Graph and Networks
Homework 1

Mansee Jadhav - 204567818
Ronak Sumbaly - 604591897

April 10, 2016

Ques 1. Creating Random Networks

Part (a)

3 random network graphs were created using **random.graph.game** method of **igraph** package. The probability p for drawing an edge between two arbitrary vertices of each graph is $0.01, 0.05, 0.1$ respectively. The **degree distribution** for each graph is as follows:

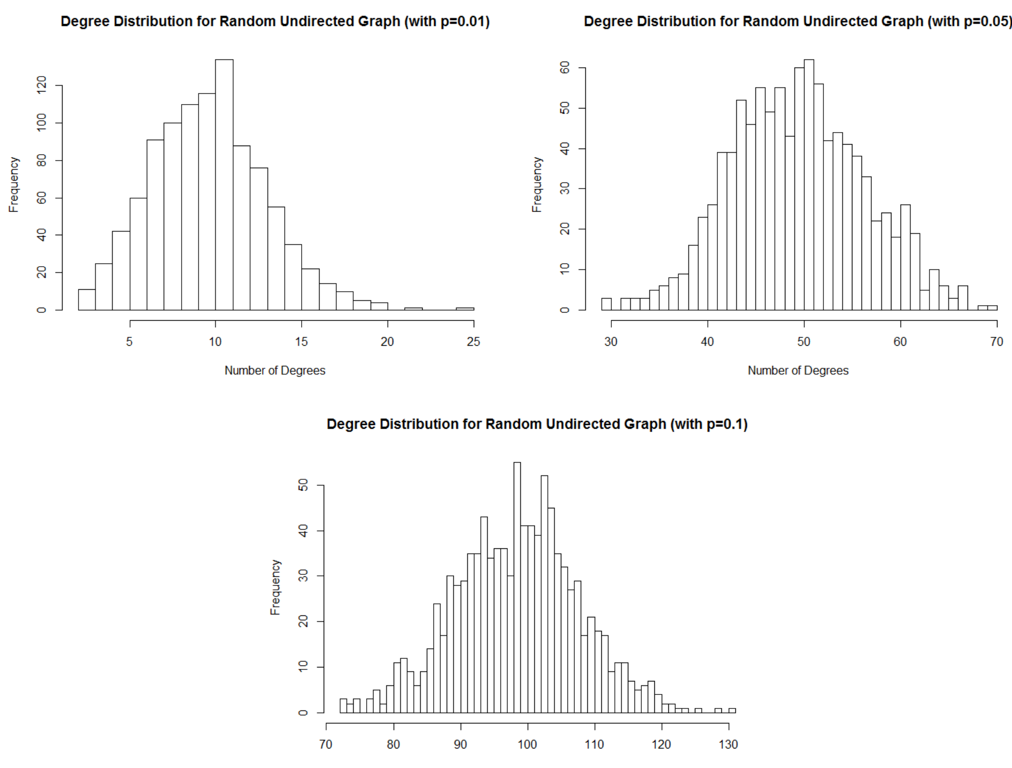

Figure 1: Degree Distribution of 3 graphs with their respective probabilities

Part (b)

The connectivity and diameters of the initial graphs were observed as follows

Probability	Connectivity	Diameter
0.01	TRUE	5.00
0.05	TRUE	3.00
0.1	TRUE	3.00

Since the graphs generated are random, we observed the graphs 100 times and averaged the results to obtain different probabilities of connectivity and diameters. The results obtained were as follows.
Note: Connectivity 0.99 signifies out of 100 observations 99 times the graph created were connected.

Probability	Average Connectivity	Average Diameter
0.01	0.95	5.39
0.05	0.99	2.97
0.1	0.99	2.97

Part (c)

A value p_c is defined such that the generated random networks are disconnected when $p < p_c$ and connected when $p > p_c$. Using **is.connected()** method we checked if the graph is connected or disconnected each time increasing the value of p by 0.001, starting from 0.000. We recorded 100 different observations for 100 such random graphs and took the average to get accurate results.

$$\textbf{Threshold Probability} = \mathbf{p_c} = \mathbf{0.00762}$$

Part (d)

Research shows that we can also analytically derive the value of threshold probability (p_c) using the **Erdos-Renyi Asymptotic Expression**, the random graph $G_p(N)$ is disconnected if the link density p is below the connectivity threshold using the formula,

$$p_c \sim \frac{log(N)}{N} \quad p > p_c$$

$$p_c \sim \frac{log(1000)}{1000} = \mathbf{0.0069}$$

Thus it can be seen that the analytically value calculated is very close to the value computed through the program.

Ques 2. Fat Tailed Distribution

Part (a)

The degree distribution resulting from **BA model (barabasi.game())** is scale free, it is power law of form P(k) $\sim k^{-3}$. We created undirected network with 1000 nodes whose degree distribution is proportional to k^{-3} using BA model. The degree distribution plotted was as follows,

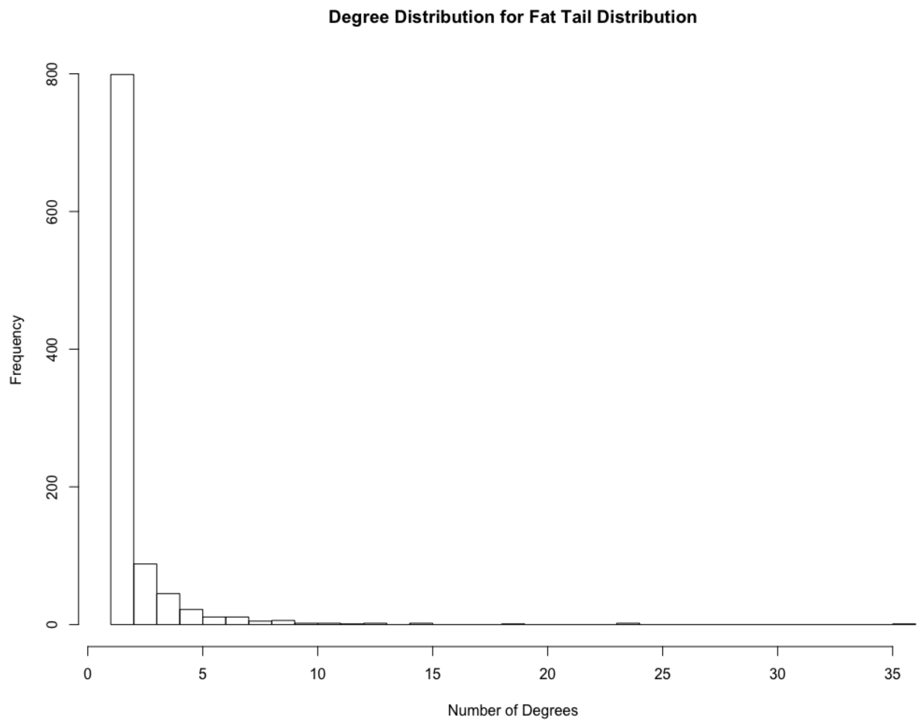

Figure 2: Degree distribution of Fat Tail Model using Barabasi Game

In order to calculate the diameter of the graph we used the **diameter()** method in the igraph package

Diameter Fat Tail Distribution = 19

Part (b)

Similar to before the connectivity of the graph generated from the Fat Tail Distribution was calculated by using is.connected() method. The connectivity found was,

Fat Tail Distribution Graph Connected = True

Modularity of the graph is the measure the strength of division of a network into modules (also called groups, clusters or communities).

In order to calculate the modularity of the graph, we defined a function which does following:

1. Gets all the clusters from graph using **clusters()** method and then get the cluster with the maximum size which represents GCC.
2. Get nodes which are not present in GCC cluster by checking their membership. Remove nodes from the graph to get only the GCC using **delete.vertices**.
3. Get community of GCC using **fastgreedy.community()** method.
4. Get modularity of GCC using **modularity()** method.

The above procedure was executed and the GCC, Community Structure and Modularity were observed. Since the network is connected the **GCC was found to be the entire network**. The other results were as follows,

$$\textbf{Community Structure} = \textbf{Number of Communities} = 35$$
$$\textbf{Modularity of Graph} = 0.9273438$$

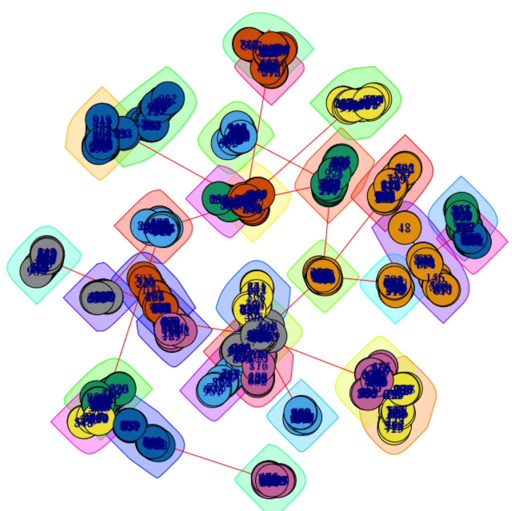

Figure 3: Community Structure of Fat Tail Graph

The high modularity of the network can be attributed to it having dense connections between the nodes within modules but sparse connections between nodes in different modules. This is due to the network being generated from a barabasi model where in new nodes are linked to nodes having higher degree.

Part (c)

The above method was repeated for a larger network using 10000 nodes. The value of modularity obtained was,

$$\textbf{Average Modularity} = \mathbf{0.9781733}$$

It can be seen that there is just a slight increase in the modularity for the larger network as compared to the smaller network since the number of nodes has increased making the graph more dense.

Part (d)

We now randomly choose a node i and randomly pick its neighbor j and plot degree distribution of nodes j that are picked in this process.

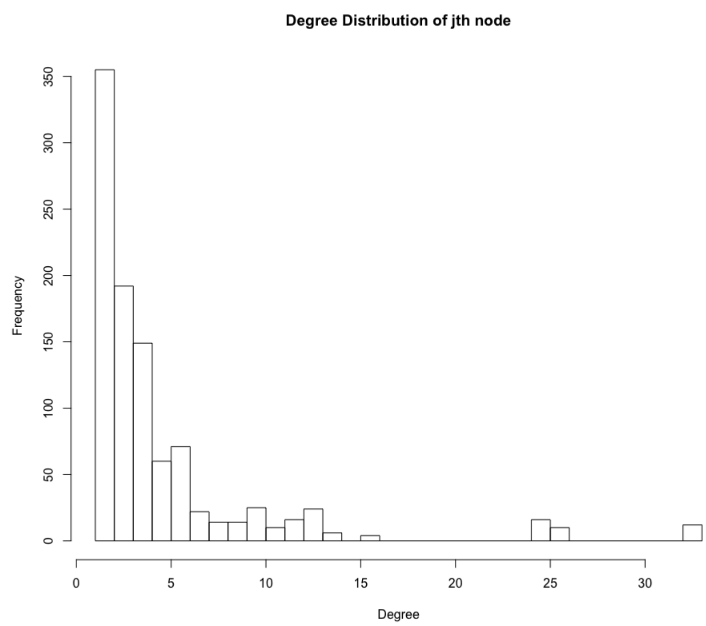

Figure 4: Degree distribution of j^{th} Node which is randomly picked neighbor of i

Ques 3. Simulating Evolution

Part (a)

In order to generate graphs that are simulated by its evolution we use **aging.prefatt.game()** method of the **igraph** package. The degree distribution of evolving graph was :

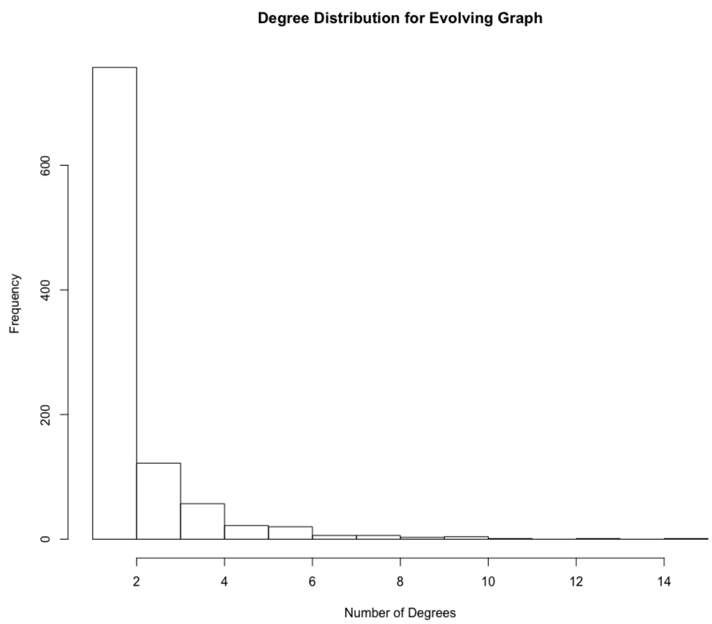

Figure 5: Degree distribution of Evolving network

Part (b)

The modularity and community structure were computed using the method described in Ques. 2.

$$\textbf{Community Structure} = \textbf{Number of Communities} = 32$$
$$\textbf{Modularity of Graph} = 0.9354179$$

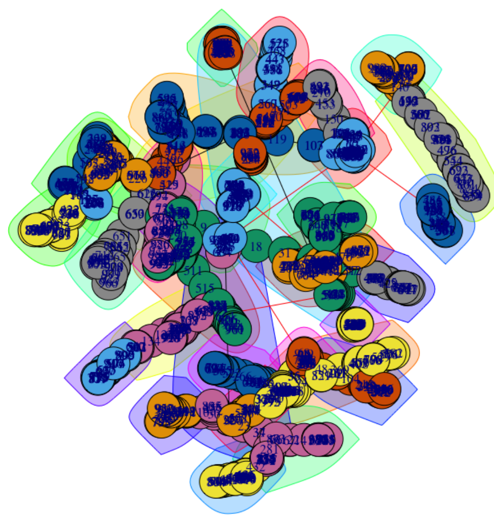

Figure 6: Community Structure of Evolving Graph

Ques 4. Forest Fire Model - Directed Network

Part (a)

Forest Fire network was created using **forest.fire.game()** method of the **igraph** package. This is a growing network model, which resembles how the forest fire spreads by igniting trees close by. We chose the forward burning probability as 0.37 and backward probability as $\frac{0.32}{0.37}$. The In and Out degree distribution for this network was as follows:

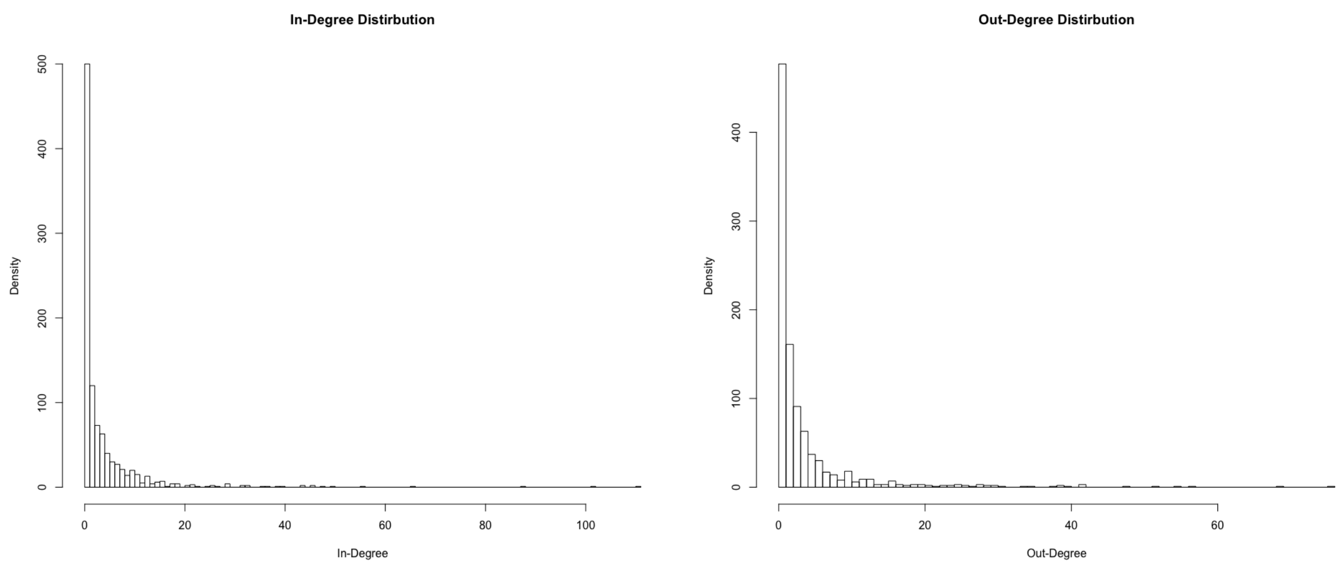

Figure 7: In and Out Degree Distribution of Forest Fire Network

Part (b)

The **diameter()** method was used to obtain the diameter of the network,

Diameter Forest Fire Graph = 11

Part (c)

Similarly the modularity and community structure of the Forest Fire Network Graph was computed using methods defined above,

$$\textbf{Community Structure} = \textbf{Number of Communities} = 25$$
$$\textbf{Modularity of Graph} = 0.5512971$$

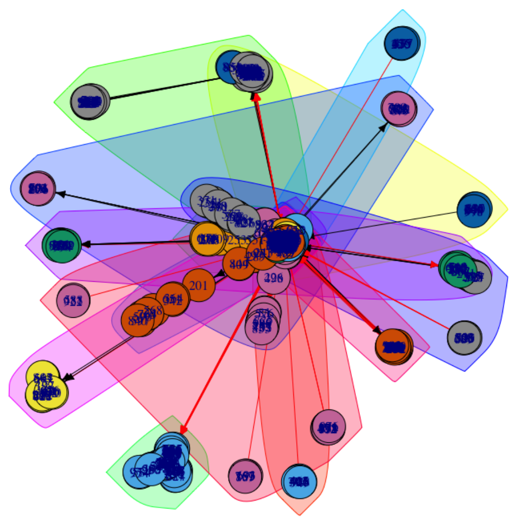

Figure 8: Community Structure of Forest Fire