UCLA ECE 232E S2021 Project #1

Random Graphs and Random Walks

Lin Fan - 505627503

Sunay Bhat - 905629072

Yi-chun Hung - 705428593

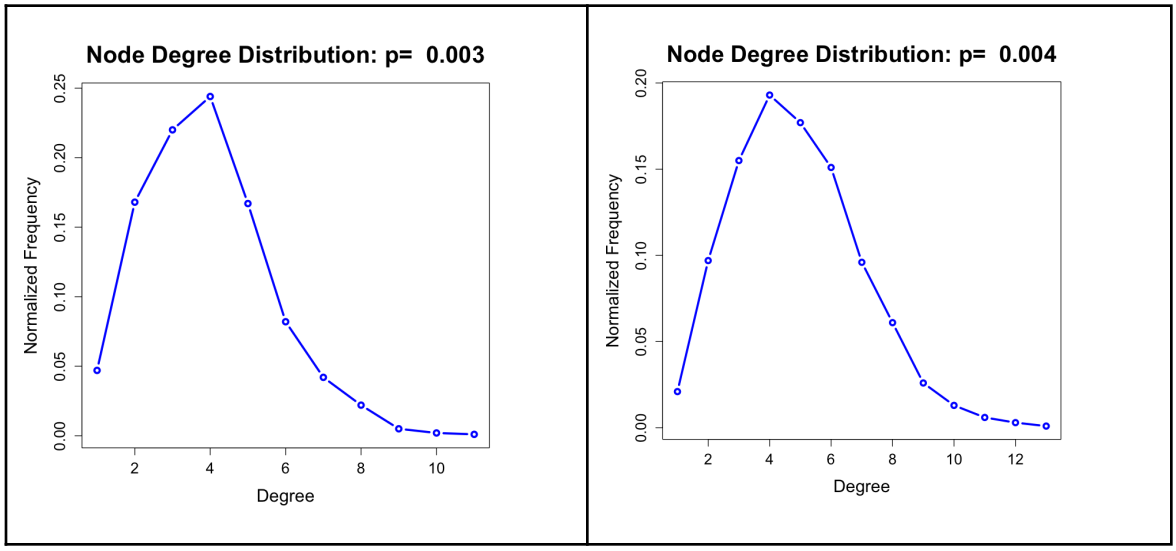
Tu Yu-Hsien - 405627283

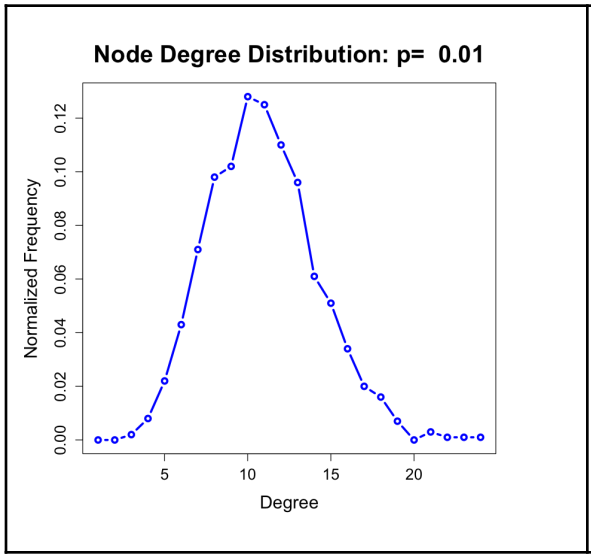
Part I: Generating Random Networks

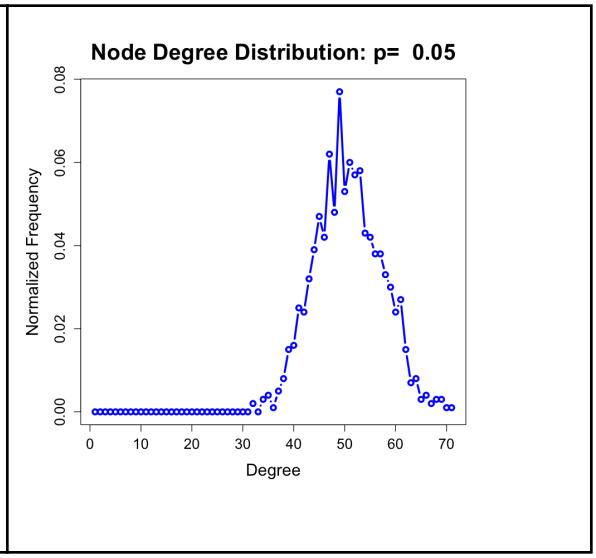
Problem 1

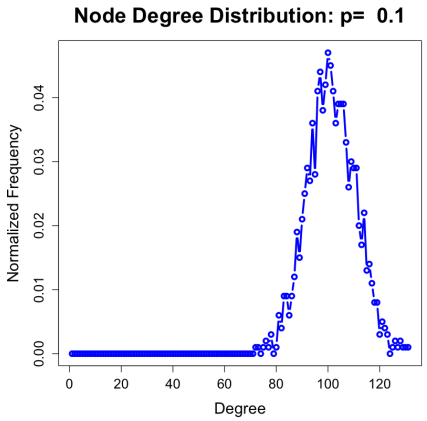
For part 1, problem 1 of the project, we are asked to generate random networks using the Erdos-Renyi (ER) model. In the ER method, a graph with n vertices is generated where every possible edge is independently generated using input probability p. We explore features of this graph in the subsequent exercises.

a) For part a, we plot the degree distribution of ER networks. The following five plots show the **degree distribution for undirected random graphs** using the ER model with n=1000 vertices and p= 0.003, 0.004, 0.01, 0.05, and 0.1 edge probabilities.









We observe a **binomial distribution of K, or node degree random variable: *b(k; n-1,p)***, since for a given node, the probability of connecting to any of the remaining n-1 nodes is p. The table below lists the mean and variances of the degree distributions compared to the theoretical values based on the binomial distribution parameters as shown in the equations below. The measured and theoretical values are very similar.

$$\mu\mu = (n-1) * p = 999 * p$$
$$\sigma^2 = (n-1) * p * (1-p) = 999 * p * (1-p)$$

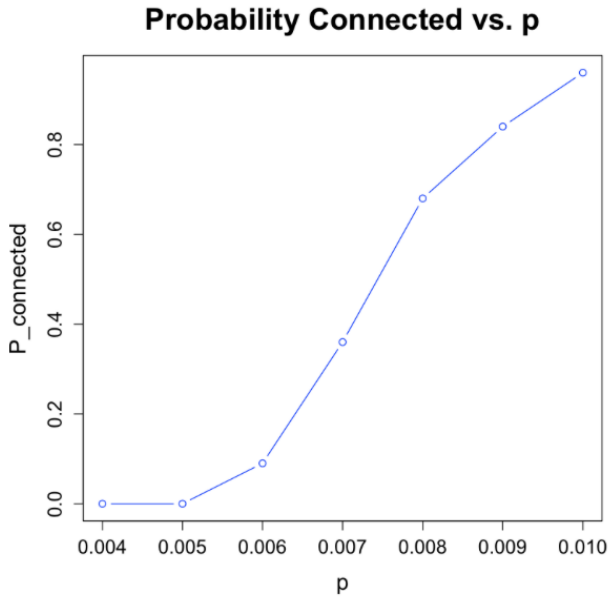
p	Mean	Mean (Theoretical)	Variance	Variance (Theoretical)
0.003	2.97	3.00	2.90	2.99
0.004	4.05	4.00	4.42	3.98
0.01	9.90	9.99	10.42	9.89
0.05	49.85	49.95	48.32	47.45
0.1	99.68	99.9	92.15	89.91

b) For part b, we are asked to determine the probability of the above ER networks being connected. We can **estimate this analytically by generating 100 instances** of each network, and counting the times the resulting graph is connected. The table below shows these results.

p	Always Connected	$P_{Connected}$	GCC Diameter (mode of 100)
0.003	No	0	14

0.004	No	0	11
0.01	No	.96	6
0.05	Yes	1	-
0.1	Yes	1	-

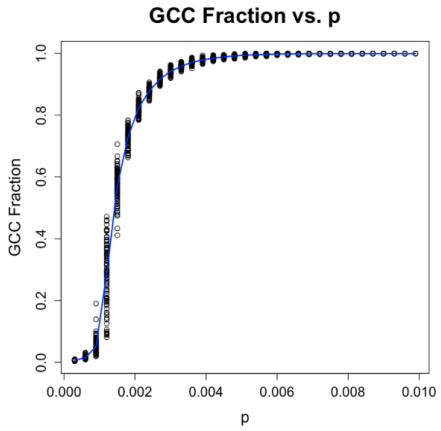
Between p = .004 and .01 the probability of a connected graph increases, and we explore this non-linear relationship and region in the plot below. **Around .0075, the probability of a connected graph is about 0.5**. Additionally the GCC diameter decreases as p increases (until the full graph is connected), which makes sense given the increase in edges within the GCC as p increases.

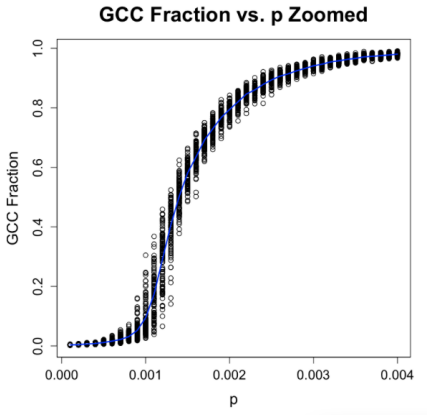


c) For part c, we explore the probability threshold in which a GCC emerges and dominates in our ER network generation scheme. Our **theoretical** assumption is that the two transition points of interest occur at ***1/n*** and ***ln(n)/n*** for emergence of a GCC and GCC Fraction > 99% (or almost always fully connected) respectively.

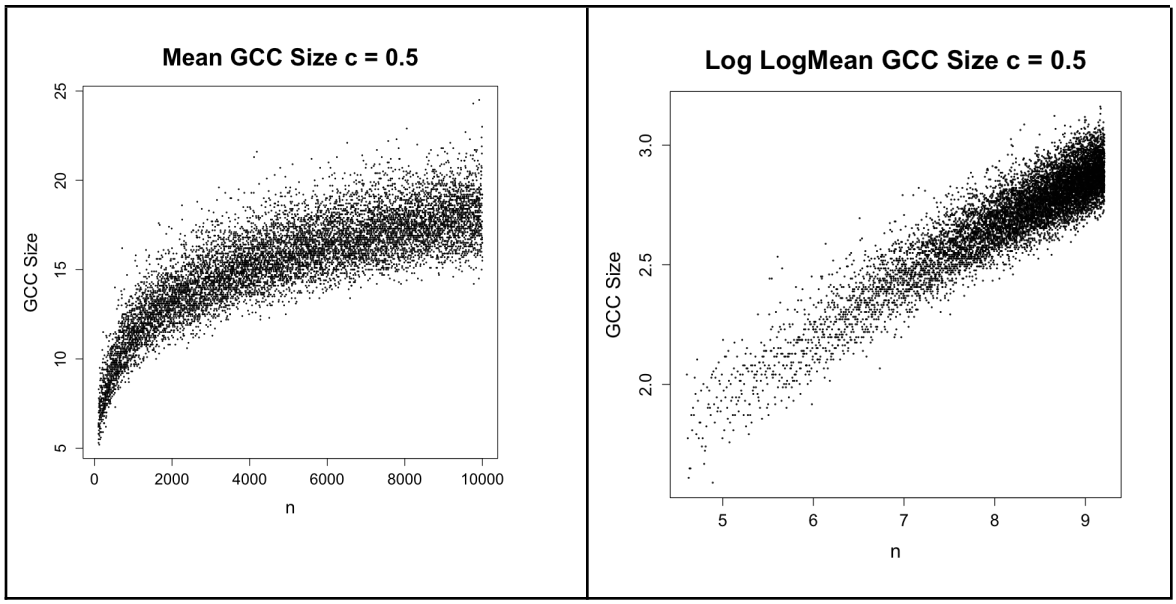
```
"Transition Point 1 (Emergence):  0.001"
"Transition Point 2 (Dominance):  0.00690775527898214"
```

The graphs below support these theoretical transition points as we clearly see a GCC emerge around ***0.001*** (see zoomed second plot), and we approach a GCC fraction of 1 after .004, but the variance clearly drops off and our iterations are almost always fully connected after ***0.007***.

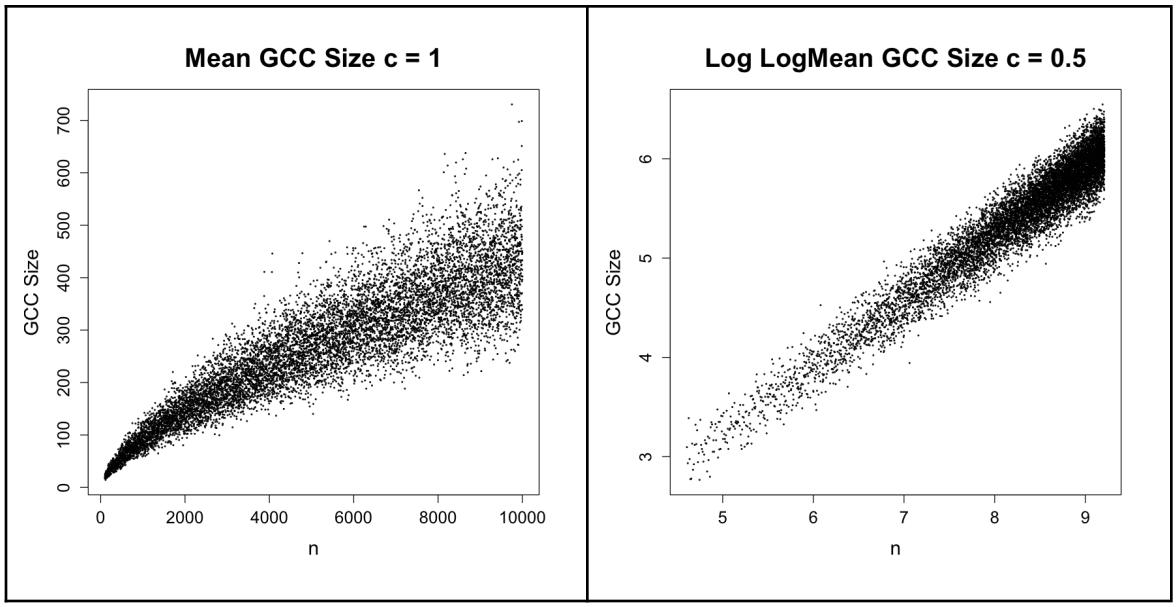




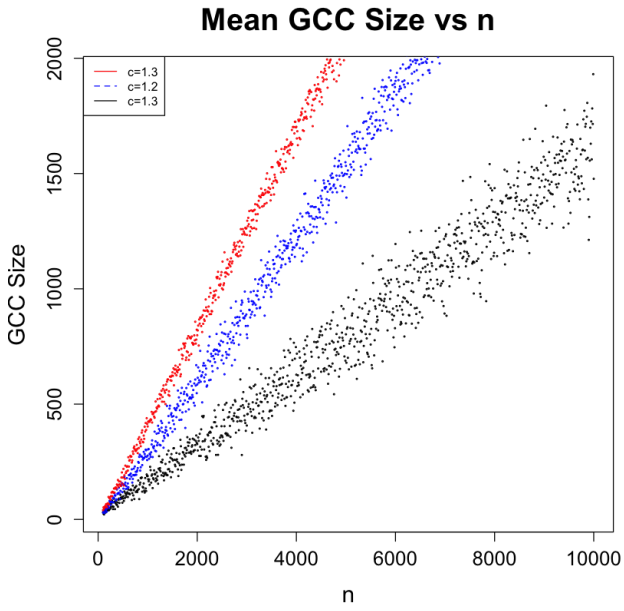
d) For part d we look at GCC size when we fix the average degree node, c = n * p. In order to compute the expected GCC size, w**e performed 10 iterations at every n** from 100:10000, and took the mean GCC size, and then plotted this against n for c= 0.5. We observe that the **overall relationship is nonlinear**, and GCC size increases faster at low n, but **eventually the relationship appears to approach a linear** regime past a certain n (~3000 in our plot below). If we plot on a **log-log scale we see a linear** relationship with a **slope** of about **0.2**.



This was repeated for c = 1. The GCC sizes have increased from c= 0.5 significantly, since p increases with c. It also appears that a linear relationship is approached much earlier at ~n=2000. Again, a **log-log** is more linear with a **slope of 0.6**.



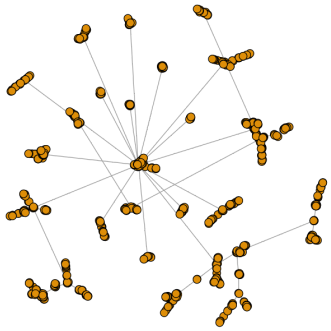
For the last part, we did this again for c=1.1,1.2,and 1.3. The plot below is the results with the same procedure except every 10th n (100:10:10000) to save computational time. It appears that **as we increase c, the relationship is more linear** (or is linear at a much earlier n). Additionally, **the GCC increases with c**, which makes sense since the probability of edge connection increases as well.



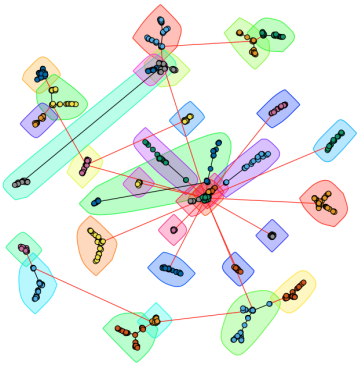
Problem 2

For part 1 problem 2, we explore the preferential attachment (PA) method of generating a graph. This is also referred to as the Barabasi-Albert method, in which a graph is initialized with a few nodes. Each new node attaches to m existing nodes (m being a specified parameter), and the probability of choosing a node is proportional to its existing degree. This graph is connected by generation method, and it also produces power-law distributions as we will see in the coming sections.

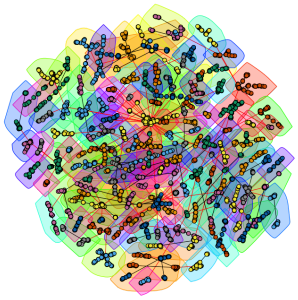
a) Such a graph **is always connected by construction method**. We repeated this construction 10 times to confirm, but since the original starting network is connected and every node gets connected to an existing node, the final network is always connected in a preferential/Barabasi network.



b) Community structure of a network is how well nodes can be grouped into densely linked regions. Modularity measures how well a network is grouped into communities by quantifying the number of edges in a community vs the number of edges linking it to other communities (while avoiding the trivial case of the whole graph). T**he modularity of our graph was 0.931**. A high modularity is expected based on the way preferential attachment works. The community structure is pictured below.



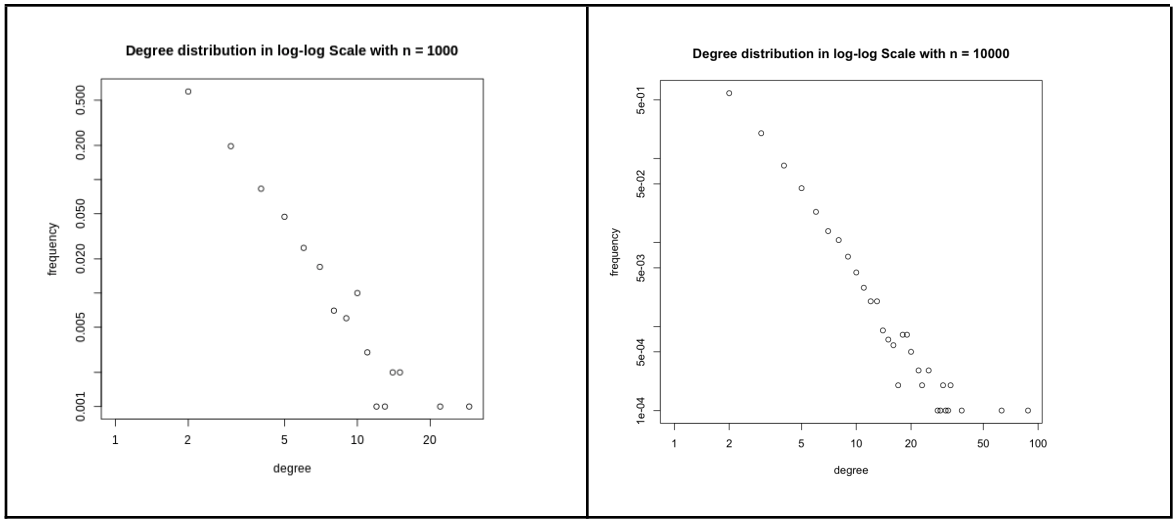
c) The graph **modularity increases to 0.978** when we go to **n=10000**. This makes sense as **preferential attachment will continue to amplify the degree** and thus connectedness of a few nodes as n increases, further clustering the communities in denser groups.



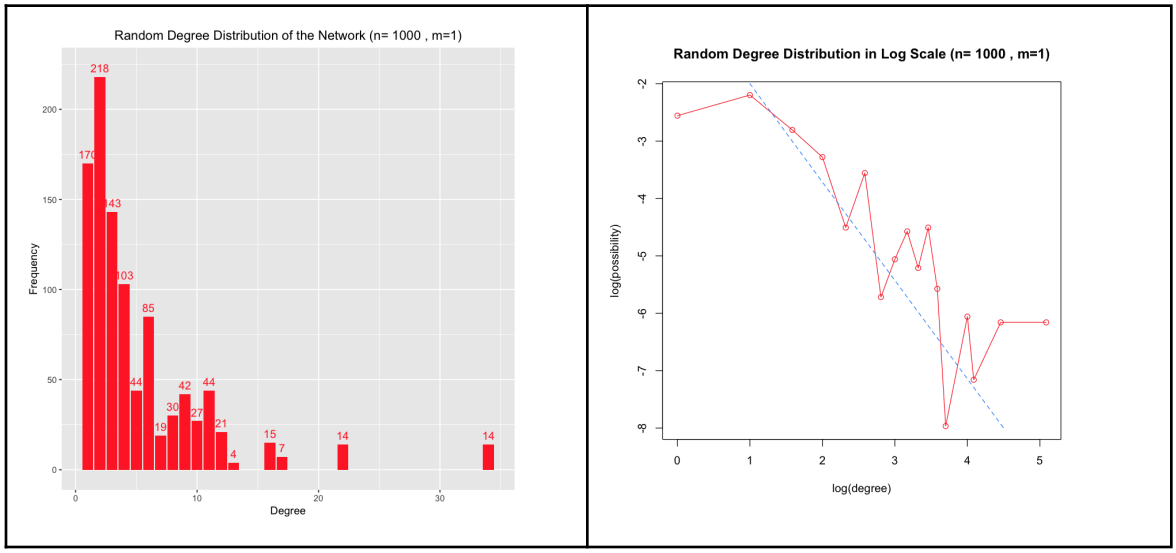
d) The preferential attachment model is a Power Law relationship, and we can derive the following linear relationship for the degree distribution with gamma being the slope.

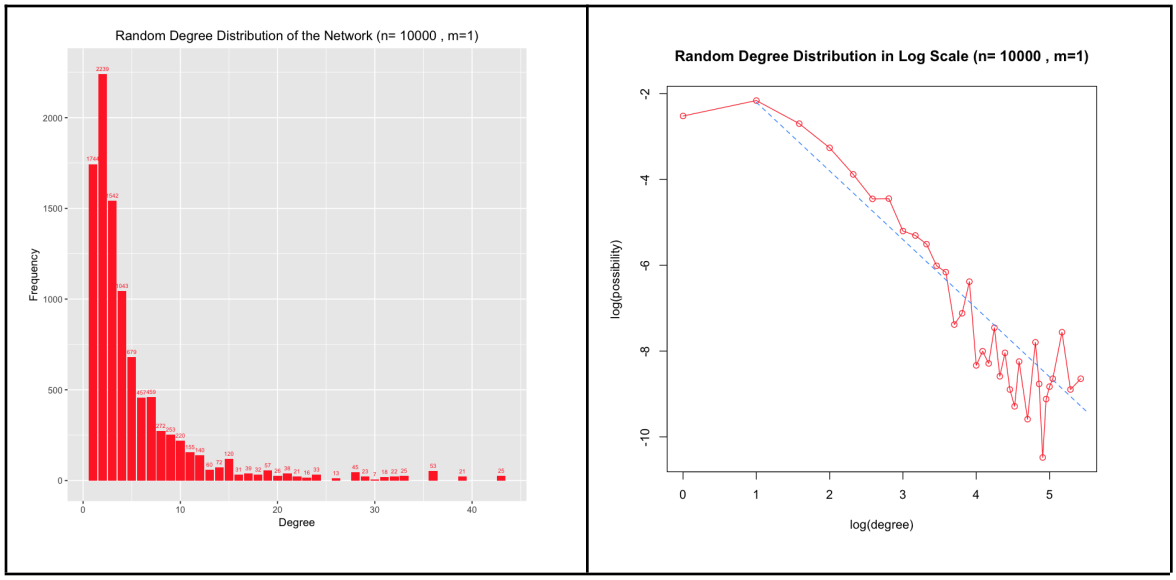
$$\log P_k = \gamma * \log k$$

We can estimate this gamma by plotting the degree distribution on a log-log scale and taking the slope of the linear regression line. The following two plots show the degree distributions, and our respective slopes for **n=1000 and 10000** was γ **= -2.26 and -2.77.**



e) We then perform a random sampling approach to obtain the degree distribution. There is a linear scale histogram and a log-log scale plot for both n=1000 and 10000 below. The **slope on each log-log plot was -0.973 and -1.64.** It is clear the initial peaks, or quantity of high degree nodes is lower when randomly selecting, as the likelihood of getting those nodes at random is much lower than the lower degree nodes.





f) In the plot below, we show the degree of a node vs its age. Each new node will come in with the same initial degree (m=1) in this case. The oldest nodes have had the longest time to accumulate higher degrees, and hence are more likely by nature of preferential attachment to further increase their degree each iteration. Hence we see an expected increase of exponential nature in the degree vs age of a node. The following image is our derivation for the expected degree of a node vs age, which we plotted against the analytical solution below.

Let k_i be the degree of node i, where node i joins the network at time t_i. i.e. $k_i(t_i) = m$.

$$\Rightarrow \frac{dk_i}{dt} = m \frac{k_i}{\sum_{j=1}^{N-1} k_j} \quad i = 1, 2 \ldots N-1. \text{ (at time } t)$$

(Increasing rate is proportional to the degree of the node.)

$$\Rightarrow \text{Since } \sum_{j=1}^{N-1} k_j = 2mt - m.$$

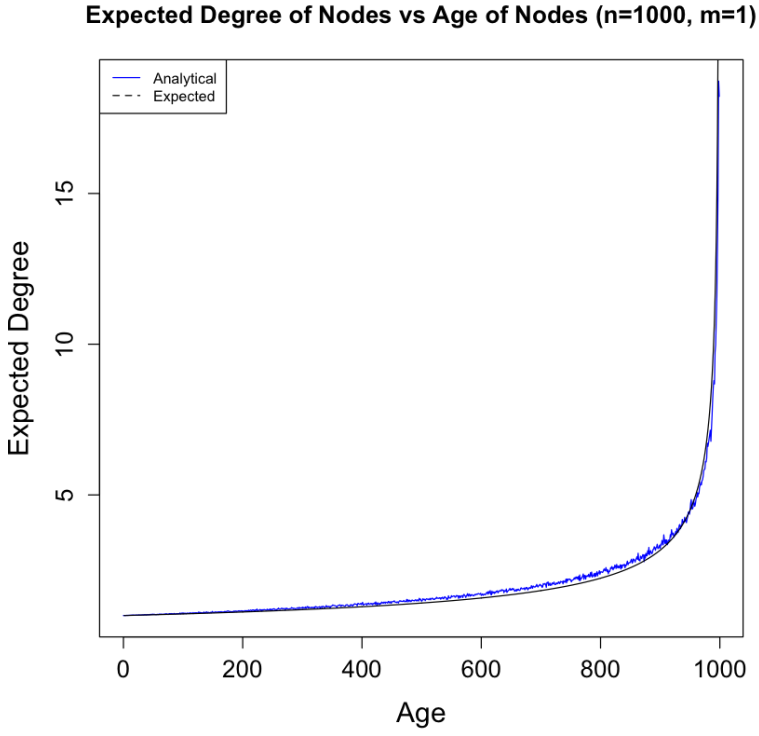
$$\Rightarrow \frac{dk_i}{dt} = \frac{k_i}{2t-1} \Rightarrow \frac{dk_i}{k_i} = \frac{dt}{2t-1}, \quad k_i(t_i) = m.$$

$$\ln(k_i) = \frac{1}{2}\ln(2t-1) + C$$

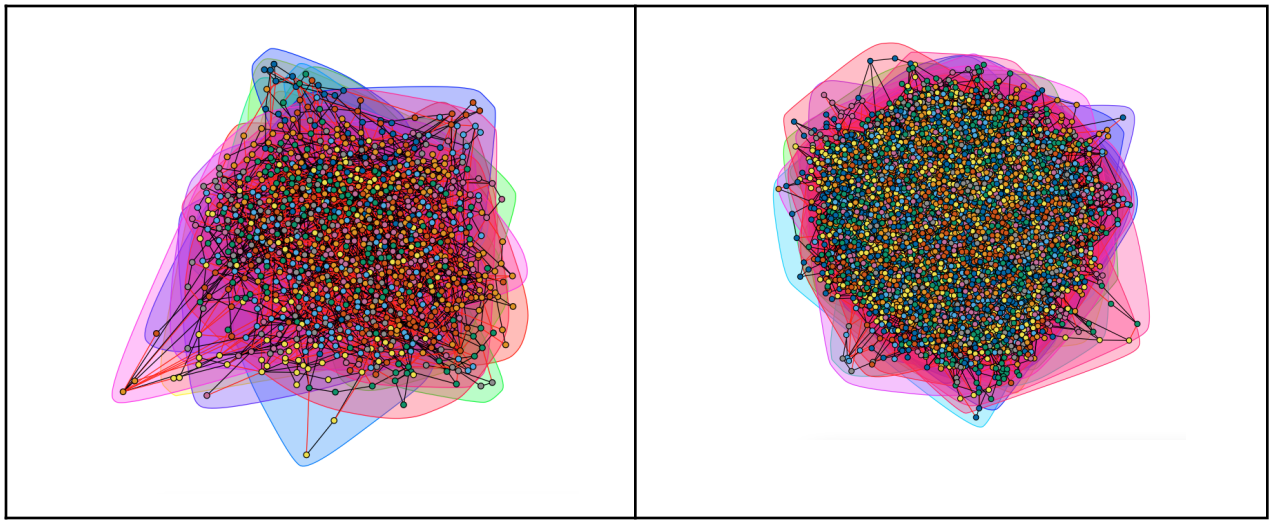
$$\ln(m) = \frac{1}{2}\ln(2t_i-1) + C \Rightarrow C = \ln\left(\frac{m}{(2t_i-1)^{\frac{1}{2}}}\right)$$

$$k_i = \frac{m(2t-1)^{\frac{1}{2}}}{(2t_i-1)^{\frac{1}{2}}} = m\left(\frac{2t-1}{2t_i-1}\right)^{\frac{1}{2}} \quad t_i = t - age.$$

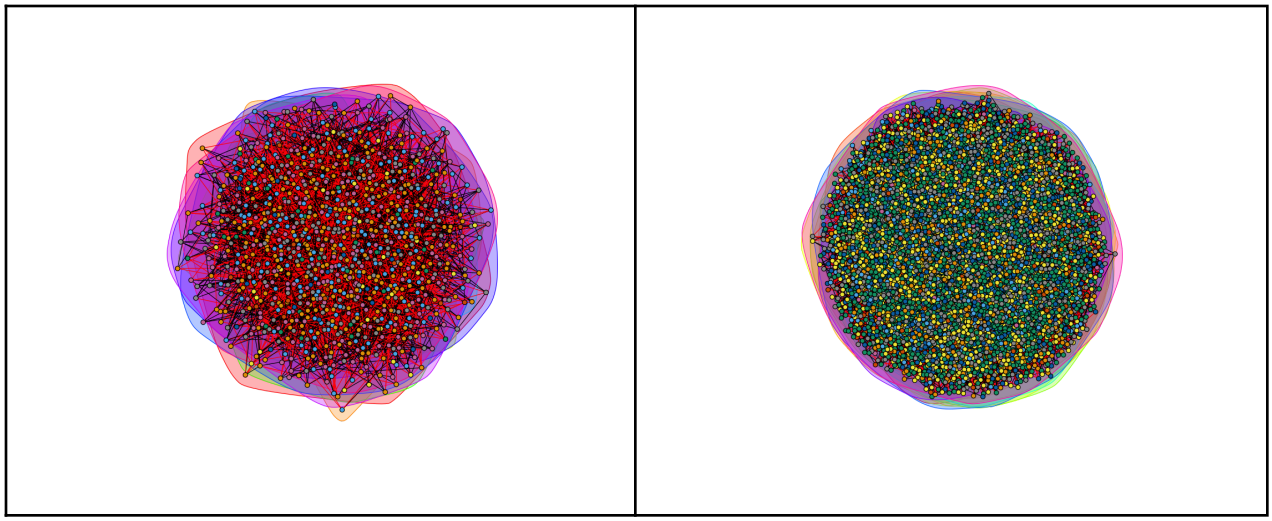
$$= m\left(\frac{2t-1}{2(t-age)-1}\right)^{\frac{1}{2}}$$



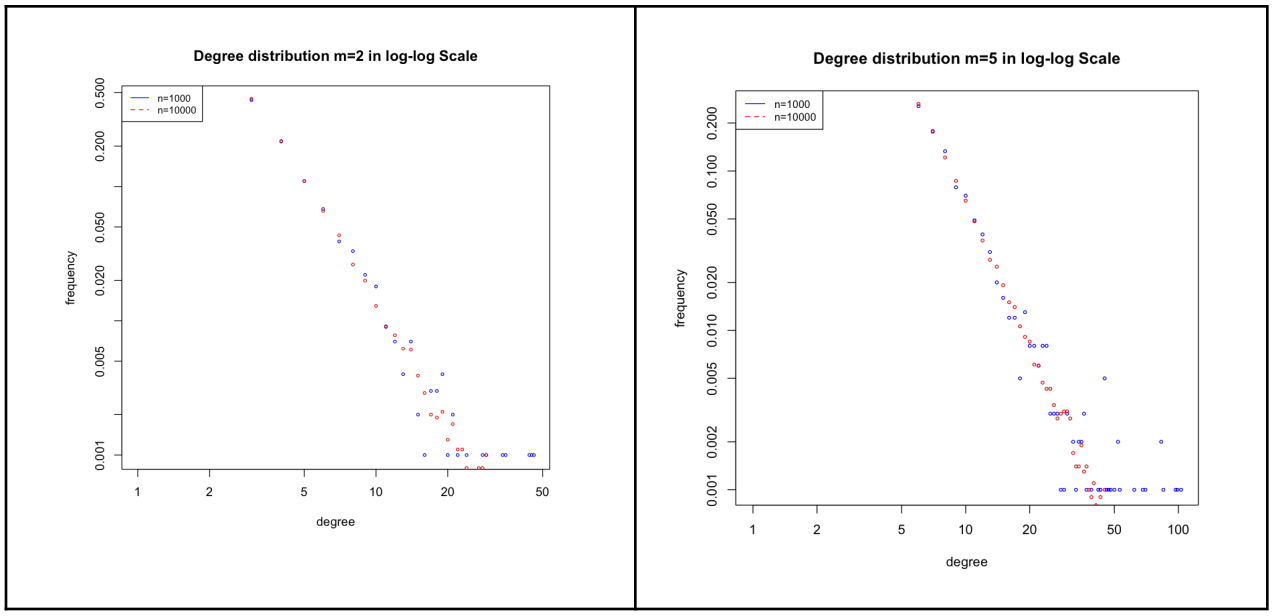
g) In this section, we perform the above analysis for m= 2 and 5, for both n=1000 and 10000. The plots highlight the major analysis, and the table below summarizes all the results for these networks.



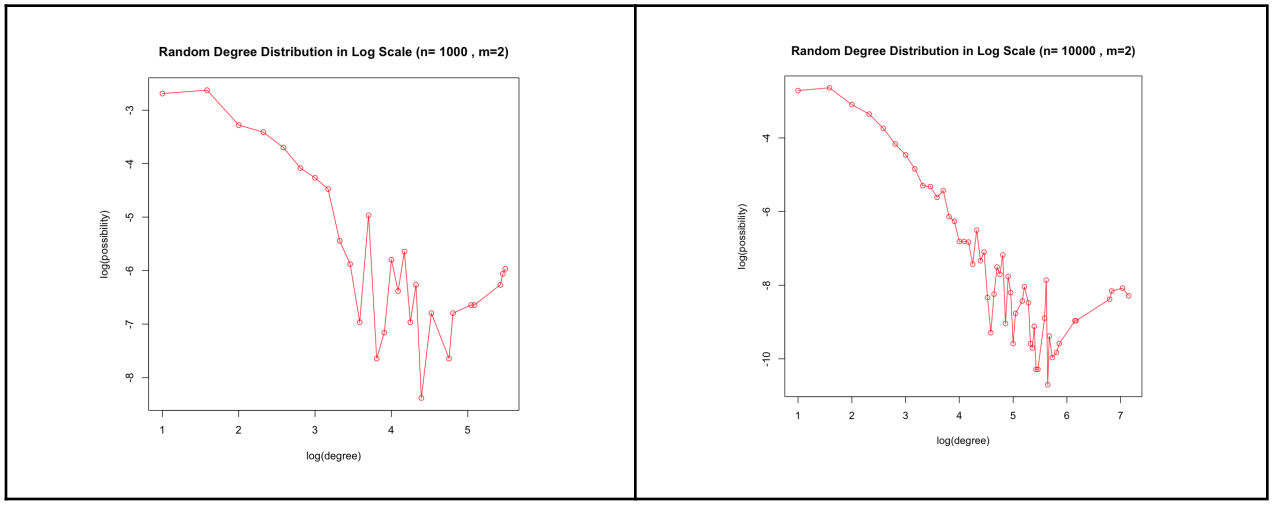
*Community Structure for **m=2**, n=1000 (left) and 10000 (right)*



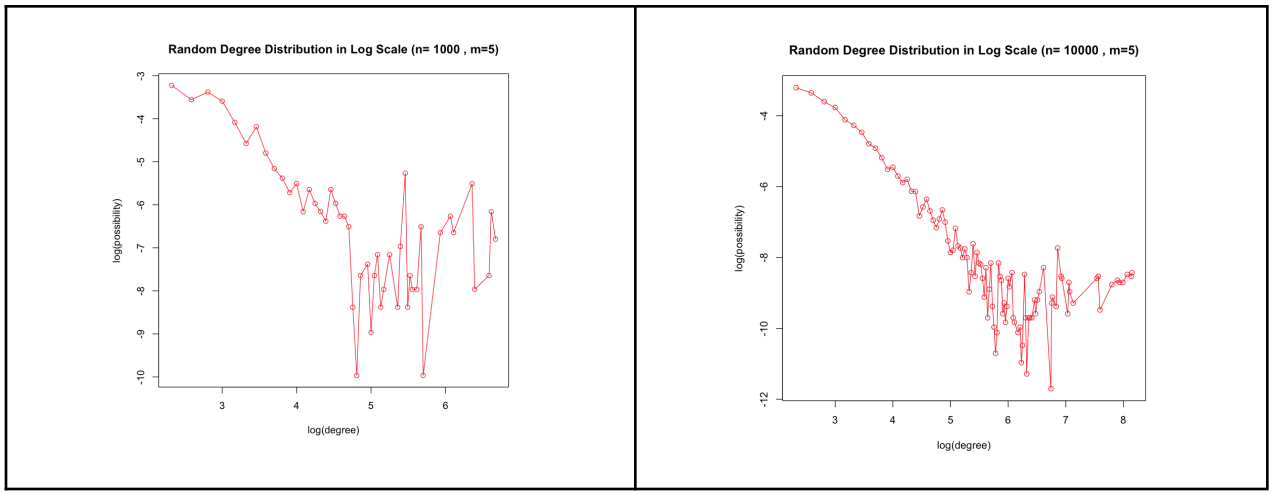
*Community Structure for **m=5**, n=1000 (left) and 10000 (right)*



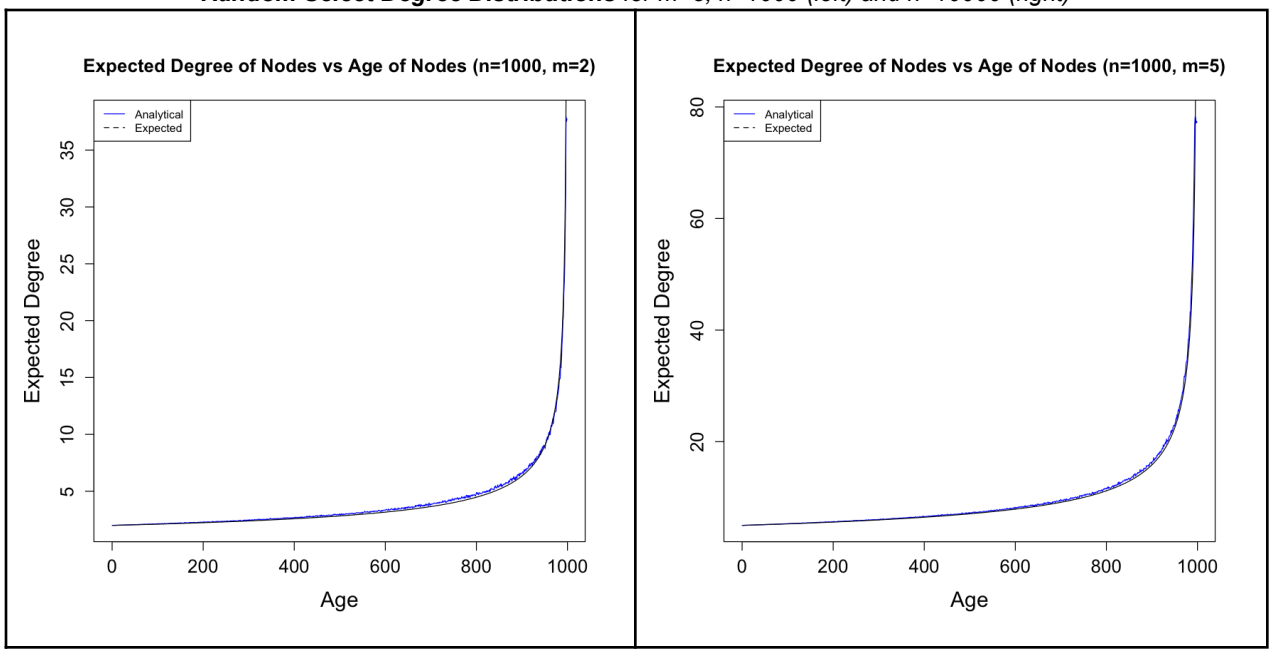
Degree Distributions *for m=2 (left) and m=5 (right)*



Random Select Degree Distributions *for m=2, n=1000 (left) and n=10000 (right)*



Random Select Degree Distributions *for m=5, n=1000 (left) and n=10000 (right)*



Age vs. Degree *for m=2 (left) and m=5 (right)*

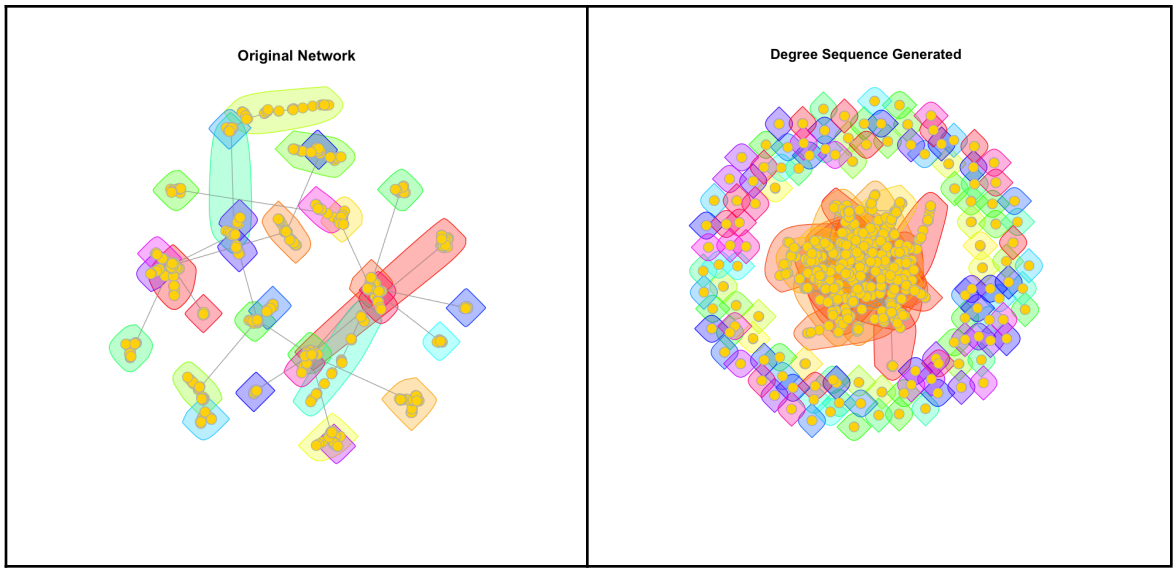
i

m	n	Modularity	Communities	γ/slope	γ/slope Rand Select
1	1000	0.934	33	-2.26	-0.934
	10000	0.978	115	-2.77	-1.64
2	1000	0.525	18	-2.41	-1.09
	10000	0.532	37	-2.50	-1.36
5	1000	0.283	9	-2.05	-1.01
	10000	0.274	16	-2.26	-1.11

We can summarize this analysis with the following observations:

- **Modularity decreases as m increases** and the communities are more likely to become interconnected into larger communities. Also, the **modularity increases slightly with n across all m's**, as this reinforces the high degree nodes.
- The **age of node vs degree is similar for all m's** in terms of shape and relationship, but the **value scales up with m** as expected.
- The **gamma/slope of the power law relationship fluctuates** with m but seems to be in the same order **of about ~2.5 for the full distribution and ~-1.2 for the randomly selected** nodes. As **m increases, the number of higher degree nodes increases** and the **tail of the distribution gets "noisier**" or there is less resemblance to a linear relationship in the log-log plot.

h) In this final portion of part 2, we make a stub-matching network using the same degree distribution as a PA generated network. The **modularities and community counts** are **0.931 with 34 communities and 0.838 with 145 communities** for the original and degree sequence generated network. It is clear the **stub-matching method is likely to leave an unconnected graph** with m=1, and this will impact the community structure with many isolated nodes as can be seen in the plots below.

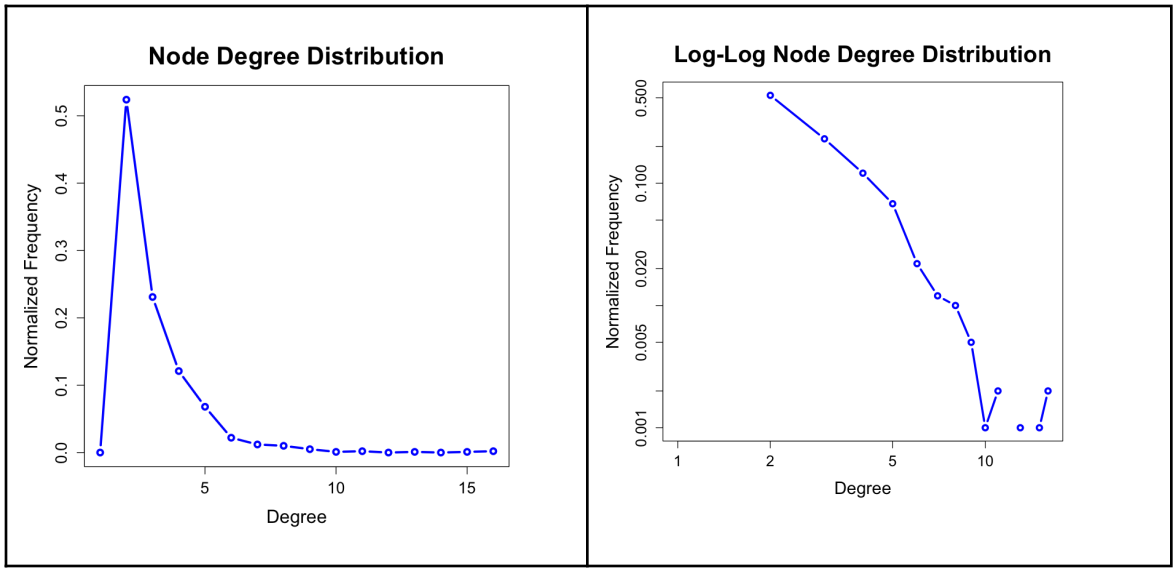

Original Network
Degree Sequence Generated

Problem 3

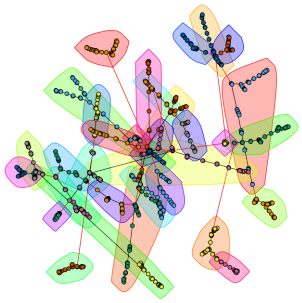
In the final problem of Part1, we explore the PA graph again, except with the added penalty for the age of a node. This was done using the “sample_pa_age” function in r, with the requested parameters in the handout: m = 1, α=1,β=−1,and a=c=d=1,b=0. With these parameters the formula is:

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

a) For n=1000, and m=1, ***γ= -3.35***. This distribution is similar but the slope is higher than the original network with no age preference.



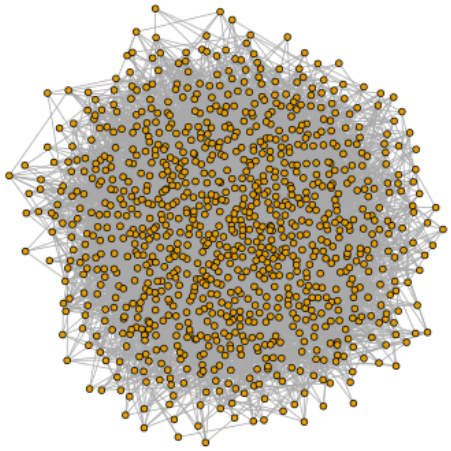
b) Again using the fast and greedy methods we find a **modularity of 0.935 and 33 communities**. This is very similar to the original PA model. The community structure is pictured below.



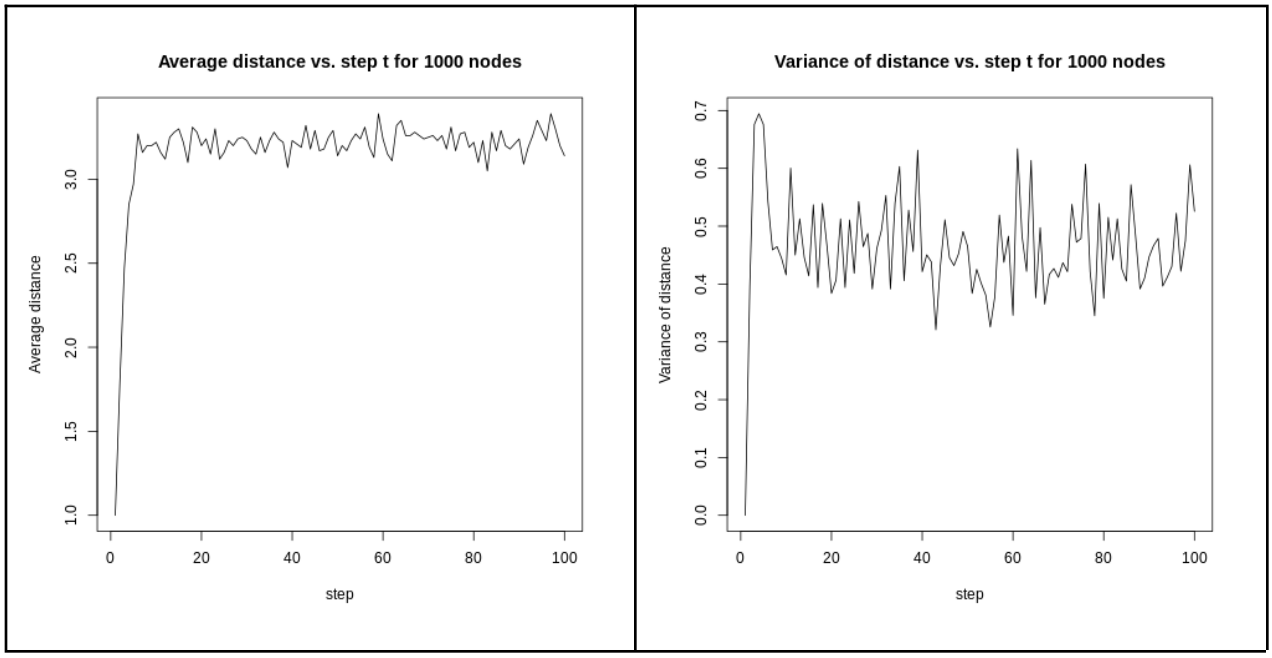
Part II: Random Walk on Networks

Problem 1

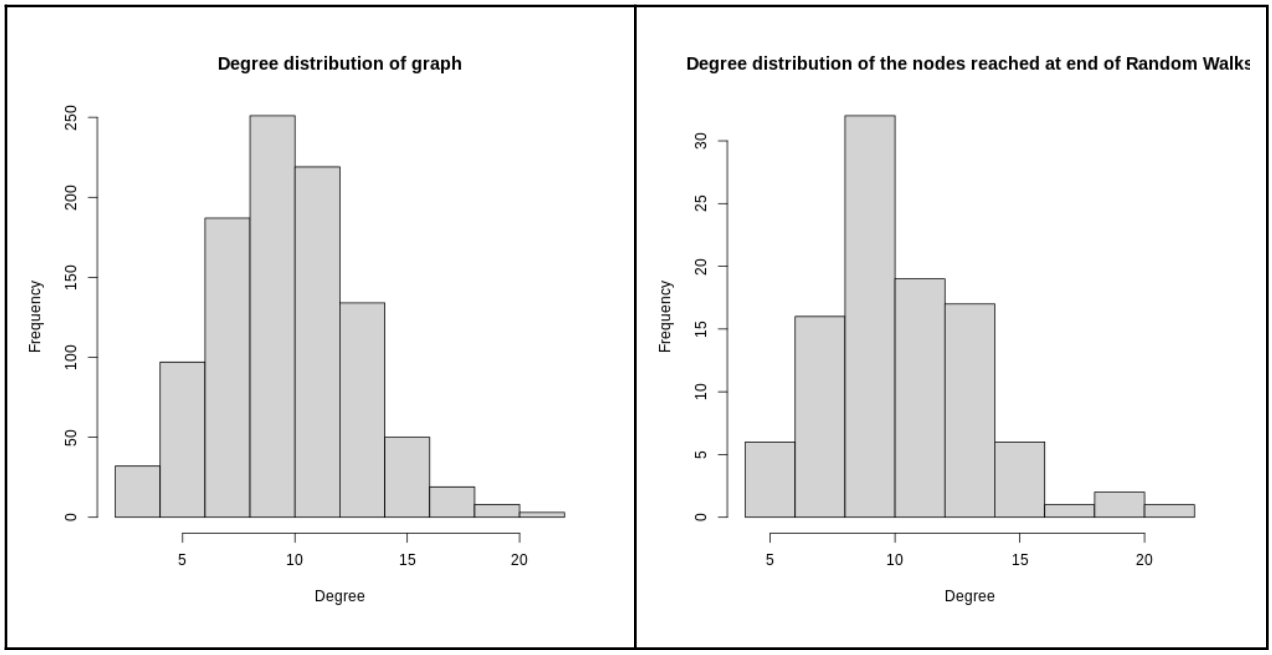
a) We created an undirected random network with 1000 nodes and probability p equals to 0.01. The graph of this network is shown below:



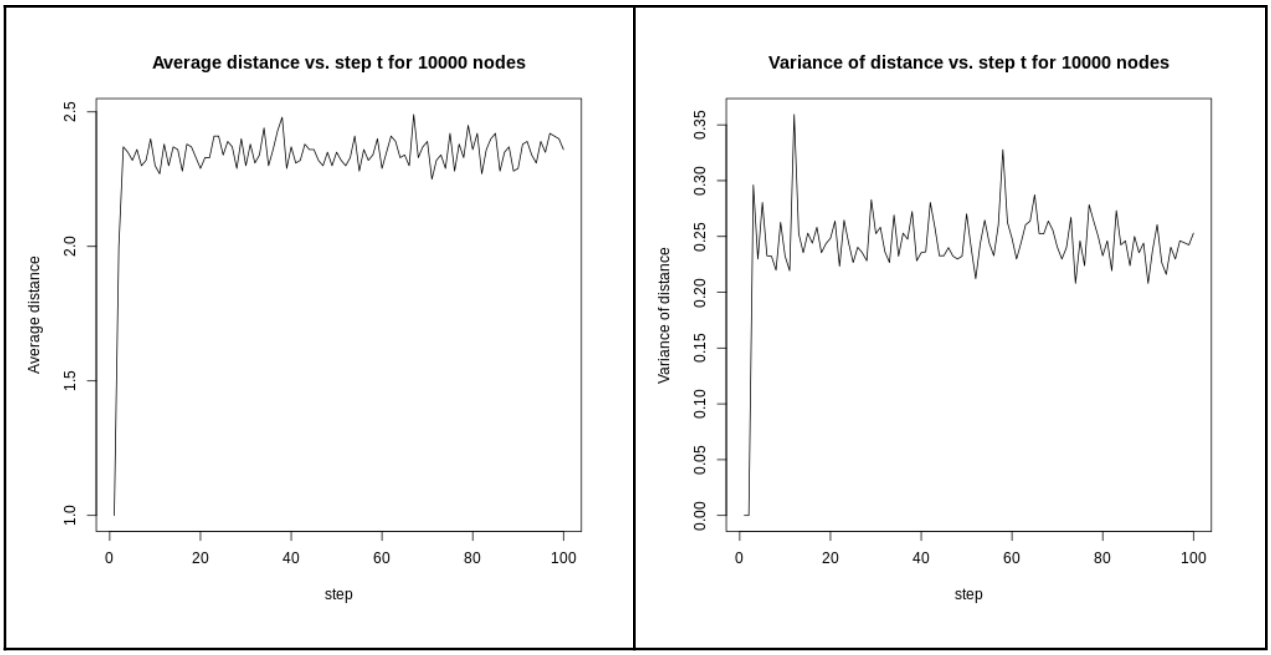
b) The plot on the left shows the average distance of the walker from his starting point at step t and the plot on the right shows the variance of distance at step t for undirected random network with 1000 nodes. The **average distance is around 3.5** and **variance of distance between range 0.3 to 0.7**. We approach these steady states after about 10 steps.



c) The degree distribution of the nodes reached at the end of random walks is shown on the right below. It shows a similar trend to the degree distribution of the graph, which is shown on the left below. **Both degree distributions are binomial**. This is true because for Erdos-Renyi model, the degree distribution of is $P(deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$ since each vertex can be connected to n-1 other vertices. For **n that are large enough**, the degree distribution of the graph approaches the binomial distribution.

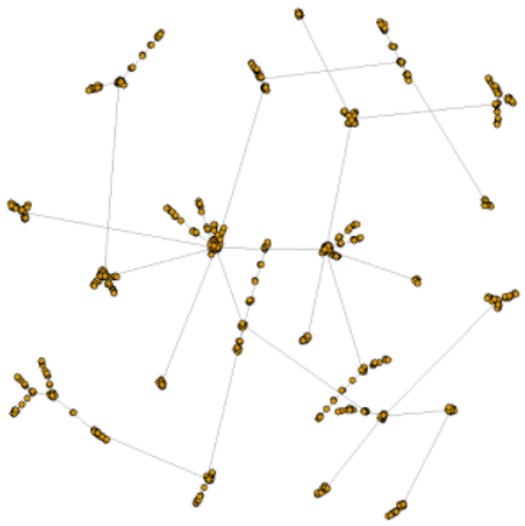


d) The plot on the left shows the average distance of the walker from his starting point at step t and the plot on the right shows the variance of distance at step t for undirected random network with 10000 nodes. The **average distance is around 2.4** and **variance of distance between range 0.2 to 0.4**. It is clear that the random network with 10000 nodes **has smaller average distance** and smaller variance compared to the random network with 1000 nodes. The random network with 10000 nodes also **converges quicker and has less fluctuation** in average and variance of distance than the random network with 1000 nodes.



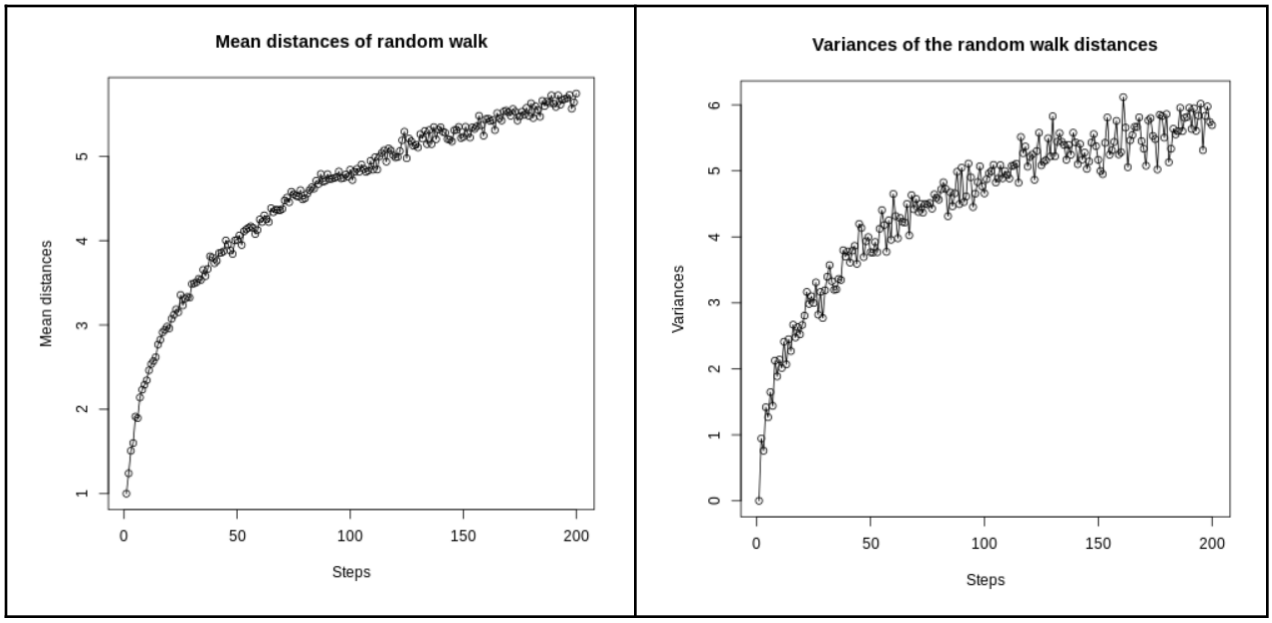
Problem 2

a)
We created an undirected preferential attachment network with 1000 nodes, each node connects to m=1 old nodes. We also find that the diameter of the graph is 18.
The graph of this network is shown below:



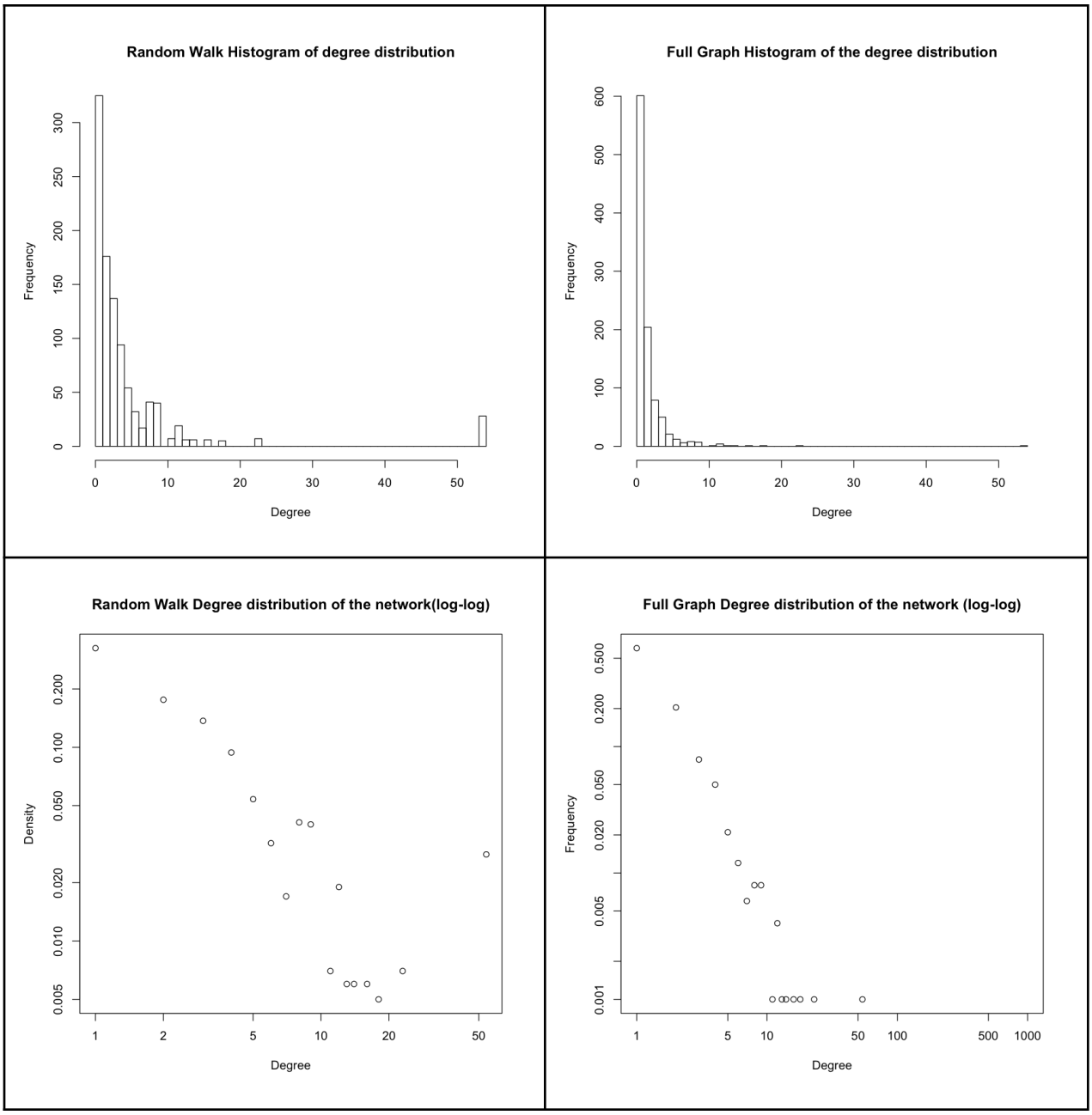
b) We let a random walk start from a randomly selected node and plot the average and variance v.s. the step t. We choose an iteration time of 1000 and the step range to be 200. **The result is different from 1(b) in part II**. The indegree for each node in the **preferential attachment**

network follows the power law distribution, while the Erdös-R´enyi network follows the normal distribution.



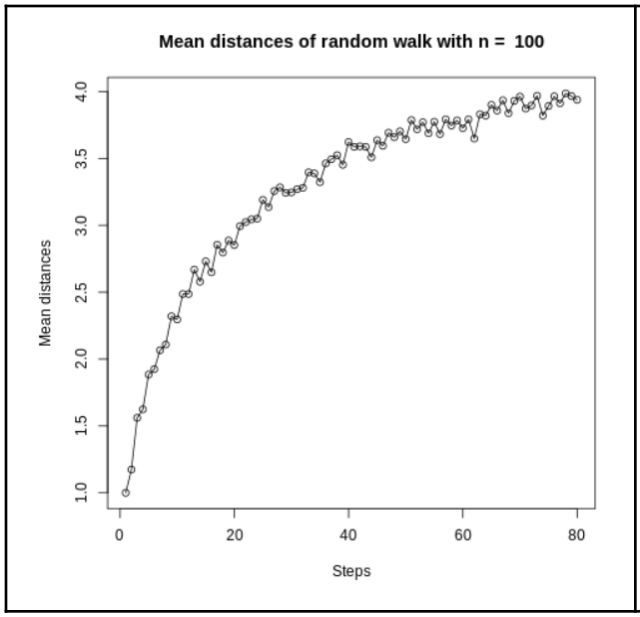
c) Comparing both the histogram of degree distribution, we may conclude that the degree distribution of the nodes reached at the end of the random walk (left part) depends on the degree distribution of the original network(right part). **Also, both degree distributions follow the power law distribution as shown in the log-log graph**.

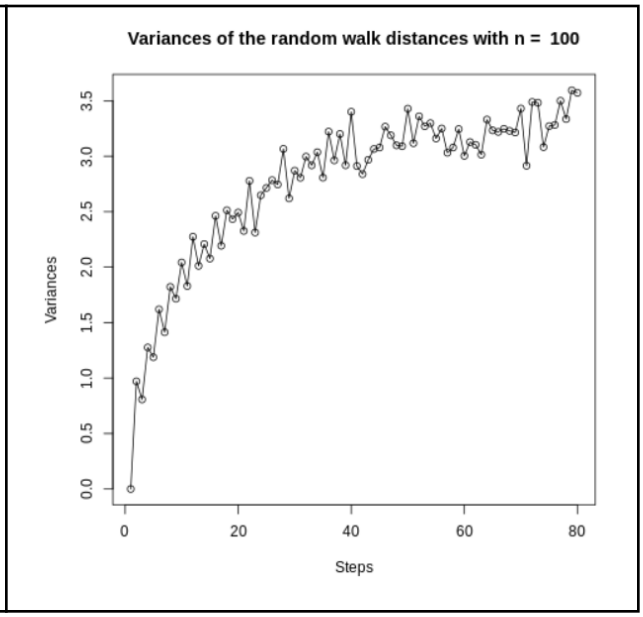
They have slightly different slopes however, **with a smaller slope in the random walk graph**. We think that the reason is that **our number of steps isn't enough** (we just run through 200 steps in this case). If we run more steps we believe the walker will eventually reach a steady state.


Random Walk Histogram of degree distribution
Frequency
0
50
100
150
200
250
300
0
10
20
30
40
50
Degree
Full Graph Histogram of the degree distribution
Frequency
0
100
200
300
400
500
600
0
10
20
30
40
50
Degree
Random Walk Degree distribution of the network(log-log)
Density
0.005
0.010
0.020
0.050
0.100
0.200
1
2
5
10
20
50
Degree
Full Graph Degree distribution of the network (log-log)
Frequency
0.001
0.005
0.020
0.050
0.200
0.500
1
5
10
50
100
500
1000
Degree

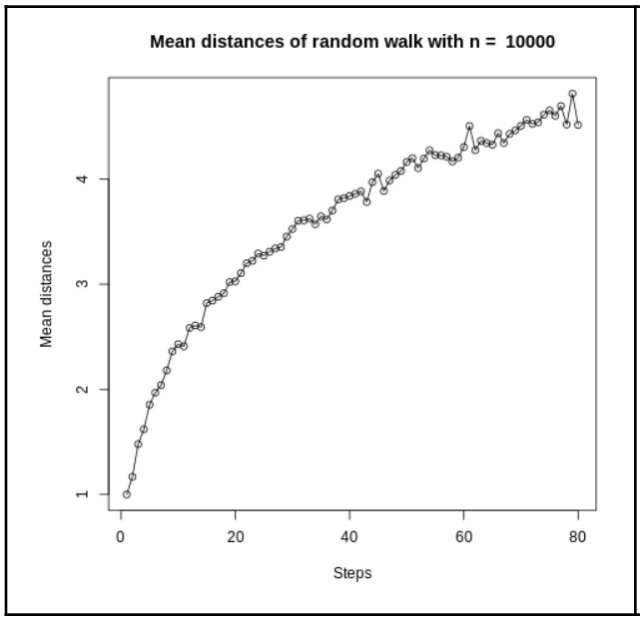
d) In part d of this problem, we repeated 2(b) for n=100 and n=10000.

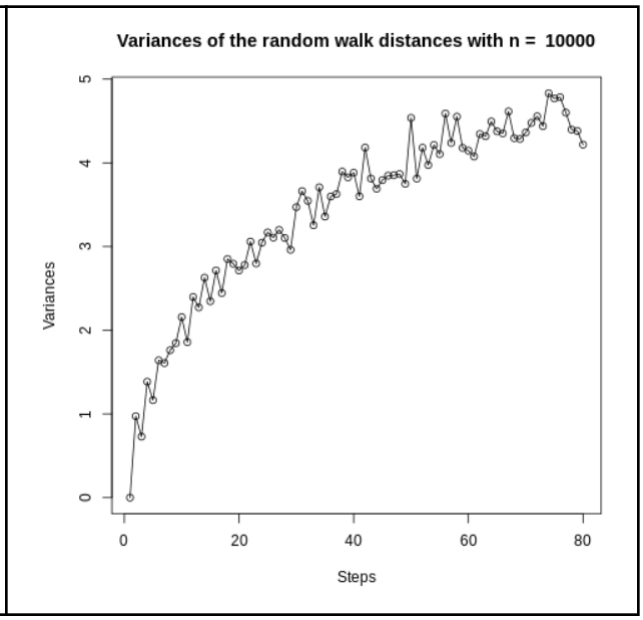
n=100





n=10000

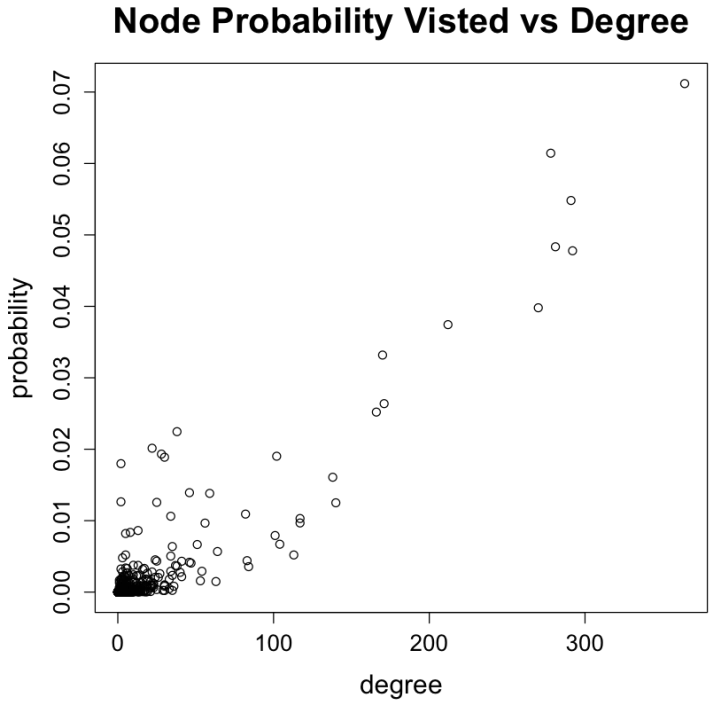




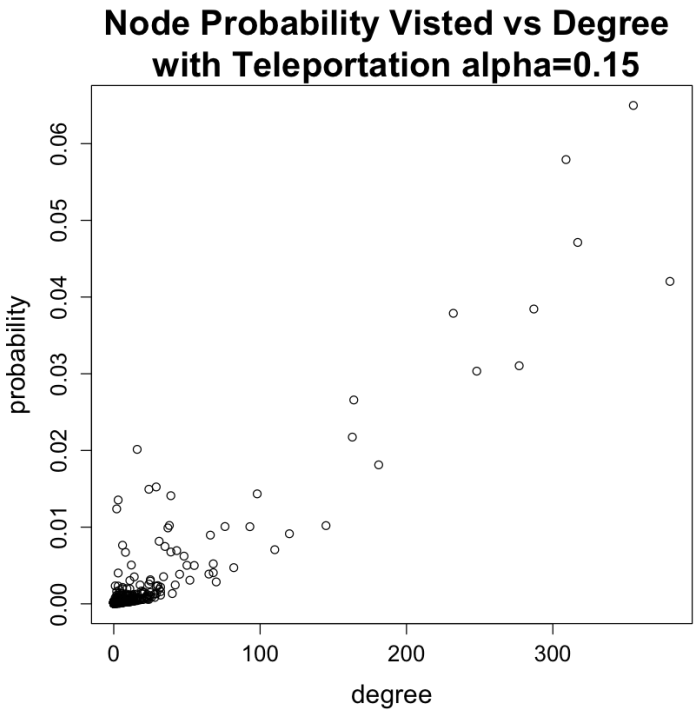
Comparing the result from 2(b), we can discover that the size of the graph will affect the number of steps required to reach to get stable. **With a larger n, the walker needs to use more steps to reach the stable state**. Also, **the variance is bigger for the larger graph** than the smaller graphs under the same number of steps.

Problem 3

a) As shown in the figure, the **possibility of visiting nodes is highly correlated** to the in-degree of nodes. To get an exact measure, the **Pearson correlation** coefficient between the two is **0.92**. This makes sense since the more in-degree of the node, the more random paths that go into the node. Additionally, by observing the distribution of visiting probability, we know that the most of the nodes are unvisited. This is similar to the practical case in web browsing - some popular websites get the majority of visitors.

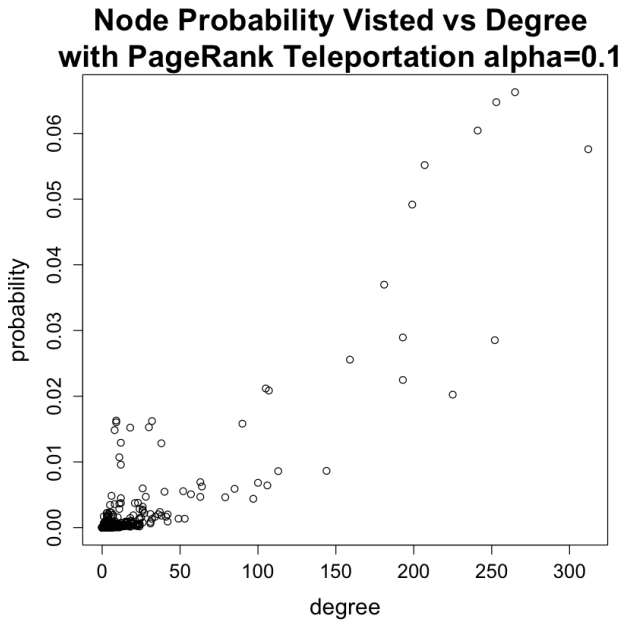


b) As shown in the figure, adding the **teleportation helps the nodes with few in-degrees get more visits**. The **correlation coefficient remains high, (0.93)**, but the **slope of the graph is lower** indicating. This can be seen in the plot scale and the probabilities of the nodes with in-degree less than 50. Teleportation is helpful for newly created websites, which normally have less in-degrees due to the nature of the graph of the internet (Preferential Attachment). **By introducing teleportation, we can help newly created websites get more visitors** as they otherwise would.

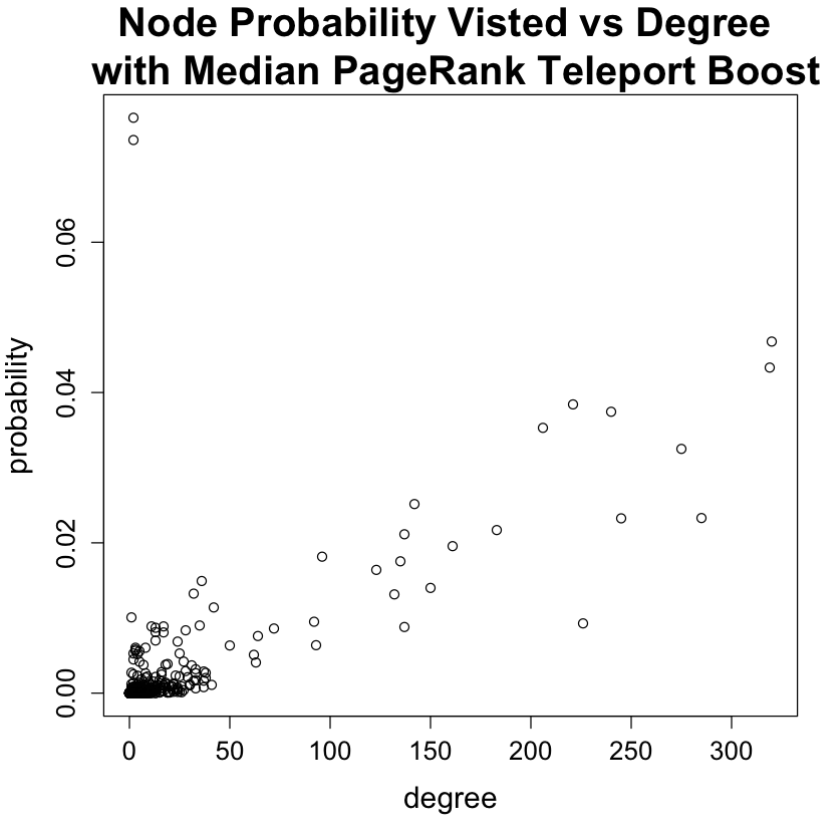


Problem 4

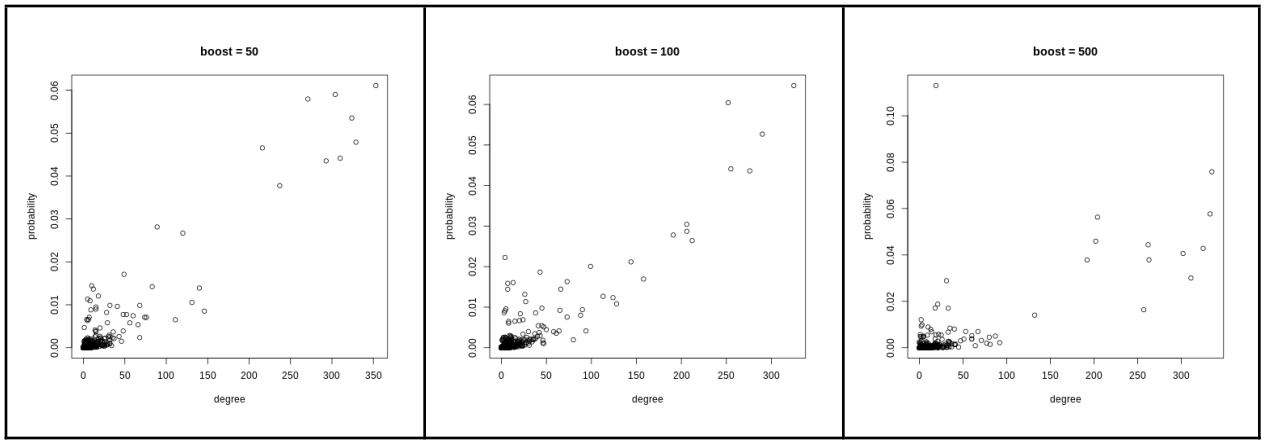
a) As discussed in the previous questions, teleportation can help the newly created websites get more visitors and survive. However, the distribution of teleportation to each website is not possible uniformly distributed. The distribution should be affected by personal interests, current events and the relationships between nodes. To include those effects, some researchers propose PageRank, which is one of the web ranking indicators for Google and other search engines. **As shown in the figure below, a few nodes significantly get more visits due to the PageRank**. This makes more sense to us, since in the practical case, those **commonly interested websites** should get more visits **although it has few in-degree**. **PageRank hits a balance between uniform teleportation and no teleportation.**



b) As shown in the figure, the nodes **with median PageRanks get significantly more visits**. However, **the rest of the nodes seem unaffected**. There is likely a slight effect on nodes adjacent to the two median nodes. But the number of visits is dominated by the random walk and only slightly impacted by the teleportation change.



c) To include personal interests into teleportation, **we randomly chose five nodes and scaled up their probabilities of teleportation**. The five chosen nodes represent the websites, which are of common interest to certain groups of people. As shown in the figures, we select different scaling numbers, 50, 100 and 500, to control the boosting effect. **The larger the number, the greater possibility these low degree nodes could be visited**. Based on the design, it is shown that the chosen nodes can get more visits as the boosting scale goes higher. **The boost value will allow a designer to turn a knob and control this feature of the network.**



References Links

https://www.geeksforgeeks.org/erdos-renyl-model-generating-random-graphs/

https://arxiv.org/pdf/cond-mat/0408187.pdf

https://web.stanford.edu/class/msande235/erdos-renyi.pdf