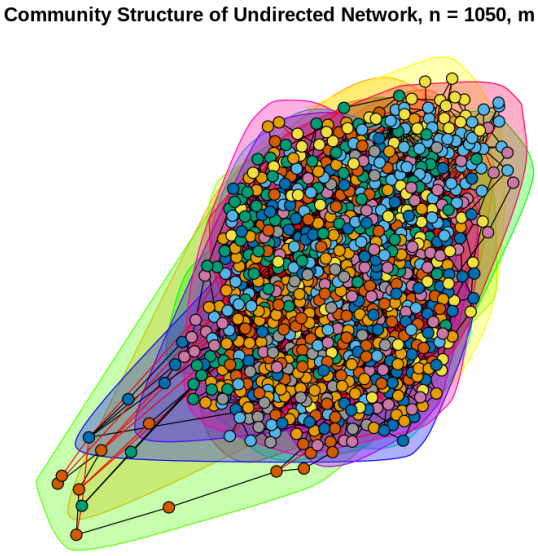
The network is connected. We tried for 100 random networks and all were connected.

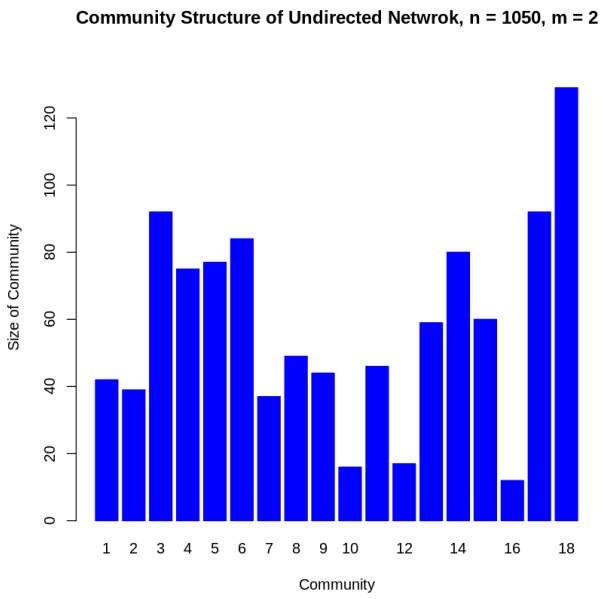
For m=6:



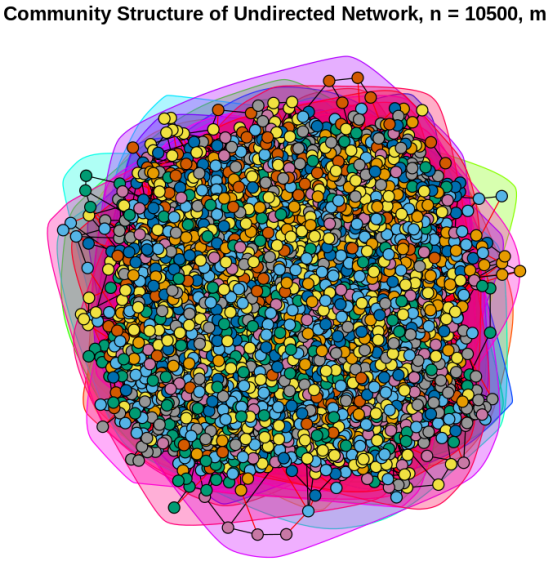
The network is connected. We tried for 100 random networks and all were connected.

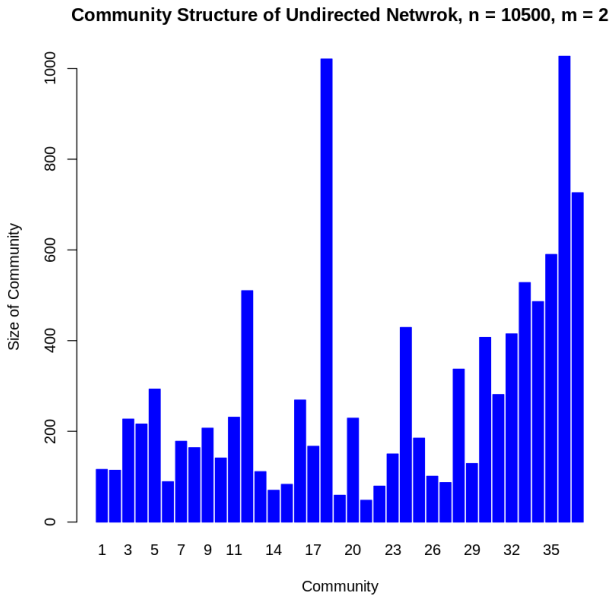
For m=2:





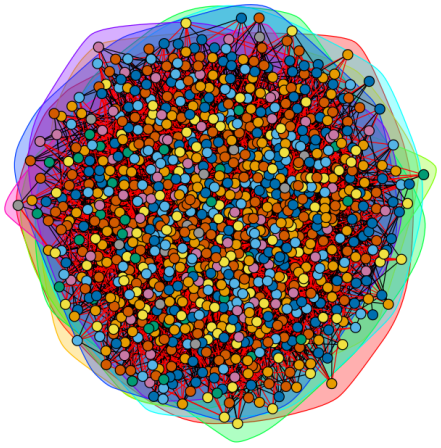
Using the fast greedy method, we get the above community structure for n=1050 and m=2. We get the below values:
Modularity: 0.5217184
Degree Assortativity: -0.04428939

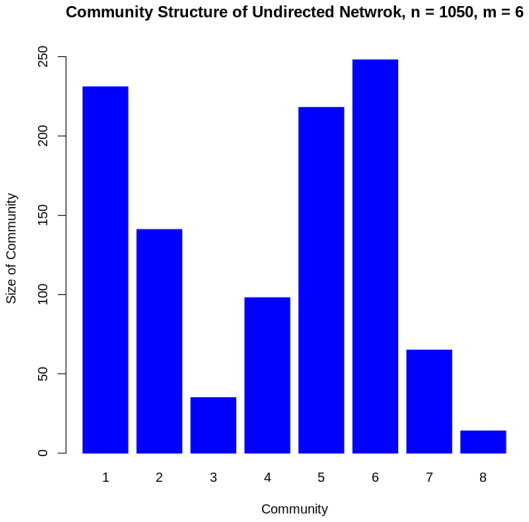




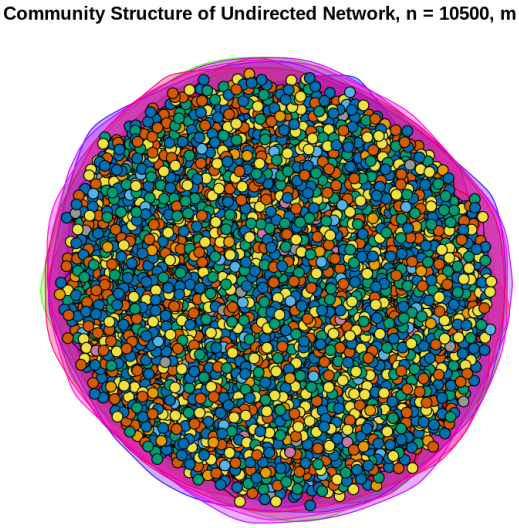
Using the fast greedy method, we get the above community structure for n=10500 and m=2. We get the below values:
Modularity of Larger Network: 0.5322066
Assortativity of Larger Network (Degree): -0.00721725

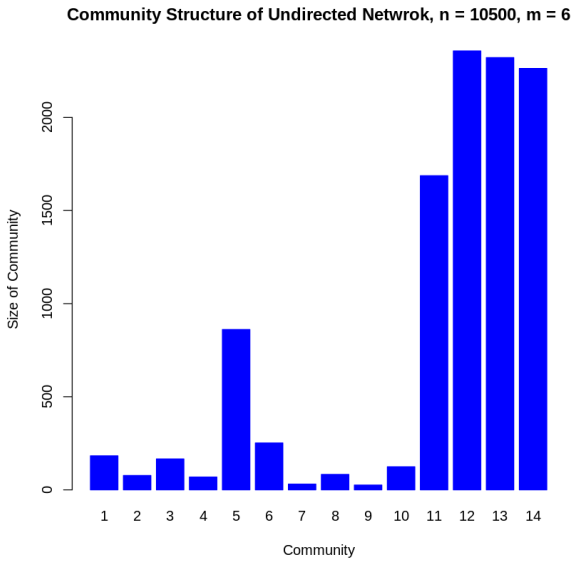
For m=6:





Using the fast greedy method, we get the above community structure for n=1050 and m=6. We get the below values:
Modularity: 0.2520199
Degree Assortativity: -0.02376997

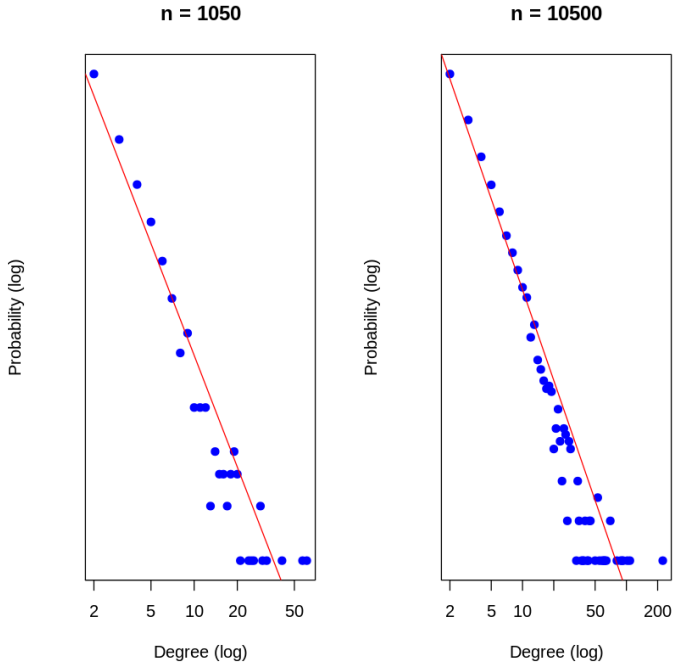




Using the fast greedy method, we get the above community structure for n=10500 and m=6. We get the below values:
Modularity of Larger Network: 0.2431556
Assortativity of Larger Network (Degree): 0.002164367

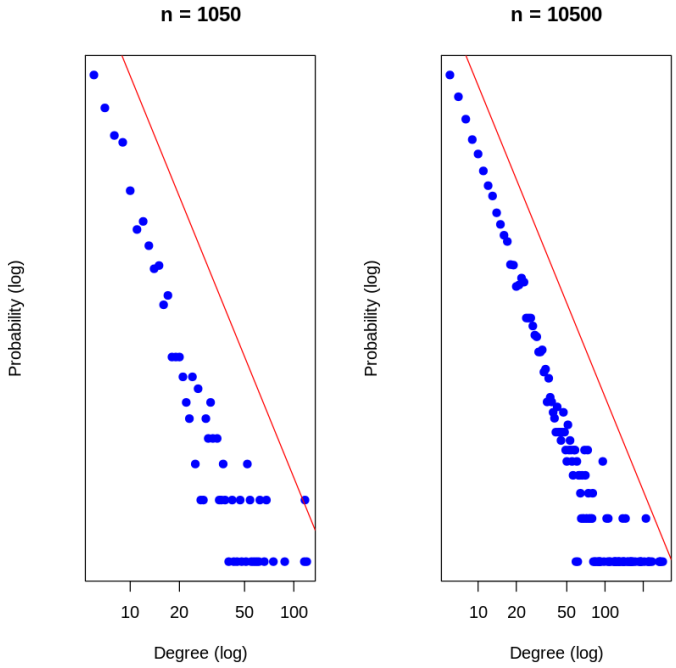
We see that there are more communities in the network with a higher number of nodes (10500) compared to the smaller network (1050 nodes). In addition, the number of vertices in each community is significantly higher as well.
We observe that as the value of m increases, the modularity index drops and the number of communities drop significantly, with a few communities having a large number of vertices. Intuitively, a higher value of m indicates that an incoming node is connected to a larger number of older nodes. While this should result in strong intra-community connectedness, the global sparsity among different communities is lost due to the connectedness requirement brought on by high values of , resulting in edges being formed among otherwise distinct clusters and hence weakening the community structures.

For m=2:



Above is the degree distribution in a log-log scale for both n = 1050, 10500. We get the slopes as below:
Estimated slope for n = 1050 : -2.048647
Estimated slope for n = 10500 : -2.281481

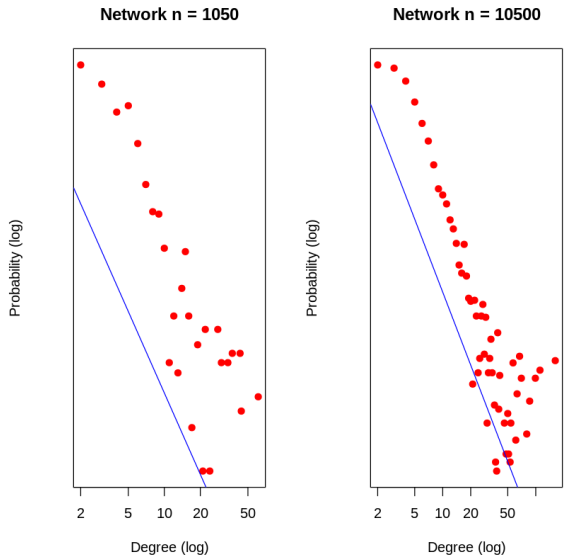
For m=6:



Above is the degree distribution in a log-log scale for both n = 1050, 10500. We get the slopes as below:
Estimated slope for n = 1050 : -1.963338
Estimated slope for n = 10500 : -2.173422

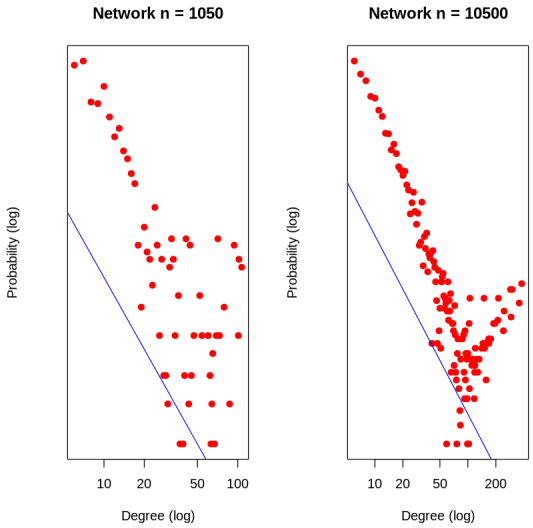
The trends are roughly linear in the log-log scale. A larger network has more vertices to better approximate the power-law degree distribution for a preferential attachment model probabilistically over the smaller networks. Since a large network has a higher number of intra-community connections, the probability that the degree of a randomly chosen nodes being bigger for large networks over small networks is high as well. We observe that as increases, the negative slope of the linear regression line decreases. This means that the degree distribution deviates from the preferential attachment model when incoming nodes are added to a larger number of older vertices. One way to explain this is that as increases, the expected degree of a node for the same degree increases.

For m=2:



Above is the plot of the degree distribution of nodes j that are picked with this process, in the log-log scale.
Estimated slope for neighbors in Network n = 1050 : -1.096612
Estimated slope for neighbors in Network n = 10500 : -1.379728

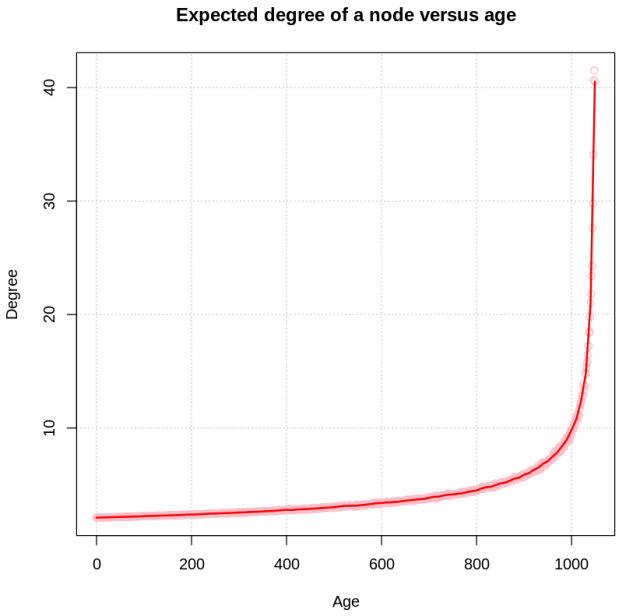
For m=6:



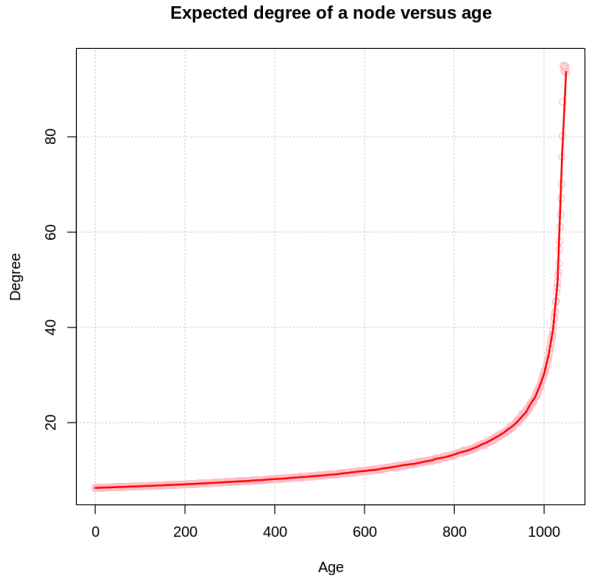
Above is the plot of the degree distribution of nodes j that are picked with this process, in the log-log scale.
Estimated slope for neighbors in Network n = 1050 : -1.050284
Estimated slope for neighbors in Network n = 10500 : -1.189399

The distributions are roughly linear. : We observe that as m increases, the negative slope of the linear regression line decreases. This means that the degree distribution deviates from the preferential attachment model when incoming nodes are added to a larger number of older vertices. The explanation is similar as for the overall degree distribution. As m increases, the expected degree of a randomly selected node for the same degree increases.

For m=2:



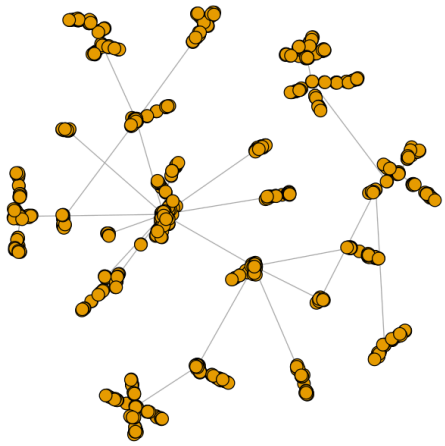
For m=6:



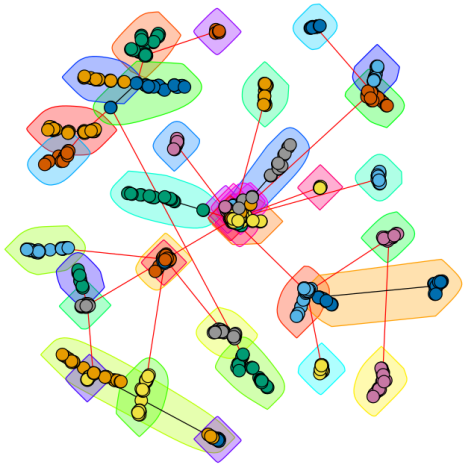
Above is the plot which shows the relationship between the age of nodes and their expected degree. For all values of m, we see that the expected degree of a node increases monotonically as it gets older. This is expected because in preferential attachment models, nodes with higher degrees or connectedness are more likely to receive edges with newer nodes. Since older nodes are more likely to be connected with newer nodes with time, the degree of older nodes increases with each timestep. As m increases, the expected degree for the same node for the same age increases.

(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 random power-law networks.
Hint In case that fastgreedy community detection fails because of self-loops, you may use "walktrap" community detection.
Useful function(s): sampledegseq

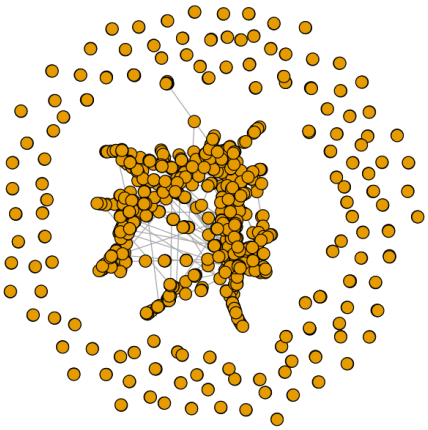
Original Undirected network with preferential attachment



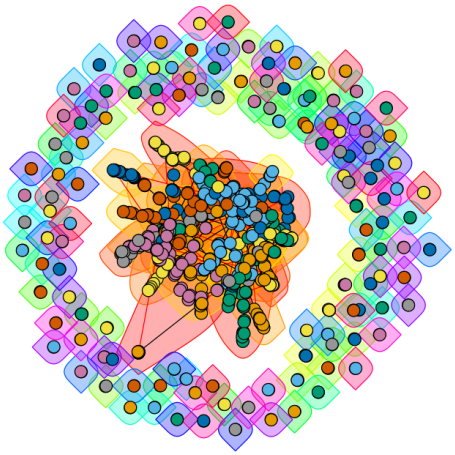
Original Community Structure



New Undirected network with same Degree Sequence



New Community Structure (same Degree Sequence)



Refer to the networks above.
The modularities are as below:
Modularity of the original community structure is 0.928853
Modularity of the new community structure is 0.836678
The simple, no multiple method avoids multiple and loop edges and restarts the generation from scratch if loops are detected. Fast greedy community detection works on this method but the method does not guarantee uniform sampling from the space of all possible graphs. In stub-matching from degree sequences, we randomly sample numbers from the degree distribution of our
choice, e.g. power law degree distribution to get the nodes of the network along with their degrees. The value of the ith sample is the degree of the ith node. Afterwards, we randomly match the stubs to connect the vertices of the network by edges. we observe that a significant number of vertices for the network generated via stub-matching are unconnected and forming their own communities, while the network generated via preferential attachment model show local regions of high-connectedness with sparse connections among individual communities. The number of communities are also significantly higher for stub-matching over preferential attachment. This is reflected in the modularity indices of the networks, with preferential attachment models having a higher modularity than stub-matching models, indicating that the preferential attachment network is better capable of being partitioned into communities with strong intra-community connectedness and sparse inter-community connectedness over stub-matching networks. In addition, since each new node is always attached to a connected node in the existing network for preferential attachment models, the final network is always guaranteed to be connected by construction. However, this is not the case for stub-matching from degree sequence, as the process is random and does not necessarily require connection to older nodes. Furthermore, stub-matching can lead to the formation of loops, which can make community detection difficult using greedy procedures. The only advantage of stub-matching is the level of control over the degree sequence that the user has, allowing the user to generate networks with custom degree distributions.

QUESTION 3:

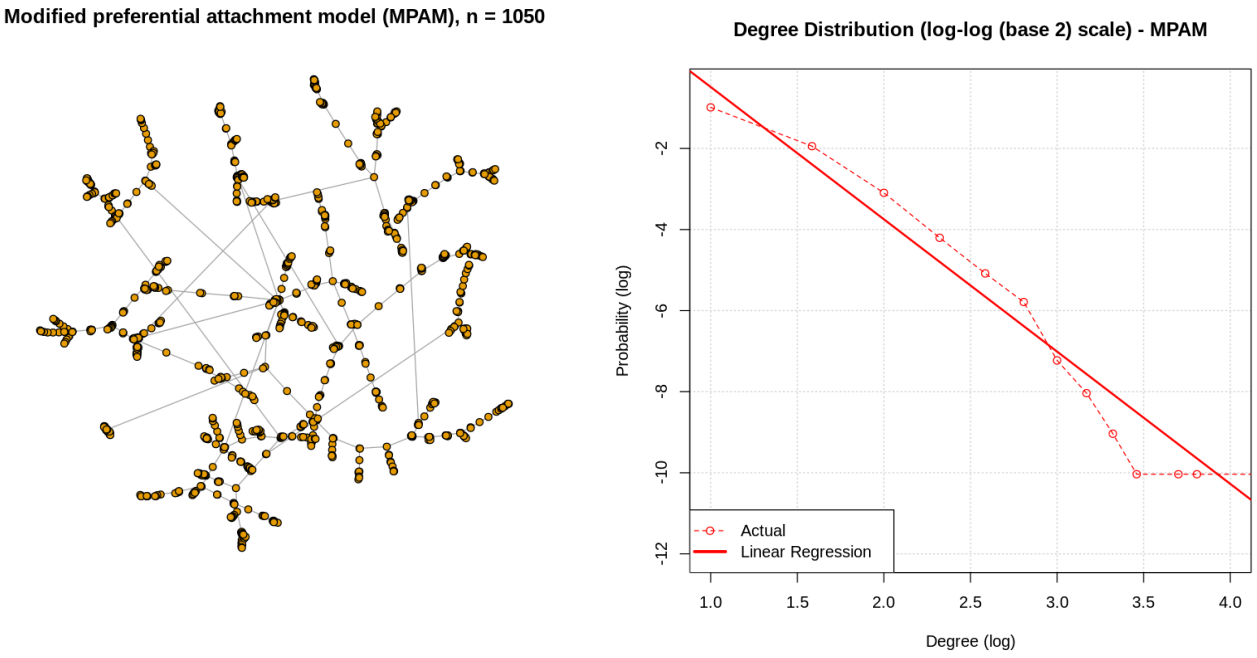
Create a modified preferential attachment model that penalizes the age of a node

(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. In particular, the probability that a newly added vertex connects to an old vertex is proportional to:

P [i] ~ (ckiα + a)(dliβ + b),

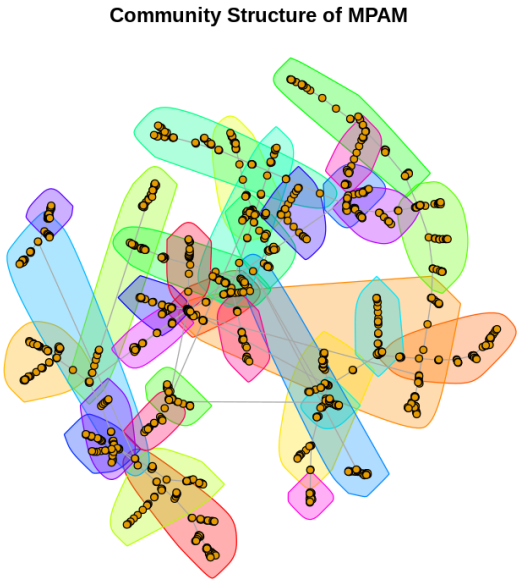
where ki is the degree of vertex i in the current time step, and li is the age of vertex i. Produce such an undirected network with 1050 nodes and parameters m = 1, α=1,β=−1,anda=c=d=1,b=0. Plot The Degree Distribution. What Is The power law exponent?

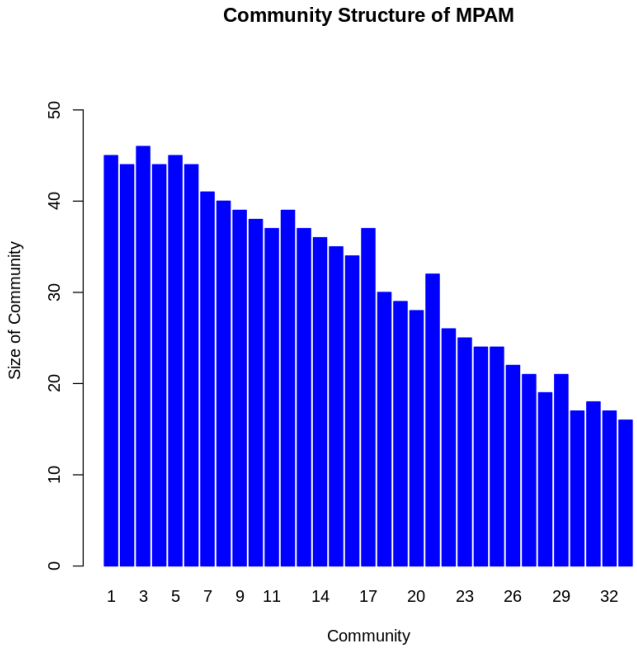
Hint Useful function(s): samplepaage



Refer to the plot above. The slope of the plot is -3.264. Therefore the power law exponent is 3.264. We observe that the degree distribution is much more linear than vanilla preferential attachment models in the log-log scale, with a lower value on the upper bound of the expected degree of the node. This means that a modified preferential attachment model gives rise to a sparse network with a finite number of high-degree nodes and a degree-distribution tail that is not as heavy as vanilla preferential attachment models.

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





Refer to the plots above. The modularity is 0.936313216727357. We see that the number of vertices in each community are much more homogeneous for the modified model over vanilla preferential attachment networks. This is expected as the modified model penalizes the network generator from selecting nodes that are too old and encourages attachment of incoming nodes to moderately aged nodes, yielding clusters with homogenous number of nodes.

RANDOM WALK ON NETWORKS

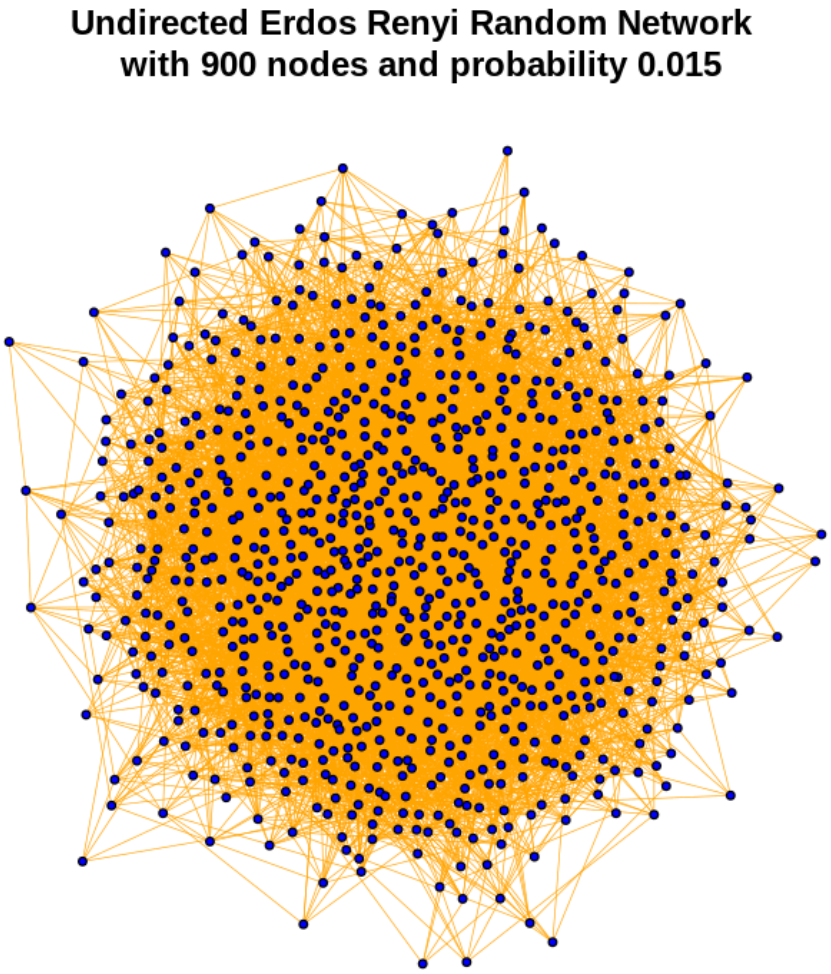
QUESTION 1:

Random walk on Erdo ̈s-R ́enyi networks

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

Using the erdos.renyi.game library we have created an Undirected Erdos Renyi Random Network G(n, p) where n = 900 and p = 0.015.

```
IGRAPH 079ff45 U--- 900 6039 -- Erdos-Renyi (gnp) graph
+ attr: name (g/c), type (g/c), loops (g/l), p (g/n)
```



The above figure shows an undirected random network with 900 nodes and the probability of drawing an edge between any two nodes as 0.015.

(b) Let a random walker start from a randomly selected node (no teleportation). We use t to denote the number of steps that the walker has taken. Measure the average distance (defined as the shortest path length) ⟨s(t)⟩ of the walker from his starting point at step t. Also, measure the variance σ2(t) = ⟨(s(t) − ⟨s(t)⟩)2⟩ of this distance. Plot ⟨s(t)⟩ v.s. t and σ2(t) v.s. t. Here, the average ⟨·⟩ is over random choices of the starting nodes.

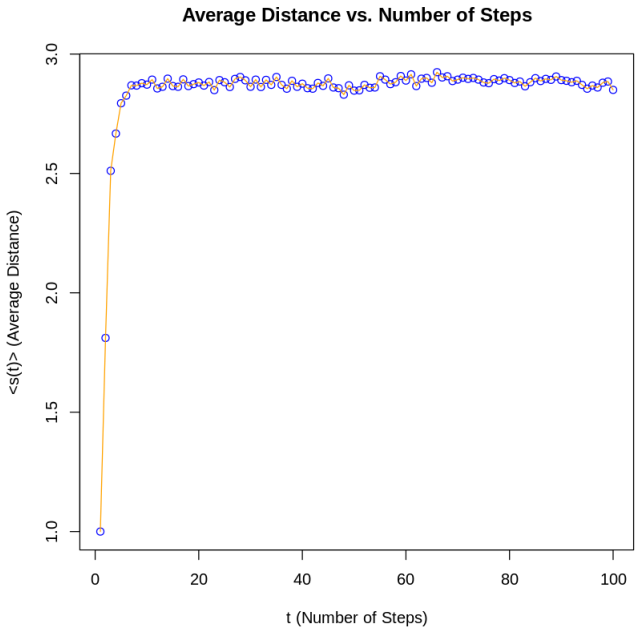
As given in the above question we implemented the following:

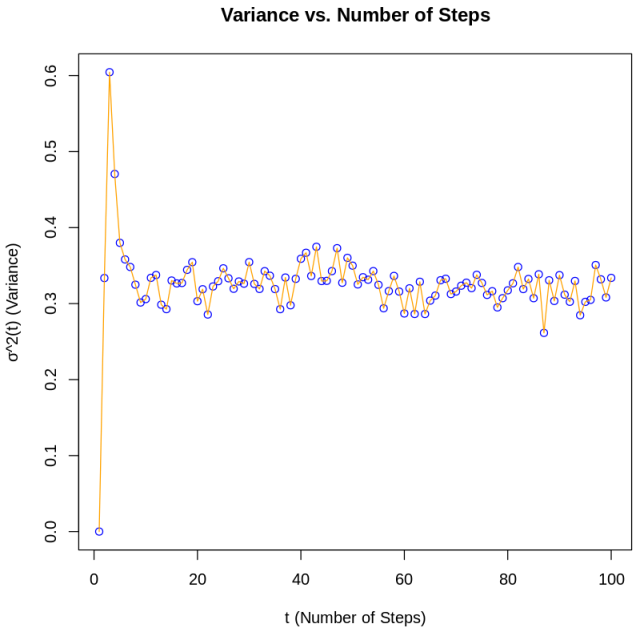
1. We created an undirected random network with 900 nodes and the probability of drawing an edge between any two nodes as 0.015. G(n = 900, p = 0.015)
2. We started a random walk on a randomly selected node in this GCC (Giant Connected Component).
3. We calculated the shortest distance s(t) for each step t from the starting node.
4. We averaged the distance and the variance over 1000 such random walks.
5. The above experiments were conducted with variable values of steps = 100, 200.

Summary of the results:

Random walk on G(n = 900, p = 0.015) with T = 100 and Iterations = 1000

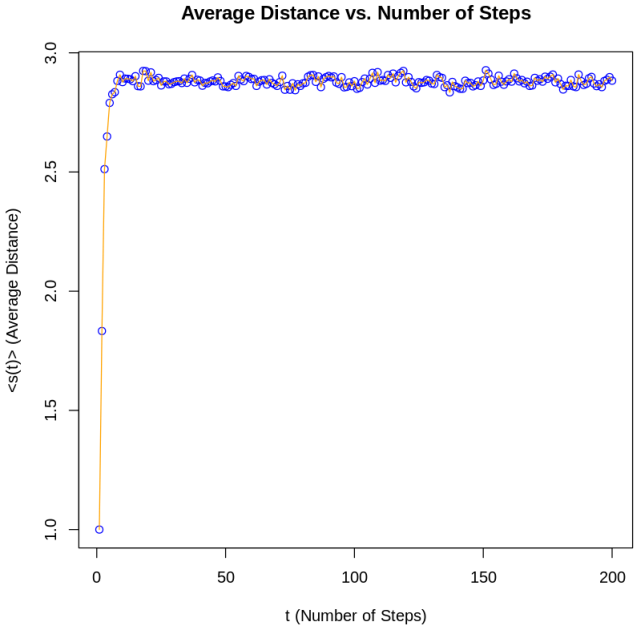
We can see from the graphs below that: ⟨s(t)⟩ and $\sigma^2(t)$ both reach a steady state around t = 10.

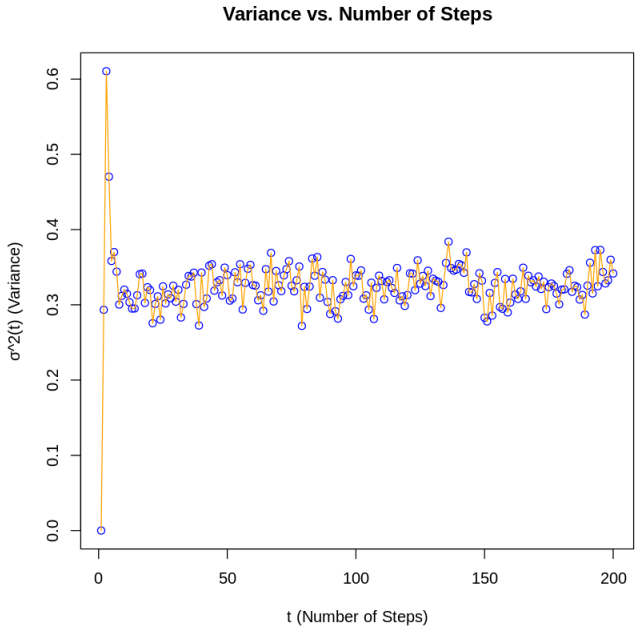




Random walk on G(n = 900, p = 0.015) with T = 200 and Iterations = 1000

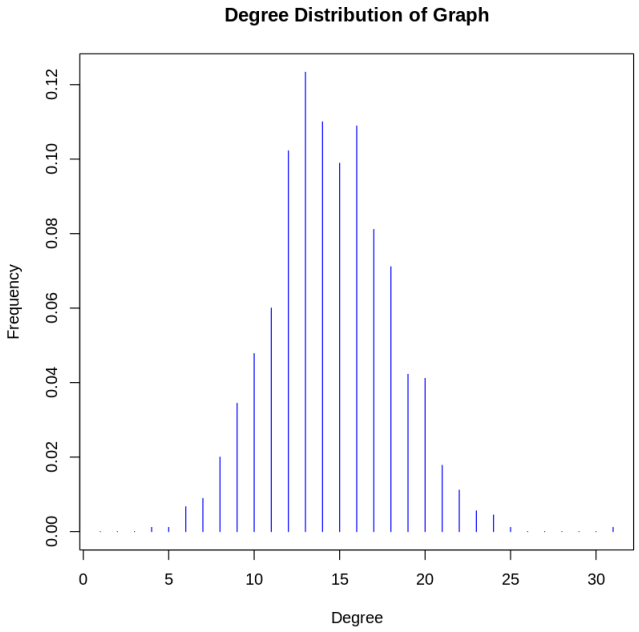
We can see from the graphs below that: $\langle s(t) \rangle$ and $\sigma^2(t)$ both reach a steady state around t = 10.

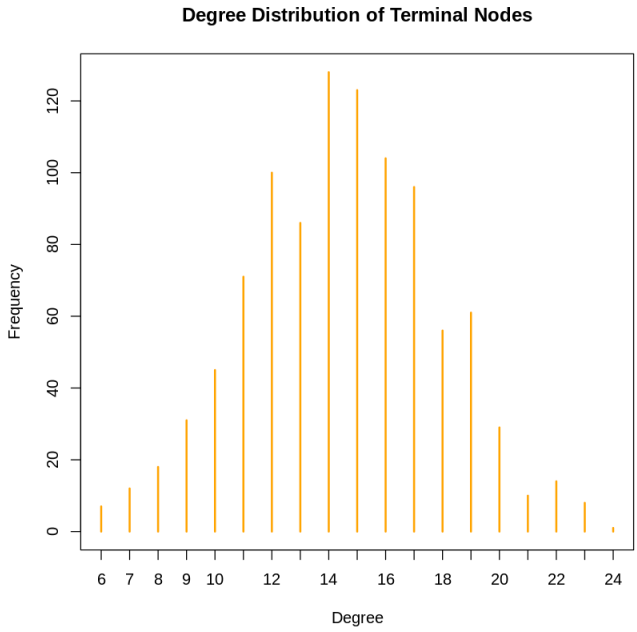


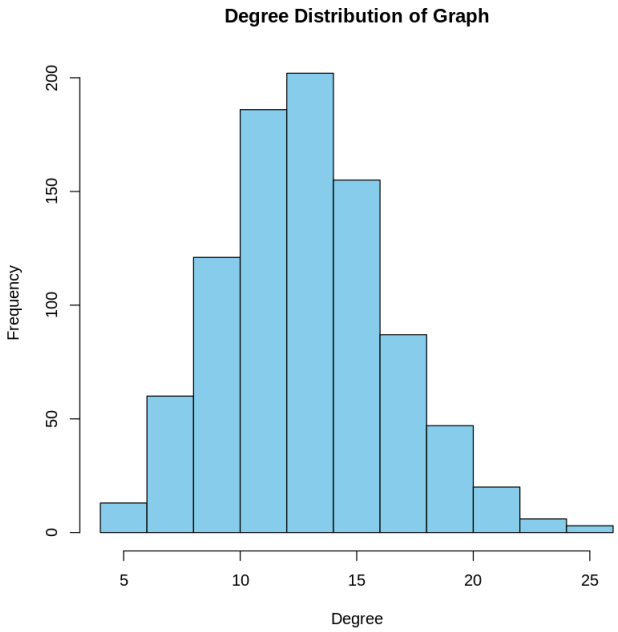


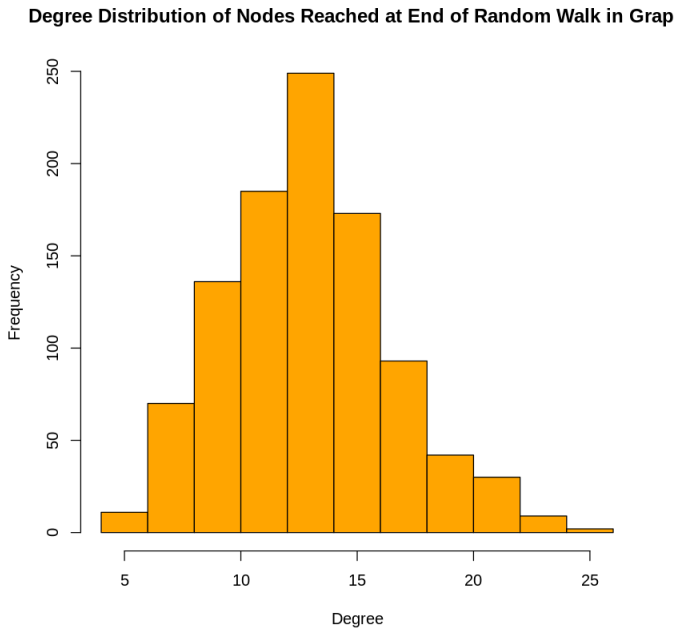
(c) Measure the degree distribution of the nodes reached at the end of the random walk. How does it compare to the degree distribution of graph?

We have plotted the degree distribution of the Original Graph and of the Nodes reached at the end of the Random Walk for G(n = 900, p = 0.015) and T = 100 for 1000 random walks.







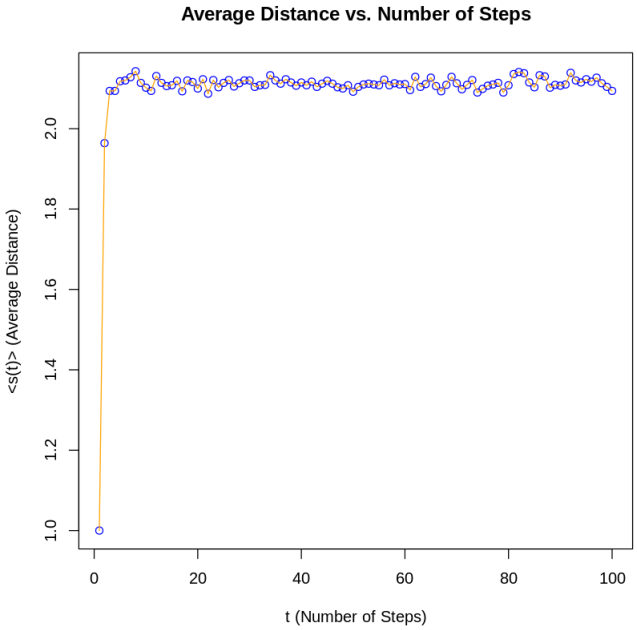


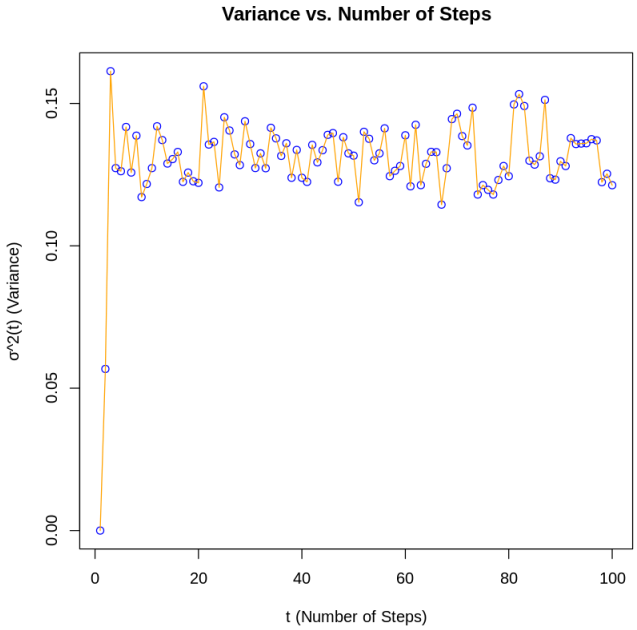
We observe that both the original graph and the graph for the distribution at the end of the random walk follow a similar binomial distribution. This results align with what is expected for E-R networks.

(d) Repeat 1(b) for undirected random networks with 9000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

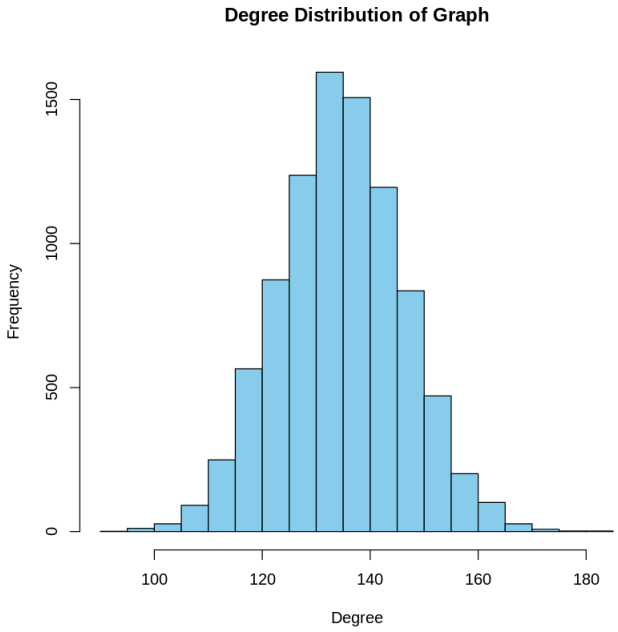
Random walk on G(n = 900, p = 0.015) with T = 100 and Iterations = 1000

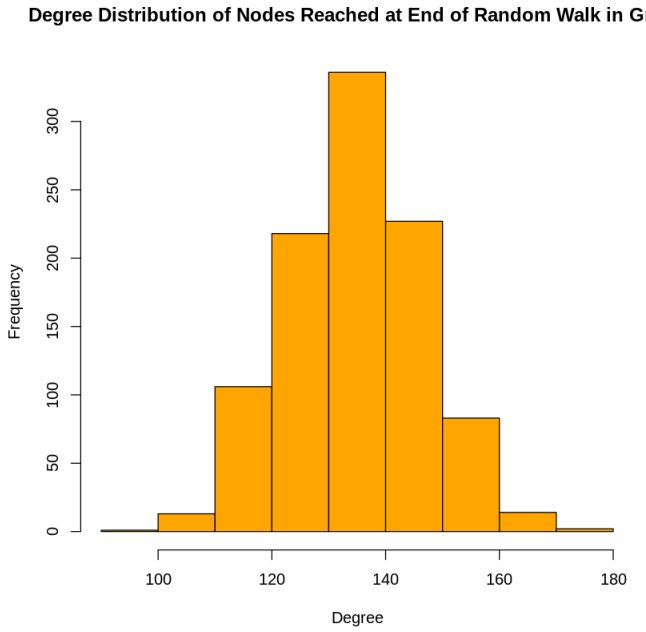
We can see from the graphs below that: $\langle s(t) \rangle$ and $\sigma^2(t)$ both reach a steady state around t = 5.

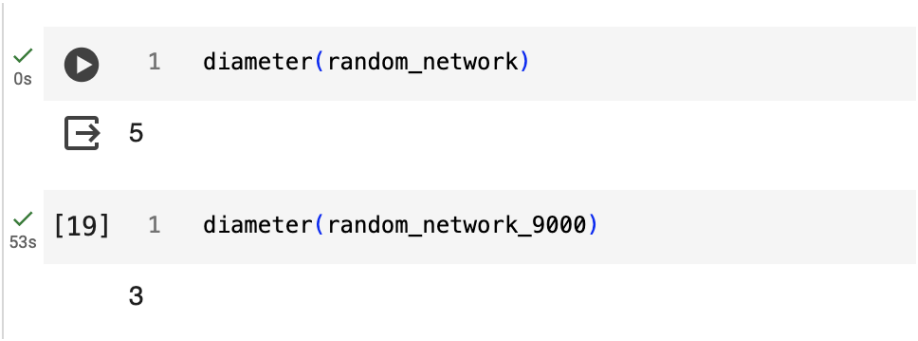




We have plotted the degree distribution of the Original Graph and of the Nodes reached at the end of the Random Walk for G(n = 9000, p = 0.015) and T = 100 for 1000 random walks.







As the number of nodes (n) in the network increases, two key observations are made:

1. **Decrease in Number of Steps to Reach Steady State**
   With the increase in n from 900 to 9000, the number of steps required for the random walk to reach a steady state t decreases. This suggests that larger networks converge to a steady state sooner compared to smaller networks.

2. **Decrease in Spread of Shortest Distances and Variance**
   Additionally, at the point of convergence, both the average shortest distance and the variance of shortest distances decrease. This decrease implies that, upon convergence, there is less variation in the shortest distances from any node to the starting node.

Qualitative Explanation and the Role of the Network Diameter:

1. Although it might seem counterintuitive that larger networks converge more quickly and with less spread in shortest distances, this observation can be understood by considering the network's diameter.
2. Despite the increase in n from 900 to 9000, the diameter of the network decreases from 5 to 3.
3. In a network with a smaller diameter, random walks are more likely to reach any node from any other node relatively quickly. This is because there are shorter paths between nodes, allowing for faster traversal of the network. In ER networks with smaller diameters, the average distance between nodes tends to be smaller. This is because there are more direct paths between nodes, leading to shorter average distances and variance.
4. In a network with a larger diameter, random walks may take longer to reach certain nodes as there are longer paths between them. This can lead to slower exploration of the network and potentially slower convergence to a steady state. In ER networks with larger diameters, the average distance between nodes tends to be larger, so does the variance. This is because nodes are more likely to be farther apart, requiring longer paths to reach each other.
5. A smaller diameter implies that, on average, the network is more tightly interconnected and concentrated around the starting node. Therefore, even though the network size increases, the decrease in diameter results in a more compact network structure, leading to quicker convergence and reduced spread in the shortest distances at the steady state. **Therefore, the diameter influences the efficiency and speed of random walks, with smaller diameters generally facilitating faster exploration and convergence.**

Overall, the diameter of an ER network significantly influences the behavior of random walks, affecting the efficiency, average distance, and variance of distances between nodes. Smaller diameters generally result in faster and more uniform exploration of the network, while larger diameters may lead to slower and more variable exploration.

QUESTION 2:

Random walk on networks with fat-tailed degree distribution

(a) Generate an undirected preferential attachment network with 900 nodes, where each new node attaches to m = 1 old nodes.

Using the barabasi.game library we created an Undirected Preferential Attachment Network with 900 nodes, where each new node attaches to m = 1 old nodes

```
IGRAPH 667a3db U--- 900 899 -- Barabasi graph
+ attr: name (g/c), power (g/n), m (g/n), zero.appeal (g/n), algorithm
| (g/c)
```

(b) Let a random walker start from a randomly selected node. Measure and plot ⟨s(t)⟩ v.s. t and σ2(t) v.s. t.

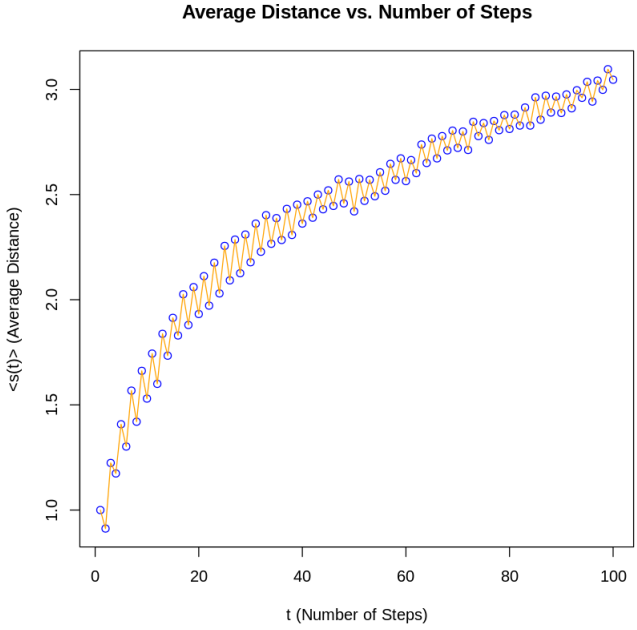
As given in the above question we implemented the following:

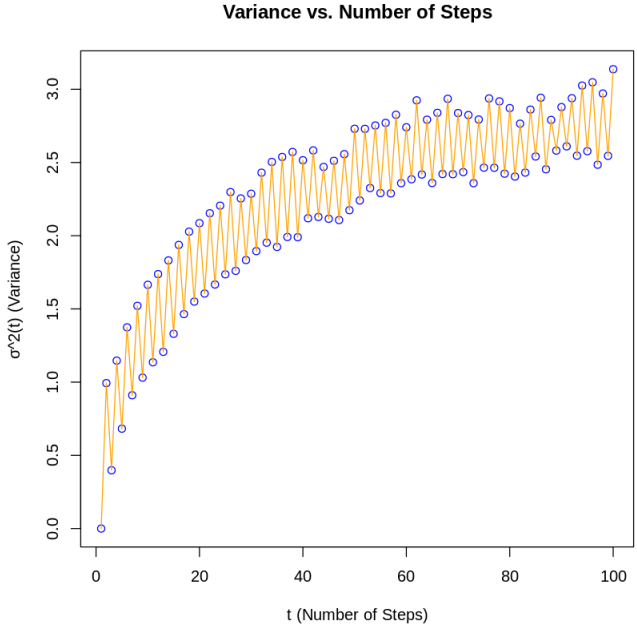
1. We created an undirected preferential attachment network with 900 nodes, where each new node attaches to m = 1 old nodes
2. We started a random walk on a randomly selected node in this GCC (Giant Connected Component).
3. We calculated the shortest distance s(t) for each step t from the starting node.
4. We averaged the distance and the variance over 1000 such random walks.
5. The above experiments were conducted with values of steps = 100, 800

Summary of the results:

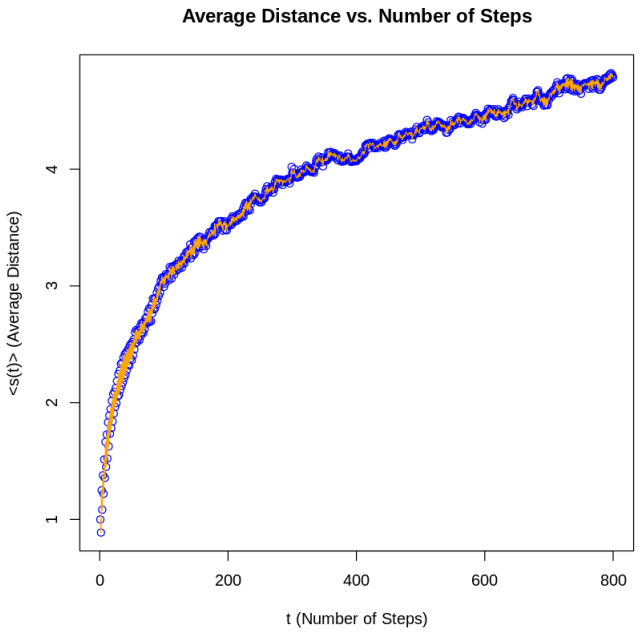
Random walk on G(n = 900, m = 1) with T = 100 and Iterations = 1000

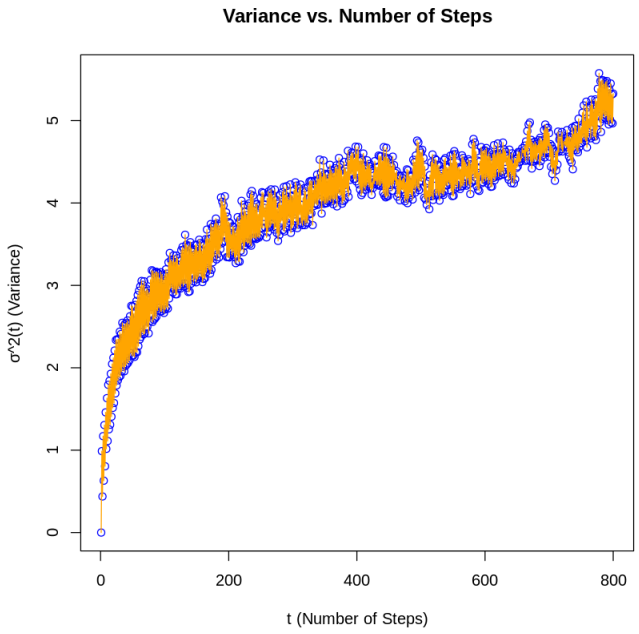
We can see from the graphs below that: ⟨s(t)⟩ and $\sigma^2(t)$ both reach a steady state around t = 700.



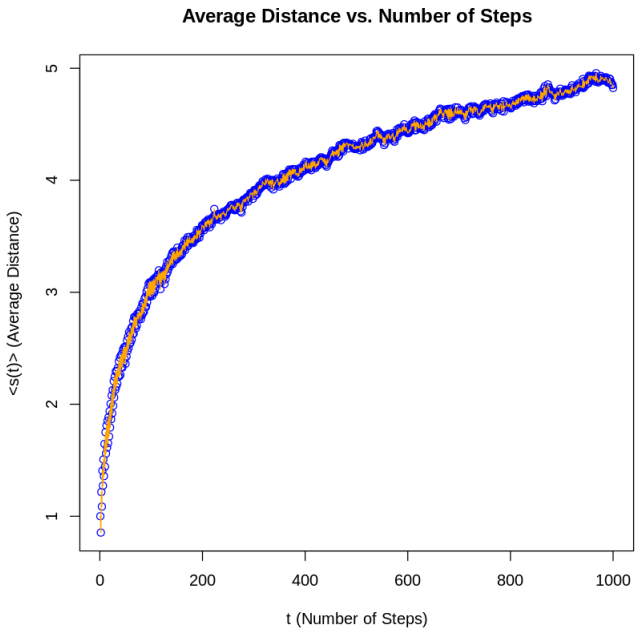


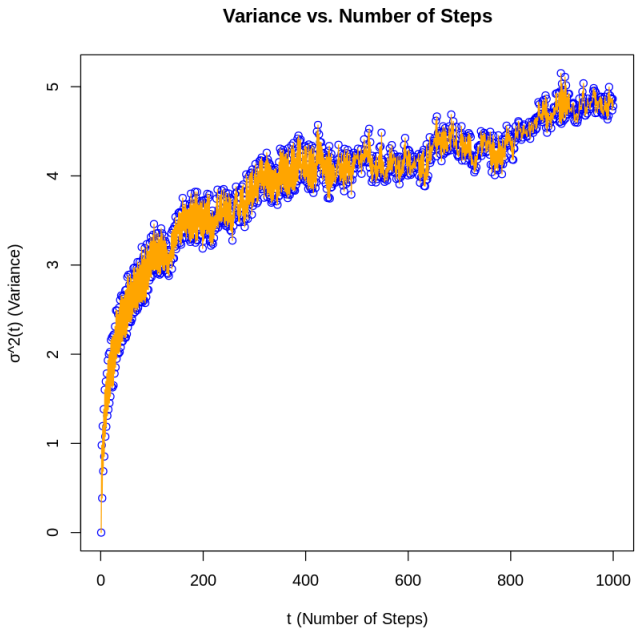
Random walk on G(n = 900, m = 1) with T = 800 and Iterations = 1000





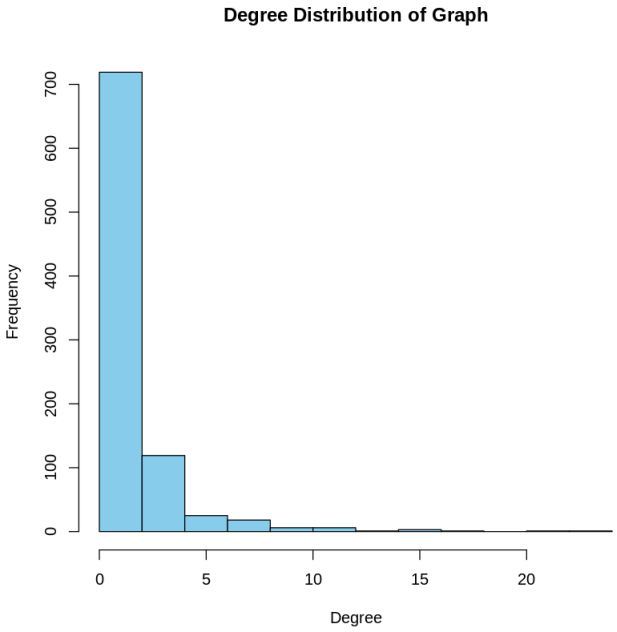
Random walk on G(n = 900, m = 1) with T = 1000 and Iterations = 1000

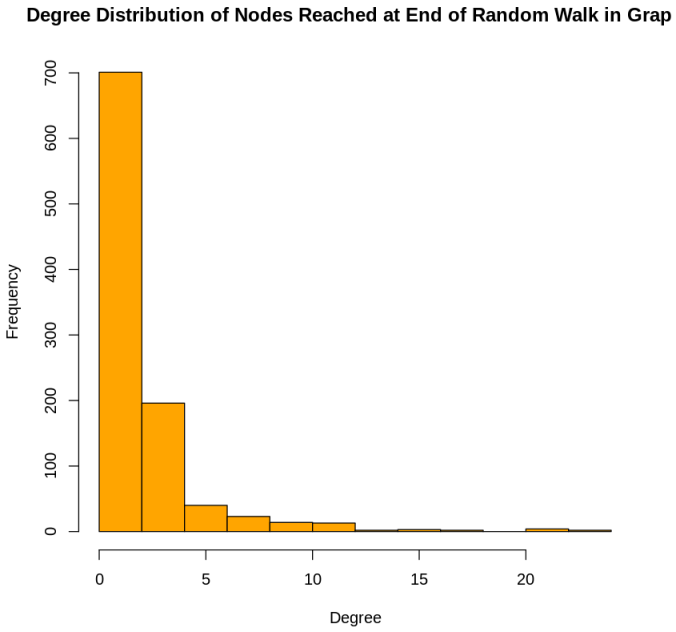




(c) Measure the degree distribution of the nodes reached at the end of the random walk on this network. How does it compare with the degree distribution of the graph?

We have plotted the degree distribution of the Original Graph and of the Nodes reached at the end of the Random Walk for G(n = 900, m = 1) and T = 100 for 1000 random walks.





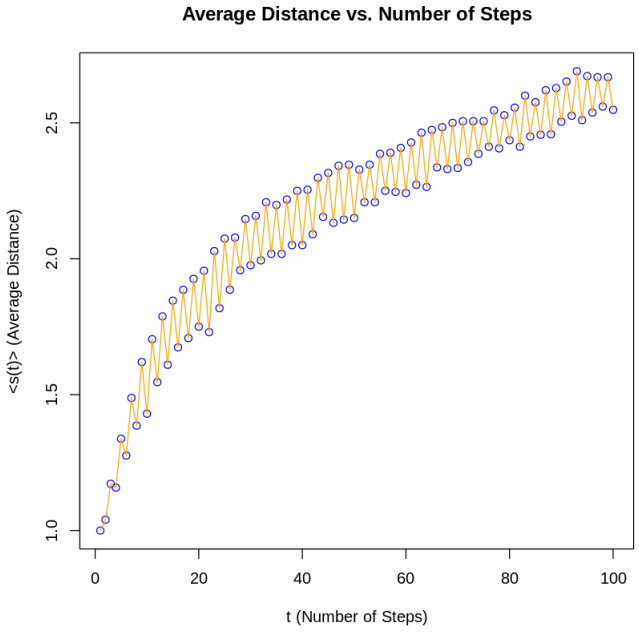
We observe that both the original graph and the graph for the distribution at the end of the random walk follow a similar power law distribution. These results align with what is expected for preferential Attachment networks where we know that the degree distribution typically follows a power-law distribution. This means that a small number of nodes have a very high degree, while the majority of nodes have relatively low degrees.

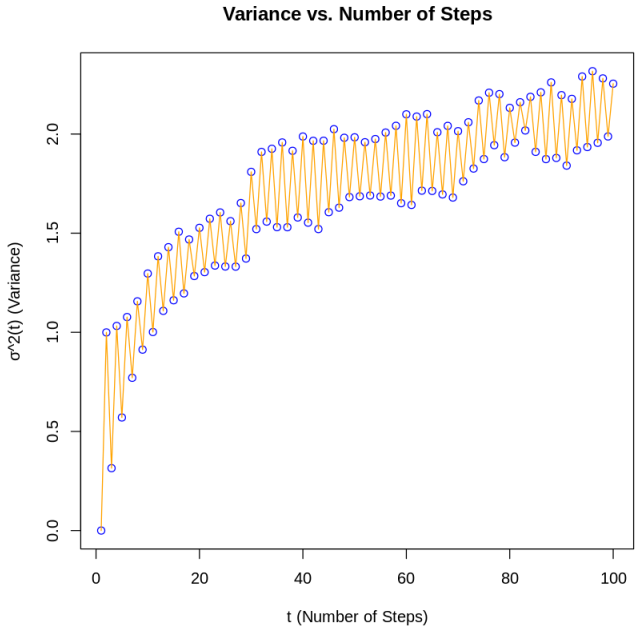
Visually, the degree distribution of preferential attachment networks often appears as a straight line on a log-log scale plot. The plot shows a linear relationship between the logarithm of the degree log(k) and the logarithm of the probability log(P(k)). This linearity indicates the power-law behavior of the degree distribution.

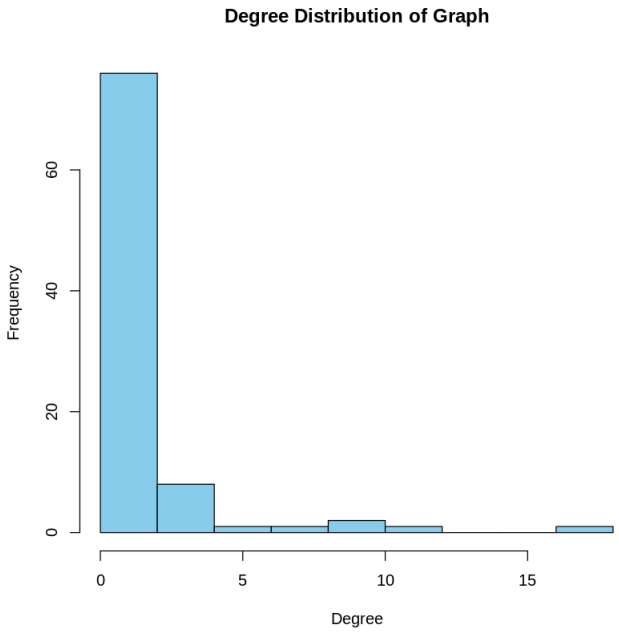
In summary, the degree distribution of preferential attachment networks exhibits a long-tailed distribution, with a small number of highly connected nodes (hubs) and many nodes with low degrees. This distribution reflects the mechanism of preferential attachment, where new nodes tend to attach preferentially to highly connected nodes, leading to the rich-get-richer phenomenon.

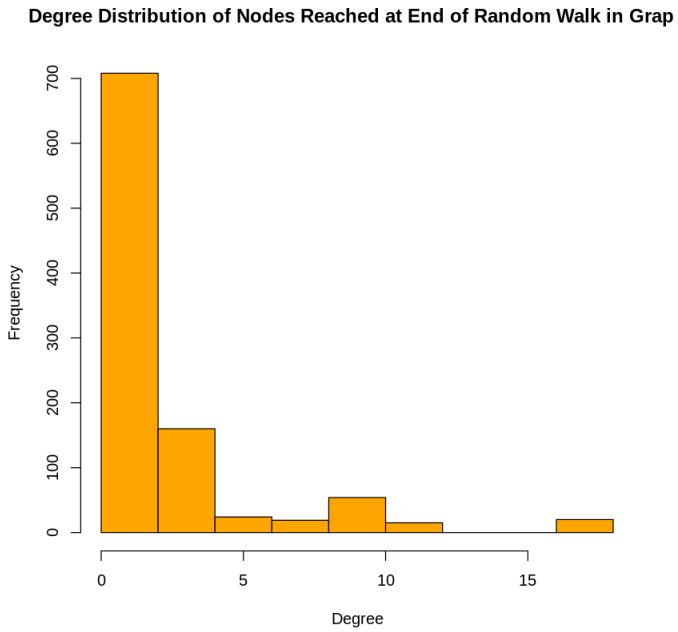
(d) Repeat 2(b) for preferential attachment networks with 90 and 9000 nodes, and m = 1. Compare the results and explain qualitatively. Does the diameter of the network play a role?

Random walk on G(n = 90, m = 1) with T = 1000 and Iterations = 1000

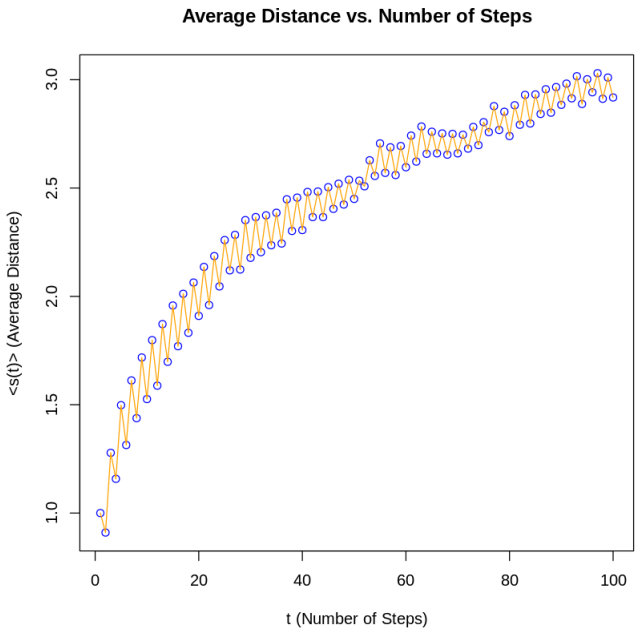


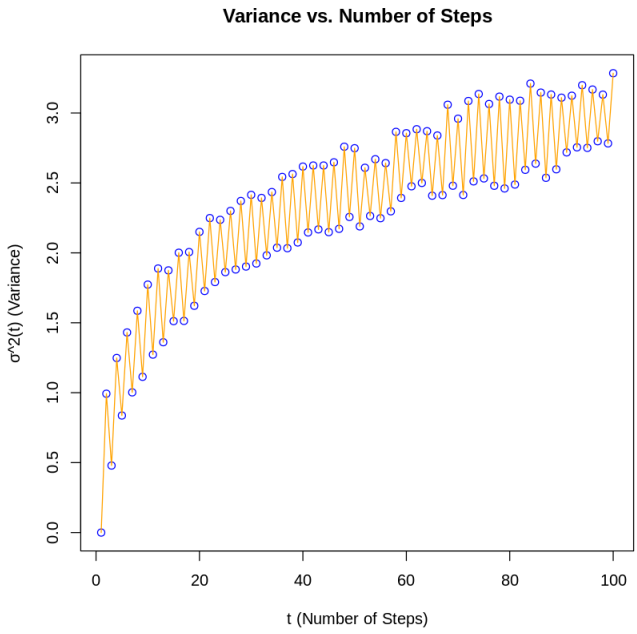


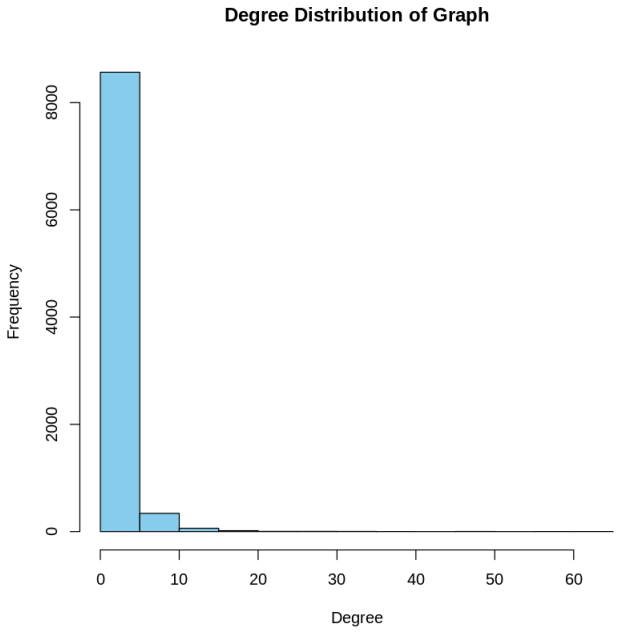


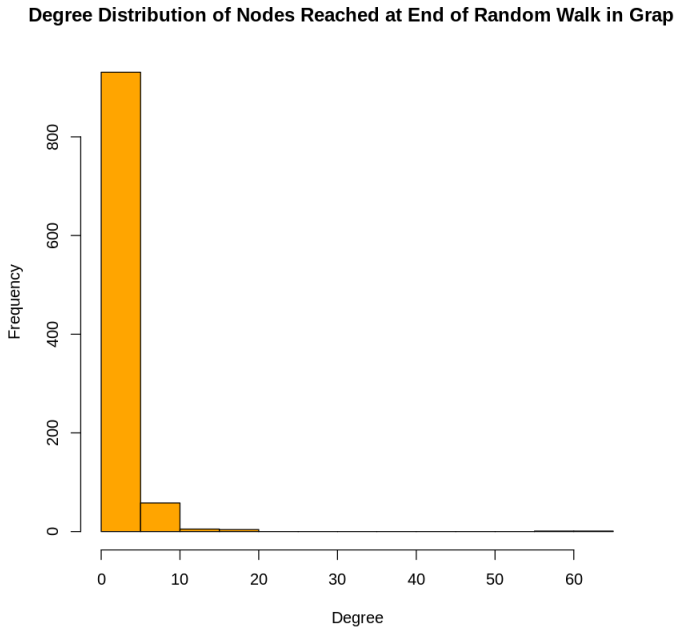


Random walk on G(n = 9000, m = 1) with T = 1000 and Iterations = 1000









```
diameter(pref_attach_network)
diameter(pref_attach_network_90)
diameter(pref_attach_network_9000)

```

```
21
12
27
```

As the number of nodes n in the preferential attachment network increases, several observations can be made:

1. **Increase in Steps to Reach Steady State**
With the increase in n, the number of steps required for the random walk to reach a steady state also increases. This indicates that larger networks take longer to converge to a steady state compared to smaller networks.

2. **Increase in Spread of Shortest Distances and Variance**
At the point of convergence, both the average shortest distance and the variance of shortest distances increase. This increase implies that, upon convergence, there is more variability in the shortest distances from any node to the starting node.

3. **Relationship with Diameter**
Upon examining the corresponding diameters of the networks, it is observed that as n increases, the diameter of the network also increases. This observation suggests that the pattern observed in terms of convergence and spread of shortest distances aligns with the changes in network diameter.

In summary, as the size of the preferential attachment network increases, it takes longer to converge to a steady state, and there is more variability in the shortest distances to the starting node upon convergence. This pattern is consistent with the increase in network diameter, indicating a relationship between network size, convergence behavior, and network structure.

QUESTION 3:

PageRank

The PageRank algorithm, as used by the Google search engine, exploits the linkage structure of the web to compute global "importance" scores that can be used to influence the ranking of search results. Here, we use random walk to simulate PageRank.

(a) We are going to create a directed random network with 900 nodes, using the preferential attachment model. Note that in a directed preferential attachment network, the out-degree of every node is m, while the in-degrees follow a power law distribution. One problem of performing random walk in such a network is that, the very first node will have no outbounding edges, and be a "black hole" which a random walker can never "escape" from. To address that, let's generate another 900-node random network with preferential attachment model, and 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. For example,
Create such a network using m = 4. Measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?
Hint Useful function(s): asedgelist , sample , permute , addedges

As per the question, we generated 2 Preferential Attachment networks with n = 900 nodes and m = 4. Subsequently we permuted the nodes of the second network and obtained a permuted edge list. We used this edge list to add the edges to the first network, thereby creating a modified Preferential Attachement network, with n = 900.

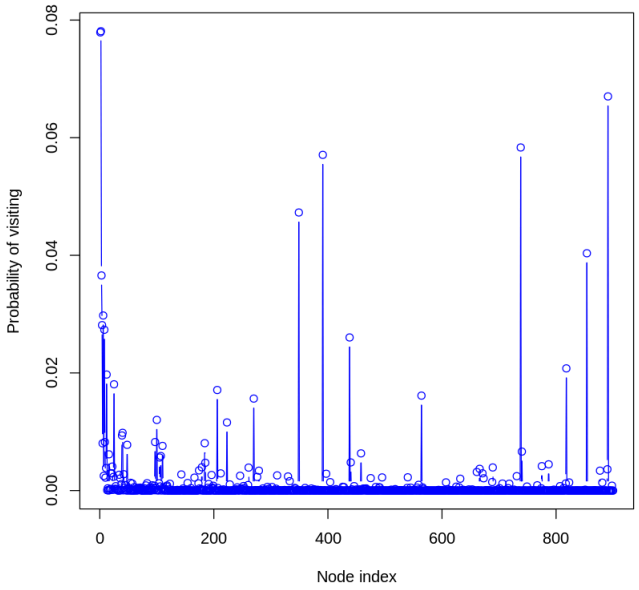
```
[74]  # Generating 2 networks with 900 nodes and m = 4

      graph_1 <- sample_pa(n=900, m=4, directed=T)
      graph_2 <- sample_pa(n=900, m=4, directed=T)
```

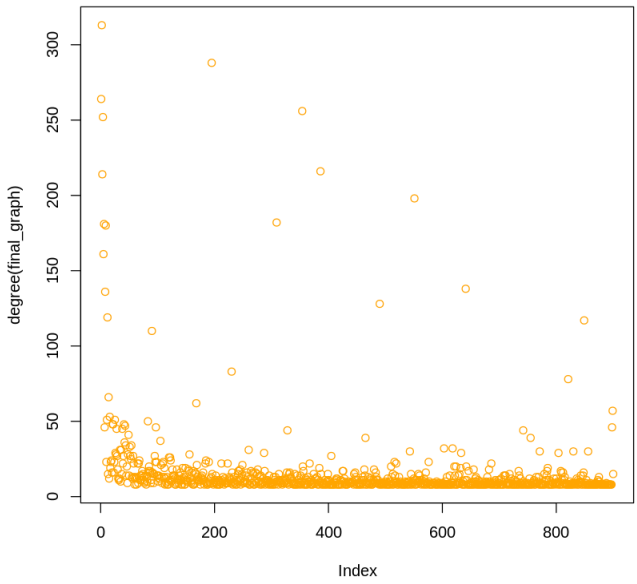
```
[75]  g2p <- permute(g2, sample(vcount(g2)))
```

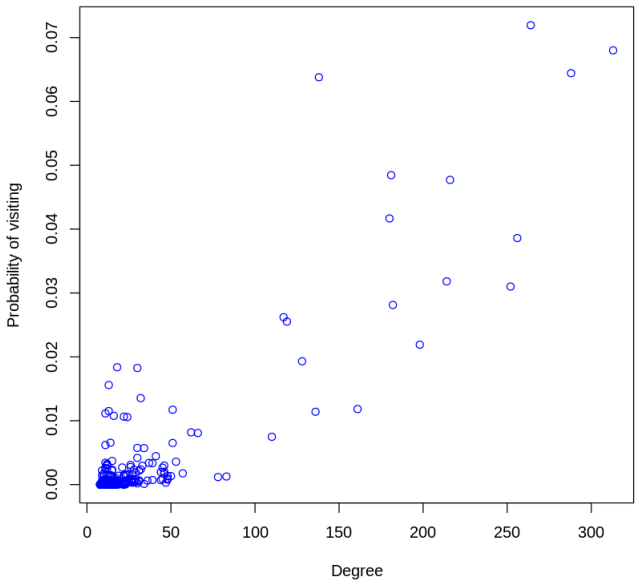
```
      final_graph = add_edges(g1, c(t(as_edgelist(g2p))))
```

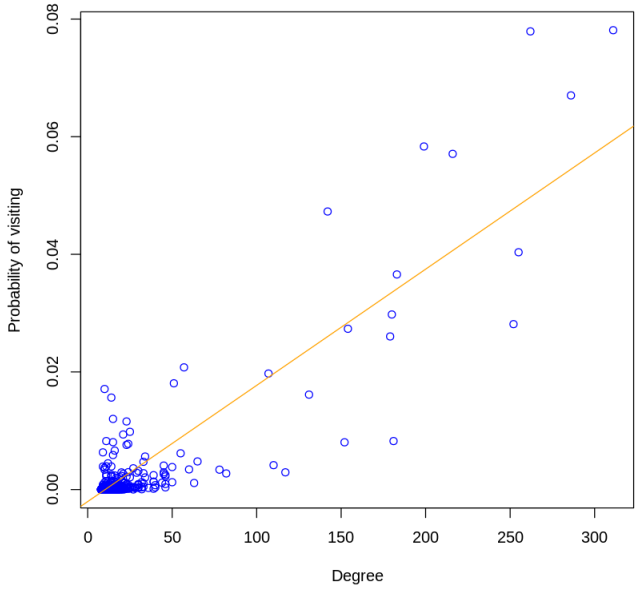
To compute the node-visitation probabilities, we conduct 100 random walks, each comprising 1000 steps, with each walk commencing at an initial node selected uniformly at random. We then tally the number of times each node is visited, averaged over the 100 random walks. It's noteworthy that we endeavor to calculate visits only after each random walk has attained its steady-state—approximated by recording visits after the first ln n steps, where ln n represents the number of steps required for exponential convergence.

The resulting probability distribution of the random walker visiting each node is illustrated in the figures below further elucidates the probability that the random walker visits nodes with a specified degree.









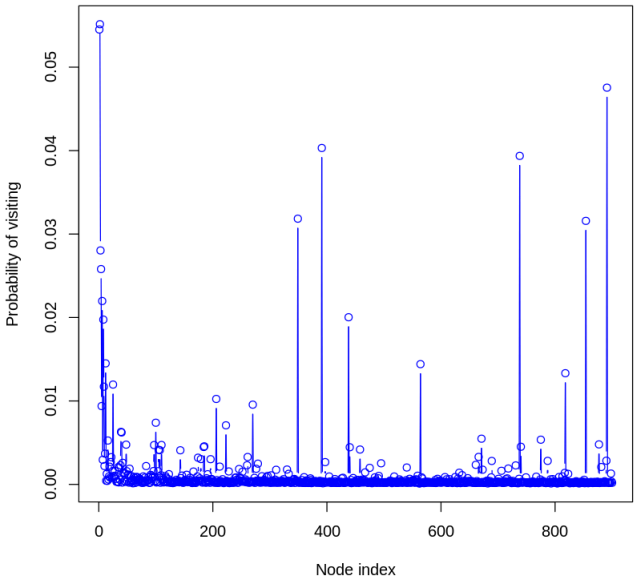

```
[1] "Slope"
[1] 0.0001975949
[1] "Intercept"
[1] -0.002041626
[1] "Correlation Coefficient"
[1] 0.8844222
```

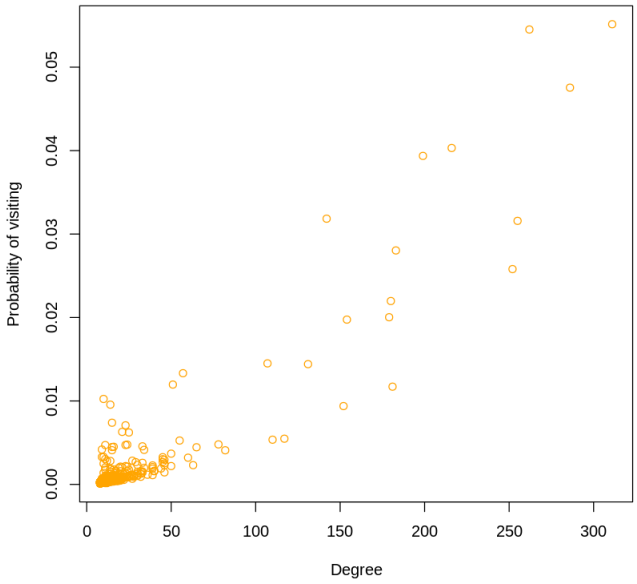
The Pearson correlation coefficient of this relationship is 0.8844222, with a linear fit yielding a slope of 0.0001975949. The relatively high Pearson coefficient indicates an approximately proportional relationship. This observation aligns with intuition, as nodes with higher in-degree should logically have a greater likelihood of being visited.

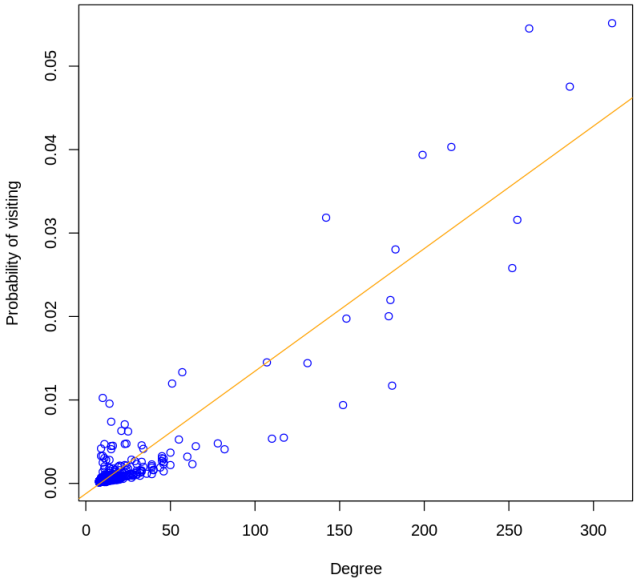
(b) In all previous questions, we didn't have any teleportation. Now, we use a teleportation probability of α = 0.2 (teleport out of a node with prob=0.2 instead of going to its neighbor). By performing random walks on the network created in 3(a), measure the probability that the walker visits each node. How is this probability related to the degree of the node and α ?

In this question, we repeated the experiment but with the addition of teleportation capability during the random walk. Consequently, at each step, the random walker opts to visit one of its neighbors with probability (1 − α) and teleports with probability α, selecting the next node with equal probability.

The resulting probability distribution of the random walker visiting each node under these conditions is depicted in the figures below. It illustrates the probability that the random walker visits nodes with a specified degree. The Pearson correlation coefficient of this relationship stands at 0.9249222, with a linear fit yielding a slope of 0.0001468477.








```
[1] "Slope"
[1] 0.0001468477
[1] "Intercept"
[1] -0.001231925
[1] "Correlation Coefficient"
[1] 0.9249222
```

1. Probability and Node Degree:
- Typically, in preferential attachment networks, nodes with higher degrees tend to attract more visits during a random walk. This is because nodes with higher degrees have more connections and, therefore, are more likely to be traversed.
- We can expect nodes with higher degrees to have higher visitation probabilities compared to nodes with lower degrees.

2. Effect of Teleportation (α):
- As we increase the teleportation probability α, the influence of node degrees on visitation probabilities diminishes. Higher values of α mean that the walker is more likely to teleport to a random node instead of traversing along edges.
- With α = 0, where there is no teleportation, the probability of visiting each node is primarily influenced by its degree.
- As α increases, the visitation probabilities become more uniform across nodes, regardless of their degrees. This is because teleportation introduces randomness, making all nodes equally likely to be visited.

Overall, as we vary the teleportation probability α, we'll observe a shift from visitation probabilities being predominantly influenced by node degrees to a more uniform distribution across nodes due to the increased randomness introduced by teleportation.

The relatively lower slope observed compared to 2.3(a) is in line with the observation that the probability of visitation for nodes with higher degrees has decreased comparatively. This trend reflects the trade-off where lower-degree nodes now have a heightened probability of being visited due to the introduction of teleportation.

QUESTION 4:

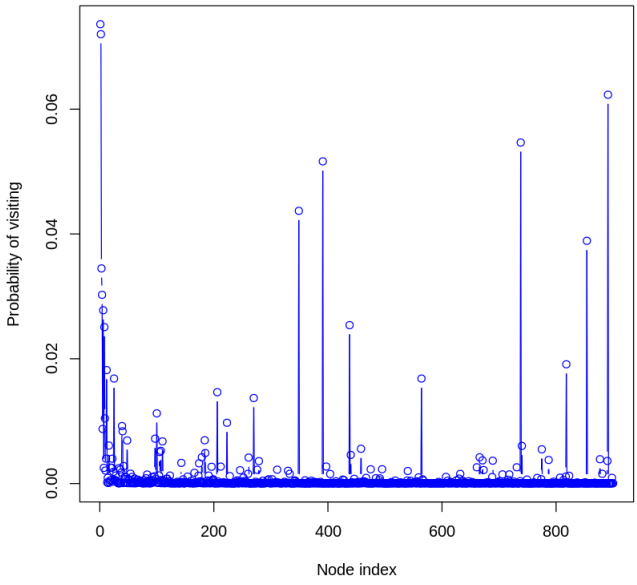
Personalized PageRank

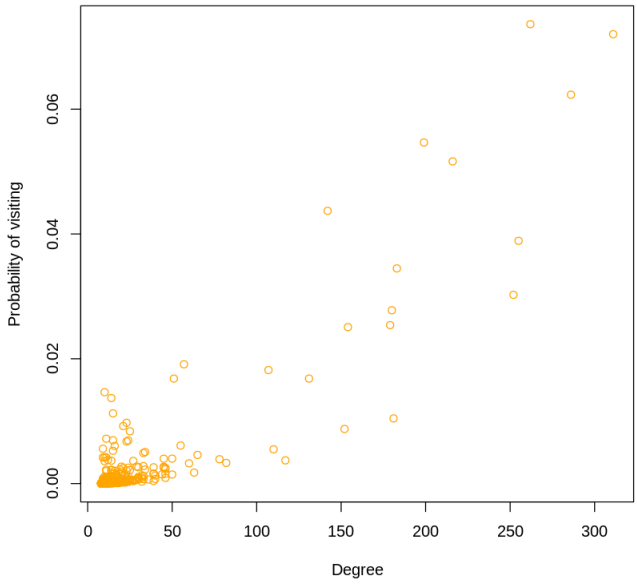
While the use of PageRank has proven very effective, the web's rapid growth in size and diversity drives an increasing demand for greater flexibility in ranking. Ideally, each user should be able to define their own notion of importance for each individual query.

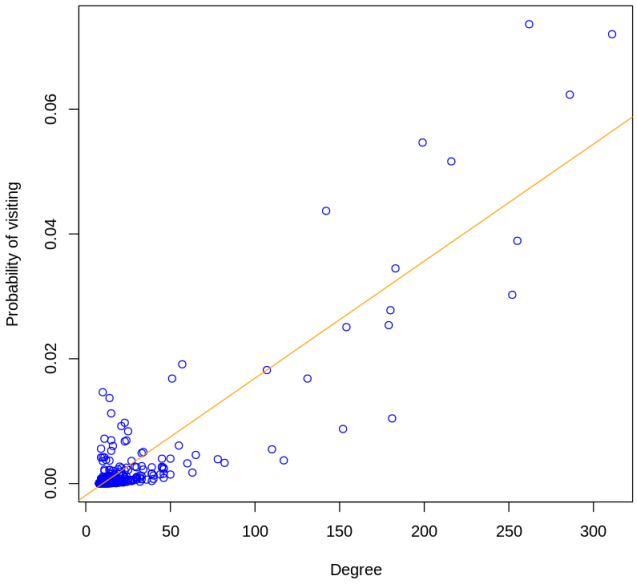
(a) Suppose you have your own notion of importance. Your interest in a node is proportional to the node's PageRank, because you totally rely upon Google to decide which website to visit (assume that these nodes represent websites). Again, use random walk on network generated in question 3 to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank (as opposed to the regular PageRank, where at teleportation, the chance of visiting all nodes are the same and equal to N1). Again, let the teleportation probability be equal to α = 0.2. Compare the results with 3(a).

Based on the above question, we can use the page_rank function to calculate the page ranks. We see in contrast to the previous questions, here when the random walker teleports, the probability that the next node that is to be chosen is proportional to the PageRank.

The resulting probability distribution of the random walker visiting each node under these conditions is depicted in the figures below. It illustrates the probability that the random walker visits nodes with a specified degree. The Pearson correlation coefficient of this relationship stands at 0.89867, with a linear fit yielding a slope of 0.0001876749.



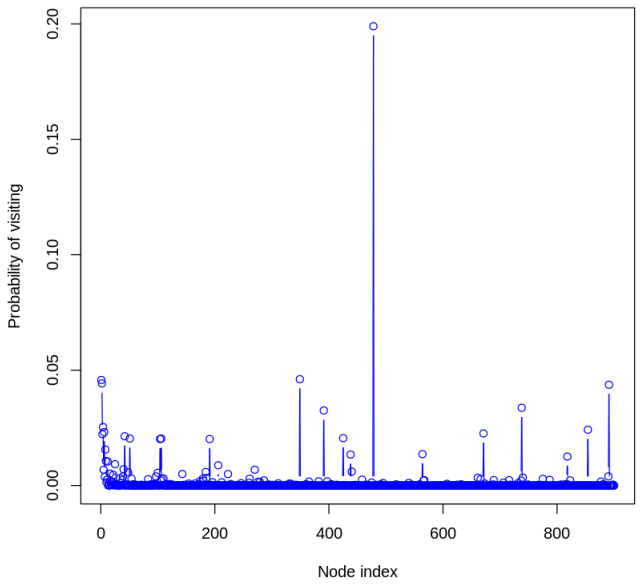


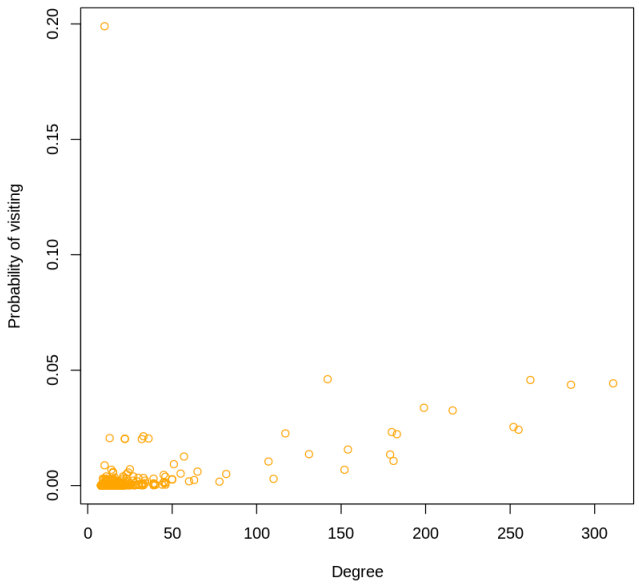


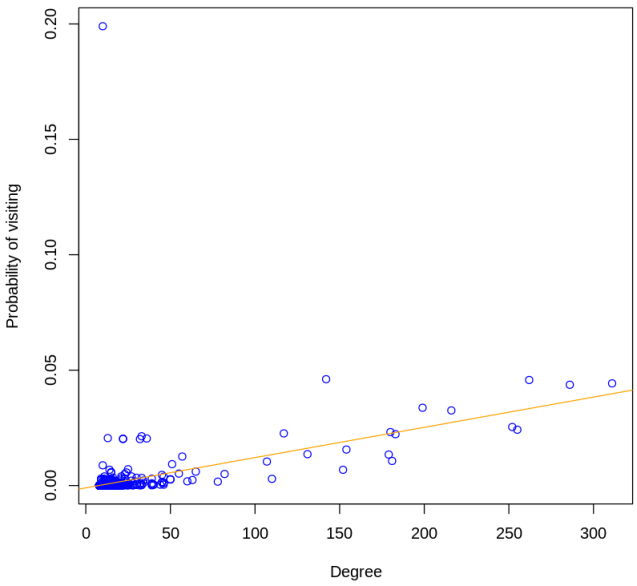

```
[1] "Slope"
[1] 0.0001876749
[1] "Intercept"
[1] -0.001883346
[1] "Correlation Coefficient"
[1] 0.89867
```

Compared to the previous question we observe that there is an increase in the slope, as well as an increase in the probability of visitation of nodes with higher degrees. This is because teleportations now have a higher probability of landing on nodes with higher PageRanks, which in turn have higher degrees.

(b) Find two nodes in the network with median PageRanks. Repeat part 4(a) if teleportations land only on those two nodes (with probabilities 1/2, 1/2). How are the PageRank values affected?





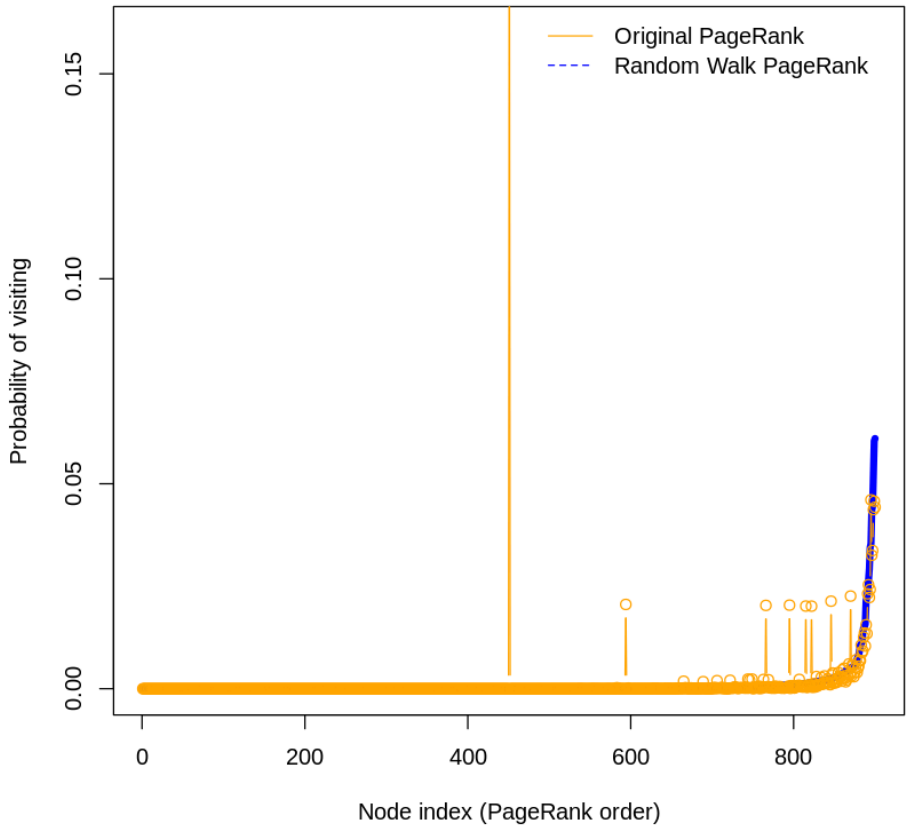



```
[1] "Slope"
[1] 0.0001310688
[1] "Intercept"
[1] -0.0009801638
[1] "Correlation Coefficient"
[1] 0.4570713
```

The approach here mirrors that of 2.4(a), with a notable tweak: during teleportation, the random walker now lands exclusively on one of the two nodes possessing median PageRanks, each selected with an equal chance. This modification ensures that the teleportation process directs the walker specifically to these pivotal nodes.

The outcomes are detailed in the above figures, where (a) provides the probability distribution of the walker's visits to individual nodes, and (b) illustrates the likelihood of visiting nodes based on their respective degrees. Notably, the Pearson correlation coefficient registers at 0.4570713, showcasing a decline compared to prior experiments. This drop stems from the significant outliers observed in the upper left quadrant of the above figure, representing the two nodes boasting median PageRanks.

The remaining data points adhere to a pattern akin to that observed in 2.3(a), unaffected by teleportation's adjustments. This distinction underscores the unique influence exerted by targeting specific nodes during the teleportation phase.



To investigate the effect of the random walk on the PageRank, we plotted the above figure:

In this context, we've made the assumption that the steady-state probabilities of node visitation by the random walker align with the adjusted PageRank values. The Node Index designation follows the ascending order of PageRanks from the original graph, resulting in a prominent peak at the center that corresponds to the two nodes holding median PageRanks.

A noteworthy observation emerges: several other nodes experience a discernible uptick in probability. These nodes likely reside in the immediate vicinity of the median nodes, possibly receiving incoming links from them. As the random walker visits the median nodes, it naturally progresses to their neighboring nodes, thereby enhancing their PageRank values as well.

(c) More or less, 4(b) is what happens in the real world, in that a user browsing the web only teleports to a set of trusted web pages. However, this is against the assumption of normal PageRank, where we assume that people's interest in all nodes are the same. Can you take into account the effect of this self-reinforcement and adjust the PageRank equation?

To account for self-reinforcement in the PageRank equation, we can introduce a damping factor that represents the probability of following a link versus randomly teleporting to any node. This modification acknowledges the inherent bias towards highly connected nodes while still allowing for random exploration of the network. Mathematically, this adjustment can be incorporated into the PageRank algorithm as follows:

$$PR(u) = \frac{1-\alpha}{N} + \alpha \sum_{v \in B_u} \frac{PR(v)}{C_v}$$

Where:

- (PR(u)) is the PageRank of node (u).
- (N) is the total number of nodes in the network.
- (α) is the damping factor (typically set between 0.1 and 0.2).
- (B_u) represents the set of nodes that link to node (u).
- (C_v) denotes the out-degree of node (v).

This adjusted formula reflects the idea that a node's importance is not only influenced by the number of incoming links but also by the importance of the linking nodes and their own link structures. By incorporating self-reinforcement in this manner, the modified PageRank equation better aligns with real-world scenarios where nodes with higher PageRank values tend to attract more attention and reinforcement.

Denote by A the Node-Node incidence matrix of a directed graph, by P the Node transition matrix for a random walker on the graph (with teleportation probability α) and by $k_{out}(i)$ the out-degree of node *i*. Also let the P_{ij} element of P represent the probability of landing on node *j* after the random walker has landed on node *i*. Then for a standard teleporting random walker,

$$P_{ij} = (1-\alpha)\frac{1}{k_{\text{out}}(i)}A_{ij} + \alpha\frac{1}{n}$$

where n = Number of nodes in the graph. Now consider the case where we only teleport to some set T of trusted nodes in the graph with equal probability, and do not teleport to other nodes at all. Then the expression for P_{ij} changes to

$$P_{ij} = \begin{cases} (1-\alpha)\frac{1}{k_{\text{out}}(i)}A_{ij} + \alpha\frac{1}{|T|} & i \in T \\ (1-\alpha)\frac{1}{k_{\text{out}}(i)}A_{ij} & i \notin T \end{cases}$$

At equilibrium, for every node *i*,

$$\pi(i) = \sum_{j=1}^{n} P_{ji}\pi(j)$$

So the PageRank equation becomes (for every node *i*),

$$\pi(i) = \begin{cases} (1-\alpha)\sum_{j=1}^{n} \frac{1}{k_{\text{out}}(j)} A_{ji}\pi(j) + \alpha\frac{1}{|T|} & i \in T \\ (1-\alpha)\sum_{j=1}^{n} \frac{1}{k_{\text{out}}(j)} A_{ji}\pi(j) & i \notin T \end{cases}$$

In particular, for 2.4(b), T = {*i* | Node *i* is one of the two median nodes with respect to PageRank} where |T| = 2.