ECE 232E: Large-Scale Social and Complex Networks: Models and Algorithms

Project 2: Social Network Mining

Akshay Sharma (504946035)

Anoosha Sagar (605028604)

Nikhil Thakur(804946345)

Rahul Dhavalikar (205024839)

I. Facebook Network

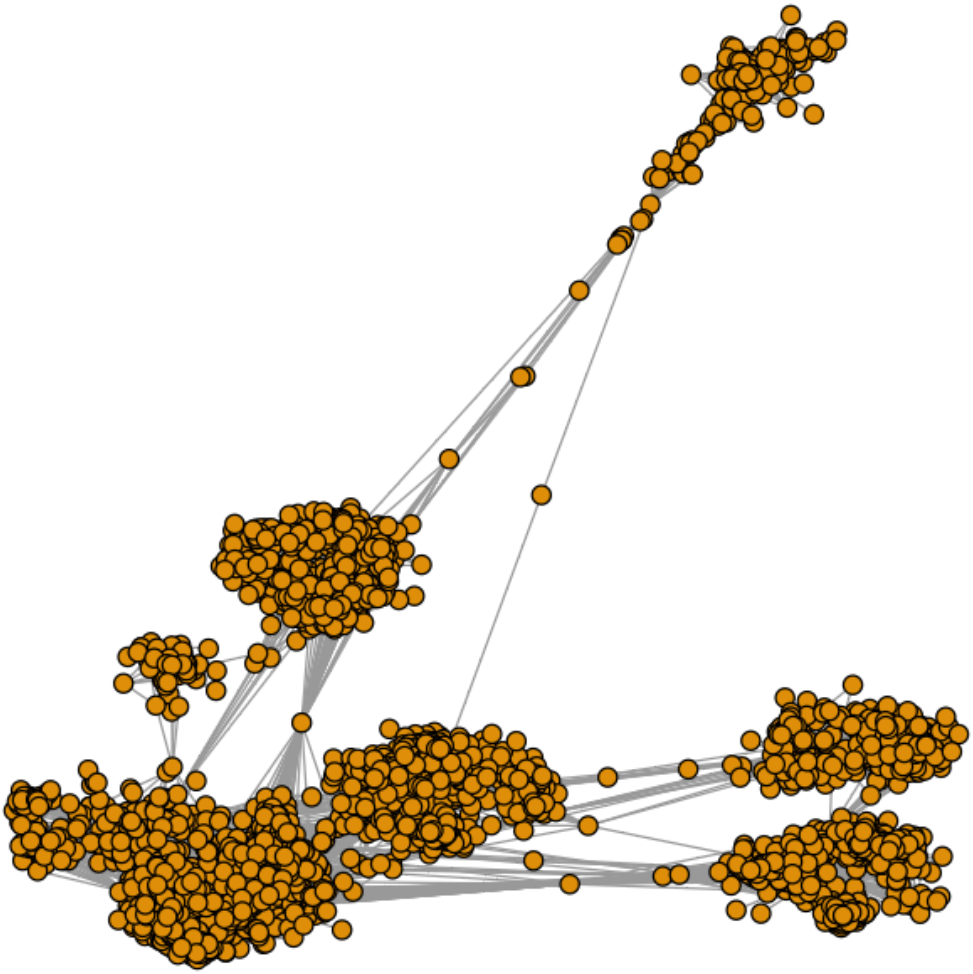
1. Structural Properties of the Facebook Network

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

Ans:

The Facebook dataset is basically an edgelist files from which we construct the network.

The Facebook network looks as follows (using Fruchterman Reingold layout):



Number of Vertices	4039
Number of Edges	88234
Is the Facebook Graph Connected	YES
Size of GCC	4039
Modularity	0.7773806

Modularity is the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. It is a measure the strength of division of a network into modules (also called groups, clusters or communities).

We can clearly see that the graph is highly modular with several distinct communities having dense connections within communities and sparse connections among communities.

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

Ans:

The diameter of a network is defined as the longest shortest path between pair of nodes.

$$d = \max_{v \in V} \epsilon(v).$$

In a connected network with n nodes, the diameter is in the range 1 (completely connected) to n - 1 (linear chain)

Diameter of the Facebook Graph	8

With respect to social networks like Facebook, LinkedIn, etc. the diameter can indicate the maximum number of hops between any two users.

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

The solution for this problem has been combined with the solution of the next problem.

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.

Ans:

On plotting the degree distribution of the facebook network, we observed that it is a type of a **scale free** network.

A *scale-free* network is a network whose degree distribution follows the *power law*.

The degree distribution of a power law network can be approximated with

$p_k \sim k^{-\gamma}$ where the exponent $-\gamma$ is its degree exponent

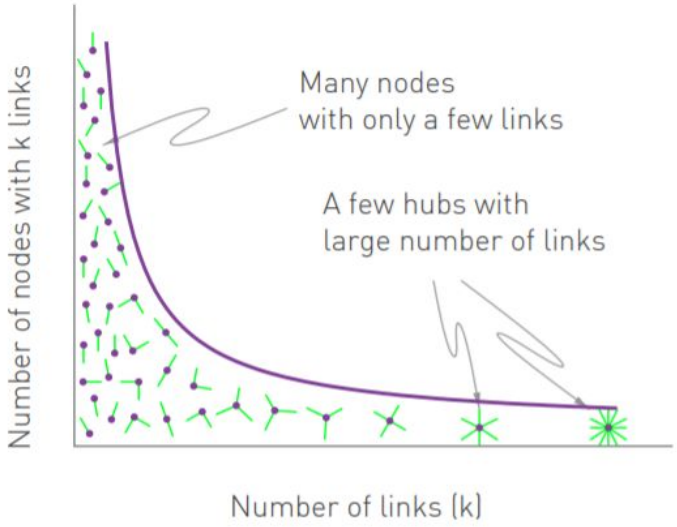
If we take logarithm of the above equation, we get

$$log\, p_k \sim -\gamma log\, k$$

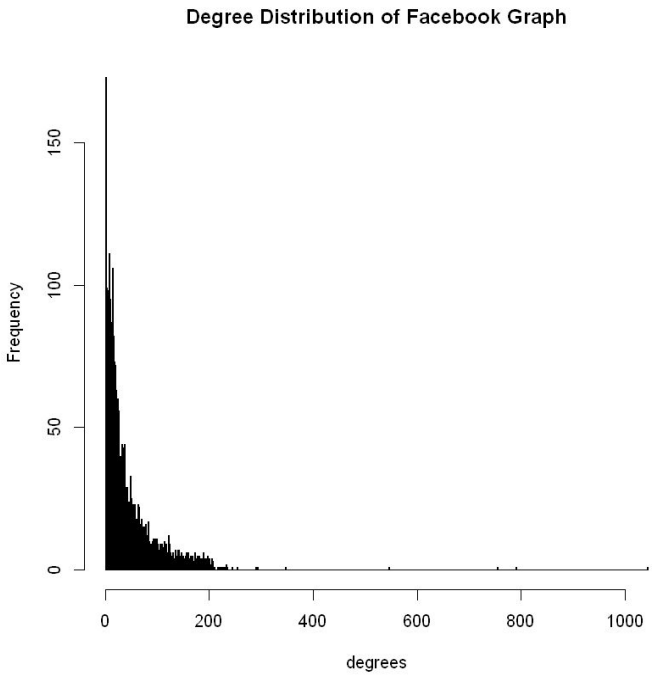
Hence, $log\, p_k$ depends linearly on $log\, k$, the slope of this line being the degree exponent γ.

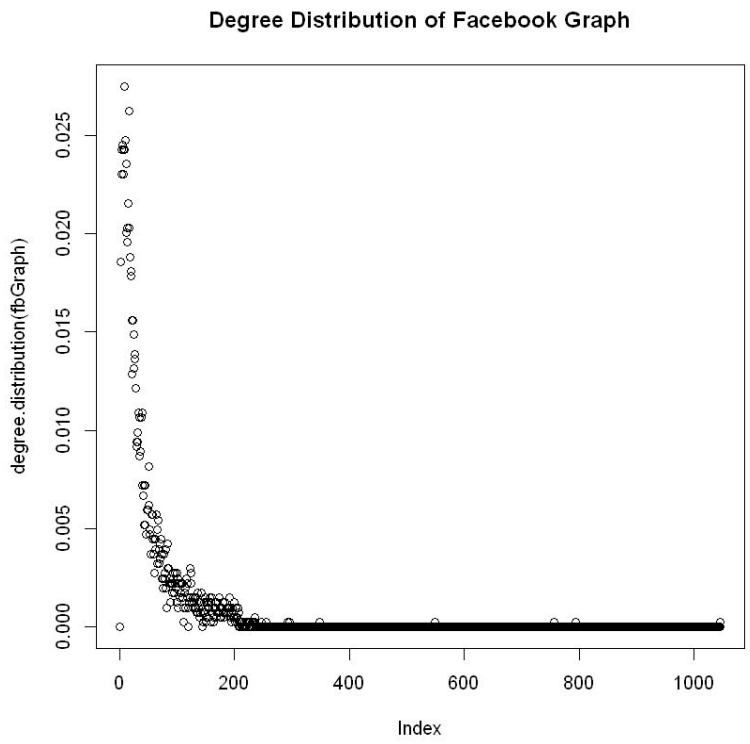
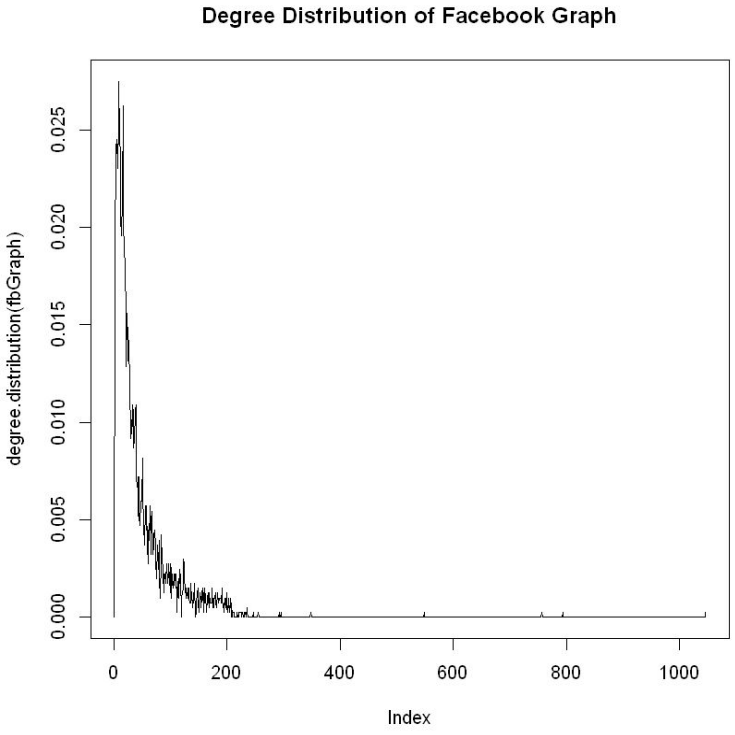
In this and the next problem, we plot both the log log plots of the degree distribution and also a histogram representing the degree distribution of the network.

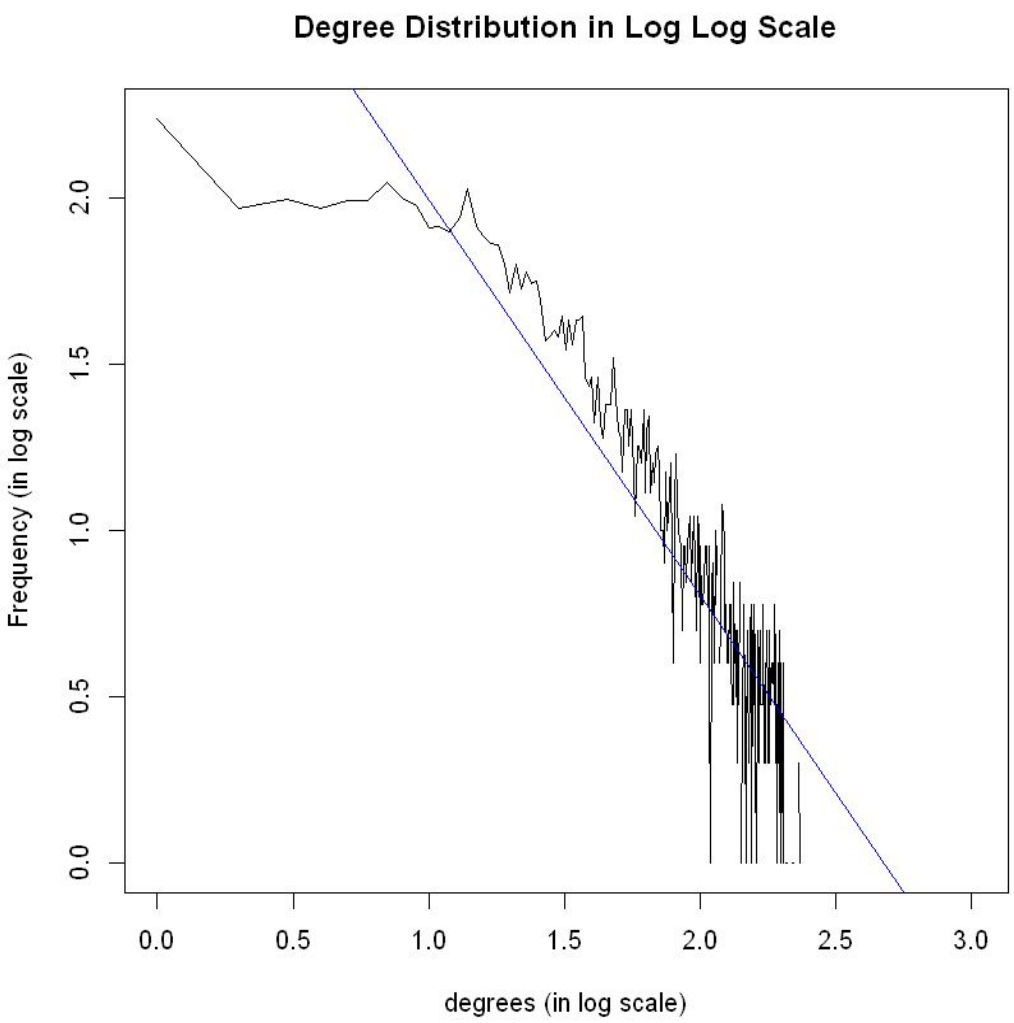
From the histograms we can observe that a large number of nodes have lower degrees and fewer nodes have large degrees. This indicates the presence of hubs which connect nodes with lower degrees. The following representative image highlights the expected degree distribution in preferential attachment models which is in line with our observations.



The preferential attachment model is called a fat tailed network as its degree distribution has a power law tail in the high-k region. As a consequence $< k^2 >$ is much larger than <k>, resulting in considerable degree variations. In scale free models such as the preferential attachment model, γ lies between 2 and 3, which is in line with our observation.

Histogram of Degree Distribution	

Scatter Plot of Degree Distribution	
Line Plot of Degree Distribution	
Average Degree	43.6910126268878

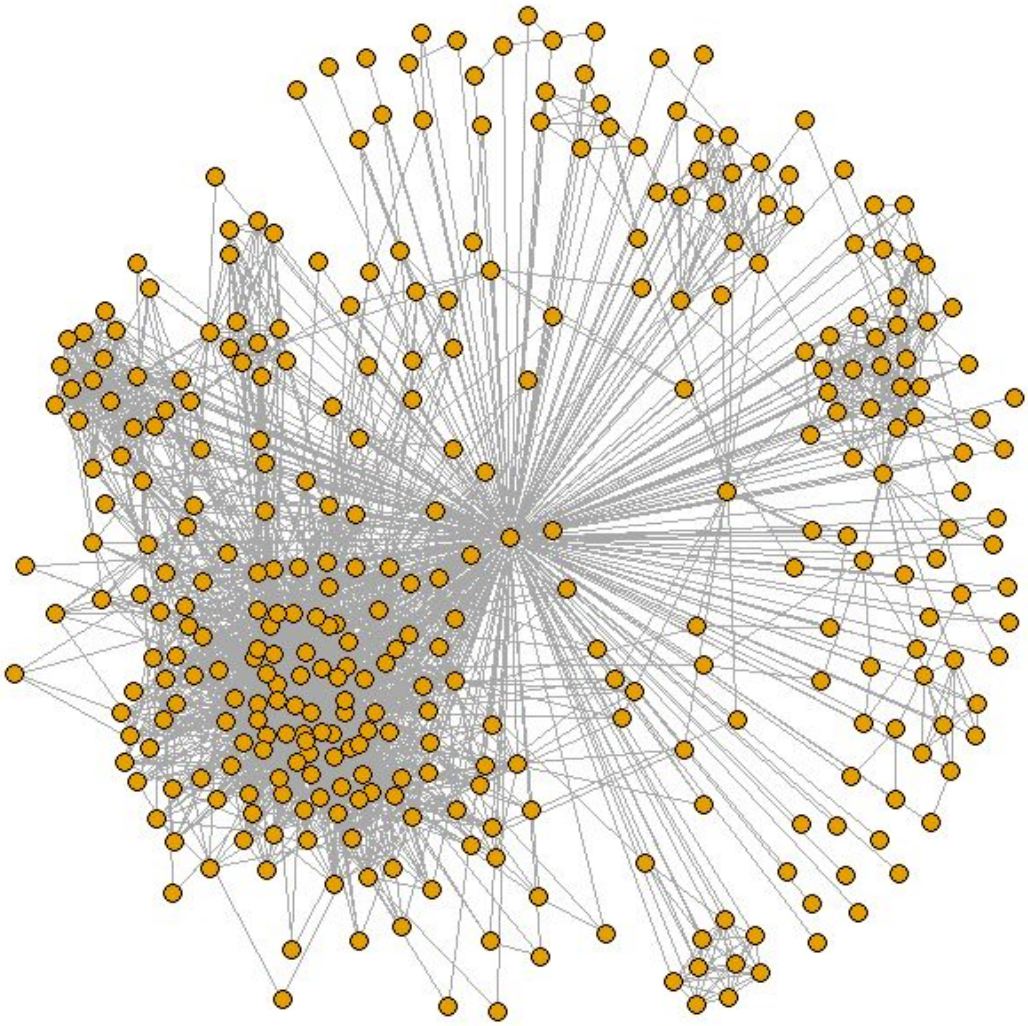
Degree Distribution in Log Log Scale	
Slope of Plot by Fitting a Line	-1.189
Power Law Exponent	2.5101754177285

2. Personalized Network

A personalized network of an user vi is defined as the subgraph induced by vi and its neighbors. In this part, we will study some of the structural properties of the personalized network of the user whose graph node ID is 1 (node ID in edgelist is 0). From this point

onwards, whenever we are referring to a node ID we mean the graph node ID which is 1 + node ID in edgelist

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



Personalized Graph for Node with Node ID 1	
Number of Vertices	348
Number of Edges	2866

We can see that the personalized graph for this node is densely connected with ratio of edges to vertices is more than 8. From our observations of the various networks studied so far, a high ratio of edges to vertices implies that even if some of the edges are deleted from the network, the network will stay connected and will not have any isolated nodes until the ratio of edges to vertices falls below a certain threshold.

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 Personalized Network	2
Trivial upper bound for the diameter	2
Trivial lower bound for the diameter	1

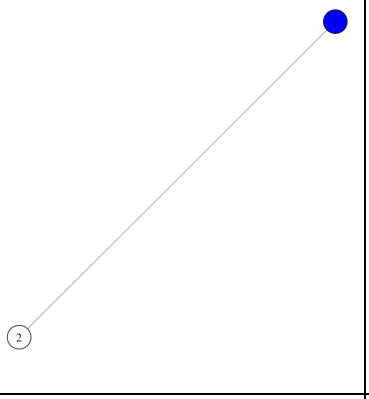
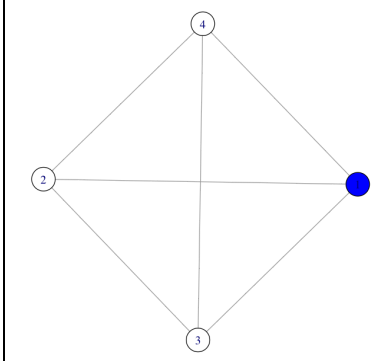
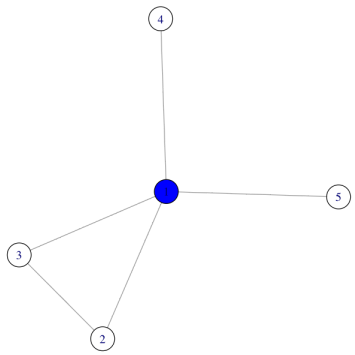
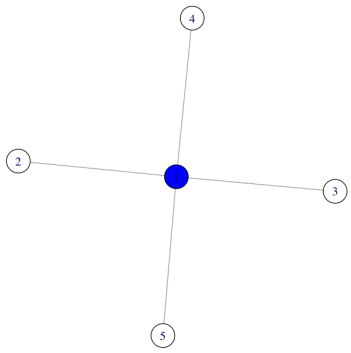
7) In the context of the personalized network, what is the meaning of the diameter of the personalized network to be equal to the upper bound you derived in question 6. What is the meaning of the diameter of the personalized network to be equal to the lower bound you derived in question 6?

A personalized network contains the core node and its neighbors. That means that all the nodes in the network are connected to the core node. So to reach any node in the network from a node, we can always go via the core node. Hence the maximum shortest distance or the upper bound of the diameter for the network will be equal to 2.

Incase the personalized network is fully connected, then every node is directly connected to every other node in the network and the shortest path between any 2 nodes is 1 and hence the diameter is 1. This is the trivial lower bound for the diameter. So when the personalized network has diameter equal to this lower bound, we can conclude that the network is fully connected. Similarly, when the diameter of the network is equal to the upper bound, we can conclude that the network is not fully connected.

Below, we have provided a graphical explanation for the same. In the first image, the core node is connected to only one node. In this case, the diameter will be 1. As we increase the nodes we come across various possibilities. In the second image, the core node is connected to three nodes and these in turn are connected with each other indicating a fully connected graph with diameter 1. The third image has one

core node and 4 neighbors and some connections in between them. As the graph is not fully connected, the diameter is 2. Lastly, the network in the fourth image has no connections between the neighbors of the core node and the diameter in this case is also 2.

			
Diameter = 1	Diameter = 1	Diameter = 2	Diameter = 2

Note: The lower bound of 1 for diameter is with respect to the network provided to us. It is also possible that the diameter can be 0. This particular scenario occurs when we introduce a new user to the network. Here the user will have no friends and hence its personalized network will only contain itself. Hence the diameter will be 0.

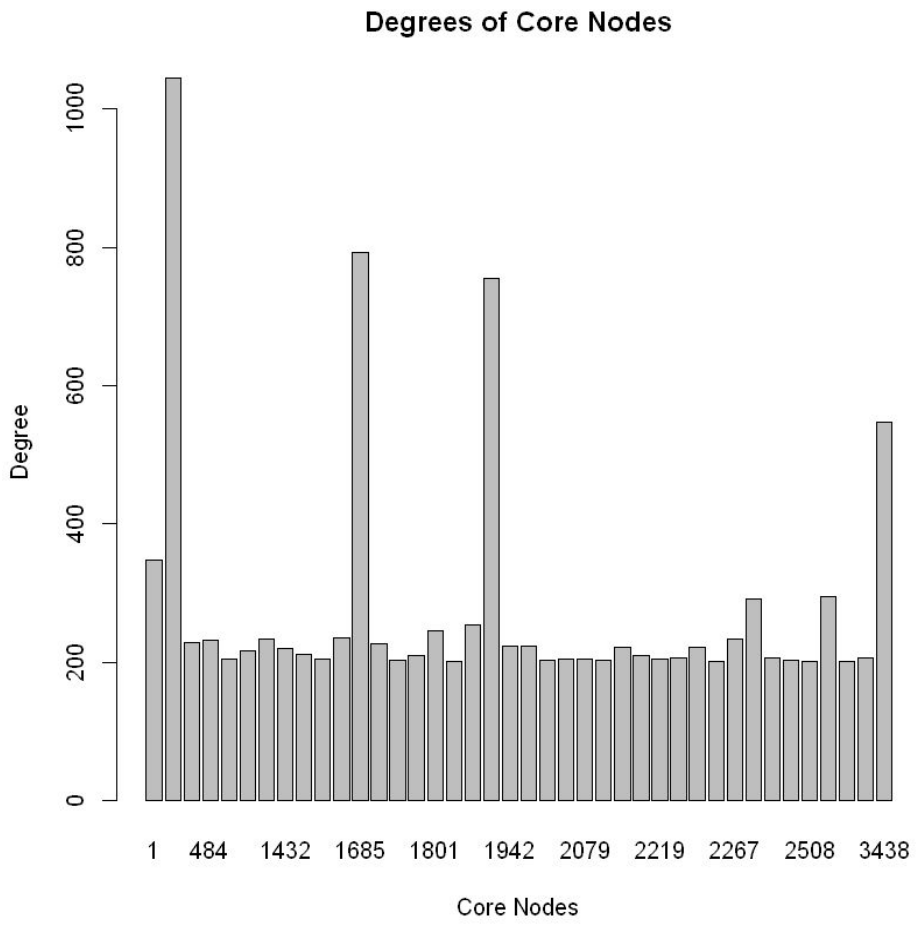
3. Core node's personalized network

A core node is defined as the nodes that have more than 200 neighbors. For visualization purpose, we have displayed the personalized network of a core node below.

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

Ans:

Number of Core Nodes	40
Core Nodes	1 108 349 484 1087 1200 1353 1432 1585 1590 1664 1685 1731 1747 1769 1801 1828 1889 1913 1942 1986 1994 2048 2079 2124 2143 2207 2219 2230 2234 2241 2267 2348 2411 2465 2508 2544 2561 2612 3438
Degree of Core Nodes	347 1045 229 231 205 217 234 220 211 205 235 792 226 202 209 245 201 254 755 223 224 203 205 204 203 221 210 205 207 222 201 234 291 207 202 201 294 201 207 547

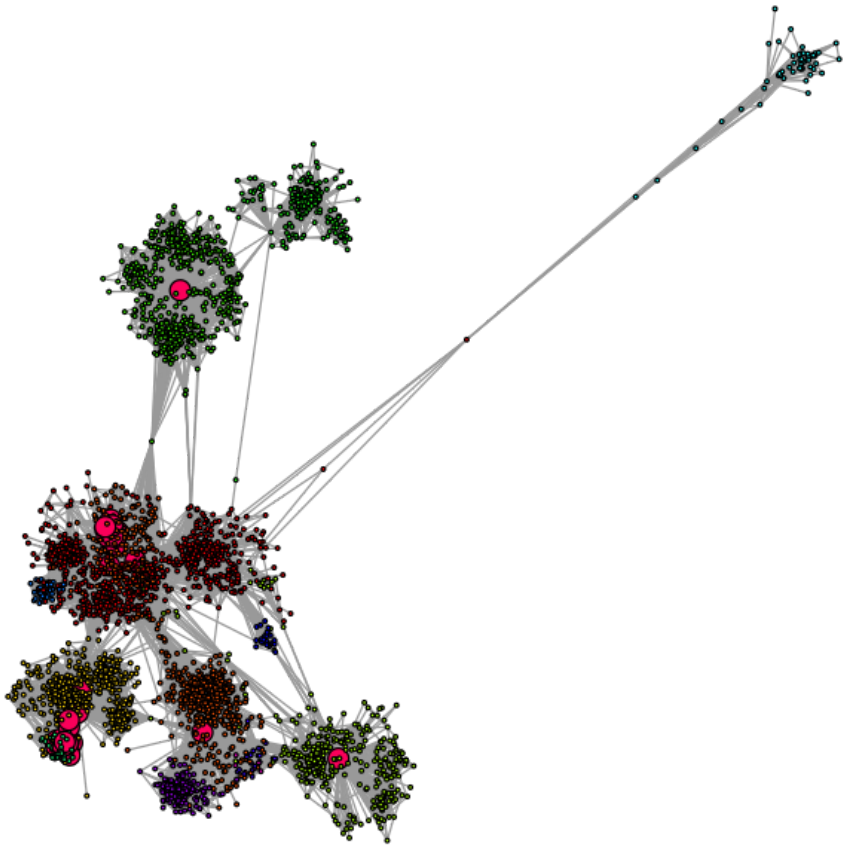
Average Degree of Core Nodes	279.375
Graphical Representation of the Degrees of Core Nodes	

The average degree of the Facebook network is 43 (calculated in question 4). We can clearly see that the core nodes have a higher average degree and these nodes will generally be part of communities and act as community centres which have maximum number of edges associated with them. The core nodes contribute to around 12% of the total edges in the network. These core nodes contribute to reducing the the modularity of the network as they have the most edges that goes across communities.

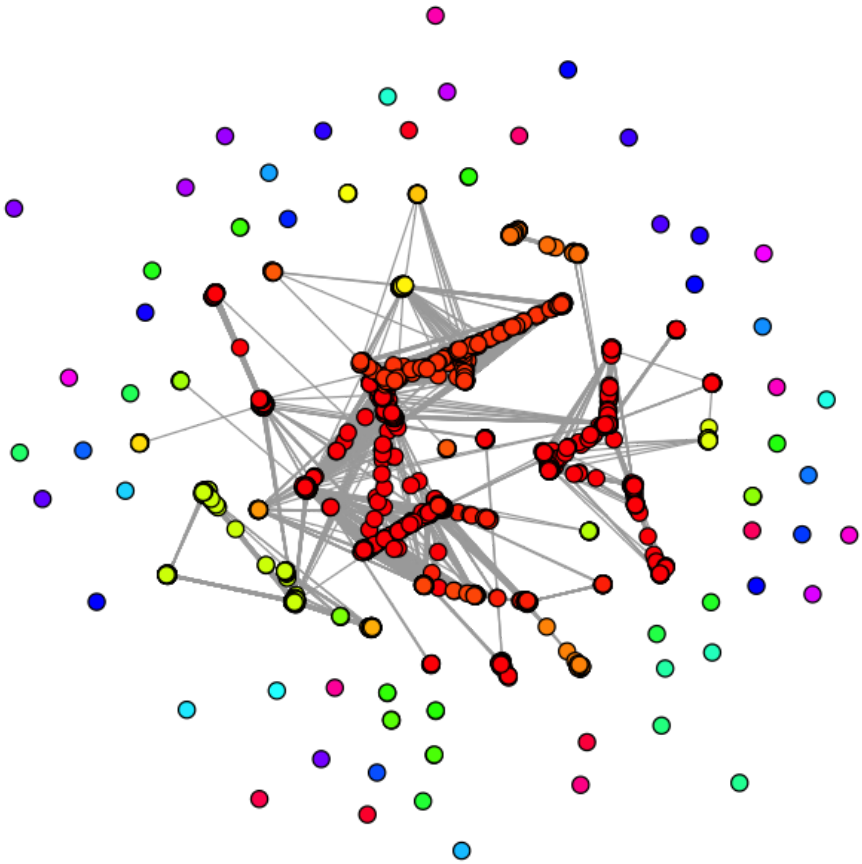
As an experiment, we deleted all the core nodes and their edges from their graph and could clearly see that the modularity of the network increased, which confirms our observation.

Number of Vertices	4039
Number of Edges	88234
Modularity	0.7773806
Number of Vertices (after deleting core nodes)	3999
Number of Edges (after deleting core nodes)	77349
Modularity	0.8242329

In the figure below, we can clearly see the core nodes (enlarged in size) are all located in communities with dense connections.



After the removal of core nodes from the Facebook graph, a lot of isolated nodes appear and it seems that the community structure has disintegrated. We could clearly see several distinct communities in the figure above, as opposed to lesser number of communities after deleting the core nodes, which confirms our observation that core nodes generally form community centres.



- **Community structure of core node's personalized network**

In this part, we study the community structure of the core node's personalized network. To be specific, we will study the community structure of the personalized network of the following core nodes:

- Node ID 1
- Node ID 108
- Node ID 349
- Node ID 484
- Node ID 1087

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.**

Ans:

Fast-Greedy: It's a hierarchical agglomeration algorithm for detecting community structure which is faster than many competing algorithms. It tries to optimize the modularity in a greedy manner. Its running time on a network with 'n' vertices and 'm' edges is O(m*d*log n) where 'd' is the depth of the dendrogram describing the community structure.

Let $$e_{ij} = \frac{1}{2m}\sum_{vw} A_{vw}\delta(c_v, i)\delta(c_w, j)$$

represent the fraction of edges that connect vertices in community i to vertices in community j and

$$a_i = \frac{1}{2m}\sum_{v} k_v\delta(c_v, i)$$

be the fraction of edges that are attached to vertices in community i, then the operation of the algorithm involves finding the changes in

$$Q = \frac{1}{2m}\sum_{vw}\left[A_{vw} - \frac{k_v k_w}{2m}\right]\sum_{i}\delta(c_v, i)\delta(c_w, i)$$

that would result from the amalgamation of each pair of communities, choosing the largest of them, and performing the corresponding amalgamation.

Edge-Betweenness: Vertex betweenness is an indicator of highly central nodes in networks. For any node, vertex betweenness is defined as the number of shortest paths between pairs of nodes that run through it. It is relevant to models where the network modulates transfer of goods between known start and end points, under the assumption that such transfer seeks the shortest available route.

The algorithm extends this definition to the case of edges, defining the "edge betweenness" of an edge as the number of shortest paths between pairs of nodes that run along it. If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to unity. If a network contains communities or groups that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness (at least one of them). By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed.

This is a divisive hierarchical approach, the result is a dendrogram. Following are the steps for the algorithms.

1. Calculate the betweenness for all edges in the network.
2. Remove the edge with the highest betweenness.
3. Recalculate betweennesses for all edges affected by the removal.
4. Repeat from step 2 until no edges remain.

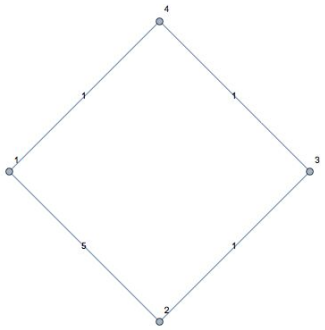
The betweenness centrality of a node v is given by the expression:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where σ_{st} is the total number of shortest paths from node s to node t and

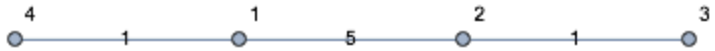
$\sigma_{st}(v)$ is the number of paths that pass through v.

If we take the example of a weighted graph below with the edge between nodes 1 and 2 with weight = 5 and all other edges with weight = 1,



It is clear that the edge with the highest betweenness is `3 <-> 4`. That's because the shortest path between `1 -> 2` is `1 -> 4 -> 3 -> 2` with a total distance of 3, instead of the distance of 5 along `1 -> 2`.

Once that is removed, the highest betweenness edge is `1 <-> 2` as it's in the "middle" of a graph like this:



It is reasonable to expect `{1,4}` and `{2,3}` as the communities, since the `1 <-> 2` coupling was less tight than all the others.

Infomap Community Detection: The hierarchical map equation measures the per-step average code length necessary to describe a random walker's movements on a network, given a hierarchical network partition. The core of the algorithm follows closely the Louvain method: neighboring nodes are joined into modules, which subsequently are joined into supermodules and so on. First, each node is assigned to its own module. Then, in random sequential order, each node is moved to the neighboring module that results in the largest decrease of the map equation. If no move results in a decrease of the map equation, the node stays in its original module. This procedure is repeated, each time in a new random sequential order, until no move generates a decrease of the map equation. Now the network is rebuilt, with the modules of the last level forming the nodes at this level, and, exactly as at the previous level, the nodes are joined into modules. This hierarchical rebuilding of the network is repeated until the map equation cannot be reduced further.

The map equation is below:

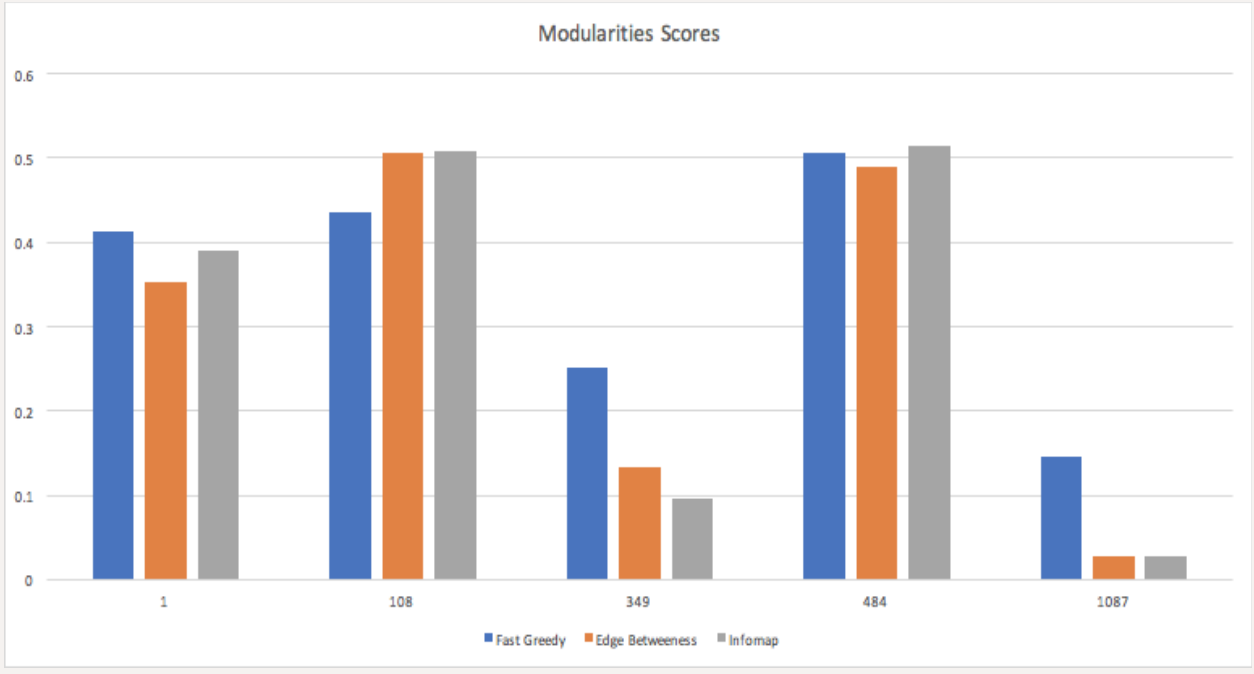
$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i)$$

Here H(Q) is the frequency-weighted average length of codewords in the index codebook and H(Pi) is frequency-weighted average length of codewords in module codebook i. Further, the entropy terms are weighted by the rate at which the codebooks are used. With qi for the probability to exit module i, the index codebook is used at a rate q = sum(qi) for all m, the probability that the random walker switches modules on any given step.

With this algorithm, a fairly good clustering of the network can be found in a very short time. The nodes assigned to the same module are forced to move jointly when

the network is rebuilt. As a result, what was an optimal move early in the algorithm might have the opposite effect later in the algorithm. Because two or more modules that merge together and form one single module when the network is rebuilt can never be separated again in this algorithm.

We used all the 3 algorithms to find the community structure using the 3 algorithms for the 5 core nodes. The following graph gives the modularity values for each of the cases.



Modularity analysis w.r.t different core nodes

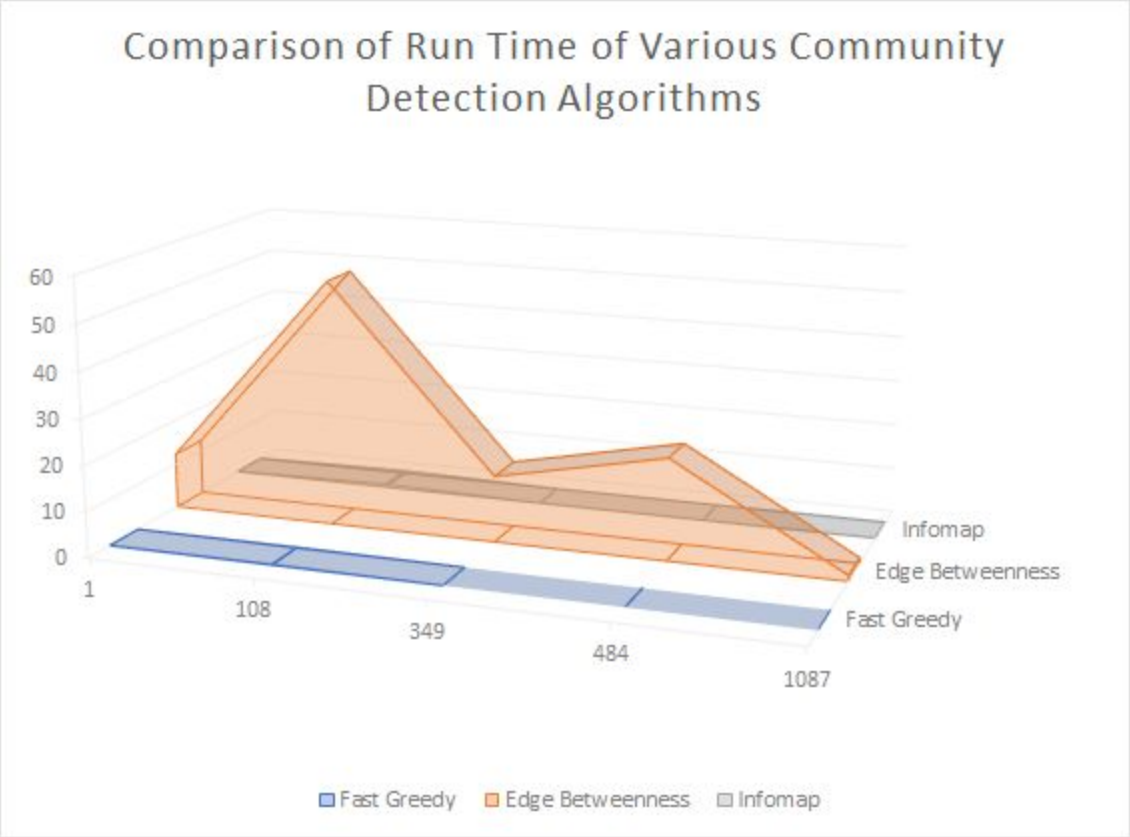
- From the point of view of core nodes, we observe high modularity values for core node with ID 484 for all the 3 algorithms. For node with ID 1087 we observe a low modularity score for all the 3 algorithms.
- Let us observe the details of these 2 nodes to get more insights.
    - For nearly the same number of vertices (232 and 206), the graph of 1087 has double the edges (7409) as compared to 484's graph(4525). This leads to the higher average degree distribution in the graph of 1087. Depending on how the communities are formed, these additional edges will affect the modularity scores.
    - Let us take a look at the communities and their sizes formed by the Infomap algorithm. Both the graphs 484, 1087 have 4 communities, what is different is the sizes of these communities.
    - Community sizes for the 4 communities of graph 484 are (85 73 70 4) and those of graph 1087 are (175 13 11 7). We observe that there are 3 equal sized communities in first cases as opposed to 1 large community in the second

graph. This results in the personalized network of node 1087 having very low modularity scores as compared to graph to 484.

- We see a similar case with communities formed by Infomap algorithm for node ID 349 where the sizes of communities are (176 19 10 8 7 3 3 2 2) and has low modularities as opposed to node ID 108 (323 127 93 102 71 72 48 38 28 29 18 16 12 11 10 6 6 5 6 6 3 3 3 3 3 2 2) which has a good distribution of nodes over all communities.
- These observations lead us to conclude that having large number of communities with lower number of members contributes to lower modularity scores as the nodes in communities with lower number of members will tend to form edges with nodes in communities that have more number of nodes and edges.

Modularity analysis w.r.t different algorithms

- Fast greedy algorithm tends to output lower number of communities due to its hierarchical agglomeration style of operation. It also tends to output larger size of communities and almost no communities which have very few nodes. This leads to a relatively higher modularity score for this algorithm for all node IDs.
- The number of communities output by the edge betweenness algorithm is the highest and it outputs a lot of communities with very few nodes. This is because of the way this algorithm functions as in each iteration, it removes the edge with highest betweenness centrality and the output of the algorithm is a dendrogram which is a tree-like structure. Unlike the fast greedy algorithm, this algorithm never attempts to combine smaller clusters into larger ones, which results in more number of clusters. In the infomap algorithm, neighboring nodes are joined into modules, which subsequently are joined into supermodules and so on. First, each node is assigned to its own module and then modules are combined into larger modules while decreasing the map equation, thus resulting in lesser number of clusters.
- We observed a degree of randomness in the result of the Infomap algorithm as it does random walks while trying to reduce the map equation described above to get the optimal result. However, this is not the case with fast greedy algorithms and edge betweenness algorithms which give fixed results in all iterations.
- Edge betweenness algorithm is the slowest among the three. The graph below shows a runtime analysis of the algorithms:



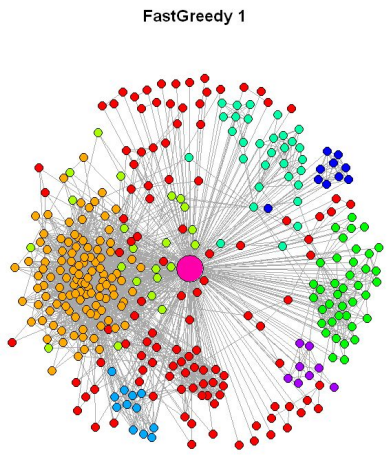
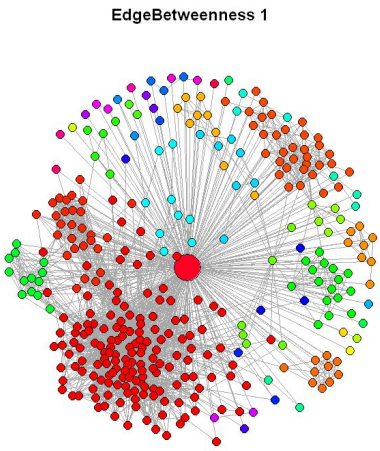
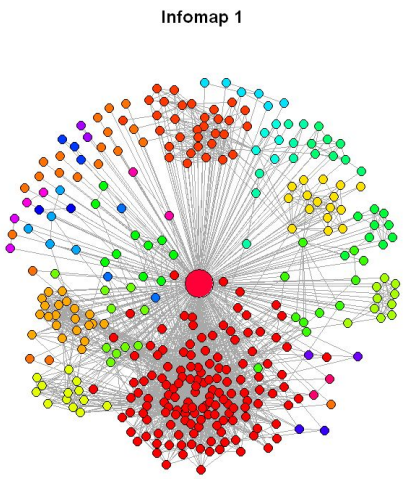
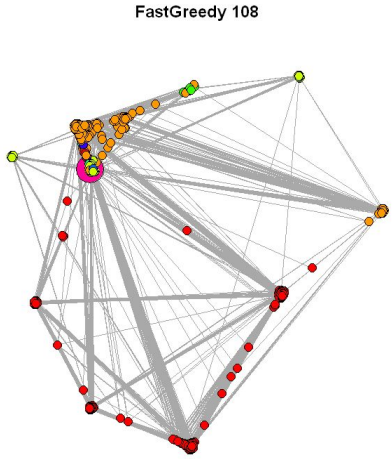
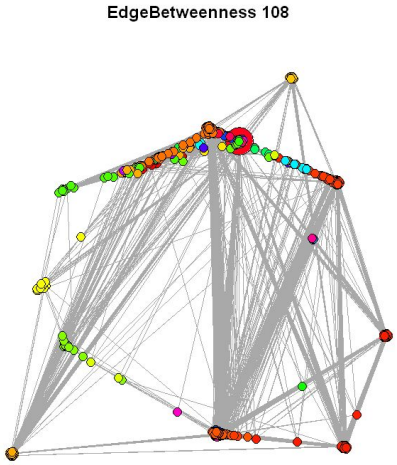
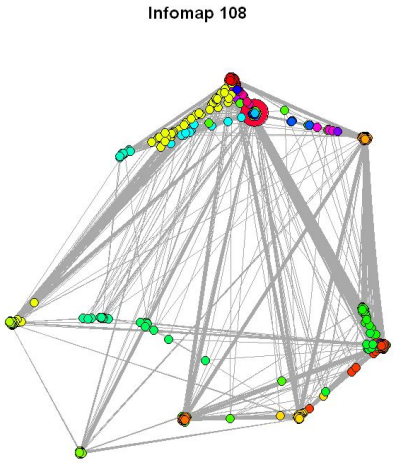
Personalized Network Statistics					
Statistics	*Core Node*				
	1	*108*	*349*	*484*	*1087*
Number of Vertices	348	1046	230	232	206
Number of Edges	2866	27795	3441	4525	7409
Connected?	YES	YES	YES	YES	YES
Size of GCC	348	1046	230	232	206
Diameter	2	2	2	2	2

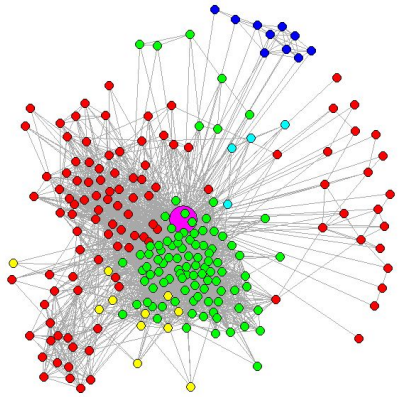
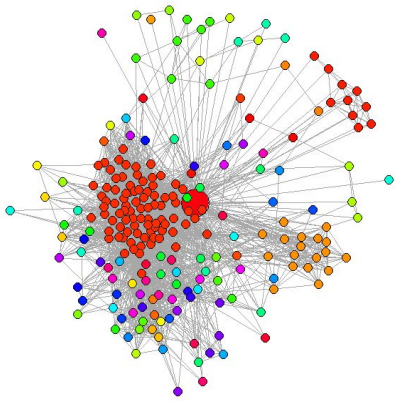
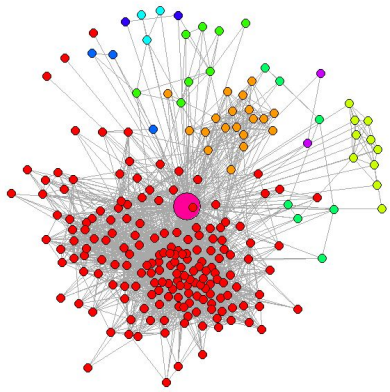
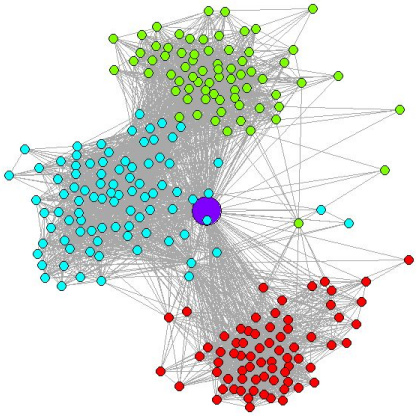
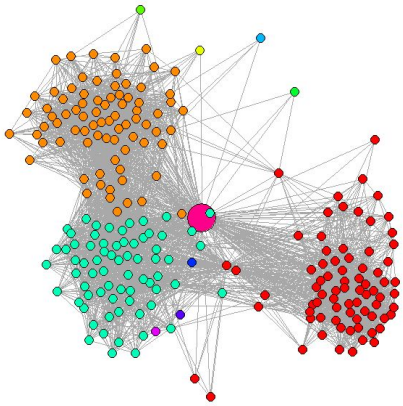
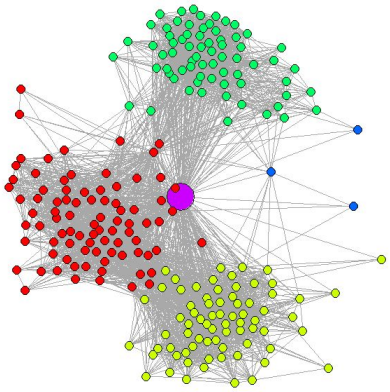
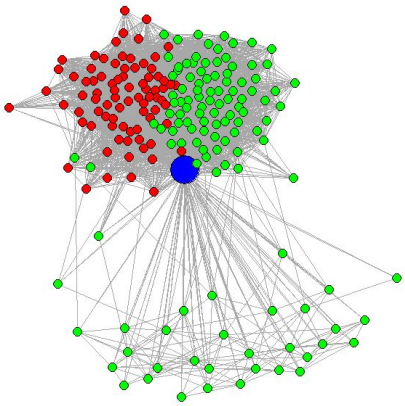
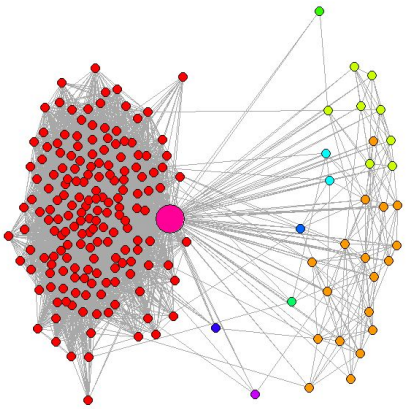
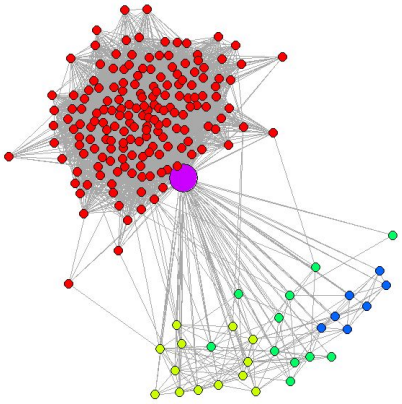
Average of Degree Distribution	16.4712643678161	53.1453154875717	29.9217391304348	39.0086206896552	71.9320388349515
Variance of Degree Distribution	555.535194938554	3156.15111200564	681.827909625973	532.302955665025	1628.05389533507

Core Node	*Community Detection Method - Modularity*		
	Fast Greedy	*Edge Betweenness*	*Infomap*
1	0.4131014	0.3533022	0.3891185
108	0.4359294	0.5067549	0.5082233
349	0.2517149	0.133528	0.0954642
484	0.5070016	0.4890952	0.5152788
1087	0.1455315	0.02762377	0.02690662

Core Node	*Community Detection Method - Community Statistics*					
	Fast Greedy		*Edge Betweenness*		*Infomap*	
	Number of Communities	*Sizes of Communities*	*Number of Communities*	*Sizes of Communities*	*Number of Communities*	*Sizes of Communities*
1	8	114 112 22 39 31 12 10 8	41	155 28 34 10 8 9 1 1 1 1 5 6 2 5 16 13 1 1 2 3 3 9 7 1 2 3 1 2 3 1 1 1 1 1 2 2 2 1 1 1 1	26	142 32 21 21 16 13 10 10 11 10 9 8 7 5 6 5 3 3 2 2 2 2 2 2 2 2
108	9	464 484 70 9 5 3 6 2 3	52	7 173 127 139 366 3 71 38 2 12 7 1 1 13 6 23 6 1 3 2 1 5 1 1 1 1 2	27	323 127 93 102 71 72 48 38 28 29 18 16 12 11 10 6 6 5 6 6 3 3 3 3 3 2

				6 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1		2
349	5	107 11 98 4 10	104	2 1 10 86 1 1 1 1 1 2 18 1 1 1 1 1 1 1 1 1 2 1 1 3 1 2 1 1 1 1 1 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	9	176 19 10 8 7 3 3 2 2
484	3	71 72 89	10	77 79 1 1 1 69 1 1 1 1	4	85 73 70 4
1087	2	81 125	9	173 18 8 1 1 2 1 1 1	4	175 13 11 7

Core Node	*Community Detection Method - Visualization*		
	Fast Greedy	*Edge Betweenness*	*Infomap*
1			
108			

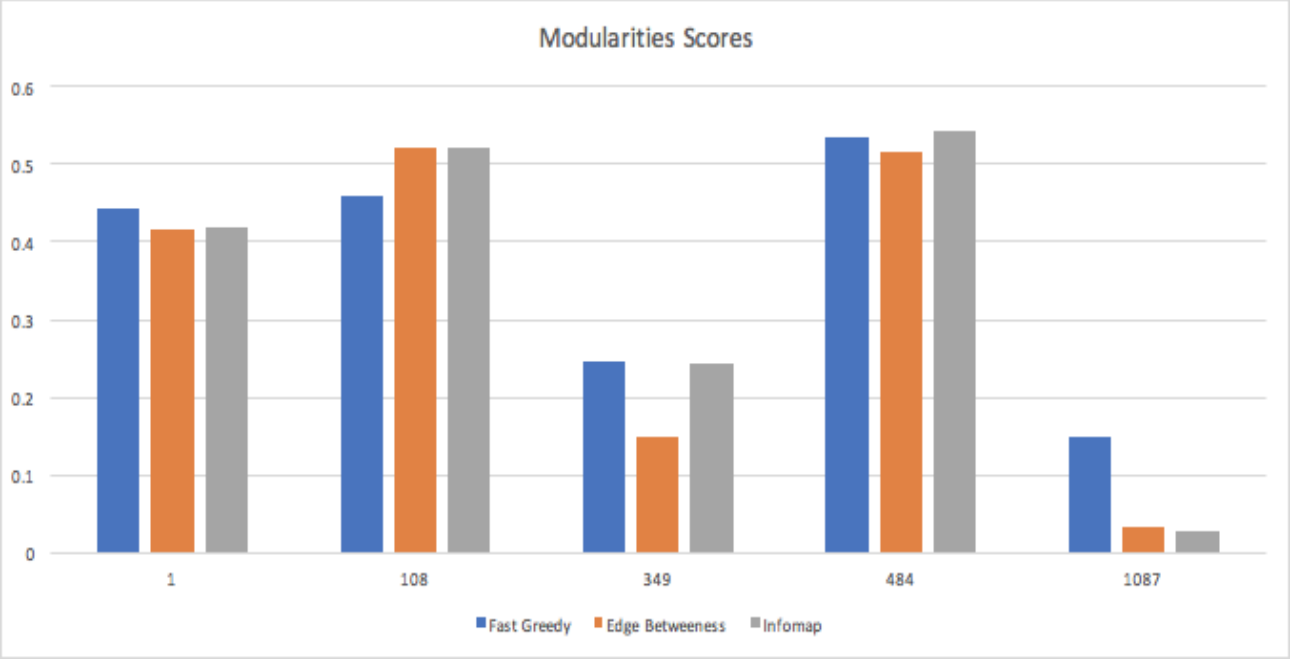
349	FastGreedy 349 	EdgeBetweenness 349 	Infomap 349 
484	FastGreedy 484 	EdgeBetweenness 484 	Infomap 484 
1087	FastGreedy 1087 	EdgeBetweenness 1087 	Infomap 1087 

- **Community structure with the core node removed**

  In this part, we will explore the effect on the community structure of a core node's personalized network when the core node itself is removed from the personalized network.

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. 4 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.

Ans: The modularities scores after modifying the personalized network as as given below.



On removing the core nodes, we observe a slight increase in the modularity scores for almost each network across all community detection algorithms. On further

investigation of our results, we observed that the increase was more prominent for edge betweenness method than the other two.

In a personalized network, core nodes act as a bridge between different communities increasing the inter community connections. On the removal of these nodes, the inter community connections reduce and these play a significant part in increasing the modularity.

To further understand the change in modularity scores, we inspected the attributes of the networks created on the removal of the core nodes. On removing the core nodes, we observe that the networks corresponding to core nodes 1, 108, and 349 are no longer connected resulting in the disintegration of the personalized network and contain few isolated nodes and slightly larger diameters. Interestingly, the diameters of these networks exceeds that of the entire Facebook network. This clearly shows the importance of these core nodes in maintaining the low diameter of the entire network.

On observing the community structure of these networks, we see that there is an increase in the number of communities with many of the isolated nodes forming their own communities as can be seen from the community sizes and the graphs.

On the other hand, the networks corresponding to core nodes 484 and 1087 retain their connectivity and hence still have lower diameters. The number of communities for these nodes is nearly the same as before with similar sizes.

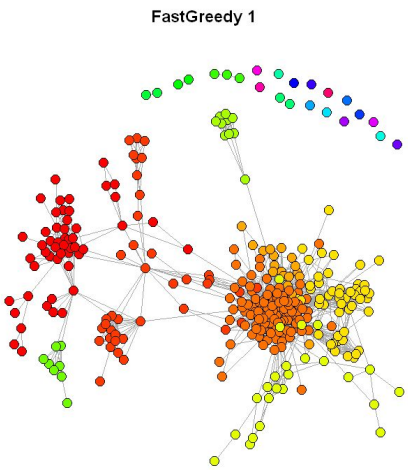
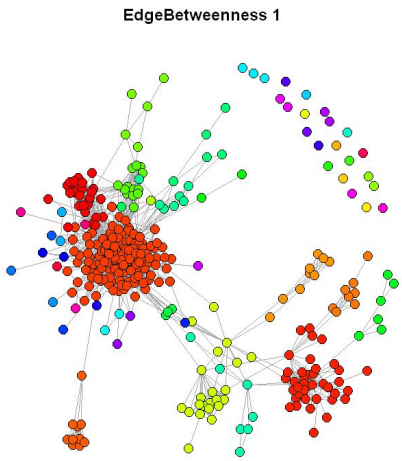
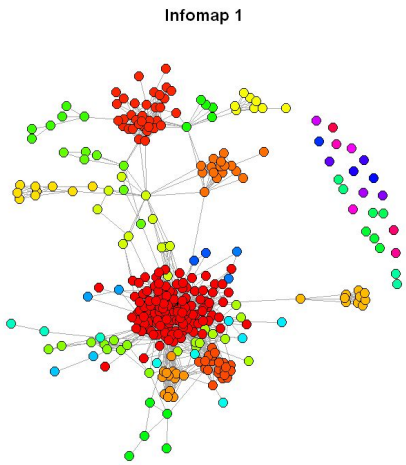
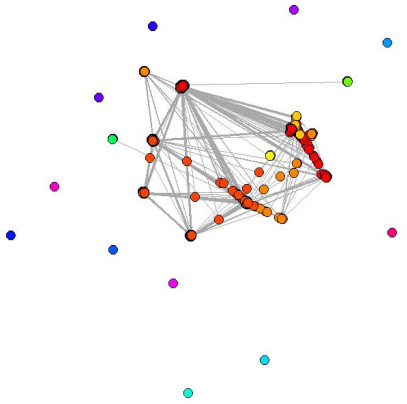
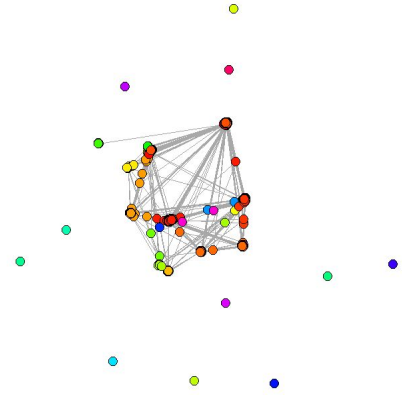
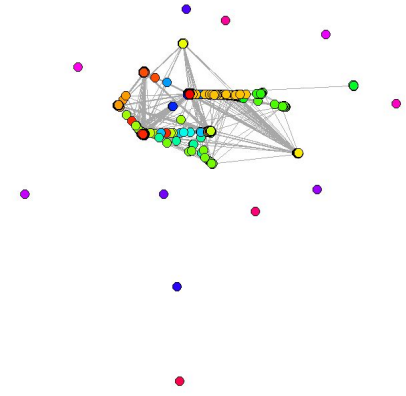
Personalized Network Statistics					
Statistics	*Core Node*				
	1	*108*	*349*	*484*	*1087*
Number of Vertices	347	1045	229	231	205
Number of Edges	2519	26750	3212	4294	7204
Connected?	NO	NO	NO	YES	YES
Size of GCC	324	1034	226	231	205
Diameter	11	9	9	3	2

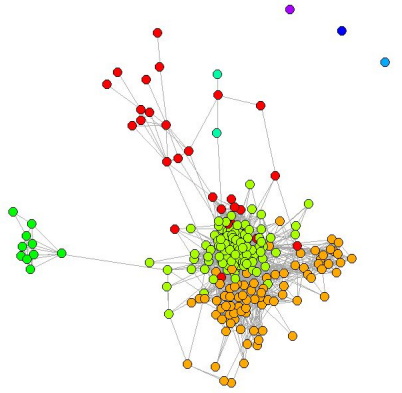
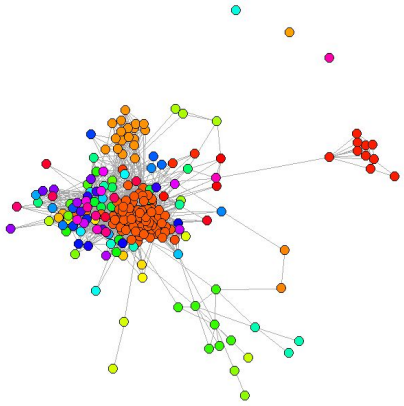
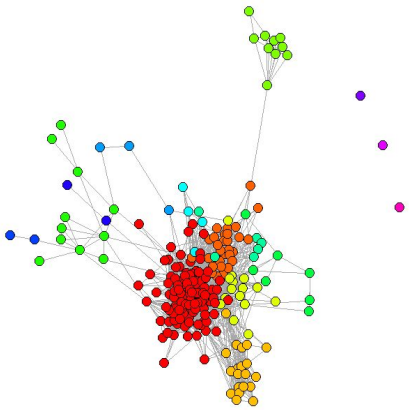
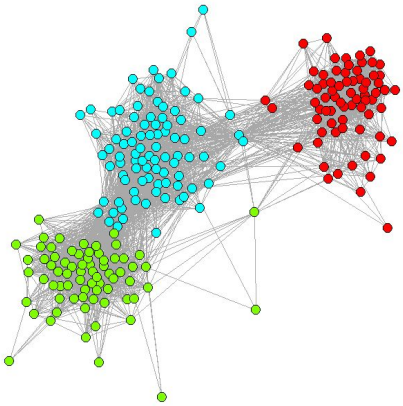
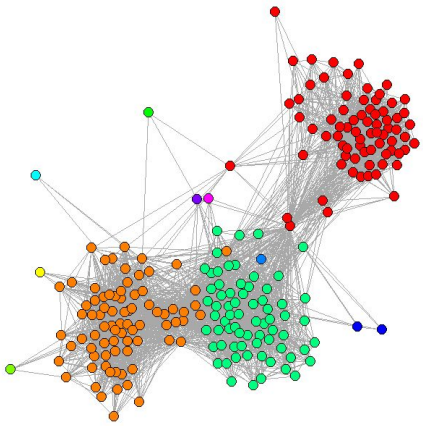
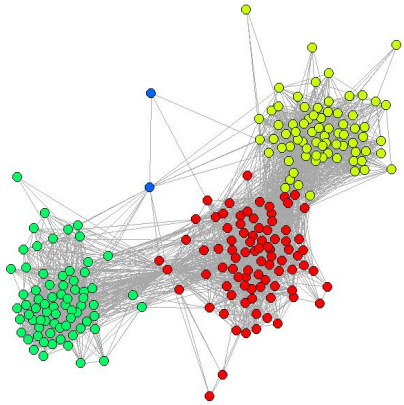
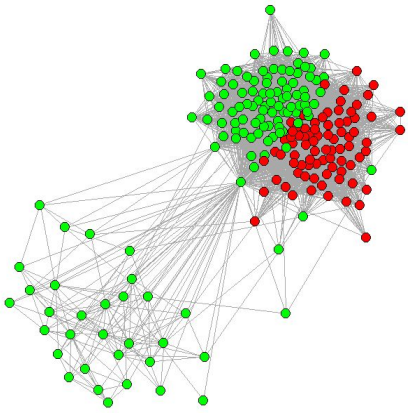
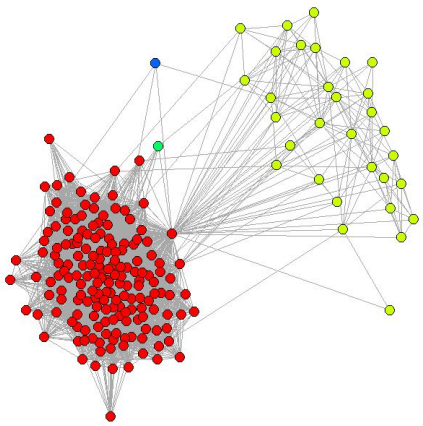
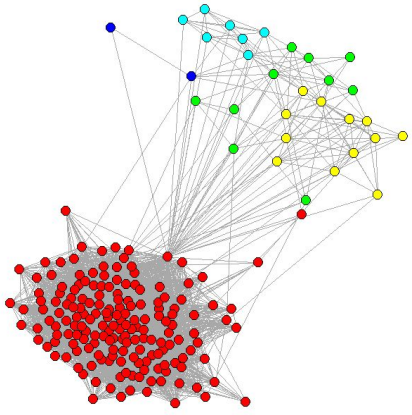
Average of Degree Distribution	14.5187319884726	51.1961722488038	28.0524017467249	37.1774891774892	70.2829268292683
Variance of Degree Distribution	240.481584514667	2215.95860602394	510.234084118593	373.659664972708	1548.81171688187

Core Node	*Community Detection Method - Modularity*		
	Fast Greedy	*Edge Betweenness*	*Infomap*
1	0.4418533	0.4161461	0.4180077
108	0.4581271	0.5213216	0.5211271
349	0.2456918	0.1505663	0.2448156
484	0.5342142	0.5154413	0.5434437
1087	0.1481956	0.0324953	0.02737159

Core Node	*Community Detection Method - Community Statistics*					
	Fast Greedy		*Edge Betweenness*		*Infomap*	
	Number of Communities	*Sizes of Communities*	*Number of Communities*	*Sizes of Communities*	*Number of Communities*	*Sizes of Communities*
1	26	52 39 121 18 52 24 10 8 3 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1	50	27 39 140 10 8 9 1 1 1 1 22 1 2 5 13 1 1 2 6 3 1 5 7 7 1 2 3 1 2 1 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 1 1 1 1 1	40	143 34 21 16 13 10 9 8 11 10 9 7 6 5 5 3 2 2 2 3 3 3 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
108	22	458 465 70 19 4 3 6 2 2 3	57	9 137 138 360 171 3 64 38 2	38	325 128 96 103 71 74 48 38 29

		2 1 1 1 1 1 1 1 1 1 1 1		12 7 1 1 13 4 3 10 6 3 14 2 1 5 4 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1		19 16 12 11 10 6 6 5 6 4 4 6 4 3 3 3 2 2 1 1 1 1 1 1 1 1 1 1 1
349	8	27 80 107 10 2 1 1 1	103	2 1 10 2 1 1 85 1 1 2 18 1 1 1 1 1 1 1 1 1 2 1 1 3 1 2 1 1 1 1 1 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	15	117 29 20 15 10 11 7 6 4 3 2 2 1 1 1
484	3	71 71 89	11	74 79 1 1 1 69 1 1 2 1 1	4	85 73 71 2
1087	2	80 125	4	172 31 1 1	5	174 12 10 7 2

Core Node	*Community Detection Method - Visualization*		
	Fast Greedy	*Edge Betweenness*	*Infomap*
1			
108			

349	FastGreedy 349 	EdgeBetweenness 349 	Infomap 349 
484	FastGreedy 484 	EdgeBetweenness 484 	Infomap 484 
1087	FastGreedy 1087 	EdgeBetweenness 1087 	Infomap 1087 

- ## Characteristic of nodes in the personalized network

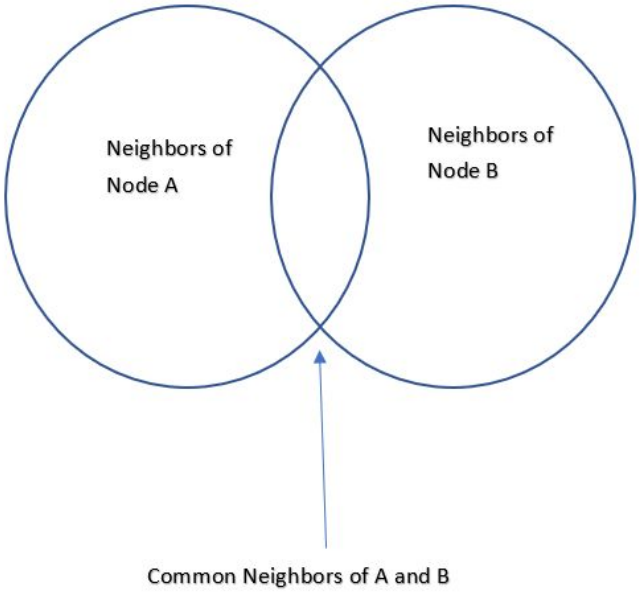
  In this part, we will explore characteristics of nodes in the personalized network using two measures. These two measures are stated and defined below:

  - **Embeddedness** of a node is defined as the number of mutual friends a node shares with the core node. It has the property of capturing the extent to which the two person's social circles overlap. One of the papers (https://arxiv.org/pdf/1310.6753.pdf) that we referred had a central finding that embeddedness is in fact a comparatively weak means of characterizing romantic relationships.
  - **Dispersion** of a node is defined as the sum of distances between every pair of the mutual friends the node shares with the core node. The distances should be calculated in a modified graph where the node (whose dispersion is being computed) and the core node are removed. We look not just at the number of mutual friends of two people, but also at the network structure on these mutual friends. Roughly, a link between two people has high dispersion when their mutual friends are not well connected to one another. In the above mentioned paper, it was observed that dispersion measure has roughly twice the accuracy of embeddedness in identifying the partner from among the user's full set of friends.

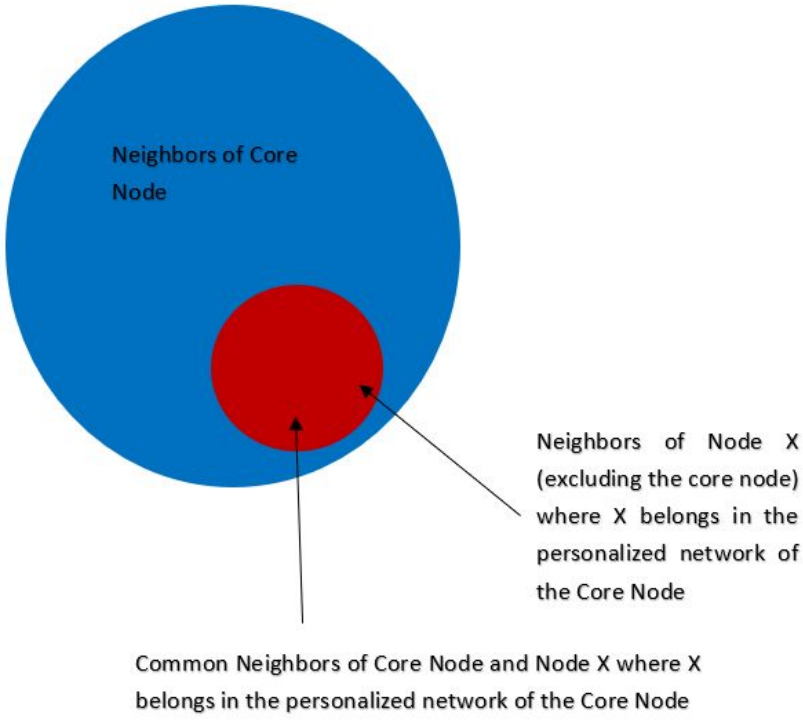
11) Write an expression relating the Embeddedness of a node to it's degree.

Ans:

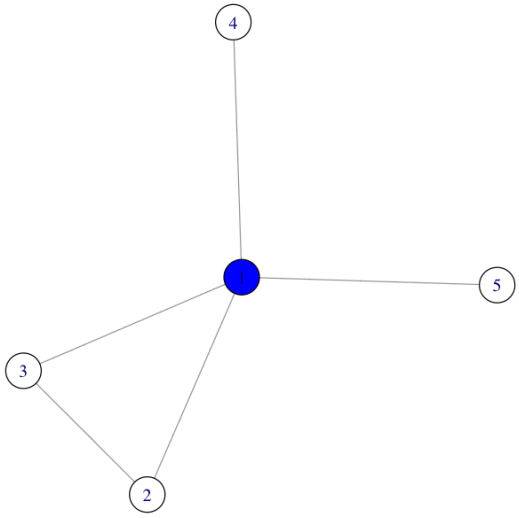
From the definition of embeddedness provided in the project statement, we can say that the embeddedness of node A wrt node B is the intersection of the neighbors of both the nodes. We can represent it graphically using the Venn Diagram as shown below:



In a personalized network, all the neighbors of any node X will belong in the set of neighbors of the core node. Hence, our venn diagram now looks as follows:



To further understand, let us consider the following personalized graph:



Here, the neighbors of the Core Node are {2, 3, 4, 5}. If we want to find the embeddedness of node 3, we first find its neighbors which are {1, 2} and then take the intersect of these two neighbor lists to find the mutual friends.

$$Mutual\ Friends = \{2, 3, 4, 5\} \cap \{1, 2\} = \{2\}$$

Hence, embeddedness is the length of the set of mutual friends which is |{2}| = 1.

The degree of node 3 is 2 which includes a connection to the mutual friend and one connection to the core node. Hence, the embeddedness in terms of degree will be degree(3) - 1.

This can be further extended to any node in a personalized graph, i.e., in a personalized graph the degree of any node apart from the core node will consist of all connections to the mutual friends and an extra connection to the core node.

For a personalized network, the embeddedness of a node can be given as

$$Degree(Node) - 1$$

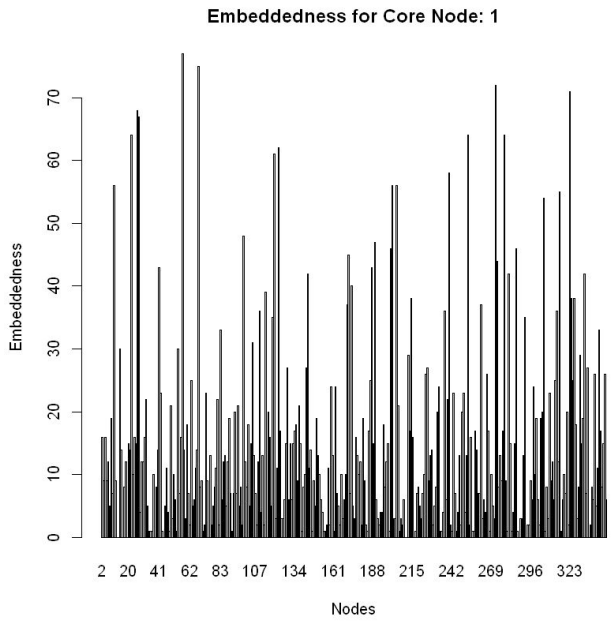
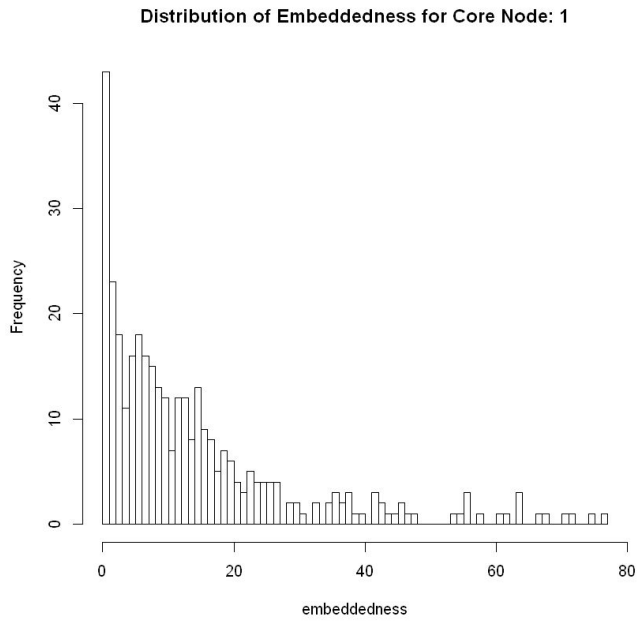
12) For each of the core node's personalized network (use the same core nodes as question 9), plot the distribution of embeddedness and dispersion. In this question, you will have 10 plots.

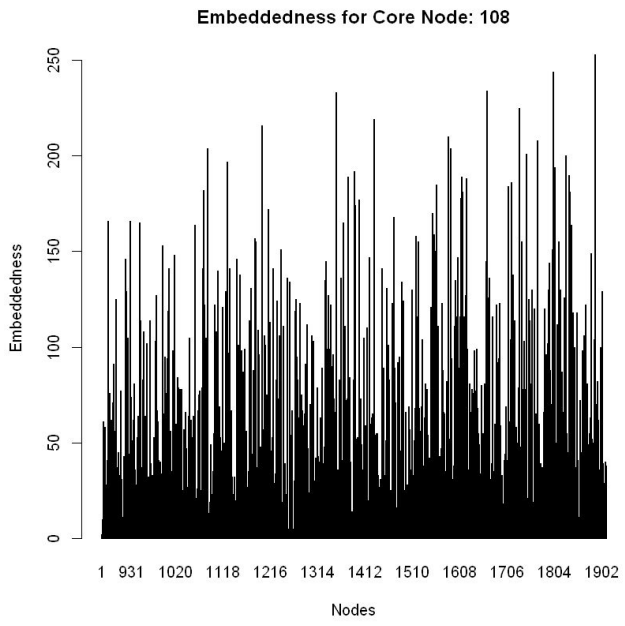
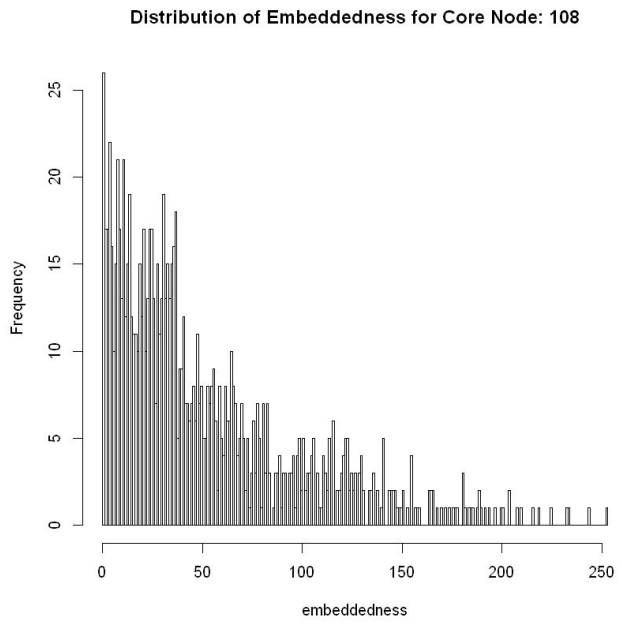
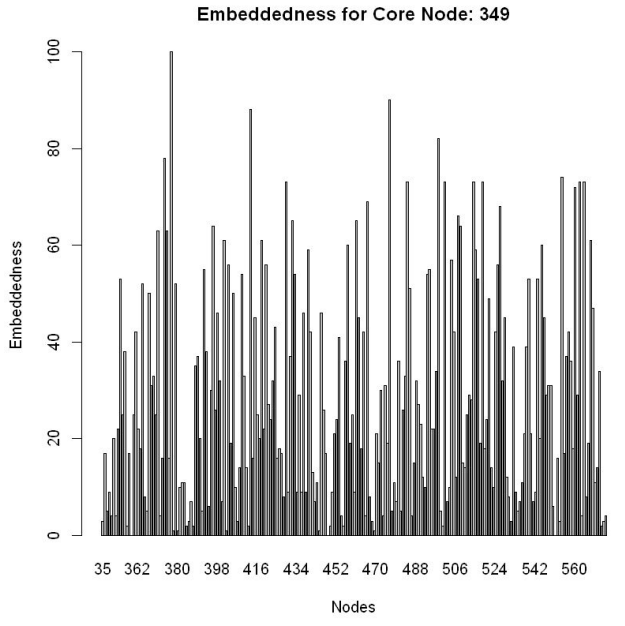
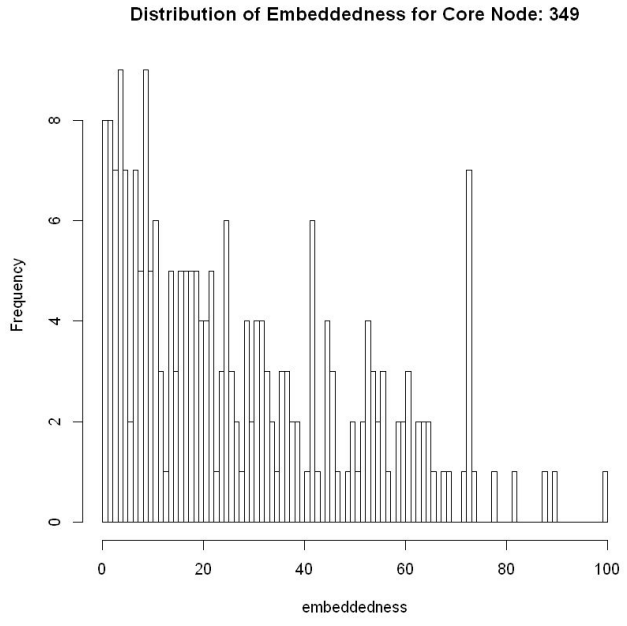
Ans

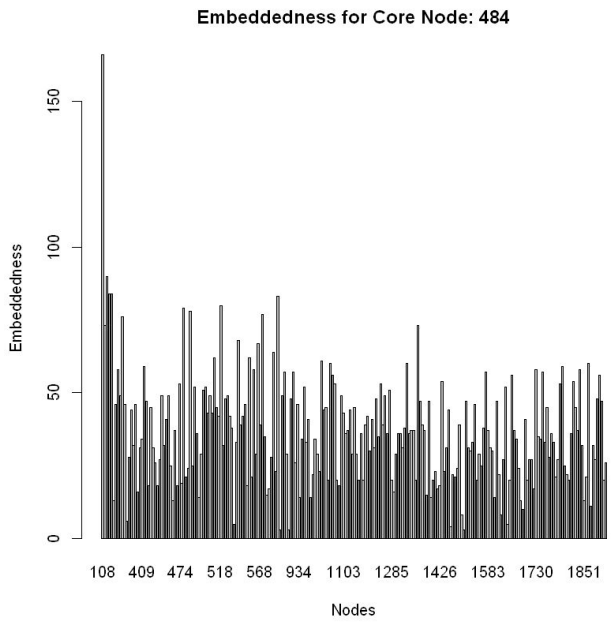
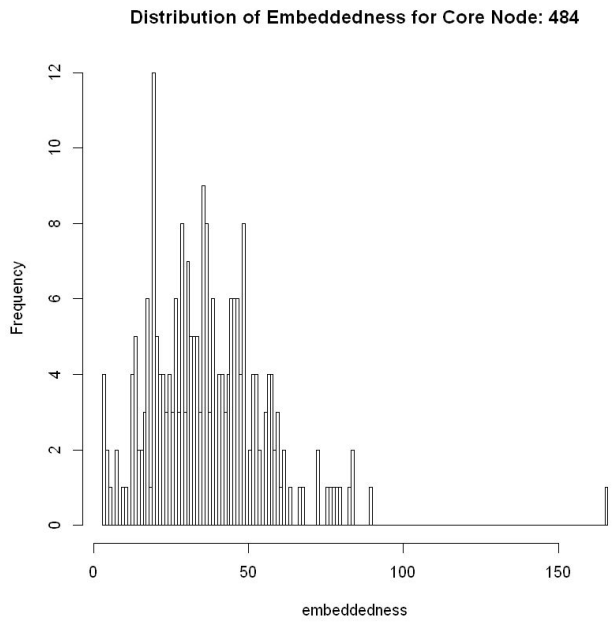
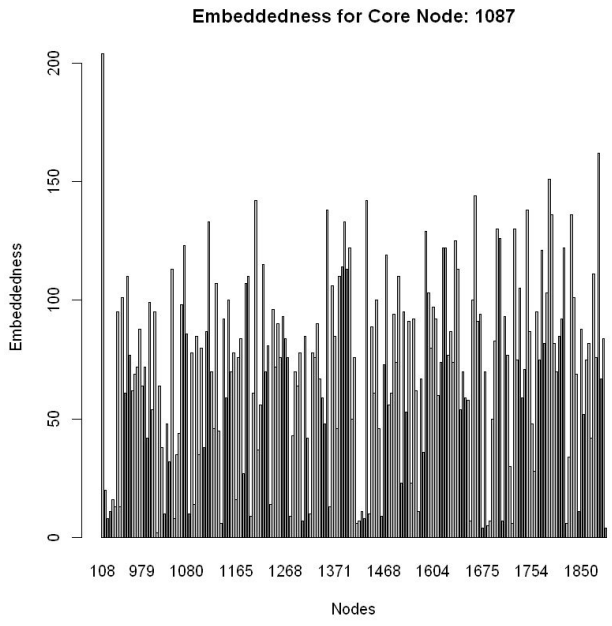
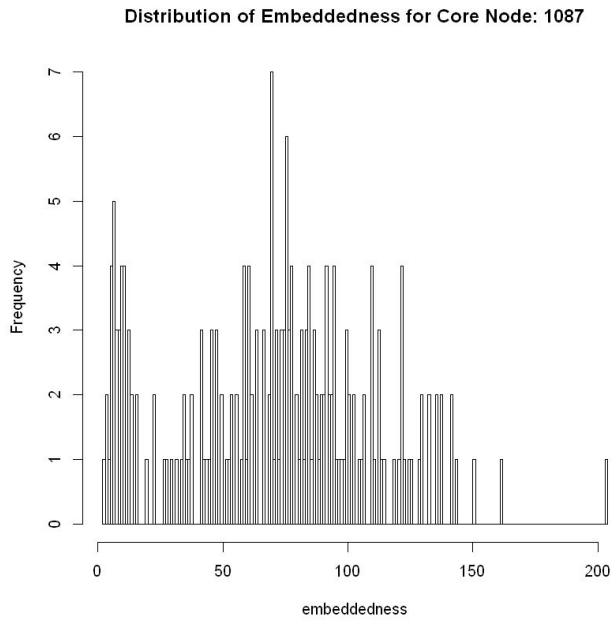
For each core node's personalized network we have done an analysis of the distribution of embeddedness and dispersion. We have found the embeddedness and dispersion of each node in the network and the bar plot in the below table shows the value corresponding to each node. To get better insight we have plotted a

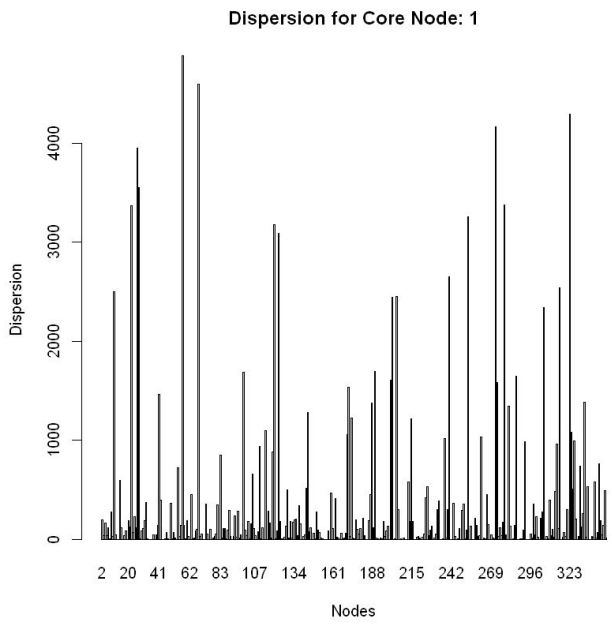
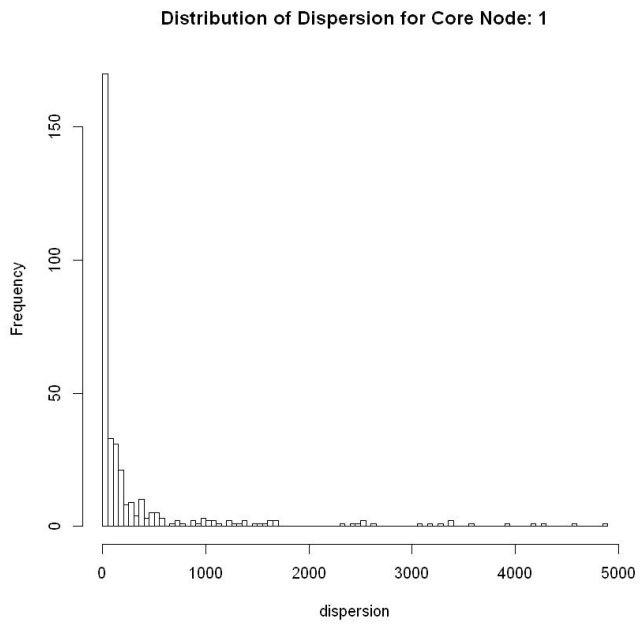
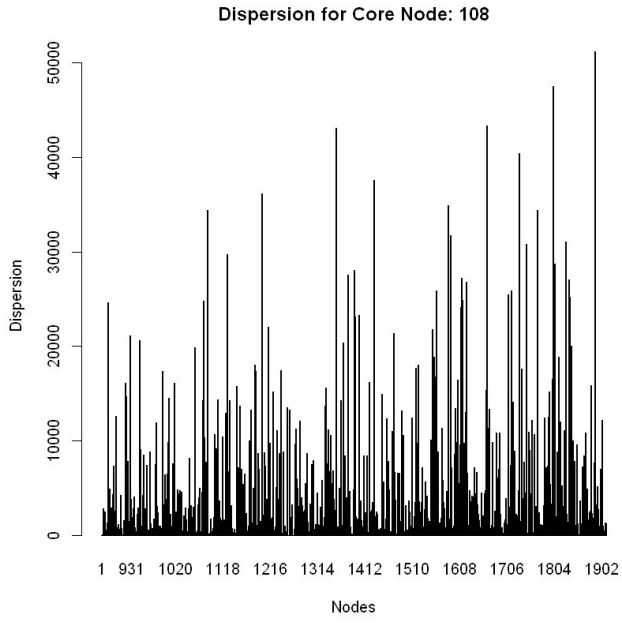
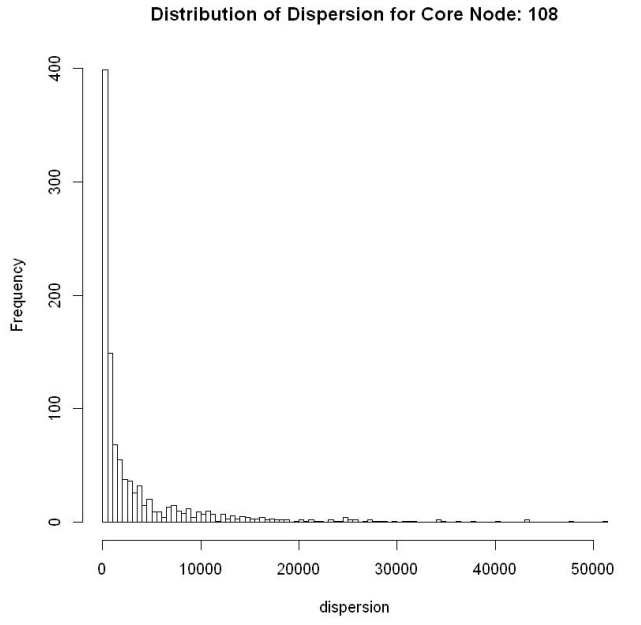
histogram with respect to the embeddedness/dispersion values to check the frequency of nodes with particular embeddedness/dispersion values and observe the structure of the histogram to get an idea about the network as a whole.

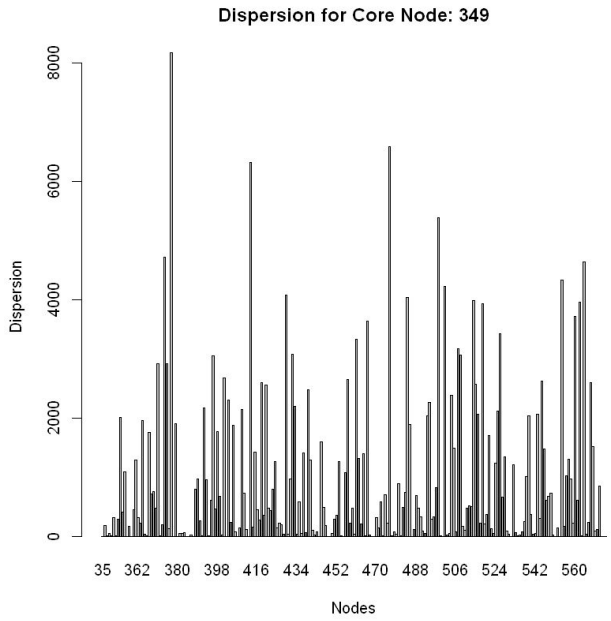
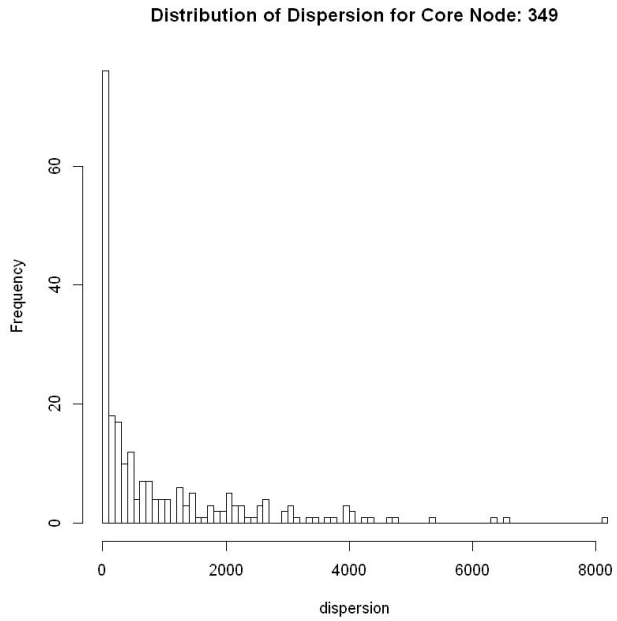
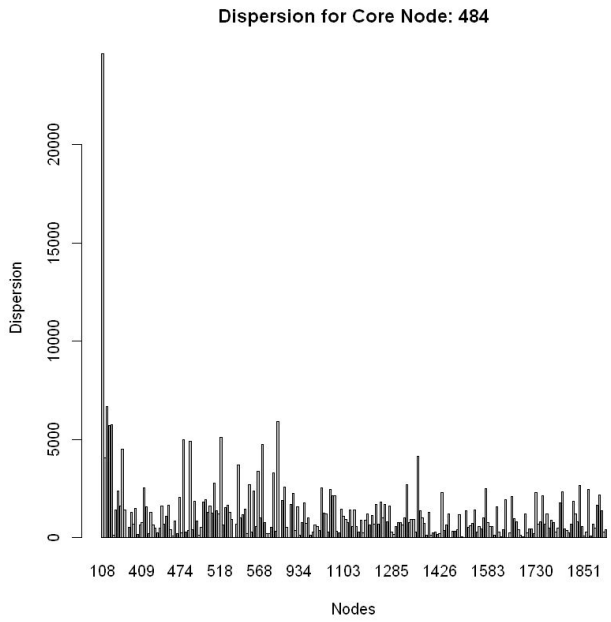
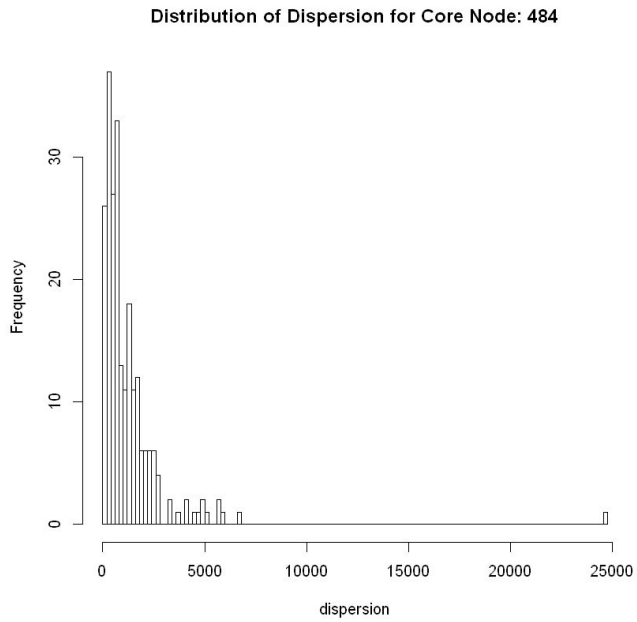
We see from the dispersion histograms for each personalized network of core node that the nodes with very high values of dispersion are not frequently encountered. A majority of the nodes have dispersion values on the lower side. This is because the networks are well connected and hence many nodes have lower dispersion values.

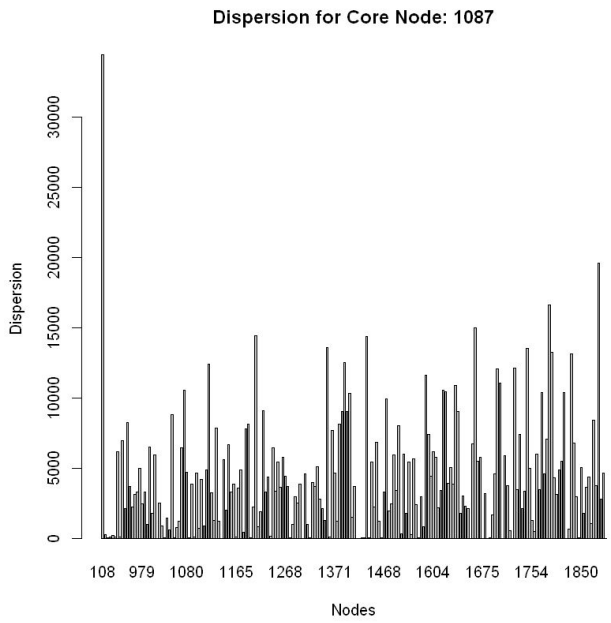
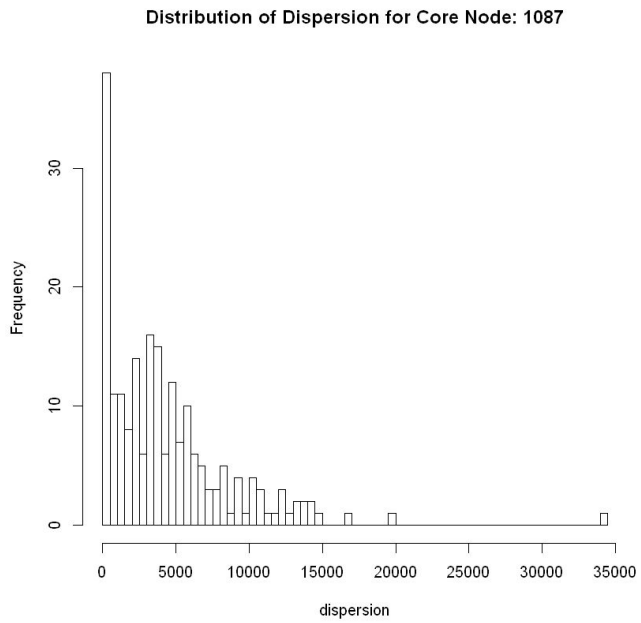
Core Node	*Bar Plot of Embeddedness Values*	*Distribution of Embeddedness*
1		

108	 Embeddedness for Core Node: 108 Embeddedness 0 50 100 150 200 250 1 931 1020 1118 1216 1314 1412 1510 1608 1706 1804 1902 Nodes	 Distribution of Embeddedness for Core Node: 108 Frequency 0 5 10 15 20 25 0 50 100 150 200 250 embeddedness
349	 Embeddedness for Core Node: 349 Embeddedness 0 20 40 60 80 100 35 362 380 398 416 434 452 470 488 506 524 542 560 Nodes	 Distribution of Embeddedness for Core Node: 349 Frequency 0 2 4 6 8 0 20 40 60 80 100 embeddedness

484	 Embeddedness for Core Node: 484 Embeddedness 0 50 100 150 108 409 474 518 568 934 1103 1285 1426 1583 1730 1851 Nodes	 Distribution of Embeddedness for Core Node: 484 Frequency 0 2 4 6 8 10 12 0 50 100 150 embeddedness
1087	 Embeddedness for Core Node: 1087 Embeddedness 0 50 100 150 200 108 979 1080 1165 1268 1371 1468 1604 1675 1754 1850 Nodes	 Distribution of Embeddedness for Core Node: 1087 Frequency 0 1 2 3 4 5 6 7 0 50 100 150 200 embeddedness

Core Node	*Bar Plot of Dispersion Values*	*Distribution of Dispersion*
1		
108		

349	 Dispersion for Core Node: 349 Dispersion 0 2000 4000 6000 8000 35 362 380 398 416 434 452 470 488 506 524 542 560 Nodes	 Distribution of Dispersion for Core Node: 349 Frequency 0 20 40 60 0 2000 4000 6000 8000 dispersion
484	 Dispersion for Core Node: 484 Dispersion 0 5000 10000 15000 20000 108 409 474 518 568 934 1103 1285 1426 1583 1730 1851 Nodes	 Distribution of Dispersion for Core Node: 484 Frequency 0 10 20 30 0 5000 10000 15000 20000 25000 dispersion

1087		

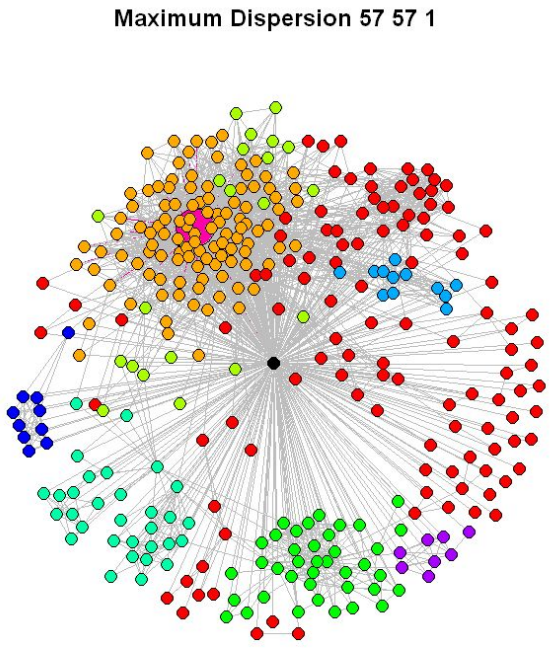
13) For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node. To detect the community structure, use FastGreedy algorithm. In this question, you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

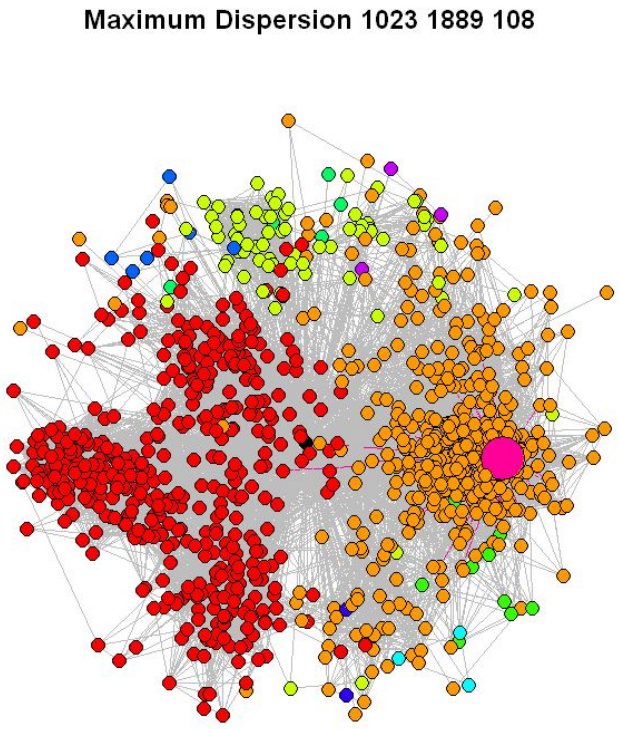
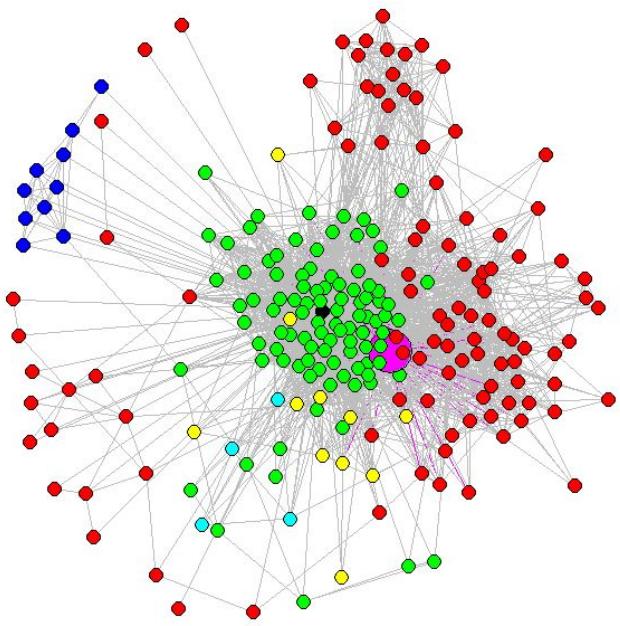
Ans: Dispersion looks not just at the number of mutual friends of two people, but also at the network structure of the mutual friends of the two people. A high dispersion for a node indicates that the mutual friends of the node and the core node are not well connected to one another.

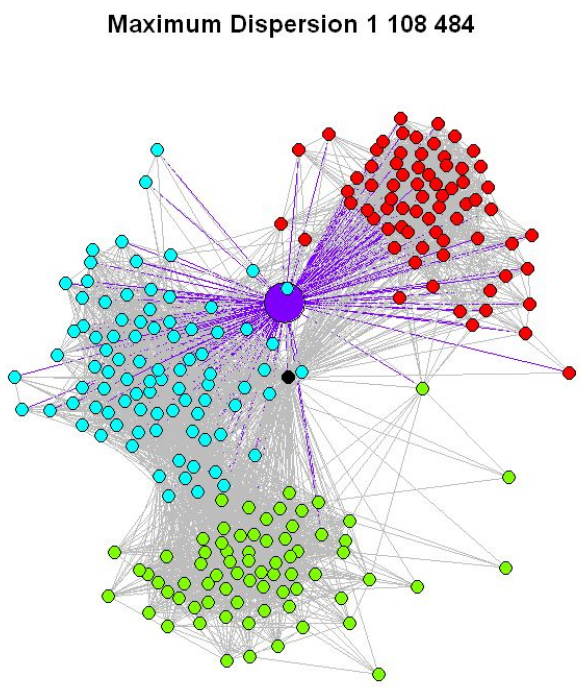
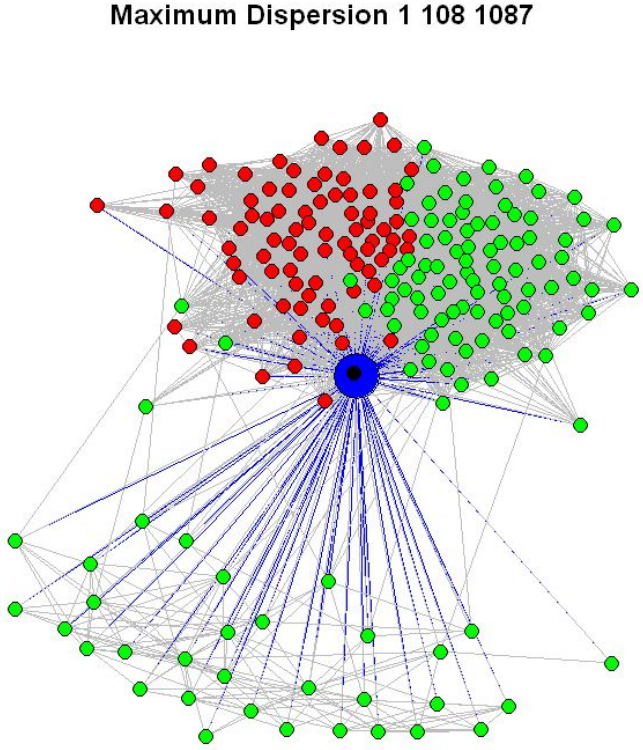
Dispersion is the sum of distances between every pair of mutual friends in the graph obtained after removing the core node and the node under consideration. As these nodes are removed, it is possible that some nodes might lose a path between them. In this case, as the path does not exist, the shortest path between such 2 nodes is infinity. We have handled this case specially by assigning a sufficiently large value to the distance between such nodes. The numerically maximum possible distance between any 2 nodes in a graph network is equal to the diameter of the graph. So for nodes with shortest path as infinity we have assigned a value of (diameter + 100) as the value of shortest path. The maximum value of diameter for a graph can be the number of vertices in the graph. As the vertices of this graph are the mutual

friends between the core node and the target node, the number of vertices in this graph or its max diameter is actually equal to the embeddedness.

The graph networks in the table below visually show all the networks. In the graph, the nodes belonging to the same community have the same color. Nodes belonging to different communities have different color. The core node is represented differently as a black color node. For each personalized network, the node with maximum dispersion is represented as a node with a larger size for easier visibility. The edges incident to this node are also represented with the same color as this node. Sometimes these edges may not be visible because they might be hidden behind some other nodes. Rest of the edges are grey in color. The table gives the number of the node in the personalized network as well as its number in the main Facebook network.

Core Node	*Maximum Dispersion*	*Node With Maximum Dispersion*		*Visualization*
		wrt Facebook Network	*wrt Personalized Network*	
1	4882	57	57	

108	51167	1889	1023	
349	8174	377	33	

484	24644	108	1	
1087	34412	108	1	

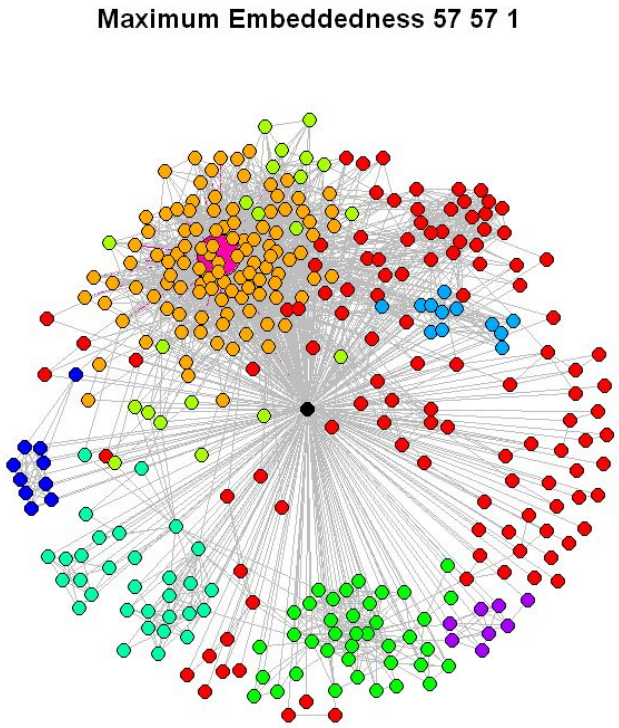
14) Repeat question 13, but now highlight the node with maximum embeddedness and the node with maximum value for (dispersion/ embeddedness). Also, highlight the edges incident to these nodes

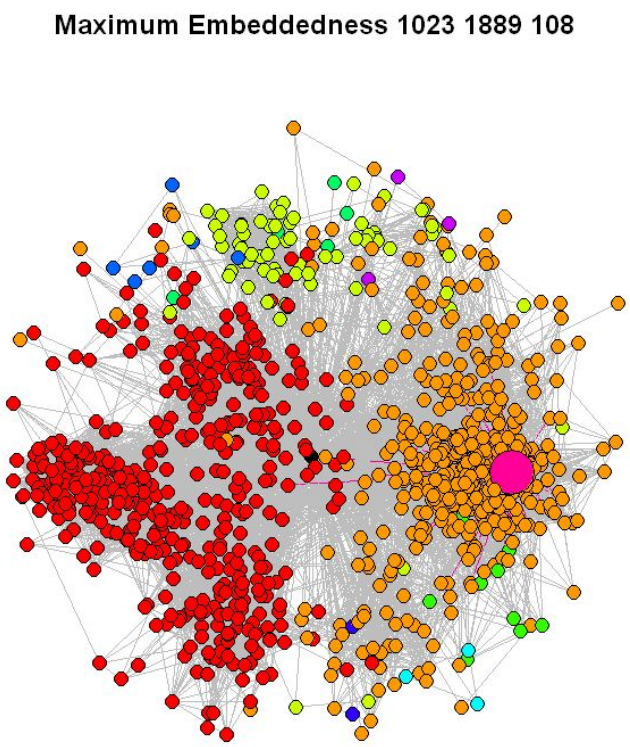
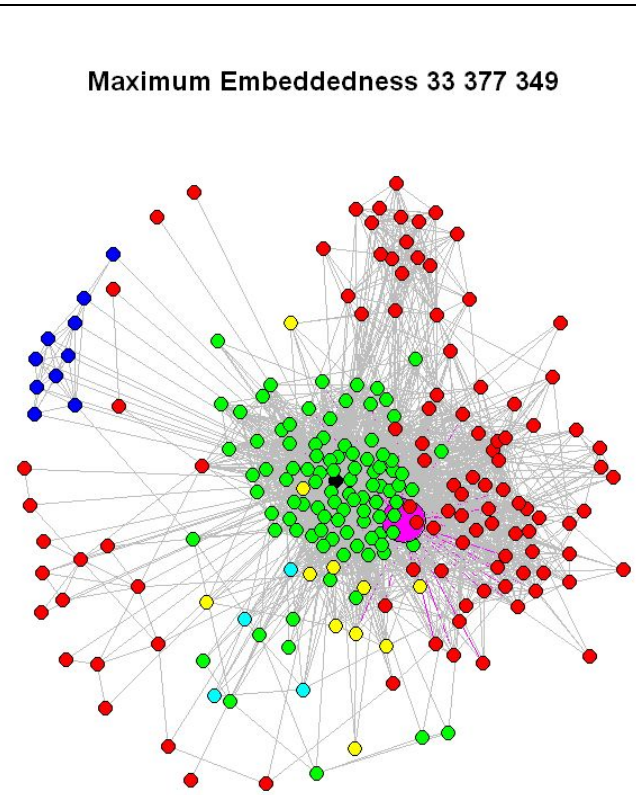
Ans:

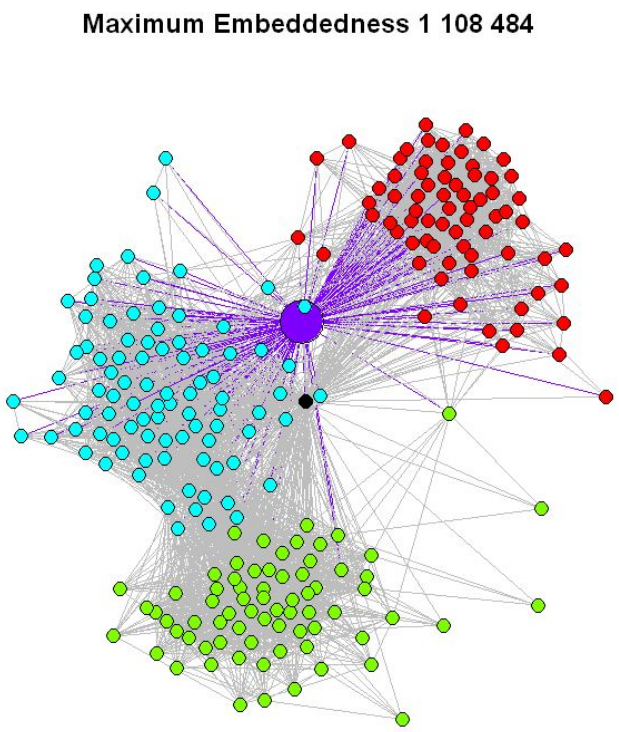
The ratio of dispersion and embeddedness is called normalized dispersion since it normalizes the absolute dispersion by the embeddedness. The ratio can be very useful in comparing the relation of two nodes as it can provide a normalized metric for comparing the mutual connectivity for different pairs of nodes.

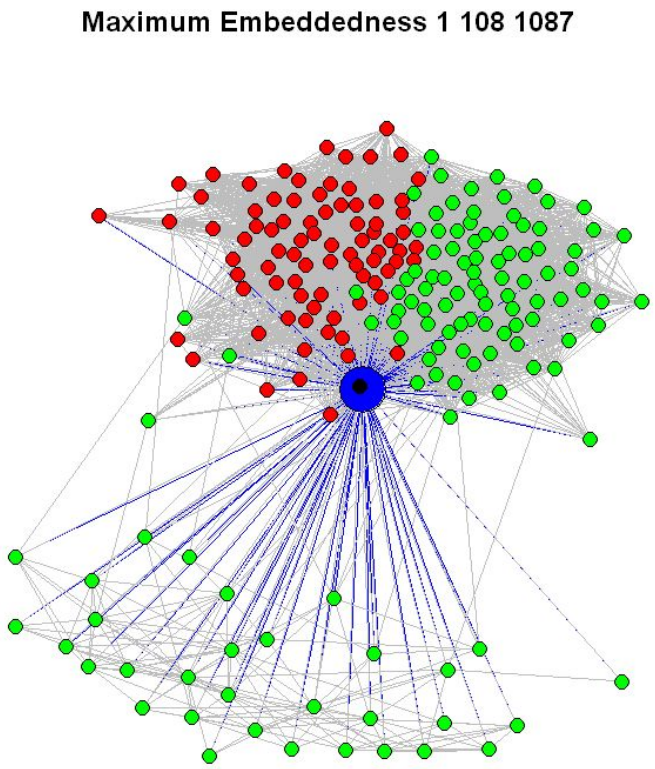
A special case in this is when the embeddedness is 0. If embeddedness is 0 then the denominator of ratio becomes 0 causing the ratio to be infinite. We handle this case by setting the ratio to 0 if embeddedness is 0. This is equivalent to skipping the node.

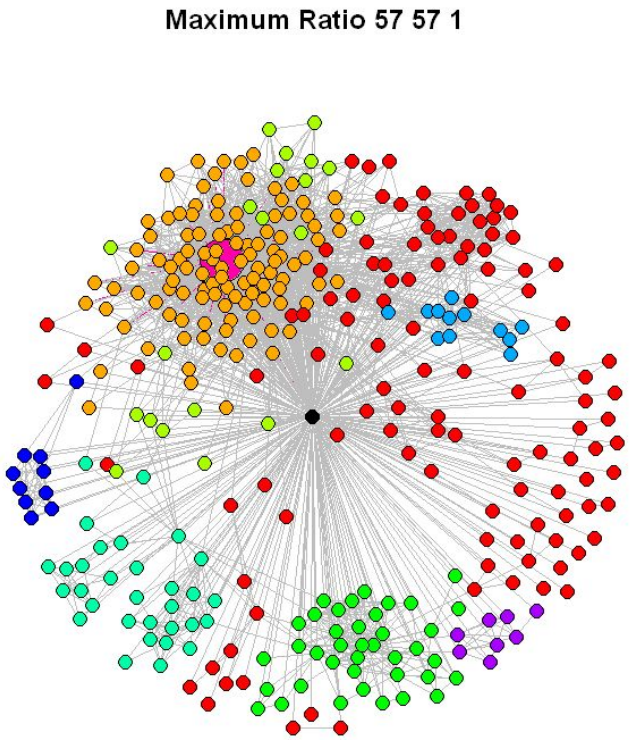
The graph networks in the table below visually show all the networks. In the graph, the nodes belonging to the same community have the same color. Nodes belonging to different communities have different color. The core node is represented differently as a black color node. For each personalized network, the node with maximum embeddedness/ratio of dispersion to embeddedness is represented as a node with a larger size for easier visibility. The edges incident to this node are also represented with the same color as this node. Sometimes these edges may not be visible because they might be hidden behind some other nodes. Rest of the edges are grey in color. The table gives the node number of this node in the personalized network as well as its number in the main Facebook network.

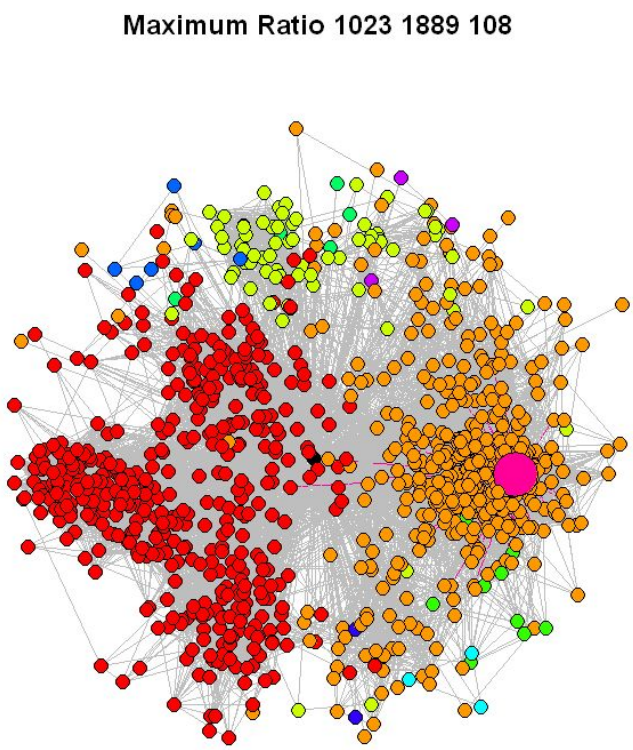
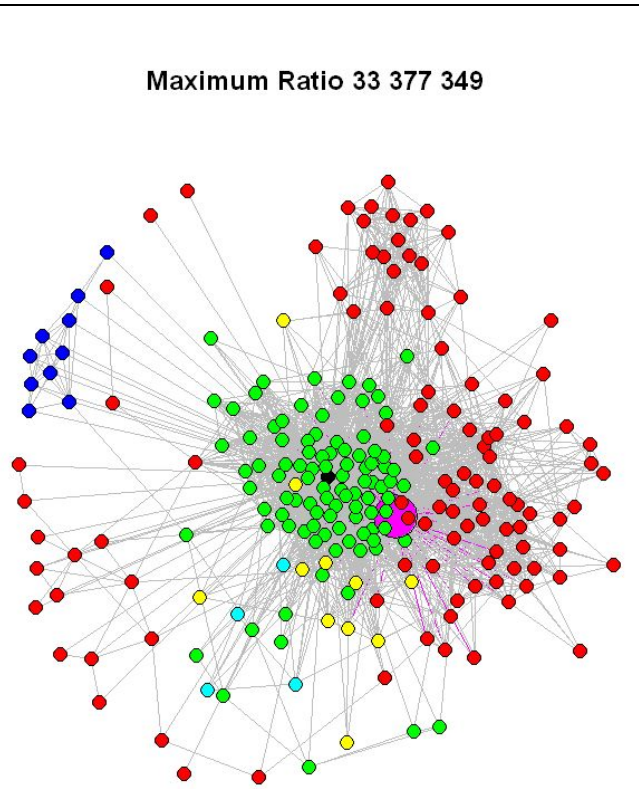
Core Node	*Maximum Embeddedness*	*Node With Maximum Embeddedness*		*Visualization*
		wrt Facebook Network	*wrt Personalized Network*	
1	77	57	57	

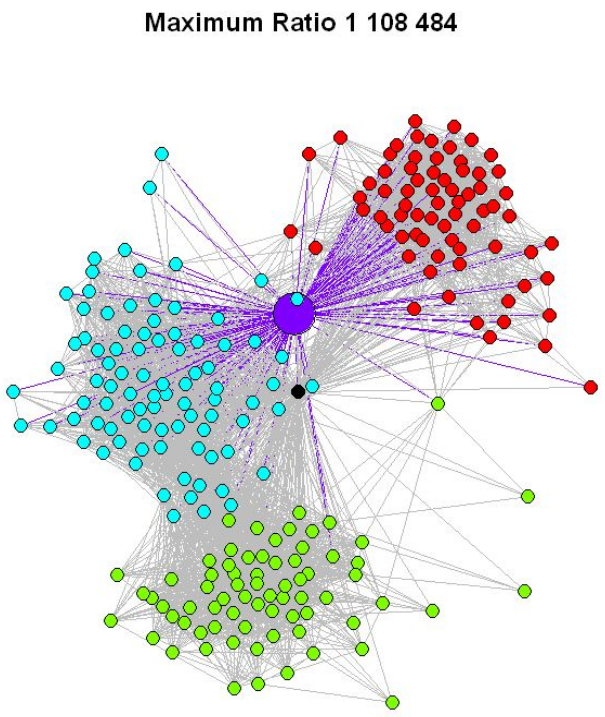
108	253	1889	1023	
349	100	377	33	

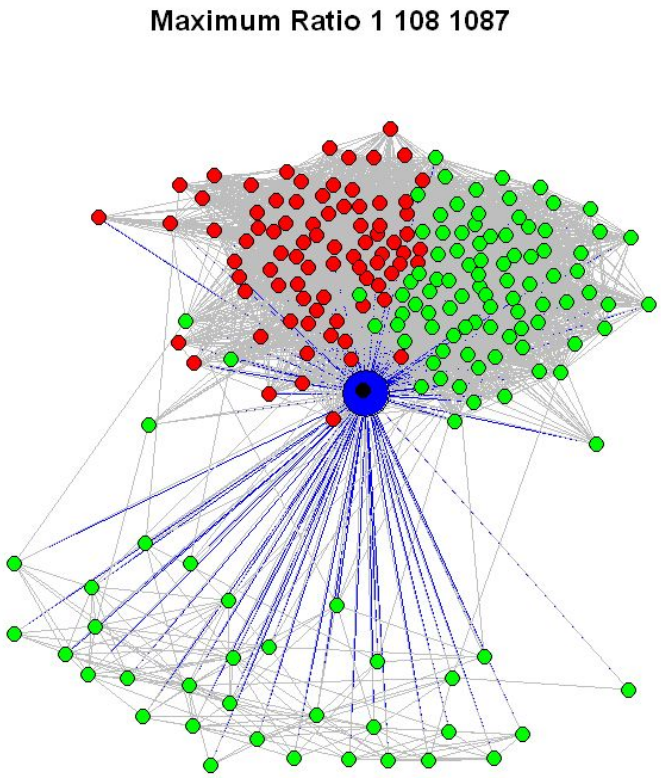
484	166	108	1	

1087	204	108	1	

Core Node	*Maximum Ratio*	*Node With Maximum Ratio*		*Visualization*
		wrt Facebook Network	*wrt Personalized Network*	
1	63.402597 4025974	57	57	

108	202.24110 6719368	1889	1023	
349	81.74	377	33	

484	148.45783 1325301	108	1	

1087	168.686274509804	108	1	

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

Ans:

One important observation from part 13 and 14 is that we get the same node in all 3 cases - that is, in each personalized network, the node with the maximum dispersion, maximum embeddedness and maximum ratio is the same.

High embeddedness implies many links lying within communities but few links lying between communities (extent to which the neighbors of a node belongs to the same community as the node itself). A **high dispersion** between the nodes indicates that the induced graph on their mutual neighbourhood has poor connectivity. The ratio of dispersion and embeddedness is called normalized dispersion since it normalizes the absolute dispersion by the embeddedness. The ratio can be very useful in comparing the relation of two nodes as it can provide a

normalized metric for comparing the mutual connectivity for different pairs of nodes.

So from the above facts we can say that the node having the highest embeddedness has a high number of nodes which lie in its community and are also present in the personalized network of the core node. As this node also has the highest dispersion among all nodes, the neighbors of this node are not as well connected as the neighbors of the other nodes. We see the same node as the node with the highest dispersion/embeddedness ratio, implying that even after the normalization this node has a higher dispersion than all other nodes in the personalized graph,

Analysis of the network based on dispersion, embeddedness and ratio values

If the embeddedness of a node in a graph is 'e', that means there are 'e' mutual friends between the core node and this node. A lower value of dispersion implies that the network of mutual friends is well connected. The minimum value of dispersion for this case will be when the graph of mutual friends is fully connected and maximum value of dispersion will be when the graph is fully isolated or none of the mutual friends are connected. If any 2 nodes are not connected, as per the previous part we have taken dispersion as (e+100). Therefore,

Minimum value of dispersion will be $1 \times {}^{e}C_2$ and Maximum value will be $(e + 100) \times {}^{e}C_2$

One more important value in this is a graph which has no isolated nodes. In this case let us consider the upper bound which can be when all the nodes are at a diameter distance away from each other (Although all nodes being at a diameter distance away from each other is not possible, we consider this hypothetical case as the upper bound). Here as diameter=e, this value will be $e \times {}^{e}C_2$.

Using this, Let us see what these values are for the graphs with core node 1 & 484

Core node	**Maximum embeddedness**	**Value of dispersion**		
		Minimum possible	**Maximum possible**	**Observed maximum**
1	77	2926	517902	4882
484	166	13695	3642870	24644

As we see from the above table, the maximum value of dispersion in the graphs is close to the minima as compared to the maxima. This tells us that even though the network is not fully connected, it is well connected to get such a low dispersion value.

4. Friend recommendation in personalized networks

In many social networks, it is desirable to predict future links between pairs of nodes in the network. In the context of this Facebook network it is equivalent to recommending friends to users. In this part of the project, we will explore some neighborhood-based measures for friend recommendation. The network that we will be using for this part is the personalized network of node with ID 415.

- **Neighborhood based measure**

  In this project, we will be exploring three different neighborhood-based measures. Before we define these measures, let's introduce some notation:

  - **Si is the neighbor set of node i in the network.**
  - **Sj is the neighbor set of node j in the network.**

- **Friend recommendation using neighborhood based measures**

  We can use the neighborhood based measures defined in the previous section to recommend new friends to users in the network. Suppose we want to recommend 't' new friends to some user 'i' in the network using Jaccard measure. We follow the steps listed below:

  1. For each node in the network that is not a neighbor of 'i', compute the jaccard measure between the node i and the node not in the neighborhood of 'i'

     $$\text{Compute Jaccard}(i, j)\ \forall j \in S_i^{\,C}$$

  2. Then pick t nodes that have the highest jaccard measure with node i and recommend these nodes as friends to node 'i'

- **Creating a list of users**

  Having defined the friend recommendation procedure, we can now apply it to the personalized network of node ID 415. Before we apply the algorithm, we need to create the list of users who we want to recommend new friends to. We create this list by picking all nodes with degree 24. We will denote this list as N_r

16) What is $|N_r|$?

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| = 11 with the 11 users having node ID as below.

31 53 75 90 93 102 118 133 134 136 137

- ## Average accuracy of friend recommendation algorithm

In this part, we will apply the 3 different types of friend recommendation algorithms to recommend friends to the users in the list Nr. We will define an average accuracy measure to compare the performances of the friend recommendation algorithms. Suppose we want to compute the average accuracy of the friend recommendation algorithm. This task is completed in two steps:

1. Compute the average accuracy for each user in the list N_r.
2. Compute the average accuracy of the algorithm by averaging across the accuracies of the users in the list N_r

Let's describe the procedure for accomplishing the step 1 of the task. Suppose we want to compute the average accuracy for user i in the list N_r. We can compute it by iterating over the following steps 10 times and then taking the average:

1. Remove each edge of node i at random with probability 0.25. In this context, it is equivalent to deleting some friends of node 'i'. Let's denote the list of friends deleted as R_i.
2. Use one of the three neighborhood based measures to recommend $|R_i|$ new friends to the user 'i'. Let's denote the list of friends recommended as P_i.
3. The accuracy for the user 'i' for this iteration is given by $|P_i \cap R_i| / |R_i|$

By iterating over the above steps for 10 times and then taking the average gives us the average accuracy of user i. In this manner, we compute the average accuracy for each user in the list N_r. Once we have computed them, then we can take the mean of the average accuracies of the users in the list N_r. The mean value will be the average accuracy of the friend recommendation algorithm.

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?

- Neighborhood based methods are based on the idea that two nodes i and j are more likely to form a link if the set of neighbors S_i and S_j have large overlap. This follows the natural intuition that such node pairs represent authors with many colleagues in common, and hence are more likely to come into contact themselves.
- One can directly use this idea in Common neighbors measure by setting the measure as the number of common neighbors of node i and node j. 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.
- The Jaccard coefficient, commonly used in information retrieval measures the number of features that both node i and node j have compared to the number of features that either node i or node j has.
- Adamic Adar measure refines the simple counting of common features by weighting rarer features more heavily. The Adamic/Adar predictor formalizes the intuitive notion that rare features are more telling.

The below table lists the details of average accuracy of the friend recommendation algorithm. We have listed the accuracy of the algorithms for each of the users individually as well to get more insights.

User #	**Average accuracy over 10 iterations for each user**		
	Common Neighbors	**Jaccard**	**Adamic Adar**
31	0.41778	0.22869	0.30369
53	1	0.9181	0.98
75	0.92405	0.89393	0.83321
90	0.84437	0.7971	0.81548
93	0.41266	0.41556	0.45167
102	1	0.95321	0.98
118	0.8746	0.83813	0.93532
133	1	0.90381	0.99
134	1	0.96	1
136	0.93365	0.87905	0.87393
137	0.96111	0.92722	0.975

Algorithm Average	**0.851655844155844**	**0.792254361799816**	**0.830753968253968**

According to the above table, we see that the Common Neighbors measure in the algorithm works the best followed by Adamic Adar measure and lastly Jaccard measure. Another observation is that depending on each user, the algorithm accuracy varies. We see a drastic difference in this between user node ID 31 and 134 as highlighted in the table. This difference is due to some difference in correlation among the neighbors.

As there are 11 users involved, we ran all 3 algorithms multiple times to see something interesting. The below table lists the accuracies of the 3 runs.

We observed that in the multiple runs, algorithm when run with the Jaccard measure always gives the least accuracy. However the best algorithm varies between Common neighbors and Adamic adar measure. Sometimes common neighbors gives better accuracy than adamic adar and sometimes adamic adar gives better accuracy than common neighbors. The accuracies of the algorithms do not differ by large value.

Algorithm	**Run #1**	**Run #2**	**Run #3**
Common Neighbor	0.823383510428965	0.813257247802702	0.851655844155844
Jaccard	0.794440736031645	0.803473597109961	0.792254361799816
Adamic Adar	0.829745507018234	0.832964712055621	0.830753968253968

II. Google+ Network

In this part, we will explore the structure of the Google+ network.

The main feature of the Google+ dataset provided to us is the rich feature set and the presence of social circles. Social circles are one of the main mechanisms for users of various social networking sites to organize their networks and the content generated by them. These circles can be used in further applications such as content filtering, privacy, and sharing of groups of users.

The Google+ data contains data from 133 ego-networks, consisting of 479 circles and 106,674 users. The 133 ego-networks represent all Google+ users who had shared at least two circles. Additionally, the Google+ network is ***directed***.

There are six files associated with every user in the Google+ network:

- ***.circles***: contains the circle information for a user

- ***.edges***: contains the edge list of the users present in its network. ***It does not contain any edges from the user in question.***
- ***.featnames***: contains the names of the features associated with the network of the user
- ***.egofeat***: associates a vector with the user indicating whether a feature is applicable to it or not.
- ***.feats***: associates a vector with every neighbor of the user indicating whether a feature is applicable to them or not.
- ***.followers***: contains the followers of the user

18) How many personal networks are there?

Ans:

The project statement defines a personal network as a network of a user which contains more than 2 circles.

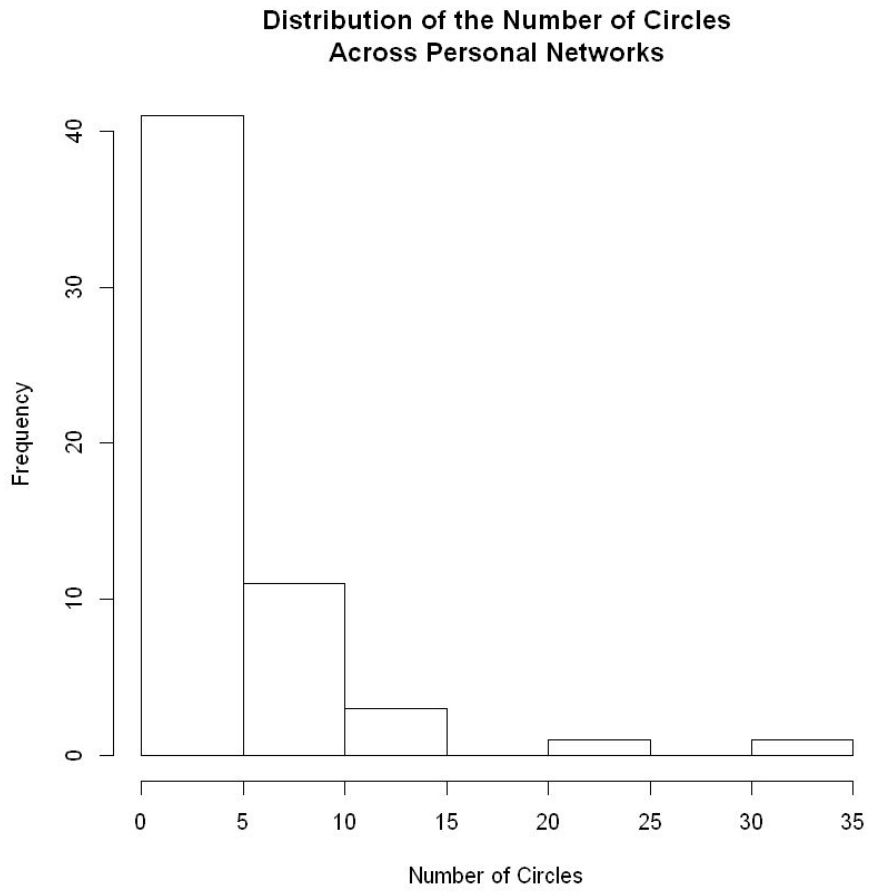
We can get circle information from the .circles file of the users. The circles file is organized as follows: each line of the file indicates one circle and contains a circle ID and the users present in the circle.

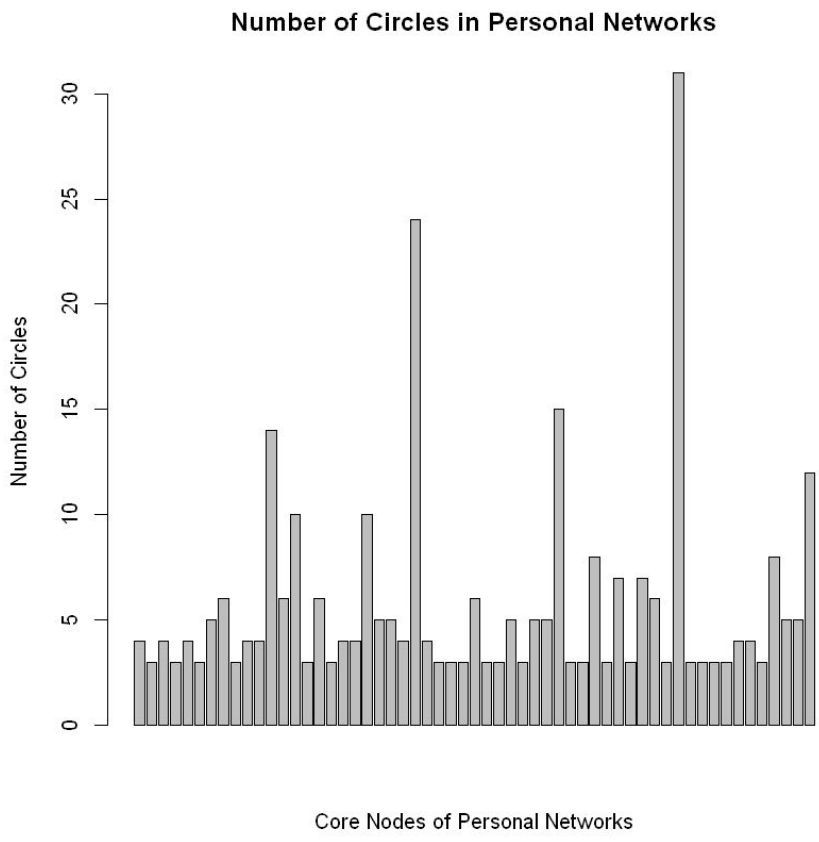
By scanning all the circle files, we found that 57 files had more than 2 circles the results of which have been summarized in the table below. Furthermore we analysed these results graphically as well.

Number of Personal Networks	57
Users having Personal Networks	1. 100535338638690515335 2. 10096287152568431589 7 3. 101130571432010257170 4. 10118574899692705993 1 5. 101263615503715477581 6. 101373961279443806744 7. 101541879642294398860 8. 101626577406833098387 9. 102170431816592344972 10. 102615863344410467759 11. 102778563580121606331 12. 103236949470535942612 13. 103892332449873403244 14. 104105354262797387583 15. 104607825525972194062

	16. 104672614700283598130 17. 104987932455782713675 18. 106186407539128840569 19. 106228758905254036967 20. 106382433884876652170 21. 106837574755355833243 22. 107040353898400532534 23. 107203023379915799071 24. 107223200089245371832 25. 107459220492917008623 26. 107489144252174167638 27. 108883879052307976051 28. 109327480479767108490 29. 109596373340495798827 30. 110538600381916983600 31. 110614416163543421878 32. 110701307803962595019 33. 110809308822849680310 34. 110971010308065250763 35. 111048918866742956374 36. 111091089527727420853 37. 112317819390625199896 38. 112724573277710080670 39. 113112256846010263985 40. 113356364521839061717 41. 113881433443048137993 42. 114147483140782280818 43. 115121555137256496805 44. 11536071097759949621 45. 115455024457484679647 46. 115625564993990145546 47. 116247667398036716276 48. 116315897040732668413 49. 116807883656585676940 50. 116825083494890429556 51. 116931379084245069738 52. 117412175333096244275 53. 117503822947457399073 54. 117668392750579292609 55. 117734260411963901771 56. 118107045405823607895 57. 118379821279745746467
Average Number of Circles	5.71929824561404
Minimum Number	3

of Circles	
User ID with Minimum Number of Circles	1. 100962871525684315897 2. 101185748996927059931 3. 101373961279443806744 4. 102170431816592344972 5. 104607825525972194062 6. 104987932455782713675 7. 107489144252174167638 8. 108883879052307976051 9. 109327480479767108490 10. 110538600381916983600 11. 110614416163543421878 12. 110809308822849680310 13. 112317819390625199896 14. 112724573277710080670 15. 113356364521839061717 16. 114147483140782280818 17. 115455024457484679647 18. 116247667398036716276 19. 116315897040732668413 20. 116807883656585676940 21. 116825083494890429556 22. 117503822947457399073
Maximum Number of Circles	31
User ID with Maximum Number of Circles	115625564993990145546

Distribution of the Number of Circles	

Visualization of the Number of Circles for every User	

We have further investigated our results by finding the finding the distribution of the number of circles of these nodes. We observe that maximum users have circles in the range of 3-5 with very few users having a larger number of circles.

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.

- **109327480479767108490**
- **115625564993990145546**
- **101373961279443806744**

Ans:

For constructing these personalized networks, we have used the information from the .edges file and .feats file.

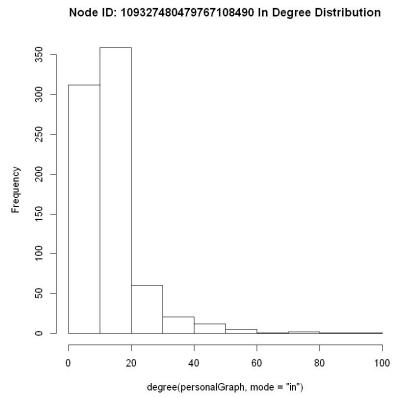
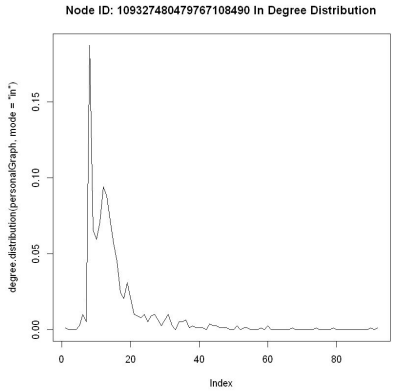
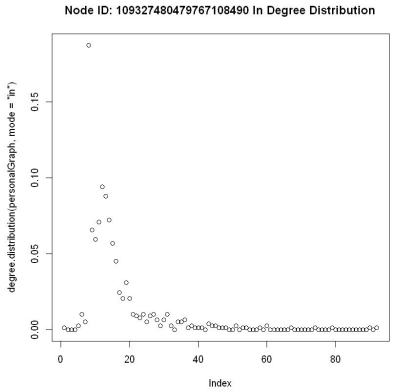
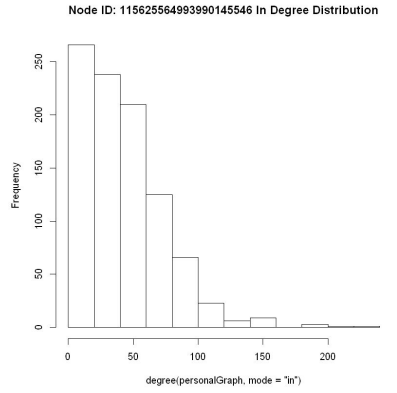
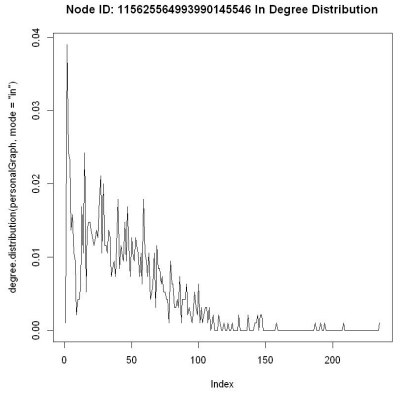
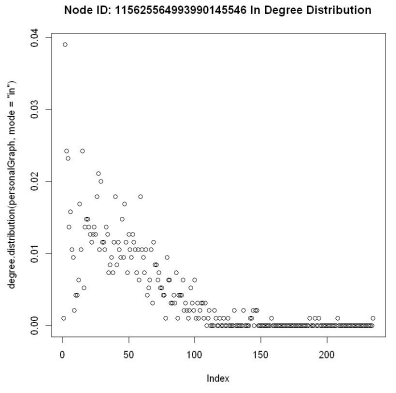
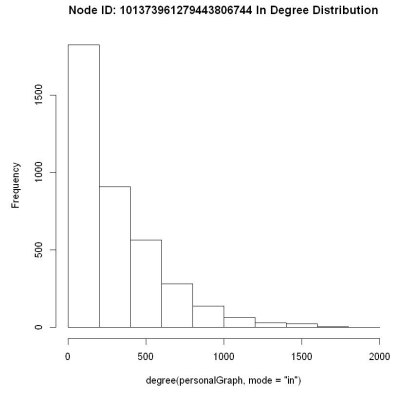
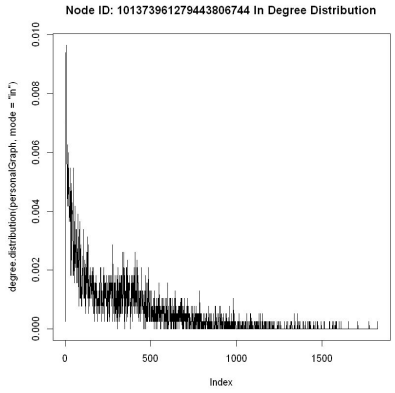
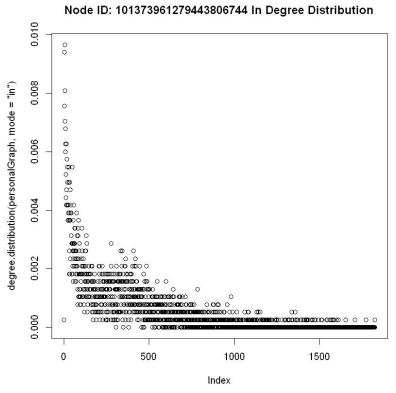
Before analyzing the degree distributions, we had a look at the networks constructed using these user IDs. we observe that these networks are connected, dense and have lower values of diameters which is the trend generally observed for social networks.

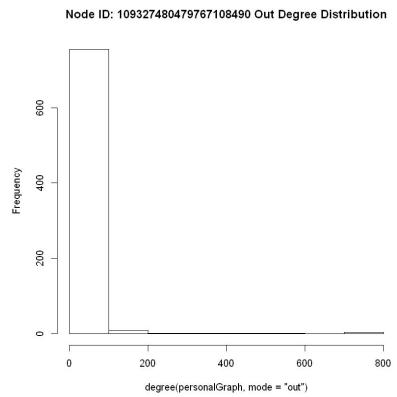
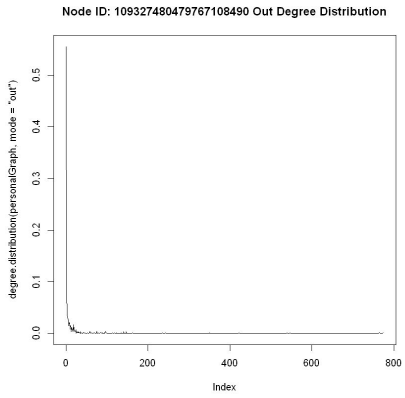
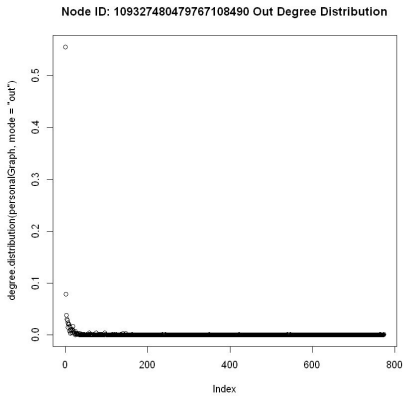
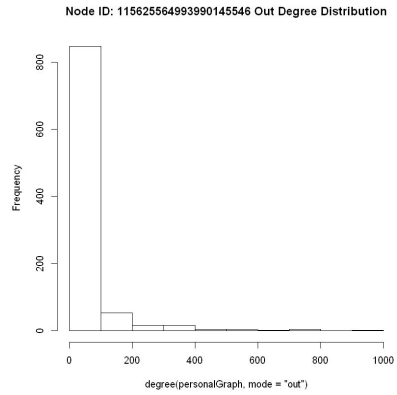
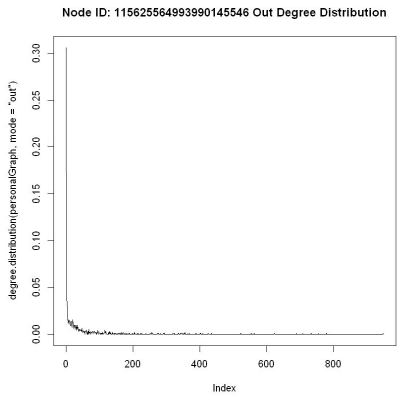
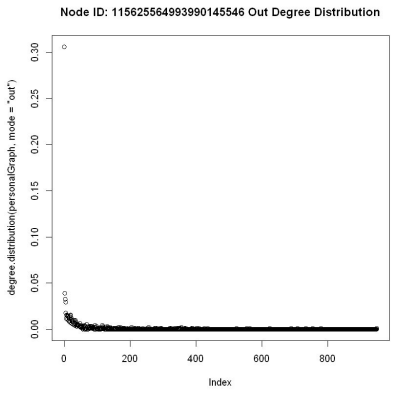
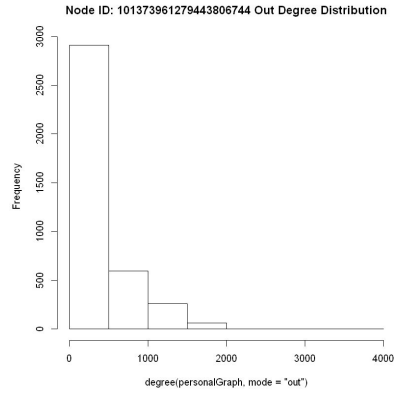
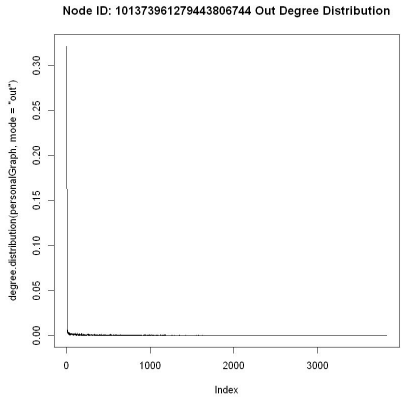
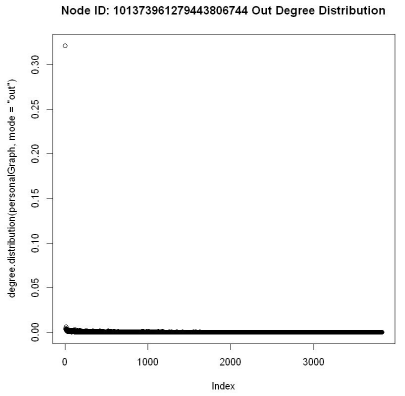
The network statistics are presented below:

Personal Network Statistics			
Network Statistics	*User ID*		
	109327480479767108490	*115625564993990145546*	*101373961279443806744*
Number of Vertices	774	948	3832
Number of Edges	10884	40347	1137338
Connected?	YES	YES	YES
GCC Size	774	948	3832
Diameter	5	7	5
Average of Degrees	28.1240310077519	85.120253164557	593.600208768267
Variance of Degrees	5345.40244893047	11911.8566960288	327685.922088184

We have constructed three plots - histogram, line plot, and a scatter plot to understand the nature of the degree distributions of these networks. We observe that all these three plots have similar degree distributions.

From the in-degree distributions we see that there are a large number of nodes which have lower degrees and as the degree increases, the number of nodes associated also decrease gradually. This is in contrast with the sharp decline observed for the number of nodes associated with larger degrees as observed in the out-degree distributions of the networks. The node distribution of real-world social networks such as Facebook and Google Plus follows a power law distribution which is evident in the degree distributions obtained.

In Degree Distribution			
User ID	*Histogram*	*Line Plot*	*Scatter Plot*
109327480479767108490			
115625564993990145546			
101373961279443806744			

Out Degree Distribution			
User ID	*Histogram*	*Line Plot*	*Scatter Plot*
10932748047976710849 0			
11562556499399014554 6			
10137396127944380674 4			

- **Community Structure of Personal Networks**

  In this part of the project, we will explore the community structure of the personal networks that we created and explore the connections between communities and user circles.

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.

Ans:

In this problem, we explore another community detection method called the walktrap community detection algorithm.

Similar to Infomap community detection algorithm, the walktrap algorithm approach is also based on random walks. The general idea behind this is that if we perform random walks on the graph, then these random walks tend to stay within the same community because there are very few edges connecting different communities.

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 the Fast Greedy approach.

Walktrap is an efficient hierarchical clustering algorithm that allows us to find community structures at different scales.

We start from a partition $P1 = \{\{v\}, v \in V\}$ of the graph into n communities reduced to a single vertex. We first compute the distances between all adjacent vertices. Then this partition evolves by repeating the following operations.

At each step k:

- choose two communities C1 and C2 in P_k according to a criterion based on the distance between the communities
- merge these two communities into a new community $C3 = C1 \cup C2$ and create the new partition: $P_{k+1} = (P_k \setminus \{C1, C2\}) \cup \{C3\}$

- update the distances between communities. After n − 1 steps, the algorithm finishes and we obtain P_n = {V }. Each step defines a partition P_k of the graph into communities, which gives a hierarchical structure of communities called dendrogram. This structure is a tree in which the leaves correspond to the vertices and each internal node is associated to a merging of communities in the algorithm: it corresponds to a community composed of the union of the communities corresponding to its children. The key points in this algorithm are the way we choose the communities to merge, and the fact that the distances can be updated efficiently.

  This choice plays a central role for the quality of the obtained community structure. In order to reduce the complexity, we will only merge adjacent communities (having at least an edge between them). This reasonable heuristic limits to m the number of possible mergings at each stage. Moreover it ensures that each community is connected. We choose the two communities to merge according to Ward's method. At each step k, we merge the two communities that minimize the mean σk of the squared distances between each vertex and its community.

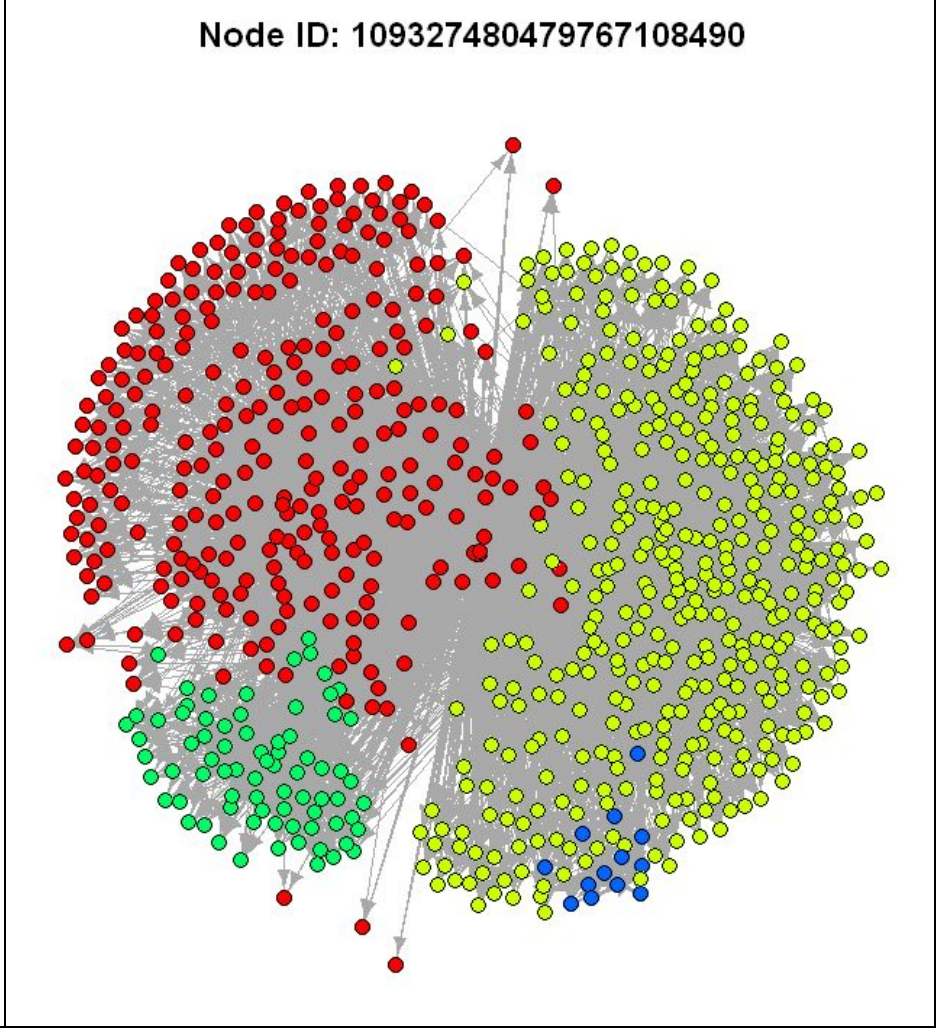
$$\sigma_k = \frac{1}{n} \sum_{C \in \mathcal{P}_k} \sum_{i \in C} r_{iC}^2$$

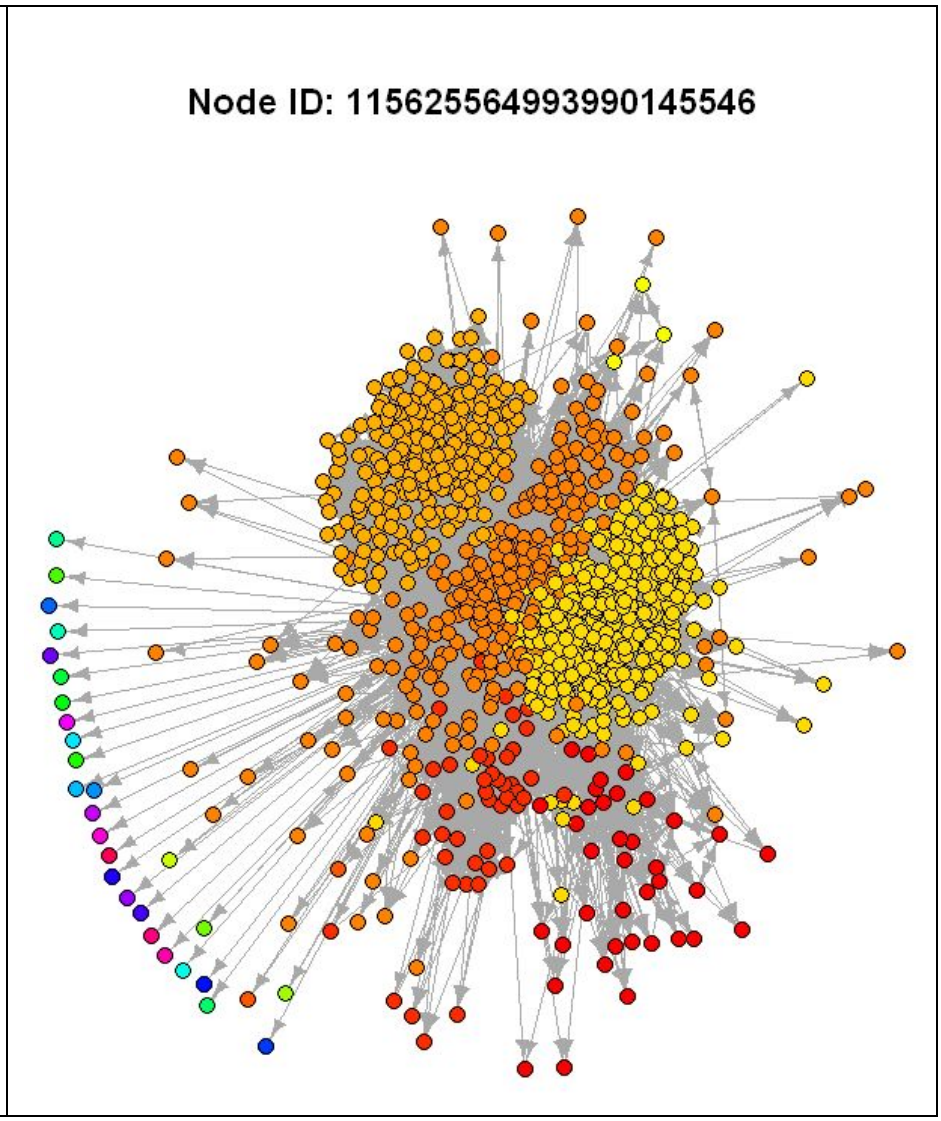
  On computing the modularities for the three selected users, we observe that the values are not particularly high implying that there are dense inter community connections. We observe the highest modularity for the personal graph associated with user ID *1156255649939901455546* and the lowest for the personal graph associated with user ID *101373961279443806744*.

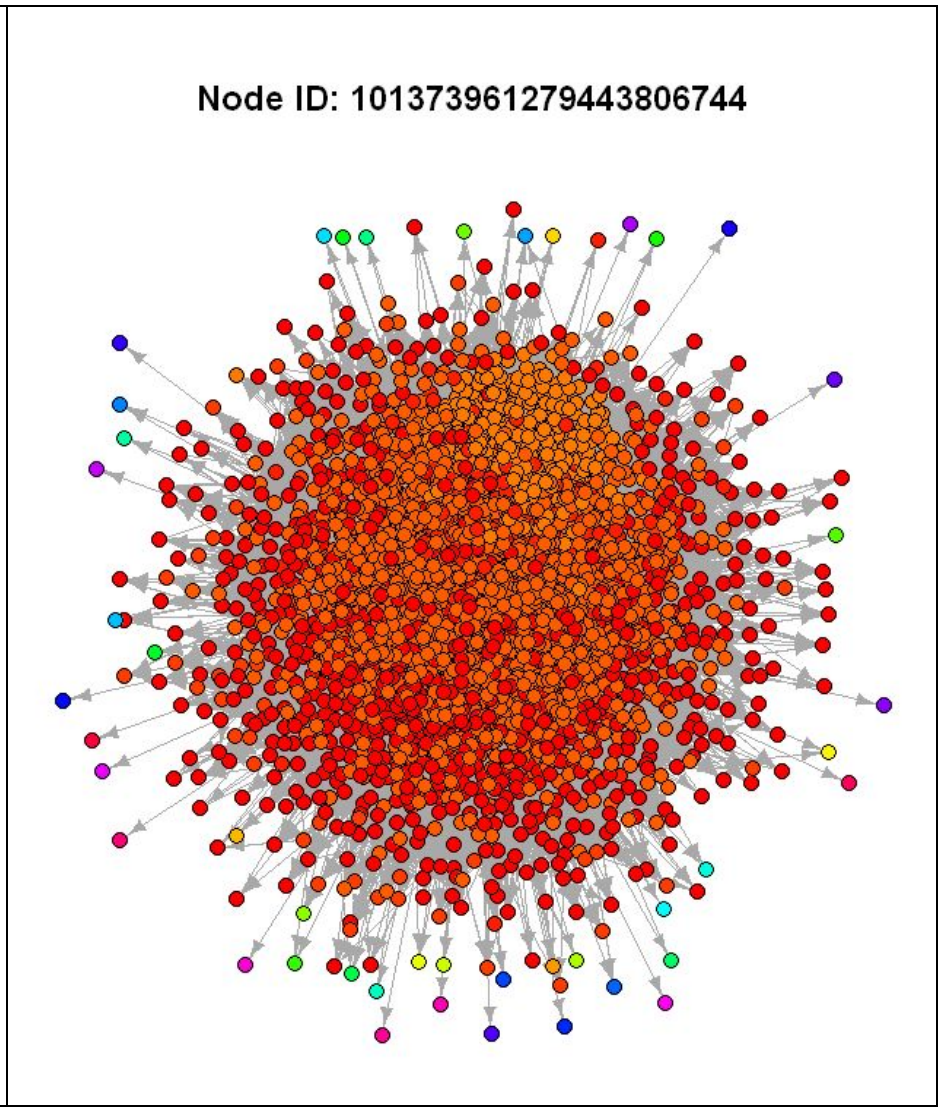
User ID	*Modularity*
109327480479767108490	0.252765387296677
115625564993990145546	0.312876313924789
101373961279443806744	0.163859650492668

User ID	*Number of Communities*	*Community Sizes*
109327480479767108490	4	288 397 76 13

115625564993990145546	34	37 46 2 263 232 338 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
101373961279443806744	49	1052 345 49 1811 531 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

User ID	*Visualization of Community Structure*
109327480479767108490	

11562556499399014554 6	

101373961279443806744	

Having found the communities, now we will explore the relationship between circles and communities. In order to explore the relationship, we define two measures:

- Homogeneity
- Completeness

Before, we state the expression for homogeneity and completeness, let's introduce some notation:

- C is the set of circles, C = {C1, C2, C3, · · · }
- K is the set of communities, K = {K1, K2, K3, · · · }
- a_i is the number of people in circle C_i
- b_i is the number of people in community K_i with circle information
- N is the total number of people with circle information

- C_{ji} is the number of people belonging to community 'j' and circle 'i'

Then, with the above notation, we have the following expressions for the entropy

$$H(C) = -\sum_{i=1}^{|C|} \frac{a_i}{N} \log(\frac{a_i}{N})$$

$$H(K) = -\sum_{i=1}^{|K|} \frac{b_i}{N} \log(\frac{b_i}{N})$$

And conditional entropy

$$H(C|K) = -\sum_{j=1}^{|K|} \sum_{i=1}^{|C|} \frac{C_{ji}}{N} \log(\frac{C_{ji}}{b_j})$$

$$H(K|C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{C_{ji}}{N} \log(\frac{C_{ji}}{a_i})$$

Now we can state the expression for homogeneity, 'h' as,

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

and the expression for completeness, c as

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

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

Ans:

There are two criteria of a successful clustering solution. First, the homogeneity criteria: each community should contain only data points that are members of a single circle. Second, the completeness criteria: all of the data points that are

members of a given circle should be elements of the same community.

The criteria of homogeneity and completeness are roughly in opposition; increasing the homogeneity of a community detection solution often results in a decrease of completeness. Consider degenerate clustering solutions. One, assigning every datapoint into a single community guarantees perfect completeness – all of the data points that are members of the same circle are trivially elements of the same community . However, this community is as far from homogeneous as possible – the maximum diversity of circle are represented in this single cluster. Two, assigning each data point to a distinct community guarantees perfect homogeneity – each community trivially only contains members of a single circle . However, by virtue of each community containing only a single member of a circle the community detection solution is as far from complete as possible.

The clustering task is to assign these data points to any number of clusters such that each cluster contains all and only those data points that are members of the same class.

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

Ans:

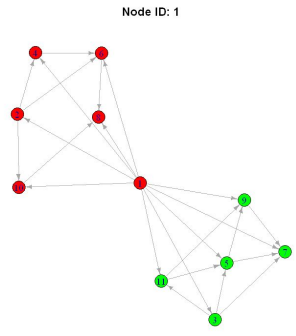
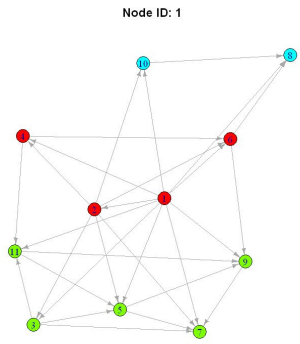
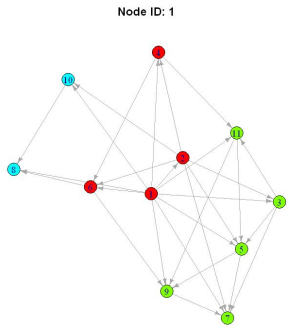
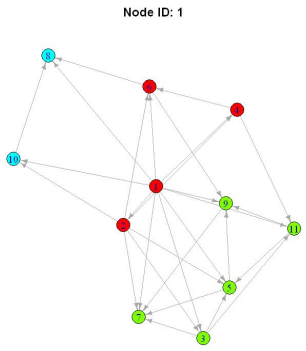
	USER ID		
	109327480479767108490	*115625564993990145546*	*101373961279443806744*
Number of Circles	3	31	3
Number of Communities	4	34	49
H(C)	1.05127807902057	8.46572937636873	0.388528191012741
H(K)	1.00520950378415	1.11953082674887	0.742602164404149
H(C|K)	0.157916044866551	4.61649018013109	0.385488255215886
H(K|C)	0.672735680256669	5.23814371424672	1.87720282723794
Homogeneity	0.849786609254066	0.454684885980695	0.00782423481017047
Completeness	0.330750776107735	-3.67887403284671	-1.52787147307088

As discussed in the previous question, homogeneity is a measure of the purity of a community, i.e., communities are considered pure if they predominantly contain members of the same classes. By observing the values obtained, we see that the personal network of user ID *109327480479767108490* has the highest homogeneity implying that the communities mostly contain members of a single circle. On the other hand, the personal network of user ID *101373961279443806744* has the lowest homogeneity indicating that the communities contain data points which belong to multiple circles.

To gain further insights, we compare the values of completeness. The values observed are on the lower end of the spectrum for all the three personal networks with the personal network of user ID *109327480479767108490* having the highest value among them. The completeness of the other two networks are surprisingly very low and in the negative range prompting us to investigate further and find possible reasons for the same.

Completeness is a measure of how members of a circle are assigned to communities. Completeness will be high if members of a circle are members of the same community. Low values will therefore be obtained if there is a possible overlap of circle members or connections between circle members. To further understand the effect of circle overlap and dense connections between circle members, we constructed a simple network of 11 nodes and two circles and experimented with circle overlap and density of connections to see the change in homogeneity and completeness.

The following are our observations:

	Case 1: Distinct Circles and No Connections Between Circles	*Case 2: Distinct Circles and Some Connections Between Circles*	*Case 3: Slightly Overlapping Circles and Considerable Connections Between Circles*	*Case 4: Completely overlapping Circles and Dense Connections between nodes*
Network				
Circles	{2, 4, 6, 8, 10}, {3, 5, 7, 9, 11}	{2, 4, 6, 8, 10}, {3, 5, 7, 9, 11}	{2 4 6 8 10 3 11}, {3 5 7 9 11 6 2}	{2 4 6 8 10 3 11 5 7 9}, {3 5 7 9 11 6 2 4 10 8}

Homogeneity	0.868482797083103	0.875488750216347	0.579359379456339	0.875488750216347
Completeness	1	0.674866511864863	0.164411197572229	-0.145688733218076

We observe that as we increase the overlapping nature of circles, the value of completeness drops and when the two circles completely overlap (case 4), the value of completeness becomes negative. Similarly, as we increase the density of connections between nodes (case 1 and case 2), we observe a decrease in completeness values.

From the insight obtained above, we inspect the overlapping nature of circles for the three personal networks in question. We compute the intersection of the circle members with all the other circles. The following are the results:

User ID: 109327480479767108490		
Circle Number	*Number of Members in Circle*	*Length of Intersection With Other Circles (Including Itself)*
1	330	330 330 1
2	346	330 346 1
3	419	1 1 419

User ID: 115625564993990145546		
Circle Number	*Number of Members in Circle*	*Length of Intersection With Other Circles (Including Itself)*
1	6	6 6 6 5 3 6 3 3 2 1 5 0 5 2 0 5 2 3 3 2 5 2 4 5 4 1 3 3 2 5 5
2	9	6 9 9 8 4 7 3 4 3 1 6 0 7 2 1 7 3 4 4 3 7 3 5 7 5 1 4 3 3 7 6
3	169	6 9 169 168 145 93 8 10 9 4 96 2 153 5 4 149 79 146 144 10 118 142 37 151 28 3 147 73 140 150 47
4	276	5 8 168 276 253 92 9 12 11 6 151 3 260 5 6 256 136 254 252 12 190 247 45 258 34 3 255 73 247 257 56
5	325	3 4 145 253 325 76 17 24 17 10 170 7 321 3 10 319 170 320 314 18 231 276 49 319 39 6

		320 62 300 320 59
6	93	6 7 93 92 76 93 5 6 5 1 60 0 81 5 2 79 37 77 76 6 64 74 30 81 26 1 77 73 71 80 38
7	73	3 3 8 9 17 5 73 73 12 62 34 7 72 2 7 19 9 17 13 12 51 9 8 19 7 6 17 2 13 19 12
8	338	3 4 10 12 24 6 73 338 33 62 126 255 133 2 258 29 14 25 19 33 232 12 10 29 9 10 26 2 19 29 17
9	46	2 3 9 11 17 5 12 33 46 10 25 10 46 1 13 19 7 17 16 46 36 11 7 20 7 10 17 2 14 19 11
10	62	1 1 4 6 10 1 62 62 10 62 27 5 61 1 5 10 4 10 8 10 41 5 4 10 3 6 10 1 7 10 7
11	338	5 6 96 151 170 60 34 126 25 27 338 89 241 7 93 193 98 172 165 26 151 148 79 202 66 5 187 42 156 192 102
12	255	0 0 2 3 7 0 7 255 10 5 89 255 51 1 255 7 5 7 5 10 175 3 3 7 3 3 7 0 6 7 6
13	485	5 7 153 260 321 81 72 133 46 61 241 51 485 6 56 363 187 326 313 48 353 275 63 373 52 10 353 63 296 361 82
14	7	2 2 5 5 3 5 2 2 1 1 7 1 6 7 1 4 2 3 3 1 6 3 7 6 6 1 3 3 2 4 7
15	260	0 1 4 6 10 2 7 258 13 5 93 255 56 1 260 12 6 10 8 13 179 5 5 12 5 3 11 1 8 12 9
16	363	5 7 149 256 319 79 19 29 19 10 193 7 363 4 12 363 187 326 313 20 262 275 56 363 45 6 353 62 294 361 69
17	188	2 3 79 136 170 37 9 14 7 4 98 5 187 2 6 187 188 173 167 7 135 146 30 187 23 4 188 30 153 188 38
18	327	3 4 146 254 320 77 17 25 17 10 172 7 326 3 10 326 173 327 314 18 233 276 50 326 40 6 327 63 295 327 60
19	314	3 4 144 252 314 76 13 19 16 8 165 5 313 3 8 313 167 314 314 17 226 276 46 313 36 6 314 62 295 314 56

20	48	2 3 10 12 18 6 12 33 46 10 26 10 48 1 13 20 7 18 17 48 37 12 8 21 7 10 18 2 15 20 12
21	489	5 7 118 190 231 64 51 232 36 41 151 175 353 6 179 262 135 233 226 37 489 200 53 271 46 9 253 51 213 261 73
22	276	2 3 142 247 276 74 9 12 11 5 148 3 275 3 5 275 146 276 276 12 200 276 42 275 32 3 276 60 273 276 50
23	79	4 5 37 45 49 30 8 10 7 4 79 3 63 7 5 56 30 50 46 8 53 42 79 62 66 1 51 21 43 55 79
24	373	5 7 151 258 319 81 19 29 20 10 202 7 373 6 12 363 187 326 313 21 271 275 62 373 51 6 353 63 294 361 78
25	67	4 5 28 34 39 26 7 9 7 3 66 3 52 6 5 45 23 40 36 7 46 32 66 51 67 1 40 19 33 44 66
26	10	1 1 3 3 6 1 6 10 10 6 5 3 10 1 3 6 4 6 6 10 9 3 1 6 1 10 6 1 4 6 2
27	354	3 4 147 255 320 77 17 26 17 10 187 7 353 3 11 353 188 327 314 18 253 276 51 353 40 6 354 63 295 354 63
28	73	3 3 73 73 62 73 2 2 2 1 42 0 63 3 1 62 30 63 62 2 51 60 21 63 19 1 63 73 57 63 26
29	300	2 3 140 247 300 71 13 19 14 7 156 6 296 2 8 294 153 295 295 15 213 273 43 294 33 4 295 57 300 295 50
30	362	5 7 150 257 320 80 19 29 19 10 192 7 361 4 12 361 188 327 314 20 261 276 55 361 44 6 354 63 295 362 68
31	102	5 6 47 56 59 38 12 17 11 7 102 6 82 7 9 69 38 60 56 12 73 50 79 78 66 2 63 26 50 68 102

User ID: 101373961279443806744		
Circle Number	*Number of Members in Circle*	*Length of Intersection With Other Circles (Including Itself)*
1	471	471 406 417
2	445	406 445 362
3	430	417 362 430

As we can see, there is significant overlap between the circles of the second and third personal networks which severely impacts the completeness as can be evident from the results obtained.