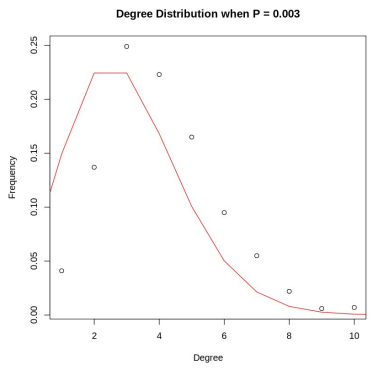
Project 1:Random Graphs and Random Walks

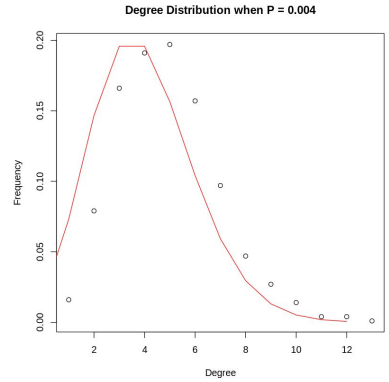
Chaohao,Li 705430930
Yifei,Chen 705444102
Yifei,Tang 605431831

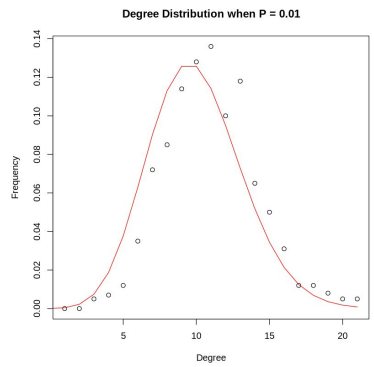
Part 1:

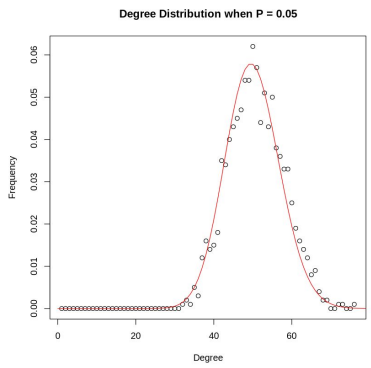
1(a):

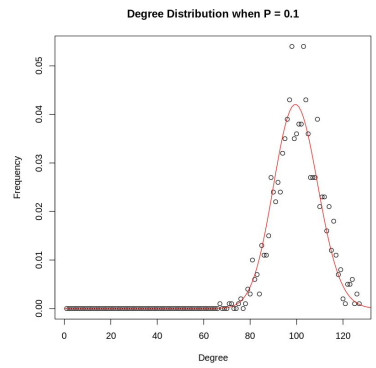
Here we plotted the degree distribution of undirected random networks with 1000 nodes under probability p = 0.003, 0.004, 0.01, 0.05, 0.1 separately and below are the results:











From the definition, we can figure out that the degree of a random node has the relation N ~ p^(N)*(1-p)^(n-1-N), which is related to binomial distribution. And we indeed found that the distributions are close to the binomial distribution in the graph we got.
Here are the mean and variance we got compared with the theoretical values:

p	mean	mean(theory)	var	var(theory)
0.003	2.952	2.997	2.962659	2.988
0.004	4.042	3.006	4.112348	3.980
0.01	9.922	9.99	9.739656	9.890
0.05	50.126	49.95	46.61073	47.453
0.1	99.864	99.9	89.41892	89.91

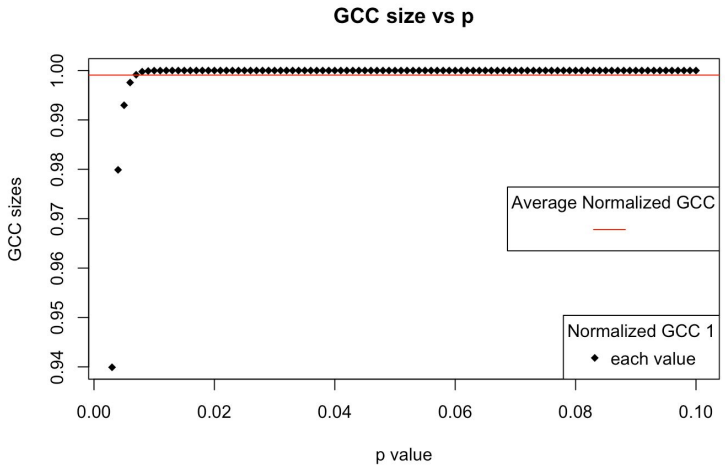
1(b):

For each probability p, we tried 100 instances and below are the results.

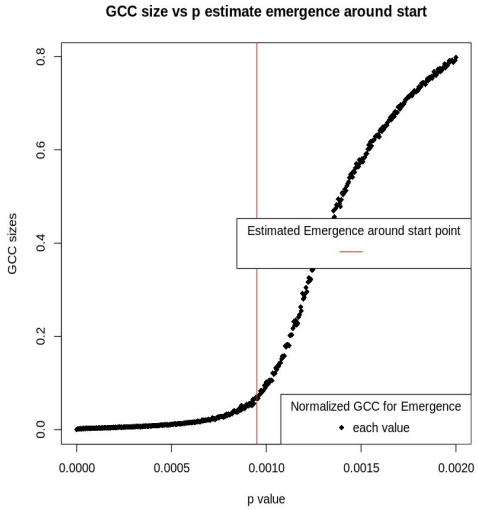
p	Prob. of connected (100 instance)	Connected (1 instance)	Dia of GCC (1 instance)
0.003	0	No	14
0.004	0	No	11
0.01	0.98	Yes	5
0.05	1	Yes	3
0.1	1	Yes	3

1(c):

For n =1000, we got ln(1000)/1000 = 0.0069 and 1/n = 0.001, then we roughly set pmax to 0.01 here and plot the GCC size/number of nodes (fraction) along with p for 100 times and calculated the average.
We observed that there is a threshold around 0.007, GCC size starts to be linear with the number of nodes (fraction is close to 1).

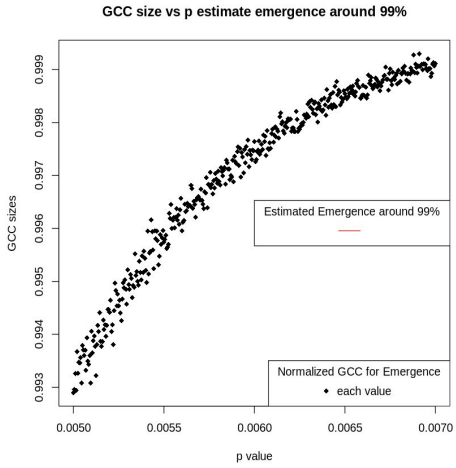


(i) to find the emerge point, we zoomed the graph around 0.001:



Here we find the emerge point (fraction starts to increase obviously) is around 0.001 (1/n).

(ii) to find the point that the fraction is over 00%, we zoomed the graph around 0.007:

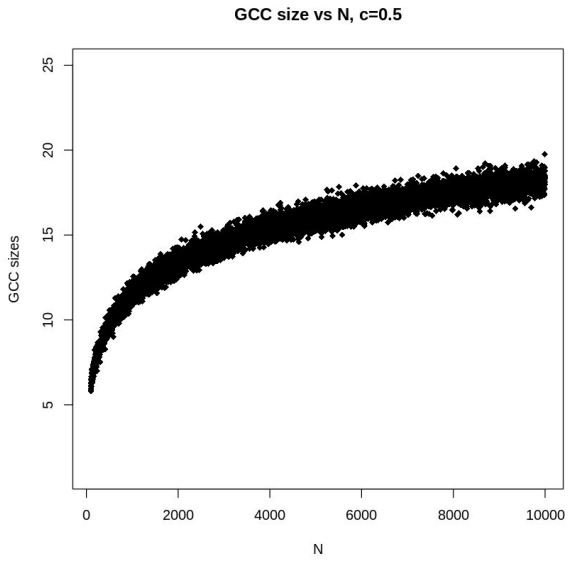


Here we find the point the fraction 99% is around 0.0069 (ln/n).

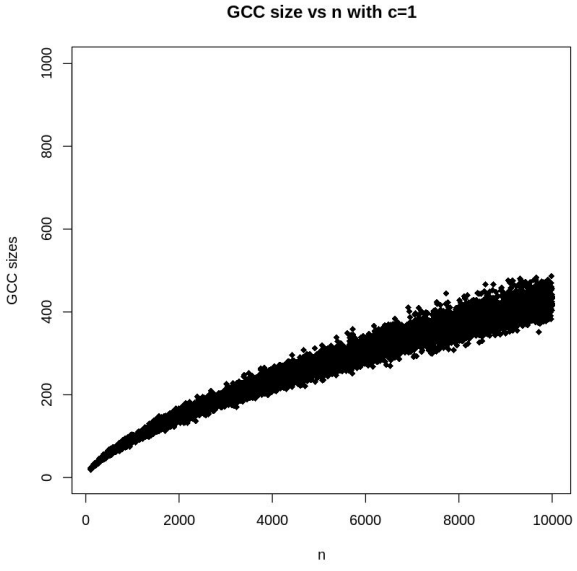
1(d):

(i)

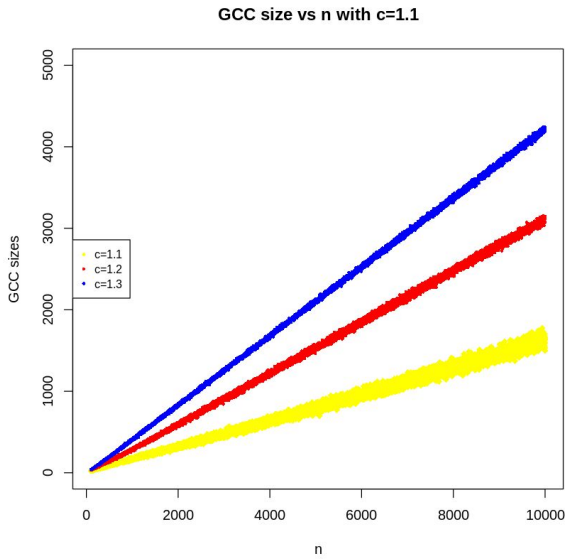
We observed that as the average degree of nodes becomes larger, the expected GCC size tends to be larger for the same node number as well. When c is equal to 0.5, the number of nodes and the GCC size seem to have a logarithmic relationship.



(ii)



(iii)



(iv)

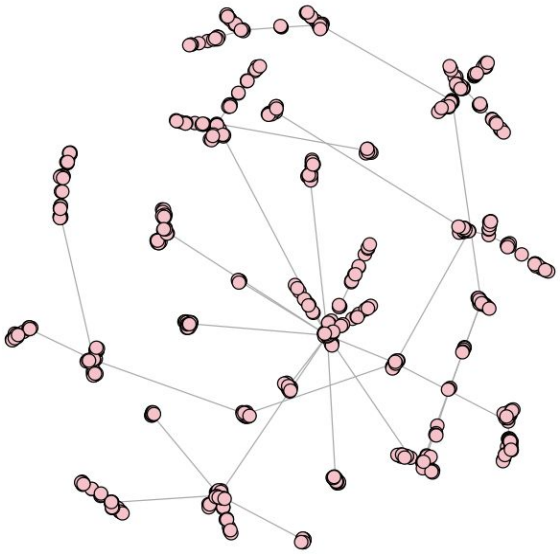
For c = 0.5, the expected GCC size and n seem to have a logarithmic relation.

For c = 1, the relation is nearly linear.

For c = 1.1,1.2,1.3, the relations are all linear between the expected GCC size and n.

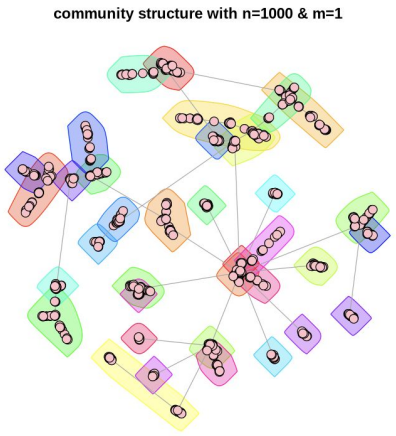
2(a):

Here we tried to construct a preferential attachment network (1000 nodes, m =1) for 100 times. We find all of them are connected and below is the visualization of one instance:



2(b):

```
0.931014598181767
IGRAPH clustering fast greedy, groups: 36, mod: 0.93
+ groups:
  $`1`
   [1]   6  16  55  97 107 109 112 114 129 137 146 199 247 271 276 282 318 351
  [19] 370 381 385 415 462 481 489 504 543 576 590 592 640 690 725 732 747 752
  [37] 771 785 807 828 866 885 909 920 941 947 974

  $`2`
   [1]   45   50   52   89  100  149  159  236  267  286  300  331  336  337
  [15]  362  390  418  466  482  528  530  550  556  557  661  663  671  672
  [29]  675  681  697  710  731  758  776  792  810  822  834  850  882  922
  [43]  967 1000
  + ... omitted several groups/vertices
```

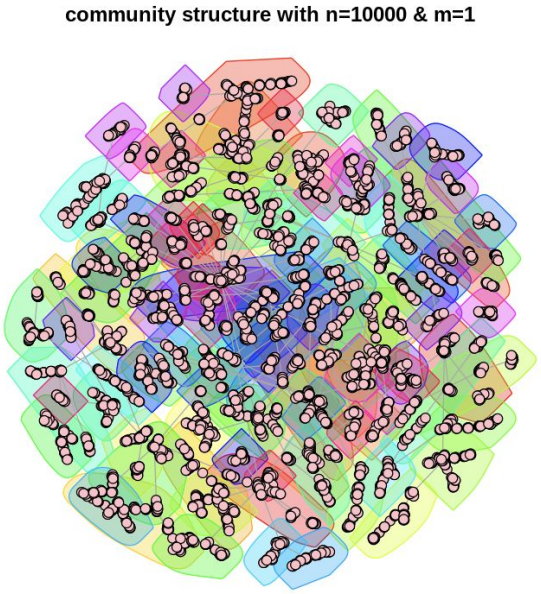


Here the number of communities is 36 and the modularity is around 0.931.

2(c):

With the same procedure with previous part, we produced a network with 10000 nodes:

```
0.977589733170751
IGRAPH clustering fast greedy, groups: 111, mod: 0.98
+ groups:
  $`1`
    [1]    5   44  114  116  169  283  310  427  469  473  503  543  544  594
   [15]  708  832  963 1000 1077 1086 1092 1094 1195 1269 1339 1385 1398 1432
   [29] 1536 1547 1548 1557 1593 1628 1693 1768 1878 1972 2013 2072 2130 2132
   [43] 2223 2312 2364 2519 2555 2595 2599 2724 2746 2832 2836 2882 2896 2914
   [57] 2993 3017 3060 3080 3164 3185 3206 3270 3318 3337 3400 3411 3477 3579
   [71] 3655 3717 3798 3848 3910 4023 4066 4103 4132 4140 4181 4202 4253 4301
   [85] 4313 4378 4429 4438 4446 4472 4543 4563 4648 4670 4701 4707 4712 4808
   [99] 4857 4933 4956 4971 5011 5017 5080 5100 5103 5109 5142 5188 5226 5302
  [113] 5430 5435 5462 5474 5492 5505 5530 5544 5578 5603 5646 5652 5665 5718
  + ... omitted several groups/vertices
```



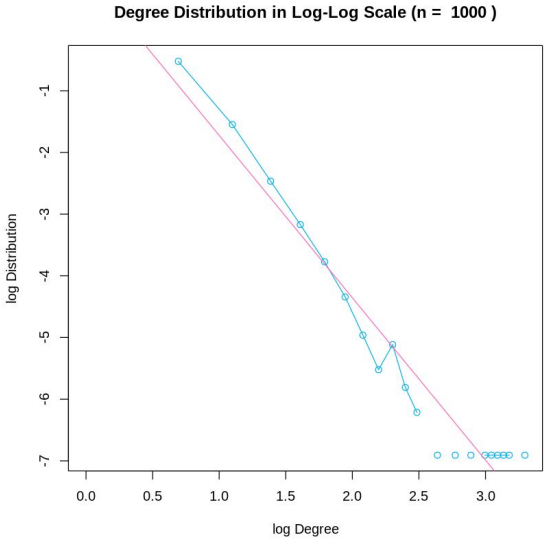
Here we got 111 communities and the modularity is around 0.98.

With a higher number of nodes, we found that the modularity increased, which means that the relations between the members of every community were intensified and the relations among the communities were weakened.

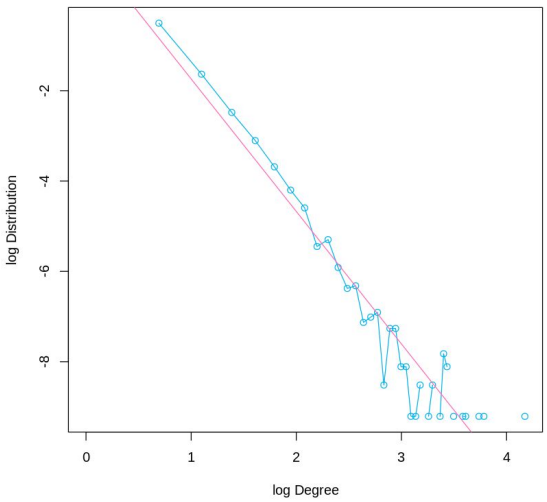
2(d) :

The estimated slope of the linear regression is -2.6318, when n = 1000.
The estimated slope of the linear regression is -2.9338, when n = 10000.

```
Call:
lm(formula = log_distribution ~ degree)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87035 -0.39952  0.05211  0.33779  0.85815

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.9080     0.3562   2.549   0.0201 *
degree       -2.6318     0.1447 -18.185 4.95e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4708 on 18 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.9484,    Adjusted R-squared:  0.9455
F-statistic: 330.7 on 1 and 18 DF,  p-value: 4.946e-13
```



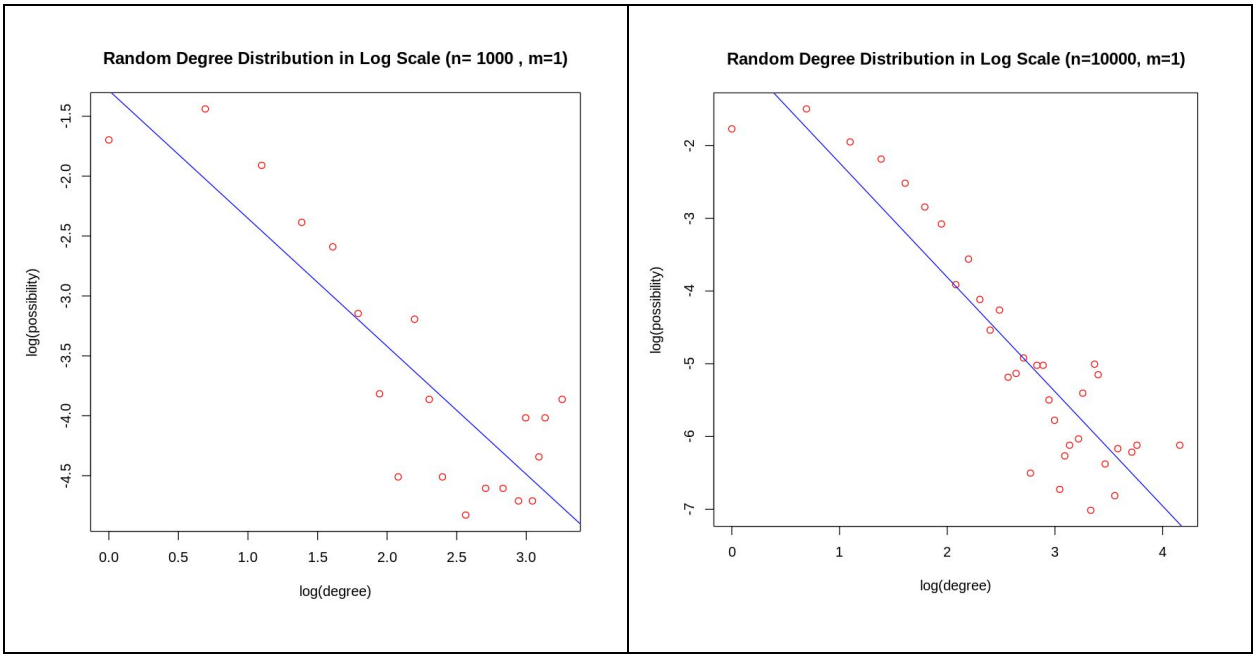

```
Call:
lm(formula = log_distribution ~ degree)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3995 -0.3509  0.0265  0.3666  1.8421

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.1944     0.4121   2.898  0.00673 **
degree       -2.9338     0.1434 -20.462  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6638 on 32 degrees of freedom
  (31 observations deleted due to missingness)
Multiple R-squared:  0.929,     Adjusted R-squared:  0.9268
F-statistic: 418.7 on 1 and 32 DF,  p-value: < 2.2e-16
```



2(e) :

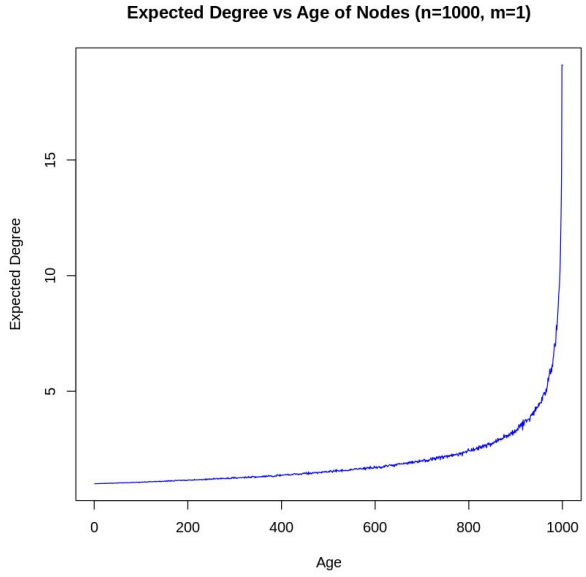
Yes. The distributions are linear in the log-log scale for both n = 1000 and n = 10000.
The slope for n = 1000 is -1.0676.
The slope for n = 10000 is -1.5724.
These linear distributions are lower than those from the node degree distribution. For the randomly picking model, the Pk will be higher, which yields in lower value of gamma.

```
Call:
lm(formula = distri_map ~ deg_map)

Residuals:
     Min       1Q   Median       3Q      Max
-1.00444 -0.41679 -0.03392  0.44430  0.90051

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.2854     0.3365  -3.820  0.00125 **
deg_map      -1.0676     0.1420  -7.517 5.88e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5521 on 18 degrees of freedom
Multiple R-squared:  0.7584,    Adjusted R-squared:  0.745
F-statistic:  56.5 on 1 and 18 DF,  p-value: 5.882e-07

Call:
lm(formula = distri_map ~ deg_map)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4769 -0.3794  0.1173  0.4559  1.0858

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.6659     0.3491  -1.907   0.0655 .
deg_map      -1.5724     0.1242 -12.657 5.32e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.658 on 32 degrees of freedom
Multiple R-squared:  0.8335,    Adjusted R-squared:  0.8283
F-statistic: 160.2 on 1 and 32 DF,  p-value: 5.317e-14
```

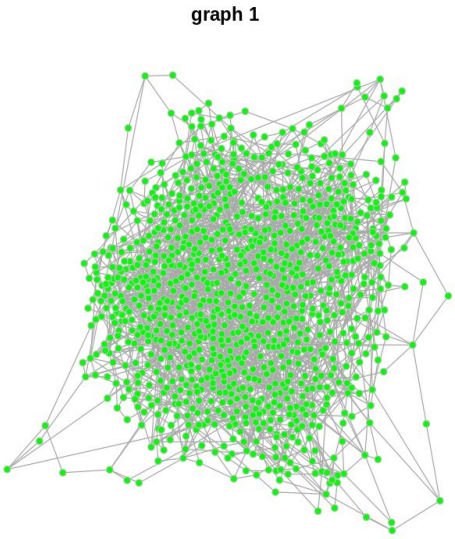


2(f):

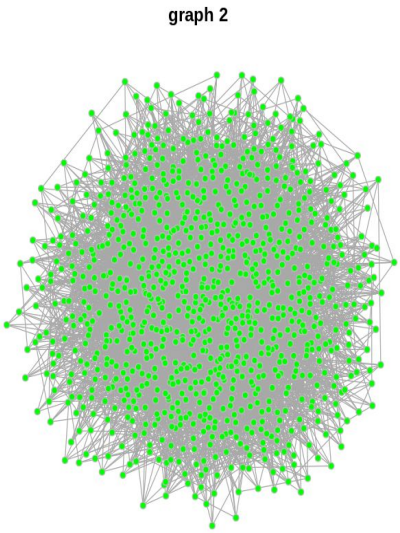
The relationship between the age of nodes and the expected degree is shown below. The estimated expected degree was calculated by computing the mean of all degrees, and was equal to 1998.



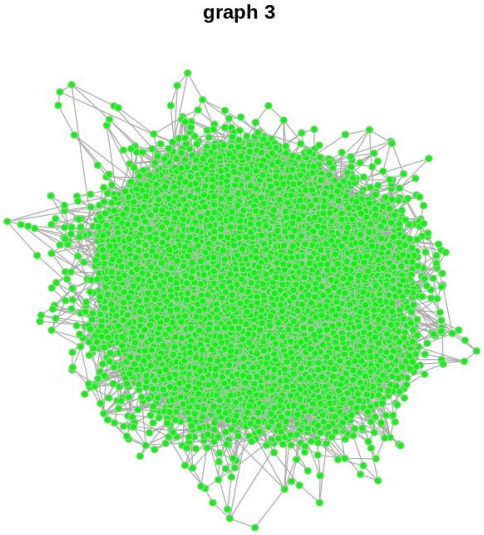
2(g):



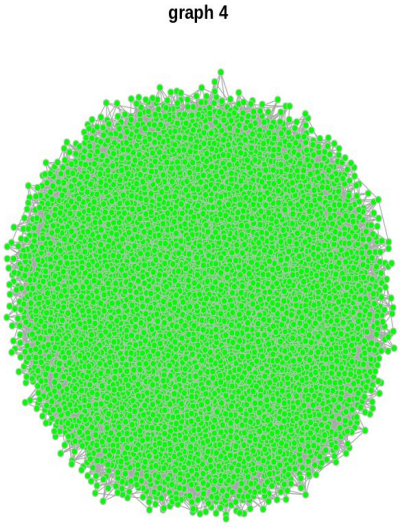
n= 1000, m = 2



n= 1000, m= 5

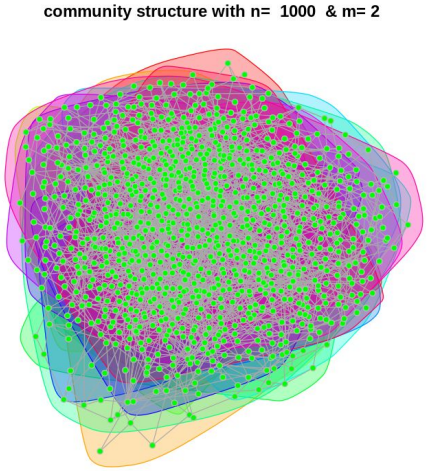


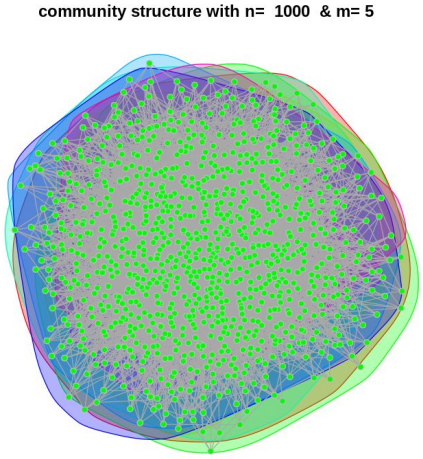
n= 10000, m= 2

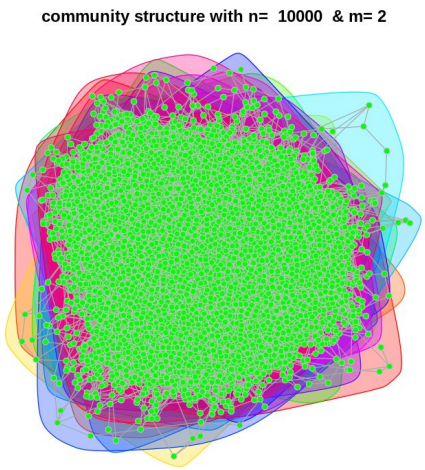


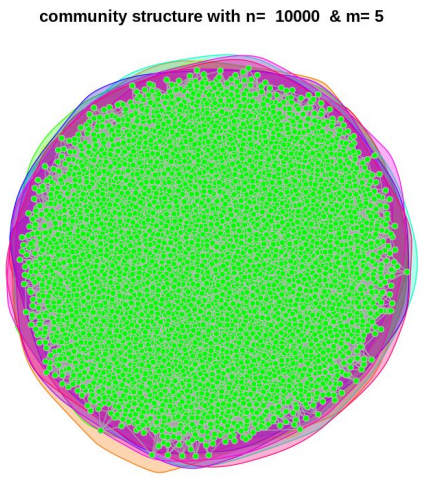
n= 10000, m= 5

These are the four graphs generated using preferential attachment models, with different numbers of nodes and different values for m. Then we use the fast greedy method to measure modularity for these graphs.



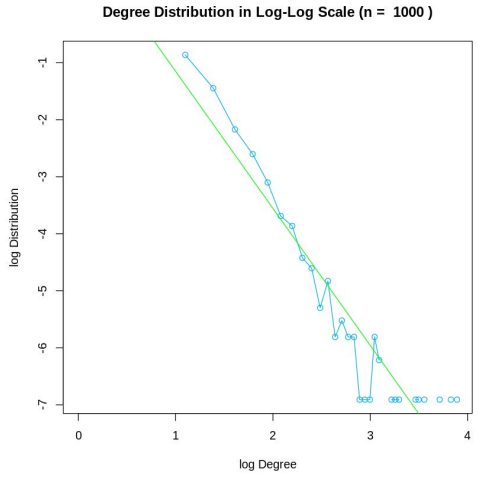
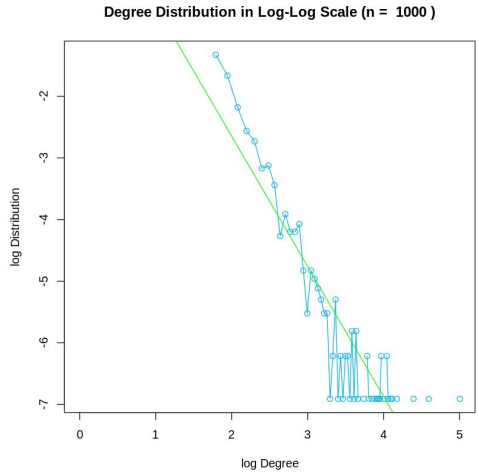
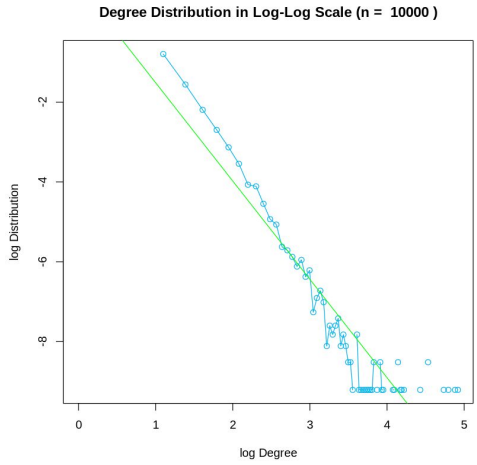
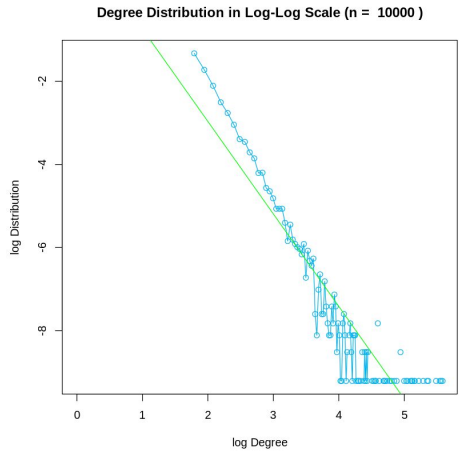






The modularity for the graph when n = 1000 and m = 2 is 0.525.
The modularity for the graph when n = 1000 and m = 5 is 0.279.
The modularity for the graph when n = 10000 and m = 2 is 0.531.
The modularity for the graph when n = 10000 and m = 5 is 0.275.

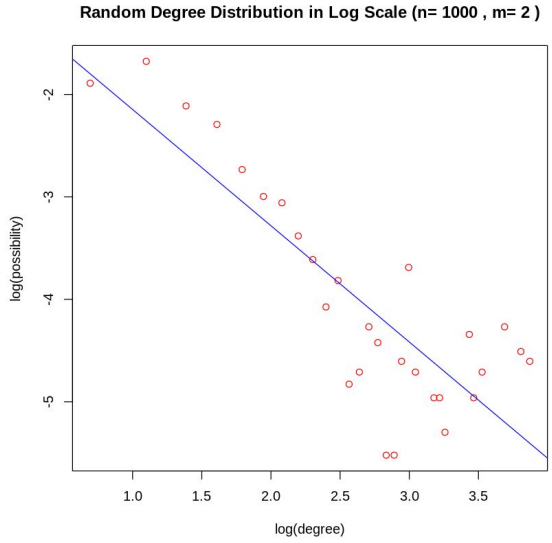
If we look at the results here, when the number of nodes is fixed, it seems that the modularity of a graph is lower, when m is larger. So it potentially shows that lower modularity means dense connections in different modules.

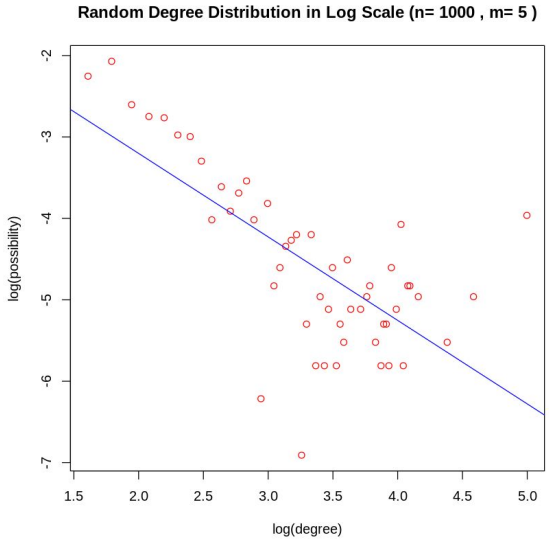
	m = 2	m = 5
n = 1000		
n = 10000		

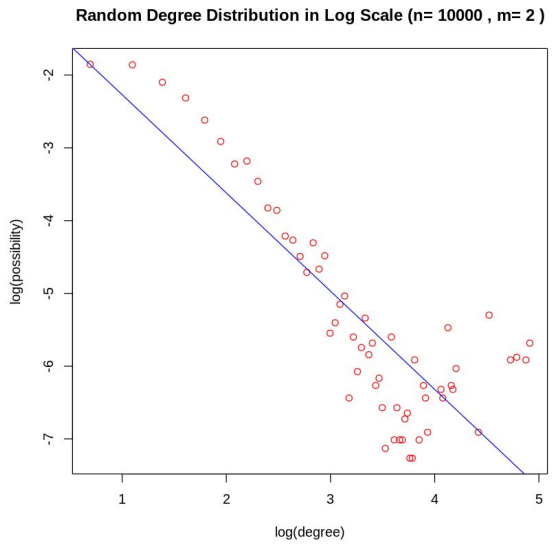
The slope for the degree distribution when n = 1000 and m = 2 is -2.4.
The slope for the degree distribution when n = 1000 and m = 5 is -2.1.
The slope for the degree distribution when n = 10000 and m = 2 is -2.46.
The slope for the degree distribution when n = 10000 and m = 5 is -2.2.

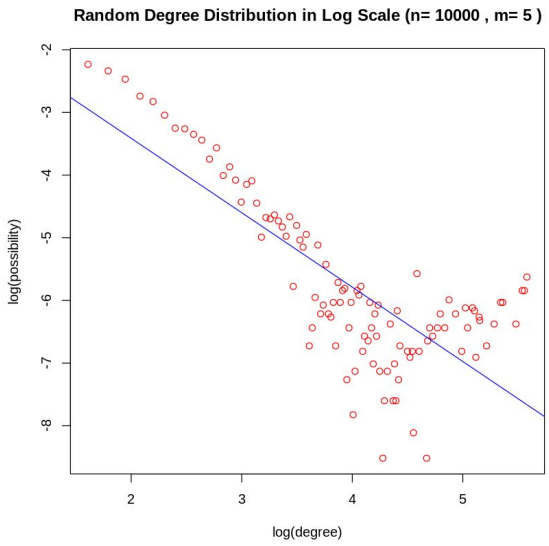
As for degree distribution, with the same amount of node, a distribution that has higher m value, shows a less inclined behavior in its linear regression slope.

Then for each m values and numbers of nodes, we computed the randomly picked degree distributions, which are shown as follows.





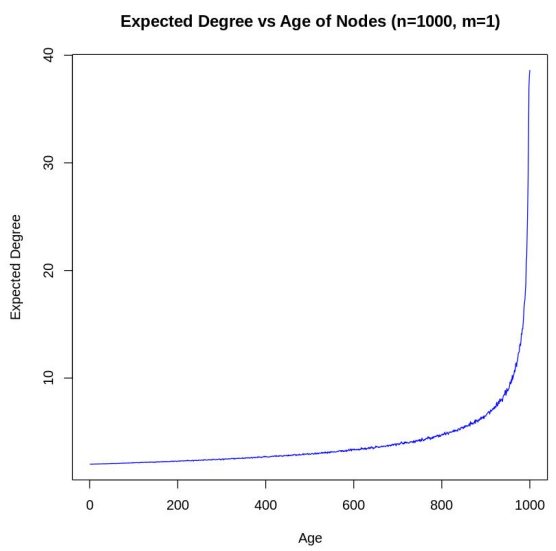
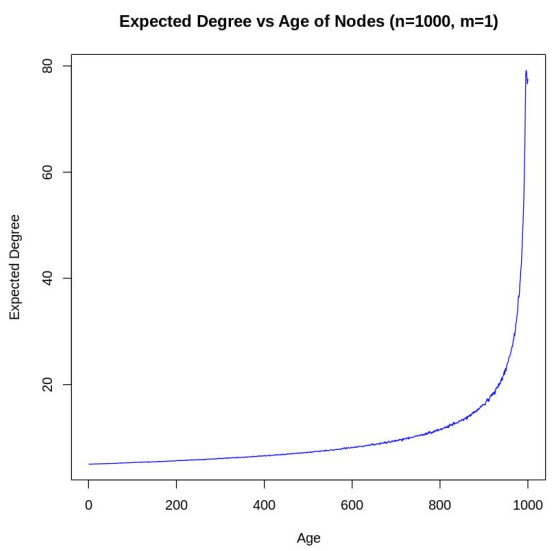




The slope for randomly picked degree distribution when n = 1000, m = 2 is -1.13.
The slope for randomly picked degree distribution when n = 1000, m = 5 is -1.03.
The slope for randomly picked degree distribution when n = 10000, m = 2 is -1.35.
The slope for randomly picked degree distribution when n = 10000, m = 5 is -1.19.

And for randomly picked degree distribution, the linear regression slopes have the same pattern. With the same amount of nodes, when the value of m is larger, the degree distribution shows a less inclined behavior in its linear regression slope.

Last but not least, we implemented the age of nodes VS the expected degree plots for m = 2 and m = 5, in order to figure out the difference between them.

m = 2	m = 5
	

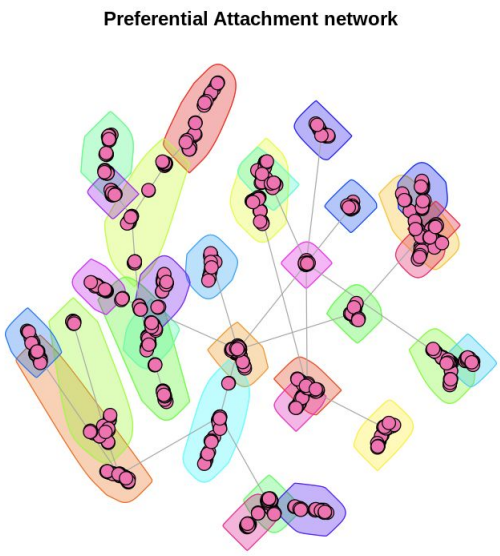
The graphs of age versus expected degree for m = 2 and m = 5 are shown above. And the estimated values for expected degree of nodes are 3994 for m = 2, and 9970 for m = 5.

This explains that with higher edge value, m, the expected degree becomes higher when the age gets larger towards the end. But the shape of the relation between them still remains a form of exponential.

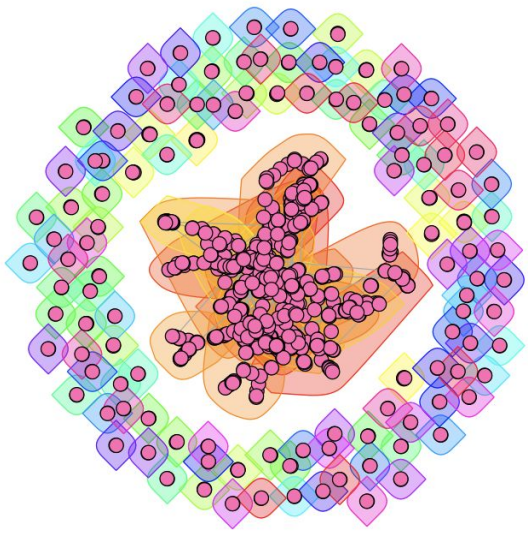
2(h):

In this question, we created the preferential attachment network (n=1000, m=1) firstly and then used its degree sequence to create stub-matching network via “simple.no.multiple” method as well as “vl” method.

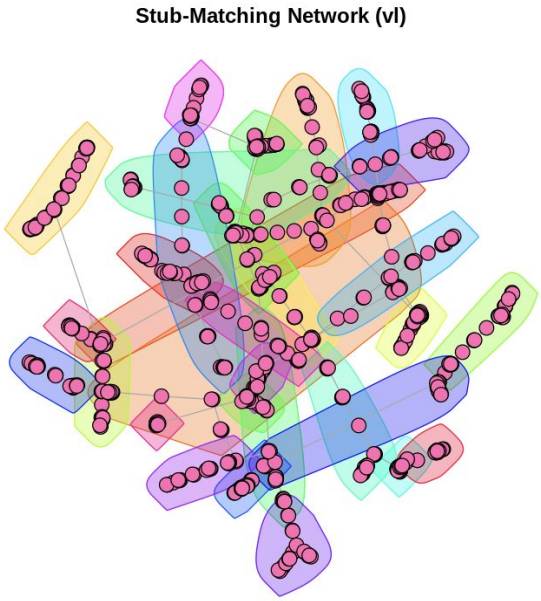
```
[1] "Preferential Attachment network: "
[1] "Modularity  0.933942951960973"
```




```
[1] "Stub-Matching Network (simple.no.multiple): "
[1] "Modularity  0.844237130022918"
```



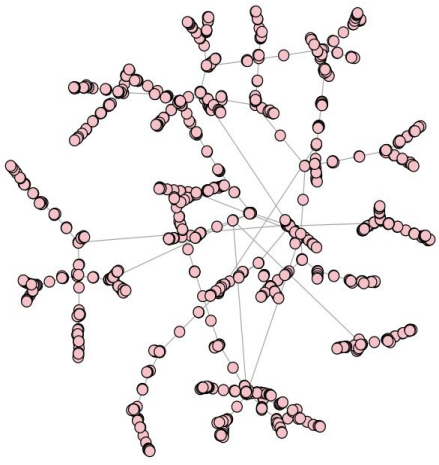
```
[1] "Stub-Matching Network (vl): "
[1] "Modularity  0.934340747153561"
```

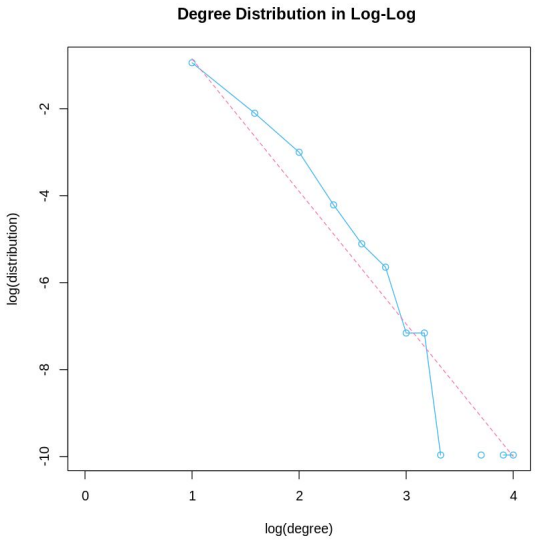


Under the preferential attachment models, new nodes are always connected to the old ones, which makes sure that the final network is connected and there won't be self-loop. However, in the stub-matching procedure, we generate all the nodes in advance and then try to establish connections among them. As a result, the final graph might not be connected and there might be self loops.

3(a):

With the given parameters, we created the preferential network:



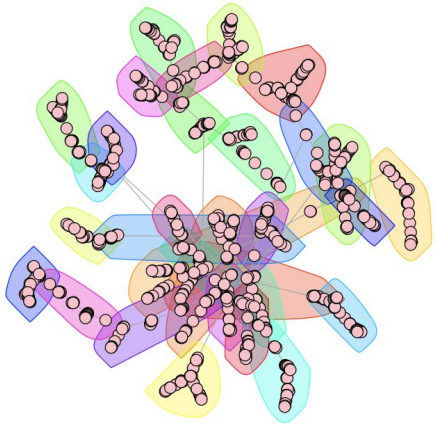


Here the power law exponent is around -3.05

3(b):

With the graph generate in 3(a), we got:

```
[1] "modularity:"
0.935067199331468
IGRAPH clustering fast greedy, groups: 31, mod: 0.94
+ groups:
  $`1`
   [1] 131 132 135 137 139 142 153 161 173 188 241 285 287 478 520 579 581 583
  [19] 591 614 621 624 693 695 727 733 734 739 740 744 746 751 756 764 768 778
  [37] 779 811 816 829 836 840 865 866 869 870 871 928 938

  $`2`
   [1]  10  11  19  21  22  25  26  35  36  38  41  42  46  53 100 120 122 123
  [19] 138 140 149 150 164 183 254 316 330 332 375 388 434 490 508 528 568 637
  [37] 721 856 863 878 880 881 882 883 884

  + ... omitted several groups/vertices
```

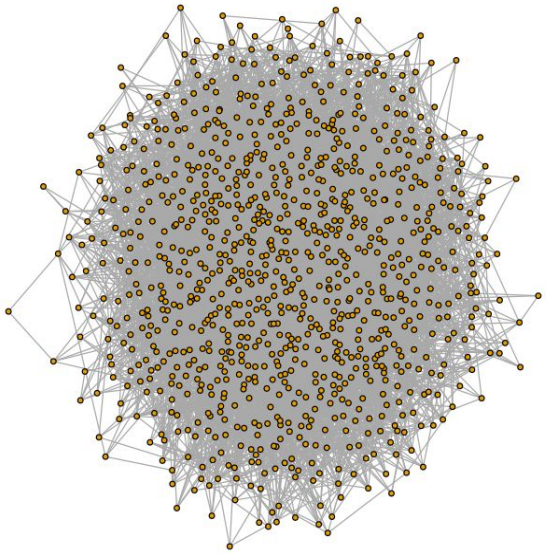


There are totally 31 communities and the modularity is around 0.9351.

Part 2:

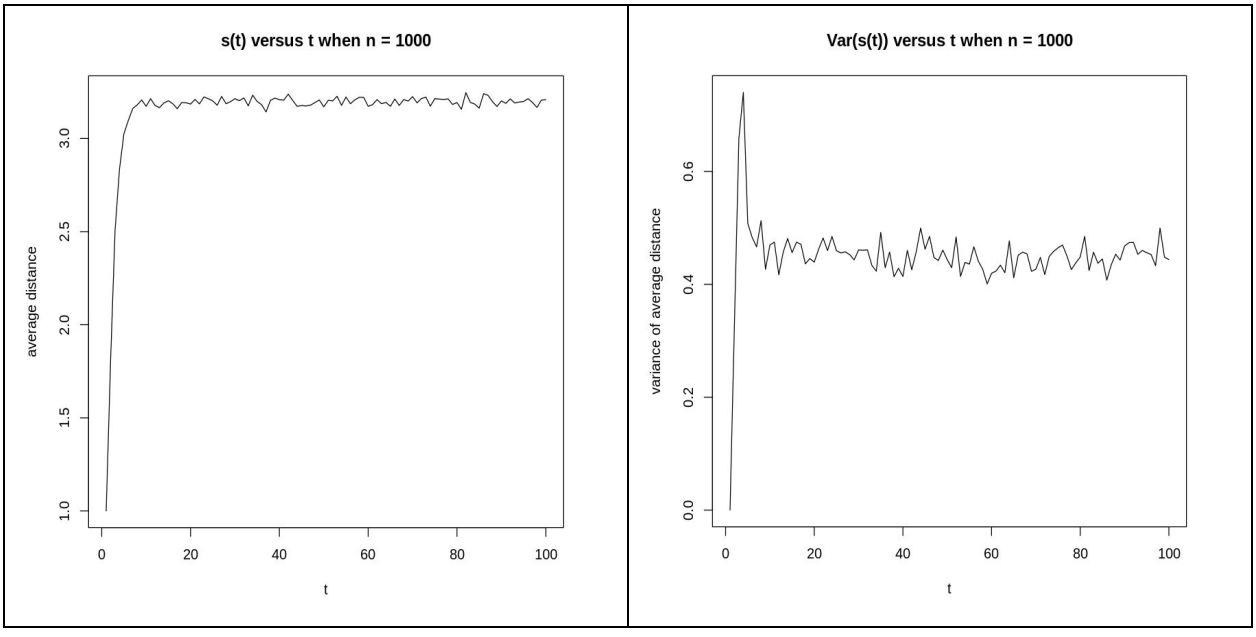
1(a):

Below shows the undirected random network with 1000 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.01.



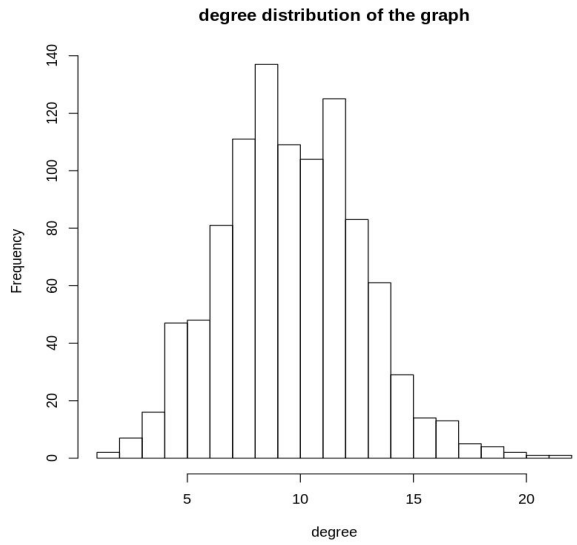
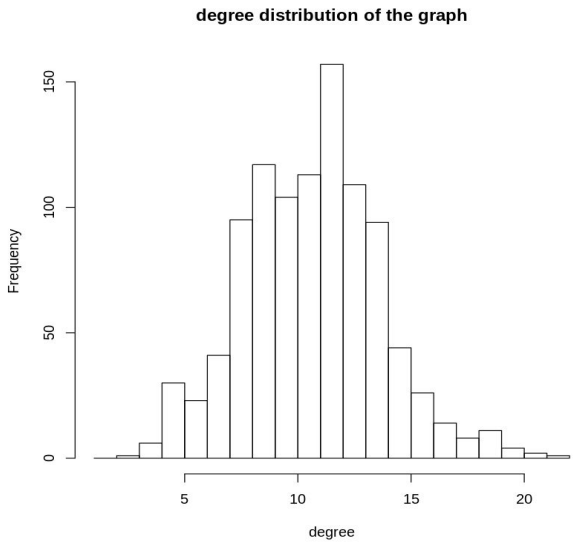
1(b):

Below shows the s(t) and Var(s(t)) v.s. t.
Here we only take the first 100 steps into account due to the size of the graph.
For both figures, the statistics we are interested in converges to a specific value and fluctuates around that.



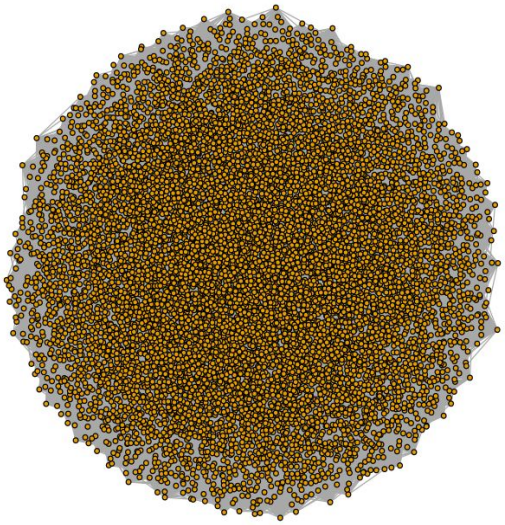
1(c):

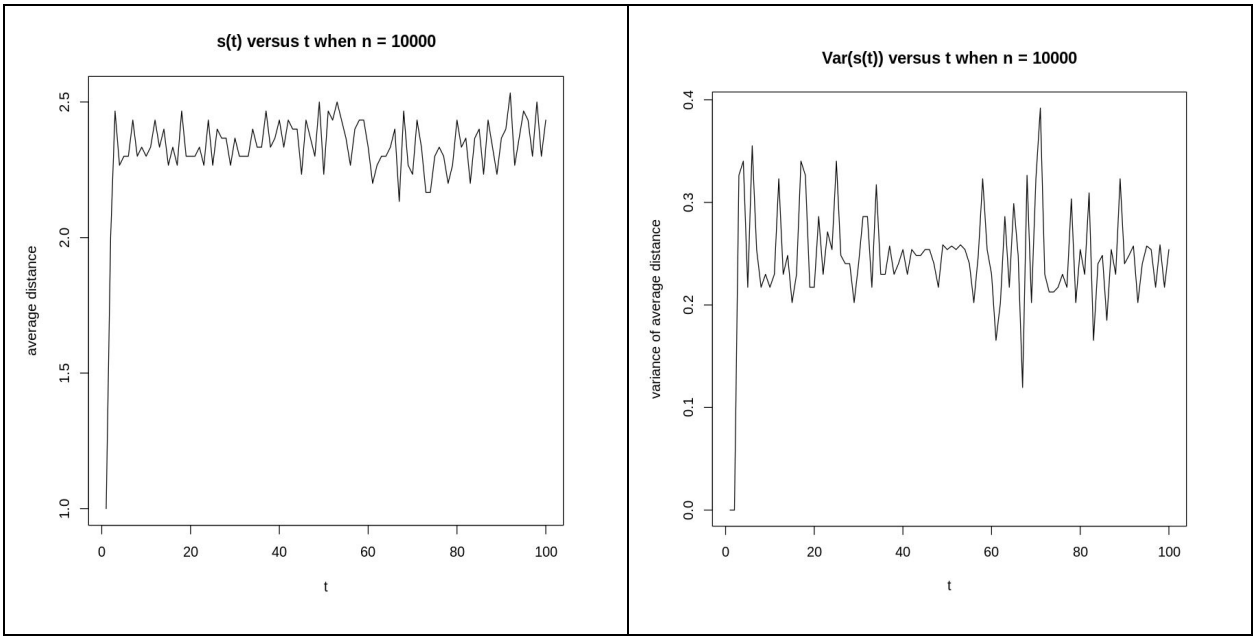
Below shows the degree distribution of nodes of the graph and the degree distribution of the reached nodes of the graph after random walk.
From comparing the two figures, we can find that the degree distribution of the nodes seems not to be influenced from random walking a lot as the two figures follow almost the same distribution.

Degree Distribution of nodes of the graph	Degree Distribution of the reached nodes
	

1(d):

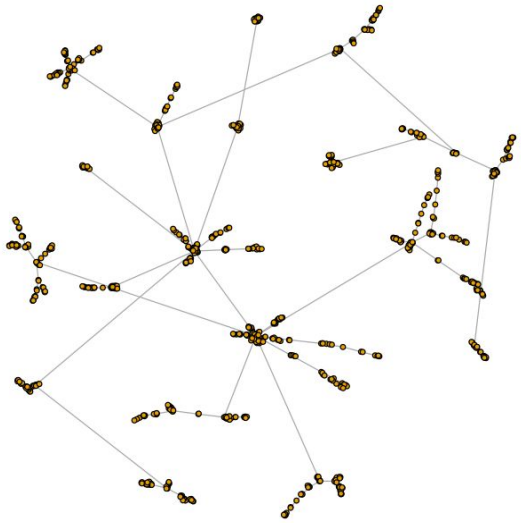
Below is the figure of an undirected random network with 1000 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.01 and its s(t) and Var(s(t)) versus t.
From comparing the s(t) and Var(s(t)), we found that the greater diameter seems to make the degree distribution more stable as the variance is lower. Also the average distance is lower as there are much more edges than before.





2(a):

Below shows the undirected preferential attachment network graph with 1000 nodes and each new node attaches to m = 1 old nodes.

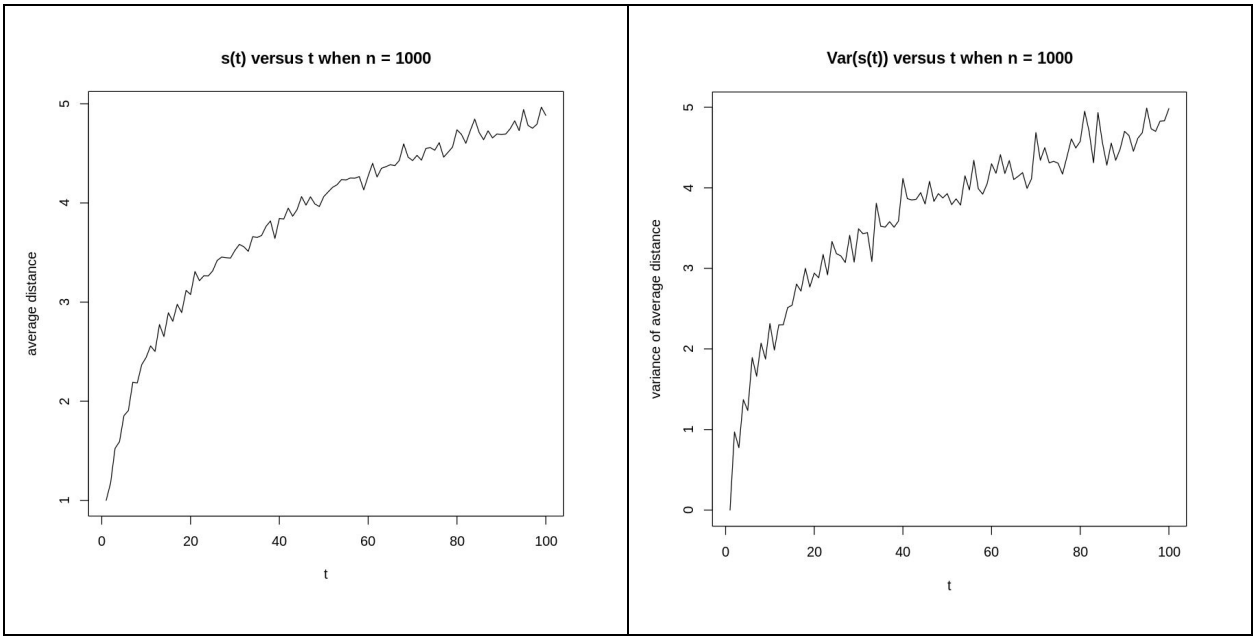


2(b):

Below shows the s(t) and Var(s(t)) v.s. t.

Here we only take the first 100 steps into account due to the size of the graph.

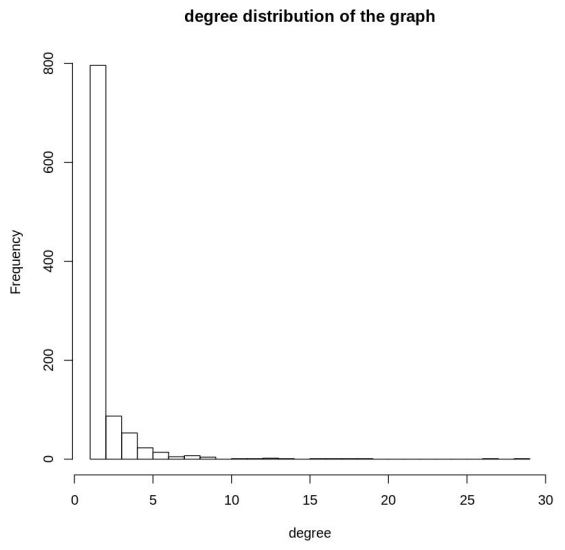
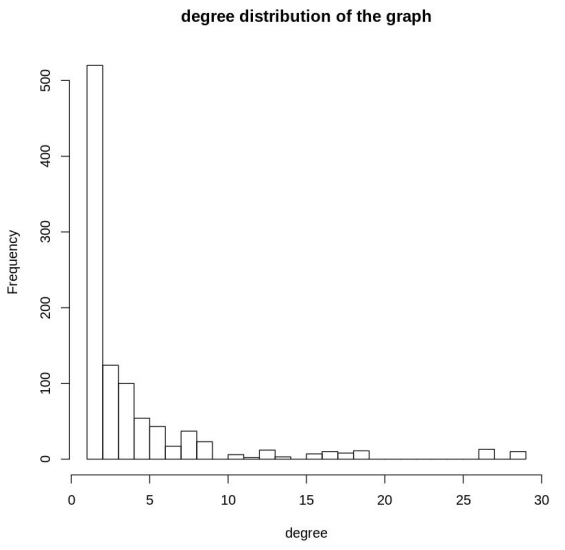
Unlike in problem1, both figures do not converge into a specific number but they both show a tendency to fluctuate around a log-like function.



2(c):

Below shows the degree distribution of nodes of the graph and the degree distribution of the reached nodes of the graph after random walk.

From comparing the two figures, we can find that for the degree distribution for the nodes reached, the variance is clearly smaller than the degree distribution of the actual graph. As the degree 2 nodes get much smaller and quite a number of high-degree nodes appear in the figure.

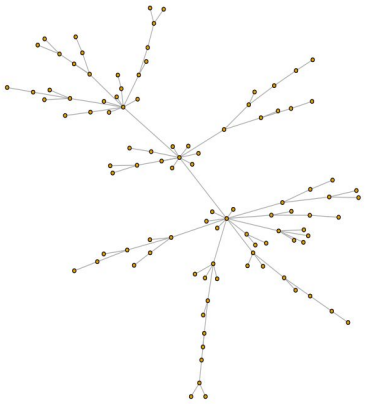
Degree Distribution of nodes of the graph	Degree Distribution of the reached nodes
	

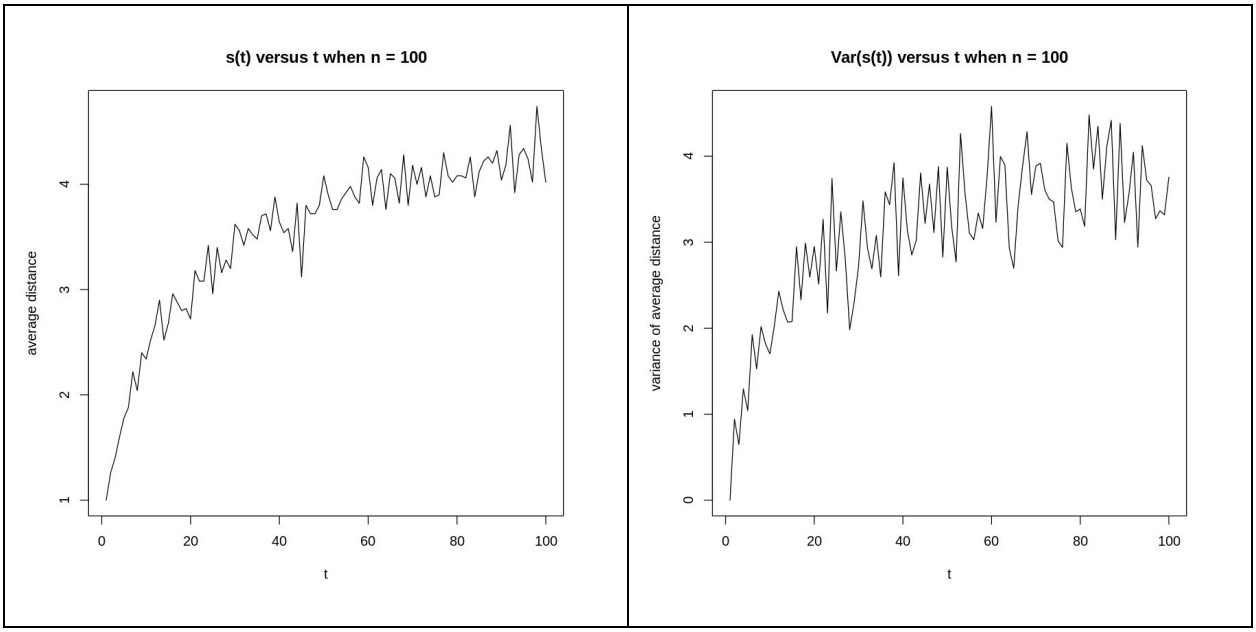
2(d):

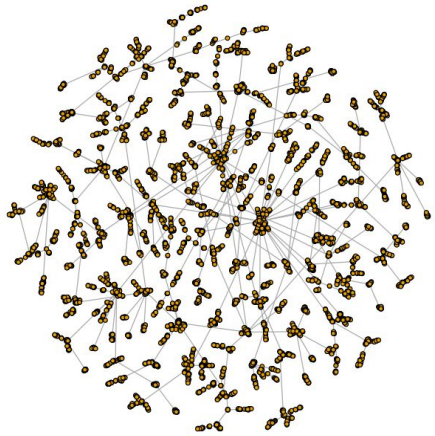
Below shows the undirected preferential attachment network graph with 100 and 10000 nodes and each new node attaches to m = 1 old nodes.

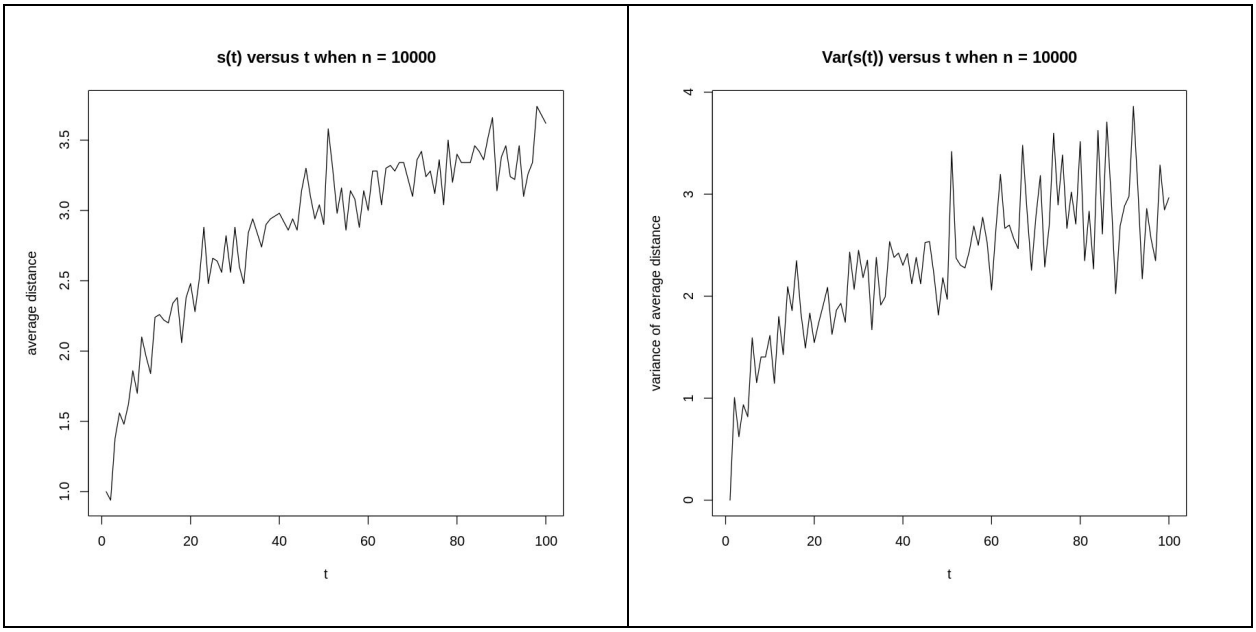
Then the s(t) and Var(s(t)) v.s. t. for both graphs are shown.

From comparing we can see, unlike the situation for Erdos-Renyi graph model, for the preferential attachment model, diameter will not greatly influence s(t) and Var(s(t)). They follow an almost the same figure for different diameters.



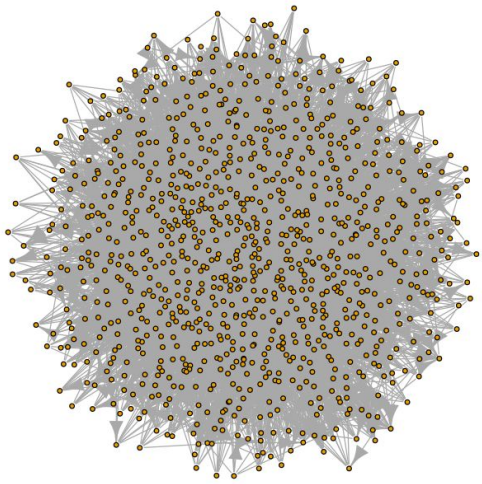

s(t) versus t when n = 100
average distance
t
Var(s(t)) versus t when n = 100
variance of average distance
t





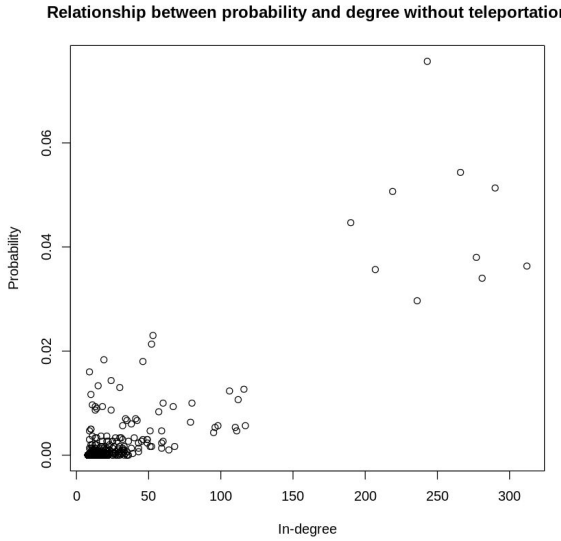
3(a):

Below shows the figure by combining two directed preferential attachment network graphs with 1000 nodes and each new node attaches to m = 4 old nodes with the second graph's edges shuffled.



Then is the relationship of the visited probability and in-degree of nodes

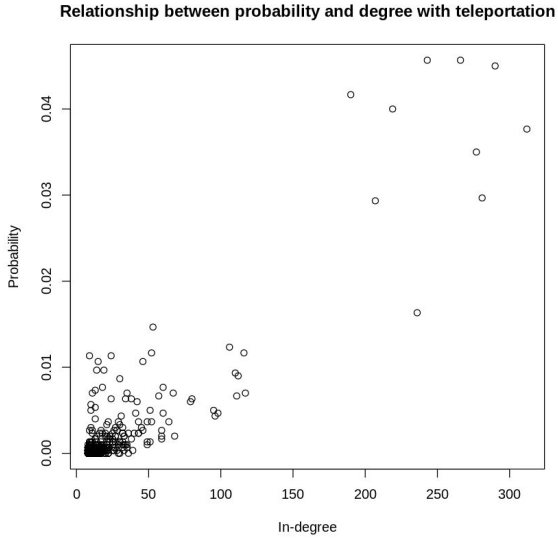
For the correlation of the two parameters, the correlation is 0.8675534. Which is pretty high. So we can say that higher in-degree nodes have a higher chance to be visited during random walk which also follow the intrinsic.



3(b):

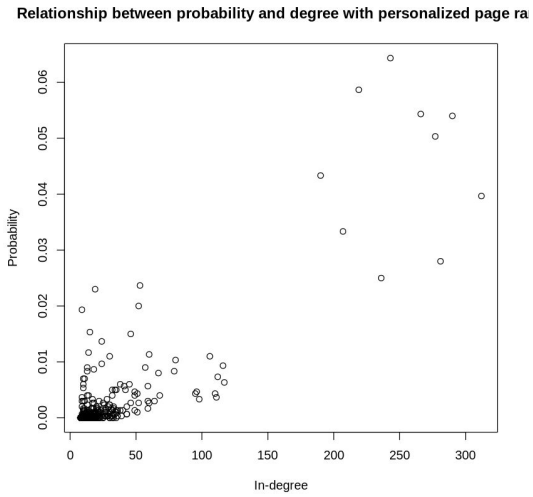
If allowing a teleportation rate of 0.15 during the random walk, the relationship between the visited probability and in-degree of nodes will be as follows, the correlation is 0.9070502.

Same as before, a very high correlation and even higher than before, this may be a result of teleporting actually can be seen as reducing the number of steps taken. Actually increase the chances that it will be stuck around the high in-degree nodes.



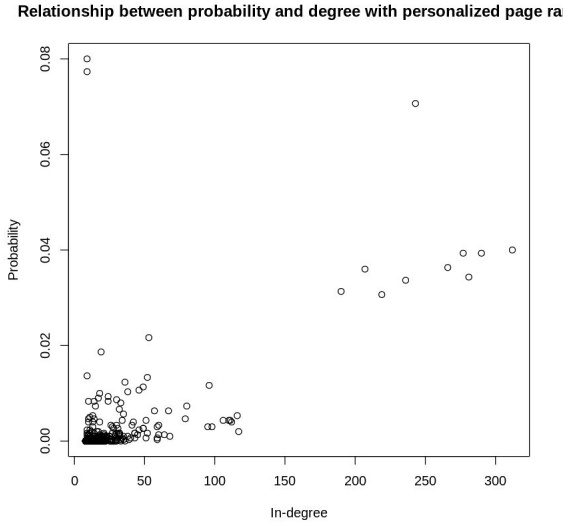
4(a):

If using the personalized pagerank, the relationship between visited probability and in-degree of nodes is shown below. The correlation is 0.8668288 which is very close to the correlation of random walk without teleportation. Therefore, we can conclude that using the pagerank from random walk for transportation will actually lead to the similar result of random walk without teleportation



4(b):

If using the personalized pagerank, and the median page rank two nodes have 0.5 each probability to be teleported to, the relationship between visited probability and in-degree of nodes is shown below. The correlation is 0.6713628, greatly lower than before. Therefore, letting the median pagerank nodes to be focused can help reduce the effect of in-degree of nodes.



4(c):

Originally, pagerank is only related to the transition matrix and teleport probability. Considering what we have done in 4(a) and 4(b). We will only teleport to some trusted pages. Therefore, for the teleportation probability to each node, we can modify it to be node-related or person-related such that for given nodes, if you teleport from that, it will only teleport to several trusted links of this site or several trusted links of the given user. This will make the self-reinforcement take into account our PageRank equation.