EE232 Graphs and Network Flows
Homework #1

Sidharth Gulati – 104588717
Tushar Sudhakar Jee – 004589213
Chandan Dhal – 904588105

Introduction

Igraph is a tool used to generate graphs and networks. In this homework we generate different kinds of random networks and analyze the different parameters of a network such as, diameter, connectivity and modularity of the community structures or clusters.

Exercise #1

Static Random Networks

We used Erdos-Renyi model to generate random undirected network. **Erdos-Renyi** model has two closely related variants [1].

- In the $G(n, M)$ model, a graph is chosen uniformly at random from the collection of all graphs which have n nodes and M edges.

- In the $G(n, p)$ model, a graph is constructed by connecting nodes randomly. Each node is included in the graph with probability p independent from every other node. As a result, the nodes of the network generated using this model have a binomial degree distribution with parameter p. $p^M(1-p)^{\binom{n}{2}-M}$.

The G(n,p) was used to generate three undirected random networks with 1000 nodes with probability p 0.01, 0.05 and 0.1 respectively. The various parameters of a network can be defined as:

- If all the nodes in the network are connected, or there are no isolated nodes, then the network is **connected**.
- **Diameter** is the length of the longest geodesic.
- The **degree** of a node in a network is the number of connections it has to other nodes and the **degree distribution** is the probability distribution of these degrees over the whole network.

The various igraph functions used in this homework are:-

Functions	Task
is_connected()	To check if network is connected
diameter()	The longest geodesic length
get_diameter()	Returns the path of the diameter
farthest_vertices()	Extreame vertex idsconnected by the diameter
degree()	Degree of the network
degree.distribution()	Degree distribution of the network

Network analysis for p=0.01

The network plot for the probability 0.01 is depicted as below in figure 1.

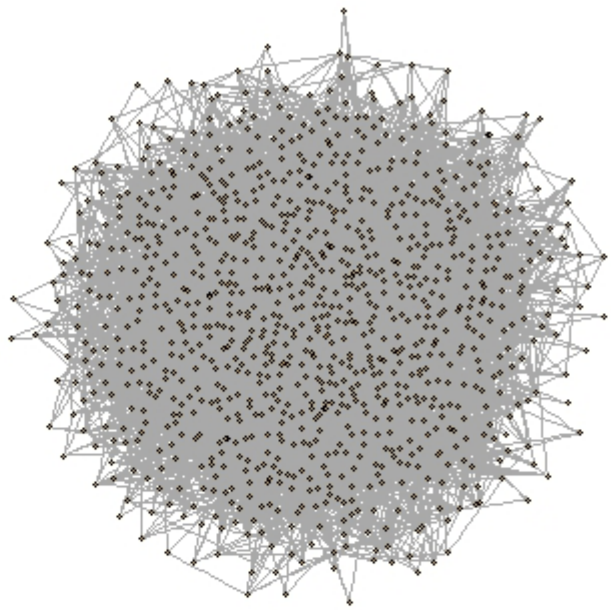


Figure 1. Network with probability 0.01. It can be inferred that the network are connected

The various network parameters are given below:

Parameters	Values
Connected	TRUE
Diameter	5
Path of diameter	1 -> 359-> 25-> 110-> 543-> 933
Farthest vertices of the diameter	1 , 933

```
> degree(g_0.01)
   [1]  6  7 10 11  7  8  6 14 11  8  9 22  5  6  8  5 11 15 12  9 16 10  5  6  9 10  7
  [28] 11 10 19 12 11  7  3 10 11  9 10 10  8 11  8  4  9 10 17 16  9 13  5  9 10 15 10
  [55] 10  8  7 15 10 13 11 12  7  7 15 12  9 10  8 10 14 11 10 11  7  9  8 18  6  7 13
  [82]  9  8 12 10  9  5  8  8  8 15  7 18  8 11  8 13 16  2 12 10  7  9  9  6  7 12  8
 [109]  8 10 10  5  7  9  7 12 11 12 16  4  4 14 12 12  9  9 14  5 10 10 10 11 12  5 13
 [136] 11 10 15 11 10  8 10  7 15 10  9 11 11 12 10  8 16 12 11  6  6 14 15  7  7 15  9
 [163] 10 14  8  8 15 16 18 15 10 18 12 14  9  8 16  8 13  9  7 17 10  6 12 15 13  9 10
 [190] 13 11  7  9 13 15 13 12 13 15 15  8  6 12 17 10 14  6  7  8 10  8  9  9  7  6 11
 [217] 10  8 12  6  7 16  9 12 10 13 11  8 13  7  8 13 13 12  9 11  9 11  9  6 10 14 14
 [244] 12  9  7 12  6  7  7  5 19  7  8  9  6  7 12 12 10 11  8  9 15 10 12 10 10 13 10
 [271] 13 14 11 13  4 10  9 11  9 14 10 13 12 12 11  9  3  9  6  6 10 10 11  9 16  3  8
 [298]  9  4 14 10 13 11  8  8  4 12  8 11 12  2 11 10 15 12 17  7 11  9 13  8 15 12 13
 [325] 18  9 16  8 14 16  8 15 10  9 12 13 14 12  9  8 12  9 11 13  5 11 14 11  9  8 14
 [352] 10  7 11 12 13 17  7 11 10  8 14 14 15  8 11  7 15 12 11 13  8 11 14 11 11 14  6
 [379]  8 12  7 13 16 14 16  8 11 11 12 15 10  9 10 18 12  9 11 13  7  7 11  7 11 10 13
 [406] 13 16  5  7 13 12  9 12  8 12 14  8 11 10  7 10 12 12 13 10  8 10  8  9 13 13 14
 [433]  7 12  9  4  8 15 19  7  9  6  5 13 10 14 12 12 10 10 11 12  7 13 13  8 12 12  5
 [460]  9 11  5 12  4 12 12 10 13 15 14 11 14 10  7  9 12  8  8 12 11 12 10 14 12  9  9
 [487]  3 10  8 11 11  8  6  8  9 10 13 11 11 12 14 25 11 14  7 11 10  8  9  8  9 11  5
 [514]  9 13 14  9  9 16 17  8  8  7 10  6 10  6  9  9 10 13  9  9 10  3  7 13  5  5  8
 [541]  9 12  8  9  7 14  5  9  6 10 10 14 10 12 12  7 11  9  8  6  9 10  7 13  5 12  9
 [568]  7  9  9 15 12  7 13  6  6  7 13 13 15  5 10 10 10 21  9  9  8  6 13 12  6 16  8
 [595]  7  4  8 13 10  5 10 11 13 13  6 10  9 10 11  7  8  9 11 13 16  8 14 10 10  7 10
 [622]  9 10  8  6 10  8  5 10  9 17 10 15  8 12 14 11 13  5 10 15  9 12 11 12  8 13 10
 [649]  4  9 13  5 14  7 12 10  9  6  5 11  8 10 13  7 12 10 15  3 11 14 14 11 12  8 12
 [676] 10 13 13 12  7  3  3 13 13  8 11 14  8  7 13  5 12  9 17  6  7  9  4 13 11  6 11
 [703] 14  4 11 12  9  8 10  8 12  6  5 10  7  7  9 10 13 14  9 12  7 11 10  8  6 15 11
 [730] 11 12  8  8 12 16 11  9 10 10  7  3 10 10  4 13  8  9 16 12 11  5  6 10 10  6  8
 [757]  6 13  9 10 12  8 12  7 11 10 10  4  6 14  9  8  8  8 15  9  6  5 15  9  6 12  4
 [784]  6  9 13 13 18 19  9 13 10 12  6 12  9  5 14 11  9 11 11  7 10 13 13  7 12 11  8
 [811] 13 15 11  9 13  9 17 10  7  6 18  9 17  7  6 10  7  9 10 11 12 11  7  8 10 16  8
 [838] 10  8 12 10 13 10  7 11 16 11 10 10  8  5  9 16 11  7 10 11  9 12 15 13 14  9 11
 [865]  8 11 13 11 10 13 12  4  9 13  7  3  9  6 11 10  6 17 13 15 10 10 10  3 10 10 12
 [892]  5  9  8  3  8  8  8 11  7 11  5 18  6 12  4 15  7  8 10 13 13  7 11  9  6 13  8
 [919] 11  7  8 11 12  7  8  7 16  7  5 14 14  9  5  6 12  3 14 10  6  5 16 11 14  8 12
 [946]  7  5  6 11  7 11  8  9 10  6 12 11 13  8  8  7  8  9  3  6 15 11 17 12  6 14 11
 [973] 12  8  8 11  4 14 10 10 14 10  7 10  7 15  9  6 12 13 11 14  8 13 10  5  7 11 13
[1000]  7
```

Figure 2:- Degree of each node. Snapshot of Rstudio console.

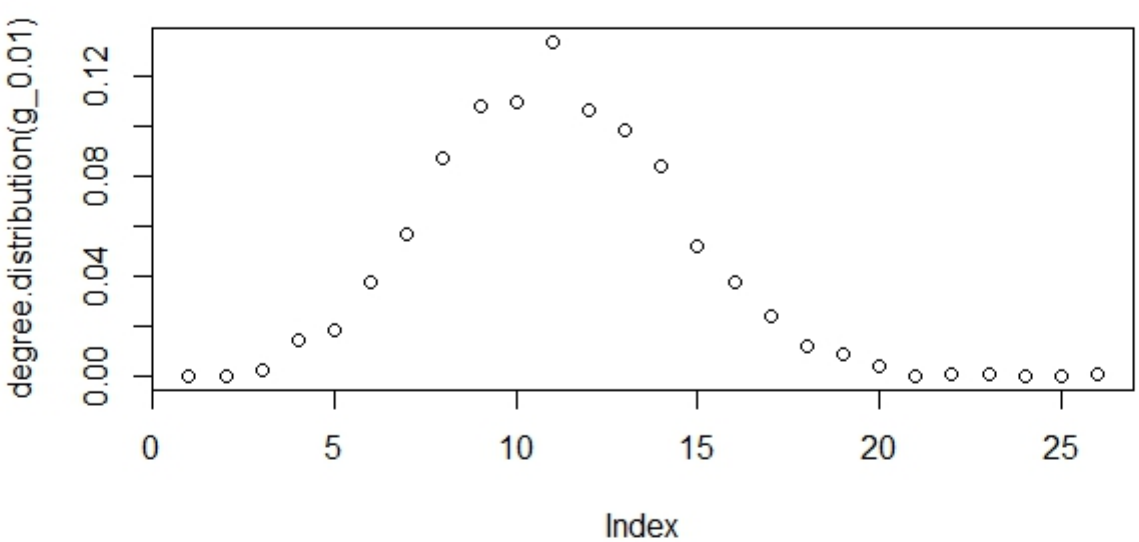


Figure 3: Degree Distribution of the network

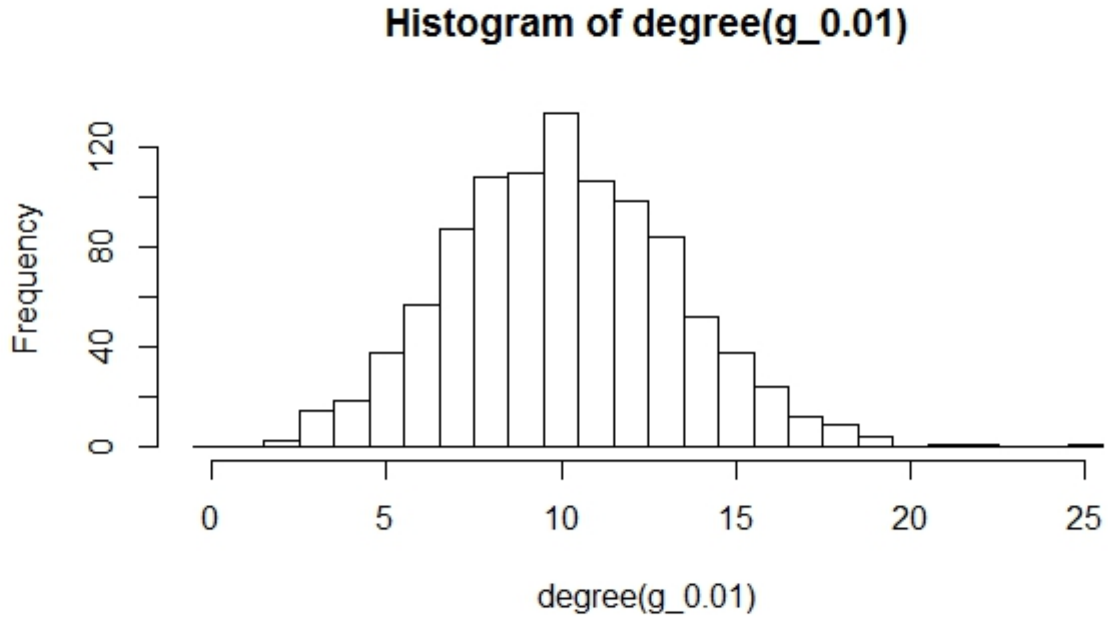


Figure 4: Histogram of degree distribution

Network analysis For p=0.05

The network plot for the probability 0.05 is depicted as below in figure 5.

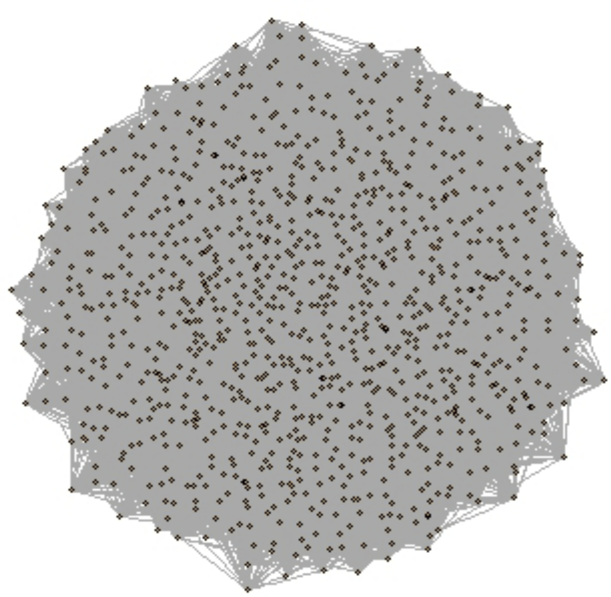


Figure 5. Network with probability 0.05. It can be inferred that the network are connected.

The various network parameters are:

Parameters	Values
Connected	TRUE
Diameter	3
Path of diameter	1 -> 70-> 30-> 69
Farthest vertices of the diameter	1 , 69

```
> degree(g_0.05)
   [1] 35 49 44 52 60 48 53 55 53 50 51 46 48 52 47 48 45 48 48 60 56 42 46 49 46 62 53
  [28] 52 61 53 46 50 60 49 44 47 57 44 50 53 40 50 50 56 47 50 49 67 48 43 59 49 51 43
  [55] 44 38 52 43 52 44 54 40 53 46 51 47 58 44 49 48 55 50 51 43 58 43 43 48 44 59 54
  [82] 53 53 62 45 55 56 61 48 44 62 51 46 47 44 49 46 60 50 47 55 68 43 54 61 58 61 51
 [109] 43 58 59 63 41 48 51 66 37 53 58 42 36 47 40 48 58 62 49 60 50 53 46 55 55 44 51
 [136] 43 47 55 46 51 47 60 51 45 31 53 55 41 47 50 48 41 62 53 49 59 50 49 64 51 48 44
 [163] 57 46 53 37 53 47 46 51 61 51 36 58 46 55 61 50 47 55 42 50 44 38 43 62 51 57 44
 [190] 52 56 50 46 48 50 62 69 44 68 51 54 48 50 46 62 56 44 62 48 49 45 64 57 49 51 60
 [217] 58 43 42 48 50 52 45 52 41 66 53 37 48 59 56 48 59 55 57 47 54 54 48 54 49 42 48
 [244] 45 44 54 49 56 44 58 51 49 47 54 50 64 45 55 43 39 56 52 49 54 37 51 53 45 56 42
 [271] 61 42 47 45 42 48 50 53 52 49 61 47 49 47 61 48 51 53 46 50 57 53 49 54 56 39 28
 [298] 48 54 51 54 54 52 57 63 48 60 52 53 51 45 63 52 51 52 60 50 49 55 62 46 47 42 45
 [325] 46 48 33 52 51 41 38 52 49 49 47 46 56 50 75 43 52 59 58 42 41 37 66 47 52 52 60
 [352] 46 44 54 58 47 41 47 42 56 60 62 64 49 53 61 59 57 42 42 60 60 51 58 43 44 42 53
 [379] 43 51 56 46 47 50 49 46 44 47 40 51 53 44 51 53 39 46 49 52 57 49 46 43 66 41 47
 [406] 57 50 56 53 50 37 53 54 45 49 42 53 64 44 57 43 50 54 43 56 35 63 37 46 44 54 51
 [433] 51 50 43 43 50 41 51 58 58 63 53 54 48 55 44 46 50 45 50 42 62 50 43 51 51 61 59
 [460] 55 54 67 50 48 42 54 63 49 39 50 62 53 65 47 65 52 51 46 64 54 51 65 56 58 50 47
 [487] 58 54 42 48 38 58 62 45 55 55 50 58 53 42 49 47 39 61 41 53 36 51 63 47 60 52 45
 [514] 42 49 48 56 49 42 38 51 46 46 55 50 49 55 49 56 40 58 40 44 44 43 46 56 60 58 42
 [541] 64 44 43 45 50 53 57 46 45 59 63 37 56 35 34 53 42 45 49 64 33 45 39 52 55 56 46
 [568] 54 48 39 60 26 47 52 49 39 49 45 46 50 52 50 50 55 44 58 50 44 46 57 51 36 47 52
 [595] 48 52 45 56 49 61 39 52 47 45 49 66 54 57 45 40 56 48 57 41 47 60 36 51 53 47 52
 [622] 48 42 52 54 51 46 53 46 50 39 54 39 46 61 52 61 44 40 41 45 53 44 58 53 45 39 48
 [649] 56 53 51 56 43 37 42 61 56 44 49 53 46 52 44 47 48 50 54 40 51 54 52 54 50 45 41
 [676] 57 42 53 49 50 58 57 57 42 54 57 49 45 50 48 52 59 47 40 53 42 49 58 51 58 63 42
 [703] 57 52 48 46 47 55 55 54 54 66 45 38 44 54 46 48 42 47 47 54 55 45 50 47 38 44 47
 [730] 41 44 59 58 57 45 51 54 51 53 46 51 45 62 54 51 51 49 48 42 43 59 44 41 42 38 44
 [757] 54 57 54 45 51 45 62 44 40 59 46 44 53 55 52 55 50 45 50 49 54 55 53 48 58 51 44
 [784] 59 55 46 47 49 47 56 49 57 54 53 36 47 49 50 62 50 53 56 55 59 53 43 47 62 51 53
 [811] 42 53 60 45 49 44 61 53 55 59 53 41 42 56 52 57 54 44 58 49 39 57 41 42 48 54 55
 [838] 67 49 55 54 41 53 62 52 52 53 54 61 49 50 62 52 44 47 53 64 55 57 45 49 54 49 46
 [865] 43 58 60 60 48 52 51 47 58 62 51 49 57 48 50 34 59 41 54 47 47 50 57 57 50 59 60
 [892] 46 44 50 55 51 46 50 58 44 40 56 47 63 50 47 50 67 48 56 47 46 50 56 47 52 54 46
 [919] 55 56 59 47 59 49 52 52 57 46 46 48 53 59 52 49 50 53 49 51 46 44 54 36 54 53 60
 [946] 50 40 44 56 48 44 49 37 39 53 57 53 68 59 46 52 56 46 60 45 66 53 54 55 59 59 46
 [973] 64 51 64 50 47 49 42 46 40 62 47 55 46 43 52 59 56 44 50 57 44 52 51 53 35 44 51
[1000] 56
```

Figure 6: Rstudio console sample of degree of nodes

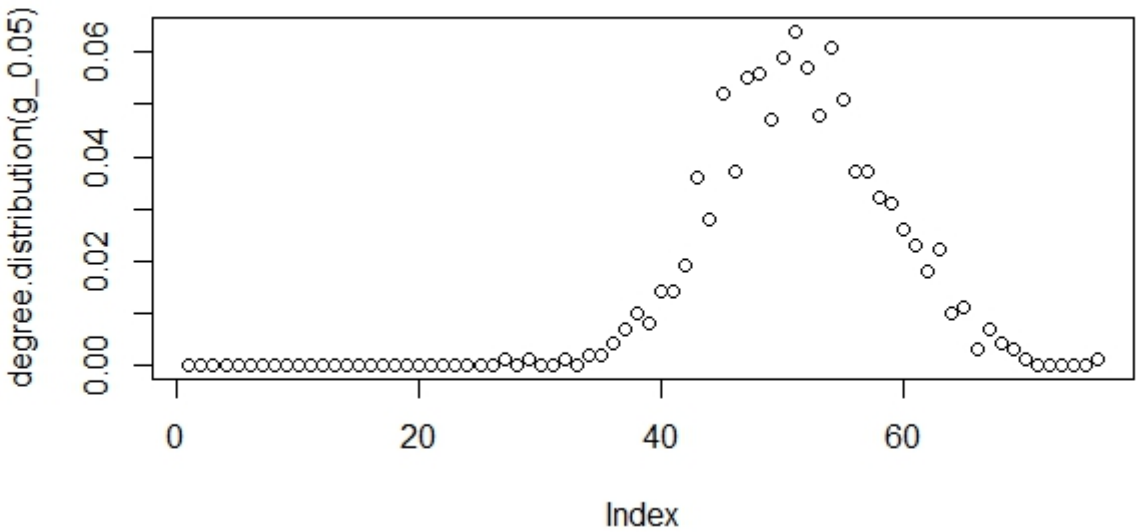


Figure 7: Degree distribution of the network

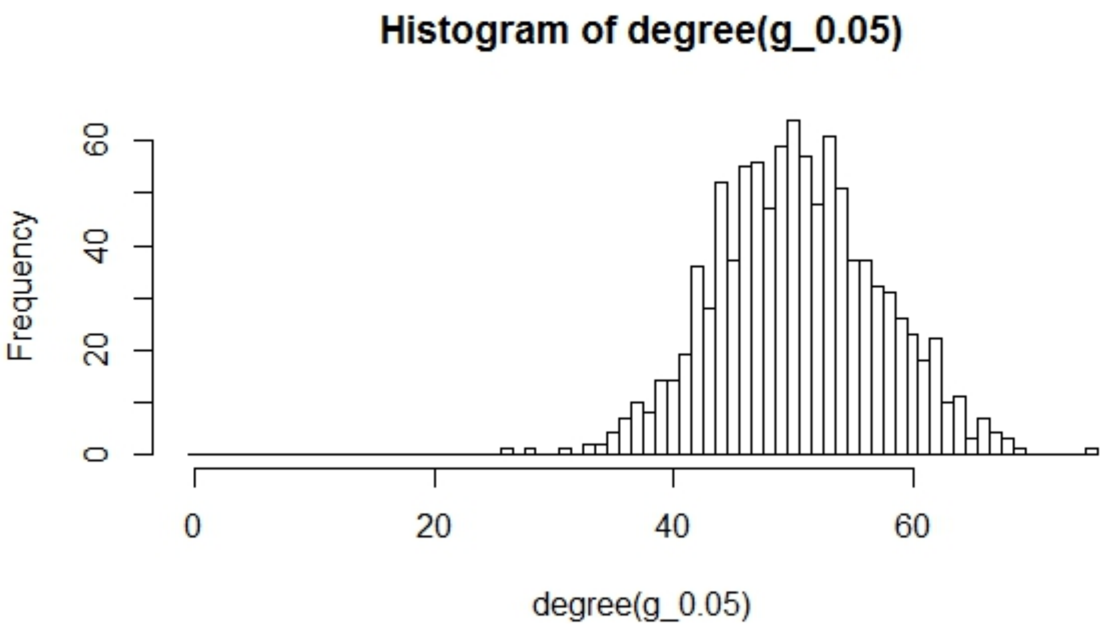


Figure 8: Histogram of degree distribution

Network analysis For p=0.1

The network plot for the probability 0.1 is depicted as below in figure 9.

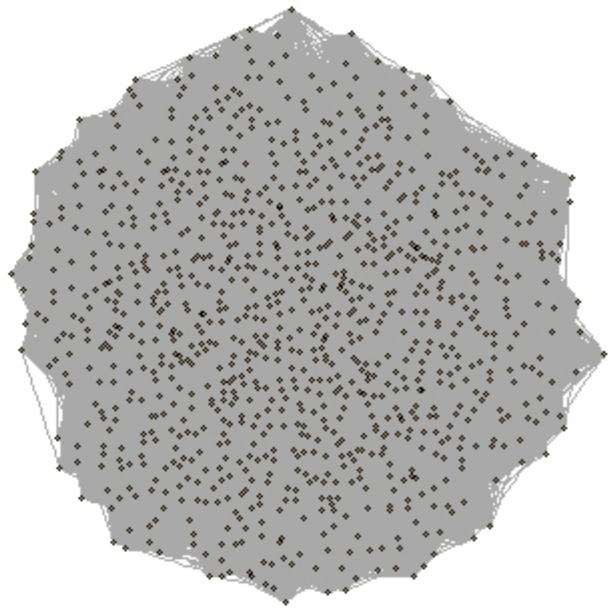


Figure 9. Network with probability 0.1. It can be inferred that the network are connected.

The various network parameters are:

Parameters	**Values**
Connected	TRUE
Diameter	3
Path of diameter	1 -> 6-> 448-> 802
Farthest vertices of the diameter	1 , 802

```
> degree(g_0.1)
  [1] 102 109  88  69  90  97 113  97  92  96  97 113 100 104  85  93 104 115  96  96
 [21] 110  94 103 107  93 117  93  99 102 107 102 105 130 110 114 101 118  87  93  95
 [41] 110 103 100 108 100 100  96  92  95  96 102 100 109 115  98 101  92 102 105 106
 [61] 104 104  94  82  83 104  98 103 102  92  88  87 111 106 105 114 109 118 105  97
 [81]  96  87  99  88  98  99 100  96  96  98  94  96 119 104  97  93  98 102  89 115
[101] 100 105 109  88  79 108  95 111 101  97  98 106 115 101 116 110 114  97  90 111
[121] 113  83 102  97 104  96 105 104 120 107 103  88  92  90 101 102  94 110 102  93
[141] 112 112 112 114  98 119  97 113  98 103  90  84 105  96 104  94 107  81  97 120
[161]  93 117  92 107 107  86  95  78  91  97  93  95  93  94 105 102 109  97  97  97
[181]  97  88 104  97  90  90  98  83  96 104  88 104 101 113 110 107  91  96  91  94
[201] 120  89  98  80 105  93  95  91 109 102  91  96 104  88  93 104 111  97  97  88
[221] 100 113  80 107  89 110  90 116 112  86  88  95 113  94 105  96  97 104  99 115
[241] 108 105  92  93 113 107 105  96  96  96 106  89  99  94  90 114  95 113  95  81
[261] 109 121  88 103 112 106 118 100  99  97  79  90  97  93  91 102 102  95 119  91
[281] 106  89 128  89 104 107 116 108 109 101 108  99 103 101 123  99  93 112  90 102
[301] 101 103  99 101 102  91 104 128  92  97 109  88  96  79  93  99  77  93 112  96
[321] 105 107  94  84  97  99  98 117  96 119  85  98  97 108 114  99 100  94  79 102
[341] 106 112 108 112  93 106 109 102 113 101 106 115 103  98  94  95 122 104 102  95
[361]  85 109  88 107  91  92  80 114 103 109  91  96 107  90 106  99 106  97  91 112
[381] 103  88 113 108  97  98  99  81  81 109  95  90  98 103 102  96  97  86  80  98
[401] 105 104  98  86  96 108  98 103 109  98 102 107 102 107 103  95 119 102 105 113
[421] 104 105  89 106  97 110 117 103  83  97  94 102  88 118  88 105  99 103 111 112
[441]  87 102  92  97 104  97  89  94  82 100  89 113 104 102 109 114  93  98 110 105
[461] 109  91  93  90  91 114 103  86  98 112  88 107 105  97 109  95  85  95  92  99
[481]  89  97 108 100 103  96 114 107  99  96 112  96  96  86 100 100 102  87  96 110
[501] 101 102  89 112 124  86 108  95 109  89 114 107  85  90  81  96  92  97 108  96
[521]  99  85  97  92 108 105 103  96 113  86  90  96  93 103 109  92 117  95 103 108
[541]  99  92  91  80  94  99 106 104 101  84 107  95  97 106  94  98  95  95  91  93
[561]  91  84  93  85  96 104 108 123  96  95  86 110 107  99 105  83  96  87 109  97
[581]  79  98 102  92  88  98  99  84  78  92 108  91  85  96  84  85 106 114 101 106
[601] 104  89  91  94 106  99 104  99  93 106 108 107  85 117  92  89  98  98  87 101
[621] 104 121  97 100 103 110  82 111 102  87 106  81  99 113 114 102 101  90  96 114
[641]  90  65 104 102 101  91 102  96 118 114  83  95 117 101  98 106  99  96  99 103
[661] 106 111 108 117 113  97  93  92  83  96  95  87  85 110 105  94  99  96  97  96
[681] 106 106  93  95 101  98  86  91  94 109 103 110 108  95 112  90 104 106  99 111
[701]  98 101  99 110 109  85 102  94  93 116 106 105  97  95  85  86 101 100 113  95
[721] 100 106 104  94 109 106  94  86  97  99 117 108  84 110  96  85  98  98  75  97
[741]  92  96  92 116  90 107  93  91 107 109  88 104  94 111  98  79 112 100  98 108
```

Figure 10. Rstudio console sample of degree of nodes

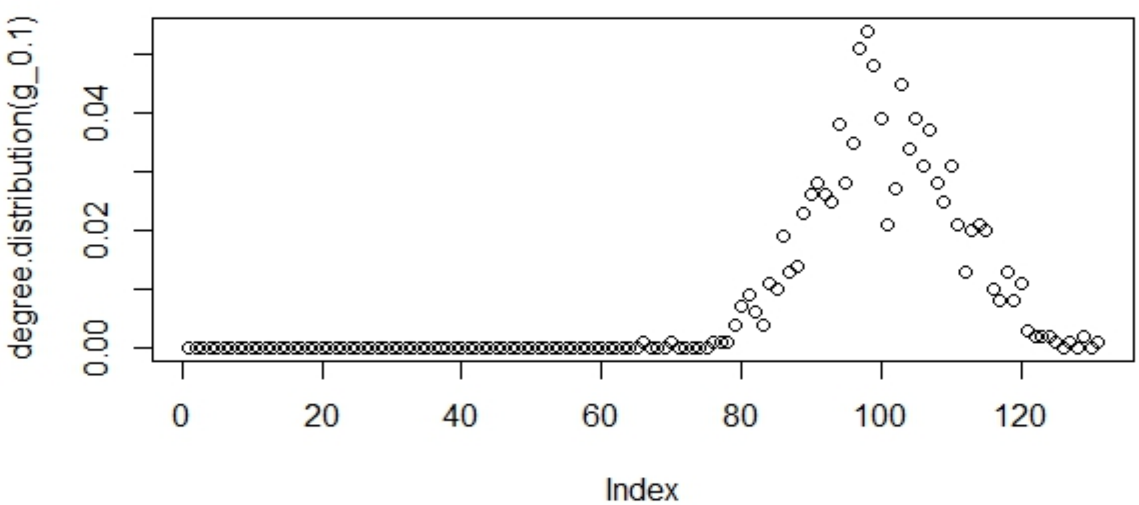


Figure 11: Degree distribution of the network

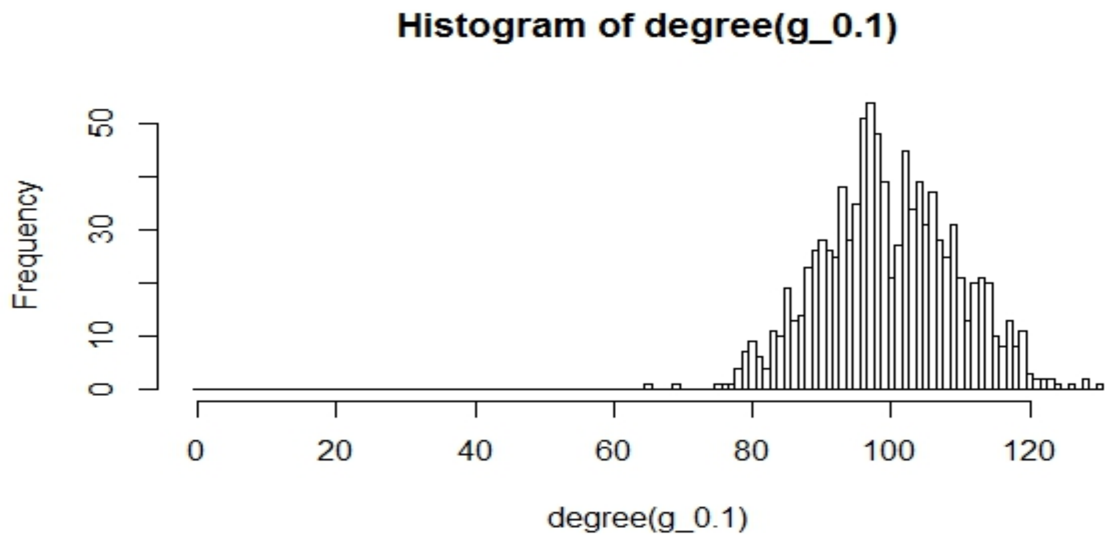


Figure 12: Histogram of degree distribution

Thus, the maximum node degree increases as the probability of drawing an edge between two arbitrary vertices increases.

Percolation Threshold

In statistical physics and mathematics, **percolation theory** describes the behavior of connected clusters in a random graph [2]. The threshold is the probability below which the network does not contain any large connected community (Clusters) with in the network; while above it there exists a giant component of the order of system size. We plotted the ratio of the size of the giant-connected component to the number of nodes in the network (1000) versus the probability sequence from 0.0001to 0.01 in steps of 0.00001.

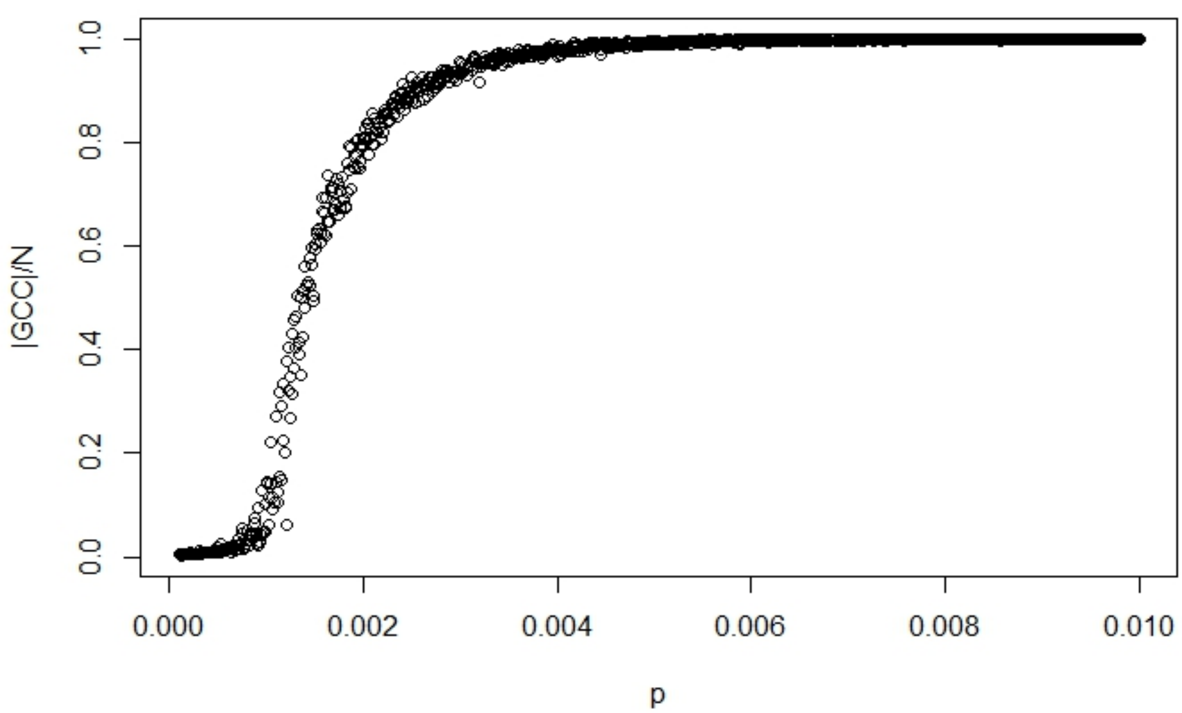


Figure 13:- Percolation threshold plot

Connectivity test

From the above plot in figure 13, we can infer that the percolation threshold is around 0.07 approximately. It is intuitive that the below the percolation threshold the network will not be connected, hence we tested the connectivity test for the network for various probabilities to determine the threshold point.

Probability (p)	**Connected**
0.05	FALSE
0.06	FALSE
0.07	FALSE
0.08	TRUE
0.09	TRUE

From the above table we can conclude that the random network we created has the percolation threshold of 0.07 One of the common inference of percolation threshold is the rate of information being viral, i.e. everyone gets the access of the information. If the probability of drawing an edge between two arbitrary vertices is above the percolation threshold then everyone can access the information as the entire network is connected.

Exercise #2

Flat Tailed Distribution

Using the power – law distribution, we generated random networks with 1000 nodes. In this model, we first created a random variable with probability mass function (following a power-law distribution) given by:

$$\mathbf{Pr(X = x) = x^{-3}}$$

Then, we sample 1000 points from the distribution with replacement. These 1000 points form the nodes of the network and their degree distribution follows the power law. Then, we create stubs using these nodes and randomly match up the stubs to generate a random network with 1000 nodes. The network generated is plotted below.

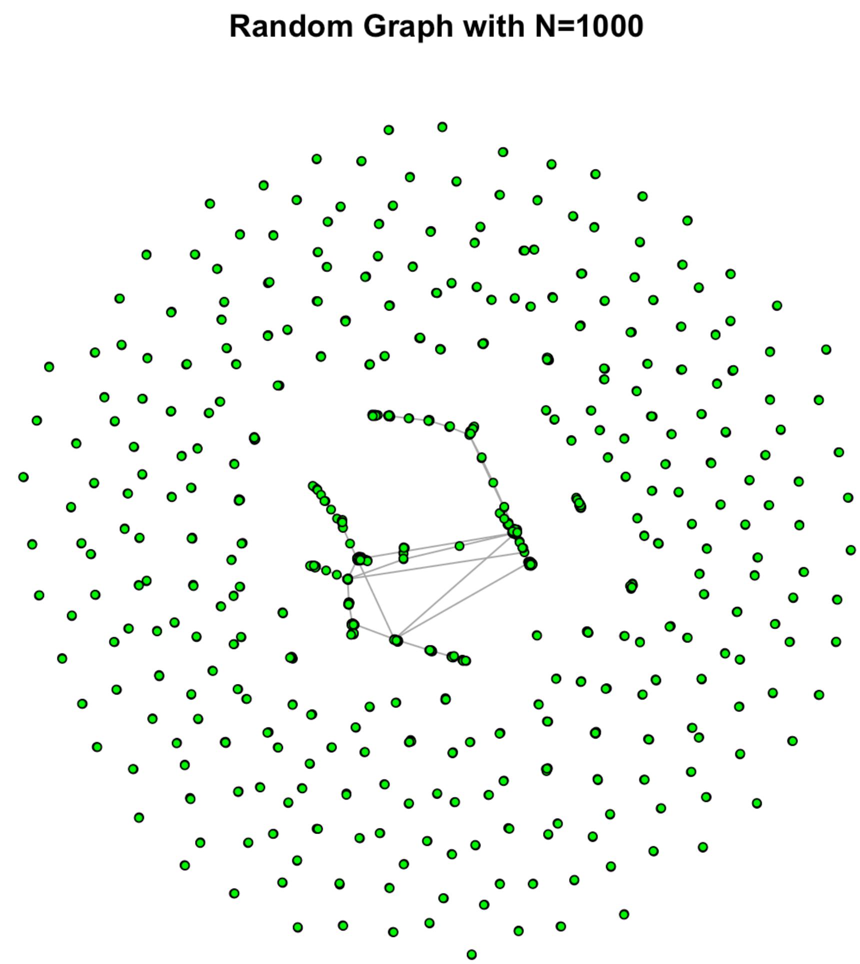


Figure 14: Fat-Tailed distribution network

From the plot, it is evident that the network is disconnected (as there are many isolated nodes), but there is a giant connected component (GCC) in the middle with some other connected components. The degree distribution of the network is plotted below.

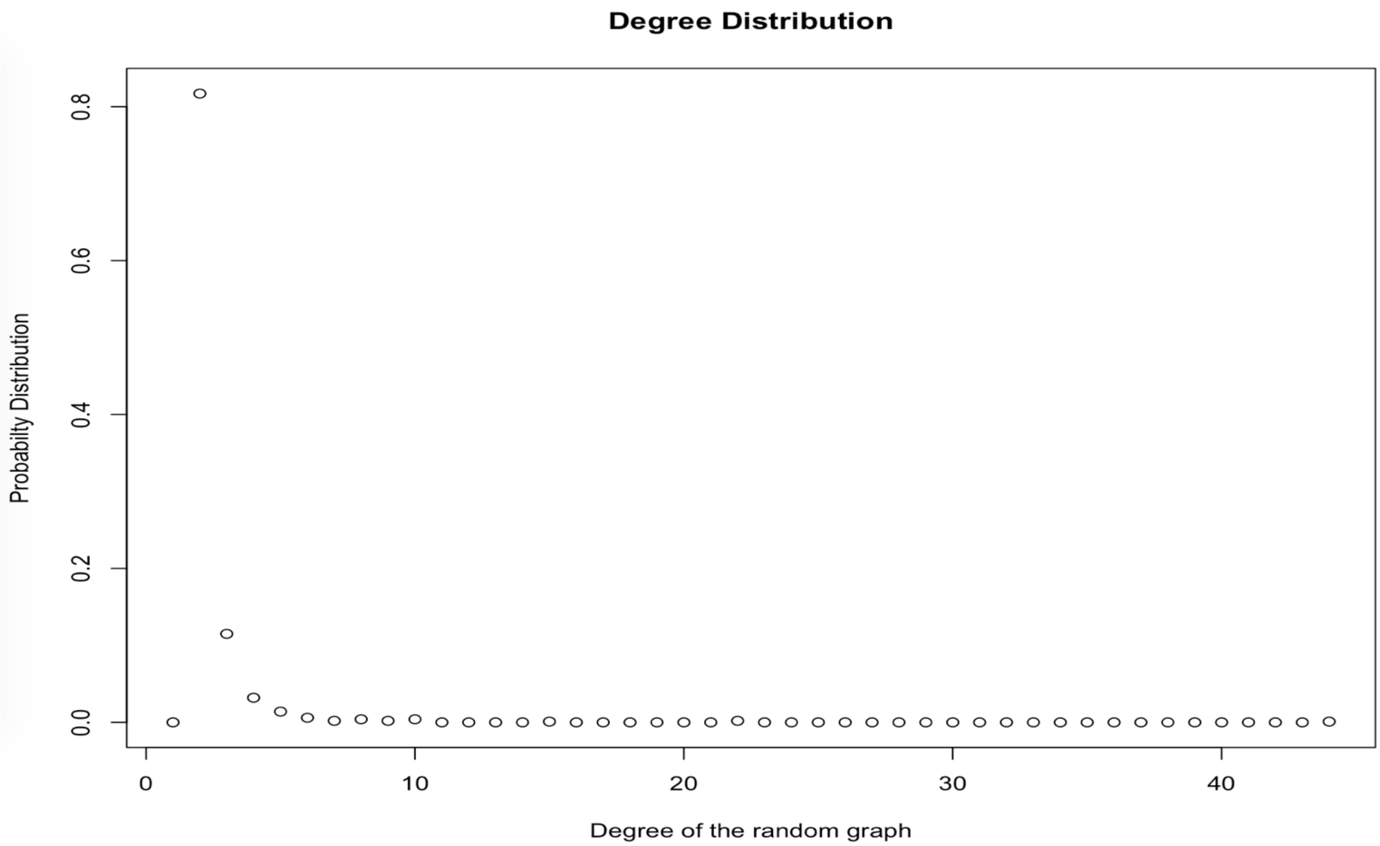


Figure 15: Degree distribution of Fat-Tailed distribution network.

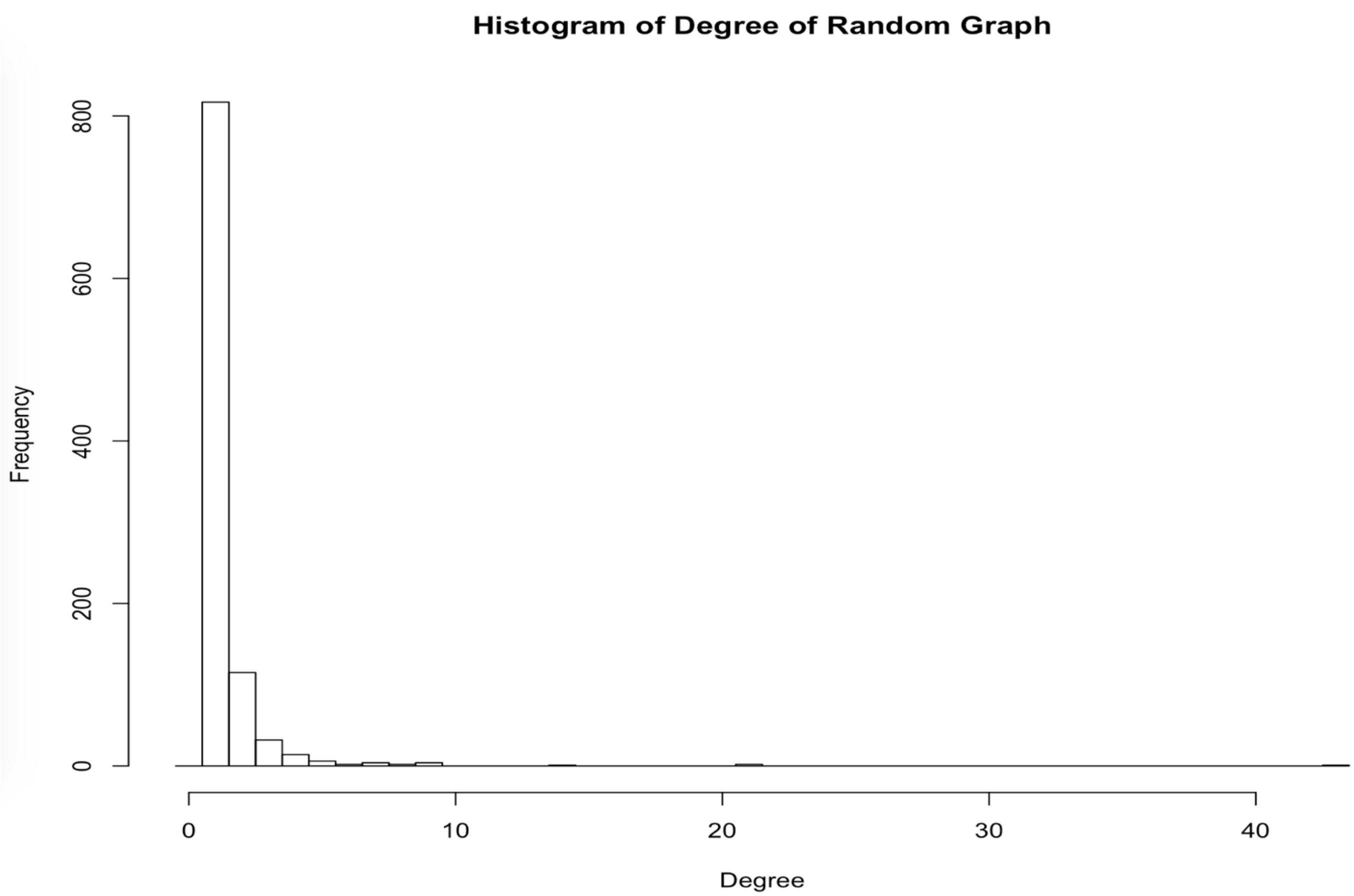


Figure 16: Histogram of degree distribution of Fat-Tailed distribution network.

The various network parameters are:

Parameters	Values
Connected	FALSE
Diameter	18
Path of diameter	1 -> 476 -> 546 ->519 ->925 -> 42-> 950-> 700 ->92 ->158 ->701 ->284 ->632-> 662-> 439-> 279-> 241-> 343
Farthest vertices of the diameter	26 ,343

Giant Connected Component

The Giant Connected Component is plotted below for N=1000. **It can be seen that the number of nodes in the Giant Connected Component is 273**.

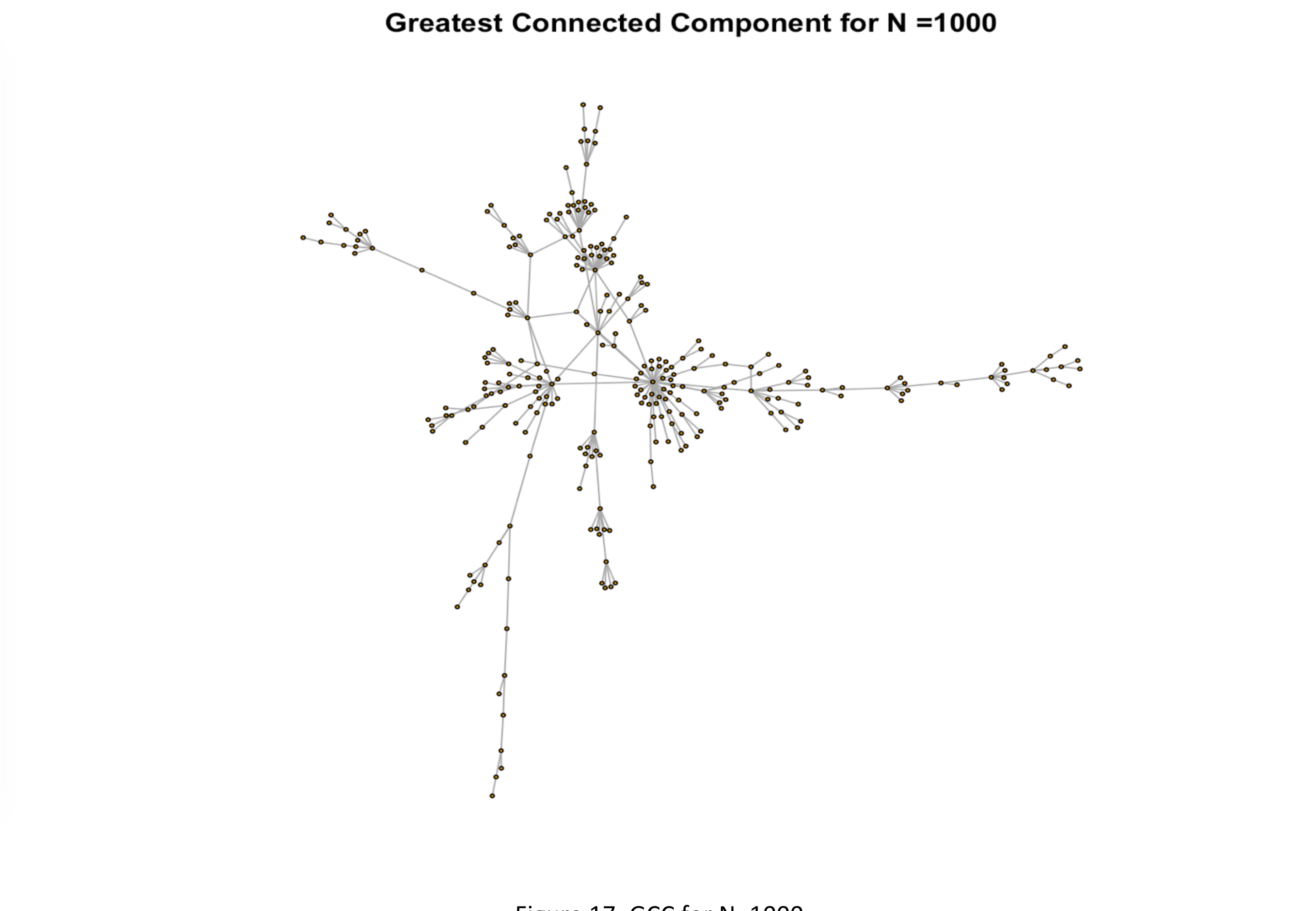


Figure 17: GCC for N=1000

Community Structure

Community detection forms an integral part of network analysis. Community detection tries to find the dense clusters, also called communities, by optimizing an objective score. In this homework, fast greedy method is used for detection community structure. This function finds the communities by optimizing a modularity score.

The modularity of a graph with respect to some division (or vertex types) measures how good the division is, or how separated are the different vertex types from each other. It defined as

*Q=1/(2m) * sum((Aij-ki*kj/(2m)) delta(ci,cj),i,j)*

Where, *m* is the number of edges,

Aij is the element of the *A* adjacency matrix in row *i* and column *j*,
ki is the degree of *i*,
kj is the degree of *j*,
ci is the type (or component) of *i*, *cj* that of *j*,
the sum goes over all *i* and *j* pairs of vertices, and *delta(x,y)* is 1 if *x=y* and 0 otherwise.

The modularity score of the partitioning using fast greedy method is 0. 9560871, calculated as follows :

```
> graph_a_community <- cluster_fast_greedy(graph_a)
> modularity(graph_a_community)
[1] 0.9560871
```

Parameters	Values
Modularity	0. 9560871

Since the network can be divided into well defined clusters, it results in a very good partitioning thereby, giving a large modularity. Partitioning into well defined clusters is due to the fact that there are fewer number of nodes, so the degree of randomness in the network is low.

The random network generated with 10000 nodes is plotted below.

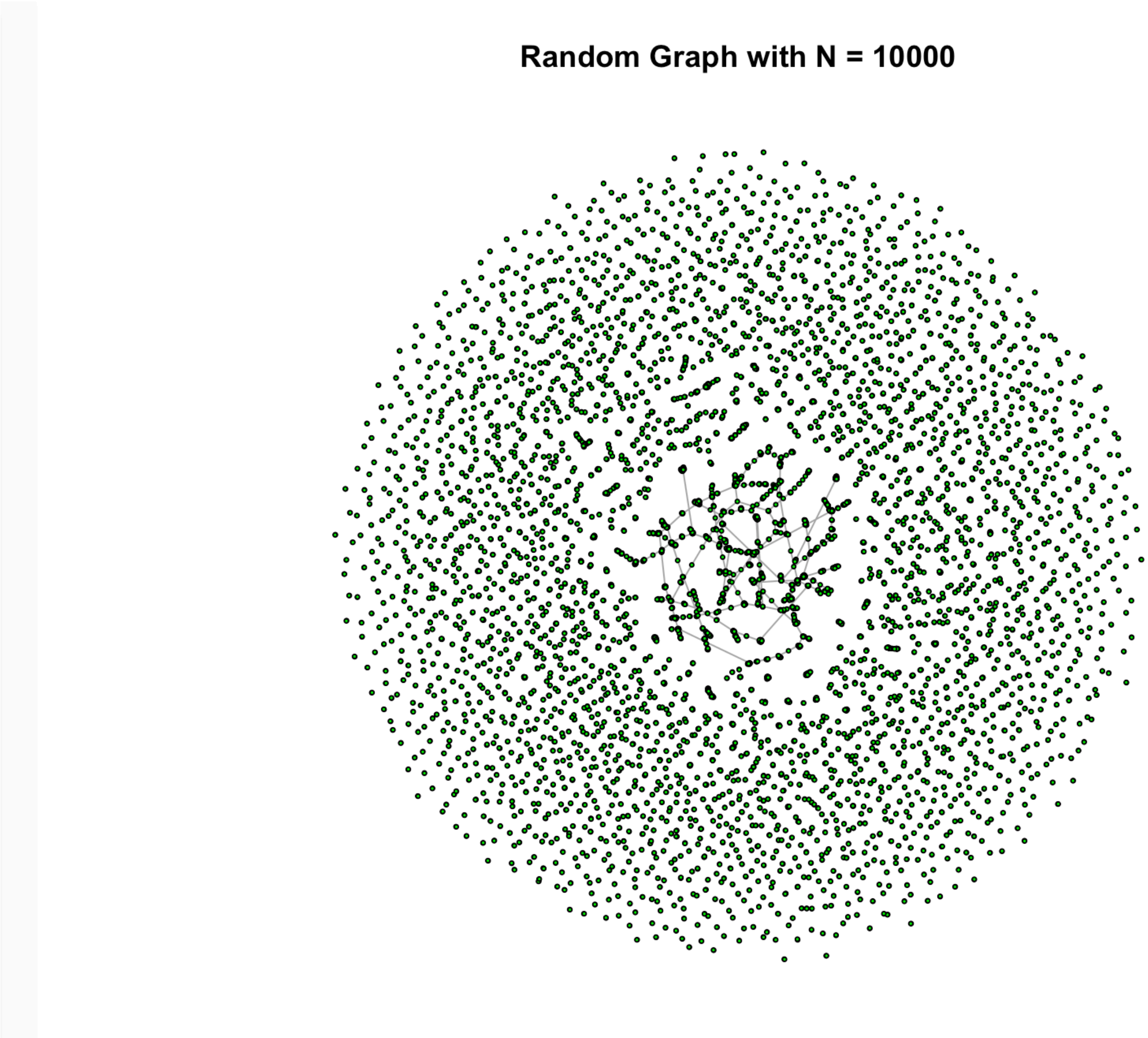
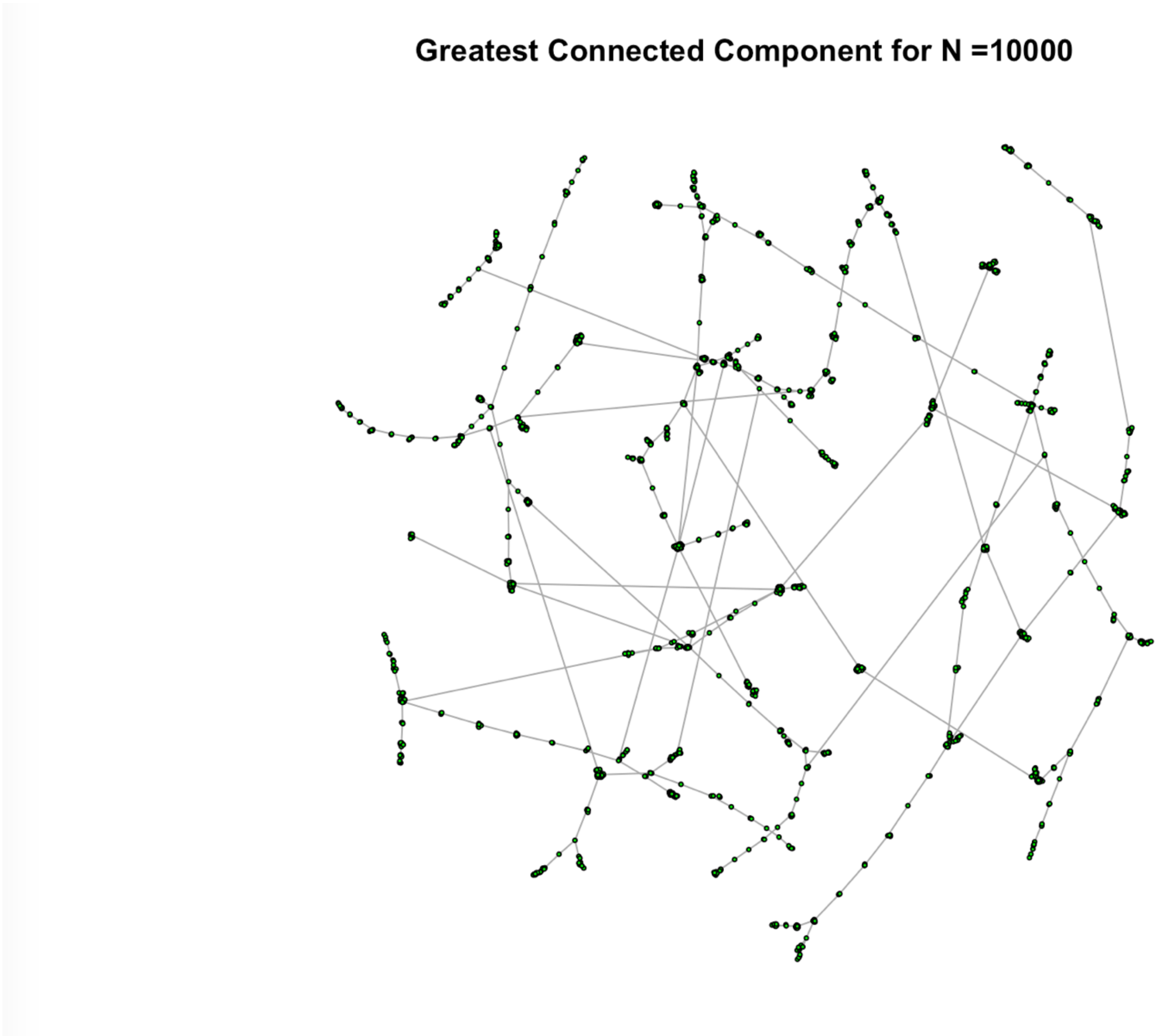


Figure 18: Fat-Tailed distribution for N=10000.

The Greatest Connected Component for N=10000 is plotted below.



The number of nodes in Greatest Connected Component are 1492.

The degree distribution for the above random graph is plotted below.

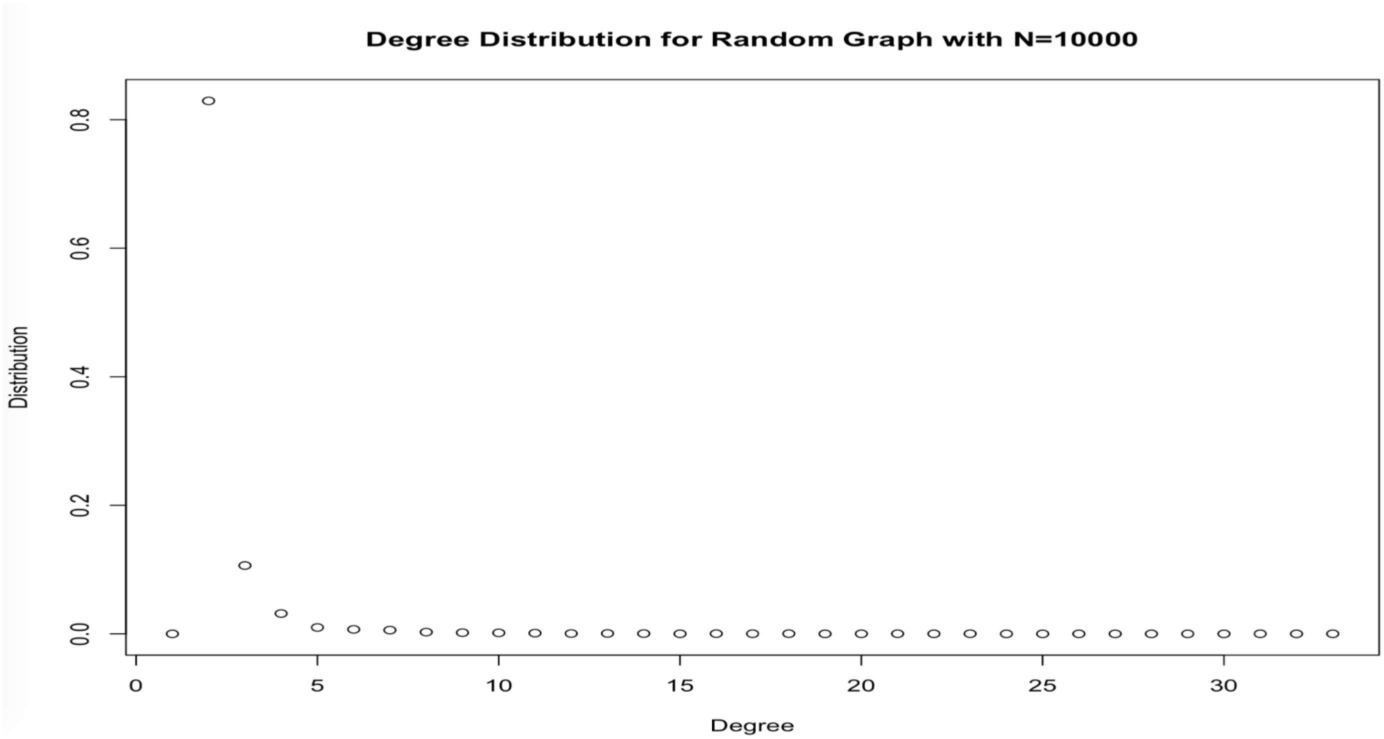


Figure 19: Degree distribution of Fat-Tailed distribution network for N=10000.

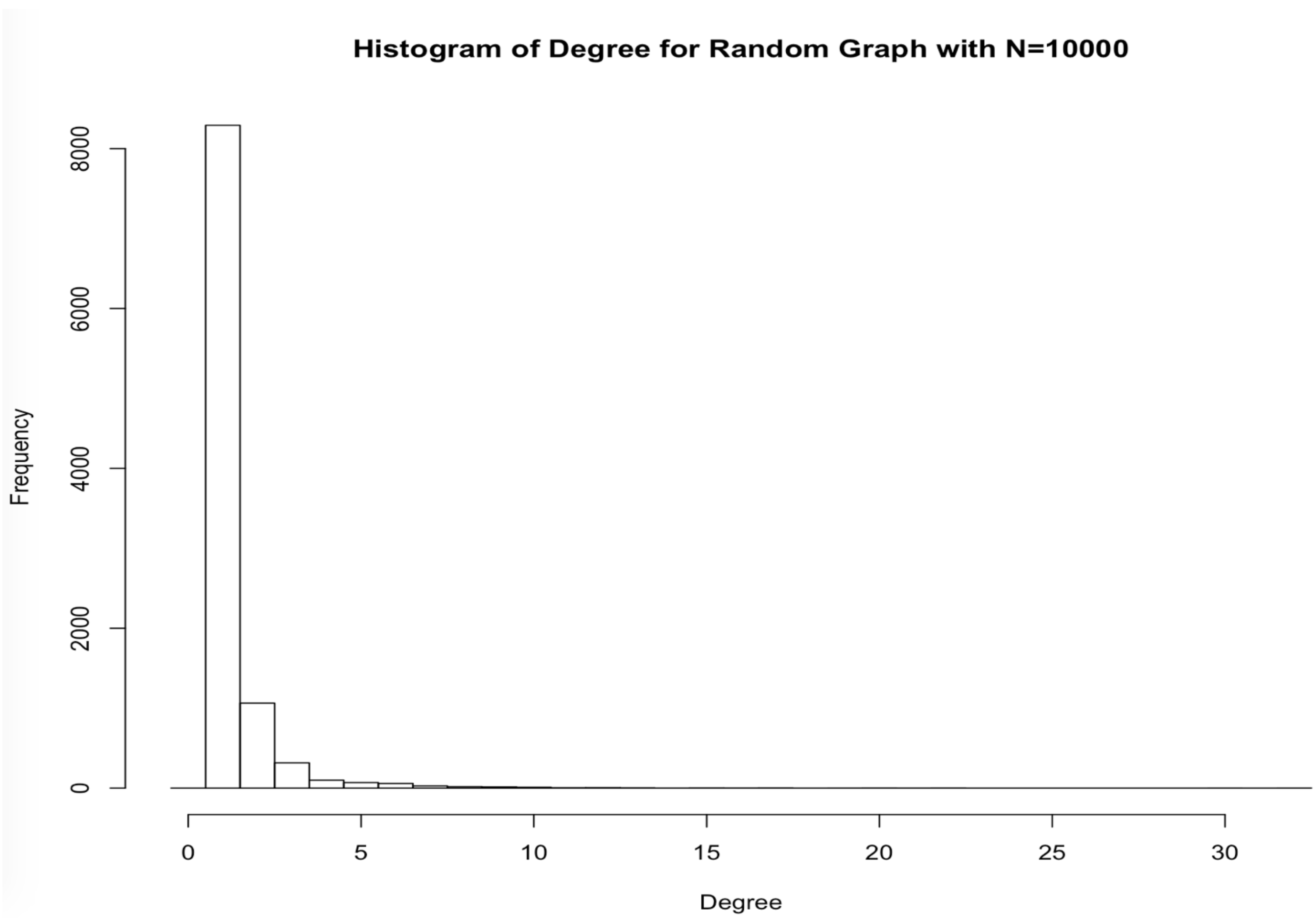


Figure 20: Histogram of Degree distribution of Fat-Tailed distribution network for N=10000.

As seen from the above network plot, the graph with N=10000 is not connected and we can check using following code:

The various network parameters are:

Parameters	Values
Connected	FALSE
Diameter	39
Modularity	0.9925116

No, the modularity of the 2 random graphs that is, with N=1000 and N=10000 is not the same. Modularity of random graph with N=10000 is greater than modularity of random graph with N=1000. This is in compliance with the theoretical concept.

Part d) A node was randomly selected from the network and then a random neighbor of the node was selected. The degree of that neighbor was then stored. If the first node selected did not have a neighbor, then a zero was stored. A point to note is that since the network was created from a power law degree distribution, a majority of the nodes will not have neighbors at all. So the vector of degrees will contain mostly zeros.

The degree distribution of the random graph with N=10000 when a node and its neighbor are picked randomly is plotted below.

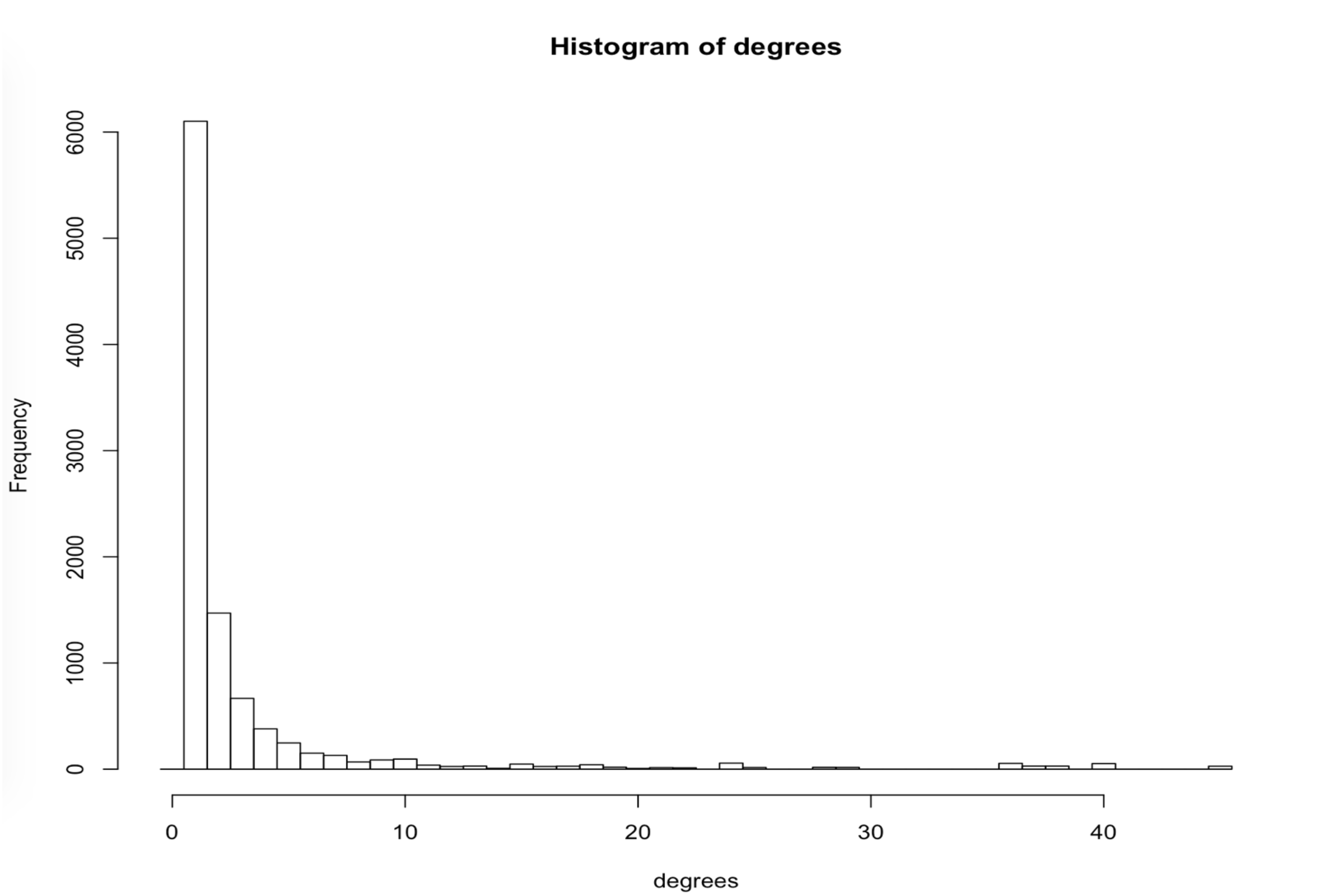


Figure 21

It is evident from the above histogram, that the distribution is similar to that of the power law degree distribution of the graph. The slight differences are due to the fact that the random sampling occurs with replacement, so some nodes might be sampled twice. However, the trend in general shows that the power law is followed.

Functions used are given below:

Functions	Task
sample	takes a sample of the specified size from the elements of x using either with or without replacement
sample_degseq	It is often useful to create a graph with given vertex degrees.
is_connected	Calculate the maximal (weakly or strongly) connected components of a graph
decompose.graph	Creates a separate graph for each component of a graph.
vcount	Order (number of vertices) of a graph
cluster_fast_greedy	This function tries to find dense subgraph, also called communities in graphs via directly optimizing a modularity score.
modularity	This function calculates how modular is a given division of a graph into subgraphs.
diameter	The diameter of a graph is the length of the longest geodesic.

Exercise #3

Growing Random Networks

Each time a new vertex is added it creates a number of links to old vertices and the probability that an old vertex is cited depends on its in-degree (preferential attachment) and age. This method of connection based on degree and age of a node is called preferential attachment and aging model.

Preferential attachment and aging model

It is an algorithm for generating random scale-free networks using a preferential attachment mechanism [3]. Scale-free networks are widely observed in natural and human-made systems, including the Internet, the world wide web, citation networks, and some social networks.

The network begins with an initial connected network of m_0 nodes and the new nodes are added to the network one at a time. Each new node is connected to $m \leq m_0$ existing nodes with a probability that is proportional to the number of links that the existing nodes already have. Formally, the probability p_i that the new node is connected to node i is

$$p_i = \frac{k_i}{\sum_j k_j},$$

where $k_i$is the degree of node $i$and the sum is made over all pre-existing nodes j(i.e. the denominator results in twice the current number of edges in the network).

We used the igraph function sample_pa_age for generating random growing networks. The function creates a random graph by simulating its evolution. Each time a new vertex is added it creates a number of links to old vertices and the probability that an old vertex is cited depends on its in-degree (preferential attachment) and age. The appendix contains the detail of this igraph function [4].

Fixed preferential attachment exponent

We created and compared the degree distribution of growing random network with fixed pa.exp =1 and varying the aging.exp. Usually aging.exp are negative, hence we chose the following values [0,-1,-2] for aging.exp i.e. a decreasing sequence (according to sign).

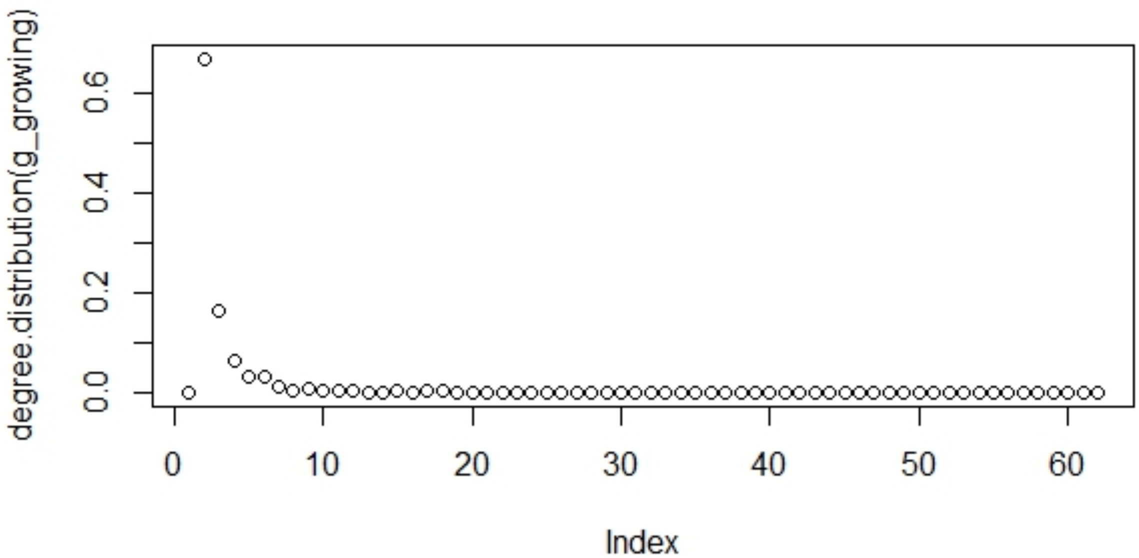


Figure 22: Degree distribution with aging exp.= 0.

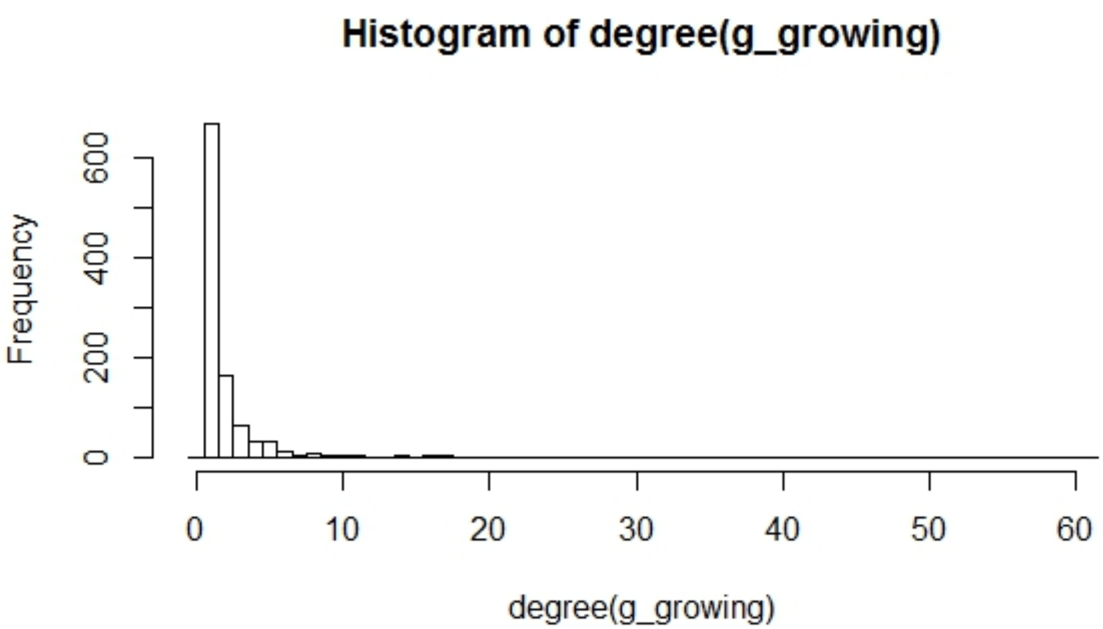


Figure 23: Histogram of degree distribution of aging exp. =0

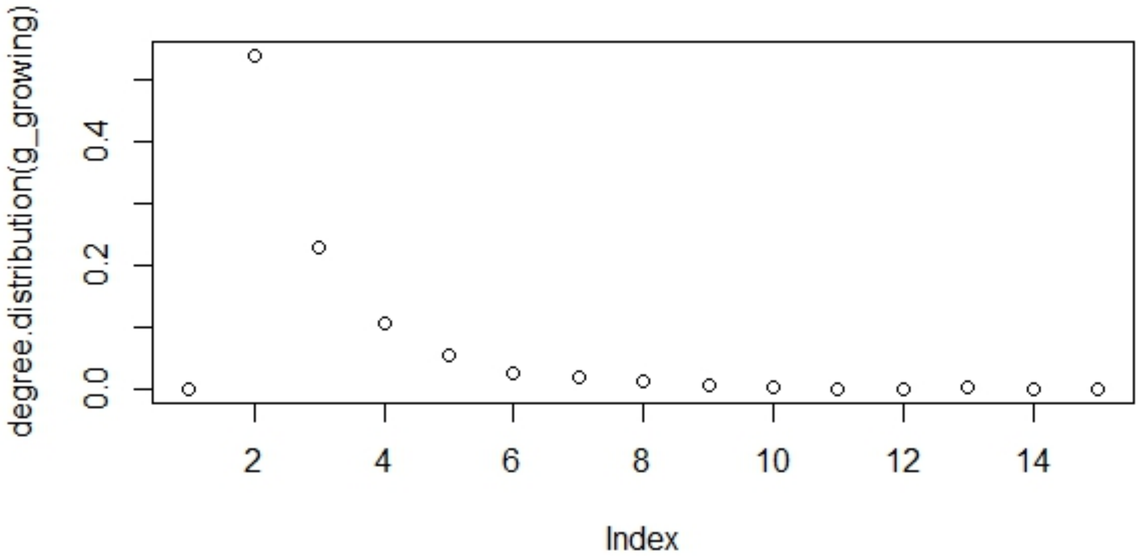


Figure 24: Degree distribution with aging exp.= -1.

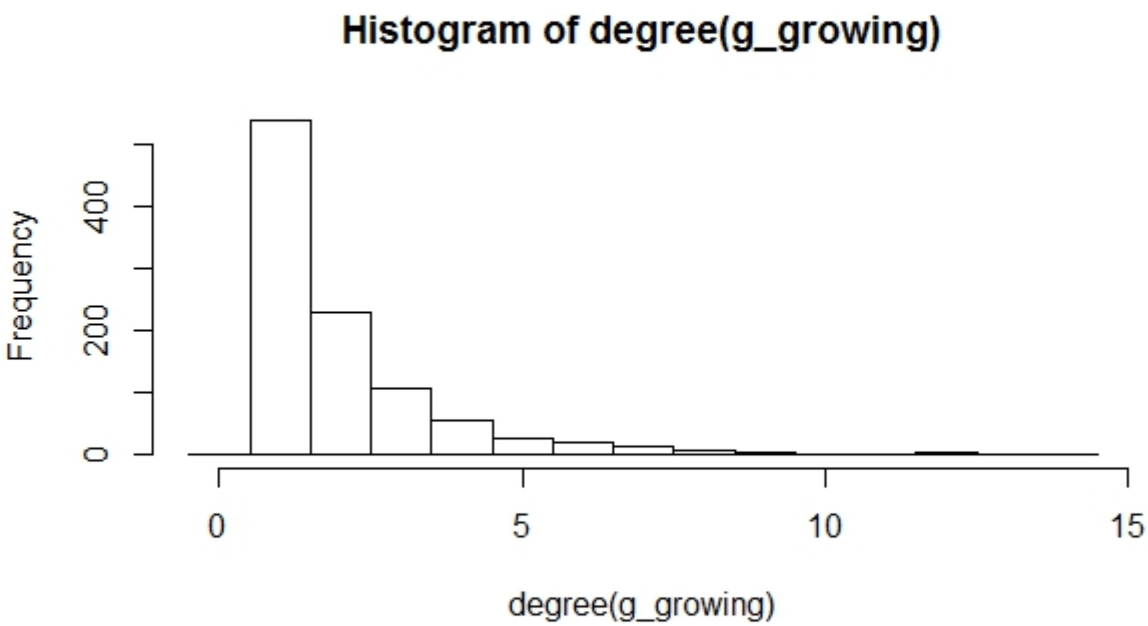


Figure 25: Histogram of degree distribution of aging exp. = -1

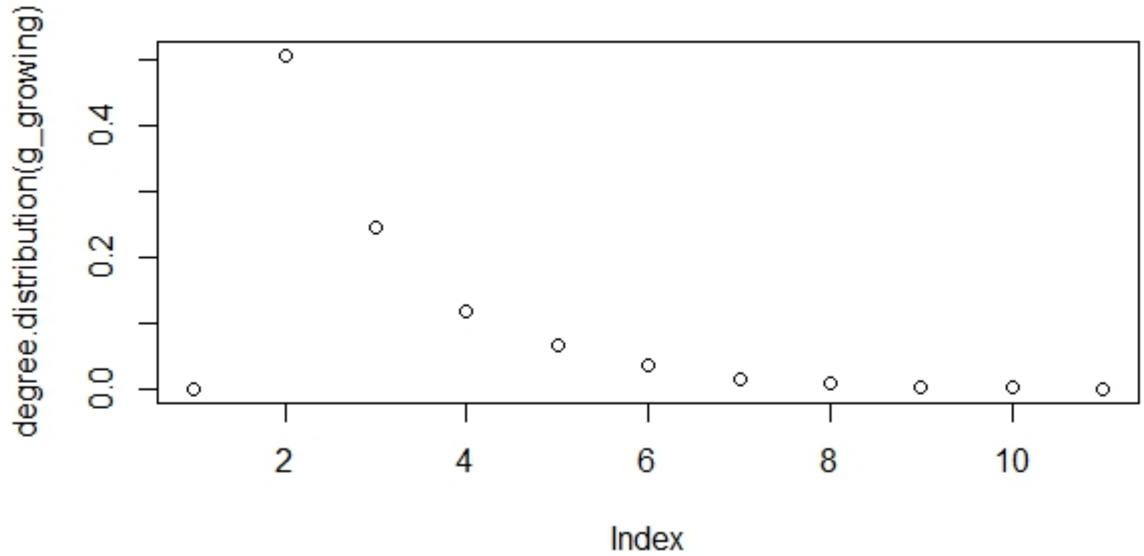


Figure 26: Degree distribution with aging exp.= -2

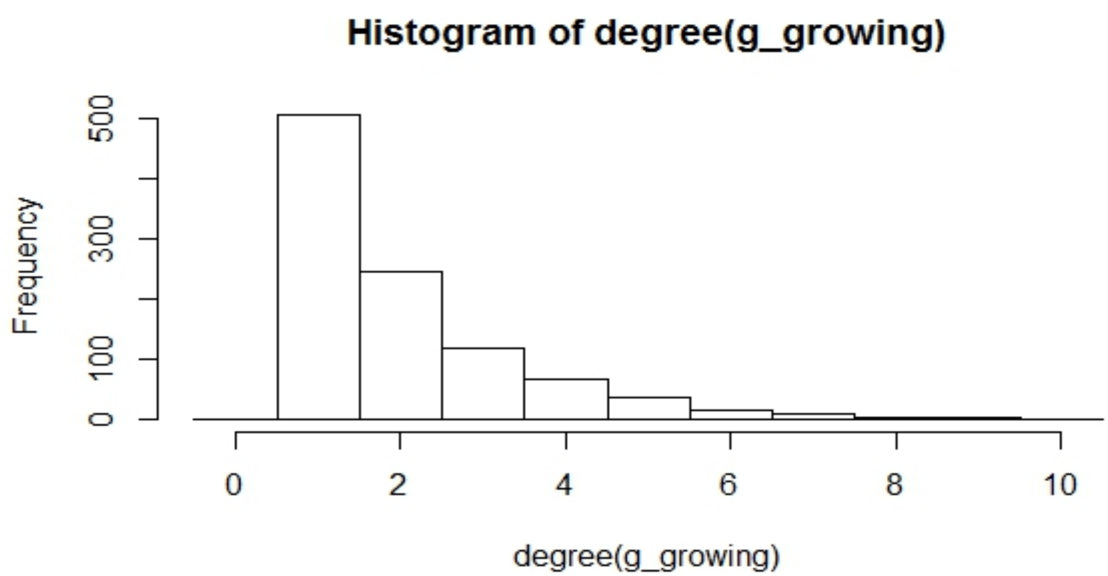


Figure 27: Histogram of degree distribution of aging exp. = -2

The observations from the following network parameters are:

Aging parameter(pa.exp = 1)	Maximum Degree of the Node
0	81
-1	13
-2	13
-3	9

We can infer from the above degree distribution plots that the maximum degree decreases for decreasing aging exp. The pattern follows similar to the power law distribution. We also performed the same operations for aging.exp of -3, to confirm the decreasing pattern.

Fixed aging exponent

We created and compared the degree distribution of growing random network with fixed aging.exp = -3 and varying the pa.exp. We chose the following values [1,2,3].

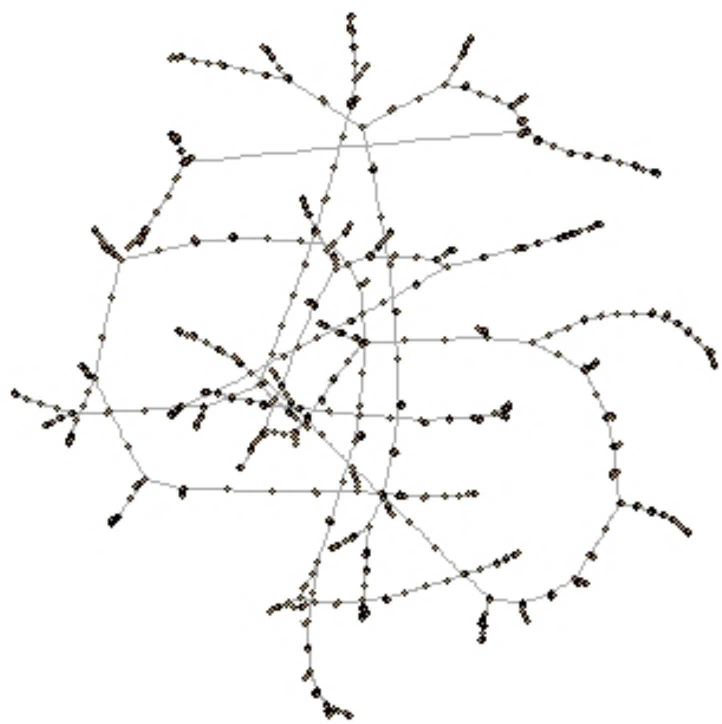


Figure 28: Network for pa.exp = 1.

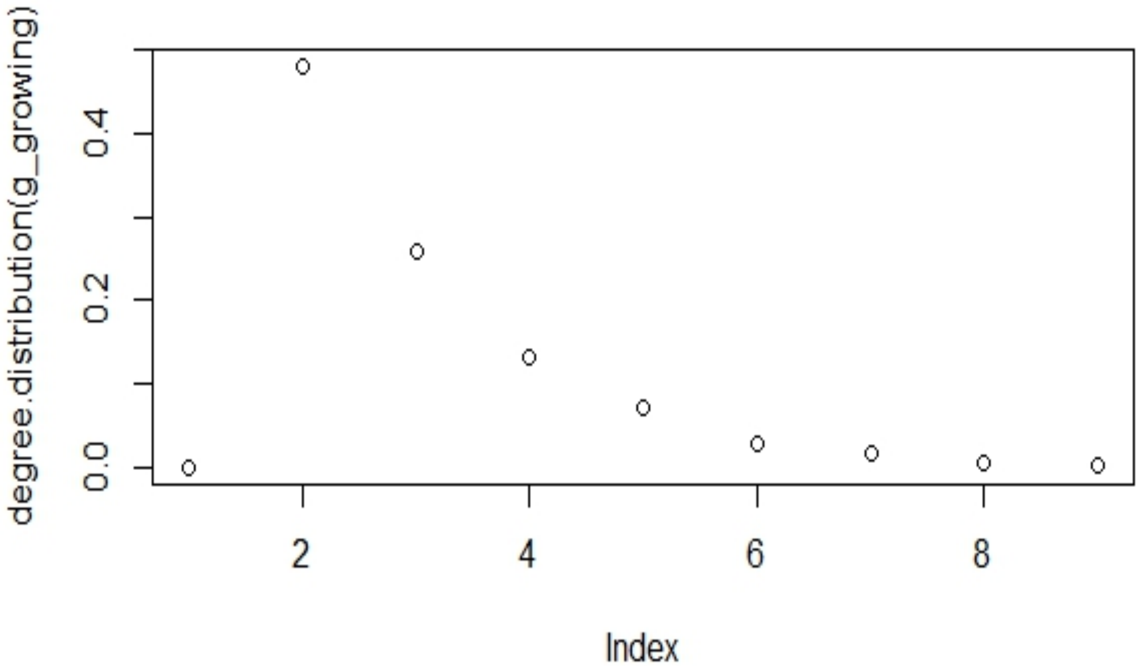


Figure 29: Degree distribution with pa.exp.= 1.

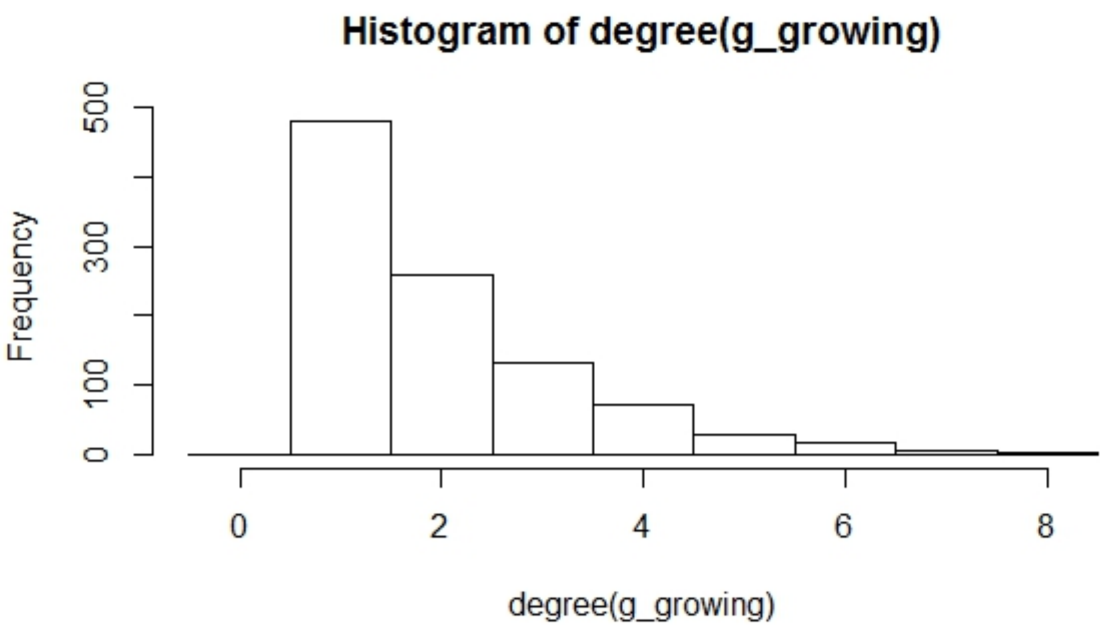


Figure 30: Histogram of degree distribution of pa.exp. = 1

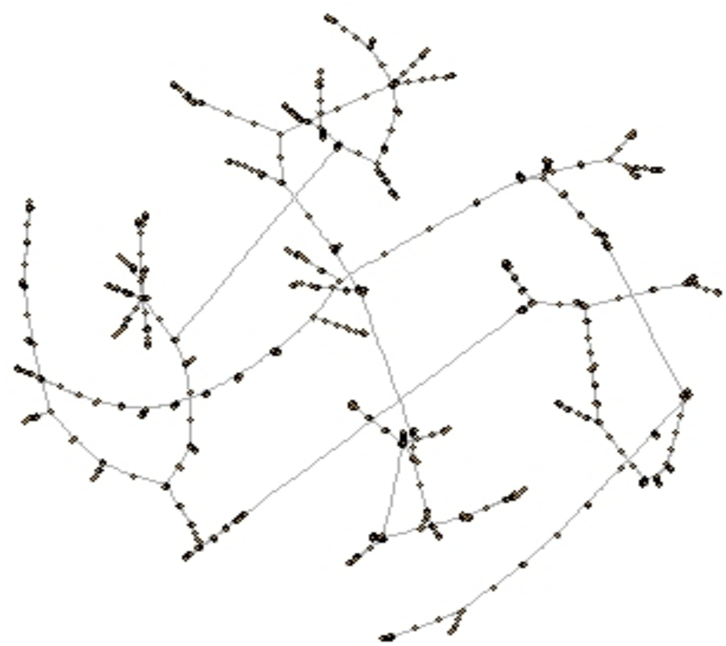


Figure 31: Network for pa.exp = 2.

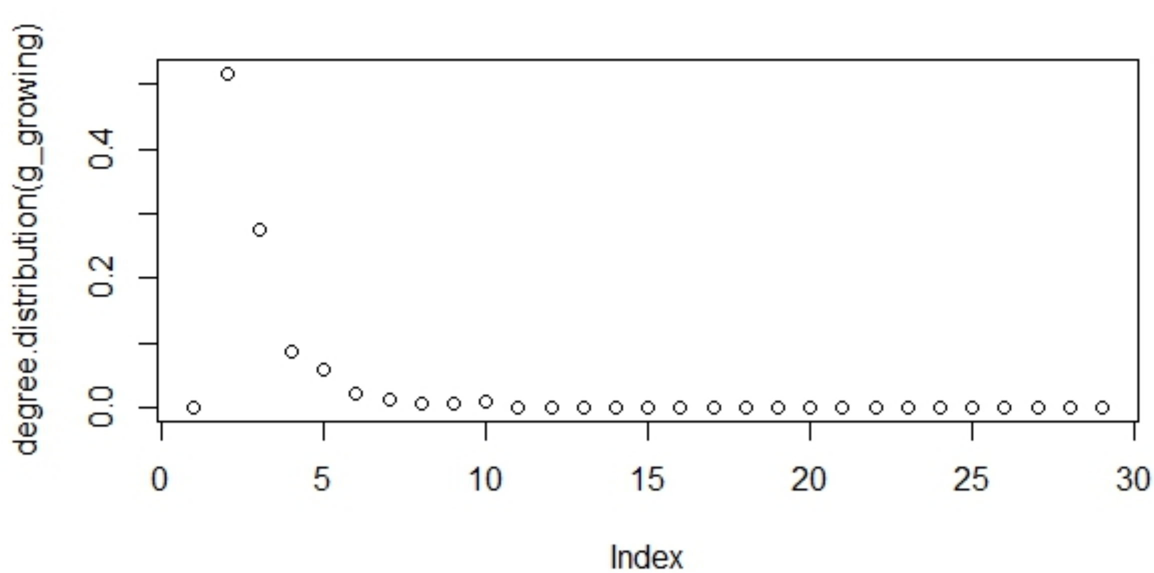


Figure 32: Degree distribution with aging pa.exp.= 2.

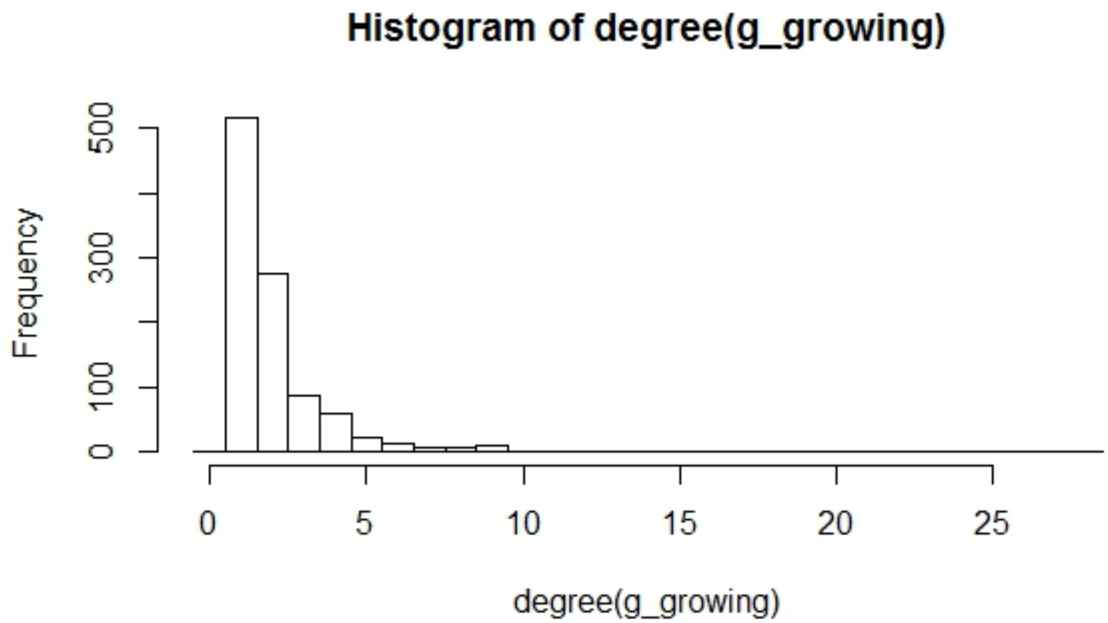


Figure 33: Histogram of degree distribution of pa.exp. = 2

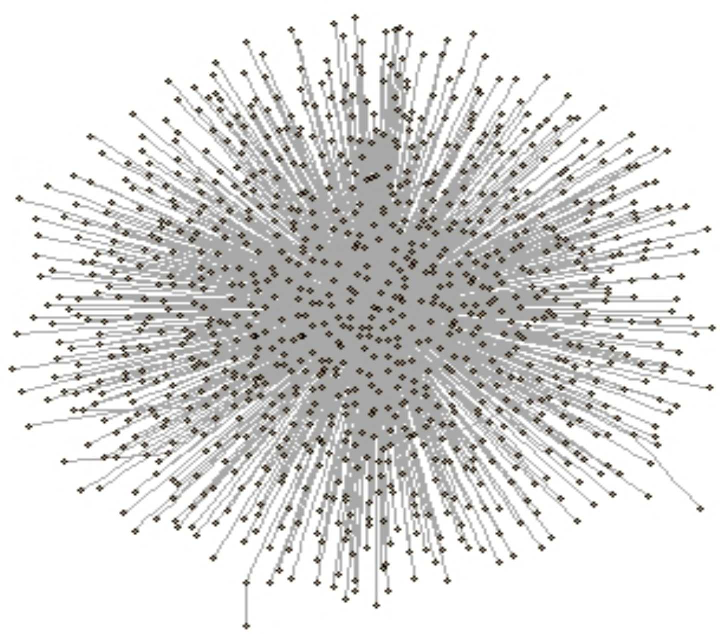


Figure 34: Network for pa.exp=3

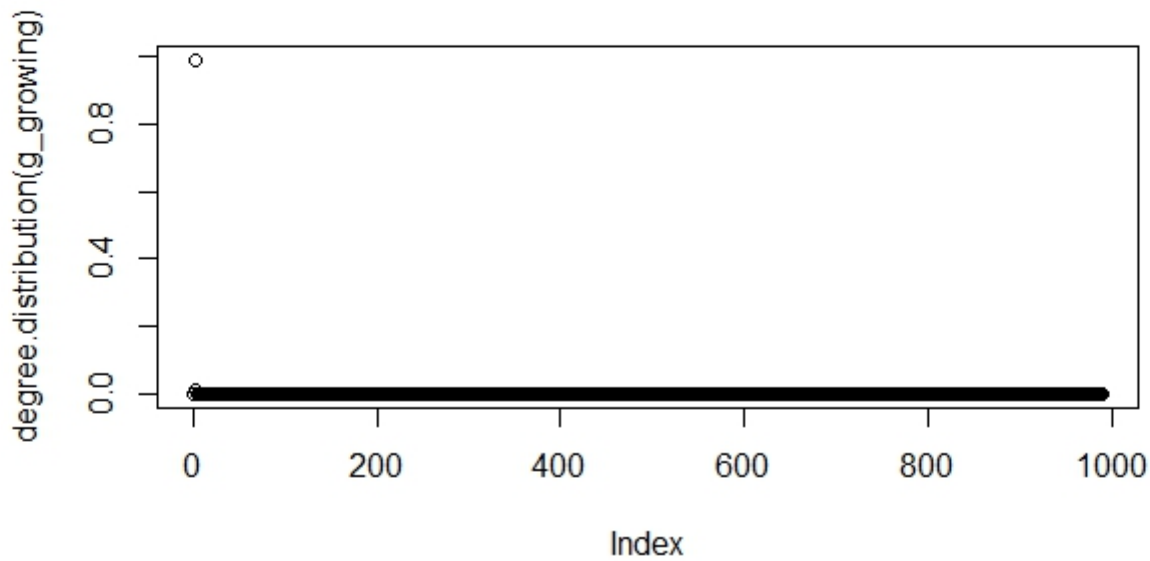


Figure 35: Degree distribution with pa.exp.= 3

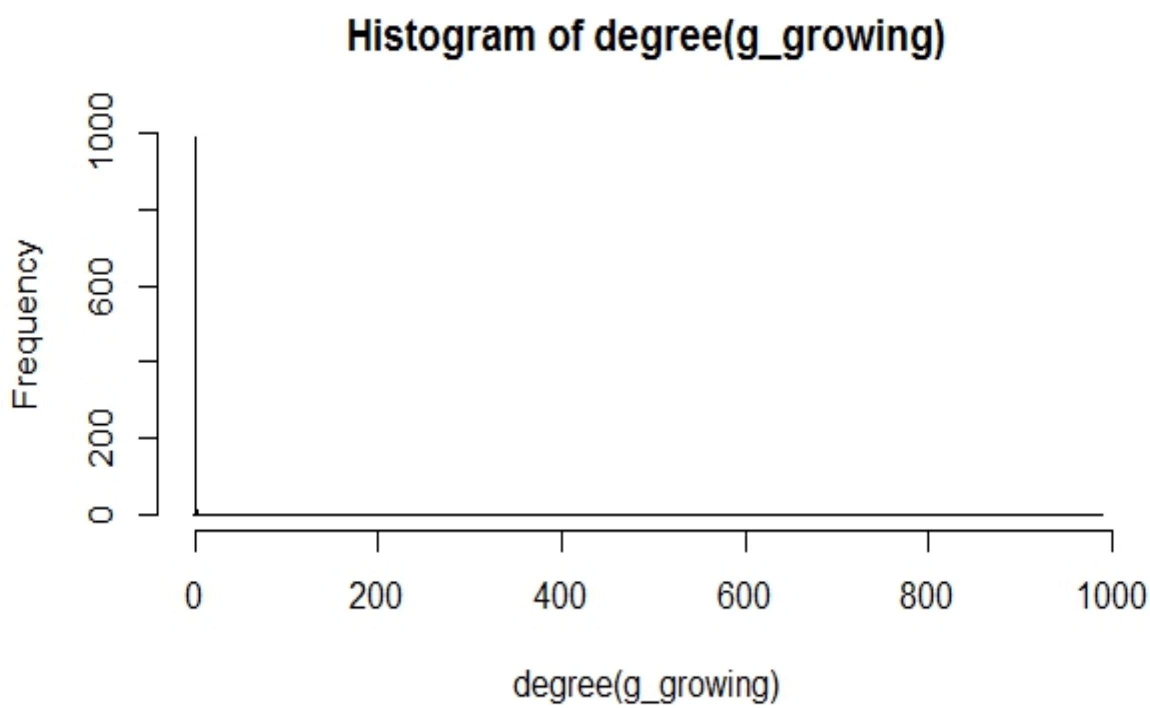


Figure 36: Histogram of degree distribution of pa.exp.= 3

When aging exponent is fixed we observed a drastic change in the structure of the network. For any value of pa.exp above or equal to 3 i.e. greater or equal to mod of aging.exp |aging.exp|, the nodes of the network seems to be connected to one node instead of having even distribution among the network. The structure more similar to star topology network, where all the nodes/elements are connected to a common node/elements called hub [4]. This transformation matches with the preferential attachment model, since the nodes with a higher degree gets preference and increasing the power (pa.exp) transforms the network into a star network. Speaking only in terms of the structure of the network.

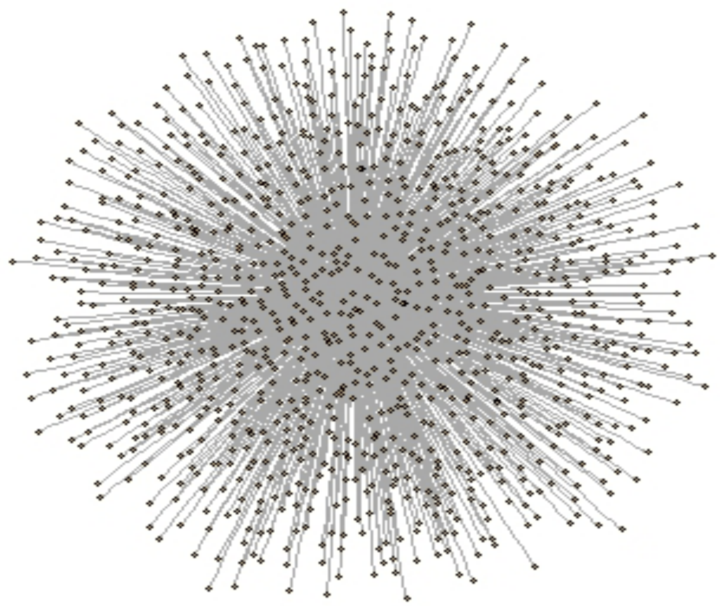


Figure 37: Network for pa.exp = 4. Hence forth for any value of pa.exp the network has star topology.

The observations from the following network parameters are:

Pa.exp (aging.exp = -3)	Maximum Node of the degree
1	7
2	18
3	988 -> connected to node 1
4	999 -> connected to node 2

We can also observe from the above degree distribution plots that the maximum degree increases for decreasing aging exp.

Community structure

We used the fast greedy method to find the communities in the 1000 node network. The modularity score of the partitioning for various values of pa.exp and aging.exp are tabulated below

pa.exp	aging.exp	Modularity
1	0	0.9240106
1	-1	0.9342571
1	-2	0.9353698
1	-3	0.9354394
1	-3	0.9359605
2	-3	0.9329695
3	-3	0.009942375
4	-3	2.950862e-14

From the above table

- We used fast greedy method to find community structures. From the above table we can also infer that the modularity increases for decreasing aging exp given the pa.exp is fixed.

- we infer that the modularity decreases for increasing pa.exp given the aging.exp is fixed.

- we can see that the modularity score starts to decrease drastically as soon as the network starts to become a star network (for pa.exp >=3). This is because the partitioning is poor when the networks transforms into a star network.

Exercise #4

Forest Fire Model (Directed Network)

The forest fire model intends to reproduce the following network characteristics, observed in real networks:

- Heavy-tailed in-degree distribution.
- Heavy-tailed out-degree distribution.
- Communities.
- Densification power-law. The network is densifying in time, according to a power-law rule.
- Shrinking diameter. The diameter of the network decreases in time.

The Forest Fire model was firstly proposed in [6] and [7]. The basic setting of the model is as follows. The model has two parameters, a *forward burning probability p* and a *backward burning ratio r*. Consider a node *v* joining the network at time t > 1, and let *G_{t}* be the graph constructed thus far. *G_{1}* will consist of just a single node. Node *v* forms outlinks to nodes in *G_{t}* according to the following process:

(a) *v* first chooses an ambassador node *w* uniformly at random and forms a link to *w*.

(b) Generate two random numbers, *x* and *y* that are geometrically distributed with means *{p} / {1-p}* and *{rp} / {1- rp},* respectively. Node *v* selects *x* outlinks and *y* in-links of w incident to nodes that were not yet visited. Let *w_{1}, w_{2},……w_{x+y}* denotes the other ends of these selected links. If not enough in or outlinks are available, *v* selects as many as it can.

(c) *v* forms outlinks to *w_{1}, w_{2},……w_{x+y}* and then applies step (b) recursively to each of *w_{1}, w_{2},……w_{x+y}* . As the process continues, nodes cannot be visited a second time, preventing the construction from cycling.

Thus, the burning of links in the Forest Fire model begins at *w*, spreads to *w_{1}, w_{2},……w_{x+y}* and proceeds recursively until it dies out. The original authors claim that this model can preserve the following three properties of a graph as growing:

- Heavy-tailed in-degree and out-degree distribution
- Densification power law
- Shrinking diameter

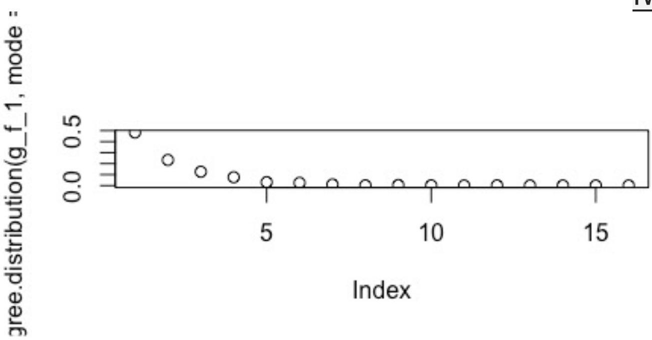
Most of the social network are based on this model. One of the interesting examples is the facebook friends suggestion. It can be noted that most of the suggestions have some basic similarity with the users and potential friend connection. This model is also used in any communicable disease breakout to determine the potential persons who might be exposed to virus or disease. Similarly it can be implemented on the research paper citation count or track and h-index can be calculated.

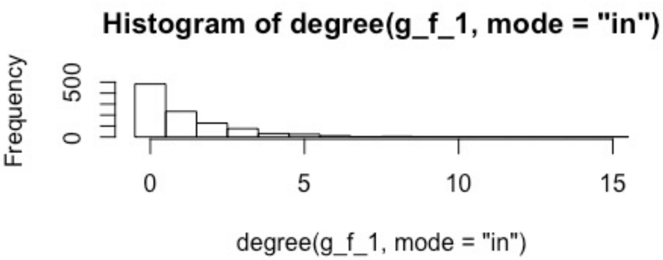
We created a 1000 node network (directed) using this forest fire model. To get a better understanding of the functionality of these parameters we varied the parameters fw.factor (forward burning probability) and bw.factor (the backward burning ratio).

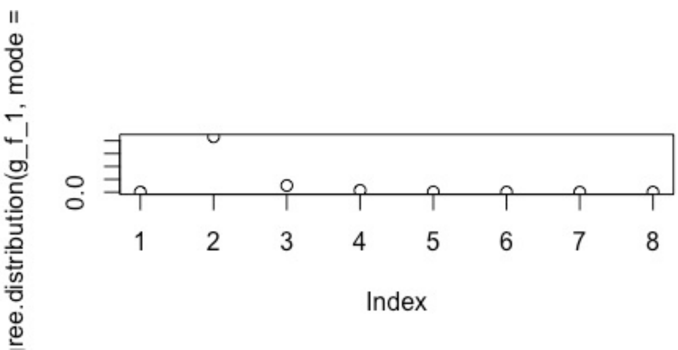
Varying fw.prob (keeping bw.factor = 1)

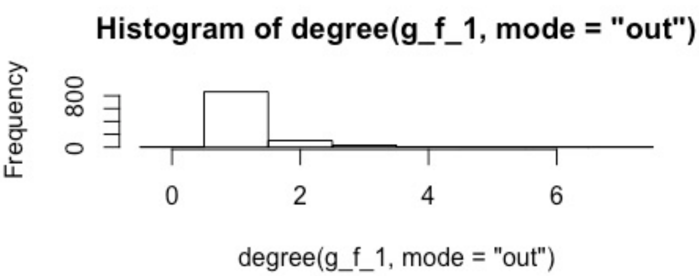
We used fw.prob = [0.1,0.2,0.3,0.4] to generate 4 directed random networks with 1000 nodes. The plots are as below:

fw.prob=0.1

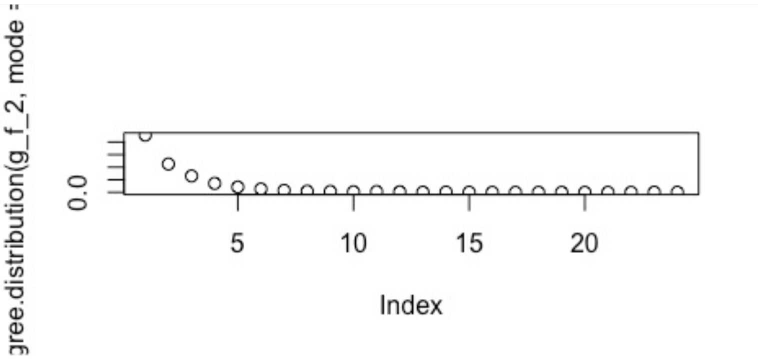


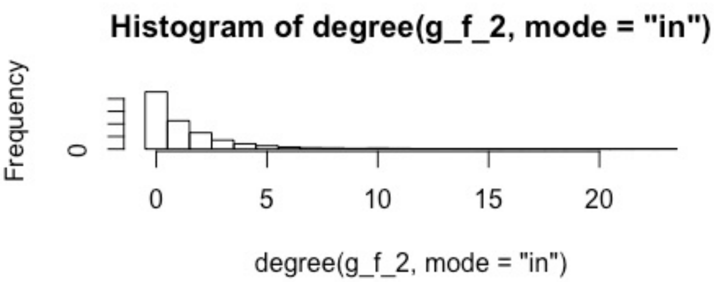


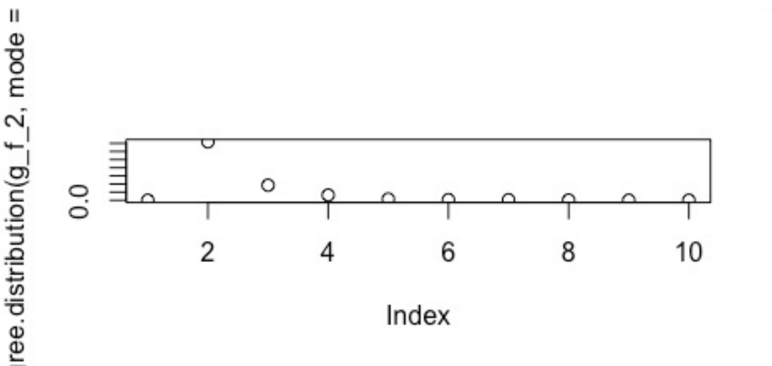


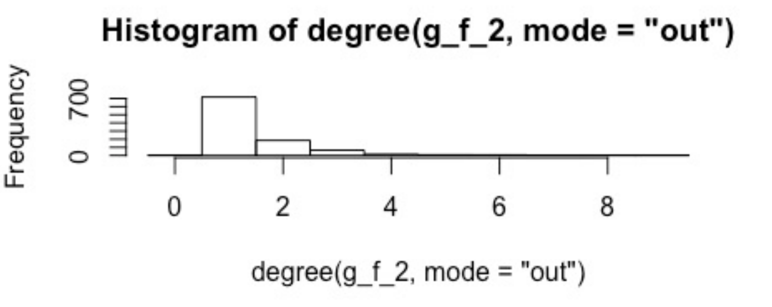


fw.prob=0.2

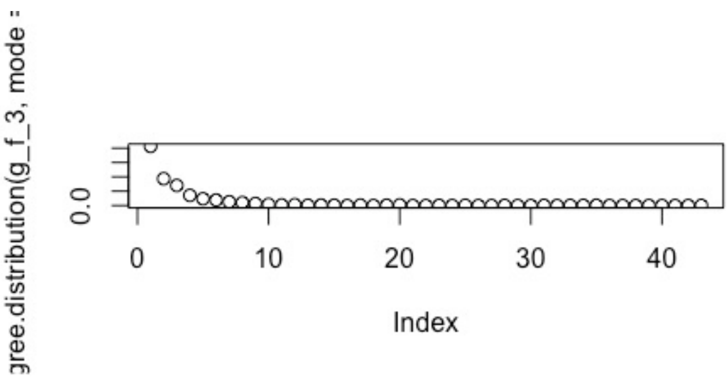

gree.distribution(g_f_2, mode =
0.0
5
10
15
20
Index

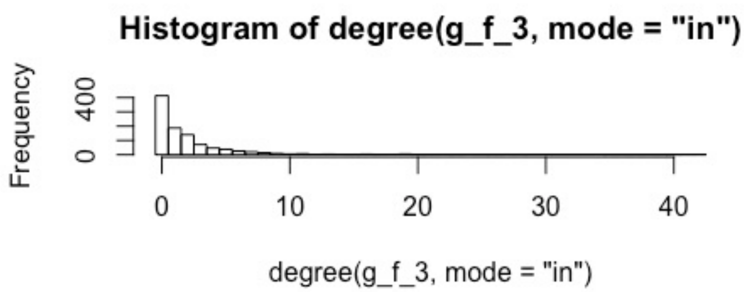

Histogram of degree(g_f_2, mode = "in")
Frequency
0
0
5
10
15
20
degree(g_f_2, mode = "in")

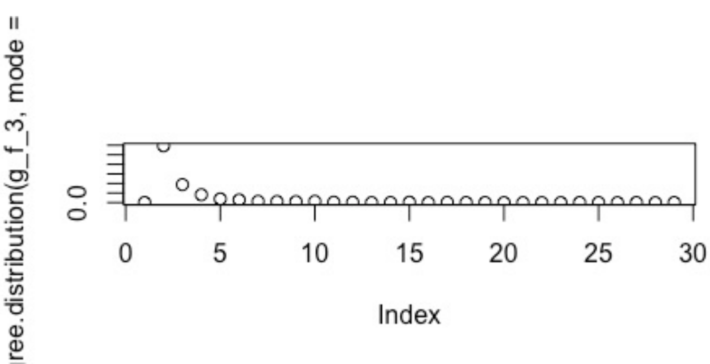

ree.distribution(g_f_2, mode =
0.0
2
4
6
8
10
Index

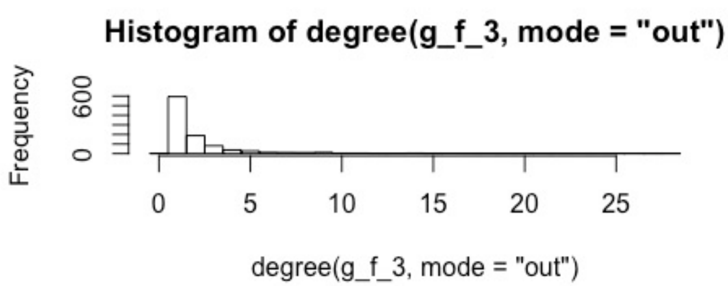

Histogram of degree(g_f_2, mode = "out")
Frequency
0
700
0
2
4
6
8
degree(g_f_2, mode = "out")

fw.prob=0.3

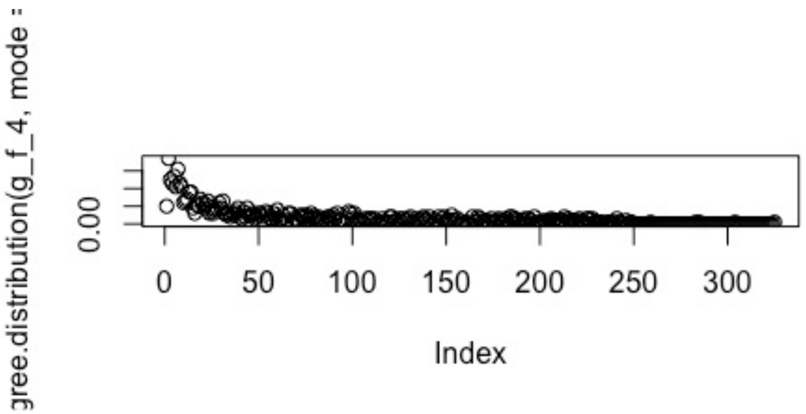

gree.distribution(g_f_3, mode =
0.0
0
10
20
30
40
Index

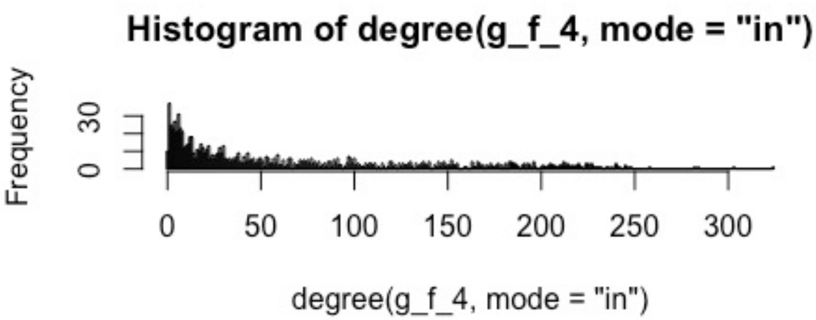

Histogram of degree(g_f_3, mode = "in")
Frequency
0
400
0
10
20
30
40
degree(g_f_3, mode = "in")

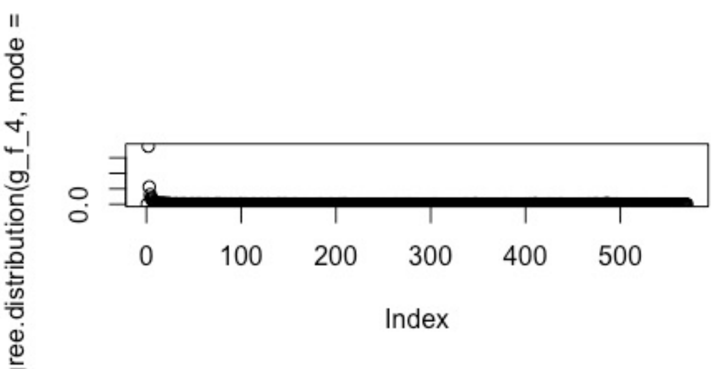

ree.distribution(g_f_3, mode =
0.0
0
5
10
15
20
25
30
Index

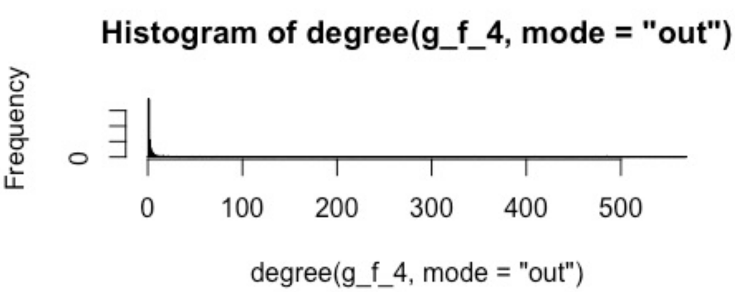

Histogram of degree(g_f_3, mode = "out")
Frequency
0
600
0
5
10
15
20
25
degree(g_f_3, mode = "out")

fw.prob=0.4









The above plots of degree distribution, we see that increasing the value of fw.prob, the maximum node degree (both in and out) increases and the distribution has a heavier tail.

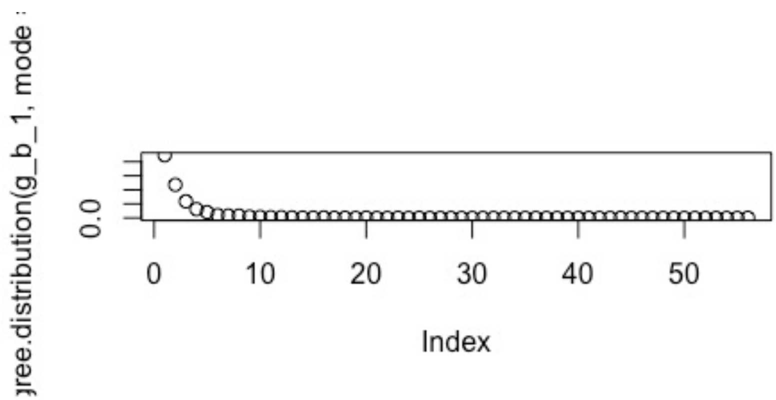
This result makes sense since as we increase the forward burning probability, the network grows faster and hence we have a heavier tailed distribution. Also, the diameter shrinks (tabulated below)

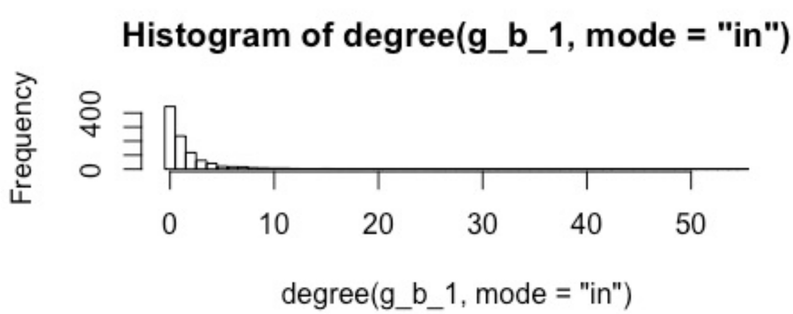
fw.prob	Diameter
0.1	15
0.2	14
0.3	12
0.4	9

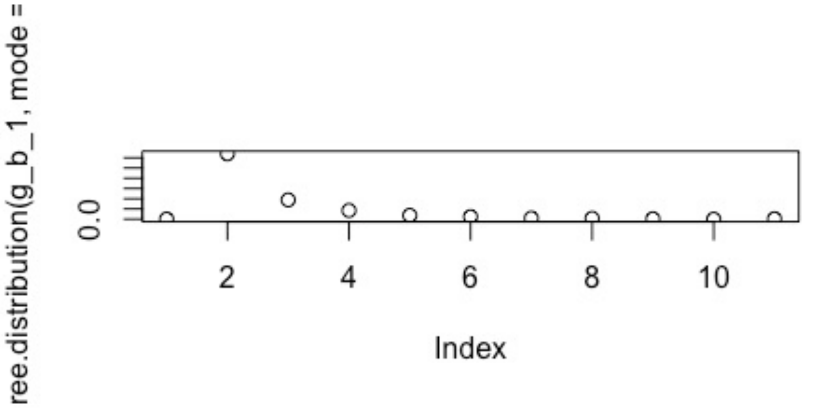
<u>Varying bw.factor (keeping fw.prob=0.3 as fixed):</u>

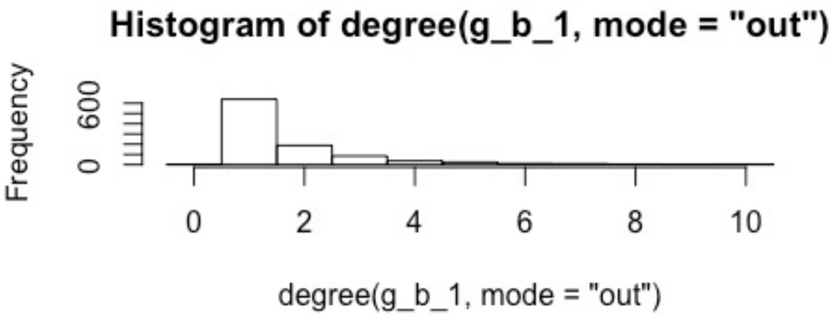
We used bw.factor = [0.6,0.8,1,1.2] to generate 4 directed random networks with 1000 nodes. The plots are as below:

<u>bw.factor= 0.6</u>

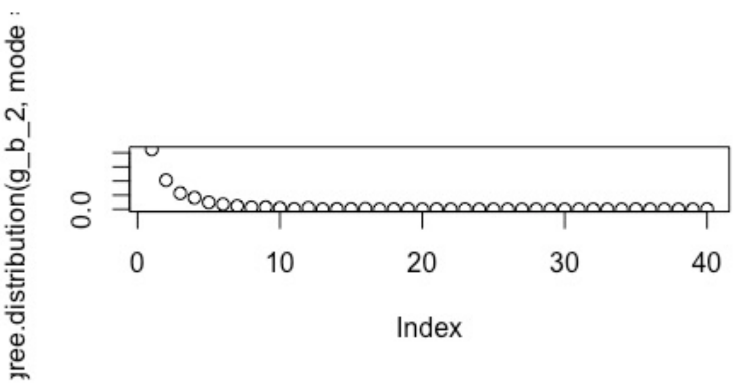


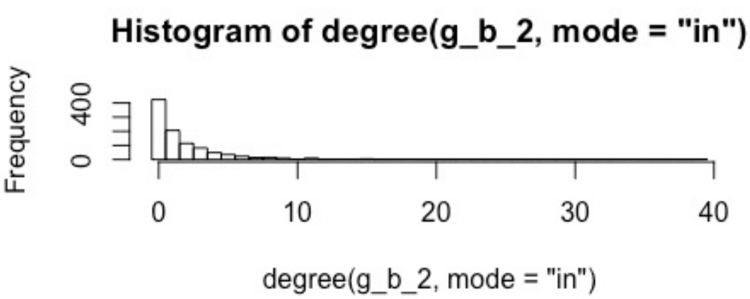


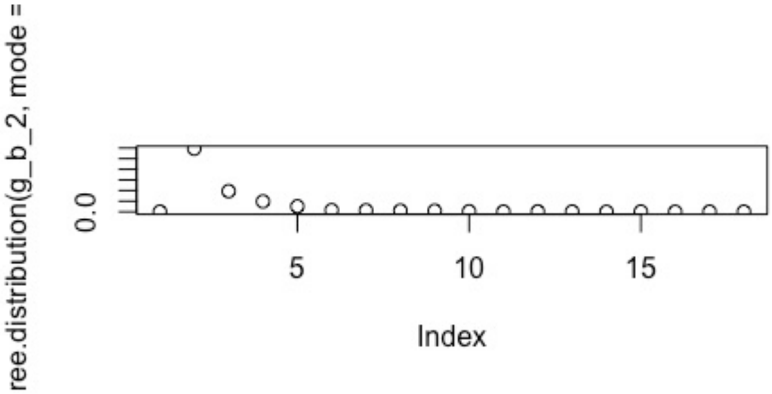

ree.distribution(g_b_1, mode =
0.0
2
4
6
8
10
Index

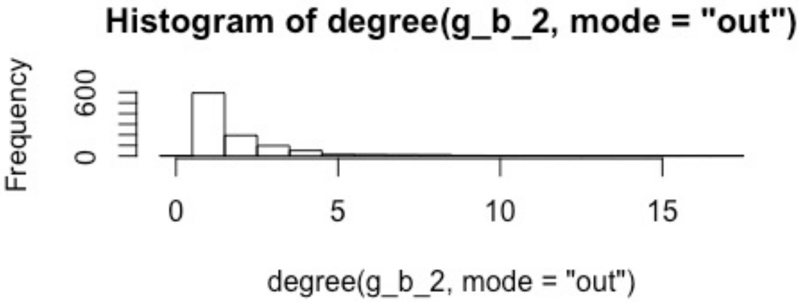

Histogram of degree(g_b_1, mode = "out")
Frequency
0
600
0
2
4
6
8
10
degree(g_b_1, mode = "out")

bw.factor= 0.8

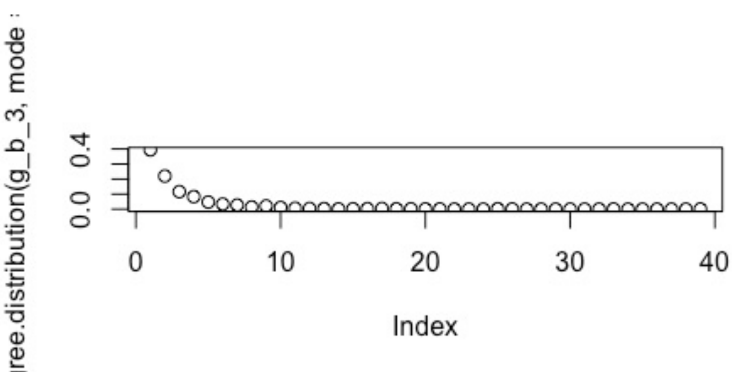

gree.distribution(g_b_2, mode
0.0
0
10
20
30
40
Index

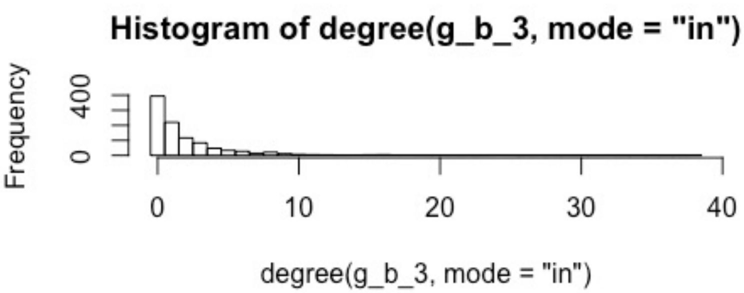

Histogram of degree(g_b_2, mode = "in")
Frequency
0
400
0
10
20
30
40
degree(g_b_2, mode = "in")

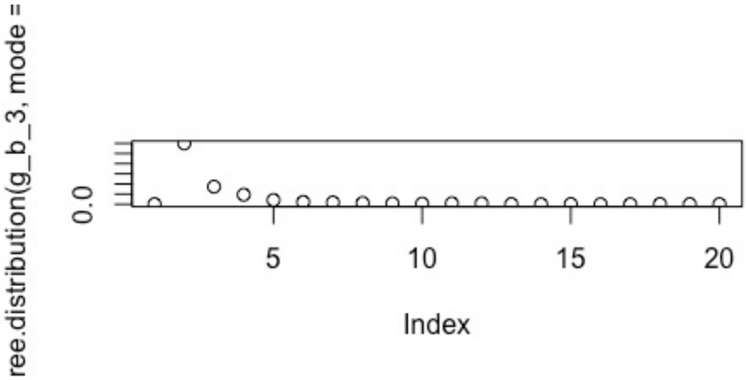

ree.distribution(g_b_2, mode =
0.0
5
10
15
Index


Histogram of degree(g_b_2, mode = "out")
Frequency
0
600
0
5
10
15
degree(g_b_2, mode = "out")

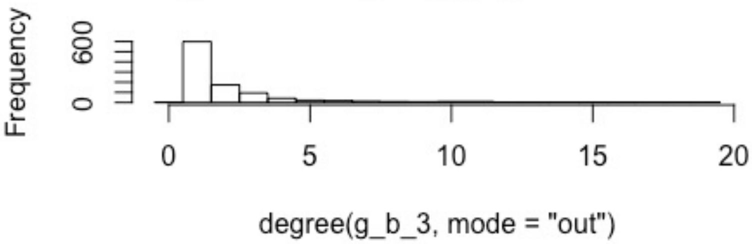
bw.factor= 1


jree.distribution(g_b_3, mode
0.0
0.4
0
10
20
30
40
Index

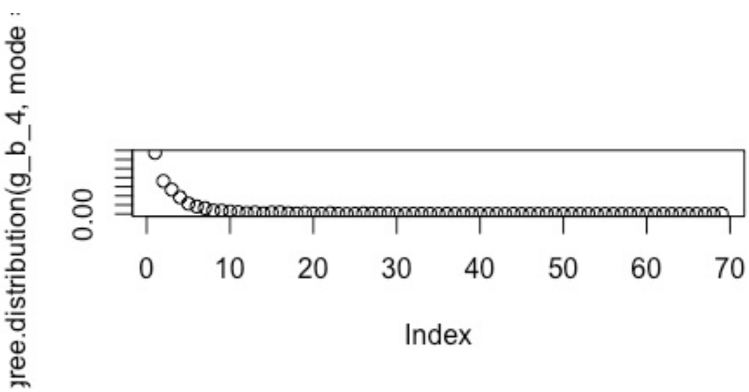

Histogram of degree(g_b_3, mode = "in")
Frequency
0
400
0
10
20
30
40
degree(g_b_3, mode = "in")



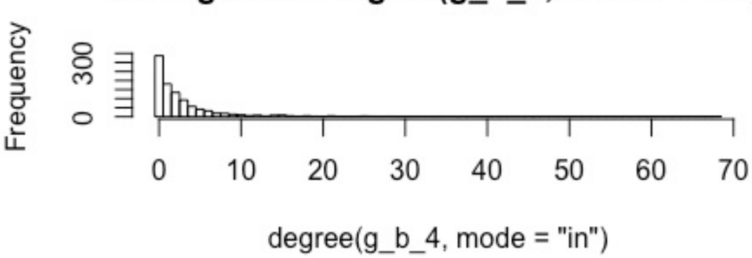
Histogram of degree(g_b_3, mode = "out")

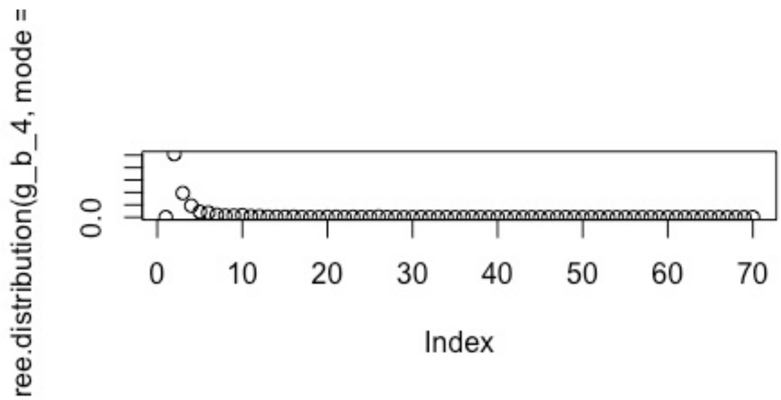


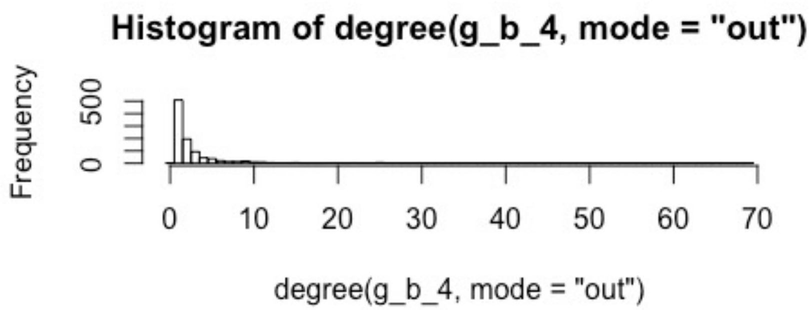
bw.factor= 1.2



Histogram of degree(g_b_4, mode = "in")







Community structure

We used the cluster edge betweenness algorithm to find the communities in the 1000 node directed network. The number of communities and the modularity of the partitioning for various values of bw.factor and fw.prob are tabulated below:

fw.prob	bw.factor	Number of comms.	Modularity
0.1	1	132	0.78
0.2	1	304	0.66
0.3	1	284	0.35
0.4	1	385	0.20
0.3	0.6	373	0.50
0.3	0.8	212	0.58
0.3	1	307	0.45
0.3	1.2	150	0.22

From the above table it can be seen that in most of the cases as the number of communities increases the modularity decreases. It adds up since as the number of communities increase it becomes more difficult to partition them well.

Conclusion

In this homework, we generated various types of static and growing networks and analyzed different network parameters like connectivity, diameter and node distribution. We observed that for the maximum node degree increases as the probability of drawing an edge between two arbitrary vertices increases and the network connectivity depends on percolation threshold which is approximately proportional to 1/N. We discussed about fat-tailed distribution random network and briefly discussed about modularity. We observed that the modularity increases as the number of node increases in the network. We also compared the different types of growing network based on preferential attachment and aging model. We observed that the variation in the network structure for varying preferential attachment exponent and varying aging exponent. In the last exercise, we found from the plots of degree distribution, as we increase the value of bw.factor the maximum node degree (both in and out) increases and the distribution has a heavier tail. However, the increase is much lower as compared to the increase obtained by increasing the value of fw.prob. This makes sense, since bw.factor controls the backward burning probability and the model grows at a slower rate compared to the growth obtained by increasing forward burning probability. Also, another observation is that max(node in degree) > max(node out degree) if bw.factor < 1.

Dictionary References

[1] Erdős–Rényi_model – Wikipedia -https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model

[2] Percolation Theory – Wikipedia- https://en.wikipedia.org/wiki/Percolation_theory

[3] Barabási–Albert_model - Wikipedia - https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model

[4] http://www.inside-r.org/packages/cran/igraph/docs/pa_age

[5] https://en.wikipedia.org/wiki/Star_network

[5] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, KDD ’05, pages 177–187, New York, NY, USA, 2005. ACM.

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD), 1, March 2007.

[7] Fortunato, Santo. "Community detection in graphs." Physics reports 486.3 (2010): 75-174.

Appendix

Preferential attachment and aging model:

This is a discrete time step model of a growing graph. We start with a network containing a single vertex (and no edges) in the first time step. Then in each time step (starting with the second) a new vertex is added and it initiates a number of edges to the old vertices in the network. The probability that an old vertex is connected to is proportional to

$$P[i] \sim (c \cdot k_i^{\alpha} + a)(d \cdot l_i^{\beta} + b) \cdot$$

Here k_i is the in-degree of vertex i in the current time step and l_i is the age of vertex i. The age is simply defined as the number of time steps passed since the vertex is added, with the extension that vertex age is divided to be in aging.bin bins.

c, α, a, d, β and b are parameters and they can be set via the following arguments: pa.exp (α, mandatory argument), aging.exp (β, mandatory argument), zero.deg.appeal (a, optional, the default value is 1), zero.age.appeal (b, optional, the default is 0), deg.coef (c, optional, the default is 1), and age.coef (d, optional, the default is 1).