EE232 Project 2
Social Network Mining

Team members:
Sonali Garg (104944076)
Aashna Agarwal (404943216)
Shweta Sood(905029230)
Karan Sanwal(205028682)
Ashish Shah(804946005)

Part 1: Facebook network

Question 1: Is the facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

Solution:
Yes, the facebook network is **connected**.
It has **4039 nodes** and **88234 edges**.

Observation:
For this section, we have used the function read.graph() to construct a graph from the edge list file facebook_combined.txt. We found the network to be connected, with 4039 nodes and 88234 edges. As the network is connected, each node is reachable with every other node. There is no mutually exclusive group of nodes. Every node is friend of friend of friends of.. every other node.

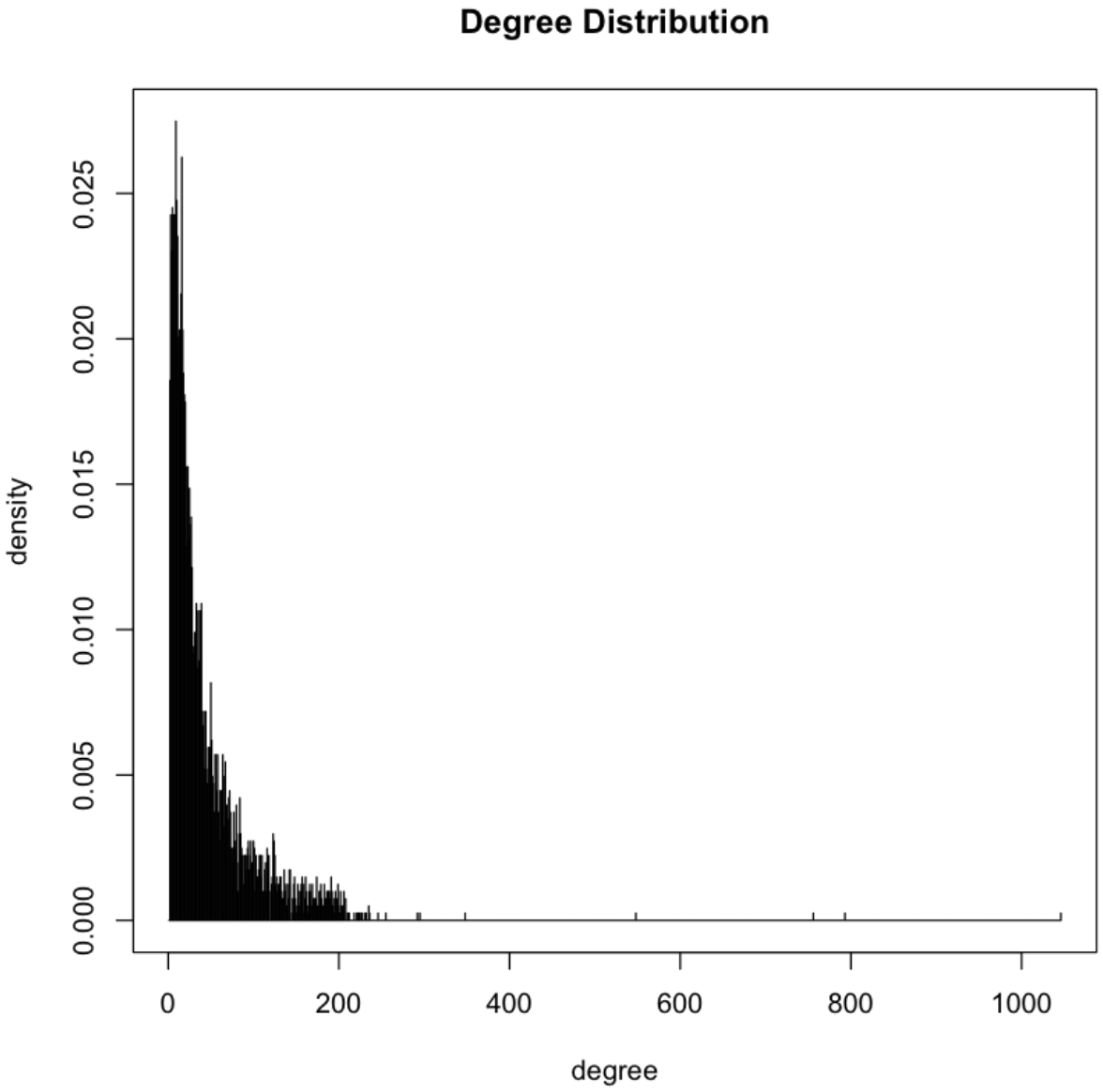
Question 2: Find the diameter of the network. If the network is not connected, then find the diameter of the GCC.

Solution:
Diameter of the network is **8**.

Observation:
Since the network was connected, we got the diameter as 8. Every node is reachable with every other node by maximum hop distance as 8.

Question 3: Plot the degree distribution of the facebook network and report the average degree.

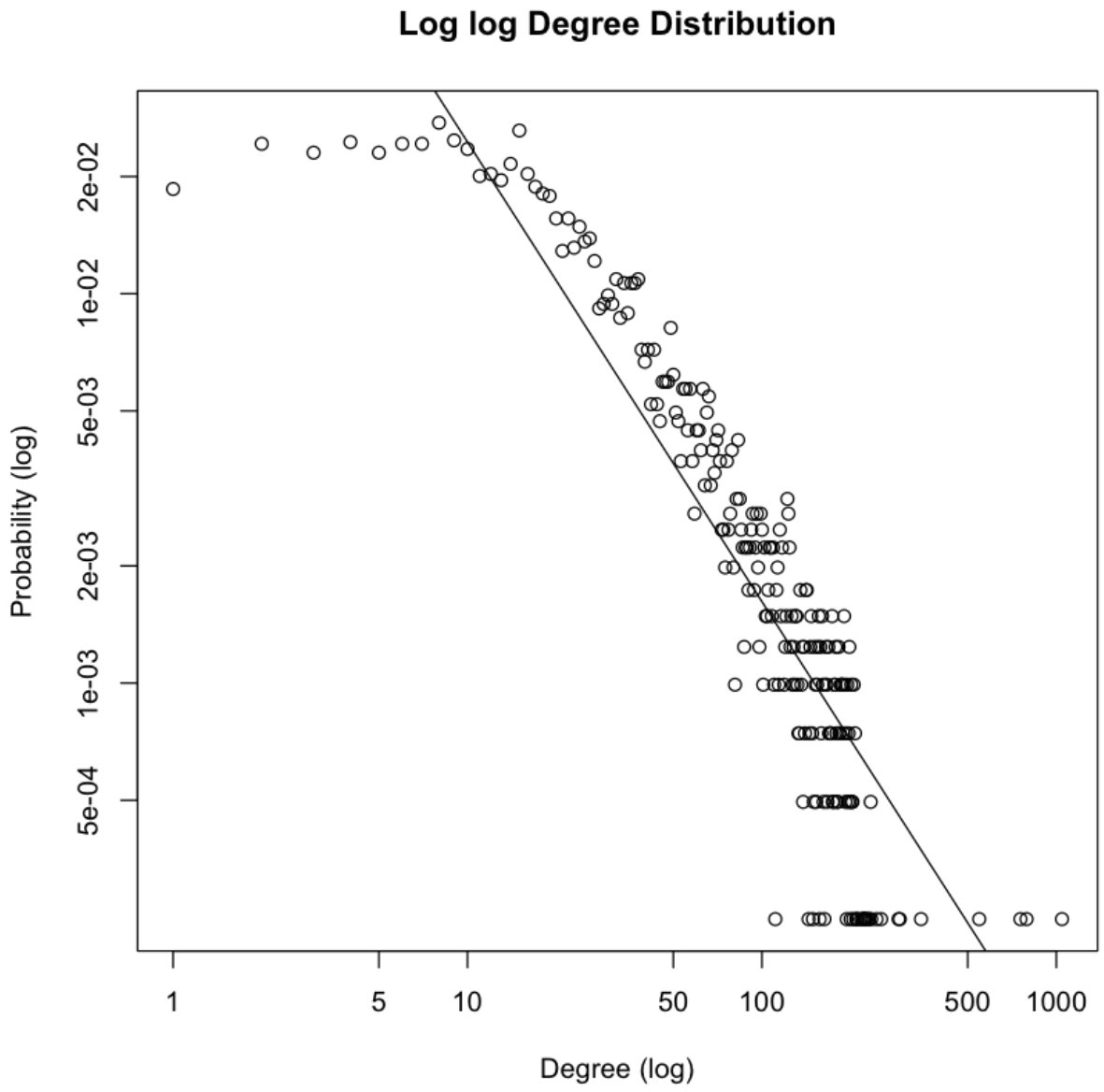
Solution:
The average degree is 43.6910126268878.



Observation:
The average degree tells us about the number of friends, a facebook user from our dataset has, which is found to be approximately 43.
One can observe that as degree increases, the density of the nodes having that degree value decreases, which is how it should be.

Question 4: Plot the degree distribution of question 3 in a log-log scale. Try to fit a line to the plot and estimate the slope of the line.

Solution:



Slope of the line is -1.1802.

Observation:

lm(formula = log10(probability) ~ log10(degree))

Coefficients:

(Intercept)	log10(degree)
-0.4309	-1.1802

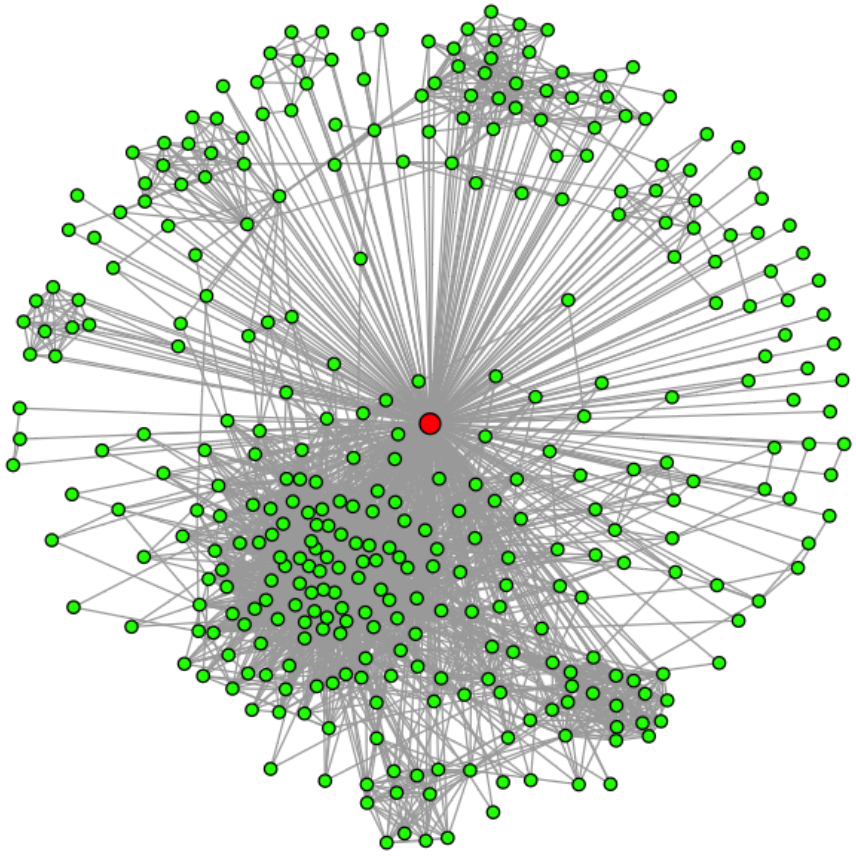
First, we removed the outliers by removing those with probability as zero, and considering only those which had the probability as non-zero. For plotting the line we used the command abline which had the argument as linear function between the log values of probability and degree.

Question 5: Create a personalized network of the user whose ID is 1. How many nodes and edges does this personalized network have?

Solution:

Number of nodes in the personalized network of the user with node ID 1 is 348

Number of edges in the personalized network of the user with node ID 1 is 2866



Observation:
For this question we take the first node in the graph and generate a subgraph consists of node 1 and its neighbors and the edges that have both ends within this set of nodes. From the figure below we can see that among all the nodes in the personal network of node 1 except for node 1, all other nodes are all friends of node 1. The number of nodes in this graph are 348 and the number of edges are 2866. The plot of the personal network is as shown above where the red dot indicates node 1.

Question 6: What is the diameter of the personalized network? Please state a trivial upper and lower bound for the diameter of the personalized network.

Solution:
Diameter of the personalised network for Node 1 is 2.
Trivial lower bound for the diameter of the personalized network (number of nodes in the network = 1) : 0
Trivial lower bound for the diameter of the personalized network (number of nodes in the network > 1) : 1
Trivial upper bound for the diameter of this connected personalized network: 2

Observation:
Diameter of a personalised network can take only 3 values, namely 0,1 and 2. The personalized network with atleast 1 friend has a range of 1 and 2 as the value for the diameter of the network since the graph is fully connected.

Since each node in the personalized network, except the node in question, is a friend of that node, the diameter of the node takes a value of 2 as every node (neighbour1) is reachable by every other node (neighbour2) by traversing through the path neighbour1 -> the node -> neighbour2.

Now if every node is friends with every other node, i.e. the case when we have ‘n’ nodes in a network and the degree of each node is ‘n-1’, then the diameter would be 1, as just 1 hop will be needed for every node to reach every other node since they would be friends.

And if the node in question has 0 friends(a special case), ie the graph has only 1 vertex but 0 edges, the diameter of the graph takes a value of 0.

Question 7: In the context of the personalized network, what is the meaning of the diameter of the personalized network to be equal to the upper bound you derived in question 6. What is the meaning of the diameter of the personalized network to be equal to the lower bound you derived in question 6?

Solution:
As explained above, when the diameter of the personalized network is equal to the upper bound (2) it means that there exists at least one such pair of nodes which although are friends of the node in question but aren't friends of each other. That is the graph is not fully connected.
Whereas, when the diameter of the personalized network is equal to the lower bound (1) this means that there exists the graph which is fully connected so it takes only 1 hop to go from 1 node to another.
Moreover, when a node has no friends, that is the personalised network for that node has 0 edges and 1 node in total then the diameter of that network will be (0), the lower bound of network of a new node with no friends.

Question 8: How many core nodes are there in the Facebook network. What is the average degree of the core nodes?

Solution:
There are **40 core nodes** are there in the Facebook network.
Average degree of the core nodes is **279.375**

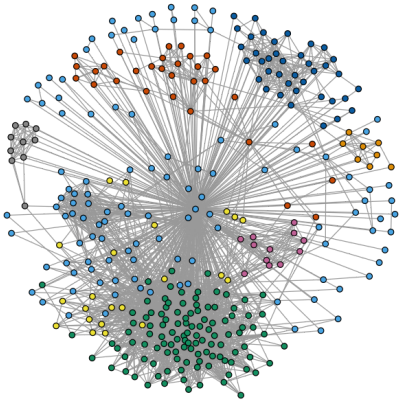
Observation:
In this question we determine the core nodes i.e. the nodes that have more than 200 neighbors. There are 40 such core nodes in this network and the average degree of these core nodes is 279.375.

Question 9: For each of the above core node's personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's personalized networks using colors. Nodes belonging to the same community should have the same color and nodes belonging to different communities should have different color. In this question, you should have 15 plots in total.

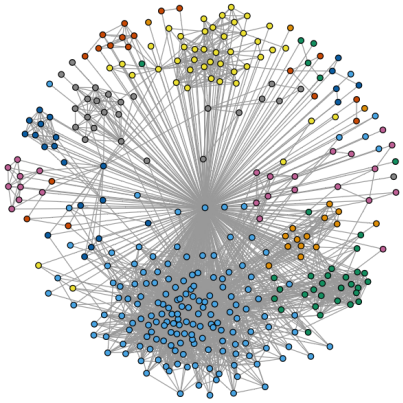
Solution:
a) Node ID 1

1. **Fast-Greedy** Modularity score: **0.413101372834235**
2. **Edge-Betweenness** Modularity score: **0.35330217254633**
3. **Infomap** Modularity score: **0.389118471050977**

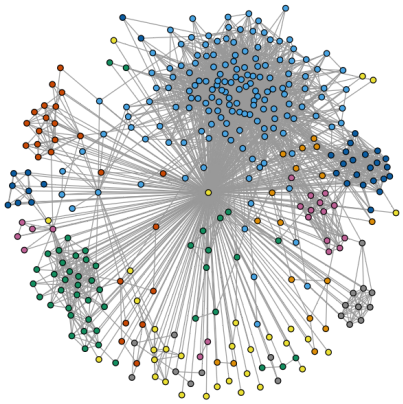
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



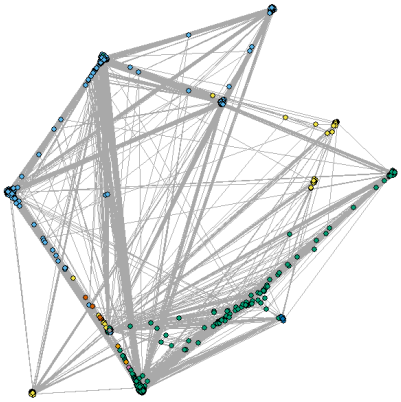
Network Plot using Infomap Algorithm



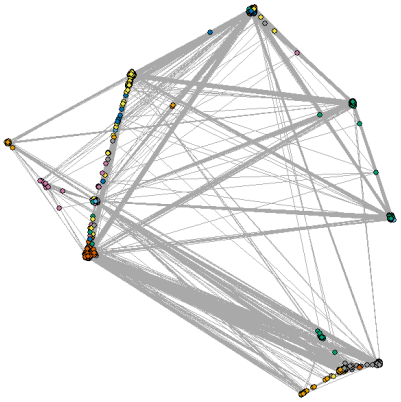
b) Node ID 108

1. **Fast-Greedy** Modularity score: **0.435929376026475**
2. **Edge-Betweenness** Modularity score: **0.506754916538902**
3. **Infomap** Modularity score: **0.508223340384871**

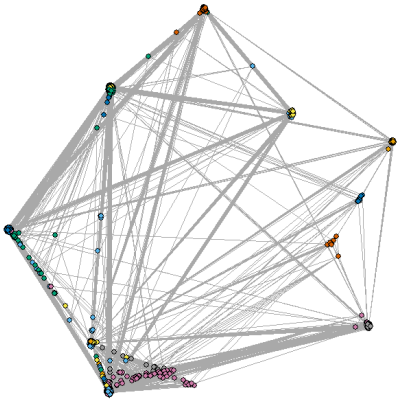
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



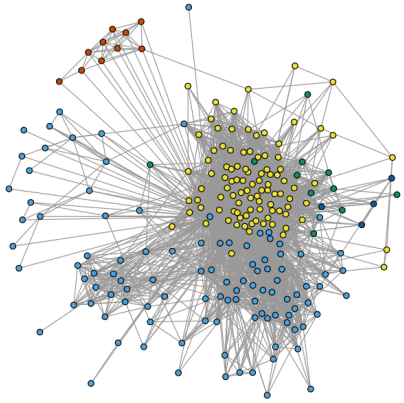
Network Plot using Infomap Algorithm



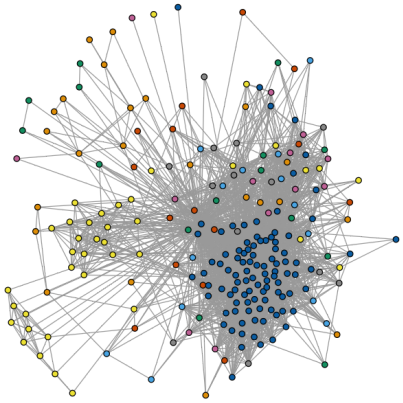
c) Node ID 349

1. **Fast-Greedy** Modularity score: **0.251714858543331**
2. **Edge-Betweenness** Modularity score: **0.133528021370078**
3. **Infomap** Modularity score: **0.0960290802375342**

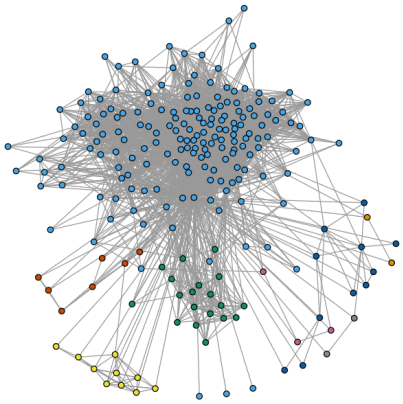
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



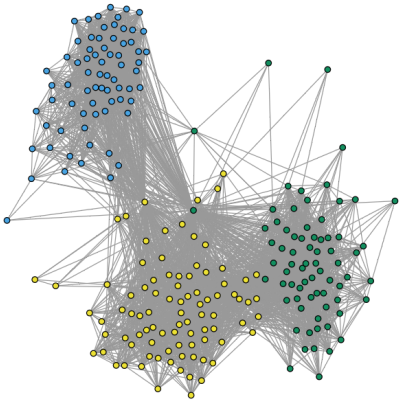
Network Plot using Infomap Algorithm



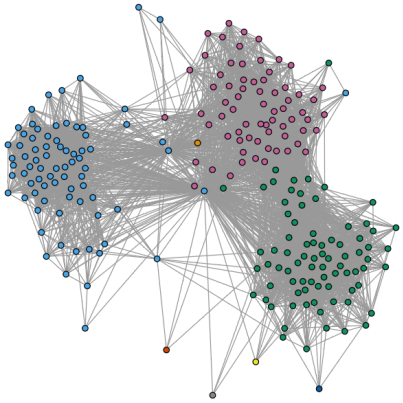
d) Node ID 484

1. **Fast-Greedy** Modularity score: **0.507001642196514**
2. **Edge-Betweenness** Modularity score: **0.489095180244803**
3. **Infomap** Modularity score: **0.515278752174842**

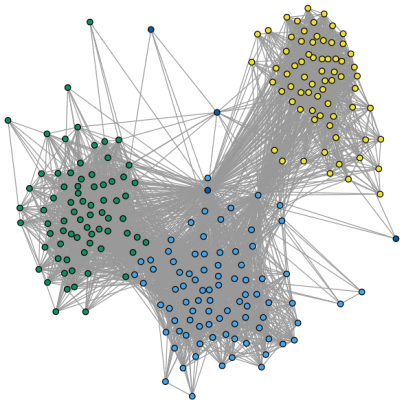
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



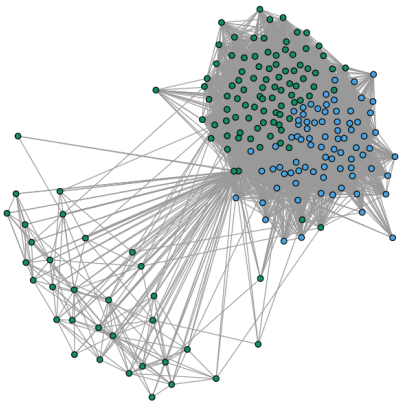
Network Plot using Infomap Algorithm



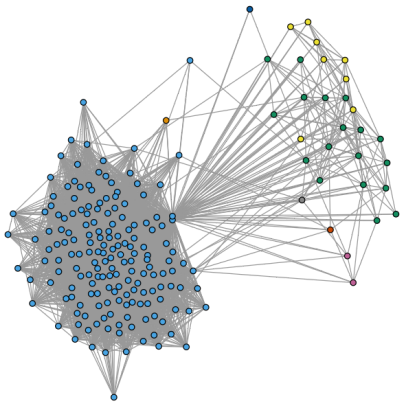
e) Node ID 1087

1. **Fast-Greedy** Modularity score: **0.145531499565493**
2. **Edge-Betweenness** Modularity score: **0.027623772388464**
3. **Infomap** Modularity score: **0.0269066172233357**

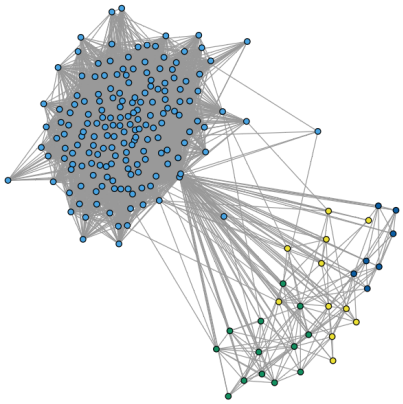
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



Network Plot using Infomap Algorithm



Observation:

We see that fast-greedy and edge-betweenness algorithms are more stable as compared to the infomap algorithm. After running the algorithms about 10-15 times we saw that the first 2 mentioned algorithms took stable values and did not move around much but when compared with infomap, we saw sometimes the values changed in a range of 0.1.

We also observed that the edge-betweenness algorithm is slower and took a lot of time to run especially for the second node in question.

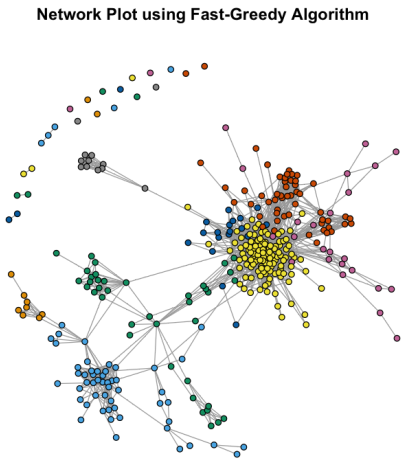
The communities in the figures above are distinguished with colors and it can be seen that in the communities plotted there is some overlap. Also it is observed that Edge-Betweenness algorithm tends to break the graph into more partitions than the other two algorithms.

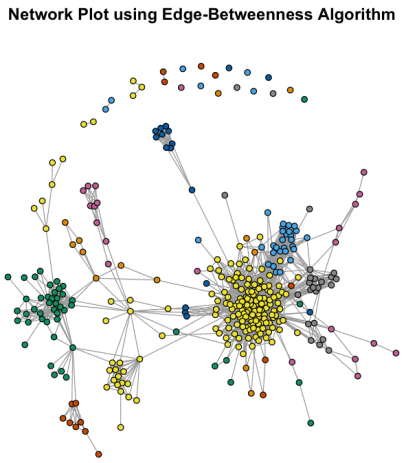
Question 10: For each of the core node's personalized network(use same core nodes as question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as question 9. Compare the modularity score of the community structure of the modified personalized network with the modularity score of the community structure of the personalized network of question 9. For visualization purpose, display the community structure of the modified personalized network using colors. In this question, you should have 15 plots in total.

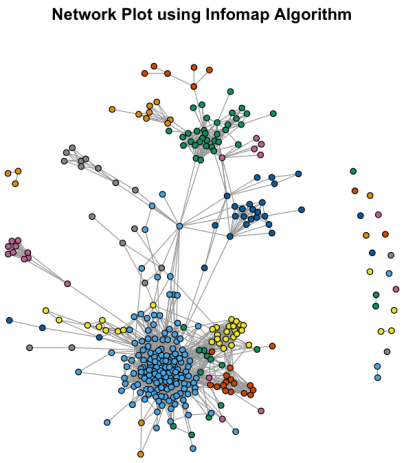
Solution:

a) Node ID 1

1. **Fast Greedy** Modularity score: **0.44185326886839**
2. **Edge-Betweenness** Modularity score: **0.41614614203983**
3. **Infomap** Modularity score: **0.418007659453891**

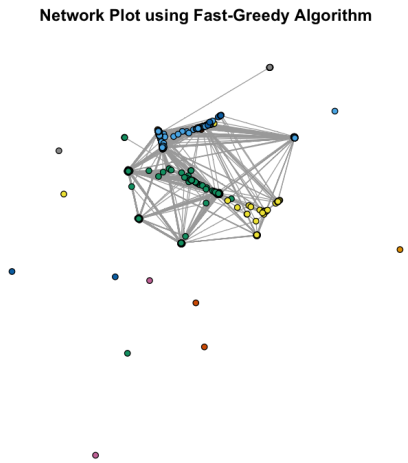


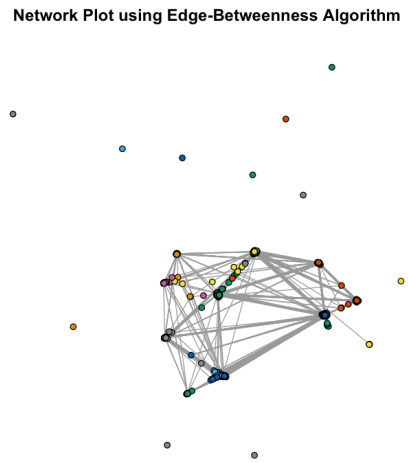


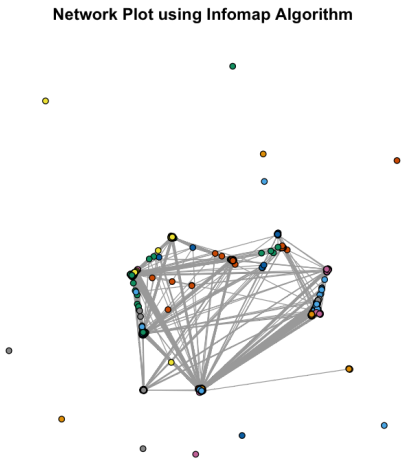


b) Node ID 108

1. **Fast Greedy** Modularity score: **0.458127093719977**
2. **Edge-Betweenness** Modularity score: **0.521321576382217**
3. **Infomap** Modularity score: **0.518217122543454**

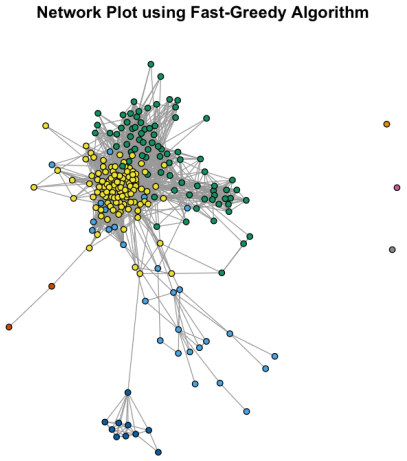


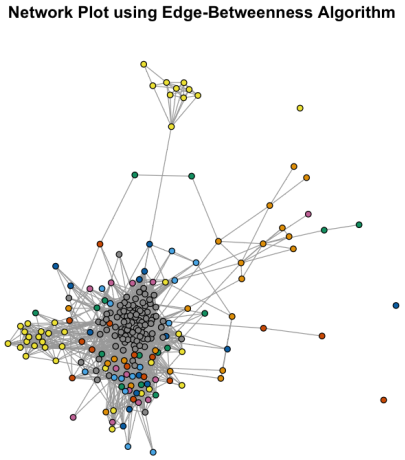




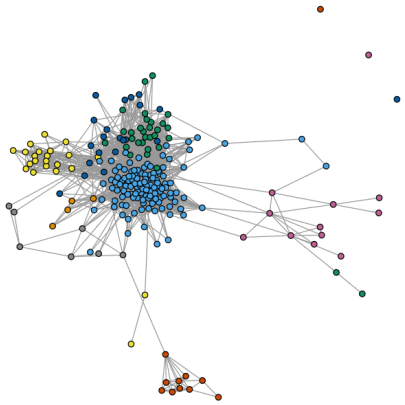
c) Node ID 349

1. **Fast Greedy** Modularity score: **0.245691795942674**
2. **Edge-Betweenness** Modularity score: **0.150566340187559**
3. **Infomap** Modularity score: **0.246578492623397**





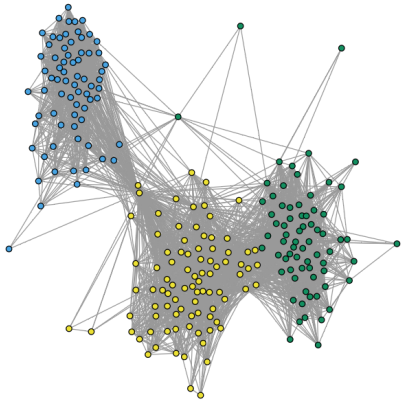
Network Plot using Infomap Algorithm



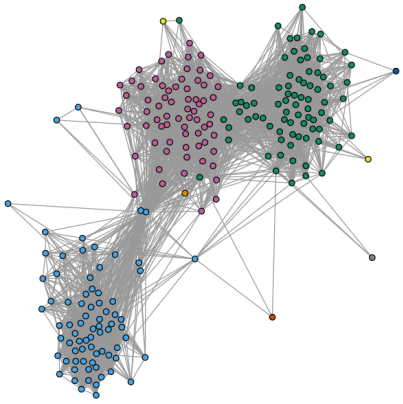
d) Node ID 484

1. **Fast Greedy** Modularity score: **0.534214154606172**
2. **Edge-Betweenness** Modularity score: **0.515441277123504**
3. **Infomap** Modularity score: **0.543443679279522**

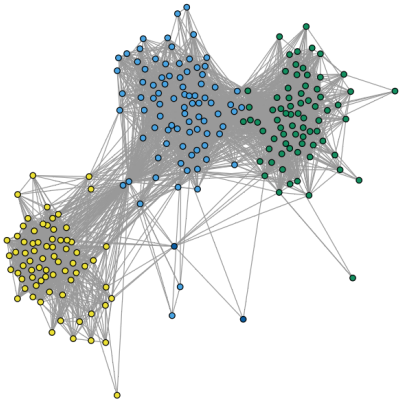
Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm



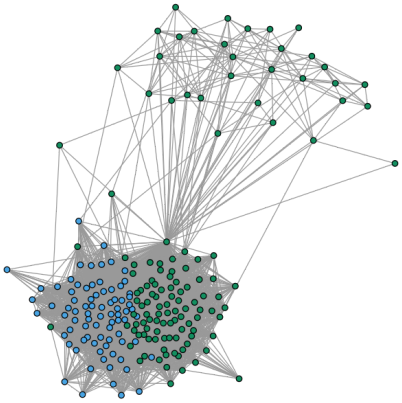
Network Plot using Infomap Algorithm



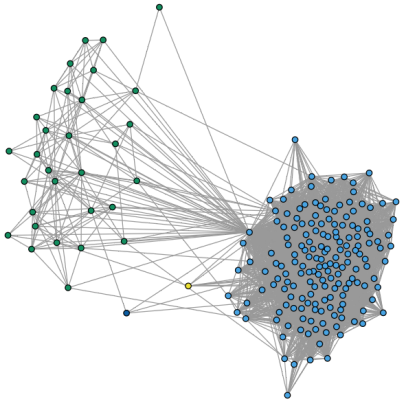
e) Node ID 1087

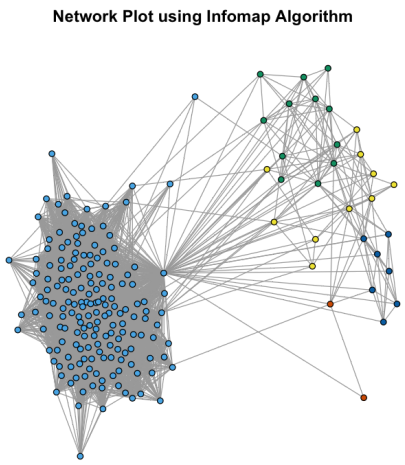
1. **Fast Greedy** Modularity score: **0.148195631953499**
2. **Edge-Betweenness** Modularity score: **0.0324952980499142**
3. **Infomap** Modularity score: **0.0273715944871148**

Network Plot using Fast-Greedy Algorithm



Network Plot using Edge-Betweenness Algorithm





Observation:
For this problem, core nodes were removed from the personal network and then the community structure was determined again using the three community detection algorithms as in question 9. It was observed that the partitions are similar to those in part 3 even though they are structured without the core node. Also there is a difference of approximately 10% in the modularity with respect to part 3. The plots are shown after.

We see that from removing the core node, the values for edge-betweenness algorithm has increased but the values for fast-greedy and infomap remains stable. This tells us the impact the graph has on modularity when a core node is removed for the edge-betweenness algorithm. We also see that there are more communities now in the edge-betweenness algorithm after removing the core node.

Question 11: Write an expression relating the Embeddedness of a node to it's degree.

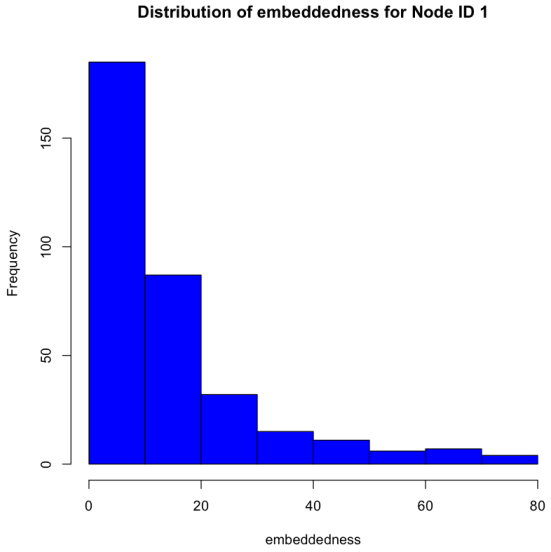
Solution:
Embeddedness of a node = D - 1,
Where D = degree of the node in personalised network.

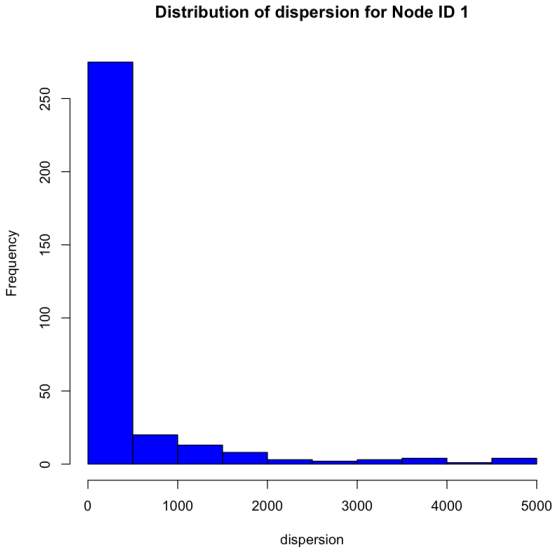
1 we have reduced for not counting the edge between the node and core node. Rest, all edges of node direct towards the mutual nodes.

Question 12: For each of the core node's personalized network (use the same core nodes as question 9), plot the distribution of embeddedness and dispersion. In this question, you will have 10 plots.

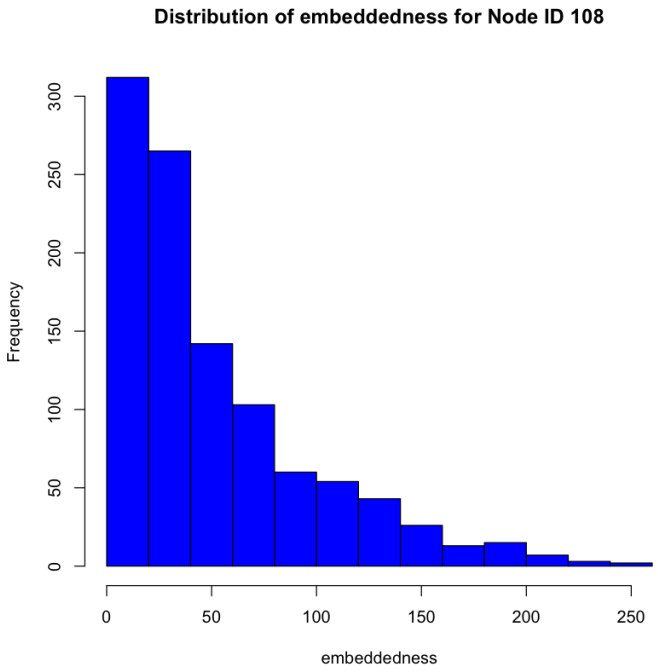
Solution:

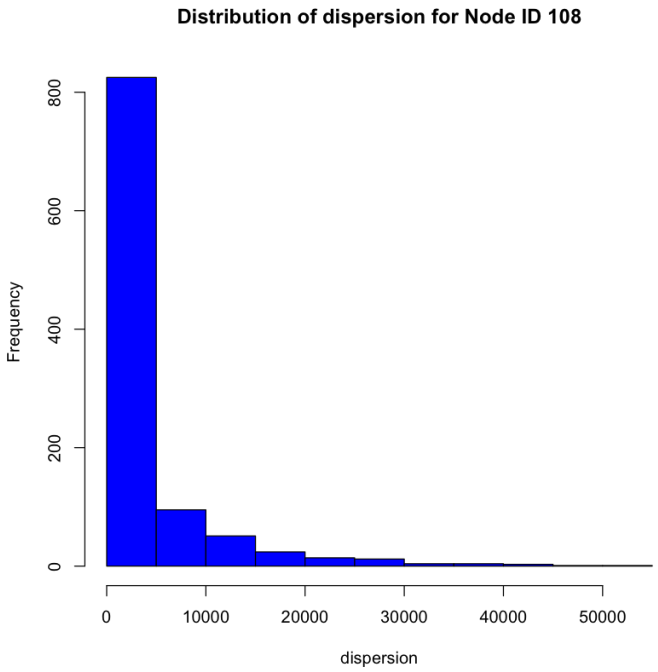
Embeddedness was calculated by counting the number of mutual friends between the ego node and other nodes in its personal network. Dispersion was calculated using Breadth-First-Search after removing both the ego node and the node for which the dispersion is being calculated. As such, the distance metric used hence becomes the length of the shortest path between two nodes.

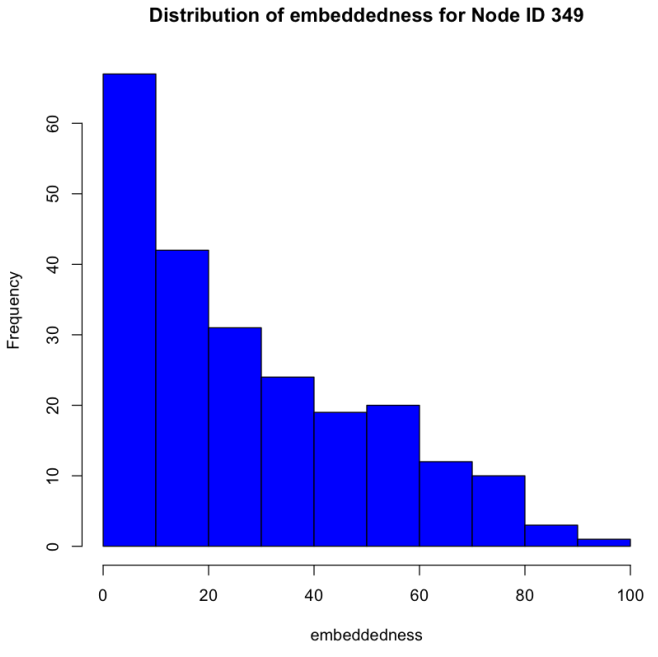


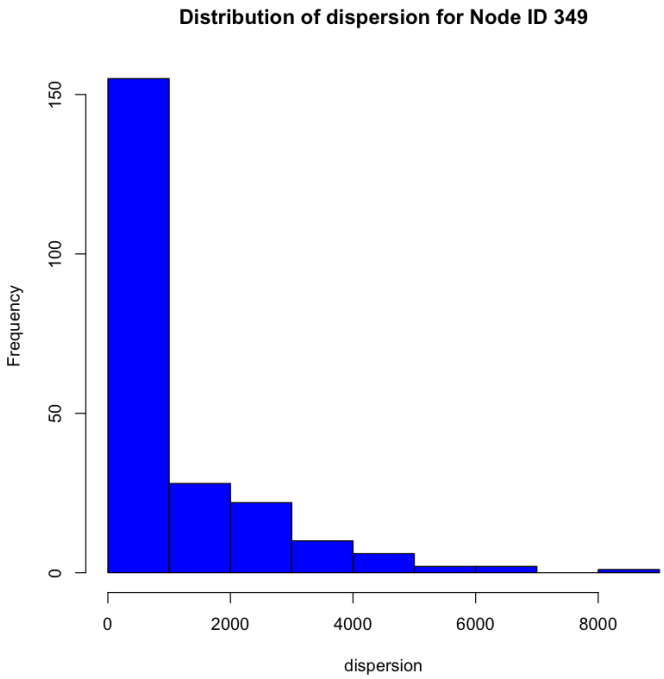


In this we see that the number of nodes with low value of embeddedness is too high meaning that the majority of the node-pairs aren't directly connected within themselves. But since we also see that low dispersion value are popular, It is fair to assume that though the the nodes being connected directly isn't a popular observed statistics, the nodes are not far away from each other, they are connected by short path consisting of intermediate nodes.

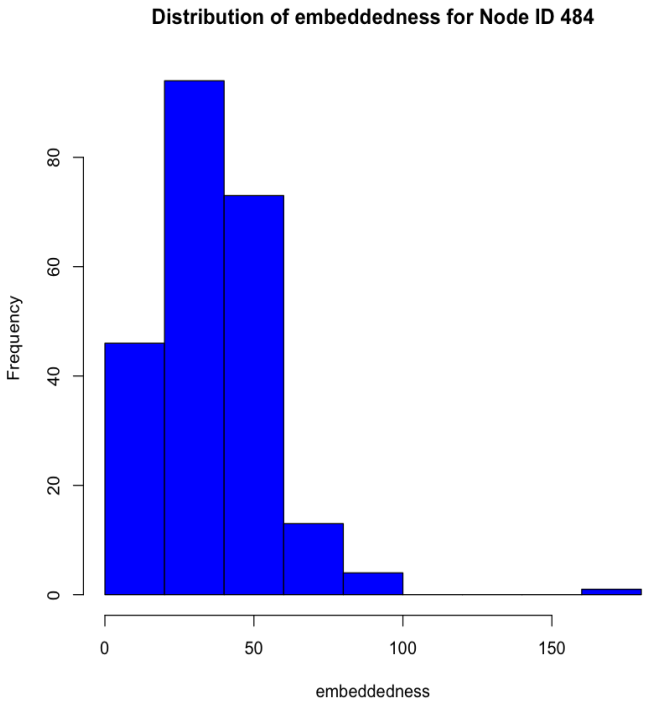

Distribution of embeddedness for Node ID 108
Frequency
0
50
100
150
200
250
300
0
50
100
150
200
250
embeddedness

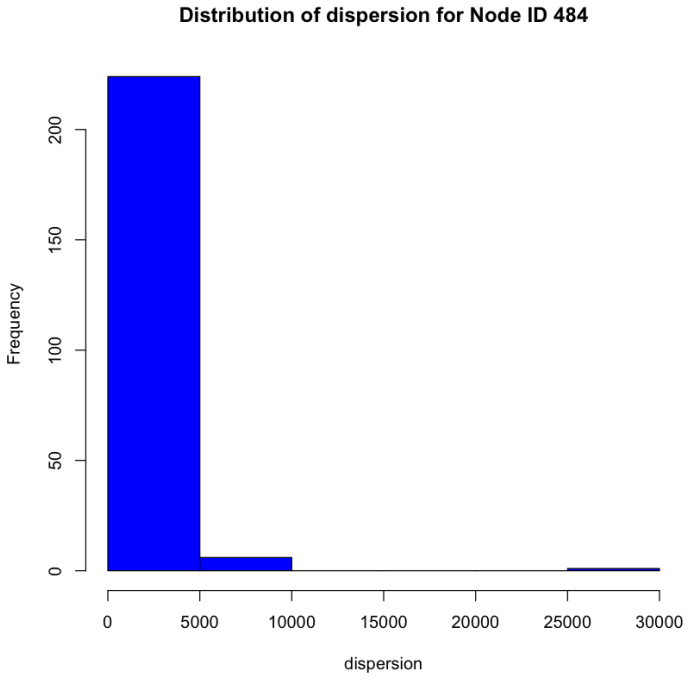

Distribution of dispersion for Node ID 108
Frequency
0
200
400
600
800
0
10000
20000
30000
40000
50000
dispersion


Distribution of embeddedness for Node ID 349
Frequency
0
10
20
30
40
50
60
0
20
40
60
80
100
embeddedness

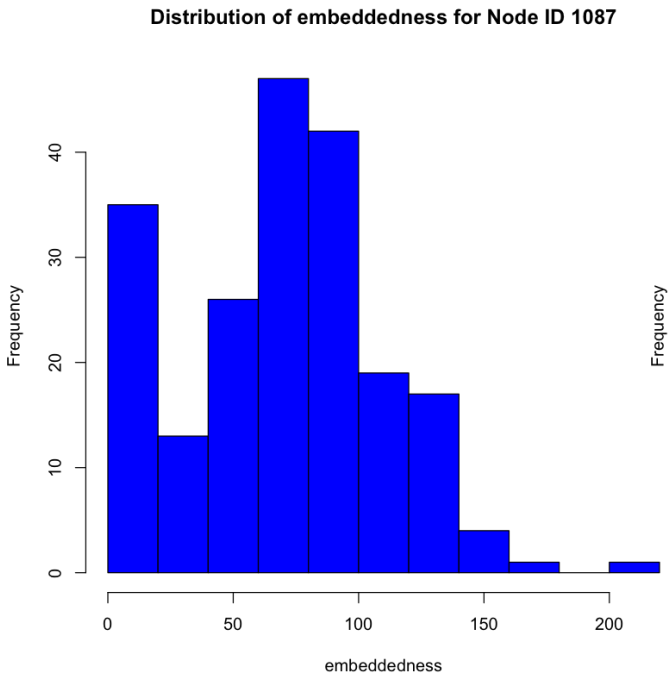

Distribution of dispersion for Node ID 349
Frequency
0
50
100
150
0
2000
4000
6000
8000
dispersion

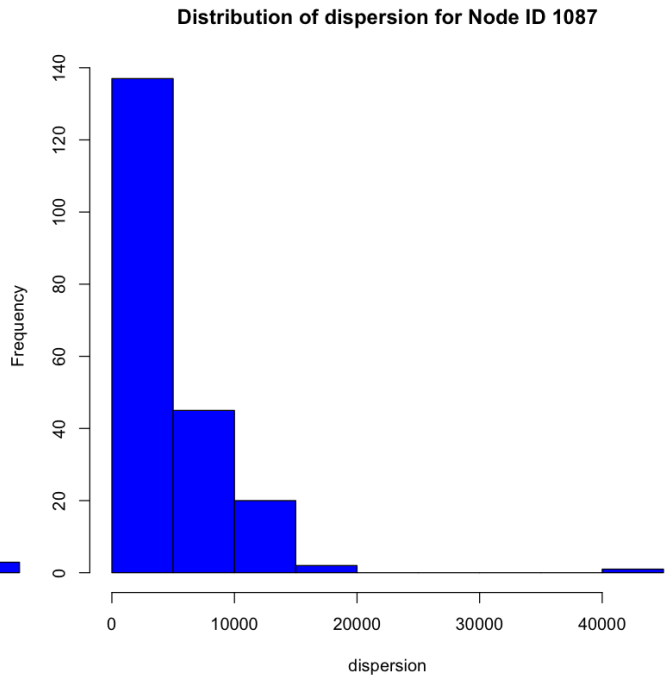
Unlike Node ID 1 , the distribution of embeddedness is better in these nodes (108,349), but still lower values seem to be more popular meaning that the majority of the node-pair in the network seem not to be directly connected. The dispersion though is similar to the one observed for Node id 1, so, the explanation is valid here as well.





The embeddedness here is comparatively a lot balanced. We see around 70-80 nodes with embeddedness value 55-75, which is good considering that the network size is 232. Lower dispersion value is expected when the embeddedness is not that low. This also means that the graph is well connected.





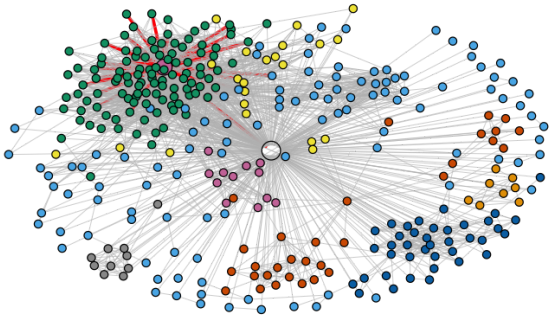
In Node ID 1087, We see a balanced distribution of embeddedness with mode values having values between 50-150 ,and we also see lower values of dispersion being more popular. Thus the Network seems to be well connected.

Question 13: For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node. To detect the community structure, use FastGreedy algorithm. In this question, you will have 5 plots.

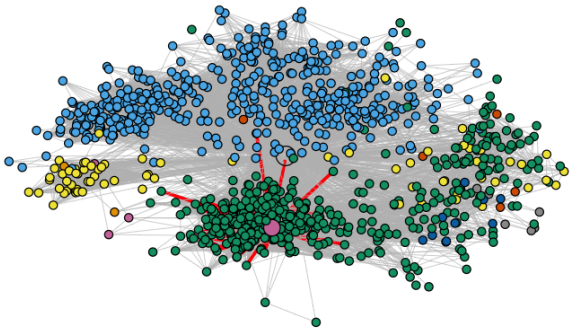
Solution:

Color codes: White for core node, Pink for node with maximum dispersion.
The incident edges to the node with maximum dispersion is red. To detect the community structure, FastGreedy algorithm was used.
For calculation of dispersion, shortest.paths (uses Breadth First Search) was used to find distance between pairs of mutual friends. If a pair of mutual friends are not connected, this distance is set to diameter of graph+5 (a constant)

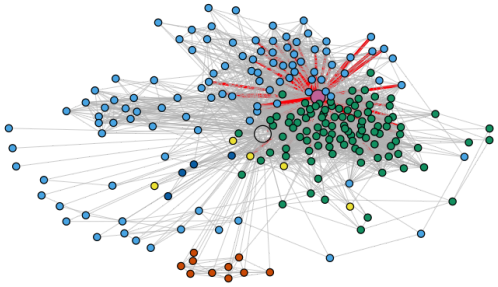
Network Plot using Fast-Greedy Algorithm for core node 1



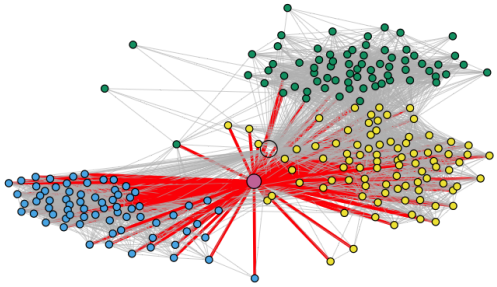
Network Plot using Fast-Greedy Algorithm for core node 108



Network Plot using Fast-Greedy Algorithm for core node 349



Network Plot using Fast-Greedy Algorithm for core node 484



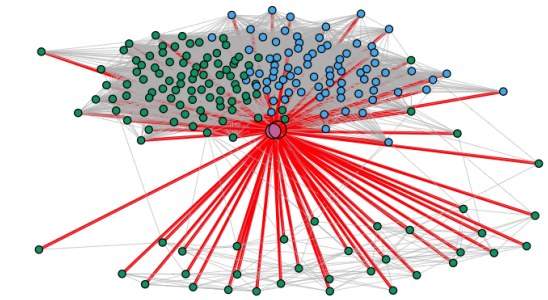


Figure_13

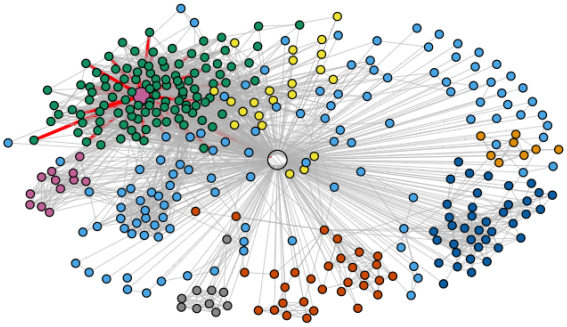
Observation:

Question 14: Repeat question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion/embeddedness . Also, highlight the edges incident to these nodes.

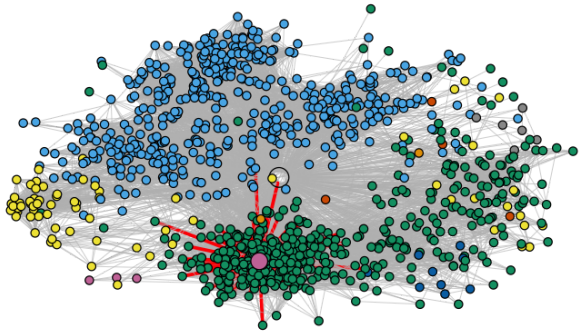
Solution:

Highlighted node with maximum embeddedness

Network Plot using Fast-Greedy Algorithm for core node 1

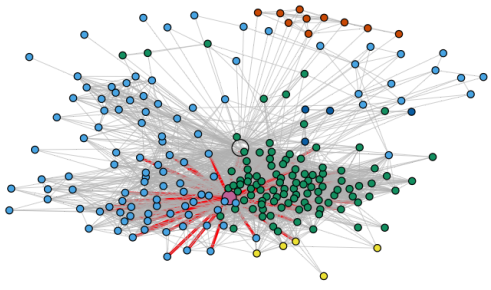


Network Plot using Fast-Greedy Algorithm for core node 108

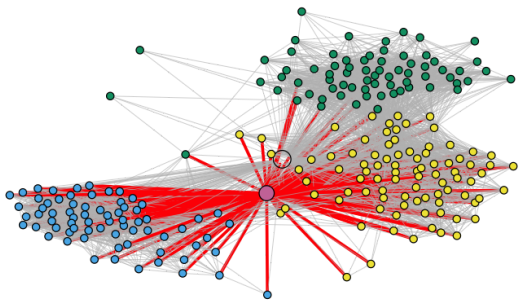


Color codes: White for core node, Pink for node with maximum embeddedness.
The incident edges to the node with maximum embeddedness is red. To detect the community structure, FastGreedy algorithm was used.
For calculation of embeddedness, shortest.paths (uses Breadth First Search) was used to find distance between pairs of mutual friends. If a pair of mutual friends are not connected, this distance is set to diameter of graph+5 (a constant)

Network Plot using Fast-Greedy Algorithm for core node 349



Network Plot using Fast-Greedy Algorithm for core node 484



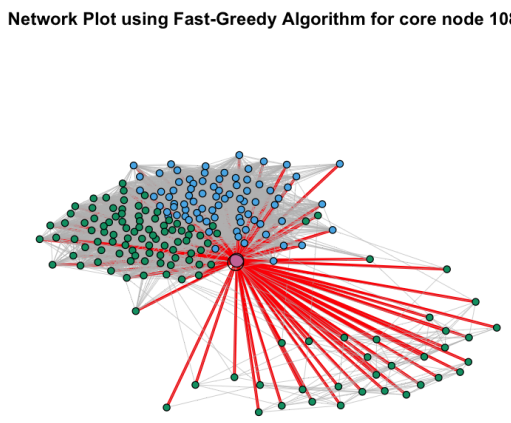
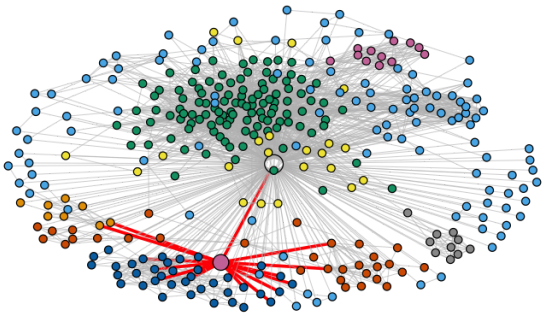


Figure_14

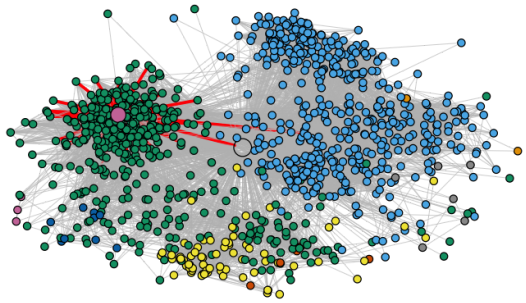
Highlighted node with maximum dispersion/embeddedness

Color codes: White for core node, Pink for node with maximum dispersion/embeddedness.
The incident edges to the node with maximum dispersion/embeddedness is red. To detect the community structure, FastGreedy algorithm was used.
For calculation of dispersion/embeddedness, shortest.paths (uses Breadth First Search) was used to find distance between pairs of mutual friends. If a pair of mutual friends are not connected, this distance is set to diameter of graph+5 (a constant)

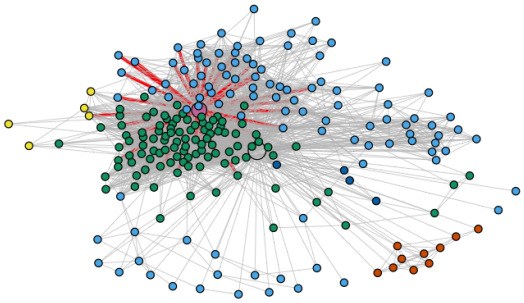
Network Plot using Fast-Greedy Algorithm for core node 1



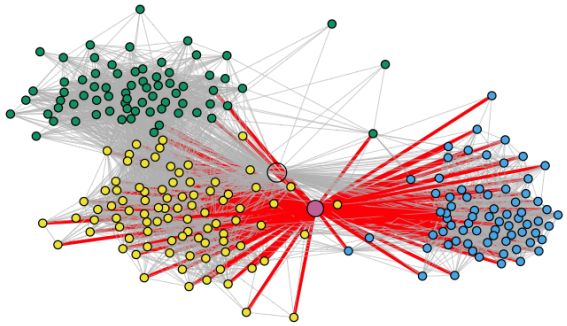
Network Plot using Fast-Greedy Algorithm for core node 108



Network Plot using Fast-Greedy Algorithm for core node 349



Network Plot using Fast-Greedy Algorithm for core node 484



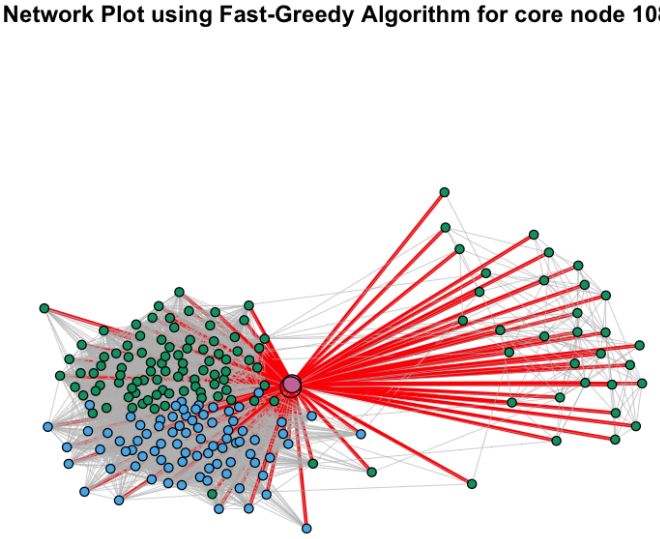


Figure_15

Question 15: Use the plots from questions 13 and 14 to explain the characteristics of a node revealed by each of this measure.

Observation:

1. Dispersion **(Figure_13)** -
    a. The pink node shares maximum dispersion with the core node. This means, the pink node is the least acquainted with the node in pink. Thus, mutual neighbors are not well connected. Also, we notice as the network size increases, the dispersion decreases because the network becomes more densely connected

ID	SIZE	DISPERSION VALUE
108	1046	51247

1087	206	40969
349	230	8258
484	232	29837
1	348	4971

2. Embeddedness (**Figure_14**) -
    a. Given an edge (u, v), its embeddedness is the number of mutual friends shared by its endpoints. Traditionally, embeddedness is associated with tie strength.
    b. The pink node has the maximum embeddedness with the core node. We also notice the pink node belongs to the core node's largest community network. This implies that in the personalized network of a core node, the community with the largest size will have a pink node that has the maximum number of mutual friends with the core node
3. Dispersion/Embeddedness **(Figure_15)** -
    a. Dispersion as a measure is not sufficient in itself. This is because it is influenced by the size of communities. If the size of community is large, the dispersion will increase, however, we still would want it to be counted as a single community relative to other nodes outside the community. To ensure this, we normalize the dispersion by embeddedness (number of mutual friends).
    b. A high dispersion to embeddedness ratio perform well to predict the correct romantic relationship - spouse detection. Eg, a couple may have different social circles implying low embeddedness and consequently their social circles will have high dispersion as they are not well connected, It has been empirically found that performance is highest for functions that are monotonically increasing in dispersion and monotonically decreasing in embeddedness. Which means, the two communities are linked together only via common nodes (here, white and pink node), indicating a romantic relationship.

Question 16: What is |Nr|?

Solution:

Nr is the list of users in the personalized network of node ID 415 with degree 24. We get 11 such nodes and hence |Nr| is 11.
31 53 75 90 93 102 118 133 134 136 137

Question 17: Compute the average accuracy of the friend recommendation algorithm that uses:

Common Neighbors measure
Jaccard measure
Adamic Adar measure
Based on the average accuracy values, which friend recommendation algorithm is the best?

Solution:
Average accuracy of the friend recommendation algorithm:

1. Common Neighbors Measure - **83.3%**

$$CommonNeighbors(i,j) = |S_i \cap S_j|$$

2. Jaccard Measure - **79.8%**

$$Jaccard(i,j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

3. Adamic Adar Measure - **84.7%**

$$AdamicAdar(i,j) = \sum_{k \in S_i \cap S_j} \frac{1}{log(|S_k|)}$$

Observation:
Based on the average accuracy values, the best friend recommendation algorithm is AdamicAdar. This is a good measure, because it gives more importance to a shared mutual node that has a small neighborhood. Eg: Person A and Person B have a mutual friend who knows very few people. This implies Person A and B have a high chance of knowing each other. Worst performing is Jaccard because it is bounded by the union of neighbors of the nodes. Hence it's value mathematically will be less than common neighbors,

Question 18: How many personal networks are there?

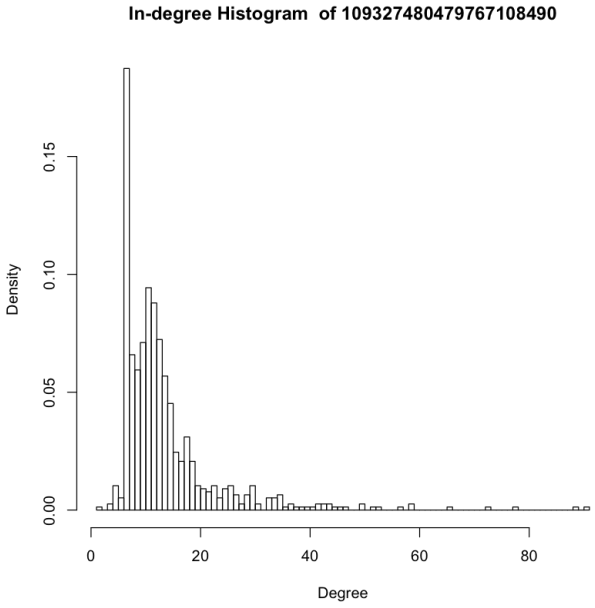
Solution : There are **57 personal networks** with more than 2 circles.

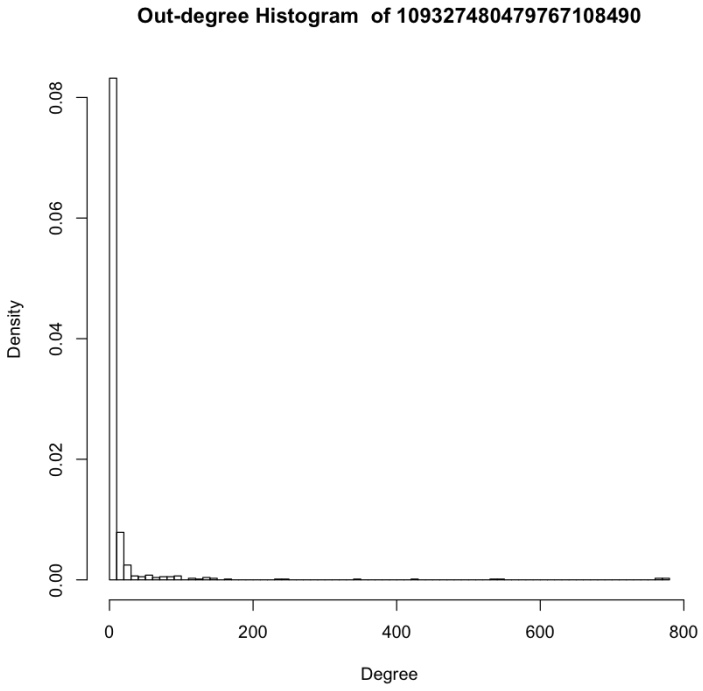
Observation : The dataset given has around 125 personal network with 2 or more circles, but only 57 out of these have strictly more than 2 circles.

Question 19: For the 3 personal networks (node ID given below), plot the in-degree and out-degree distribution of these personal networks. 9 Do the personal networks have a similar in and out degree distribution. In this question, you should have 6 plots.
• 109327480479767108490
• 115625564993990145546
• 101373961279443806744

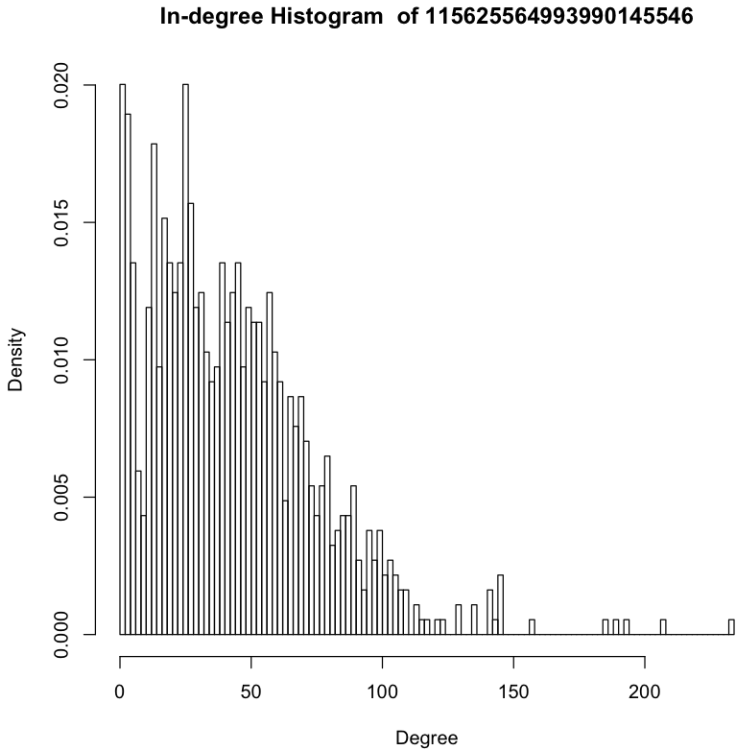
Solution :

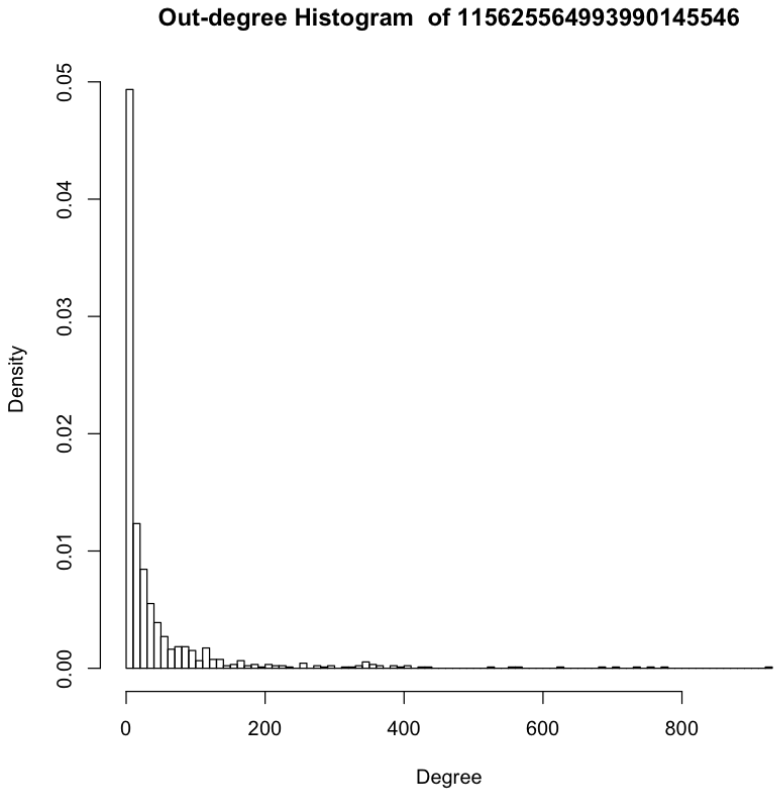
Node 1 :



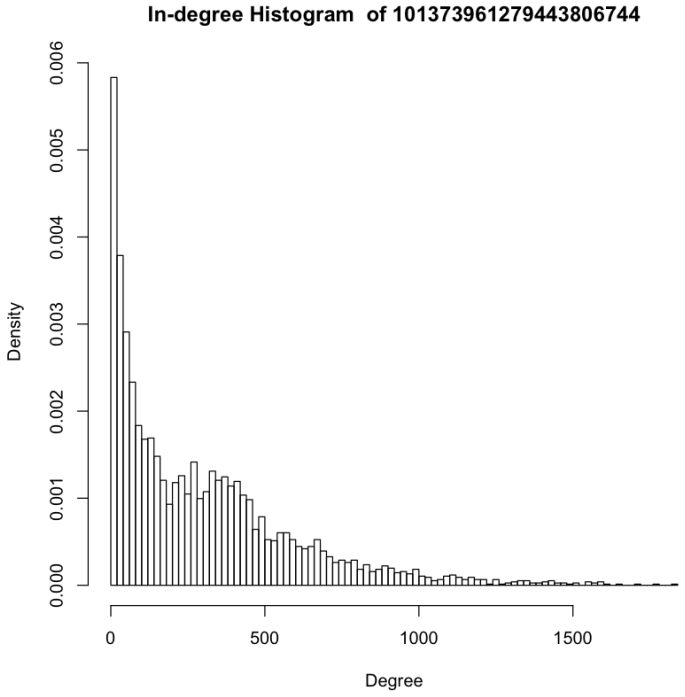


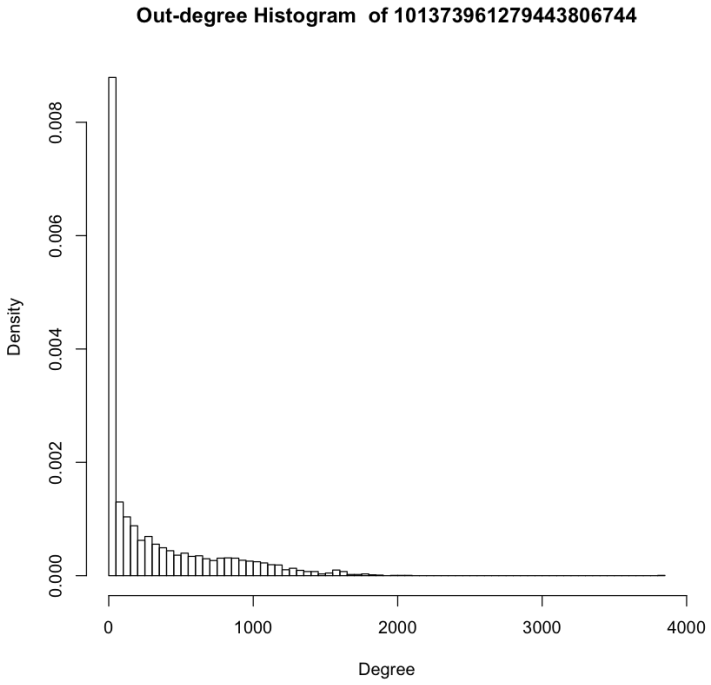
Node 2 :


In-degree Histogram of 11562556499399014554 6
Density
0.000
0.005
0.010
0.015
0.020
0
50
100
150
200
Degree


Out-degree Histogram of 11562556499399014554 6
Density
0.00
0.01
0.02
0.03
0.04
0.05
0
200
400
600
800
Degree

Node 3 :


In-degree Histogram of 10137396127944380674 4
Density
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0
500
1000
1500
Degree


Out-degree Histogram of 10137396127944380674 4
Density
0.000
0.002
0.004
0.006
0.008
0
1000
2000
3000
4000
Degree

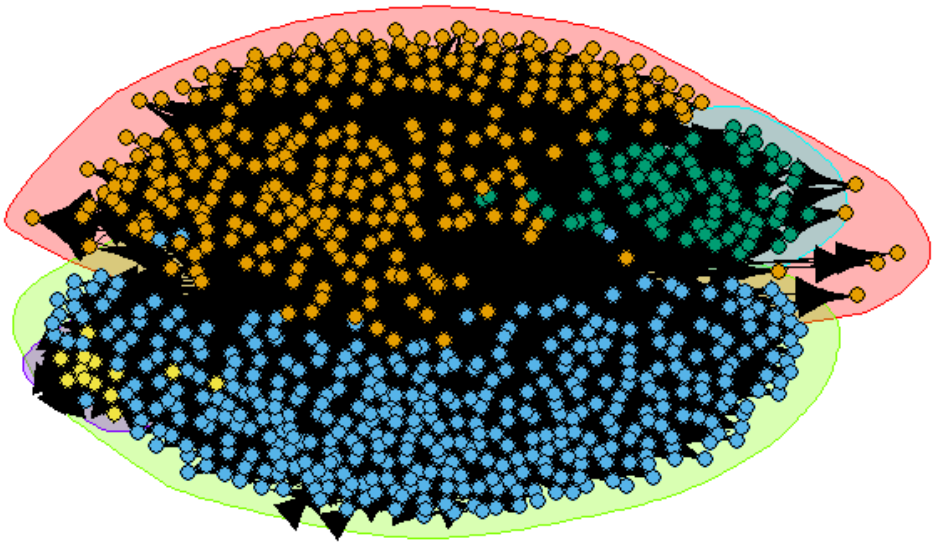
Observation

We observe that the in-degree for all three graphs have left hugging distributions, in the sense that the number of nodes with low indegree are high, but as indegree increases, the number of nodes with that indegree decreases gradually.

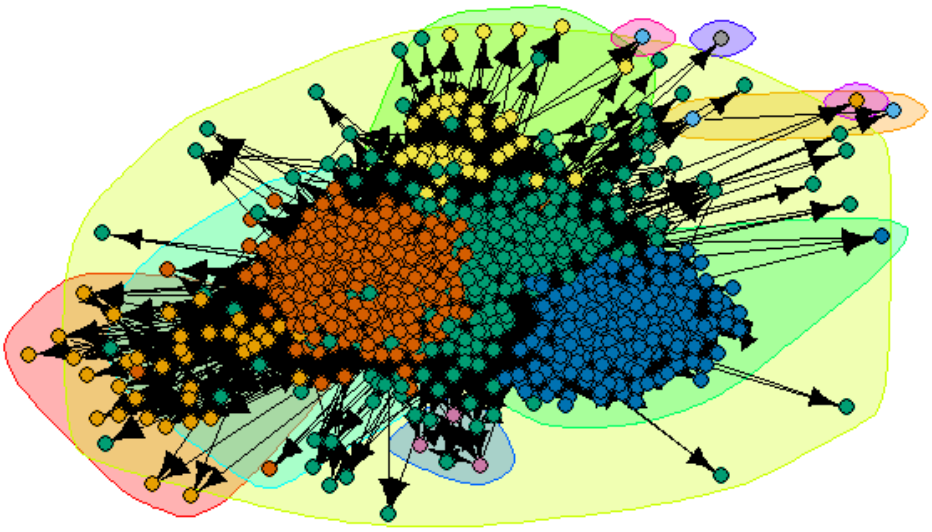
A similar trend is observed in outdegree for all 3 except that the decrease is not so gradual. Considering this, yes it appears that the distributions are similar amongst the three nodes.

We see from the above plots that it follows power law.
.

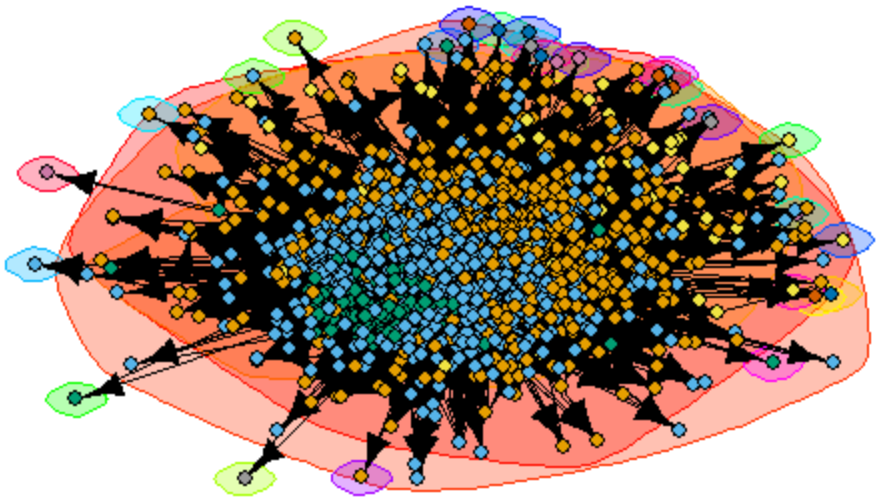
Question 20: For the 3 personal networks picked in question 19, extract the community structure of each personal network using Walktrap community detection algorithm. Report the modularity scores and plot the communities using colors. Are the modularity scores similar? In this question, you should have 3 plots.



Community plot for 109327480479767108490
Modularity for Node 1 : 0.252755



Community plot for 115625564993990145546
Modularity for Node 2 : 0.3194733



Community plot for 101373961279443806744
Modularity : 0.1910928

Observation

Considering that modularity ranges from -0.5 to 1, the differences between the maximum and minimum modularity (around 0.12) can be seen significant. But between other pairs we see a difference of only about 0.06 which is relatively less. All three modularities seem to lie in the mid range (considering possible values range from -0.5 to 1).

Modularity is the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. Among the three networks, the walk_trap algorithm seems to work the best for **115625564993990145546** , as it has the highest modularity score. This means that , compared to the other two nodes, the intra_community density is high and inter-communities edges are relatively few. Amongst the three **101373961279443806744** has the lowest modularity. This is expected as the number of vertices is very high and so is the

number of edges, making it harder for the algorithm to divide it into highly partitioned communities.

Question 21: Based on the expression for h and c, explain the meaning of homogeneity and completeness in words.

Homogeneity is representative of the percentages of the same community members lying in the same circle. So, if more members within a community map to the same circle, homogeneity would increase.
Completeness, on the other hand, is representative of the percentages of the same circle members lying in the same community. So, if more members within a circle are put in the same community by the algorithm, completeness would increase.
Both Homogeneity and Completeness see their maximum value in a case when the communities formed by the algorithms are exactly same as the circles defined by the user.

Question 22: Compute the h and c values for the community structures of the 3 personal network (same nodes as question 19). Interpret the values and provide a detailed explanation.

Node ID	**109327480479767108490**	**115625564993990145546**	**101373961279443806744**
H(C)	1.050779	8.46514	0.38431
H(K)	1.005208	1.081190	0.49333
H(C|K)	0.1556361	4.639828	0.38283
H(K|C)	0.6736162	4.783148	1.235417
Homogeneity	0.8518851	0.451890	0.003866
Completeness	0.3298739	-3.423962	-1.504238

We see that homogeneity decreases as network size increases and the same is true for completeness. This makes sense, because the smaller the circles and communities, the larger chance that nodes from the same circle fall into the same communities and vice versa.

An extremely low homogeneity value is observed for Node **101373961279443806744** because of its huge size and high number of connections.

Negative completeness is only observed because while in terms of communities a node could only be in one community, the same is not true for circles ; for e.g a friend from UCLA can be in

two circles like “UCLA” and “Friends”. Because of this, nodes in multiple circles, when getting mapped to communities, end up punishing the score more than rewarding (because for correct guess there is only one reward but multiple punishment for every other circle it is in).