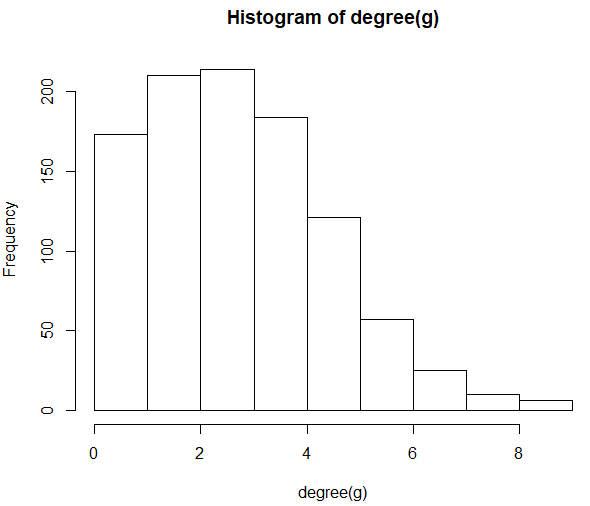
**ECE 232E Project 1**

Group members: Kushagra Rastogi (304640248)

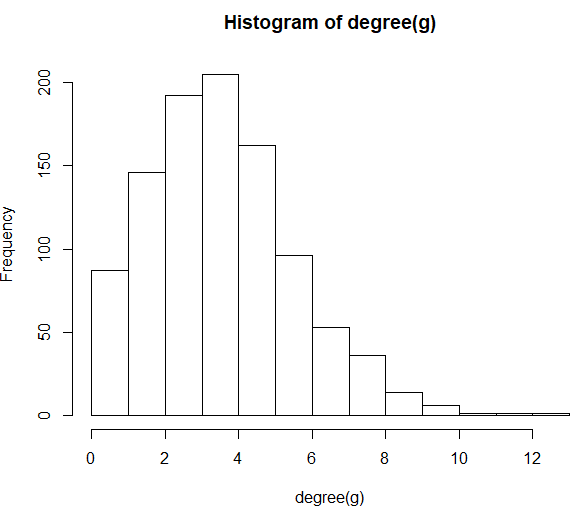
Jonathan Lee (104840173)

**Generating Random Networks**

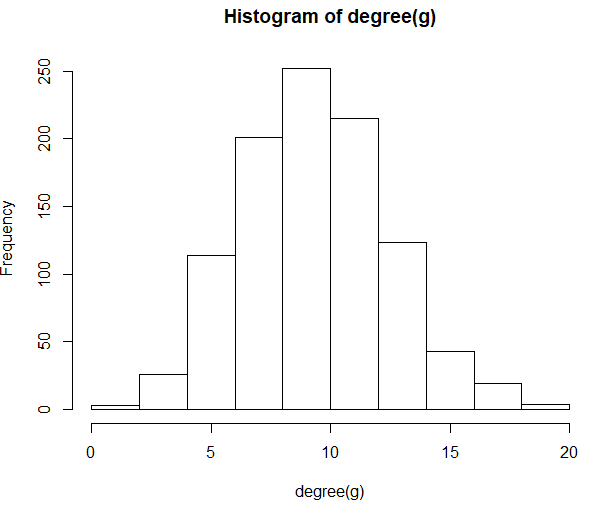
1. **a)** Plot for p = 0.003



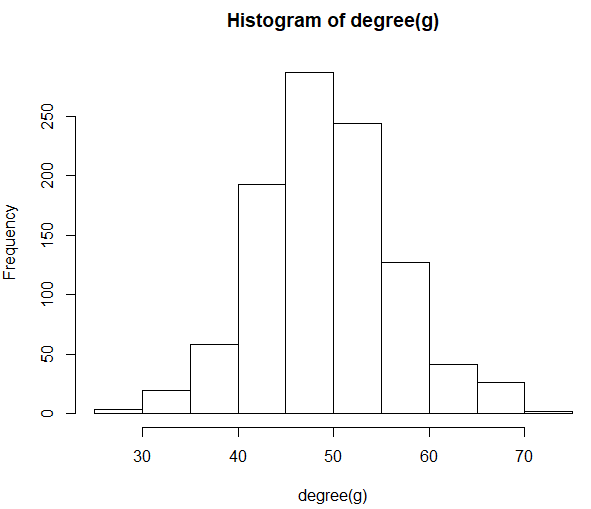
Plot for p = 0.004



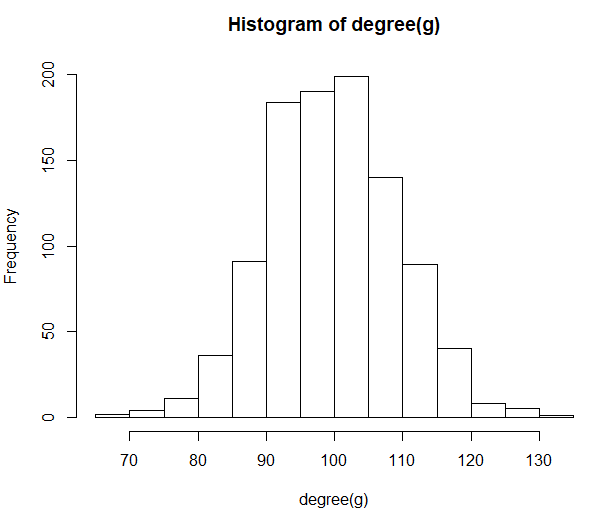
Plot for p = 0.01



Plot for p = 0.05



Plot for p = 0.1



The distribution observed is a Binomial distribution because the probability of an edge existing between a pair of nodes can be modelled by a Bernoulli random variable with probability . The mean of Binomial distribution is equal to and the variance is equal to . For , we get

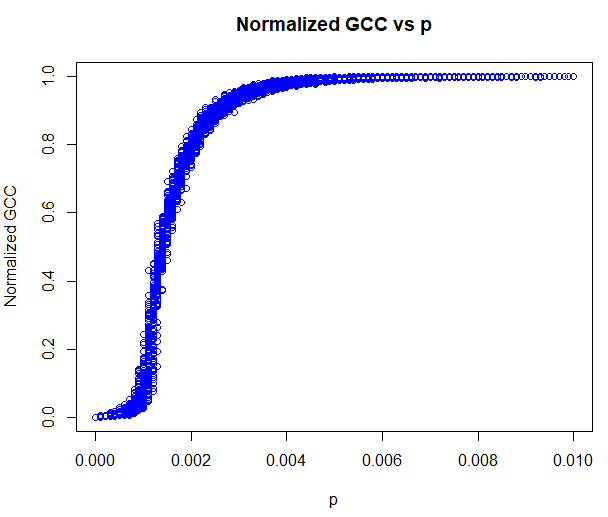
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Empirical | | Theoretical | |
| P | Mean | Variance | Mean | Variance |
| 0.003 | 3.18 | 3.154755 | 2.997 | 2.988 |
| 0.004 | 4.026 | 4.033357 | 3.996 | 3.98 |
| 0.01 | 9.844 | 9.427091 | 9.99 | 9.89 |
| 0.05 | 49.766 | 50.28353 | 49.95 | 47.45 |
| 0.1 | 100.224 | 90.04787 | 99.9 | 89.91 |

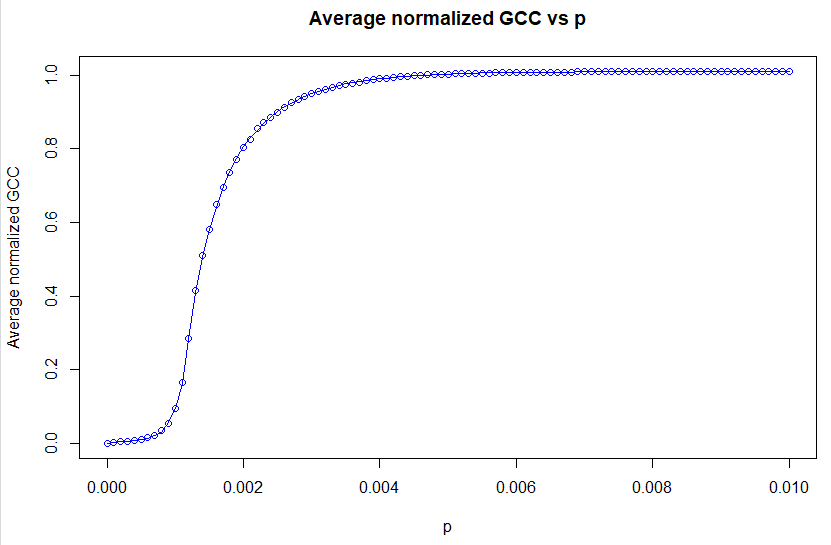
It can be seen from the table above that the empirical and theoretical values match quite well. As increases, the degree of the graph increases and the relative error between the empirical and theoretical values decreases. In fact, the difference between theoretical and empirical values is within 10% error.

**b)** All the random realizations of ER network are not connected. We generated random ER network with for 100 times and found the probability that the network was connected.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | Connected? | Probability of connected | Size of GCC | Diameter of GCC |
| 0.003 | No | 0 | 936 | 14 |
| 0.004 | No | 0 | 982 | 11 |
| 0.01 | Yes | 0.96 | 999 | 6 |
| 0.05 | Yes | 1 | N/A | N/A |
| 0.1 | Yes | 1 | N/A | N/A |

**c) (i)** There are two interesting values of which are and . Then let .

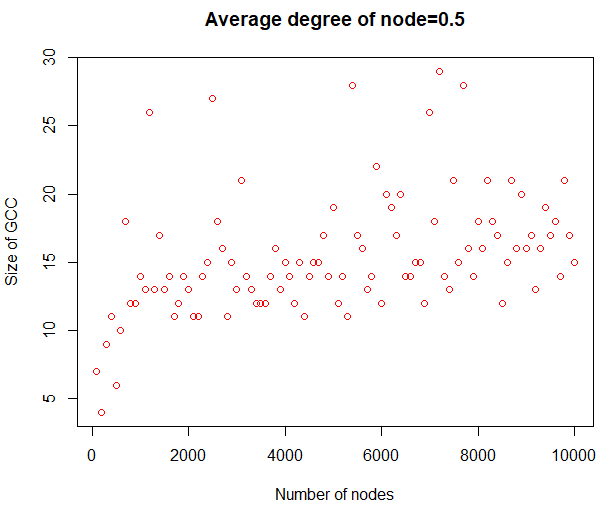




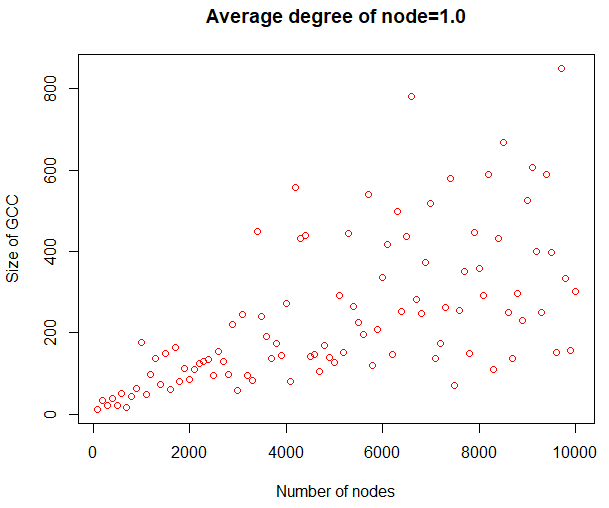
I define ‘emergence’ as the value of for which the normalized GCC is starting to become non-zero consistently. A giant connected component starts to emerge when . This value matched with the theoretical values derived in lecture.

**(ii)** The giant connected component takes over 99% of the nodes when . This matches with the two interesting values of that were derived in the previous part.

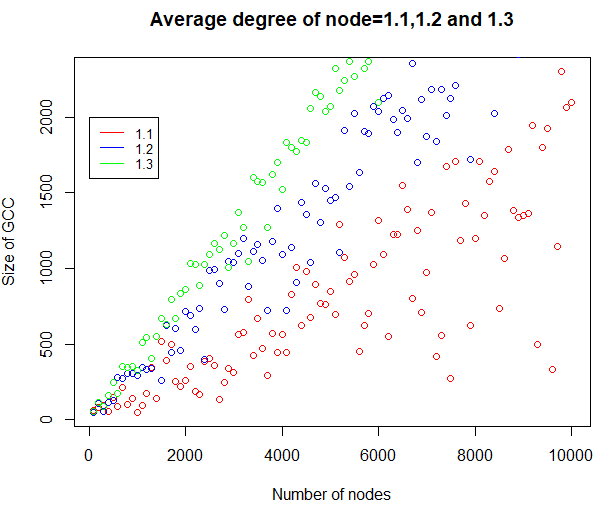
**d) (i)** As the number of nodes increases, the size of GCC increases. When , the slope of the GCC grows exponentially and when , the GCC grows linearly. This makes sense because GCC increases as increases while keeping constant since . Thus, more nods are connected and the GCC becomes larger.



**(ii)** For = 1



**(iii)** For = 1.1, 1.2, 1.3



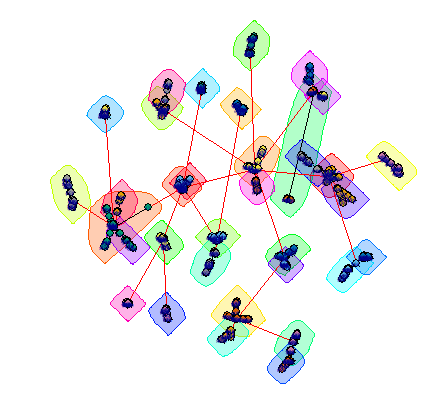
**(iv)** In each case, there is roughly an asymptotic linear relationship between the expected GCC size and which can be modelled as where . As increases, increases and thus GCC increases because more nodes are connected.

1. **a)** Yes, such a network is always connected because a new node always gets connected to an old node in a preferential attachment model.

**b)** Mathematically, modularity can be defined as

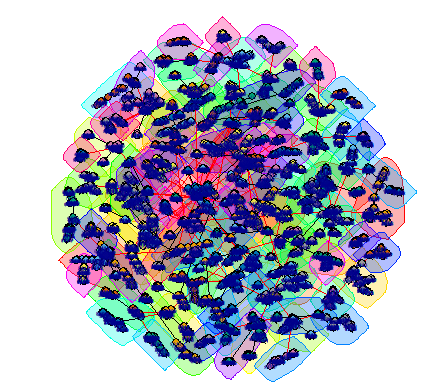
Where is the fraction of edges with one end vertices in community and the other in community and .

The community structure is



Modularity = 0.9330437

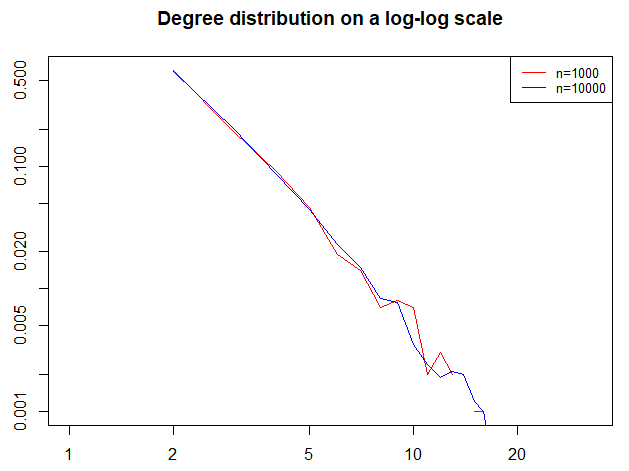
**c)** The community structure is



Modularity = 0.9780596

The modularity of the bigger network is larger than the smaller network. This is because in bigger networks, there are more communities clustered together since more nodes have higher degree.

**d)** Degree distribution

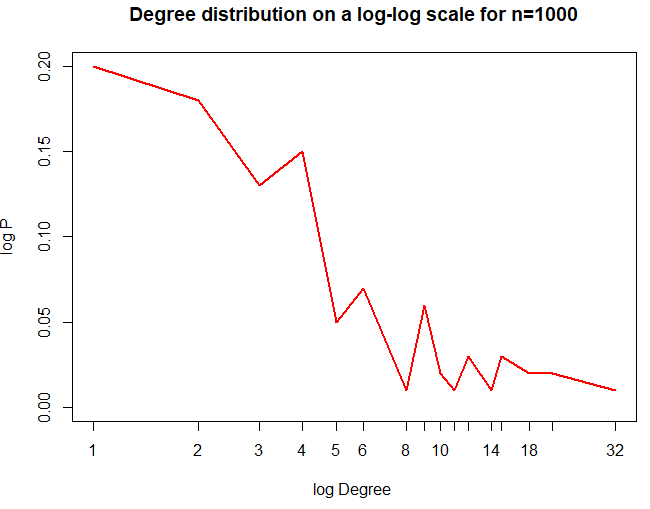


A preferential attachment model follows the power law with the degree distribution:

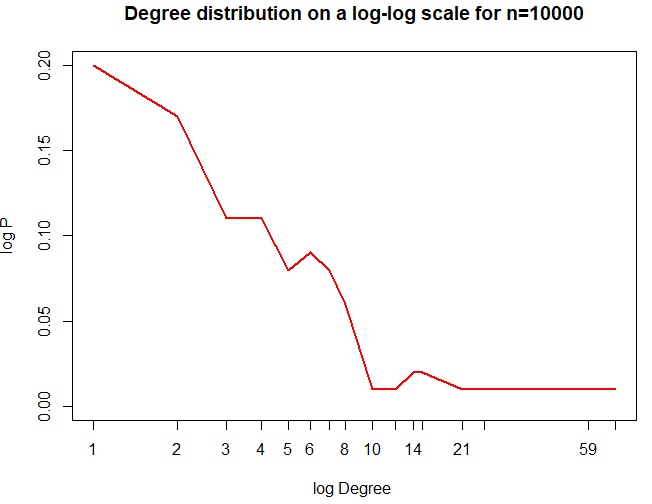
Taking log on both sides we get

Thus, the slope for is -3.242005 and the slope for is -3.418408 using linear regression. This matches with what we see in lecture since the degree of a preferential attachment varies with .

**e)**



The distribution is roughly linear. Using linear regression, the slope is -2.049911.



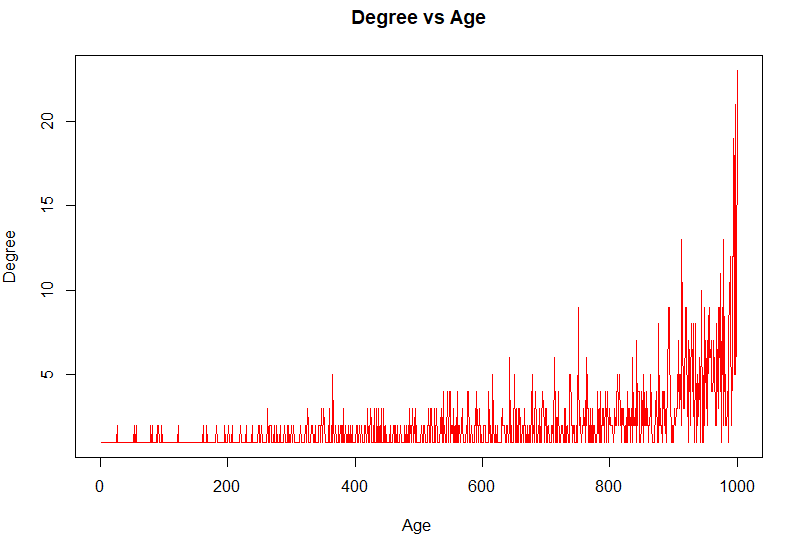
The distribution is roughly linear. Using linear regression, the slope is -2.111604.

The slopes of these randomly-picked-nodes networks is lower than the slopes of the node degree distribution for both values of . The table below compares both distributions

|  |  |  |
| --- | --- | --- |
|  | Node degree distribution | Random picking degree distribution |
| 1000 | -3.242005 | -2.049911 |
| 10000 | -3.418408 | -2.111604 |

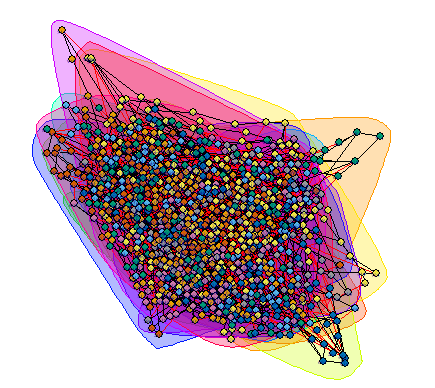
The table indicates that node degree distribution is sharper and decreases at a faster rate than the random picking degree distribution. This means the degrees are more evenly distributed in the latter distribution.

**f)**

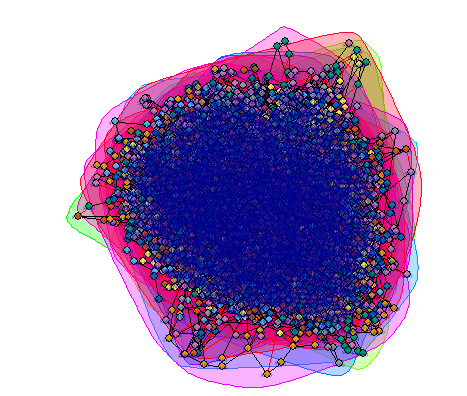


As the age of the nodes increases, the degree of the nodes increases exponentially which displays the power law in random networks

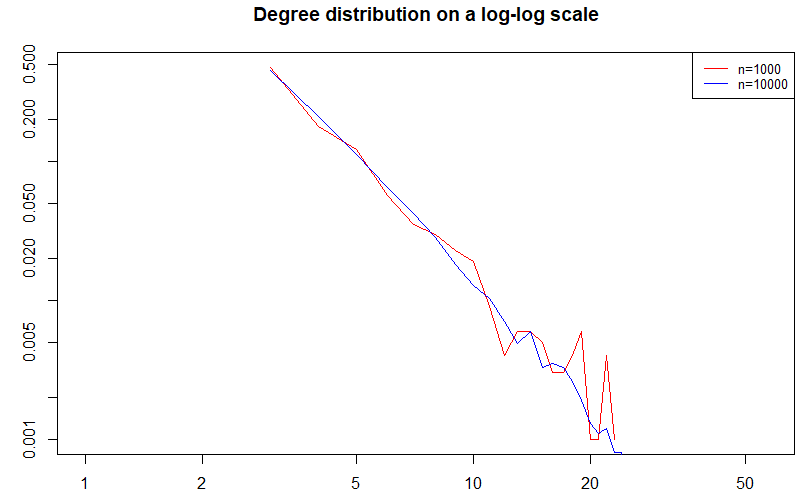
**g)** For , the network is always connected. We get the following plots



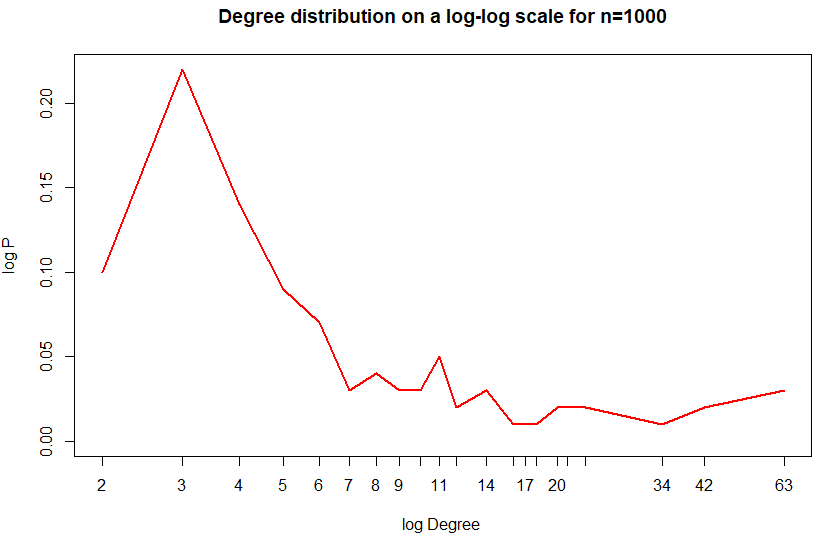
Modularity with 1000 nodes = 0.5255179



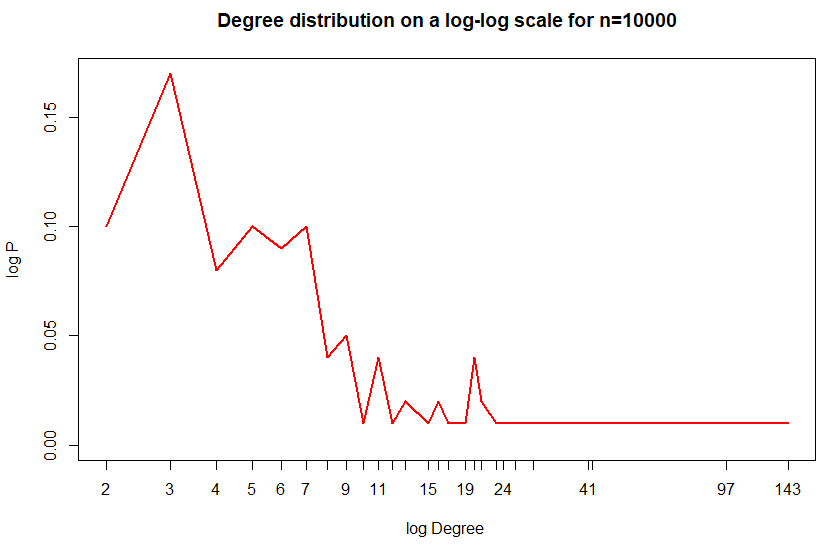
Modularity with 10000 nodes = 0.532126. The modularity for the larger network is only marginally larger than the small network. If we take the randomness into consideration, then we can see that the modularity of both the networks is essentially the same.



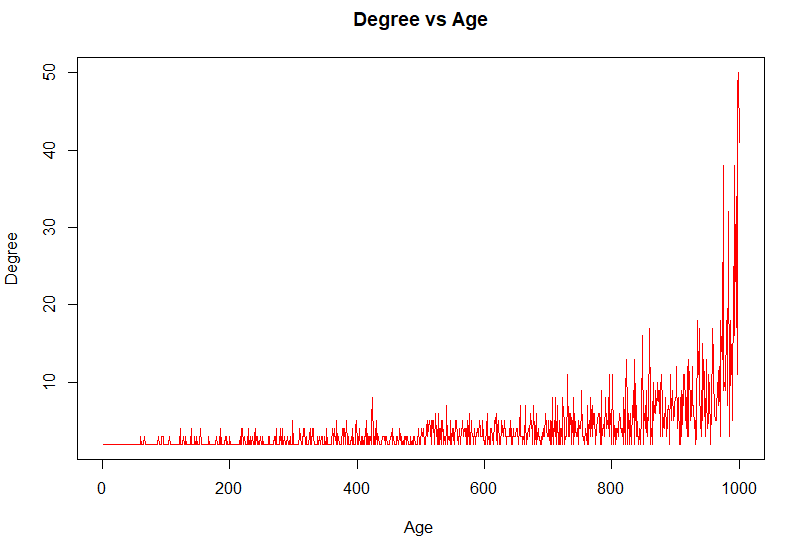
Using the techniques from previous parts, the slope for is -2.446509. The slope for is -2.969105.



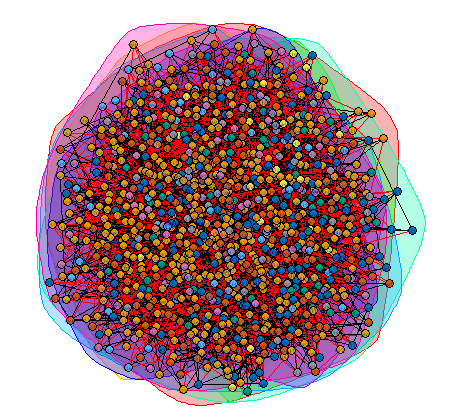
Using linear regression, the slope is -2.619996.

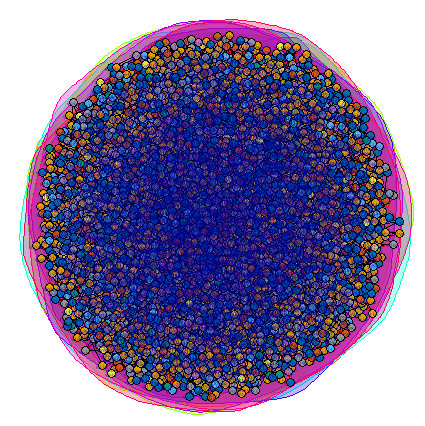


Using linear regression, the slope is -1.979664. Overall, the node degree distribution seems to sharper than the random picking distribution.

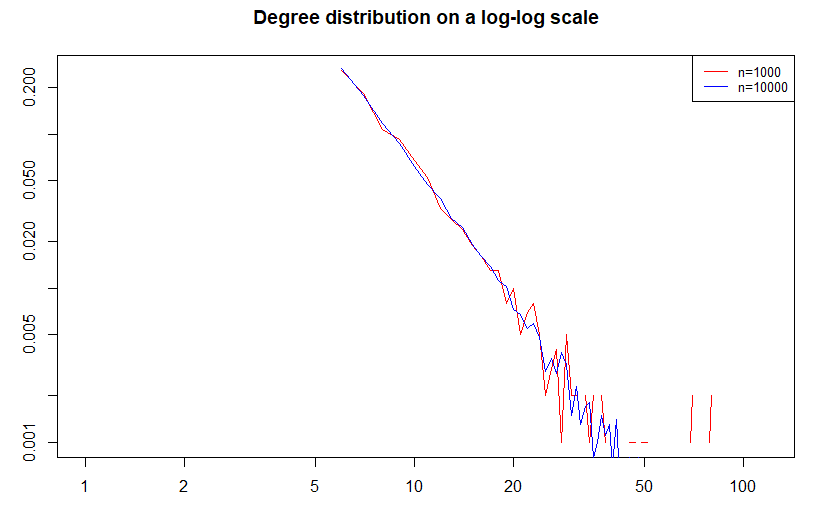


For , the network is always connected. We get the following plots

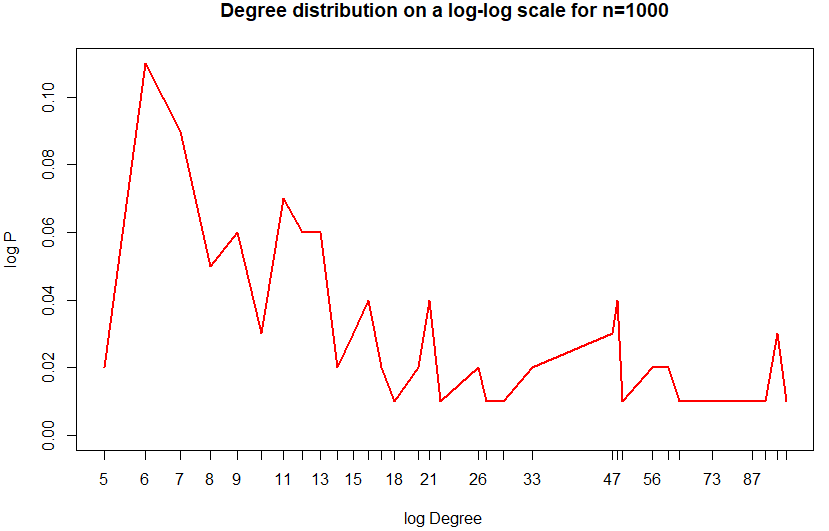
Modularity with 1000 nodes = 0.2760617

Modularity with 10000 nodes = 0.2793203.

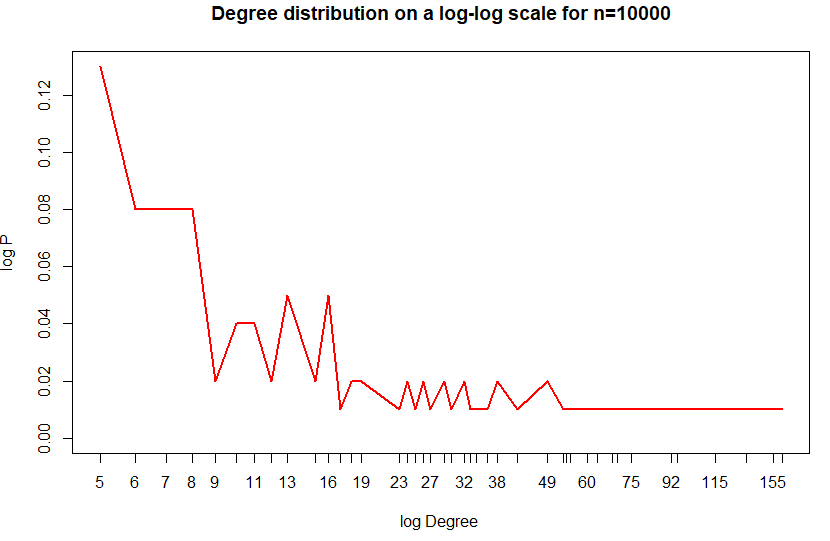
The modularity of both the networks is essentially the same.



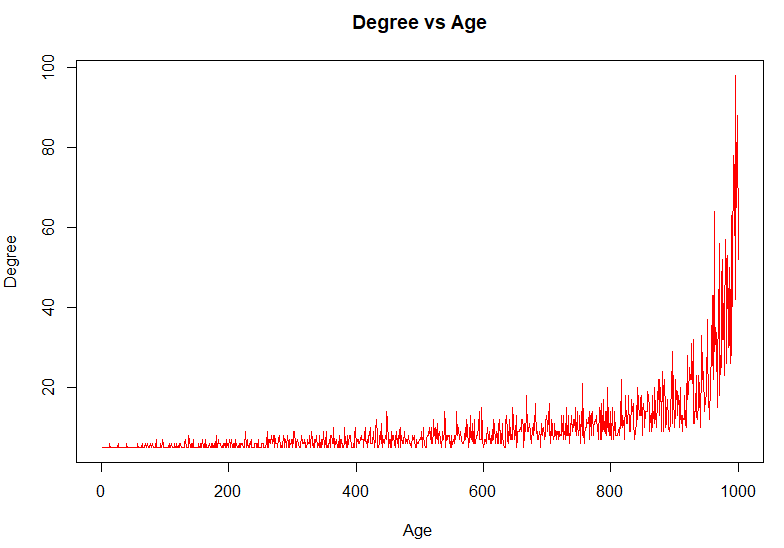
The slope for is -2.891173 and the slope for is -3.002394.



Using linear regression, the slope is -2.950684.



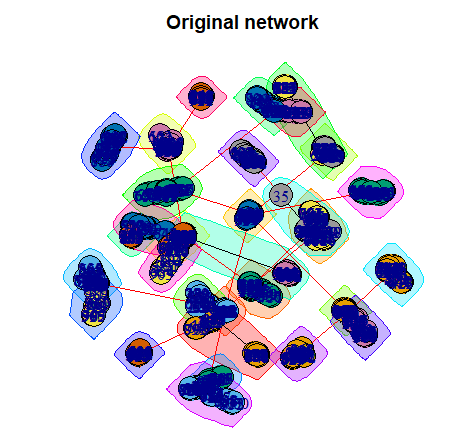
Using linear regression, the slope is -2.919648. Overall, the node degree distribution is roughly the same as the random picking distribution.



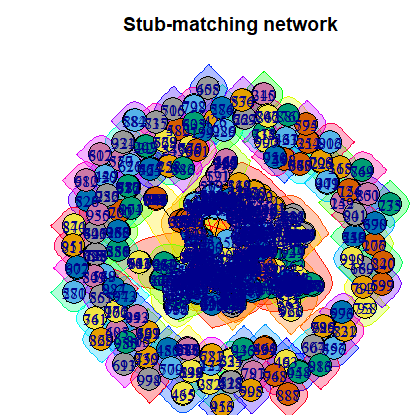
Comparison for different values of :

1. The modularity is the highest for . Modularity decreases as increases and this makes sense since as increases, more edges are created and hence the communities are not very well separated.
2. As increases, the degree decreases slower as the age of the node increases. This is evident from the sharpness of the curve. For , the degree of the nodes decreases very rapidly as the age increases. However, the decrease is much slower when .
3. The slopes of the log-log plots for all values of fall between 2 and 3. This makes sense since we know from lecture that random networks follow a power law with an exponent between 2 and 3.

**h)**



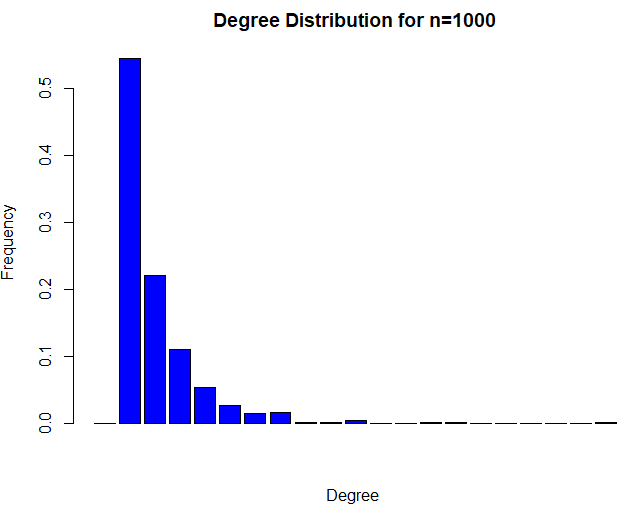
Modularity = 0.9338478



Modularity = 0.8525227

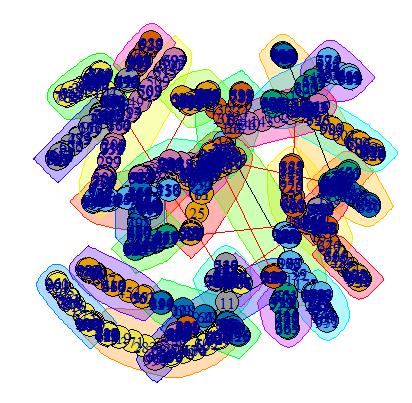
The original network has a higher modularity. This makes sense because in the original network, there are groups of densely interconnected nodes that are only sparsely connected with the rest of the network which means the communities are very well separated and hence the modularity is higher. On the other hand, all the nodes are generated first and then randomly connected in the stub-matching procedure. This usually results in networks with less dense communities and connected nodes. Hence the modularity is lower.

1. **a)**



The power law exponent = 3.673711. This was found using the fit\_power\_law() function.

**b)** Community structure

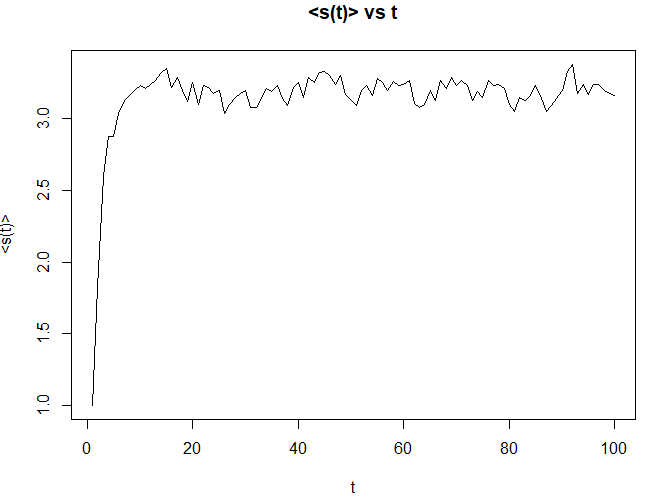


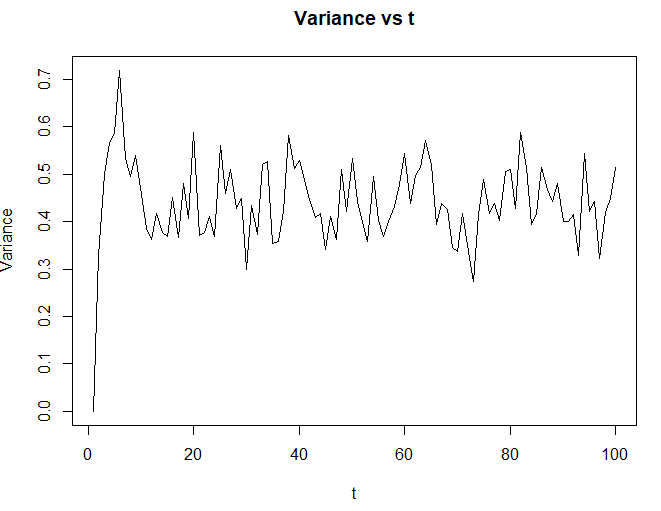
Modularity = 0.9350943

**Random Walks on Networks**

1. **a)** Created a random network with 1000 nodes. The diameter of the graph is 6.

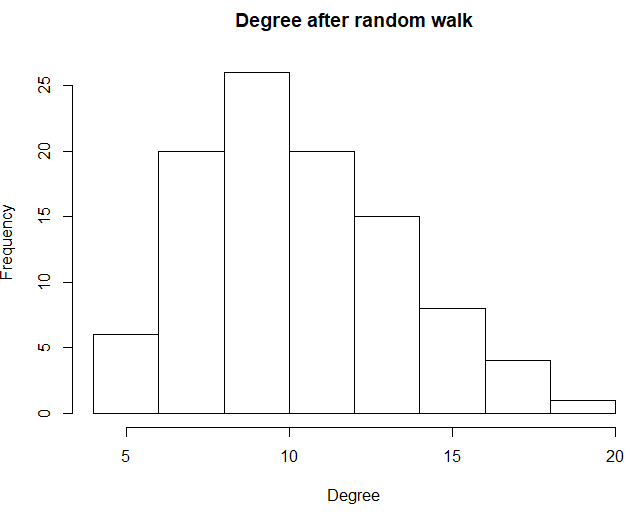
**b)**

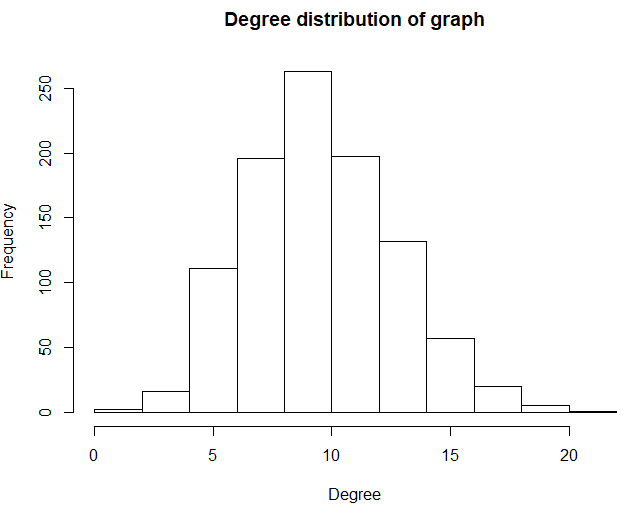




The random walker converges to around 3.2 in about 20 steps.

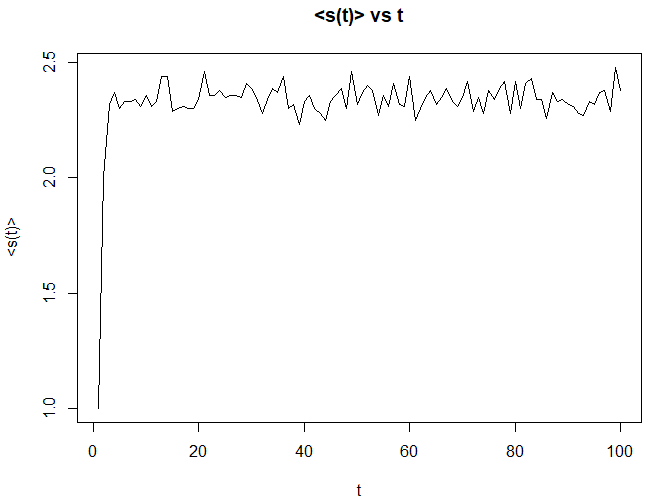
**c)**

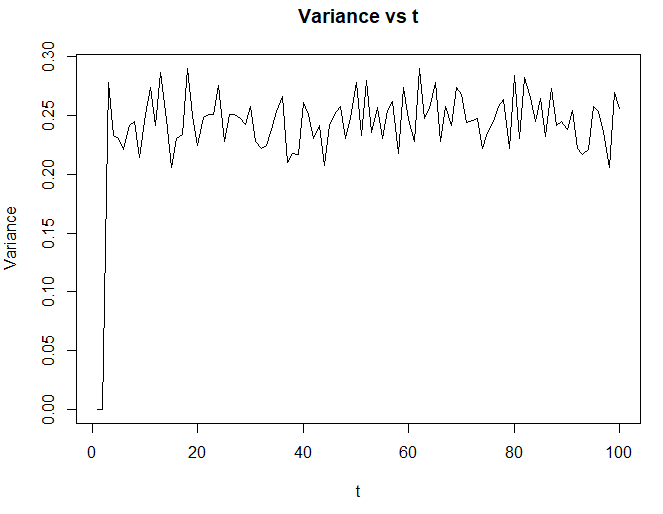




Both degree distributions are somewhat similar. The degree distribution of the graph resembles the normal distribution more closely than the degree distribution after the random walk. However, the mean of both distributions is around 10 which means both the distributions are similar to each other.

**d)** The diameter of the 1000-node random graph is 3.



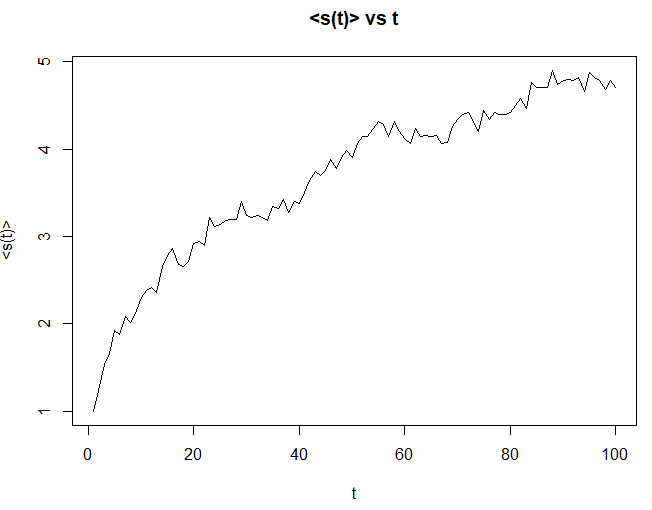


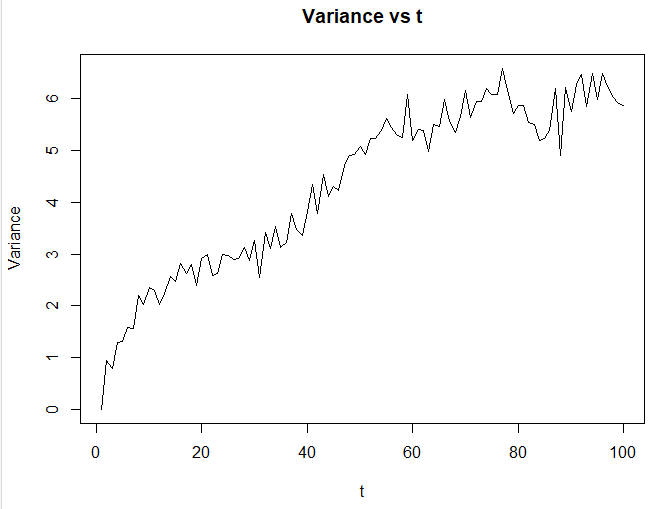
The random walker converges very quickly to about 2.4 in about 12 steps.

When , the average distance increases very rapidly and plateaus a lot faster than when . This is because the diameter is smaller and hence, the diameter plays an important role. When , the diameter is 3 which means the graph is very dense. Since the graph is very dense, the average distance converges very quickly, as seen in the plots above. Also, networks with smaller diameter have a lower variance in the distance traversed.

1. **a)** Created a preferential attachment network. The diameter is 21.

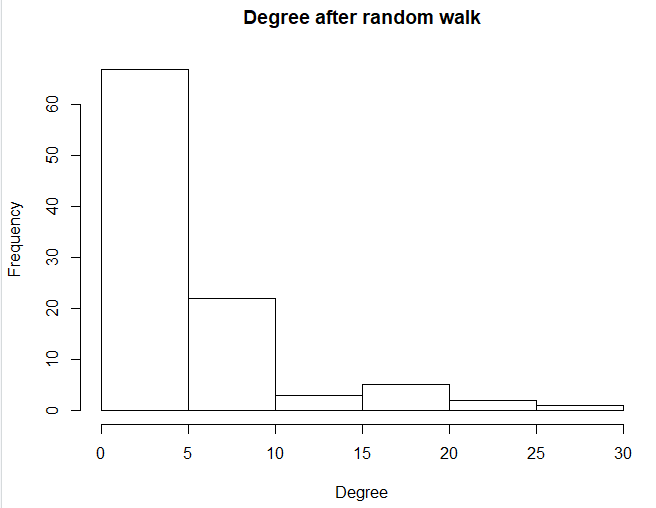
**b)**

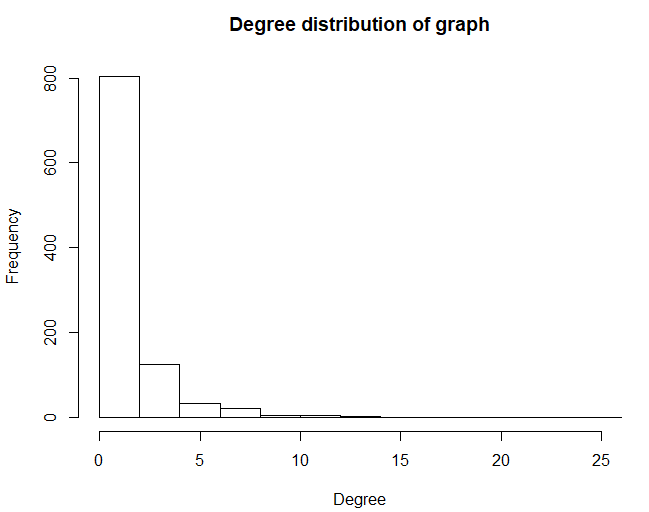




The random walker is moving quite slowly and failed to converge in the given amount of steps.

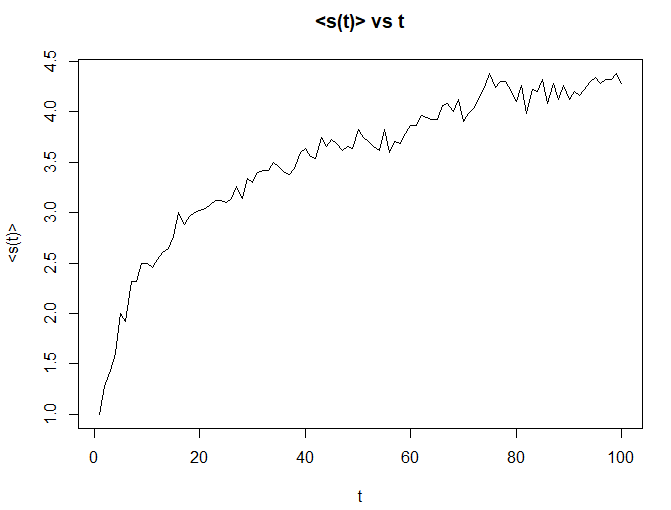
**c)**

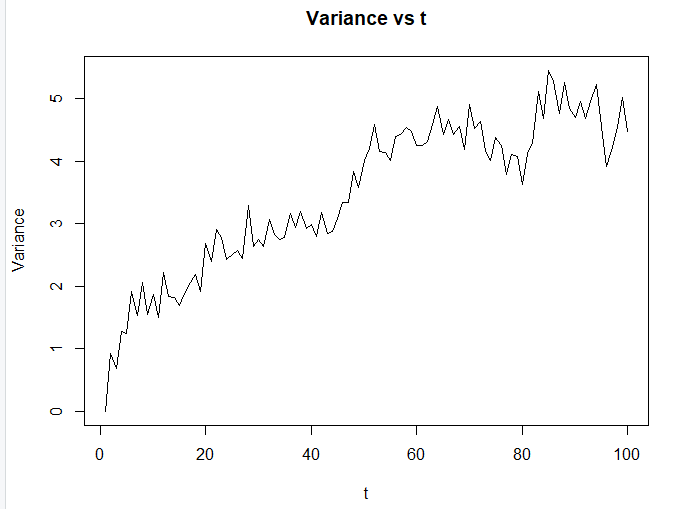




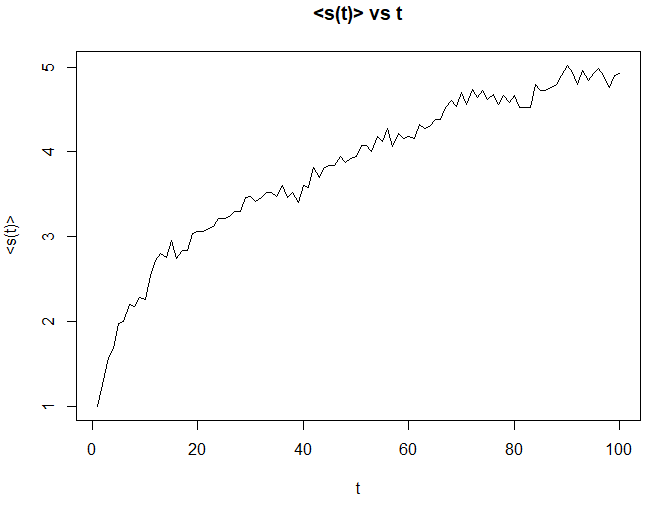
The degree distribution of the graph and the degree distribution after the random walk are again very similar. Both distributions follow the power law which makes sense since preferential attachment networks show fat-tailed degree distribution.

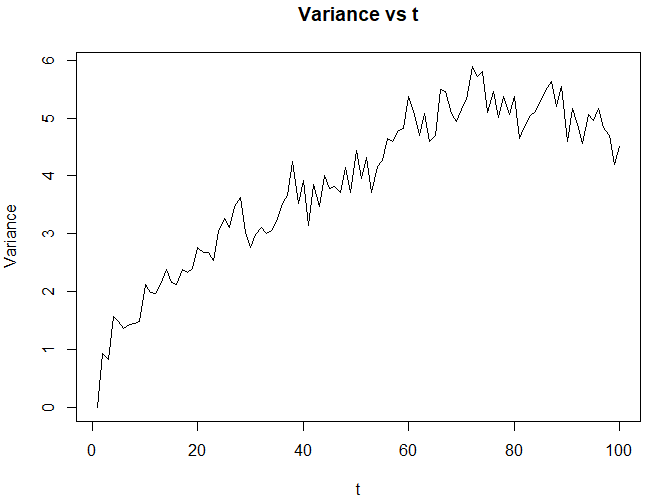
**d)** For , the diameter is 14.





For , the diameter is 29.





The average distance and variance for all three values of are very similar. Thus, diameter does not play a role in the convergence of average distance. However, an argument can be made for how long it would take for all the random walkers to converge. When , the diameter is very large and thus a random walker will take more steps to converge to a stable distance. On the contrary, a random walker should converge relatively quickly if since the diameter is smaller. This phenomenon is also depicted in the plots since the speed of the walker (given by the slope) is the largest for .

1. **a)**

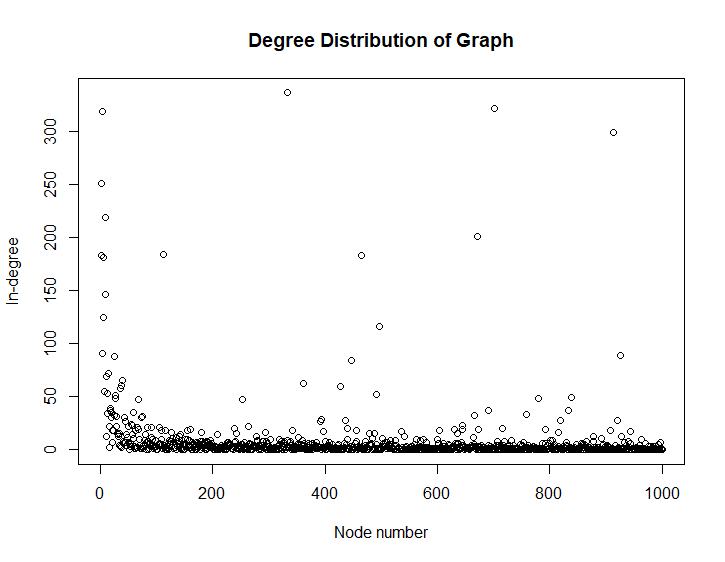
Now we take a closer look at a real-life example of a random walk, namely, one of Google’s search algorithm, PageRank. We begin by simulating a 1000 node network of webpages using the preferential attachment procedure. During the process of generating the network, each time step composes of bringing in a new node into the network, followed by m outgoing links to other existing nodes. In our simulation, we utilize m = 4. The probability of each outgoing link of the new node connecting to an existing node is proportional to the exiting node’s in-degree (or links connected to a webpage). One issue we face from using the preferential attachment network is that the first node contains no outgoing links, which causes a “black hole” that a random walker cannot escape from. To combat the phenomenon, we generated a second preferential attachment network of n = 1000, m = 4, permuted the nodes, and added its edge list to the first network. We then generate a transition probability matrix to use when performing random walks on the network. A random walk is as follows:

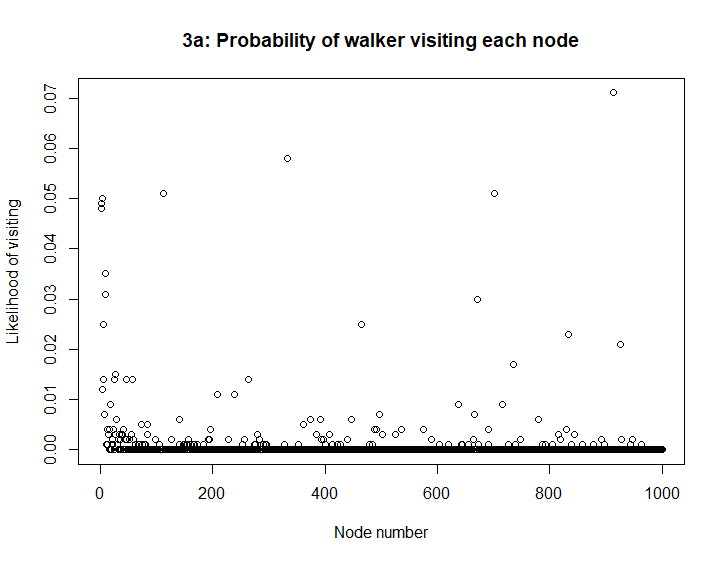
1. Start at a random node
2. Use the transition matrix to perform a random draw against the outgoing links of the node
3. Step to the outgoing node
4. Repeat steps 2-3 until the number of specified steps is reached

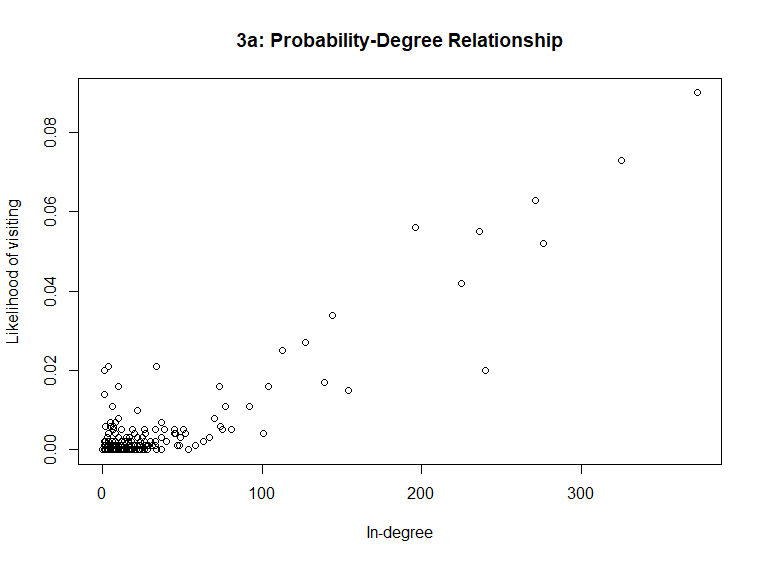
For confident accuracy in each PageRank, we chose to run 1000 walks with 1000 steps per walk.

Results of the probability of a walker visiting each node (their PageRank) are shown below. Given the fact that we added the edges of a secondary preferential attachment network to the primary network as a way of eradicating the black hole phenomenon, there appears to be a few outliers where some younger nodes have a very high likelihood of being visited. If the outliers are not taken into consideration, you see that the likelihood of visits versus the age of a node seems to follow a power law distribution. If the black hole phenomenon persisted with a sufficient step size per random walk, there will certainly most, if not all, walks ending in the first node (the probability of visiting a node is 1.0 for the first node and 0.0 for the other nodes)

Taking a look at the second graph depicting probability of visits versus its degree, there seems to be an increasing linear trend in the relationship (i.e. probability of visiting a node grows linearly in its in-degree).





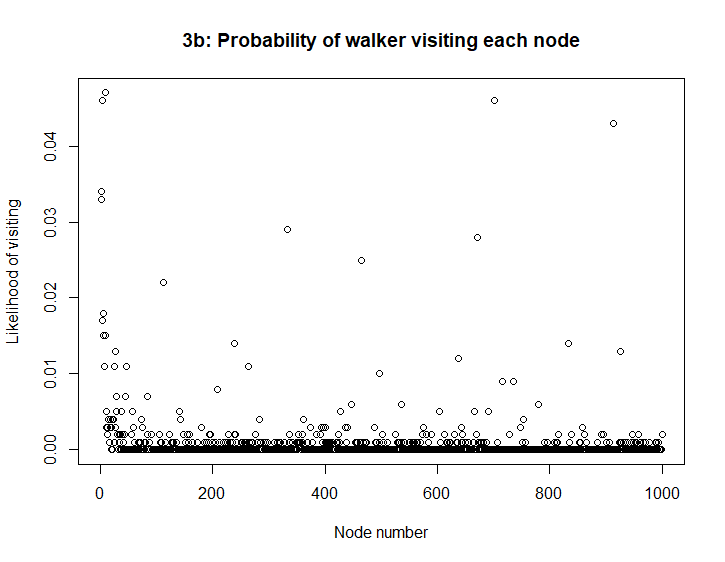


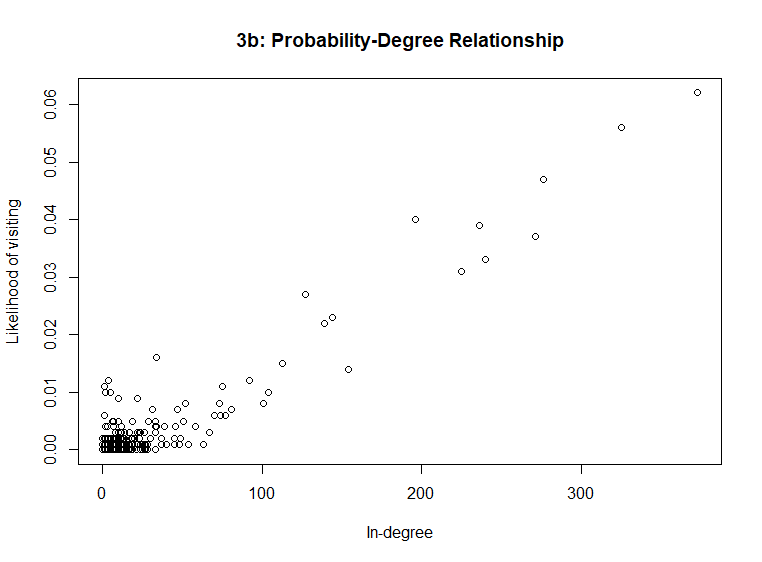
**b)** In addition to incorporating the previous procedure for random walks, we now explore a teleportation factor in an attempt to simulate a case of a random walker deciding to jump to a page that doesn’t necessarily correspond to an outgoing node of the current page. This occurs when the walker/surfer decides to go to any other page from their current page. The way we integrate this teleportation operation into the random walk procedure is by placing an additional probability step that governs whether the walker clicks on a link to another page or use the URL address bar to navigate to any other available page in the network. This transition probability is set to 0.15 for our run, indicating a 15% chance to jump to any other page and an 85% chance to click on an available link in the current webpage. If the walker decides to teleport, the probability of traversing to any other page (or remaining in the current page) is uniformly distributed across every page. In other words, each webpage is equally likely to be teleported to (probability of traversing to each webpage is 1 / n). A random walk is as follows:

1. Start at a random node
2. Take a random draw
   1. 15% chance to use uniform teleportation
   2. 85% chance to use the transition probabilities of the current node to traverse to one of its outgoing links
3. Step to the outgoing node
4. Repeat steps 2-3 until the number of specified steps is reached

Again, the probability of visiting each node is calculated by performing 1000 random walks with 1000 steps per walk.

Elaborating the results shown below, the probability of visiting each node doesn’t seem to deviate too much from part 3.a, as the weight given on the teleportation operation is small compared to using the transition matrix. If we, however, give more weight/importance to uniform teleportation, the likelihood of visiting each node will gradually tend towards a uniform probability distribution. The probability-degree relationship follows a similar trend; hence, one can deduce the probability is still linearly related to the degree of the node.





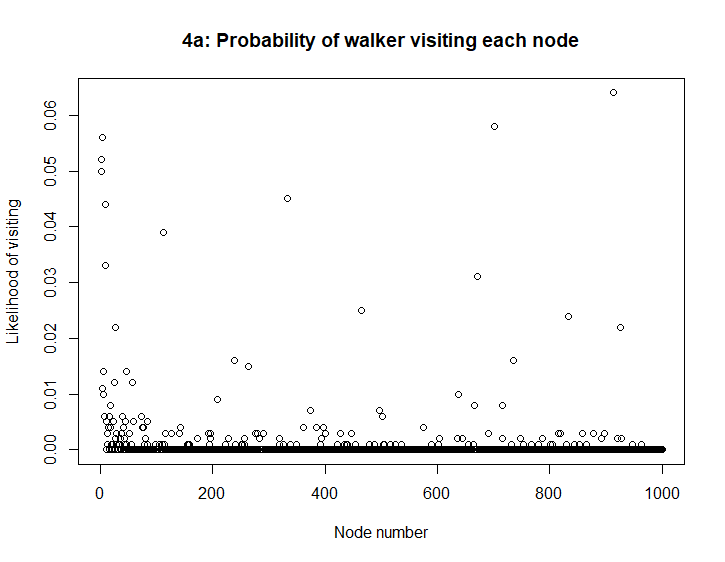
**4.**

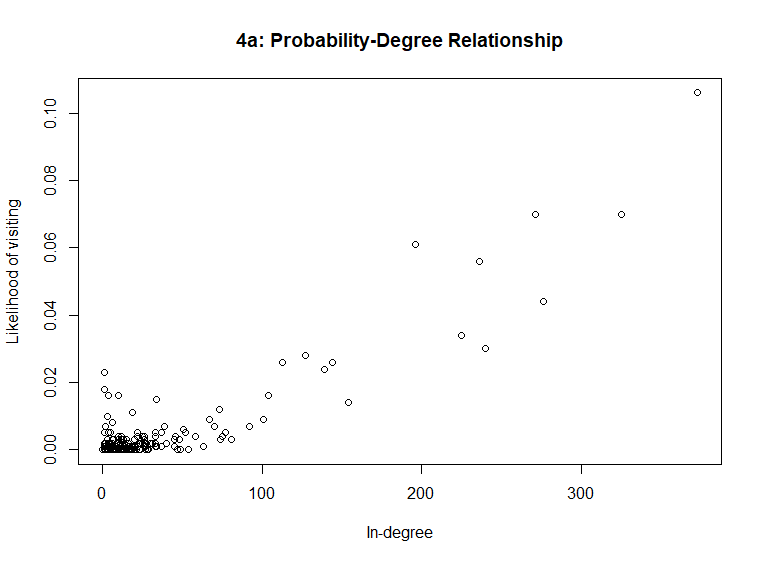
**a)**

In this example, we demonstrate a different behavior of a random walker surfing the web by establishing a notion of importance. Specifically, under teleportation, the probability of visiting each node is more likely for those with a higher number of incoming links. We quantify this probability trend through the usage of each of the node’s PageRank generated from part 3.a in replacement of uniform probability across all pages. This “personalized” PageRank teleportation scheme, along with the original transition matrix traversal procedure is as follows:

1. Start at a random node
2. Take a random draw
   1. 15% chance to use “personalized” PageRank teleportation
   2. 85% chance to use the transition probabilities of the current node to traverse to one of its outgoing links
3. Step to the outgoing node
4. Repeat steps 2-3 until the number of specified steps is reached

It is shown from the results below that the probability of a walker visiting each node is higher for the earlier nodes when using the “personalized” PageRank teleportation (4.a) versus random walking with no teleportation. This happens to be the case because in 4.a, we are taking advantage of the fact that we can teleport to the webpages with higher incoming links even if the current page is not directly connected to it. In contrast, for 3.a, it would take more steps to reach the nodes with higher in-degrees, indicating a lower-probability of visiting these nodes. This teleportation occurs on average 15% of the time over the steps of a random walk. If we put more weight into this teleportation scheme, the probability of visiting nodes with lower in-degrees decreases, while ones with higher in-degrees increases.





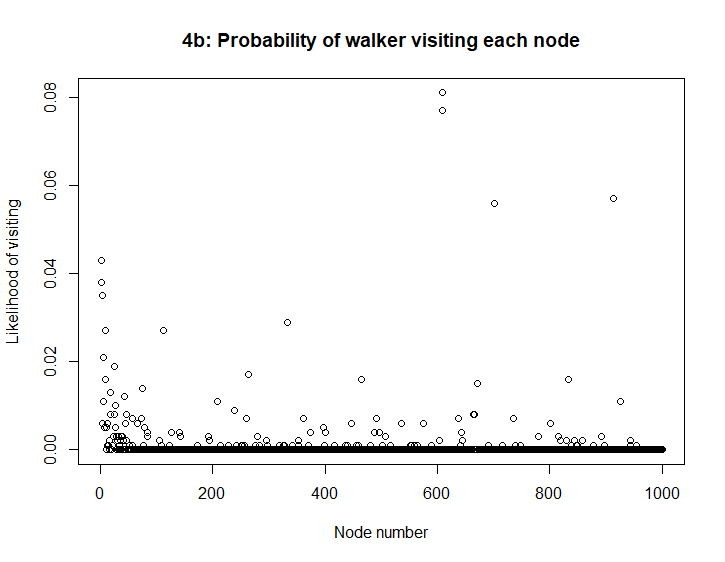
**b)**

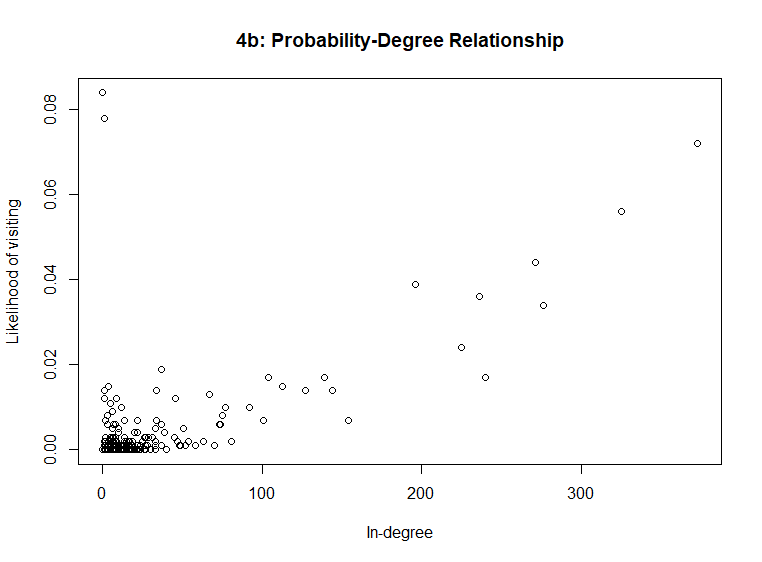
The next teleportation scheme we consider when simulating random walks comes from the fact that surfers are more likely to visit sites that are well-trusted. For example, if a surfer has knowledge of malware being contained in a particular site, he/she will be hesitant in visiting that page. We naively account for this case using a median PageRank teleportation operation. Again, with a 15% chance per step, a surfer will jump to another site per the specified scheme. We first assume that the two median PageRank pages are well-trusted, and hence traversable via teleportation (each with probability 0.5). The rest of the nodes in the network have a 0% chance to be teleported to. A random walk procedure with median PageRank teleportation is as follows:

1. Start at a random node
2. Take a random draw
   1. 15% chance to use median PageRank teleportation
   2. 85% chance to use the transition probabilities of the current node to traverse to one of its outgoing links
3. Step to the outgoing node
4. Repeat steps 2-3 until the number of specified steps is reached

Considering this schema, we plot the results from 1000 random walks with 1000 steps per random walk.

The PageRank values of using median teleportation is noticeably different than traversal with no teleportation. To compare, the probability of visiting each node with median teleportation puts more emphasis on the two trusted webpages than any other webpage. In other words, the 2 median pages have a highest PageRank (even higher than the ones with the highest in-degrees). These 2 nodes, which are approximately around the 600th node, are noticeable outliers compared to the others (at about 0.075 to 0.08 probability). However, the median nodes are virtually non-existent in terms of probability of visit in 3.a.





**c)**

To this point, we have explored the a few approaches in simulating PageRank through a directed preferential attachment network of 1000 web pages, taking into account of eliminating the “black hole” phenomenon. The page rank probabilities of each node were first measured via random walks with no teleportation of the preferential attachment network. This involves calculating a transition matrix of the network, where a random walker at a particular node is equally likely to visit any of its outgoing links. We then included a 15% chance of teleporting under each step of a random walk (85% chance of simply walking to a new node using the transition matrix).

The various teleportation operations are listed below:

* Uniform teleportation (each page is equally likely to be visited under teleportation)
* Personalized PageRank teleportation (random walker’s interest in a page is proportional to the page’s PageRank under teleportation)
* Median PageRank teleportation (random walker’s is only interested in visiting either of the two pages with median PageRank under teleportation)

There are a few advantages to each of these teleportation operations; however, they do also contain flaws. For example, uniform teleportation does not account for a popularity factor of webpages (assumes walker will equally likely jump to any page that’s available in the network). Personalized PageRank teleportation assumes a random walker will frequently visit webpages that contain many direct links to it while seldom visiting ones with fewer in-links. Although this may seem desirable, it does not account for trust. Another flaw to this teleportation scheme is that there may be well-known and well-established webpages that may have fewer incoming links than those that aren’t as popular, but the teleportation still favors those with higher incoming links. Lastly, median PageRank teleportation assumes the user to solely visit a small subset of trusted webpages with equal probability (in our example there are 2 trusted webpages each carrying a visiting probability of 0.5). Unfortunately, median PageRank neglects the fact that a random walker may also visit other webpage, even with a small probability.

Hence, a “hybrid” teleportation operation is needed to best simulate a real world walk in a massive network of interconnected webpages. We consider a damping factor, d, that measures the importance of median PageRank teleportation as opposed to uniform teleportation. In our test, we utilize a damping factor of 0.85 in favor of median PageRank if a teleportation operation is used within a step:

Each random walk utilizes this teleportation operation with a probability of 15% and the column of the adjacency matrix otherwise. With sufficient number of random walks and steps per random walk, the PageRank result is a closer representation of a real random walker than the previous teleportation operations. Shown below are the results of the newly modified PageRank equation.

