EE232E PROJECT 2

SOCIAL NETWORK MINING

Devanshi Patel	504945601	devanshipatel@cs.ucla.edu
Ekta Malkan	504945210	emalkan@cs.ucla.edu
Pratiksha Kap	704944610	pratikshakap@cs.ucla.edu
Sneha Shankar	404946026	snehashankar@cs.ucla.edu

INTRODUCTION

The growth of social media over the last decade has revolutionized the way individuals interact and industries conduct business. Individuals are producing data at an unprecedented rate by interacting, sharing, and consuming content through social media. Understanding and processing this new type of data to collect actionable patterns presents challenges and opportunities for interdisciplinary research, novel algorithms, and tool development. Social Media Mining integrates social media, social network analysis, and data mining to provide a convenient and coherent platform for students, practitioners, researchers, and project managers to understand the basics and potentials of social media. It introduces unique problems arising from social media data and presents fundamental concepts, emerging issues, and effective algorithms for network analysis and data mining. [1]

In this project, we study two social networks Facebook and Google plus. For facebook dataset,we consider an undirected graph. This means that once a friendship is established between two individuals, an undirected link is created in the graph between the user nodes. For GPlus, we consider a directed network. This implies that when a person A follows person B, a directed link is established between the two individuals from A to B. No lnk is present for B to A though.

Analyzing these two datasets reveals interesting information about the community structures of the networks, as well as helps us to identify core nodes in the network. These core nodes are important people, people who influence large number of other nodes. A crucial task in the analysis of online social-networking systems is to identify important people — those linked by strong social ties—within an individual's network neighborhood.

We will also study the properties of **embeddedness** and **dispersion** in an individual's personalized network. 'Embeddedness' is a measure of tie strength and can be used as a baseline predictor. `Dispersion' is the extent to which two people's mutual friends are not themselves well-connected. We also study the properties of **Homogeneity** and **Completeness** for Directed Networks.

DataSet

The Facebook Dataset found here contains data of 4039 users with 88339 friendship links.The dataset includes node features (anonymized user profiles), circles (friendship-lists) , and ego networks.
The GPlus Dataset found here contains data of 107614 user nodes with 13673453 links. The dataset includes node features (profiles), circles, and ego networks.

We will now analyze the properties of the Facebook Social Network as follows:

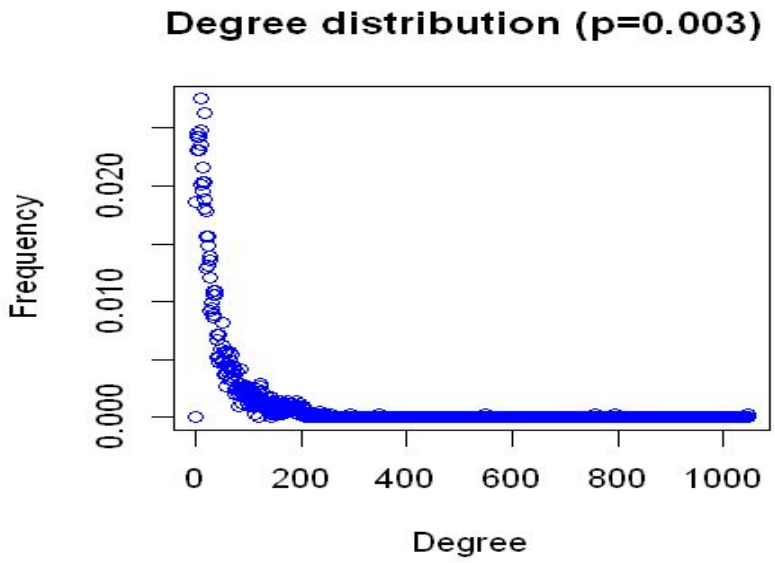
Question 1: Is the facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

Answer : Yes the graph is connected.

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

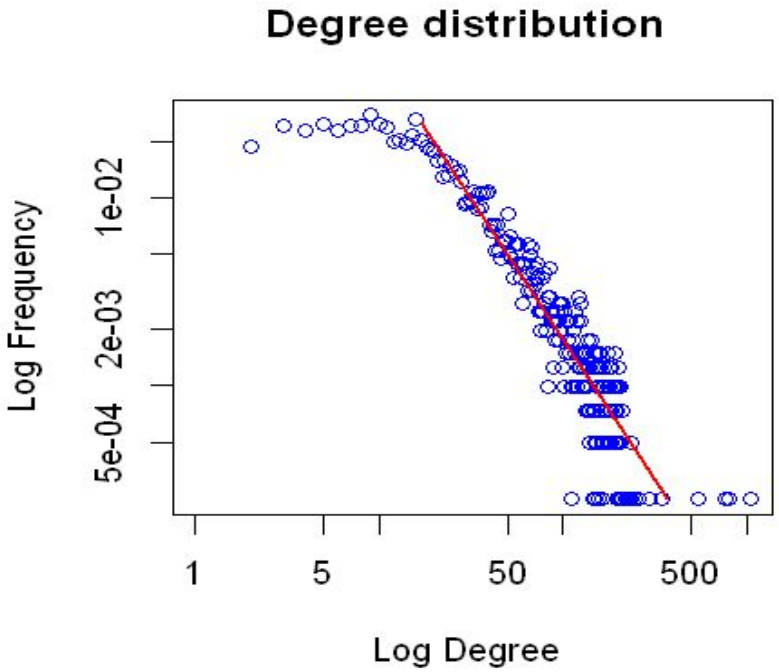
Answer : The diameter of the graph is **8.**

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



Average degree : **43.69**

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.



The Slope of the line is :

-1.1802

We found this slope using the lm function . It gives the intercept and the slope as follows:

(Intercept) log10(d)
-0.4309 -1.1802

PERSONALIZED NETWORK

A personalized network is basically a subgraph from the whole graph consisting of one particular node and all its neighbors. Analysis of personalized network of a node can lead us to interesting insights about the importance and influence that the node carries over the network. The most influential nodes are called as core nodes. For Facebook dataset, core nodes are the nodes which have more than 200 neighbors.

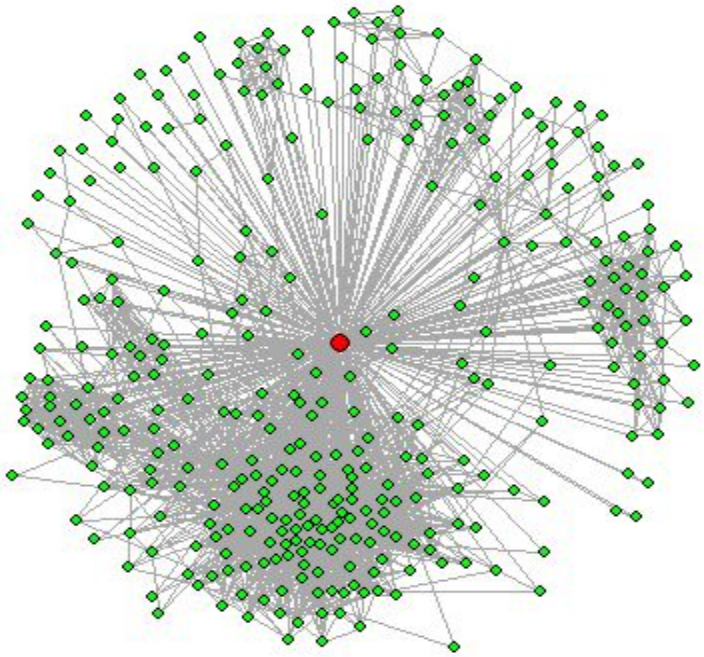
Question 5: Create a personalized network of the user whose ID is 1. How many nodes and edges does this personalized network have?

Answer :

Number of nodes in personal network **348**

Number of edges in personal network **2866**

The personal network for node 1 can be visualized as follows :



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.

Diameter of this personal network 2
Upper Bound : 2
Lower Bound : 1

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?

Answer : The diameter of a graph is the maximum eccentricity of any vertex in the graph. That is, it is the greatest distance between any pair of vertices. To find the diameter of a graph, first find the shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph. The diameter of the personalized network to be equal to the upper bound i.e. 2 means that the shortest path between any 2 vertices of this network is 2. This implies that the personalized network is not fully connected.

However, the meaning of the lower bound being 1 means that in the event that the network is fully connected, there would exist a direct edge between any two vertices. In this scenario the

diameter of the network is 1. This would mean that every neighbor of the core node in the personalized network is connected to every other neighbor of the core node.

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

We found that there are **40 core nodes** in the Facebook Network.
The average degree of the 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.
Answer:
In the study of complex networks, a network is said to have **community structure** if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.
Community structures are quite common in real networks. Social networks like Facebook and GPlus include community groups based on common location, interests, occupation, etc.

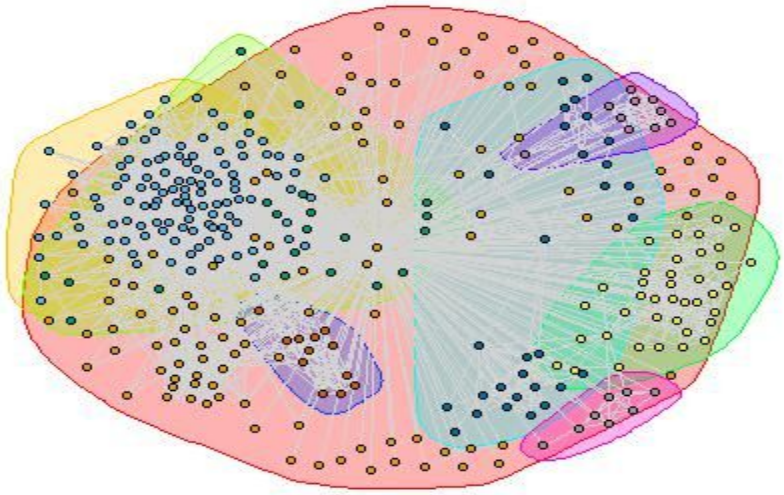
Fastgreedy is a Community Detection Algorithm that tries to optimize a quality function called modularity in a greedy manner. Initially, every vertex belongs to a separate community, and communities are merged iteratively such that each merge is locally optimal (i.e. yields the largest increase in the current value of modularity). The algorithm stops when it is not possible to increase the modularity any more, so it gives you a grouping as well as a dendrogram. The method is fast and it is the method that is usually tried as a first approximation because it has no parameters to tune. However, it is known to suffer from a resolution limit, i.e. communities below a given size threshold (depending on the number of nodes and edges if I remember correctly) will always be merged with neighboring communities.

The Girvan–Newman algorithm detects communities by progressively removing edges from the original network. The connected components of the remaining network are the communities. Instead of trying to construct a measure that tells us which edges are the most central to communities, the Girvan–Newman algorithm focuses on edges that are most likely "between" communities. According to the characteristics of edge betweenness in the complex network, if the betweenness of an edge is relative lower, a pair of nodes connected by that edge should be in the same community.

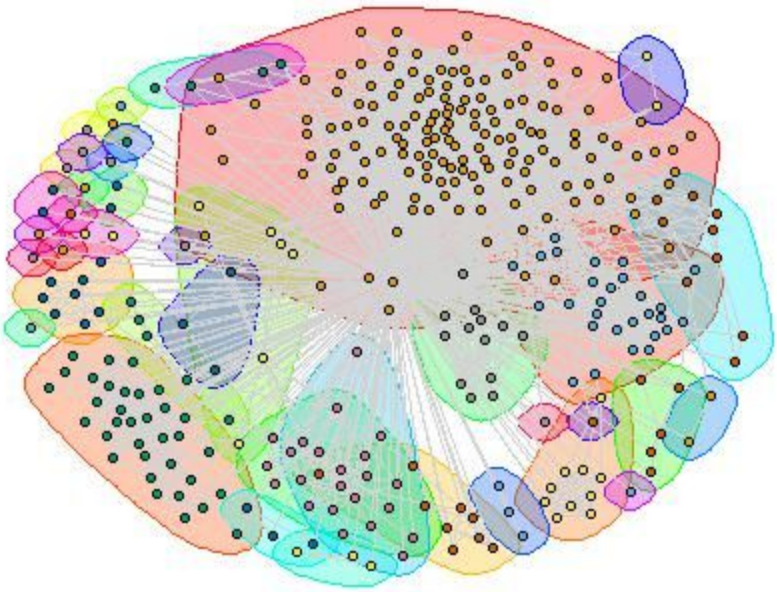
Infomap community detection algorithm is based on information theoretic principles; it tries to build a grouping which provides the shortest description length for a random walk on the graph, where the description length is measured by the expected number of bits per vertex required to encode the path of a random walk.

We now use each of these Community Detection algorithms for the 5 core nodes and compare based on the Modularity score. The higher the Modularity score, the better, as it means there is more grouping between the nodes and the communities formed are well separated.

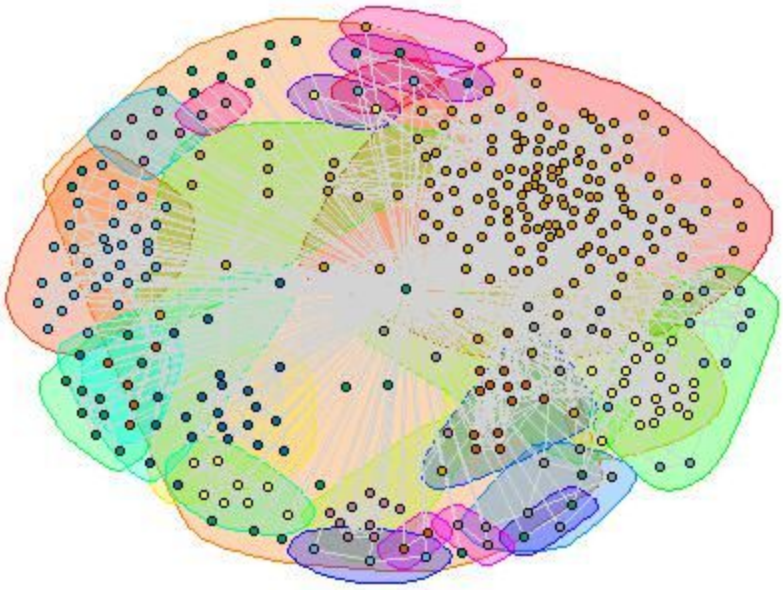
**********For node : 1 **********
Fastgreedy Modularity is 0.4131014



[1] "_____"
Edge betweenness Modularity is 0.3533022

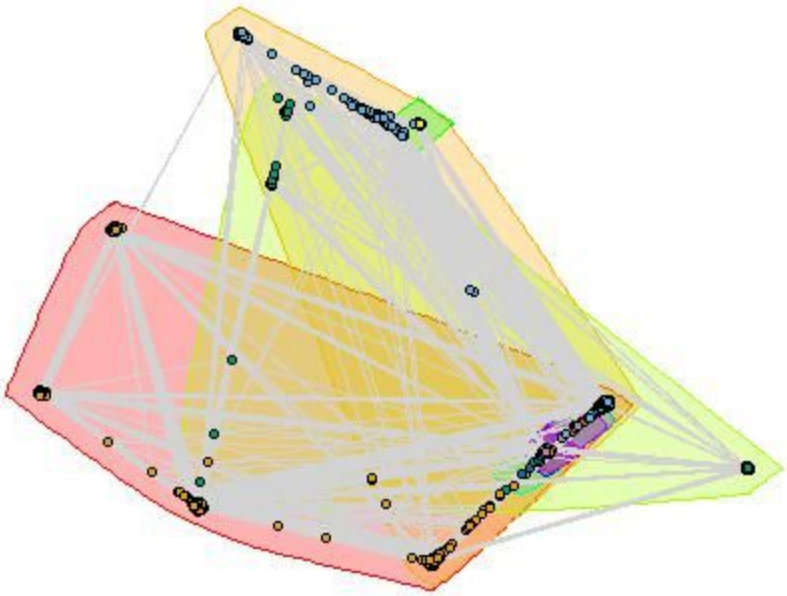


[1] "_____"
Infomap Modularity is 0.3891185



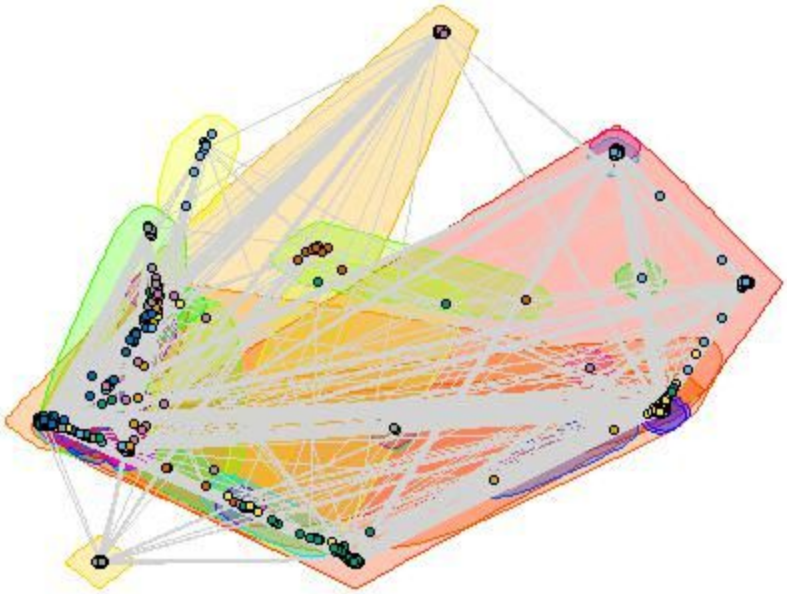
**********For node : 108 **********

Fastgreedy Modularity is 0.4359294

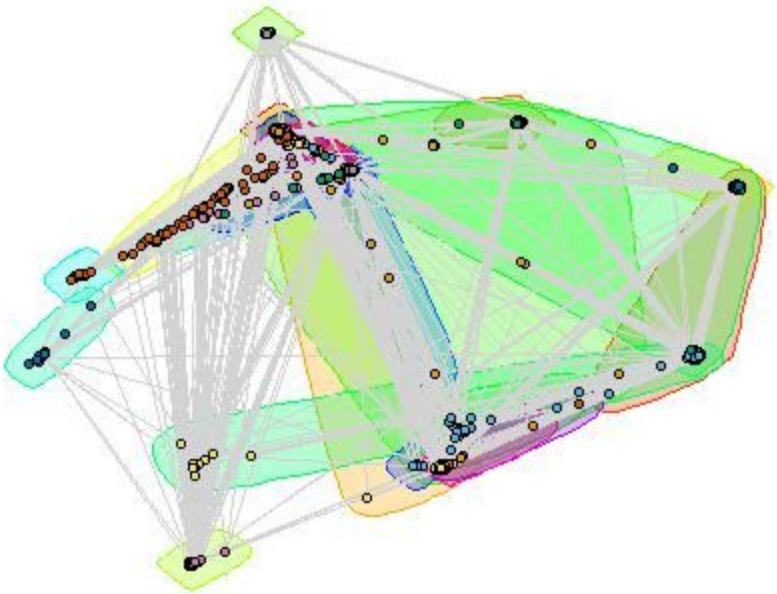


[1] "_____"

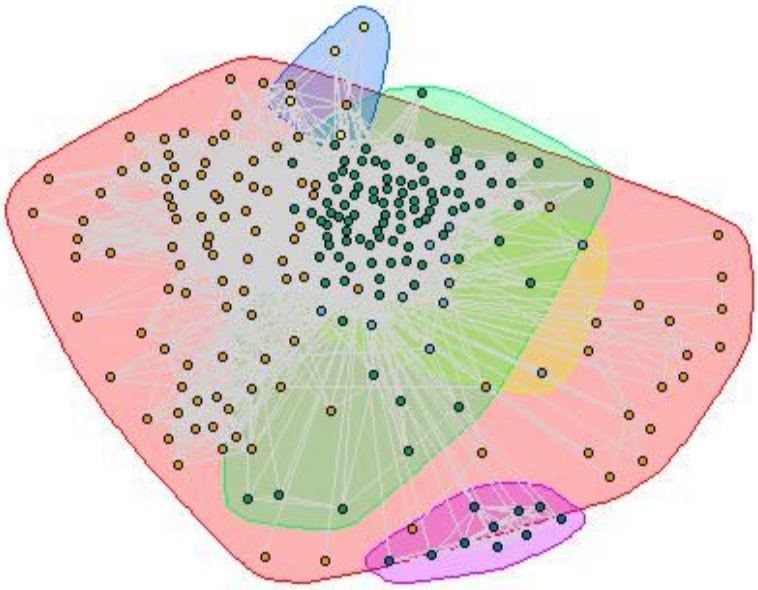
Edge betweenness Modularity is 0.5067549



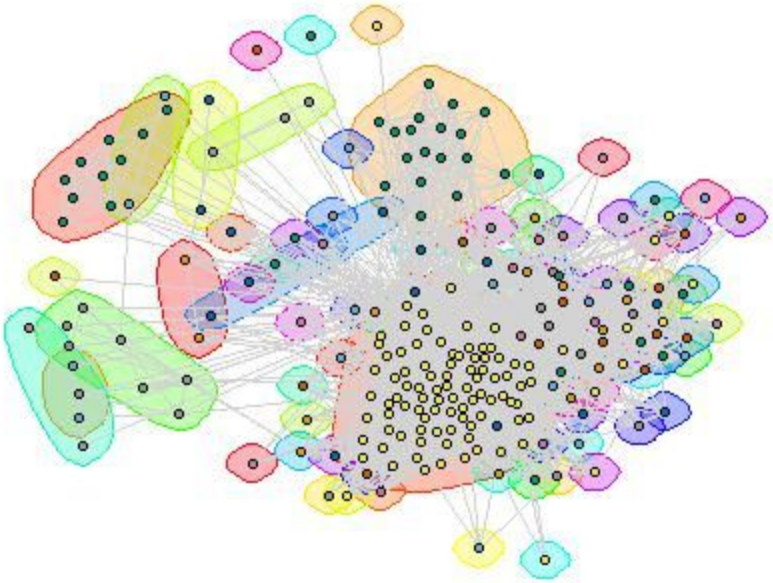
[1] "_____"
Infomap Modularity is 0.5082233



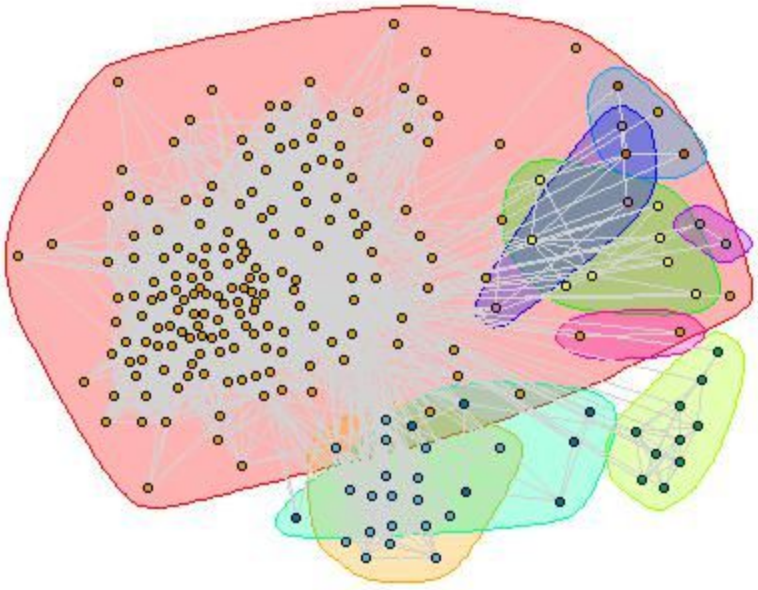
**********For node : 349 **********
Fastgreedy Modularity is 0.2517149



[1] "_____"
Edge betweenness Modularity is 0.133528

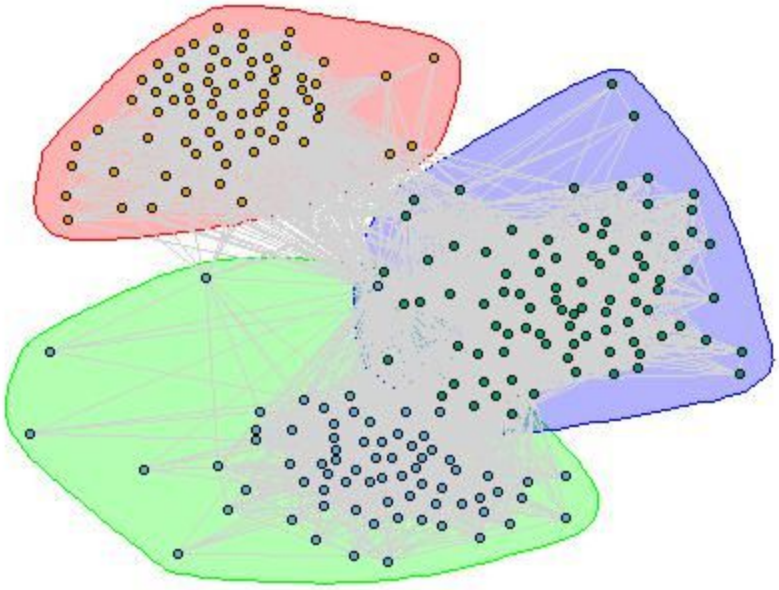


[1] "_____"
Infomap Modularity is 0.0954642



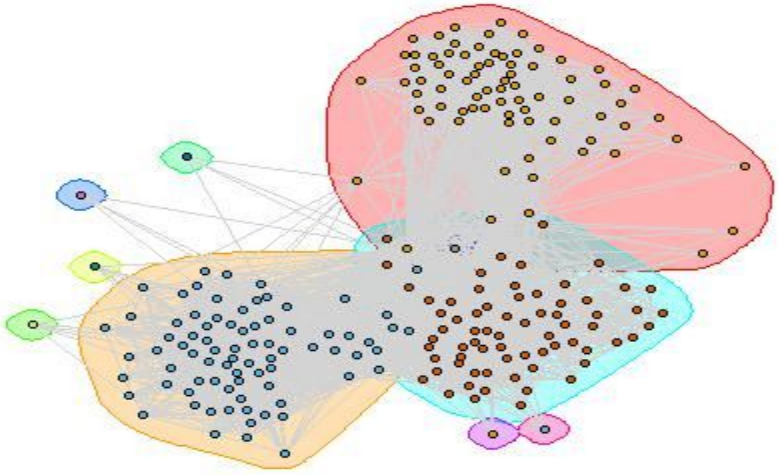
**********For node : 484 **********

Fastgreedy Modularity is 0.5070016

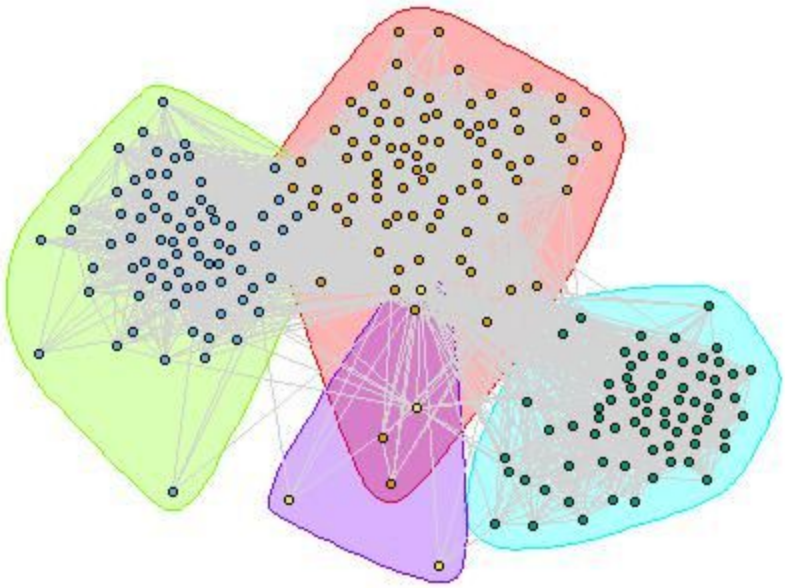


[1] "_____"

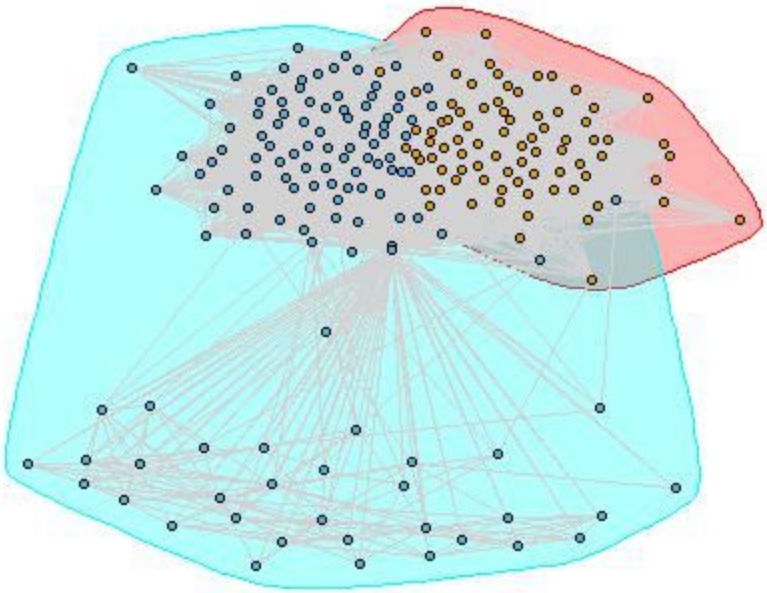
Edge betweenness Modularity is 0.4890952



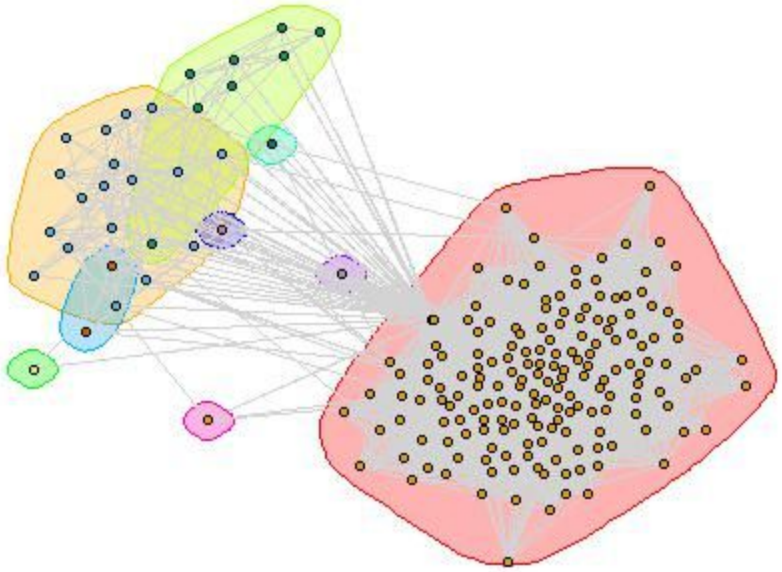
[1] "_____"
Infomap Modularity is 0.5152788



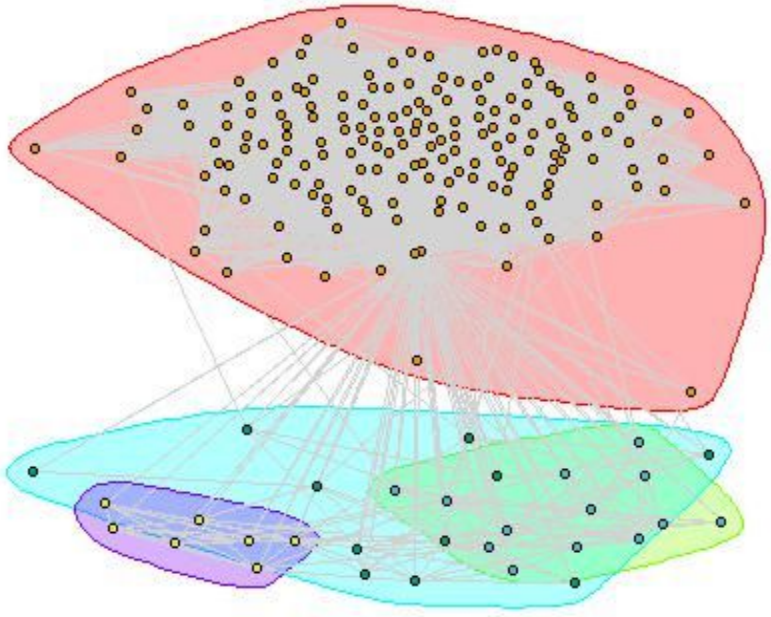
**********For node : 1087 **********
Fastgreedy Modularity is 0.1455315



[1] "_____"
Edge betweenness Modularity is 0.02762377



[1] "_____"
Infomap Modularity is 0.02651616

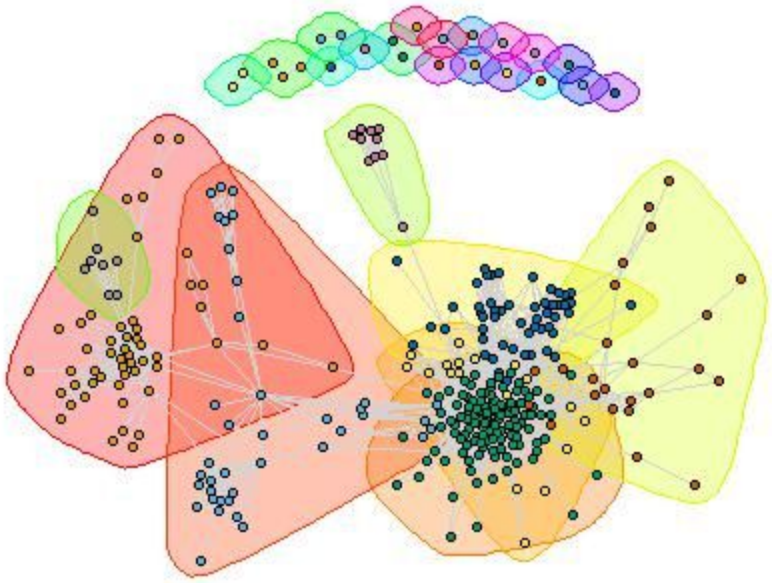


Question 10: For each of the core node's personalized network(use same core nodes as question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as question 9. Compare the modularity 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.

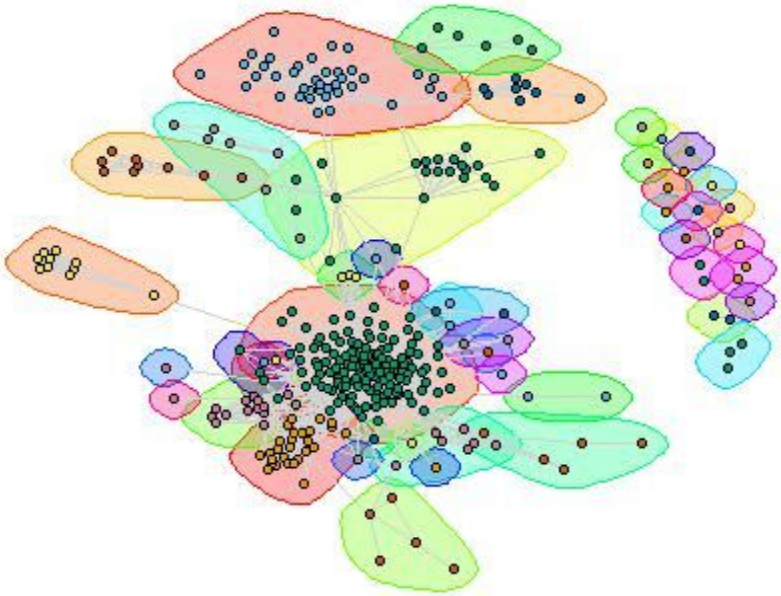
Answer: In the previous question, we found out the modularity for a core node's personalized network. Now in the same personalized network, we remove the core node and analyze the community structure of this network.

After removing the core node For node : 1

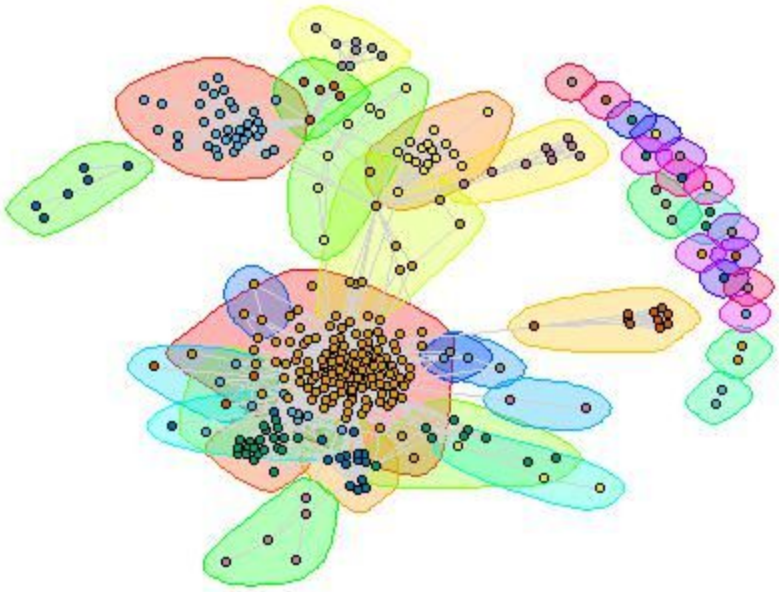
Fastgreedy Modularity is 0.4418533



[1] "_____"
Edge betweenness Modularity is 0.4161461

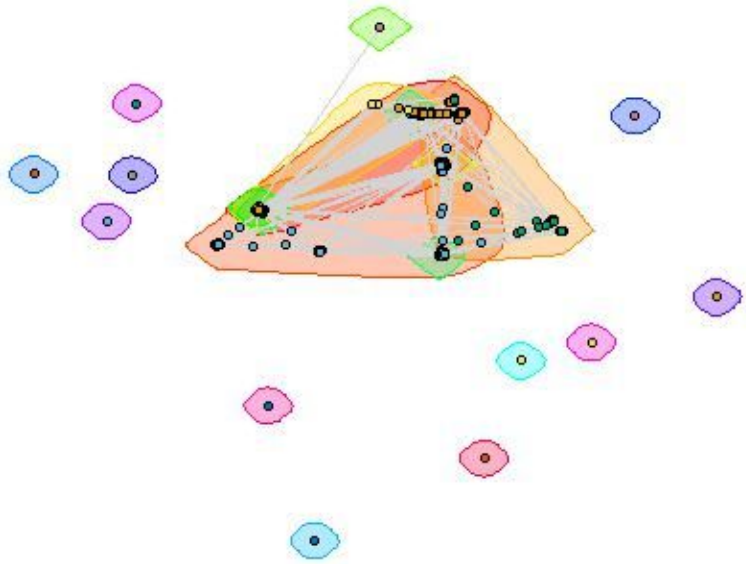


[1] "_____"
Infomap Modularity is 0.4180077



After removing the core node For node : 108

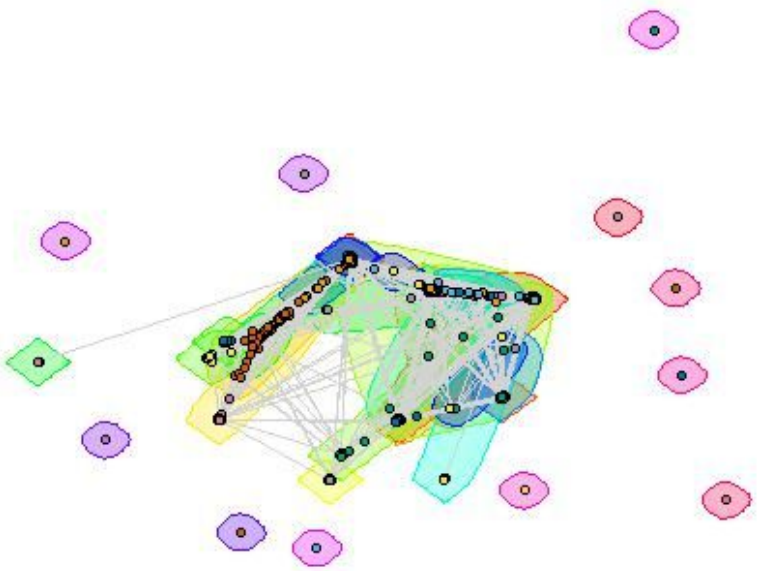
Fastgreedy Modularity is 0.4581271



[1] "_____"
Edge betweenness Modularity is 0.5213216

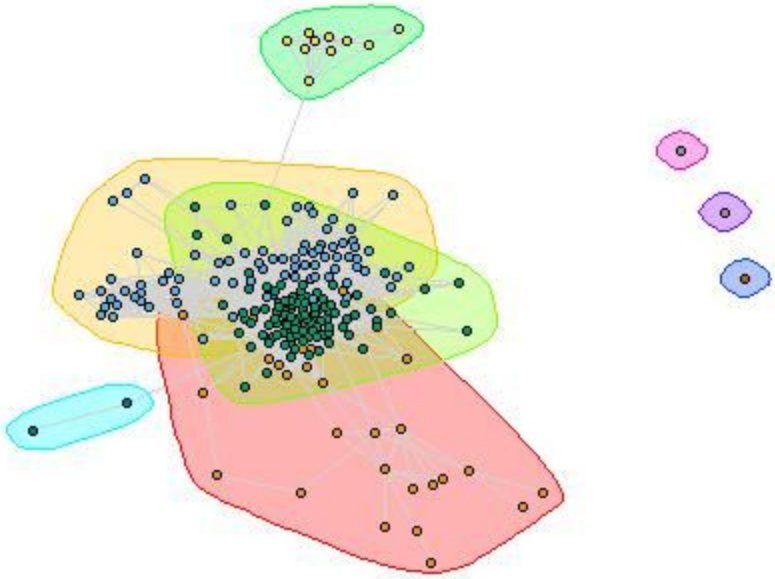
[1] "_____"

Infomap Modularity is 0.5205171



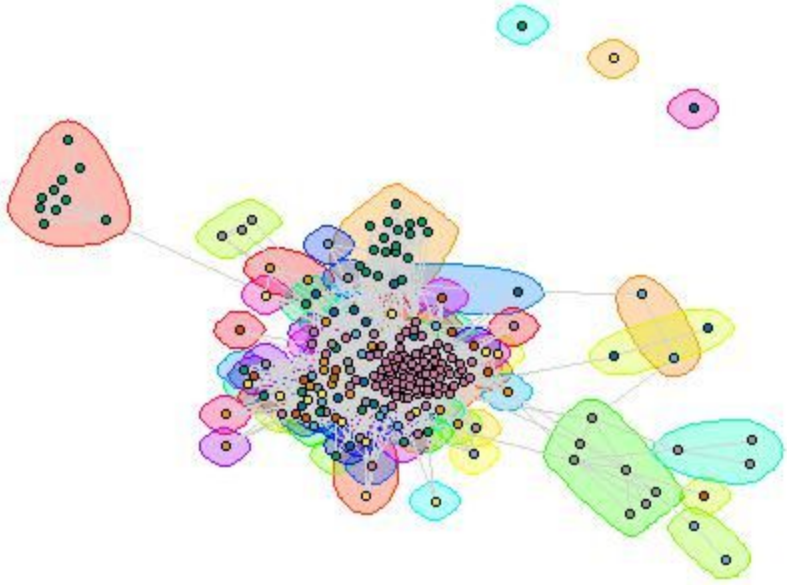
After removing the core node For node : 349

Fastgreedy Modularity is 0.2456918

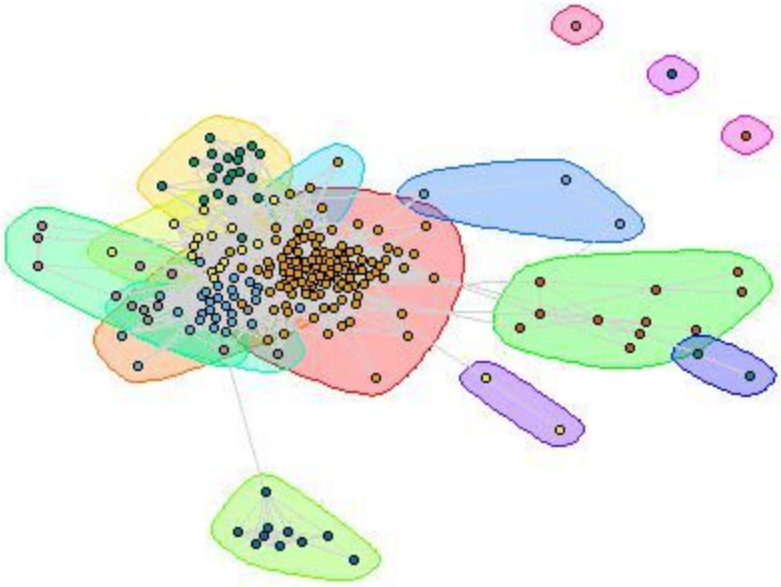


[1] "_____"

Edge betweenness Modularity is 0.1505663

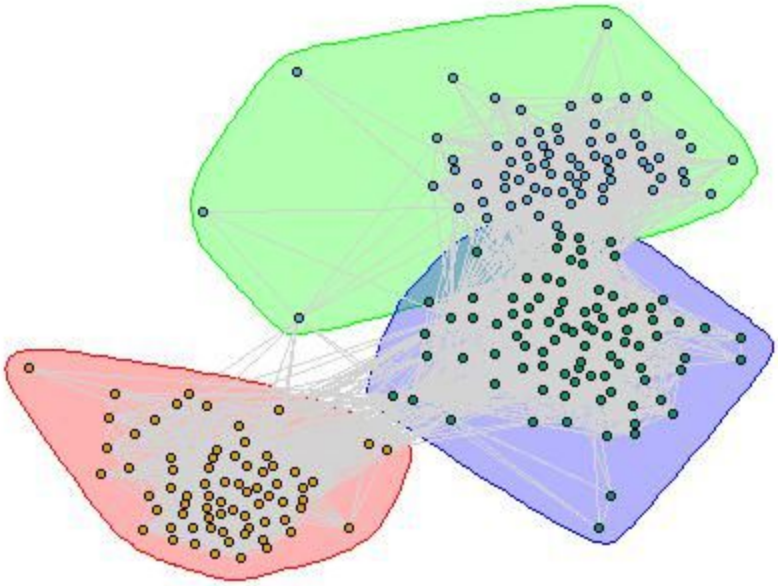


[1] "_____"
Infomap Modularity is 0.2448156



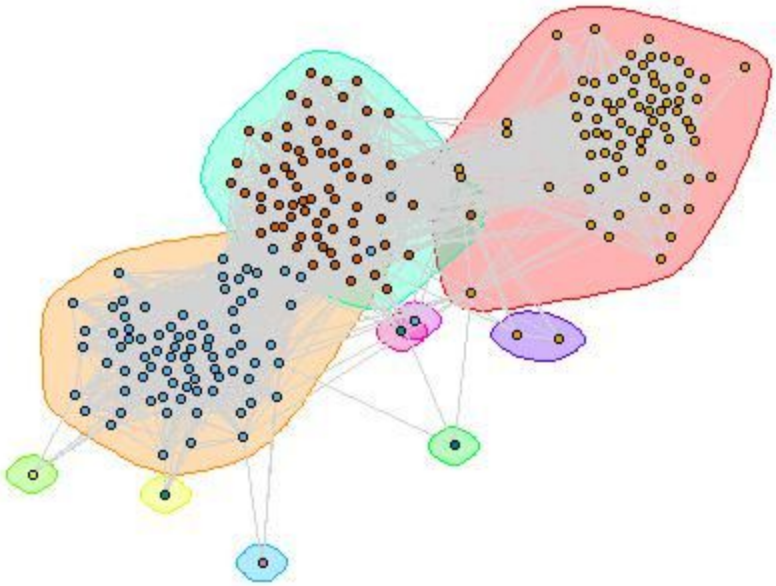
After removing the core node For node : 484

Fastgreedy Modularity is 0.5342142

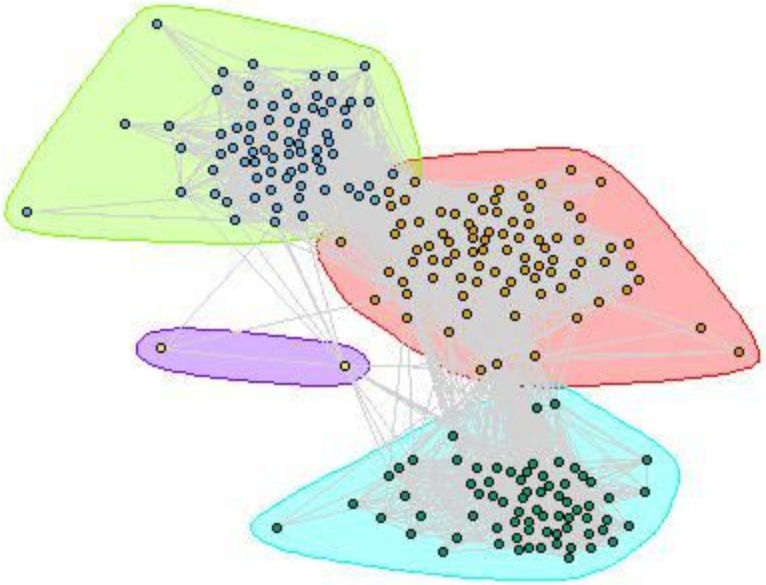


[1] "_____"

Edge betweenness Modularity is 0.5154413

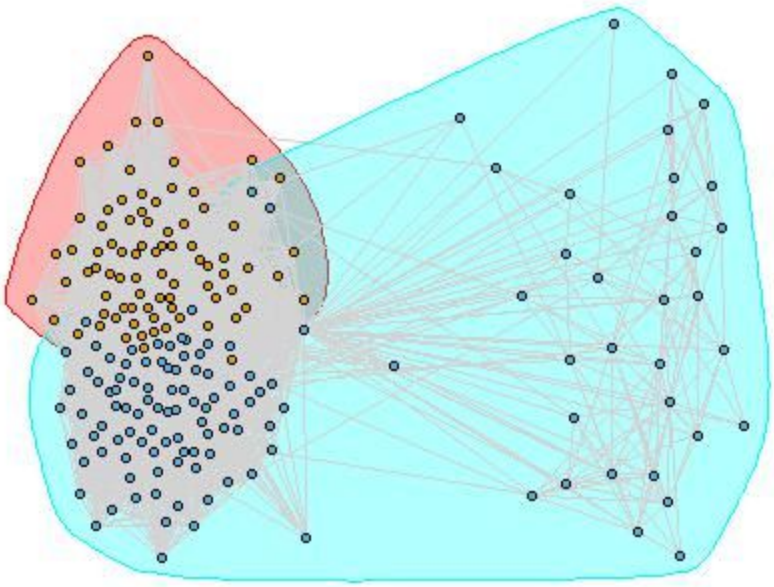


[1] "_____"
Infomap Modularity is 0.5434437

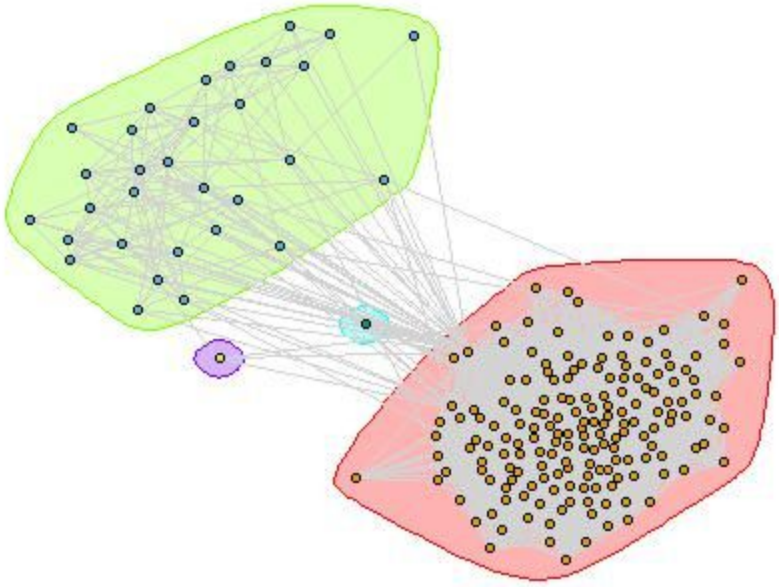


After removing the core node For node : 1087

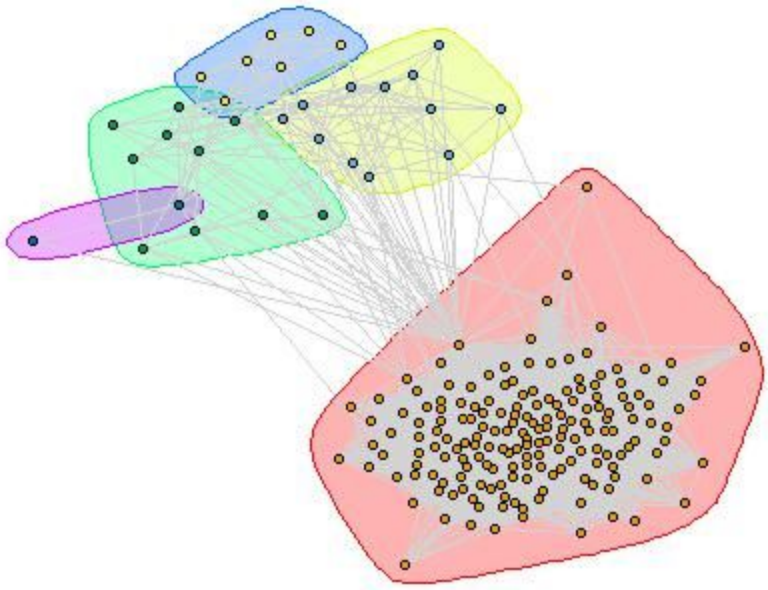
Fastgreedy Modularity is 0.1481956



[1] "_____"
Edge betweenness Modularity is 0.0324953



[1] "_____"
Infomap Modularity is 0.02737159



The modularity scores for each of the plots for Question 9 and 10 is summarized as below :

	Modularity			Modularity(after core node deleted)			
Core Node ID	Fast Greedy	Edge Betweenness	InfoMap	Fast Greedy	Edge Betweenness	InfoMap	Max Modularity
1	0.4131	0.3533	0.3891	0.4419	0.4161	0.418	0.4419
108	0.4359	0.5068	0.5082	0.4581	0.5213	0.5205	0.5213
349	0.2503	0.1335	0.0954	0.2457	0.1506	0.2448	0.2503
484	0.1	0.0169	0	0.5342	0.5154	0.5434	0.5434
1087	0.1104	0.0072	0	0.1482	0.0325	0.0274	0.1482

Analysis :

We observe that **after the core node is deleted from the network, the modularity increases**.

This is because the node node was the main connecting component of the personalized network, as it was connected to each vertex of the network. The other neighbours of the personalized network may or may not be fully connected to each other, hence the process of identification of a community by classifying nodes into one of the communities becomes each after core-node deletion. This increases modularity as can be seen from the above results.

Also, we observe that the highest overall modularity was obtained for **Fast Greedy Algorithm** as compared to the other two algorithms.

Question 11: Write an expression relating the Embeddedness of a node to its degree.

Upon plotting for embeddedness of a node with respect to its degree, a linear relationship was observed between the two. As Embeddedness is defined as the number of mutual friends shared by a node with the core node, nodes with higher degree have better chances of having more mutual friends with the core node.

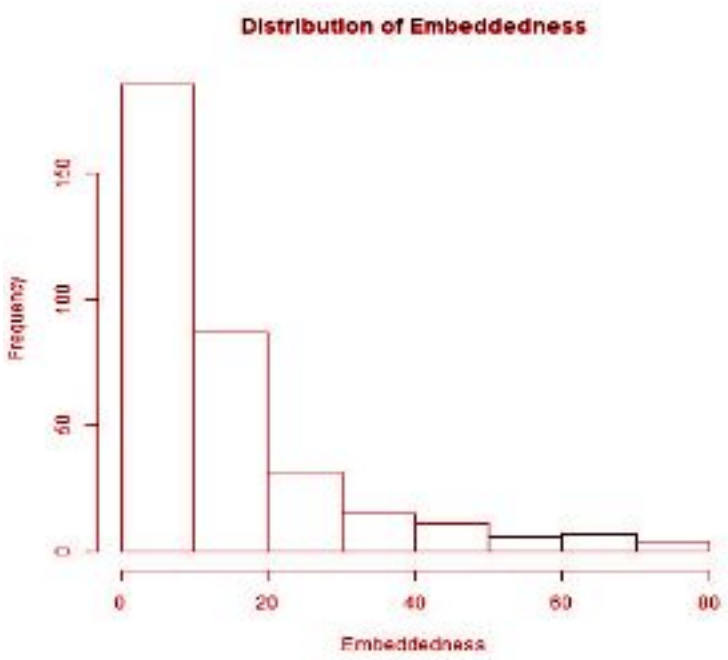
The following expression denotes the relationship between embeddedness of a node to its degree:

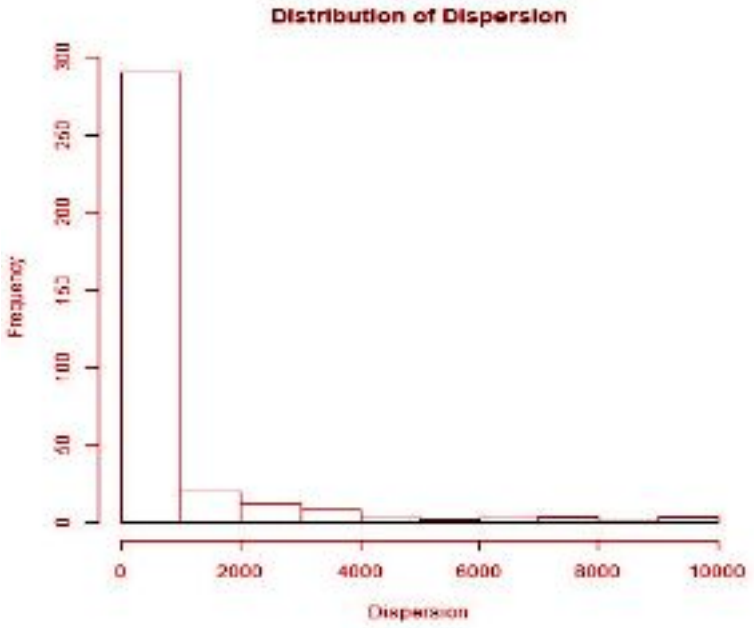
$$\textbf{Embeddedness} = D_i - N_i - 1$$

where Di denotes the degree of the node in the original network and Ni denotes the number of neighbors of node i that do not belong to the personalized network. In other words, Ni denotes the number of non-mutual friends of node i with core node.

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.

- **Node ID 1**

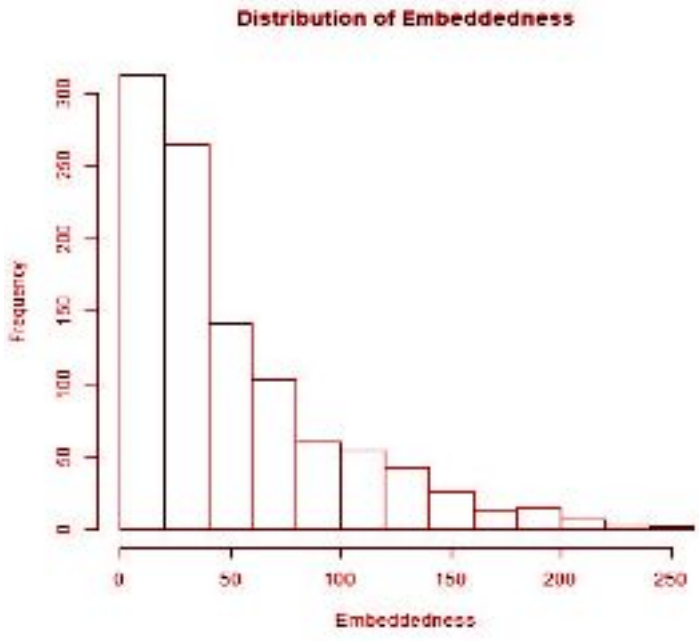


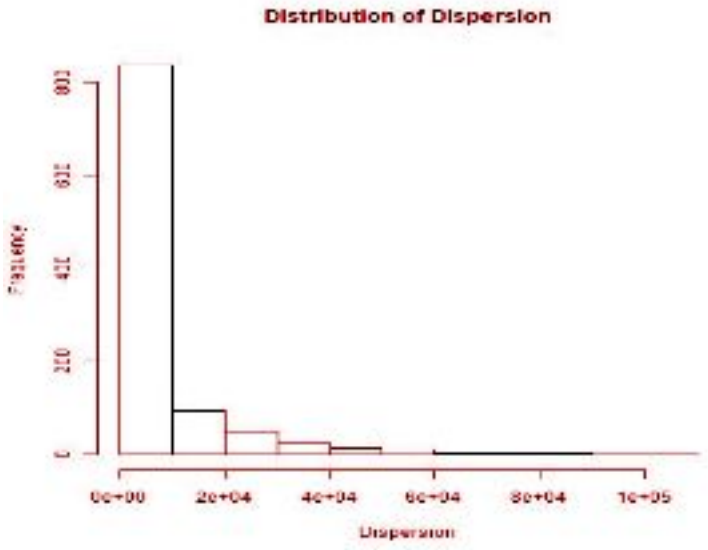


Observations:

- ➔ From the distribution of embeddedness, it can be observed that majority of the nodes have very low values of embeddedness and very few of them have higher values.
- ➔ The highest value of embeddedness observed in the personalized network of Node 1 is 80.
- ➔ The plot of dispersion shows that the majority of nodes have dispersion values less than 1000 and the maximum value of dispersion is 10,000.

- **Node ID 108**

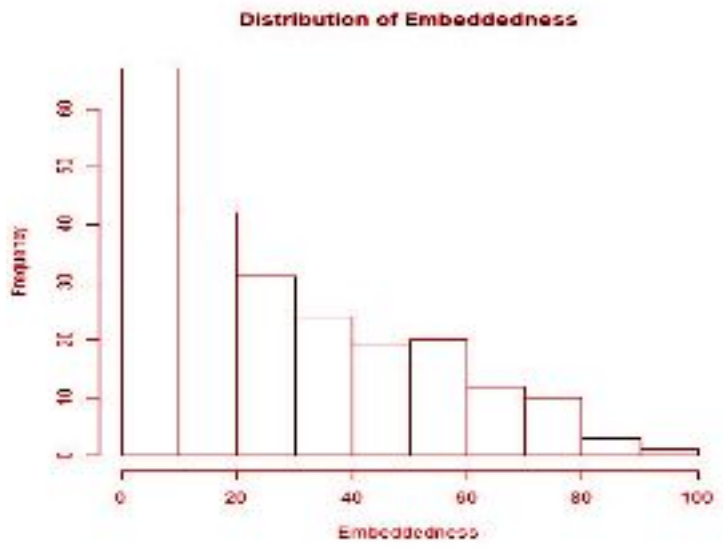


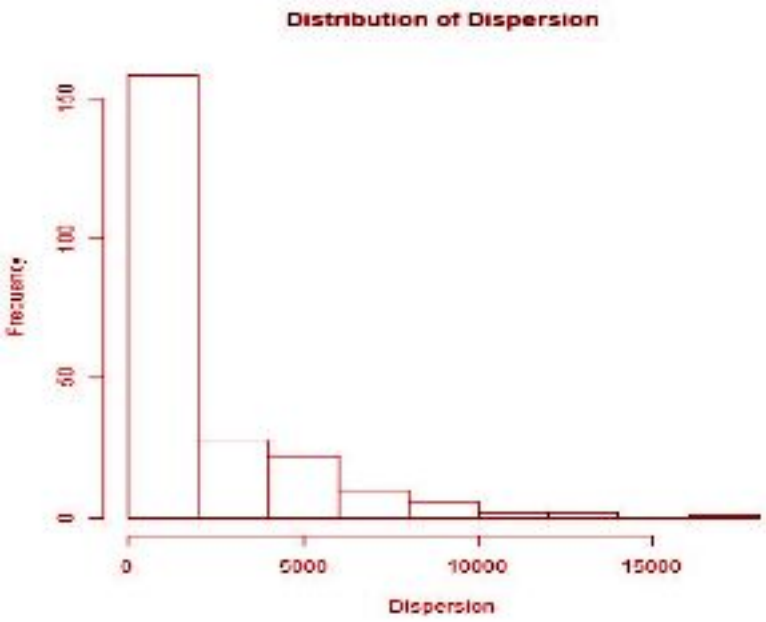


Observations:

- ➔ From the distribution of embeddedness, it can be observed that majority of the nodes have very low values of embeddedness and very few of them have higher values. The distribution is quite similar to exponential distribution.
- ➔ The highest value of embeddedness observed in the personalized network of Node 108 is 250.
- ➔ There are a large number of nodes in this personalized network and this justifies the reasoning for observing high values of embeddedness.
- ➔ The plot of dispersion shows that the majority of nodes have dispersion values less than 1000 and the maximum value of dispersion is 9000.

- **Node ID 349**

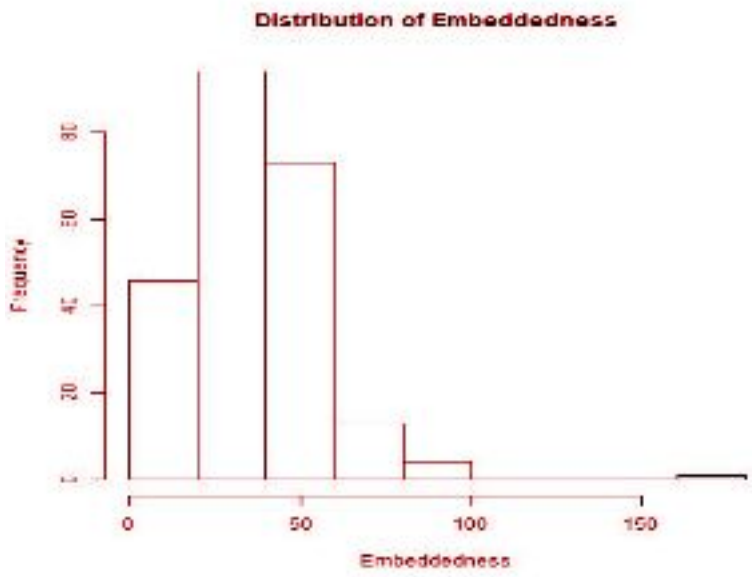


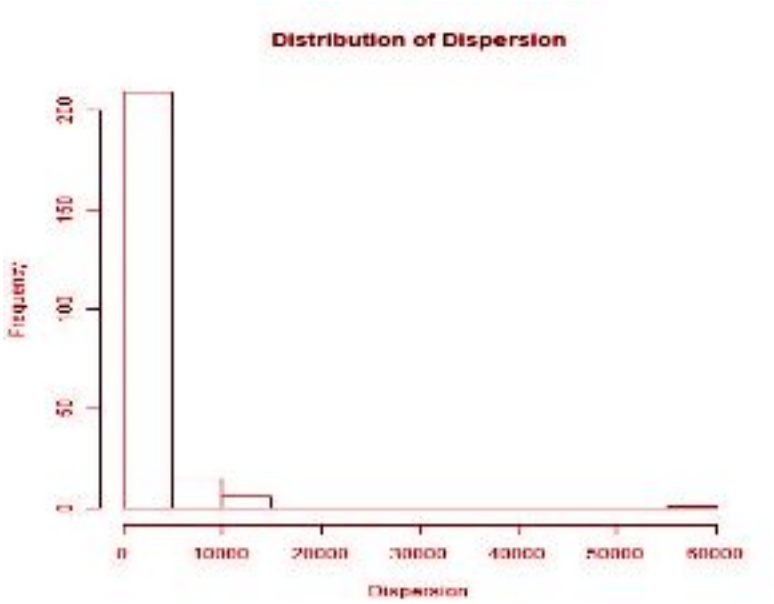


Observations:

- ➔ From the distribution of embeddedness, it can be observed that the nature is similar to monotonically decreasing plot. Also, this network has much less number of nodes compared to other networks.
- ➔ The highest value of embeddedness observed in the personalized network of Node 349 is 100.
- ➔ The plot of dispersion shows that the majority of nodes have dispersion values less than 2000 and the maximum value of dispersion is 15,000.

- **Node ID 484**

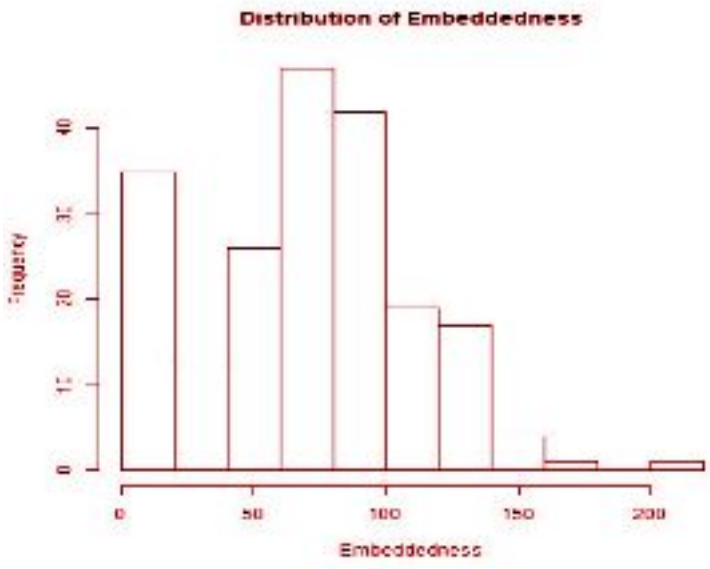


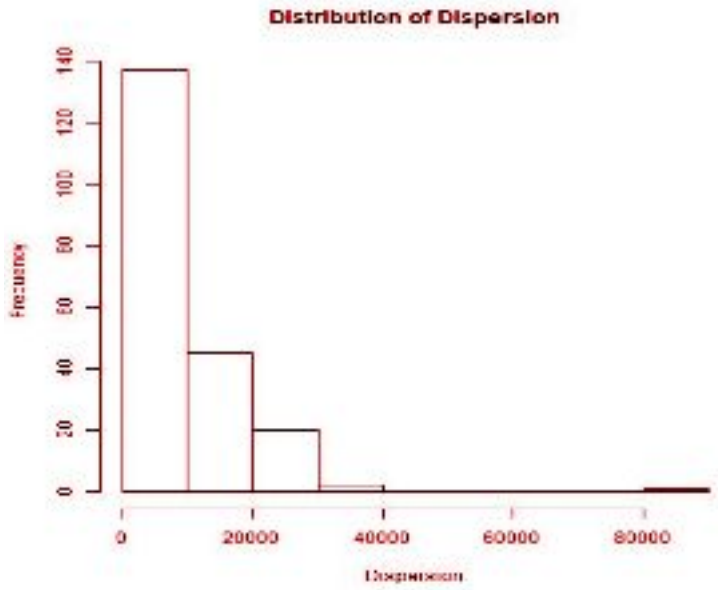


Observations:

- ➔ From the distribution of embeddedness, it can be observed that the distribution is very different from the ones we observed previously. Majority of nodes have embeddedness in the range of 20 to 60.
- ➔ The highest value of embeddedness observed in the personalized network of Node 484 is 150.
- ➔ The plot of dispersion shows that the majority of nodes have dispersion values less than 5000 and the maximum value of dispersion is 60,000.

- **Node ID 1087**





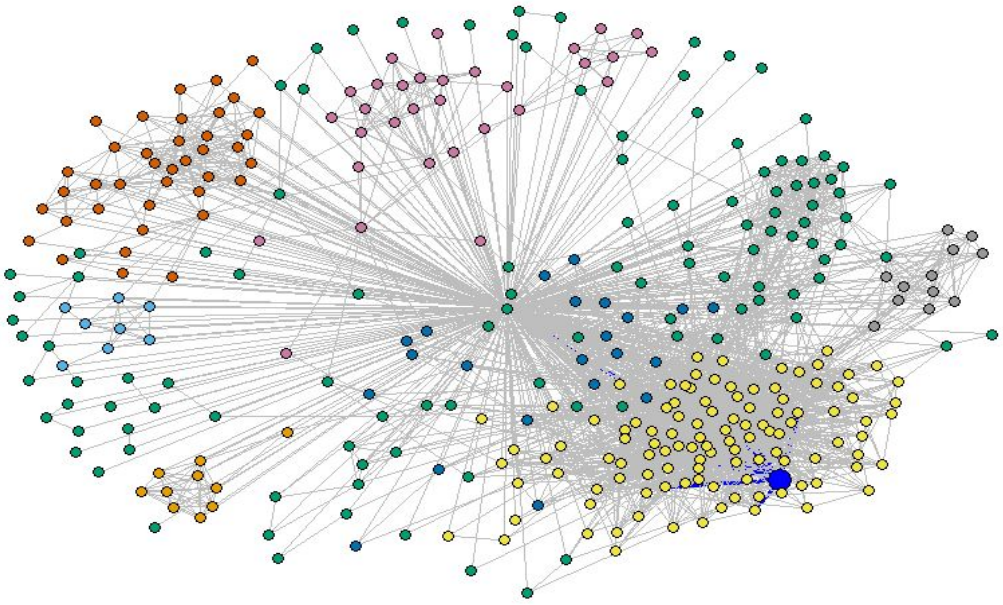
Observations:

- ➔ From the distribution of embeddedness, it can be observed that the distribution is very different from the ones we observed previously. This network has the least number of nodes out of all 5 core nodes.
- ➔ Majority of nodes have embeddedness in the range of 60 to 100.
- ➔ The highest value of embeddedness observed in the personalized network of Node 1087 is 200.
- ➔ The plot of dispersion shows that the majority of nodes have dispersion values less than 10,000 and the maximum value of dispersion is 80,000.
- ➔ This node's personalized network gives larger values for dispersion compared to any other networks.

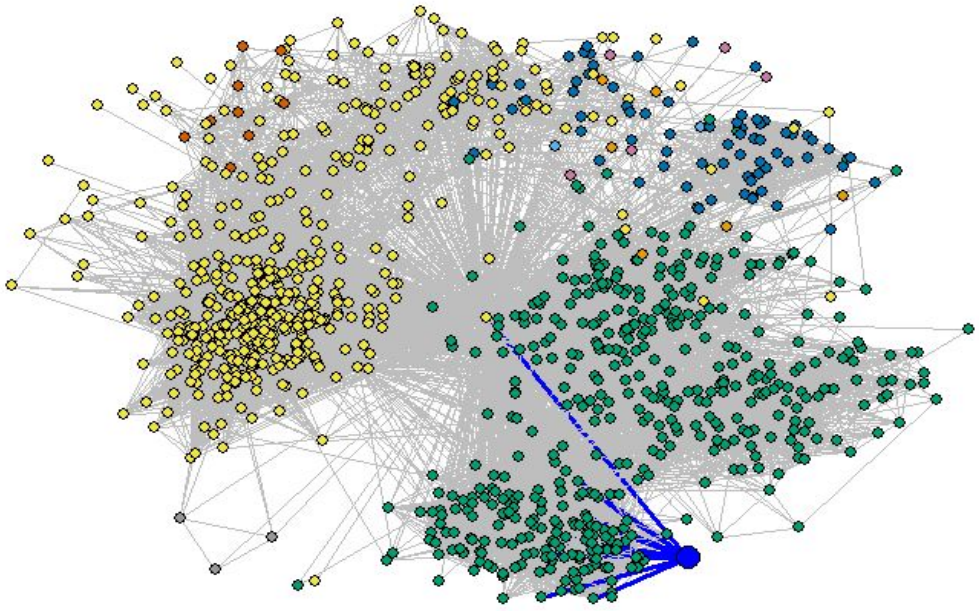
Question 13: For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node. To detect the community structure, use Fast-Greedy algorithm. In this question,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

In the plots below, we show the community structure of the personalized network of each core node. The nodes belonging to same cluster has been depicted by same colors. We have highlighted the node with maximum dispersion and its incident edges by "blue" color.

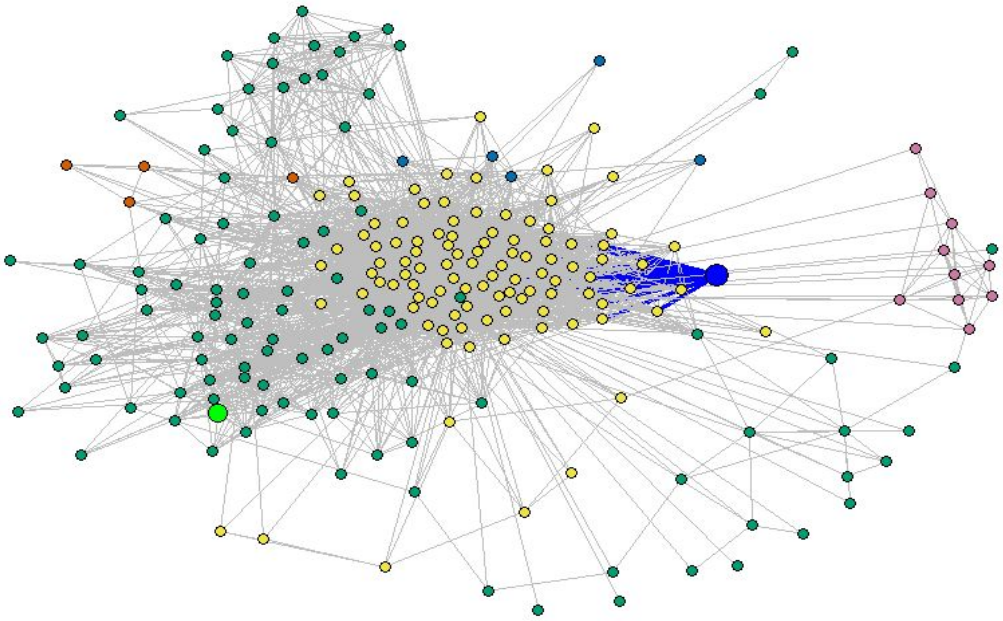
- **Node ID 1**



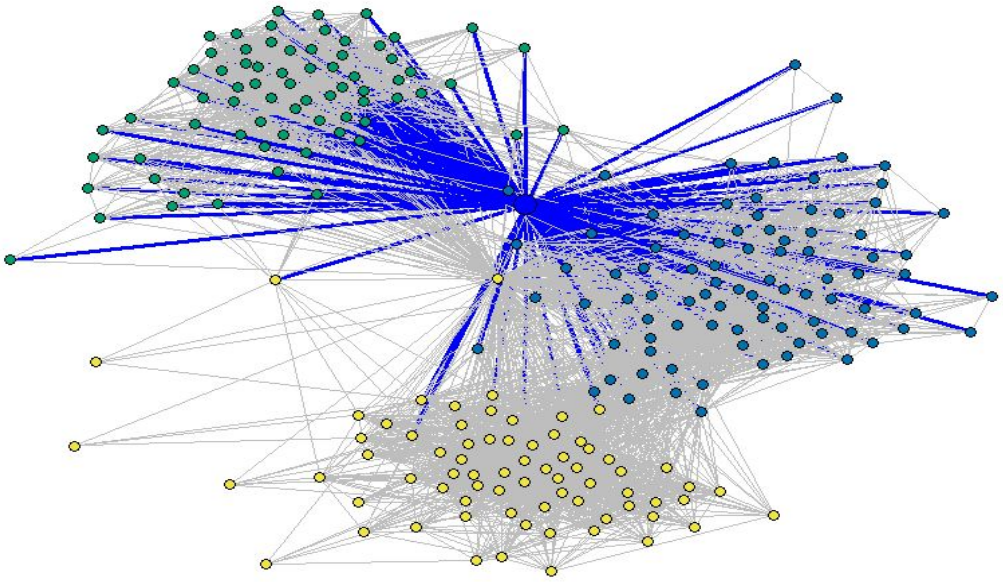
- **Node ID 108**



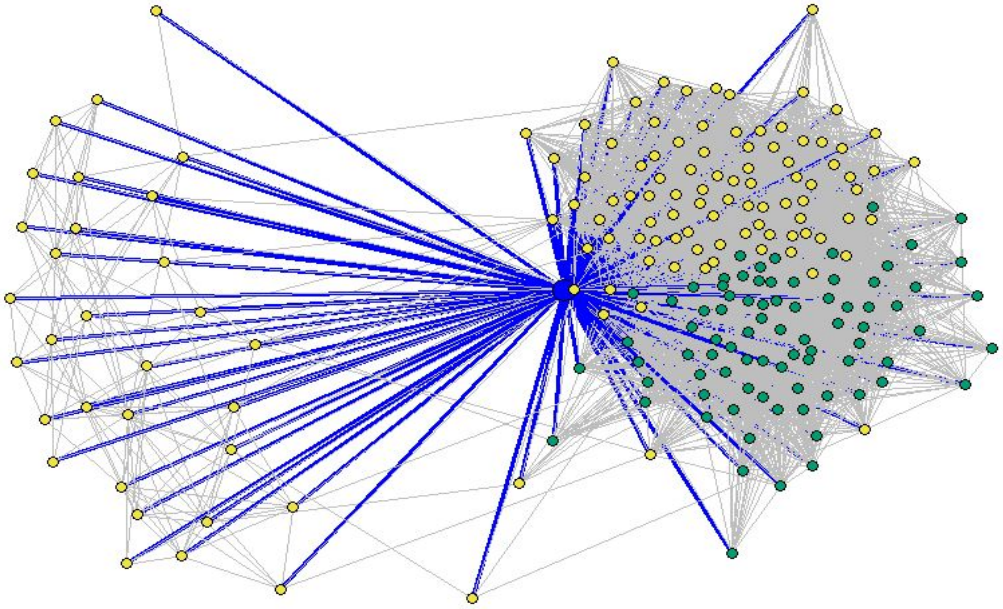
- **Node ID 349**



- **Node ID 484**



- **Node ID 1087**



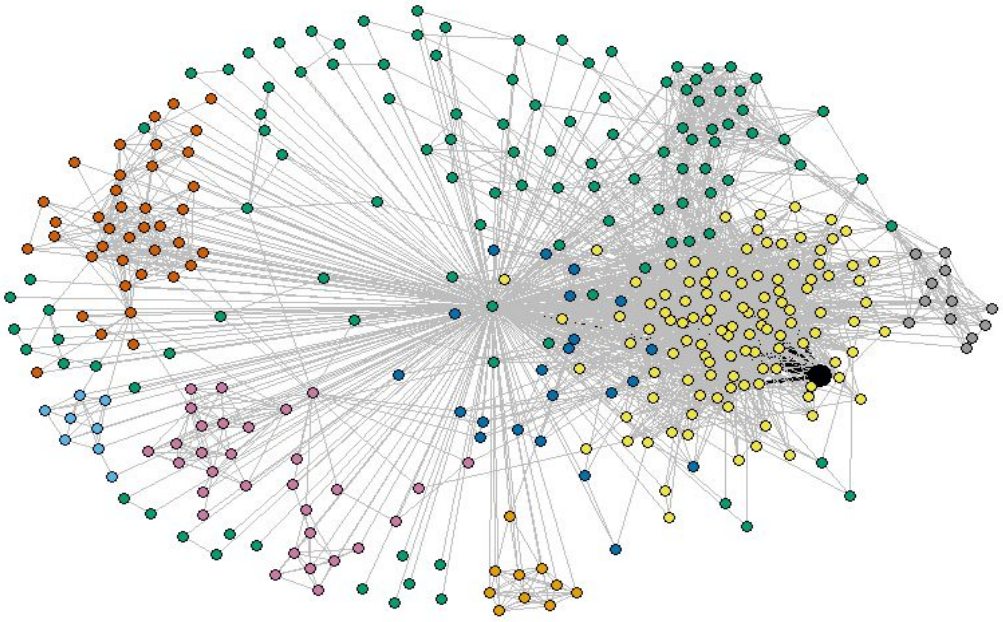
Question 14: Repeat question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion/embeddedness. Also, highlight the edges incident to these nodes.

In the plots below, we show the community structure of the personalized network of each core node. The nodes belonging to same cluster has been depicted by same colors. We have highlighted the node with maximum embeddedness and its incident edges by “black” color.

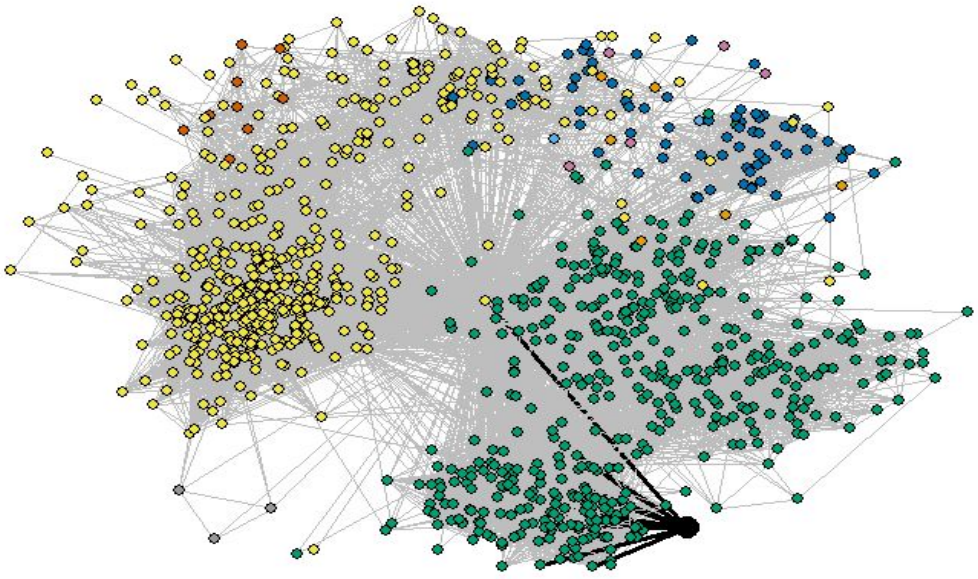
We also calculated the ratio of dispersion and embeddedness for all nodes. The node with the maximum value of this ratio and its incident edges have been highlighted by “red” color.

----------Maximum Embeddedness----------

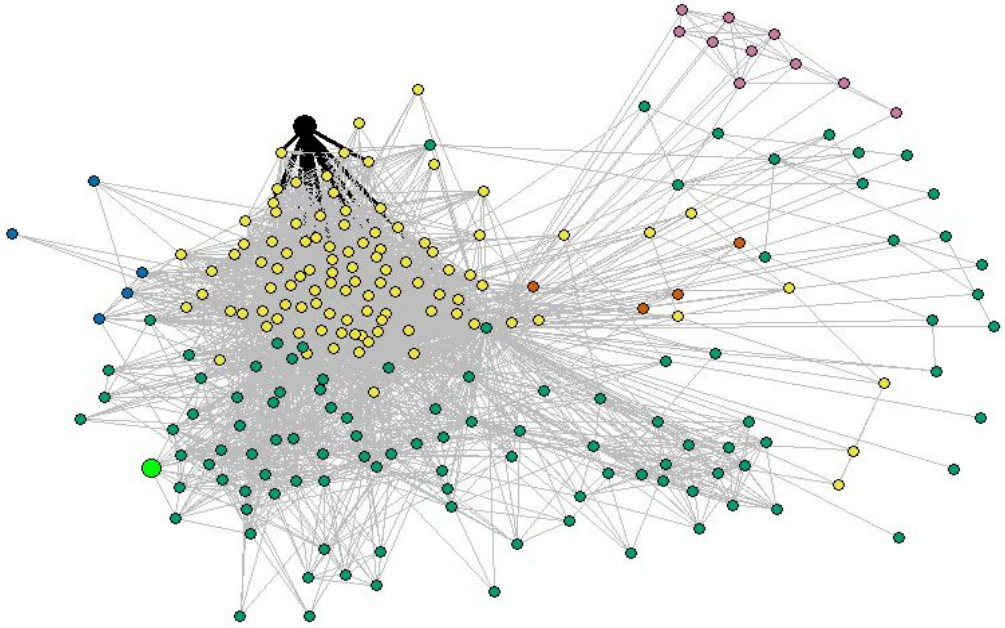
- **Node ID 1**



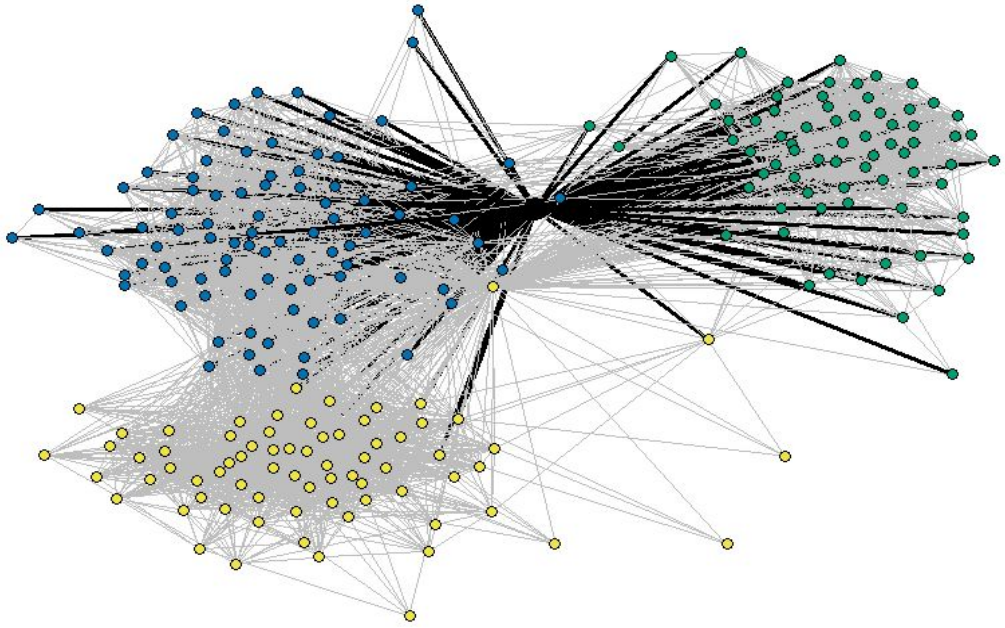
- **Node ID 108**



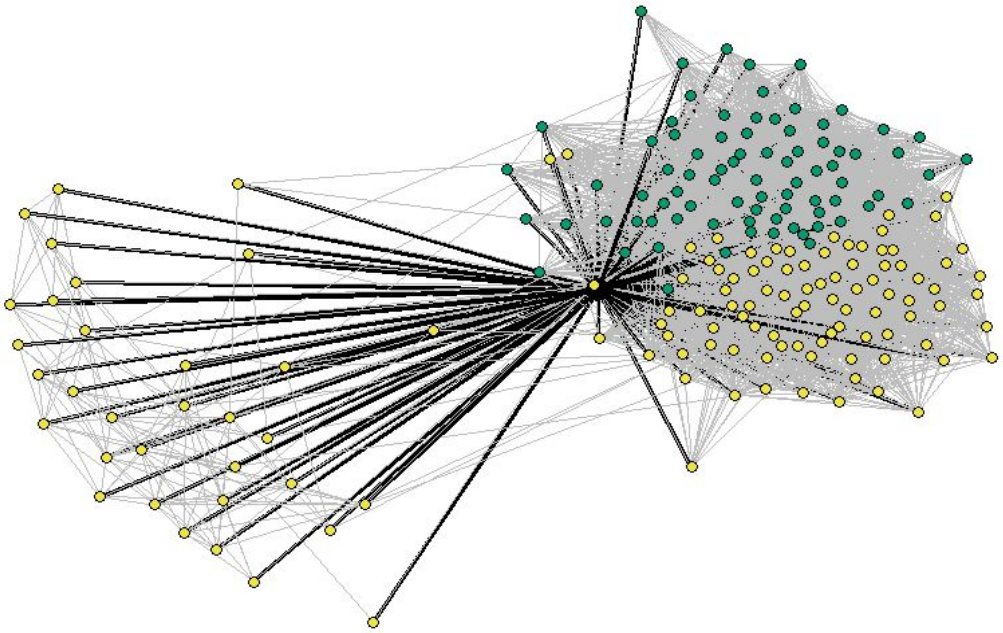
- **Node ID 349**



- **Node ID 484**

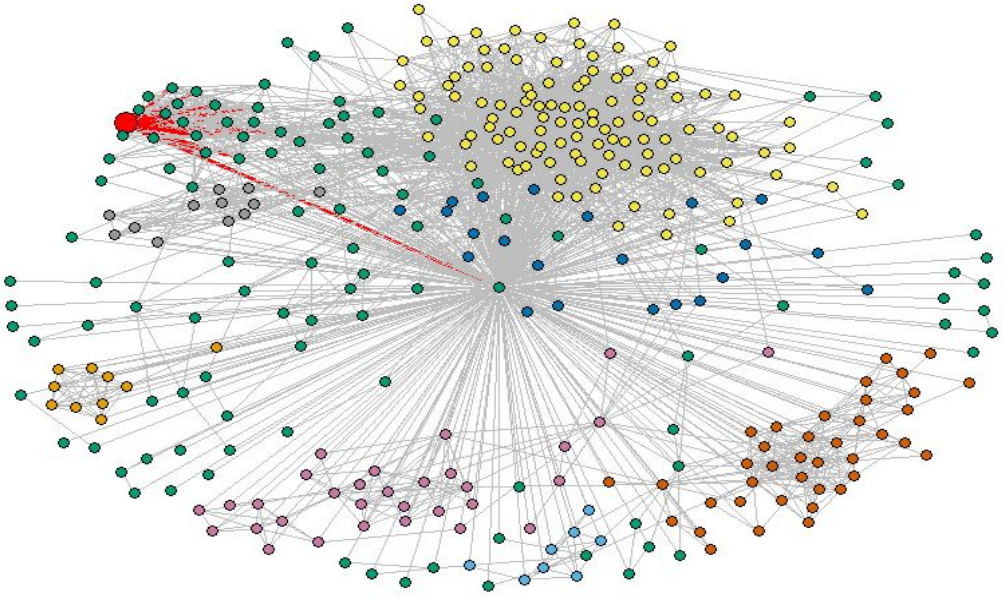


- **Node ID 1087**

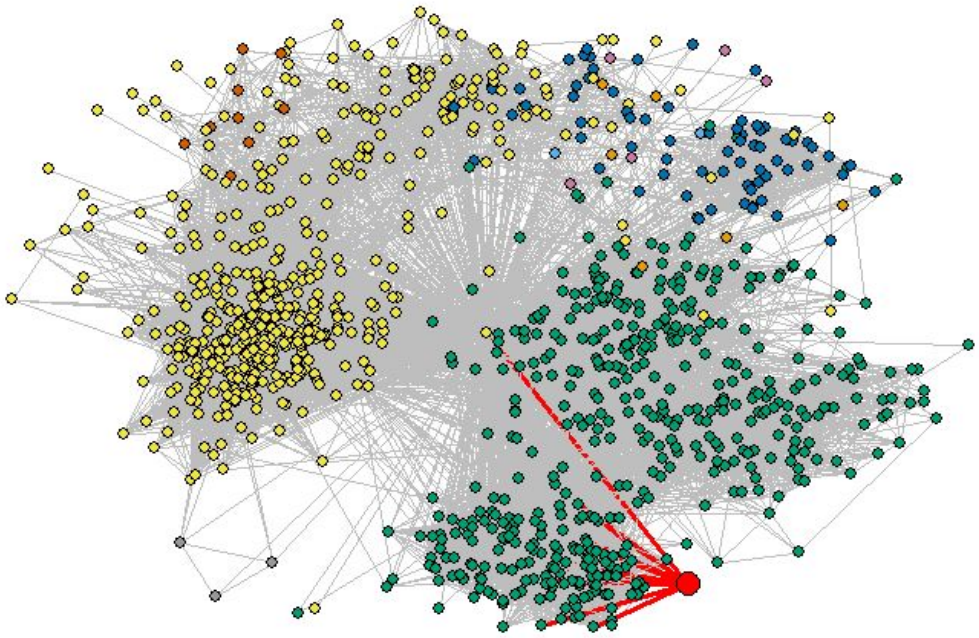


----------Maximum Dispersion/Embeddedness----------

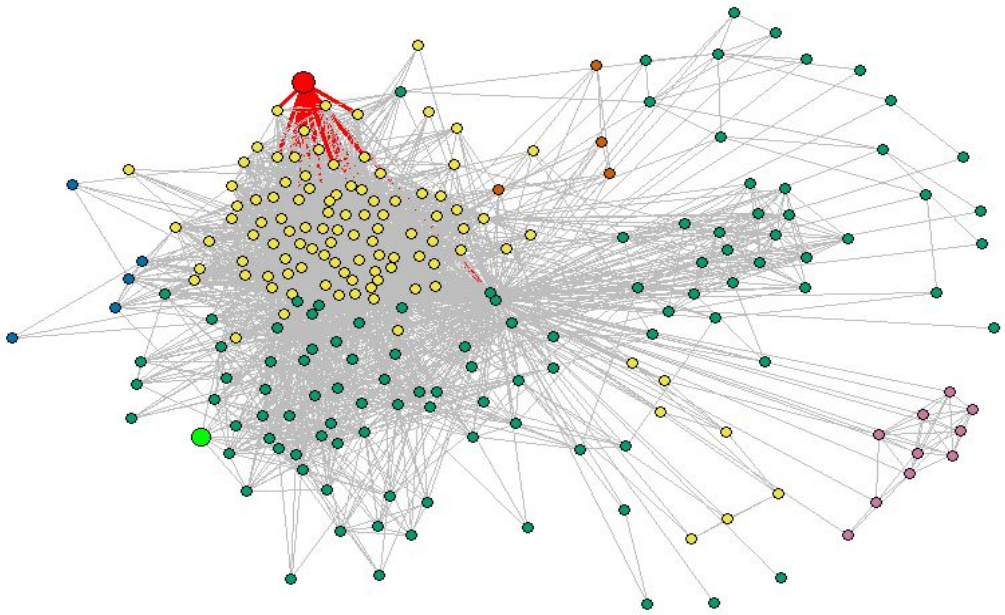
- **Node ID 1**



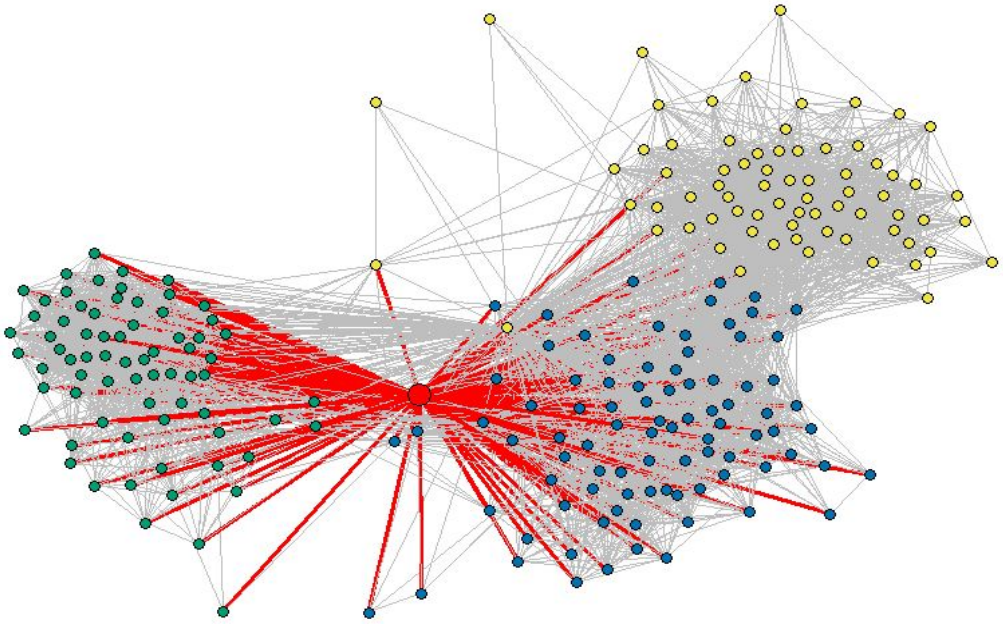
- **Node ID 108**



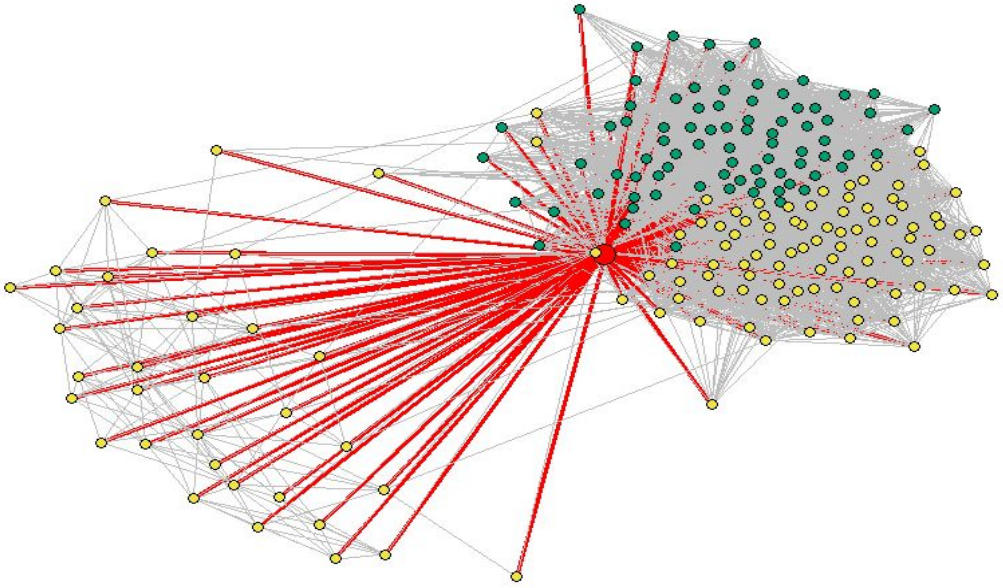
- **Node ID 349**



- **Node ID 484**



- **Node ID 1087**



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

Answer:

In the previous parts, we explored characteristics of nodes in the personalized network of 5 core nodes using the following two measures:

- **Embeddedness** is the number of mutual friends between the node and the core node. This means that higher embeddedness implies a large number of mutual friends between the two nodes. It is also a measure of tie strength among nodes. We used intersect() function available in R to calculate it.

- **Dispersion** is the sum of distances between every pair of mutual friends between the node and its core node. It measures the extent to which two people's mutual friends are not well-connected. Unlike embeddedness, dispersion also looks at the network structure of the mutual friends between two people. Dispersion can be defined by the following formula:

$$disp(u,v) = \sum_{s,t \in C_{uv}} d_v(s,t),$$

where d_v is the distance between the nodes s and t, C_{uv} indicates the common set of mutual friends between nodes u and v. Here, u is the core node and v is the node in the personalized network of u.

We calculated dispersion only for those nodes which had embeddedness greater than 1; otherwise it makes no sense. We used shortest.paths function available in iGraph to calculate distances between all pairs of mutual friends. For the nodes that were not connected by a path, we set the dispersion value as the diameter of the network plus 1.

Based on the plots in previous questions, we can make the following observations regarding the characteristics of each node:

It is noticeable that the incident edges for maximum dispersion node are not as dense the incident edges on maximum embeddedness node. This is because many people within the same clusters know each other and that leads to high embeddedness and high density of edges. However, this does not necessarily mean that there are strong ties of these nodes with the core node. As mentioned in the paper, a person's relationship partner or closest friends usually have lower embeddedness but they often involve mutual friends from several different clusters indicating that the social connections of these friends are not bounded within one cluster. This proves that embeddedness is not an accurate measure of tie strength. Instead, dispersion should be used as it measures the extent to which people's mutual friends are not well-connected.

It can be observed from the plots of **maximum dispersion** that the nodes highlighted in **blue** have maximum dispersion with respect to core node. Thus, their mutual friends are not well-connected amongst each other. Also, with the increase in the size of core node's personalized network, the dispersion value decreases.

It can be observed from the plots of **maximum embeddedness** that the nodes highlighted in **black** have maximum value of embeddedness. It is also observed that larger communities lead to larger values of embeddedness. Thus, the largest community in the personalized network will most probably contain a node that has maximum embeddedness.

We already discussed that embeddedness is not an accurate representation of tie strength. Thus, we use the **ratio of dispersion to embeddedness** to predict strength of relationships or closeness of friendships among nodes. A higher ratio indicates closer ties. It can be observed from the plots above that as the network size reduces, the ratio of dispersion and embeddedness increases.

1.4. Friend Recommendation in personalized networks

Friend recommendations in social networks is of prime importance. Recommending friends to a user whom they know and wish to connect is a fundamental problem in any social network. In terms of graphs and networks, recommending friends is nothing but to recommend new links to a node. This is the same as predicting the future links between pairs of nodes in the given network. This has always been an emerging arena and lot of algorithms have been developed so as to have the most apt recommendations rendered to a user. In this section of the project, we have implemented friend recommendation using 3 measures: Common Neighbor measure, Jaccard measure and the Adamic Adar measure. All of these three measures are neighbourhood based measures meaning that the likelihood of recommending a friend to a user depends on his existing friends/neighbours as well as their their neighbourhood too.

Question 16: What is |Nr| ?

Firstly, we create a personalized network of node ID 415. This will ensure that we operate on a network which involves only the friends of node ID 415 and itself. Obviously, we should be aware of the fact that the friends of node ID 415 may be friends amongst themselves too.

Next, as mentioned in the project statement, we choose to recommend friends only to those users who have exactly 24 friends i.e. nodes whose degree is exactly 24. Such a list of users or nodes who have 24 friends was obtained from this personalized network. Let's call this list as Nr. We got 11 such users whose degree is 24.

Therefore, **|Nr| is 11.**
The Nr nodes are: '496', '578', '600', '615', '618', '627', '643', '658', '659', '661', '662'

The Friend Recommendation Algorithm

We recommend friends to each of the Nr users. The algorithm is as follows:

1. Remove each edge of node i at random with probability 0.25. This means we randomly unfriend some people from node i. We call this list of deleted friends as Ri. We aim to recommend these friends using our 3 algorithms to this user i.
2. We then use Common Neighbors, Jaccard and Adamic Adar measure to recommend friends to this user i. This list of recommended friends is called as Pi.
3. The accuracy of our algorithm (i.e. how similar Pi is to Ri) is then computed by the following formula:

   Accuracy for user i = (Pi ∩ Ri) / Ri

4. We repeat the above steps 1-3 ten times and take the mean of all 10 accuracies. The intuition behind performing this 10 times is that since we are **randomly** deleting the links of a user and unfriending them, it is always better to compute this several times and get the average accuracy of the computations. This we call as the accuracy for correctly recommending friends to 1 user.
5. Steps 1-4 are then repeated to calculate the accuracy for correctly recommending friends to all 10 users. Again, we take the mean of these accuracies for all users in Nr. This mean is the actual accuracy of the algorithm.
6. As mentioned above, the accuracy is computed for all 3 algorithms using steps 1-5.

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?

In this section, we describe how we used each of the three measures to recommend friends to a user. We also record the accuracy of each algorithm and draw a comparison.

As mentioned in the project specs, we have the following notations:

- S_i is the neighbor set of node i in the network
- S_j is the neighbor set of node j in the network

Consider that we are recommending t friends to a user i. For each node in the network that is not a neighbor of i, we compute the three measures as follows between the node i and the node j not in the neighbourhood of i (i.e. not a friend of user i). We then recommend t friends (with the highest respective scores) to user i.

1. **Common Neighbors measure**

The common neighbour measure between node i and node j is given by the following formula:

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

Understanding and Analysis: Since this is an intersection between S_i and Sj, this formula counts the number of mutual friends of user i and user j. We then recommend t friends (amongst all j users who are not currently friends of i) with the highest common neighbor measure to user i. This essentially means that if two users have friends in common, then those two users are likely to be friends in the future. The more the mutual friends they have, the more is the likelihood of those two users being friends.

Thus, the common-neighbors predictor captures the notion that two strangers who have a common friend may be introduced by that friend. This introduction has the effect of "closing a triangle" in the graph and feels like a common mechanism in real life.

2. **Jaccard measure**

The Jaccard measure between node i and node j is given by the following formula:

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

Understanding and Analysis: Here we see that the Jaccard measure is an intersection over union. This measure or index is very widely used in statistics to compare the similarity of diverse sample sets. However, in our case, this measure gives us an idea of how many mutual friends do two users share in common out of all the friends both of them currently have. This value will always range from 0 to 1. 0 because two users may not have any mutual friends and 1 because two users may have all friends as mutual friends.

This metric solves the problem where two nodes could have many common neighbors because they have lots of neighbors, not because they are strongly related.

3. **Adamic Adar measure**

The Adamic Adar measure between node i and node j is given by the following formula:

$$\textbf{AdamicAdar(i,j)} = \sum_{k \in Si \cap Sj} 1/\log(|S_k|)$$

Understanding and Analysis

Adamic Adar is another link prediction algorithm which can be used for friend recommendation. Here, we take the inverse log of the number of neighbours of each mutual friends of user i and j. It is also interchangeably called as the Frequency-weighted common neighbors approach. This measure refines the simple counting of common neighbors by weighting rarer neighbors more heavily. The Adamic/Adar predictor formalizes the intuitive notion that rare neighbors are more telling. The rationale behind using this in recommending friends is:

If "triangle closing" is a frequent mechanism by which new edges form in a social network, then for x and y to be introduced by a common friend z, person z will have to choose to introduce the pair (x,y) from pairs of his friends. Thus an unpopular person (someone with not a lot of friends) may be more likely to introduce a particular pair of his friends to each other.

Observations and Inference

Mean accuracy obtained by **CommonNeighbors** measure is 0.8333 i.e **83.333 %**

Mean accuracy obtained by **Jaccard** measure is 0.801515 i.e **80.151 %**

Mean accuracy obtained by **AdamicAdar** measure is 0.84545 i.e **84.545%**

Based on the average accuracy values, we see that the ***'AdamicAdar'*** friend recommendation algorithm works the **best** in our case.

Analysis

Though we get the best accuracy from Adamic Adar algorithm, we see that all three algorithms have similar accuracies with only minor differences. According to us, there are two reasons for this:

- Firstly, the network on which we apply the above friend recommendation algorithms is small since it is a personalized network. The personalized network of node ID 415 comprises of only 160 nodes and 1857 edges before we delete the edges randomly. Moreover, we are recommending friends to users with only 24 friends. This is very small network as compared to actual social networks. This may be the reason why we do not get a stark difference in the accuracies of each algorithm.
- Secondly, this can also be justified by the fact that all of them are based on local similarity features i.e these algorithms are based on the idea that two nodes i and j are more likely to form a relationship in the future if their neighbor sets have large number of common nodes.

So, what is the reason for the Adamic Adar measure to render the highest accuracy?

As mentioned above, Adamic Adar measure gives more importance to rare neighbours. When we computed $\mathbf{S}_i \cap \mathbf{S}_j$, for every $\mathbf{S}_j$, we noticed that there were a lot of Sj nodes having only 1 neighbor in common with the Si node; than those having a lot of friends in common. Such nodes are given more importance (the inverse log justifies this) by the Adamic Adar algorithm and hence

the accuracy of this algorithm is slightly better than the other two for the network which we operate in.

Why does Common Neighbor measure perform better than Jaccard?

Common Neighbor simply takes the number of mutual friends between two nodes. However, Jaccard takes into account the union of the friends of both users too. Therefore, in Jaccard, the number of mutual friends are divided by the total number of friends of both users. This value is called as the Jaccard score. Thus, this score is bounded by the total number of friends. However, this is not the case in case of Common Neighbors measure where the score is simply the number of mutual friends. Jaccard measure takes into account only the strongly related nodes. It may be possible in our network that the number of common neighbors is high only because the total number of neighbors is itself high. This, however should not be the exact measure of recommending friends since users who have more friends will always benefit from this measure. The same is happening in our case. We can prevent this from occurring if we normalize the list of common friends. Thus, dividing by total number of friends (union) does the job! This is what the Jaccard measure is. However, in our network, since nodes are not quite strongly related, the Jaccard measure is slightly less than the Common Neighbor measure.

More insights

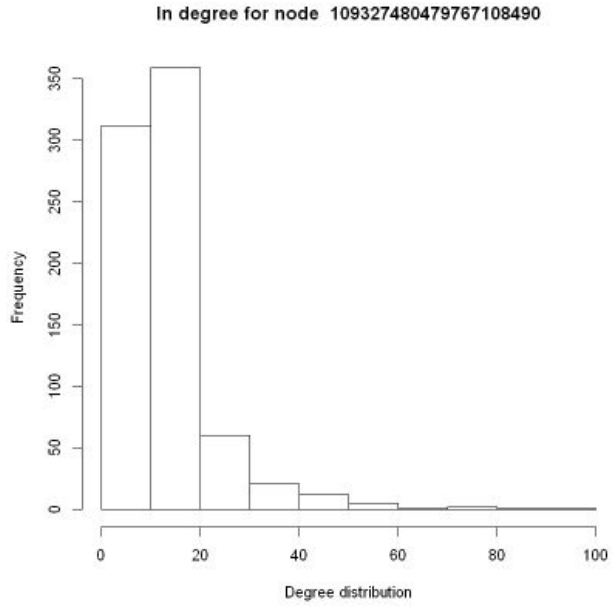
Finally, after reading a few papers online, we believe that such kind of friend recommendation or link prediction can also be done by comparing overall similarity features (and not only local similarity features) as well as node guidance capability. The Random Walk with Restart algorithm is a type of the former approach while the CNGF and KatzGf are types of the later approach. Evaluations in [2] say that CNGF and KatzGF perform better than the neighborhood based approaches we used in this project. The results obtained from the small dataset they used shows that KatzGf (which depends on node guidance capability) gives the best accuracy. It also says that the neighborhood based approaches have very similar accuracies for a small dataset (as in our case). This stands as a witness to our results and analysis.

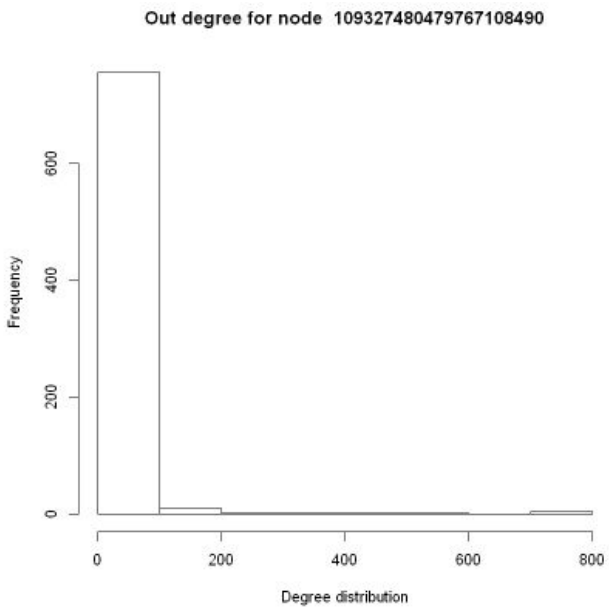
Question 18: How many personal networks are there?

The number of Personal networks is **57.**

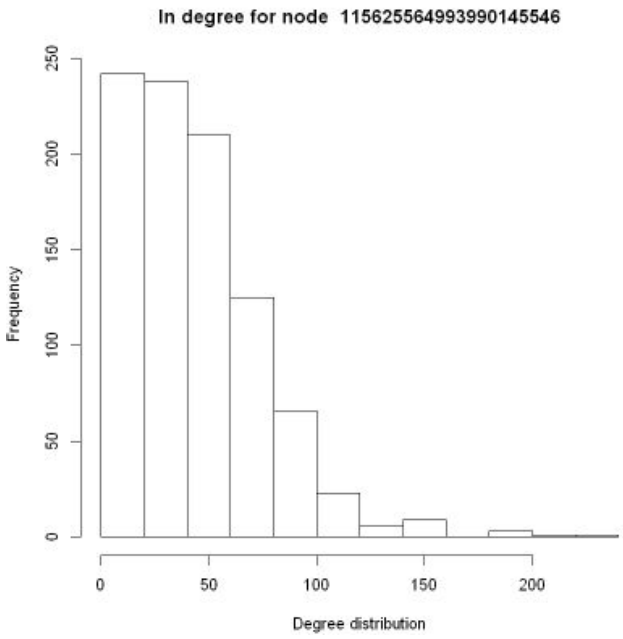
Question 19: For the 3 personal networks (node ID given below), plot the in-degree and out-degree distribution of these personal networks. Do the personal networks have a similar in and out degree distribution. In this question, you should have 6 plots.

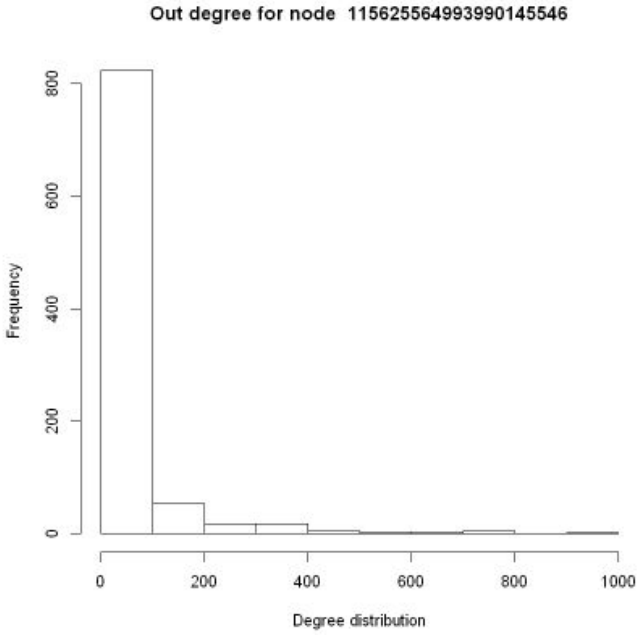
For Node 109327480479767108490:



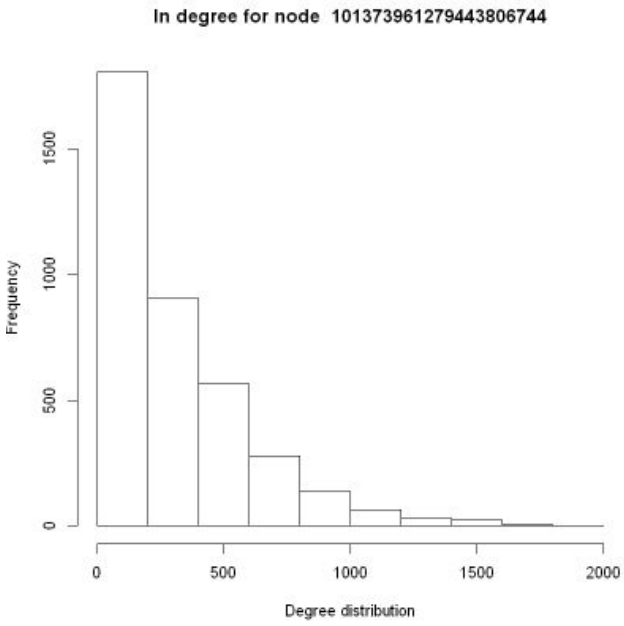


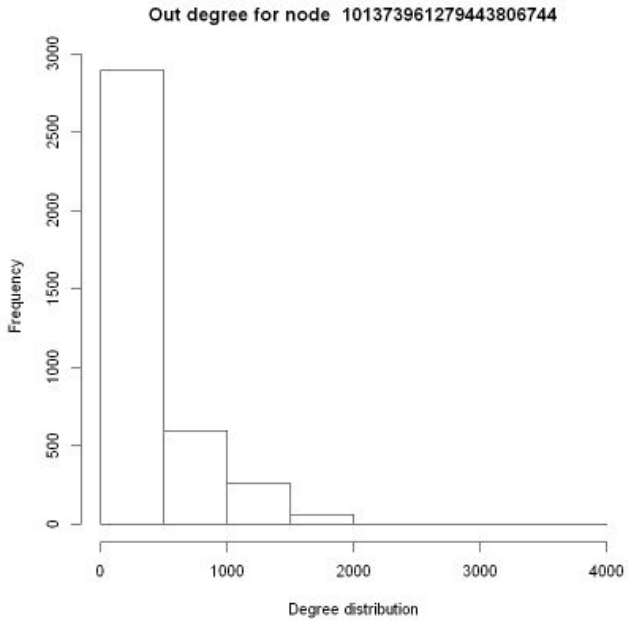
For Node 115625564993990145546:





For Node 101373961279443806744:





Answer :

Yes, we can see that the in and out degree follows a pattern here. The personal network of third node has many users in it. The indegree of three personal network seems to follows power law. We can see that there are many nodes with less indegree,but as the indegree increases, the frequency drops drastically.

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.

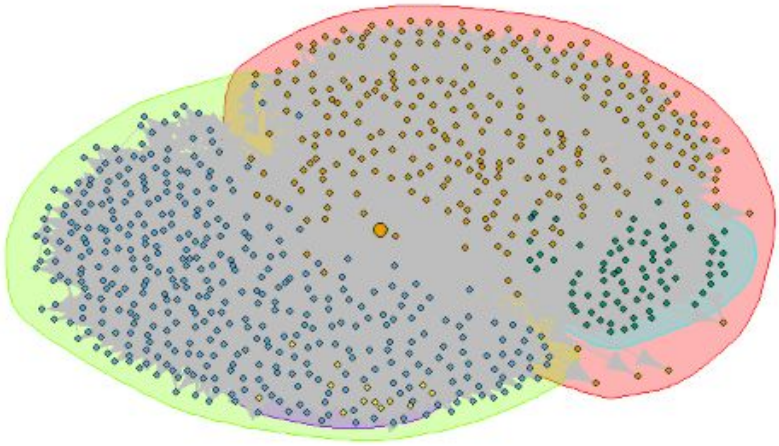
Walktrap community detection algorithm :

This approach based on random walks. The general idea is that if you perform random walks on the graph, then the walks are more likely to stay within the same community because there are only a few edges that lead outside a given community. Walktrap runs short random walks of 3-4-5 steps (depending on one of its parameters) and uses the results of these random walks to merge separate communities in a bottom-up manner like fastgreedy.community. Again, you can use the modularity score to select where to cut the dendrogram. It is a bit slower than the fast greedy approach but also a bit more accurate (according to the original publication).

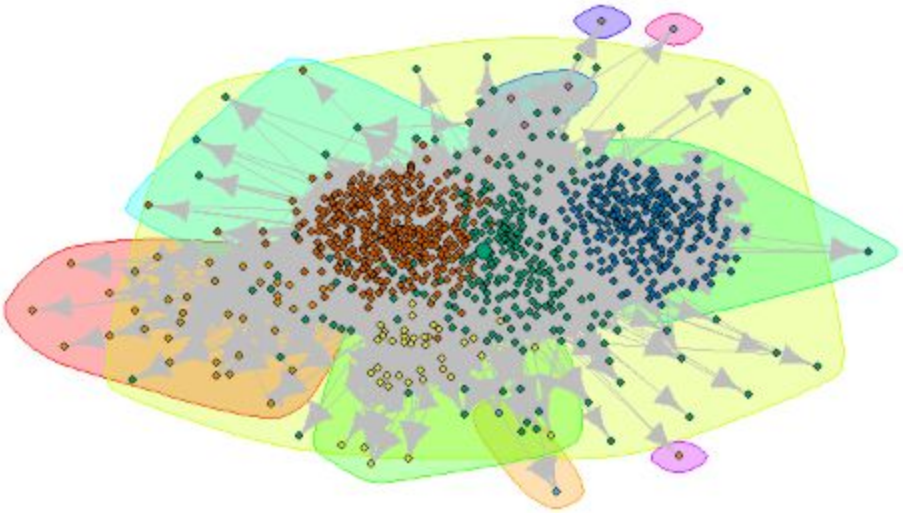
Modularity of three personal network is as follows:

Node	Modularity
Node 109327480479767108490	0.2527654
node 11562556499399014554 6	0.3194726
node 101373961279443806744	0.1910903

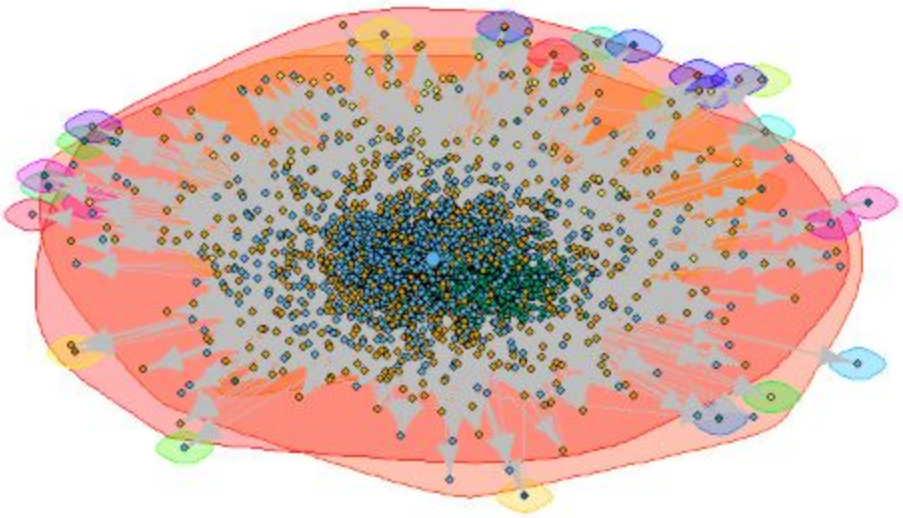
For node :109327480479767108490



For node :115625564993990145546



For node :101373961279443806744



Modularity score is not similar, however all three graphs do not have a very good modularity. The third node has the lowest modularity. From the structure we can observe that the user points in communities are not very well separated. The first node has slightly higher modularity. We can see that the points are nicely separated(we can observe the different colors) .But the communities are tightly coupled.

The second node has the highest modularity. Points are well formed in different communities also inter communities have comparatively less coupling.

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

Homogeneity and completeness and measures that are used to calculate the accuracy of any clustering technique. Homogeneity helps us to identify that a community contains users which belong to the same circle. [3]

$$h = 1 - \frac{H(C|K)}{H(C)|}$$

We determine how close a given community is to the ideal circle by examining the conditional entropy of the circle distribution given the community. In the perfectly homogeneous case, this value, H(C|K) = 0. When H(C|K) is 0 the value of h in the above equation becomes 1 which implies perfect homogeneity. But if H(C|K) is not zero, we normalize it using H(C).

Completeness ensures that all the users which belong to the same circle must be assigned to the same community.

$$c = 1 - \frac{H(K|C)}{H(K)}$$

In a perfectly complete case, all members of a class will belong to just one cluster.We evaluate this by using conditional entropy of the community given a circle, i.e., H(K|C). In a perfectly complete case, H(K|C) =0. When H(K|C) is 0, the value of c in the above equation becomes 1 which implies 100% completeness. However, if H(K|C) is not zero, we normalize it with H(K).

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.

The homogeneity(h) and completeness values for three nodes are:

Node	h	c
Node 109327480479767108490	0.85189	0.32987
node 11562556499399014554 6	0.45189	-3.42396
node 101373961279443806744	0.00387	-1.50424

The above table shows values of homogeneity and completeness for 3 nodes.

For node 109327480479767108490,

The value of homogeneity is pretty good which means that most of the communities have users belonging to the the same circle. The value of completeness is not that good which means that the all the users belonging to the same circle do not fall in the same community.

For node 115625564993990145546,

Homogeneity is close to 50%. We can interpret that 50% of the community have members from the same circle. The value of completeness is negative here . This is because the value of H(K|C) > H(K).

For node 101373961279443806744,
Homogeneity is 0.3 % which is very bad. Also the value of completeness is negative which is because H(K|C) > H(K).

REFERENCES

[1] Social Media Mining, Reza Zafarani, Mohammad Ali Abbasi, Huan Liu

[2] Gupta, Sahil, Shalini Pandey, and K. K. Shukla. "Comparison analysis of link prediction algorithms in social network." International Journal of Computer Applications 111.16 (2015).

[3] V-Measure: A conditional entropy-based external cluster evaluation measure. Andrew Rosenberg and Julia Hirschberg