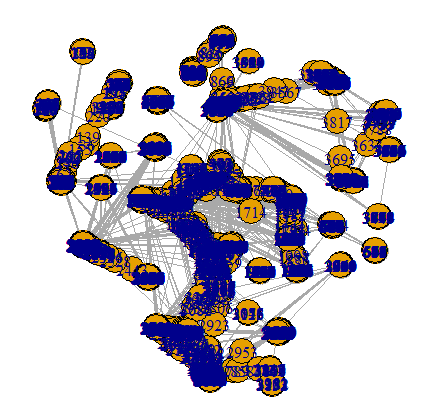
**ECE 232E Project 2**

Group members: Kushagra Rastogi (304640248)

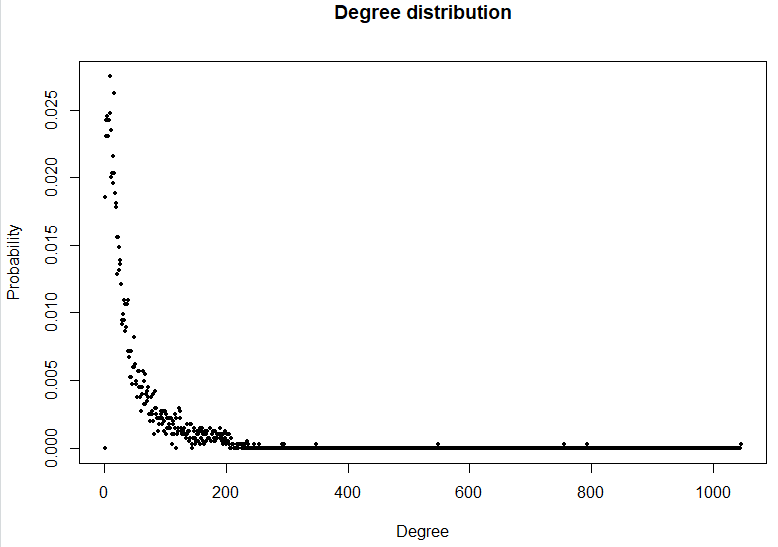
Jonathan Lee (104840173)

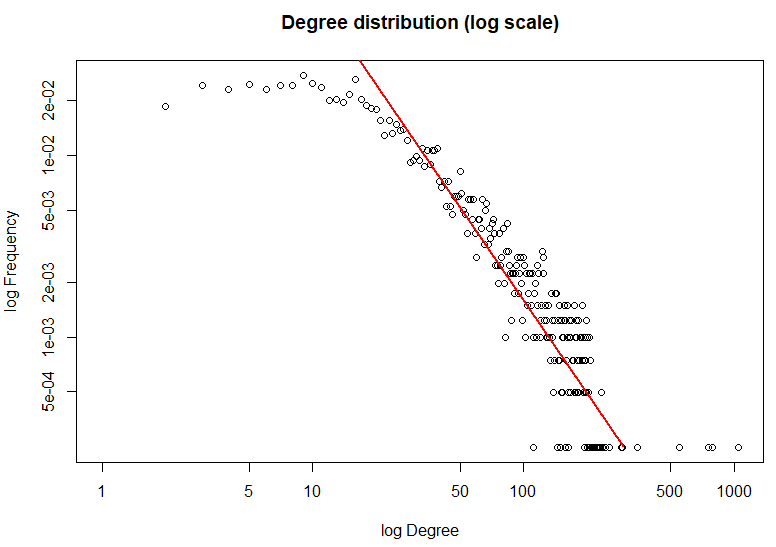
**Part I: Facebook Network**

1. The graph is connected. Nodes = 4039 and Edges = 88234.



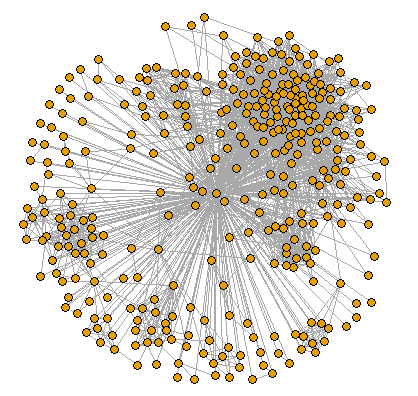
1. Diameter = 8
2. Average degree = 43.69101





The slope of the line is -1.75.

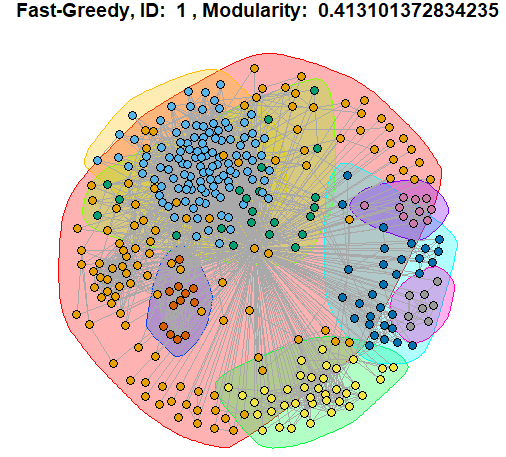
1. The personalized network looks like the following. It has nodes = 348 and edges = 2866.

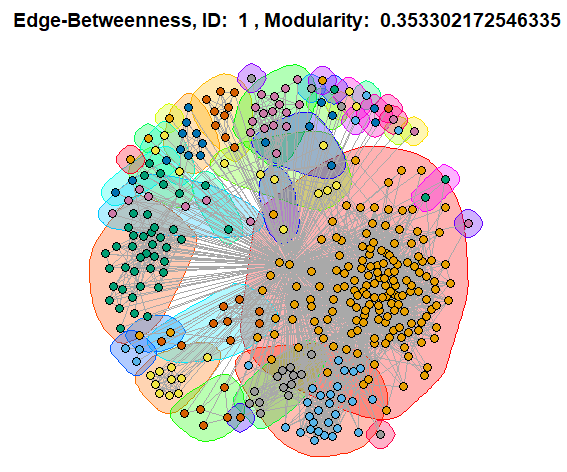
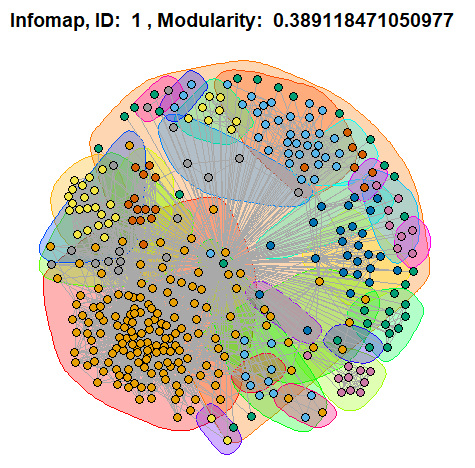


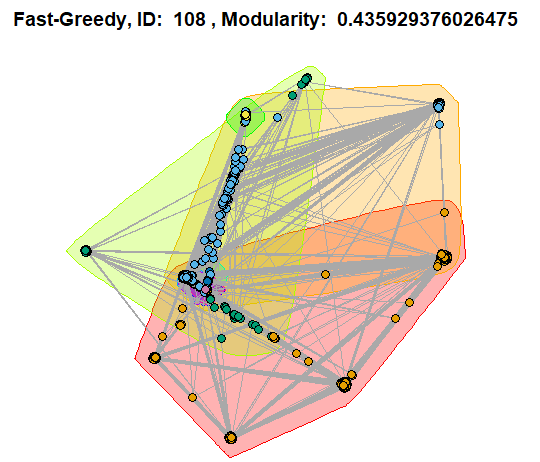
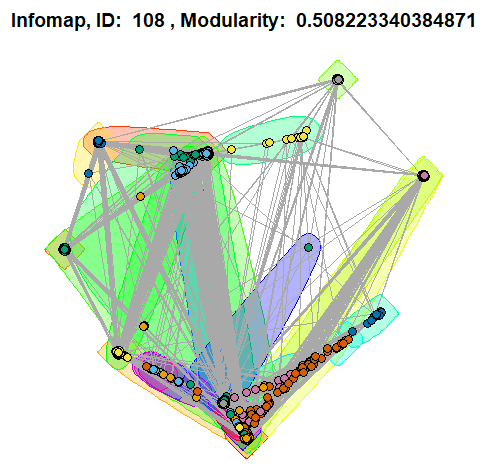
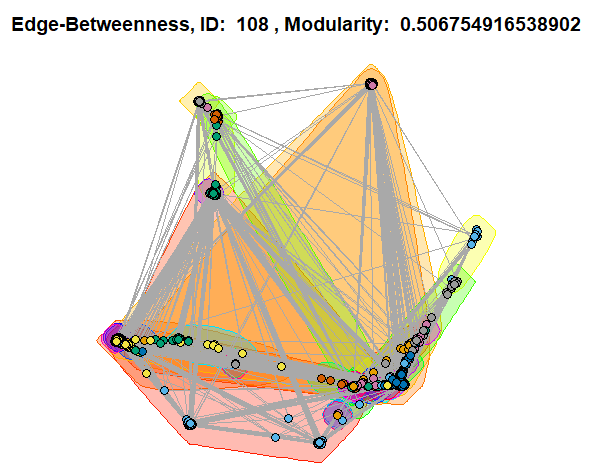
1. The diameter = 2. A trivial lower diameter lower bound = 1 and trivial diameter upper bound = 2.
2. The diameter is the longest shortest path in the graph. If the diameter is equal to the trivial upper bound, then that means 3 nodes , and are connected like or some feasible permutation of the nodes in the path. This means that there exist a pair of nodes that do not have an edge between them. However, these pair of nodes have a mutual friend (in Facebook’s terminology) between them, namely the node .

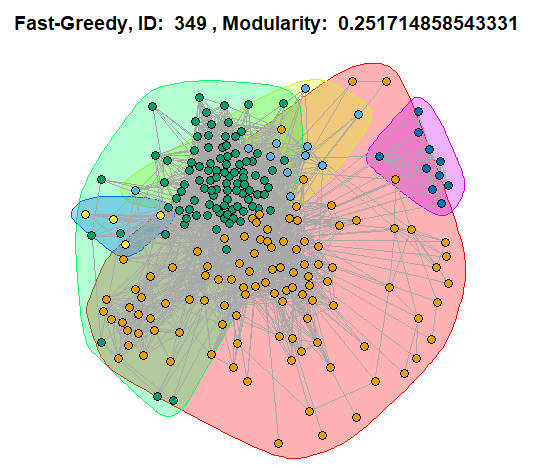
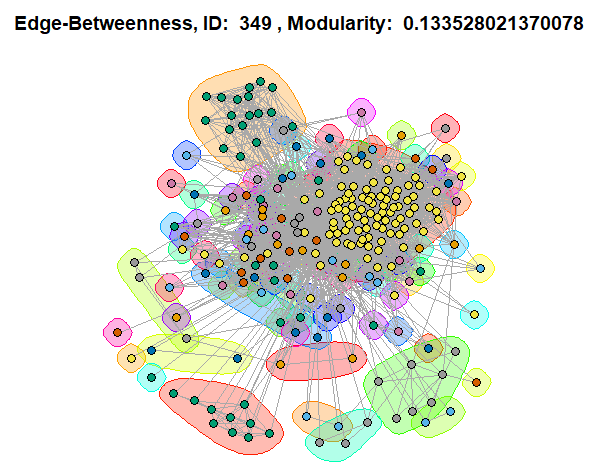
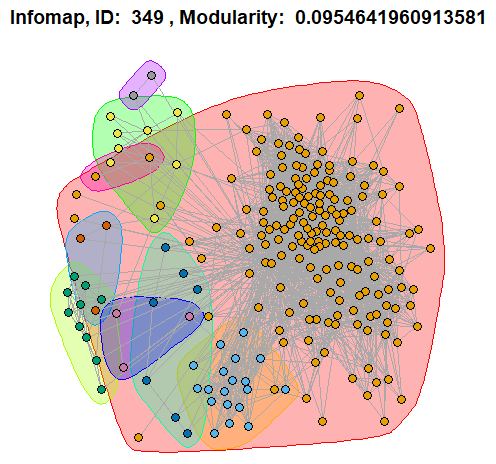
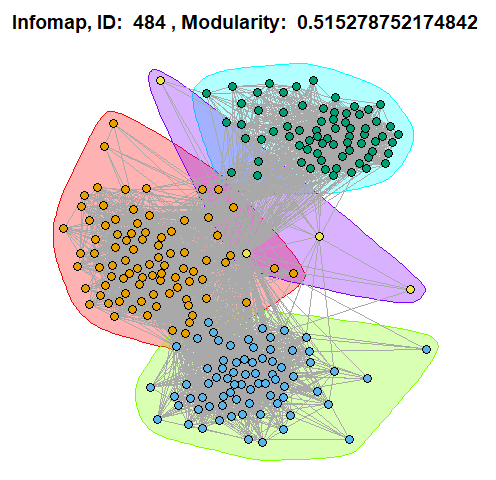
If the diameter is equal to the trivial lower bound, then this means a set of 3 nodes form a clique. Hence each node is connected to every other node. For instance, if there are 3 nodes , and , then we have , and . Thus, every node is a friend with every other node in the network.

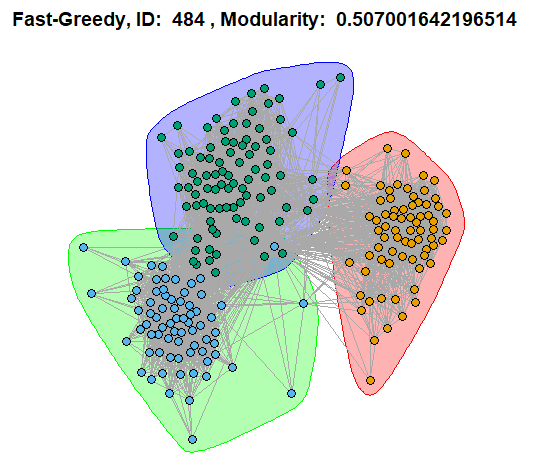
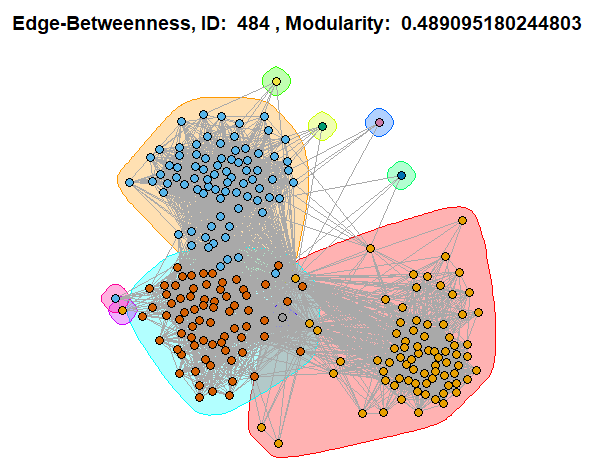
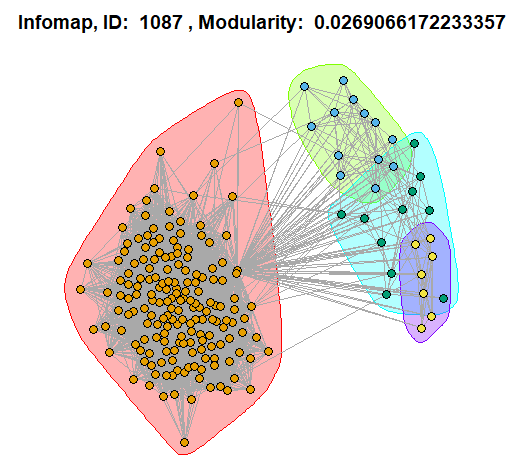
1. Number of core nodes = 40 and the average degree of core nodes = 279.375
2. The plots are shown below.

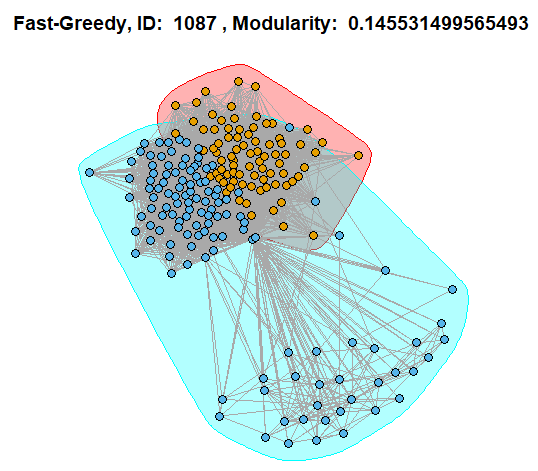
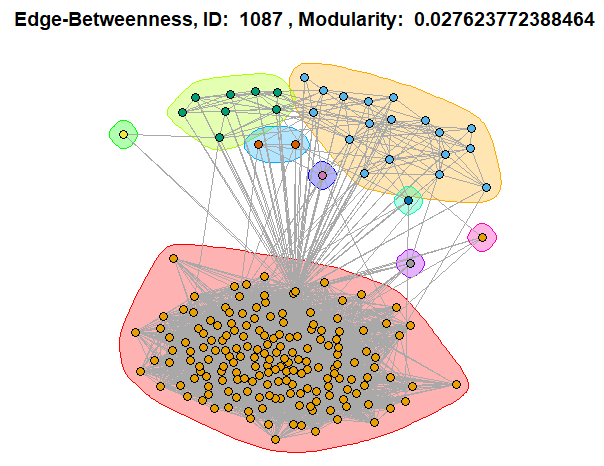








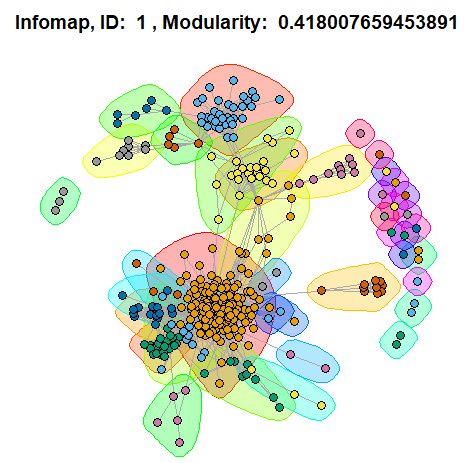


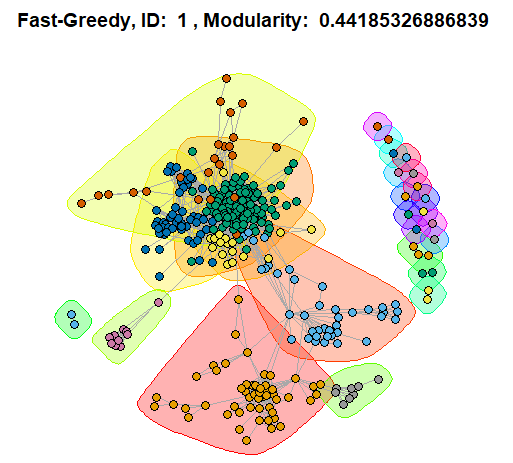
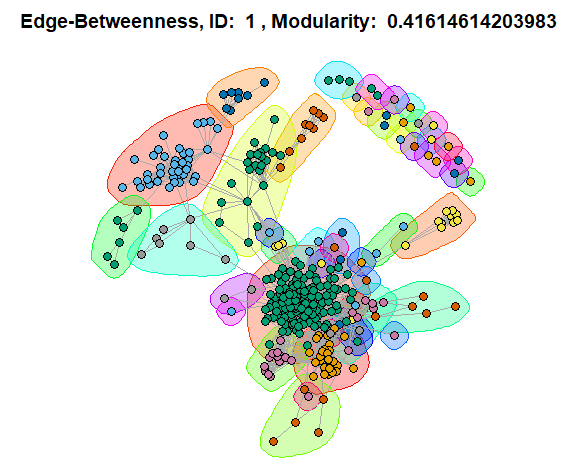


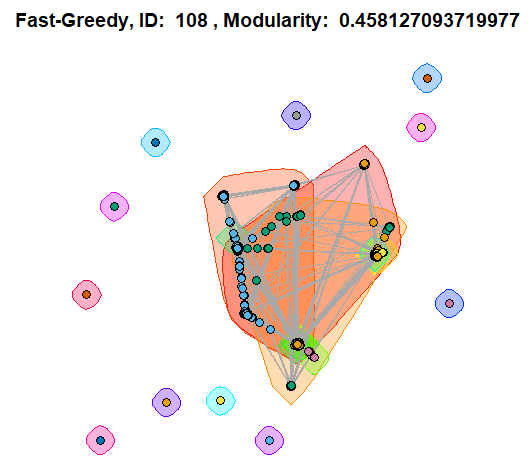
Observations:

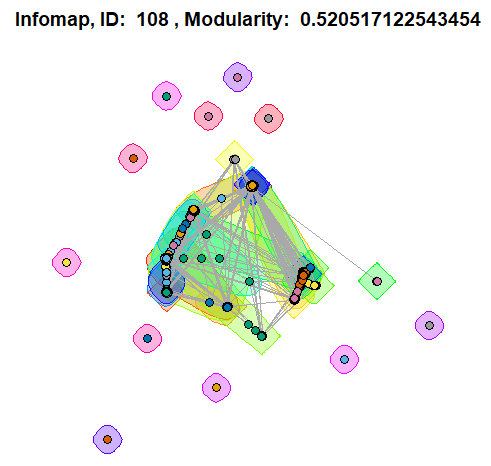
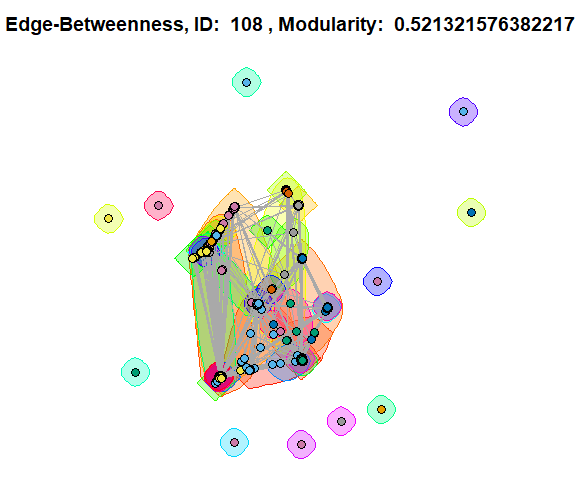
A) Node ID 484 has the highest average modularity and node ID 1087 has the lowest average modularity score. This means that node ID 484 has distinct community structure, but node ID 1087 has a very loose community structure. This can also be verified visually using the plots.

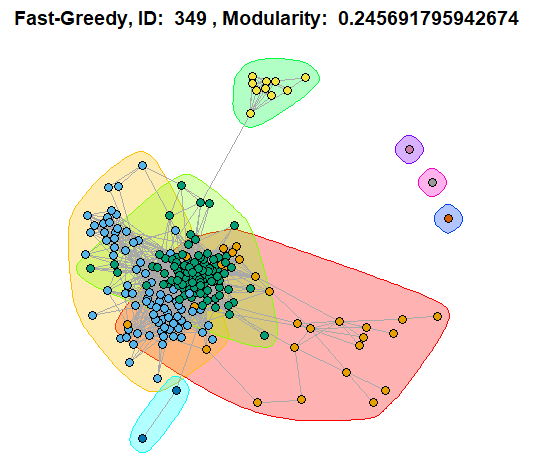
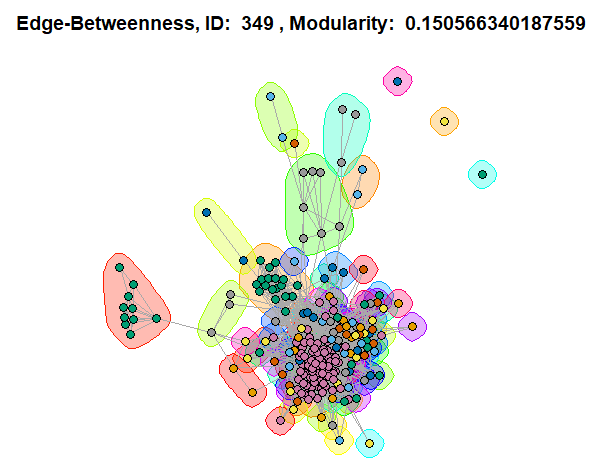
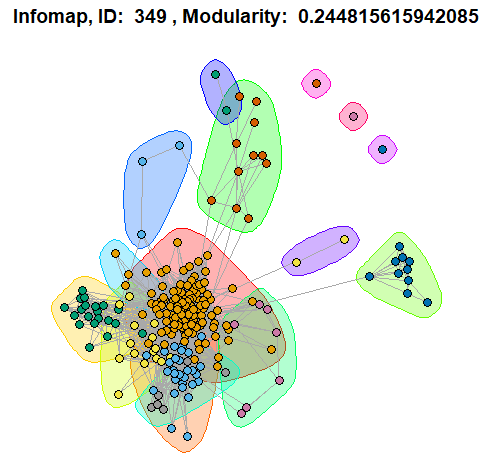
B) Fast-greedy has the best performance since it usually gives the highest modularity scores. Furthermore, fast-greedy is the fastest since it acts greedily and doesn’t perform intensive computations. It was interesting to see that edge-betweenness and infomap work better on larger networks like node ID 108 and fast-greedy works better on smaller networks like node ID 1.

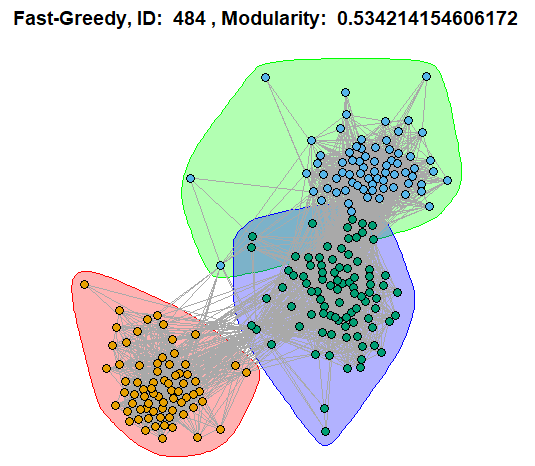
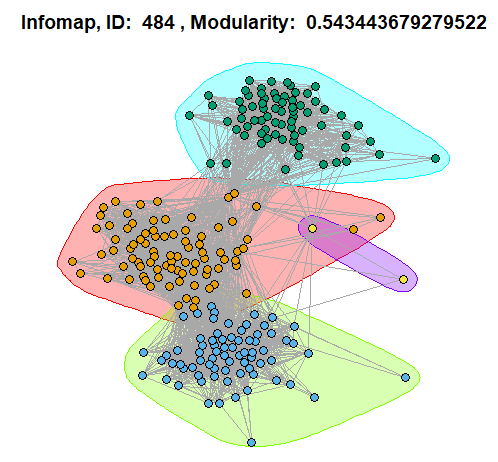
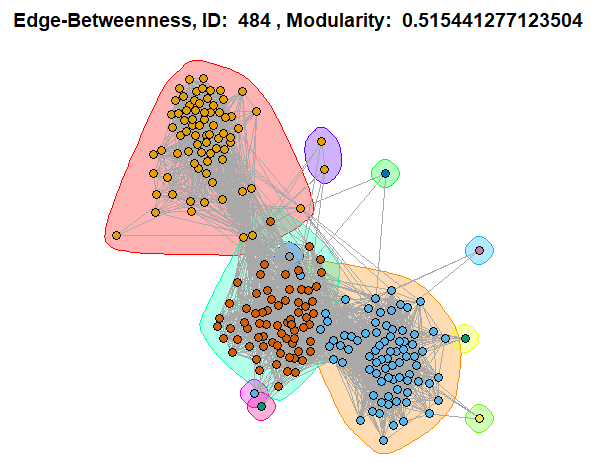
1. The plots are shown below.

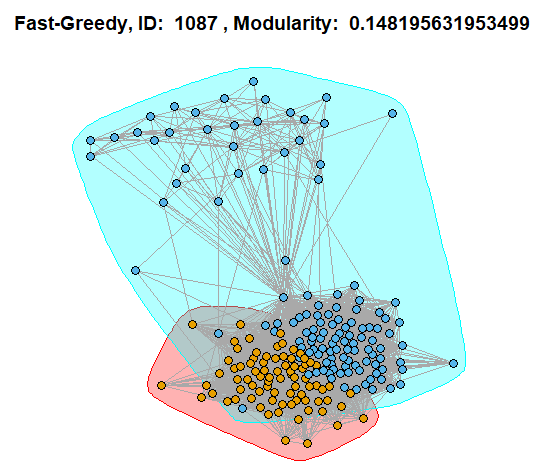
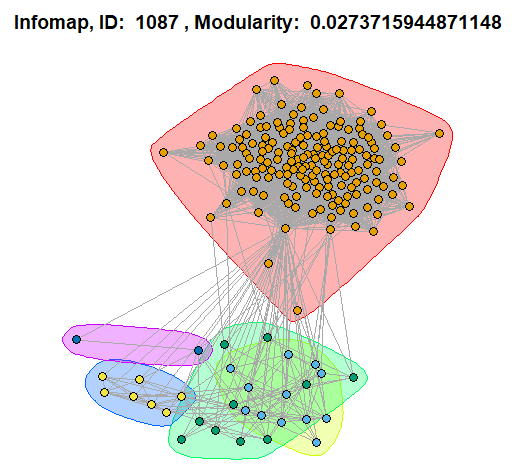


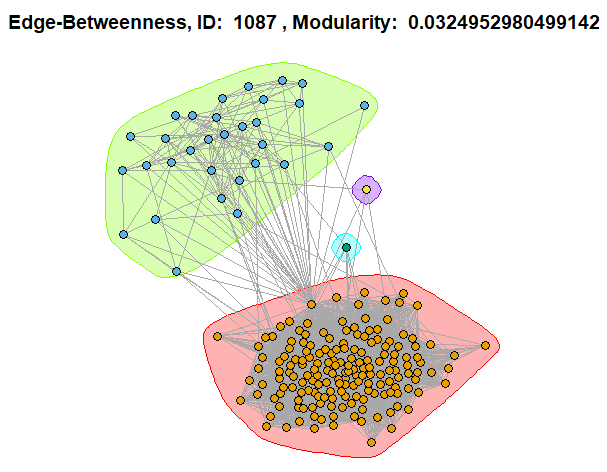






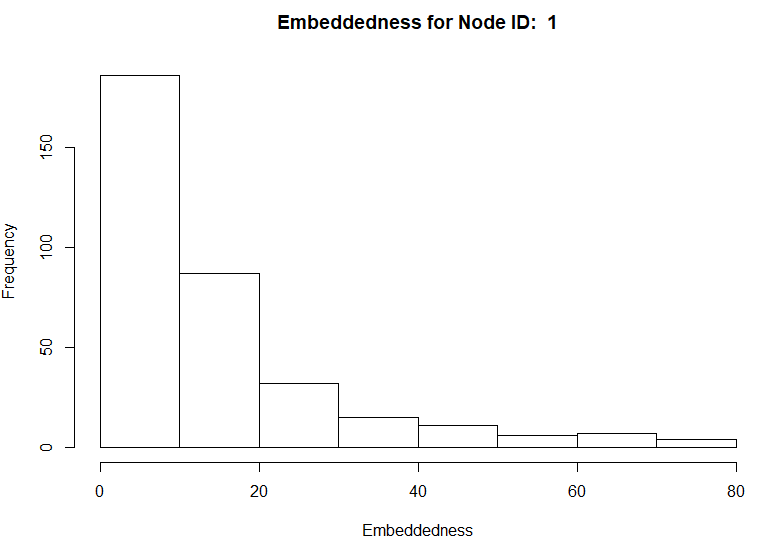
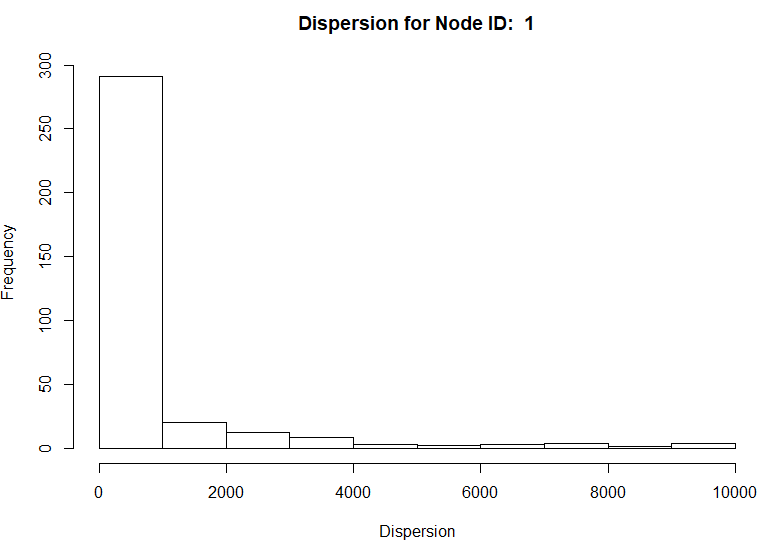


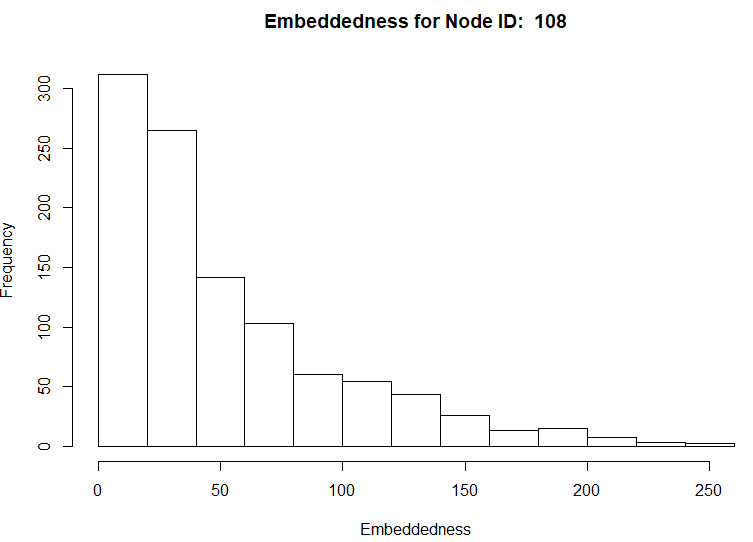
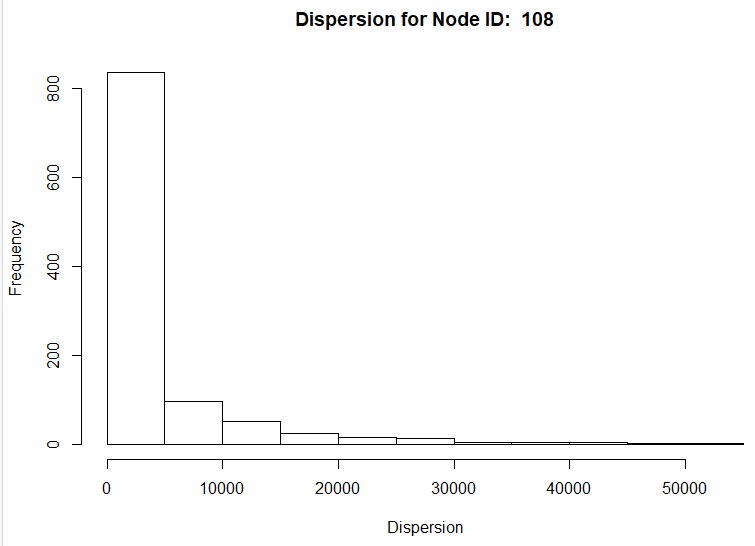


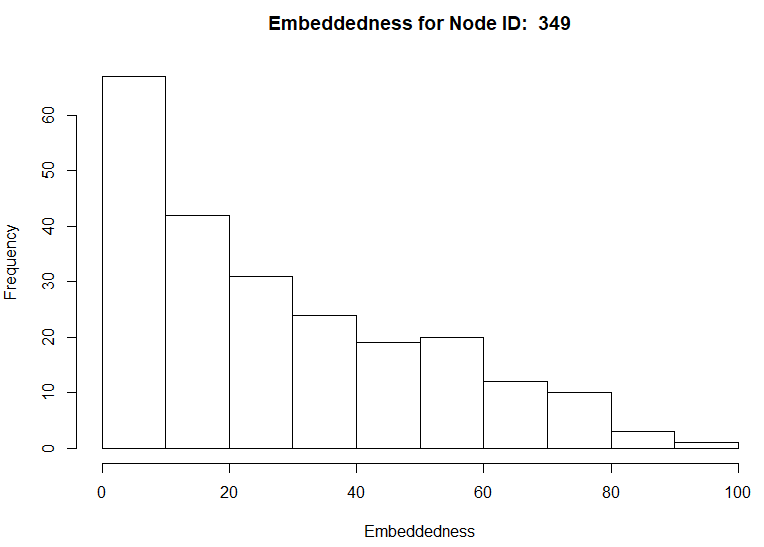


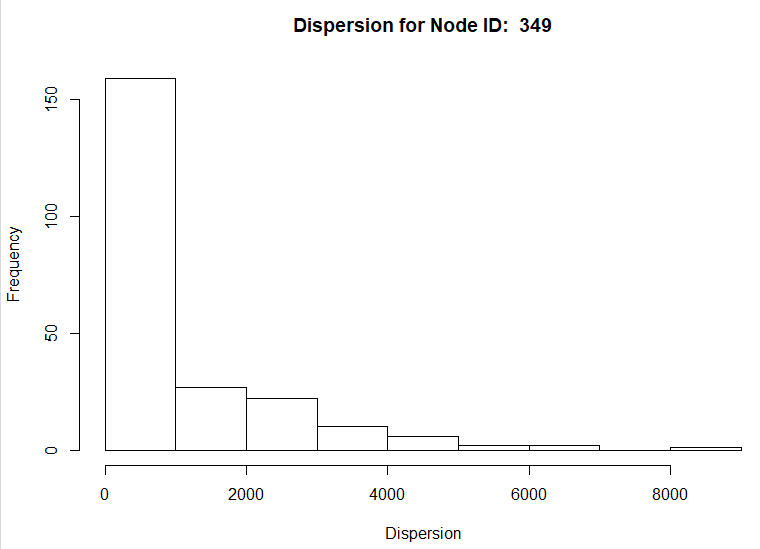
Observations:

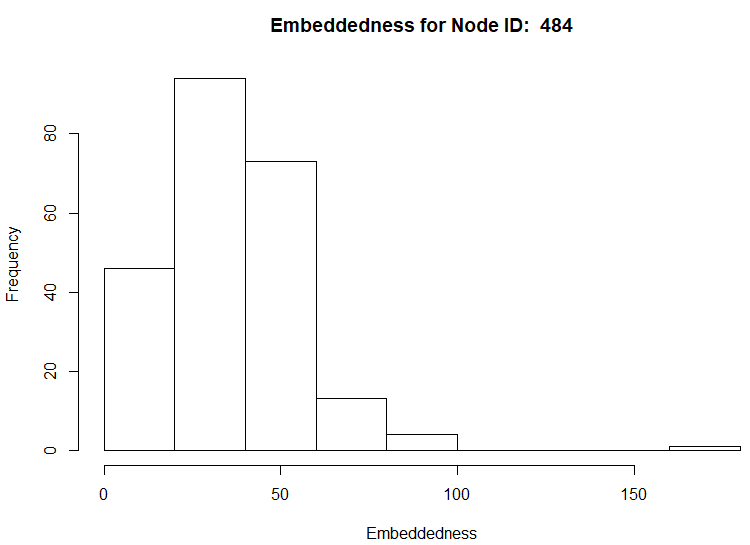
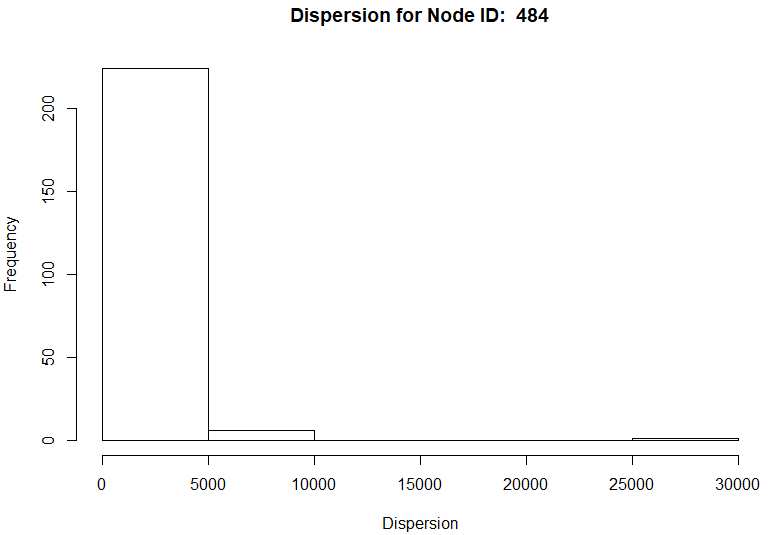
1. With the core node removes, fast-greedy still has the best performance among the 3 community detection algorithms since it gives the highest modularity score on average.
2. On average, the modularity score with core nodes removed is greater than the modularity score with the core nodes. With the presence of a core node, the communities are quite densely connected. With the core node removed, many nodes start to become their own communities, thereby augmenting the community structure and increasing the modularity scores. Hence, the removal of core nodes leads to a growth in the number of communities which increases the modularity scores calculated by all the given community detections algorithms.
3. The plots are shown below.

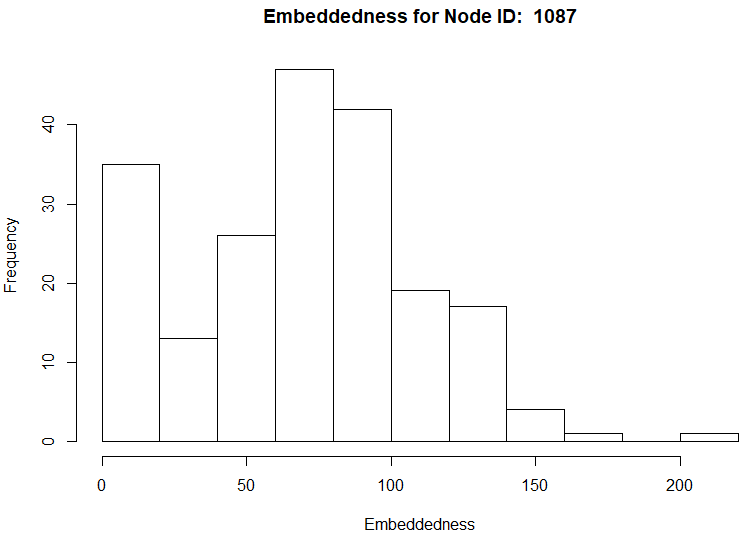
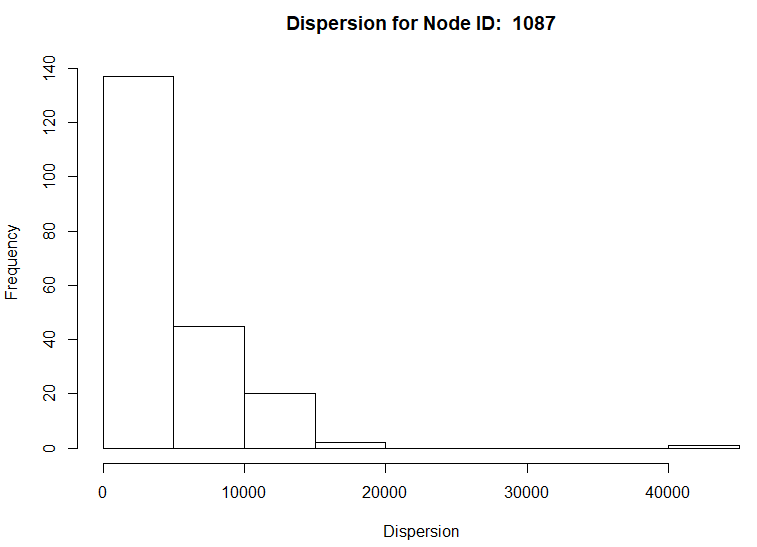




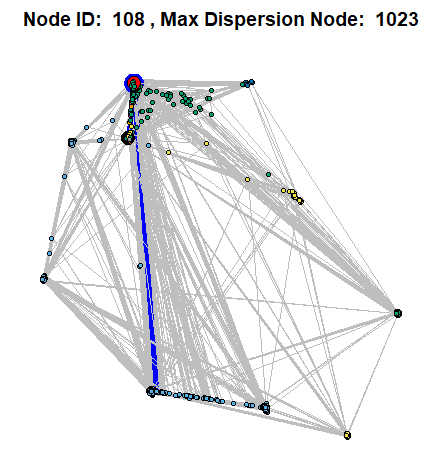
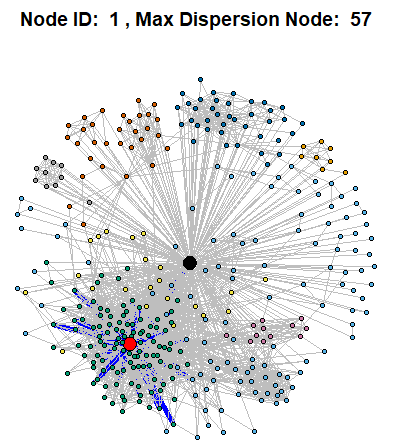


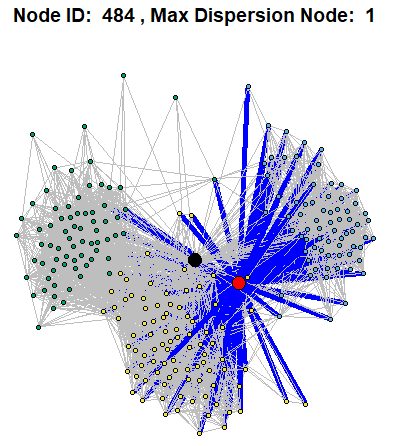
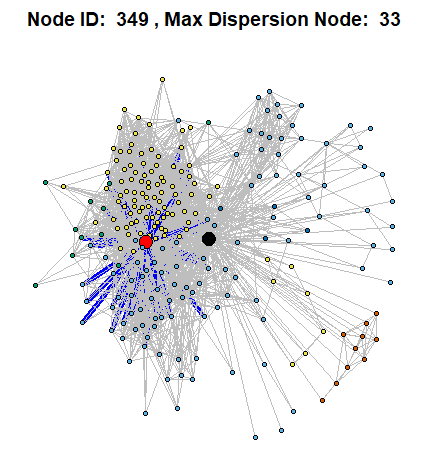


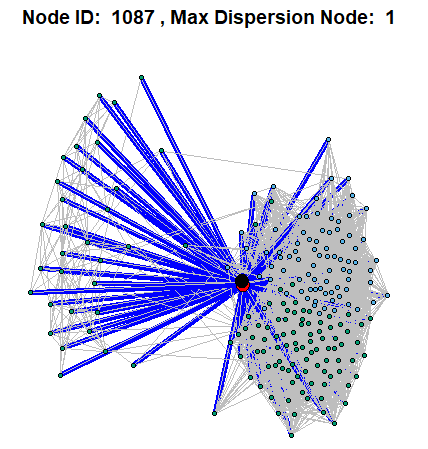




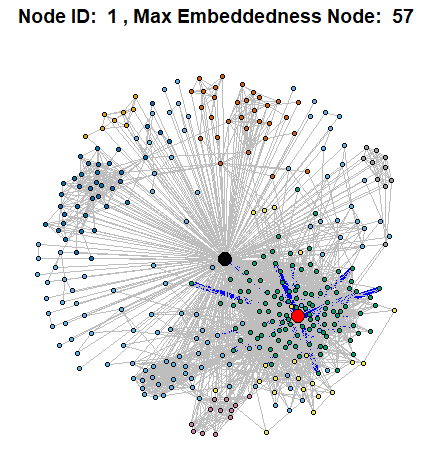
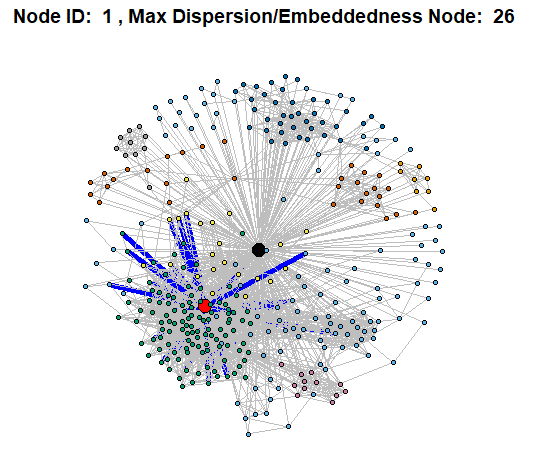
1. The plots are shown below. The node with maximum dispersion is highlighted in red with its incident edges highlighted in blue. The core node is highlighted in black. The node ID of the node with the maximum dispersion is given in the title of the plot.

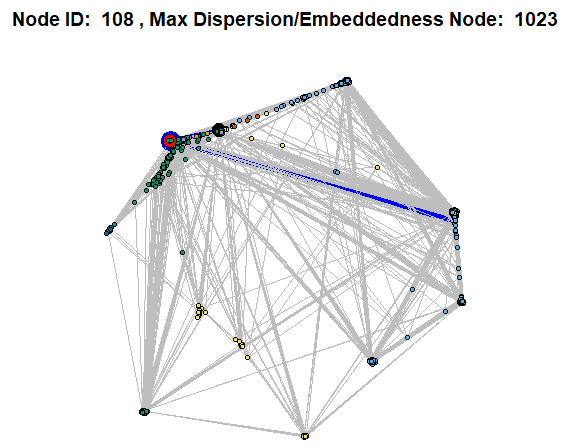
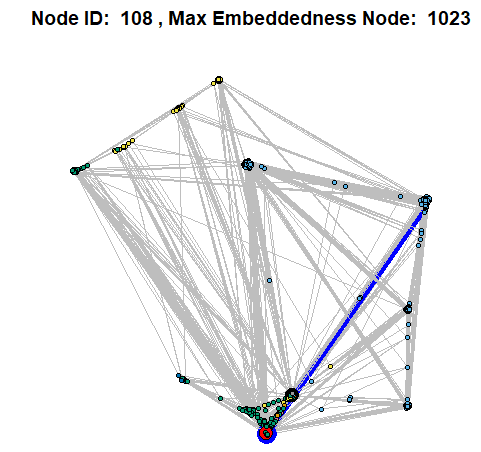


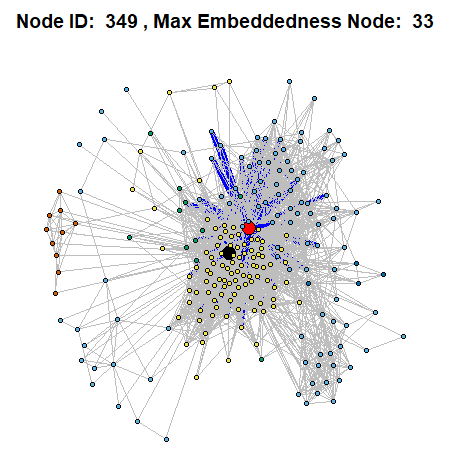
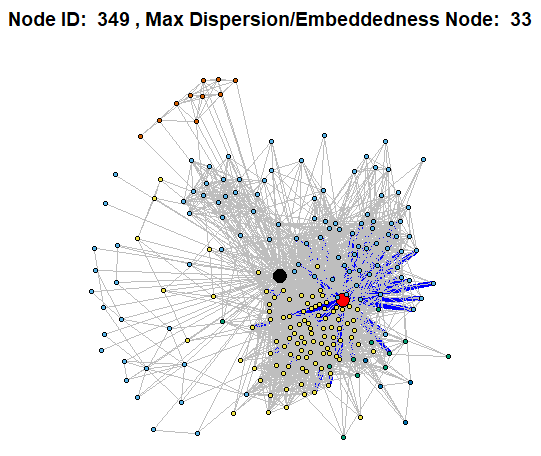


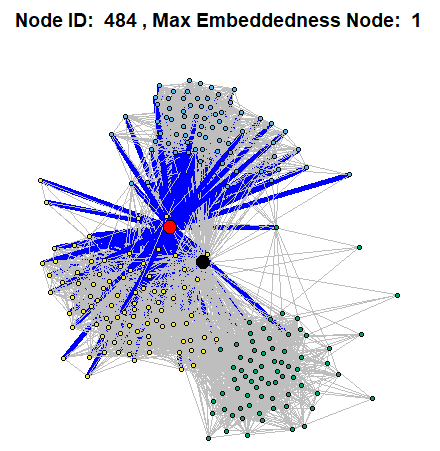
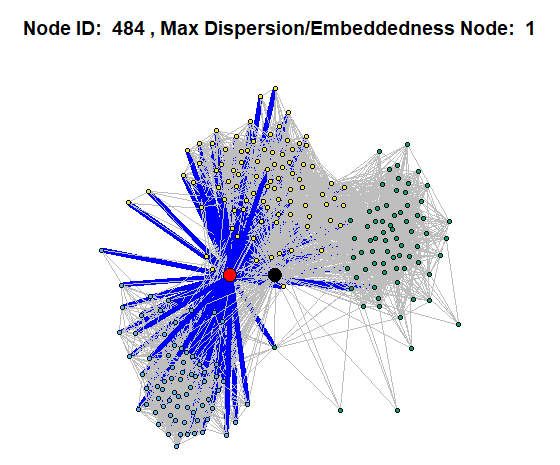


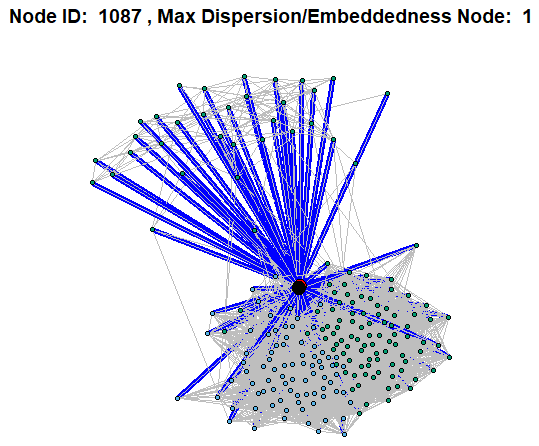
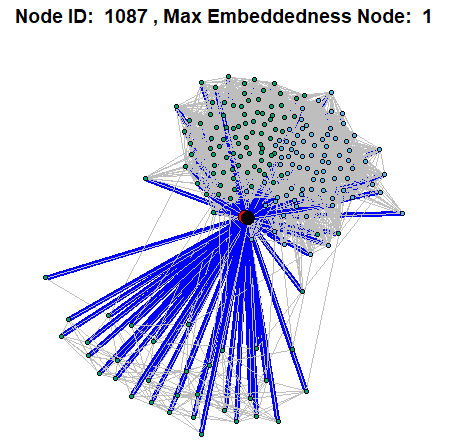
1. The plots are shown below. The node with maximum embeddedness and maximum dispersion/embeddedness ratio is highlighted in red with its incident edges highlighted in blue. The core node is highlighted in black. The node IDs are in the title of the plot.











1. Dispersion is a very good indicator of strong ties. A high dispersion value indicates that two nodes share many mutual friends, but the mutual friends are sparsely connected. A classic example of this phenomenon would be two childhood friends who study at different colleges. They might have a lot of mutual friends, but the mutual friends may not know each other. From the plots, it can be seen that incident edges for maximum dispersion nodes are not as dense as incident edges for maximum embeddedness. Another interesting fact is that the dispersion of a node decreases as the size of a code node’s personalized network increases. This is evident in the case for node ID 108 since it has the most neighbors.

Embeddedness has historically been used as an indicator for strong ties, but it is not as good as dispersion. A high embeddedness means that two nodes share a lot of mutual friends, but this does not necessarily mean that the two nodes are close or somehow intimately connected. From the plots, larger communities have larger value of embeddedness since the degree of a non-core node is likely to be larger in big communities.

The ratio dispersion/embeddedness is a normalized feature that combines the concepts of both dispersion and embeddedness. The ratio emphasizes the connectivity of the nodes’ neighbors. Hence, a high ratio would indicate that two nodes share a mutual community instead of just mutual friends. According to the paper in the project statement, a high ratio would also suggest that two nodes are likely in a romantic relationship. From the plots, it can be seen that as the network size decreases, the ratio increases indicating closer and stronger ties.

1. For this example, we explore the personalized network of ID# 415 within the Facebook network. We take a look at all of the neighboring nodes of this personalized network and generate a list of people/users to recommend friends to. Afterwards, we will analyze the expected accuracy of these recommendation algorithms with the people/users that have a degree of exactly 24 in the list. For the Facebook dataset, we have |Nr| = 11
2. Taking a look at the friend recommendation algorithms, we utilize the list of people/users with a degree of exactly 24. For each people, we randomly delete a neighbor of the node with 25% chance. With this new list of deleted neighbors, we append a new list of people to the current node of interest as recommendations. Using three different recommendation algorithms to determine this list of people, namely, common neighbors, Jaccard, Adamic Adar, we calculate the mean accuracy of the results.

The mean accuracies of the 3 algorithms are shown below:

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Common Neighbors | 0.8508369 |
| Jaccard | 0.8625738 |
| Adamic Adar | 0.8528024 |

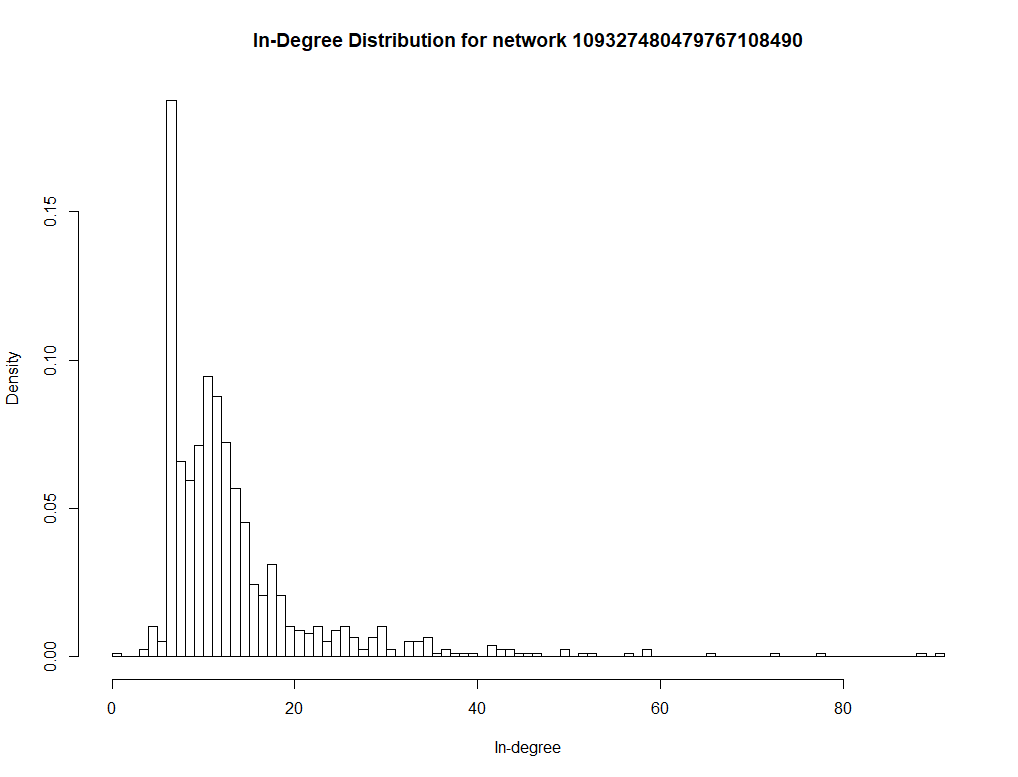
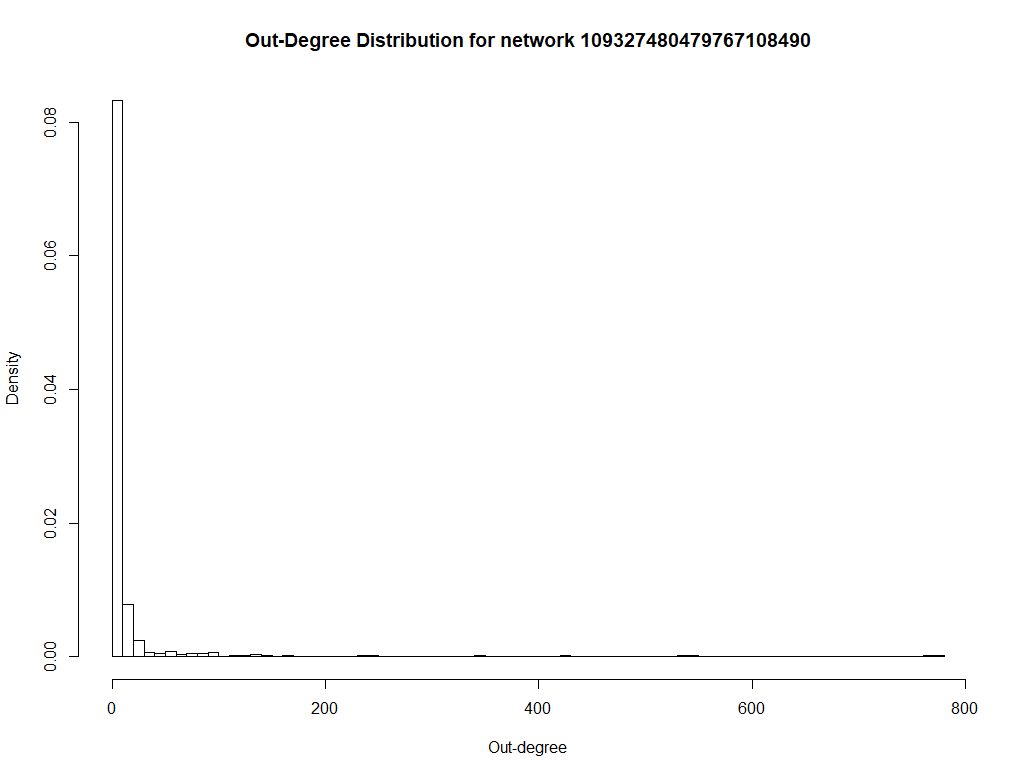
Upon observing the results of the three algorithms, there doesn’t seem to be a clear-cut distinction for the best recommendation algorithm, but all three do seem to perform above satisfaction for its accuracy values. During runtime, however, the Adamic Adar algorithm does seem to take noticeably longer than the former two methods.

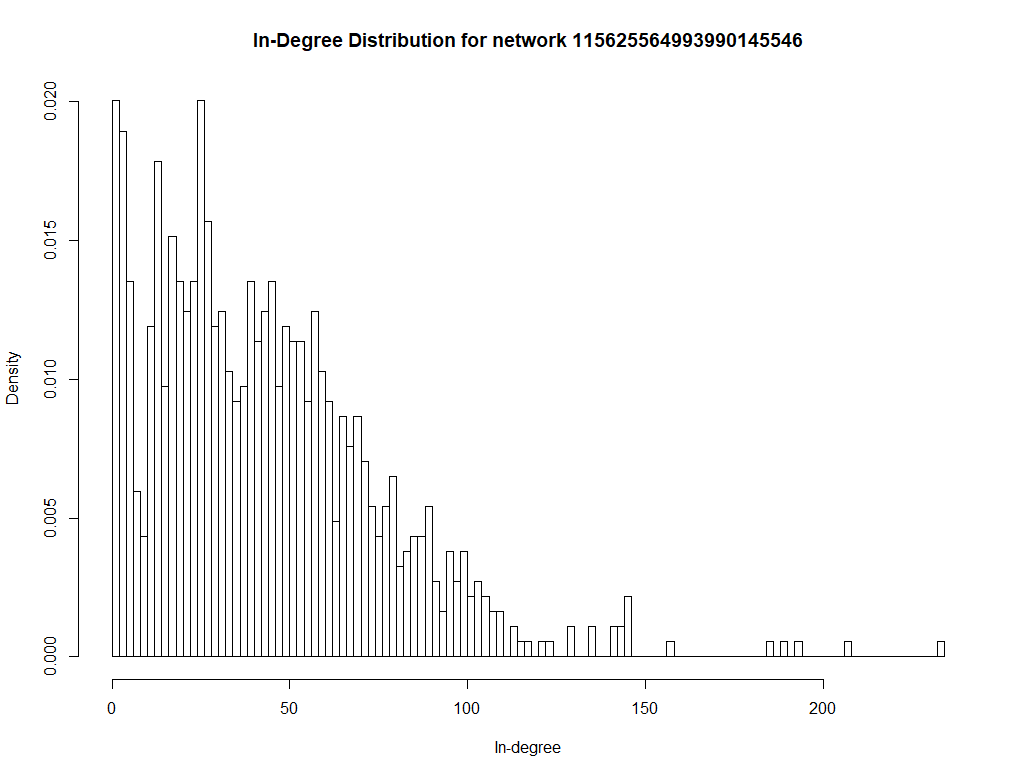
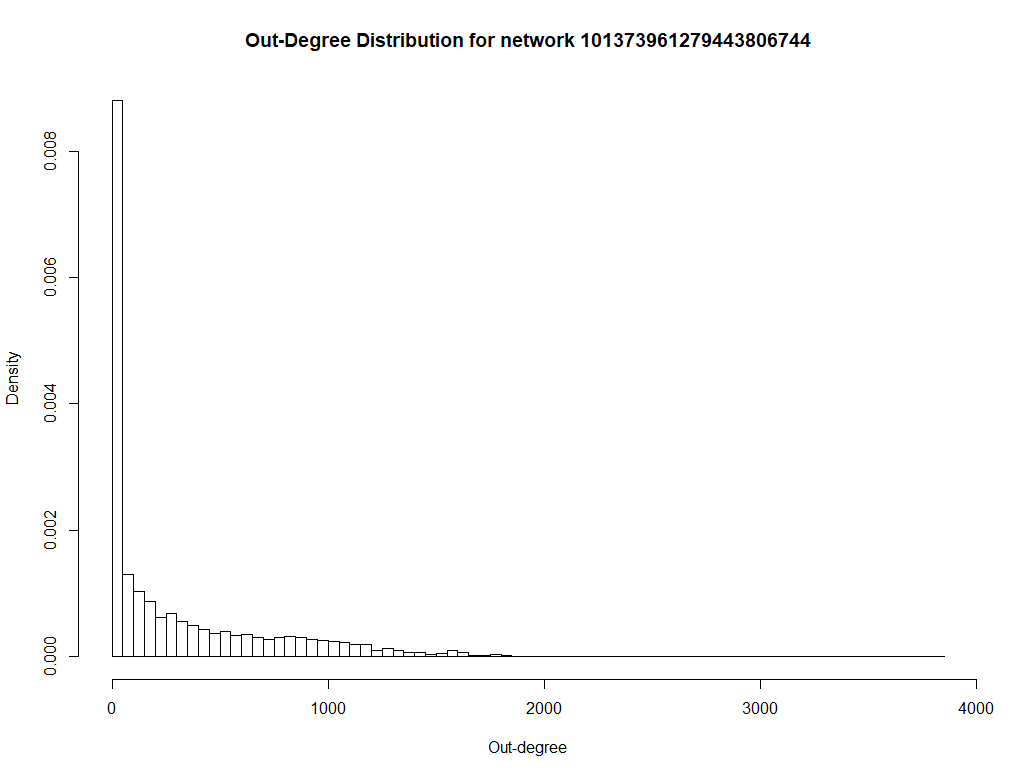
**Part II: Google+ Network**

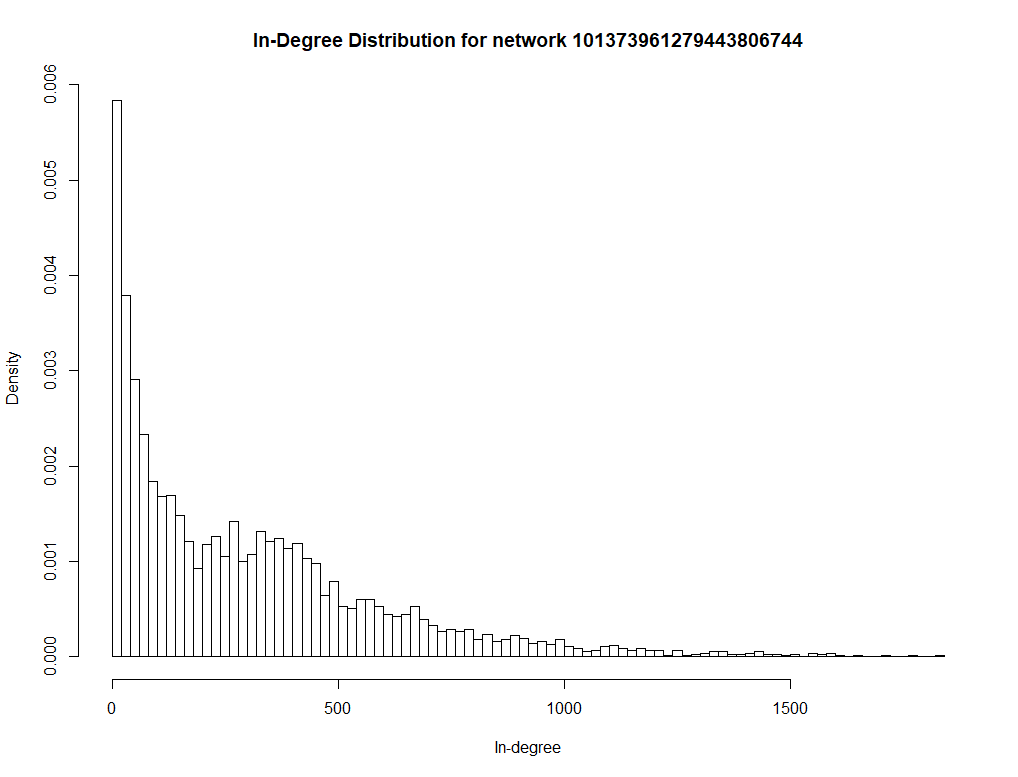
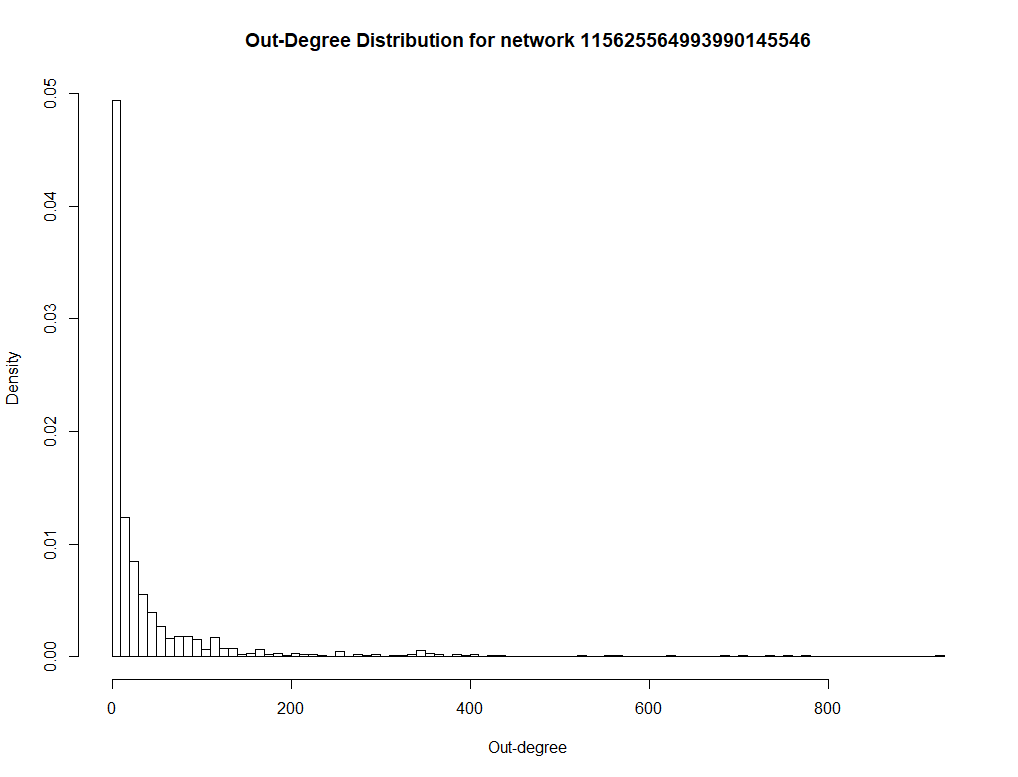
**18.** In this example, we explore the structure of the Google+ network by observing a few of its separate/personal networks. Upon navigating its massive dataset, we have concluded that there is a total of 57 personal networks that contain more than 2 circles.

**19.** We specifically analyze 3 personal networks from Google+. The network IDs are listed below:

|  |  |  |
| --- | --- | --- |
| Network ID | Number of Circles | Number of Communities |
| 109327480479767108490 | 3 | 4 |
| 115625564993990145546 | 31 | 10 |
| 101373961279443806744 | 3 | 31 |

Note that the number of communities were calculated using the Walktrap algorithm, which we will look further into in the subsequent analysis. Results for the in-degree and out-degree distributions for the personal networks are shown below:





**20.** As mentioned previously, we utilized the Walktrap algorithm to extract the community structure of the 3 personal networks. In doing so, each node/person in each personal network will be associated with at most one densely connected subgraph, or community. The modularity scores of each community structure are shown below:

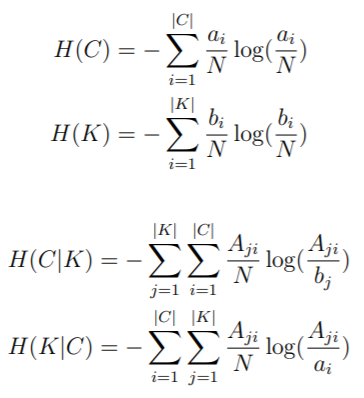
|  |  |
| --- | --- |
| Network ID | Modularity |
| 109327480479767108490 | 0.2527654 |
| 115625564993990145546 | 0.3194726 |
| 101373961279443806744 | 0.1910903 |

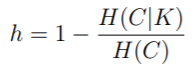
The modularity scores for the community structures of each personal network are dissimilar. It seems that the second personal network has the best modularity, while the third personal network has the worst of the 3. This implies that the number of connections between nodes in different communities are comparable to the number of connections between nodes within communities for the third personal network. If a modularity score is high, however, the inter-connections and intra-connections among communities are high and low, respectively.

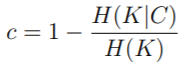
**21.** At this point, community structures are defined for the 3 personal/user networks of interest. Given this extraction, we now focus our attention on the relationship between communities and user circles/classes. There are 2 measures that we explore to perform this task—homogeneity and completeness. The following expressions will guide us in computing these measures:

1. C­ is the set of circles, C = {C1, C2, C3, …}
2. K is the set of communities, K = {K1, K2, K3, …}
3. ai is the number of people in circle Ci
4. bi is the number of people in community Ki and bi Є Ck, where Ck Є C
5. N is the total number of people with circle information
6. Aji is the number of people belonging to community j and circle i

We will make use of these expressions to calculate the entropies and conditional entropies:



Lastly, these equations are used to calculate the homogeneity and completeness of each of the 3 personal/user networks:



Based on these expressions, homogeneity implies how well clusters/communities contain nodes/people from only a single circle/class. On the other hand, completeness measures how well represented nodes/people from a single circle/class are within a single cluster/community. Examples of both perfect homogeneity and perfect completeness are shown below:

|  |  |
| --- | --- |
| Perfectly Homogeneous Clustering | Perfectly Complete Clustering |
| Class/Circle 1  Class/Circle 3  Class/Circle 2    Community D  Community A  Community C  Community B | Community B  Community A  Class/Circle 1  Class/Circle 3  Class/Circle 2 |

**22.** From the above expressions, we can now solve for the homogeneity and completeness of each personal network.

Looking at the results of the first network, it can be seen that it has the highest homogeneity and completeness scores out of the three. This implies that the communities do well in encapsulating people/nodes of only one class/circle. In other words, each community in the network does a good job in representing only a single class/circle (very few different types of classes within a community). The network’s completeness, however, implies that it does not perform well in keeping a particular class/circle inside a single community (i.e., people of class “x” is spread across multiple communities instead of just one community). The main takeaway is that network 1 does well in ensuring that each community only contains one class/circle (high homogeneity), but it lacks in keeping each class within only one of the communities (low completeness).

As for network 2, it does seem to have some homogeneity, but not to an extent as the first network. However, it does seem to have the lowest completeness with a value of -3.423962. We believe that the reason for the negative value is because of the way the Aij ­matrix is calculated. There may be a redundant number of nodes that were accounted for. We eliminated duplicate nodes when computing “b” and “N” during this process.

Lastly, it seems that the final network virtually has no homogeneity. This may be because of there are many more communities (31) than circles (3). Hence, the probability of having a community in the network composing of only one class diminishes. As for its completeness, the communities and circles are essentially mixed with each other (i.e., they are not well-separated), causing a low completeness score, as it does not do a good job in ensuring that all nodes/people of a particular class are within only one community. Rather, the people of a particular class are spread across multiple communities. We believe the reasoning behind the negative completeness score is the same as network 2’s completeness score that the computation of the Aij matrix may be accumulating redundancy, causing it to be greater than bj and/or aj.

The results for the homogeneity and completeness of the personal networks are shown below:

|  |  |  |
| --- | --- | --- |
| Network ID | Homogeneity | Completeness |
| 109327480479767108490 | 0.8518851 | 0.3298739 |
| 115625564993990145546 | 0.4518903 | -3.423962 |
| 101373961279443806744 | 0.003866707 | -1.504238 |

NOTE: Base e was used to calculate the conditional entropies.