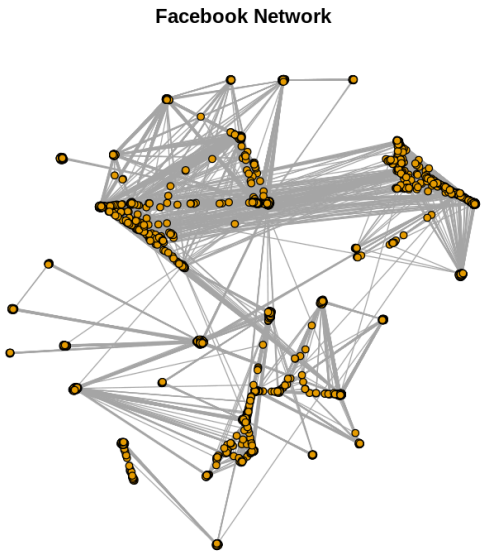
Project 2 - Social Network Mining

Aryaman Gokarn **Mugdha Bhagwat** **Tania Rajabally**
UID:506303588 **UID: 606297799** **UID: 806153219**

FACEBOOK NETWORK

QUESTION 1:
A first look at the network:

QUESTION 1.1:
Report the number of nodes and number of edges of the Facebook network.



Refer to the network above.
The number of nodes are: 4039
The number of edges are: 88234

QUESTION 1.2:
Is the Facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

The Facebook network is connected. This means that there are no isolated users (nodes) without any friends in the network and all users have some connection (edges) with other users. Since this graph is connected, we do not find the GCC. But, the code above will first check if the network is connected. If not, it calculates all connected components, identifies the largest one (i.e., the GCC), and then extracts this subgraph to determine its size. The size of the GCC is given in terms of the number of vertices it contains.

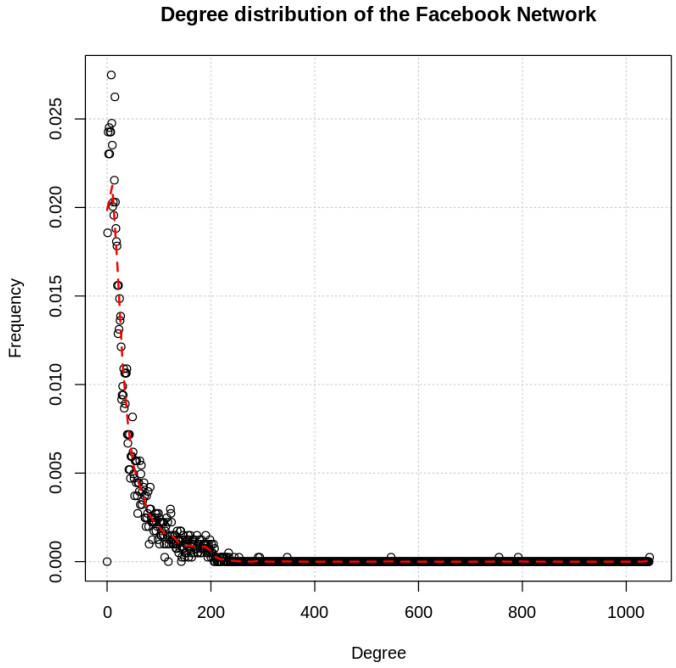
QUESTION 2:

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

The diameter of the network is 8.
The diameter function finds the longest of all the shortest paths between any pair of vertices in the graph. It gives you the diameter based on unweighted paths. Since the network is connected, we find the diameter of the network.

QUESTION 3:

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



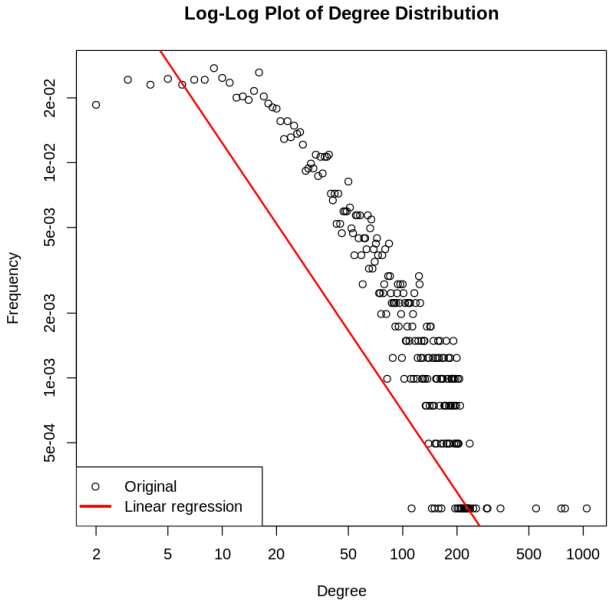
Please refer to the degree distribution above.

The average degree is 43.6910126268878.

The function degree(graph) retrieves the number of connections for each node in the network. The average degree is calculated as the mean of the degree data.

QUESTION 4:

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



Refer to the plot above.

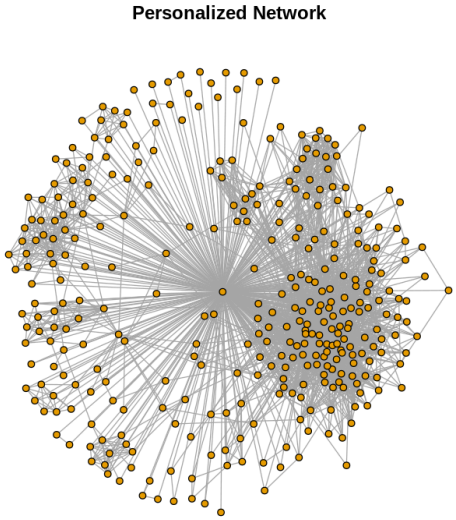
The estimated slope of the line is -1.18016441925088.

We see that the degree-distribution in the log-log scale is approximately linear. In preferential attachment models, nodes with higher degrees or connectedness are more likely to receive edges with newer nodes without any heuristics on whether an incoming node is related to a high-degree node or not. This results in a sparse network with a large number of high-degree nodes. However, for social networks, users tend to form communities with known (mutual) friends, leading to densely packed communities of known users with heterogeneous connectedness.

QUESTION 5:

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

Hint Useful function(s): makeegograph



Refer to the personalized graph above.

The Number of nodes in the personalized network is 348.

The Number of edges in the personalized network: 2866.

QUESTION 6:

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

The diameter of the personalized network is 2.
The trivial upper bound for diameter is 2.
The trivial lower bound for diameter is 1.
Trivial upper bound for diameter: It is one less than the number of nodes in the network because in the worst-case scenario, the diameter occurs when the graph is a linear chain.
Trivial lower bound for diameter: If the personalized network contains only one node, its diameter is trivially 0. Otherwise, it's 1, as there must be at least one edge between two nodes for them to be connected.

QUESTION 7:

In the context of the personalized network, what is the meaning of the diameter of the personalized network to be equal to the upper bound you derived in Question 6. What is the meaning of the diameter of the personalized network to be equal to the lower bound you derived in Question 6 (assuming there are more than 3 nodes in the personalized network)?

The diameter of a graph is defined as the largest distance between any pair of nodes (greatest shortest path between any two nodes). If the diameter of a personalized network is 2, it means that there exists users who are not connected directly in the subgraph, but are connected through a mutual user. On the other hand, if the diameter of a personalized network is 1, then all users in the network know each other directly and there exists direct edges between all vertices in the network. The network in this case is fully connected. If the diameter of the personalized network equals the upper bound (N-1, where N is the number of nodes in the network), it indicates a linear or path-like structure. The network forms a path where each node is connected sequentially to the next, and there are no shortcuts or additional connections that decrease the maximum distance between any two nodes. In this configuration, each node (except for those at the ends) has exactly two neighbors, and the end nodes have only one neighbor. This implies that communication or interaction within the network must pass through multiple intermediaries, potentially reducing efficiency and increasing the steps required to reach from one end of the network to the other.

If the diameter of the personalized network is equal to the lower bound it indicates a complete graph or clique. Every node is directly connected to every other node in the network. This configuration indicates a highly interconnected network where information or influence can spread quickly and efficiently. Each node has direct and immediate access to all other nodes, facilitating rapid communication and potentially leading to a robust network where nodes can quickly mobilize or respond to changes.

QUESTION 8:

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

The number of core nodes in the Facebook network is 40.
The average degree of the core nodes is 279.375.

QUESTION 9:

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

Hint Useful function(s): clusterfastgreedy , clusteredgebetweenness , clusterinfomap

Community structure of a graph is defined as a clustering of the vertices in the network such that the number of inter cluster edges are much smaller than the number of intra cluster edges. In other words, community structures segregate regions of high connectedness from the overall sparse network structure.

The Fast-Greedy algorithm is an efficient community detection method designed to optimize the modularity of a network. This algorithm begins by treating each node as its own community and iteratively merges communities to maximize modularity gains. In each step, it calculates the potential increase in modularity for every possible pair of community merges, selecting the pair that offers the highest increase. This process is repeated until no further increase in modularity is possible, indicating the formation of distinct communities. Due to its computational efficiency, the Fast-Greedy algorithm is particularly useful for analyzing large networks quickly.

The Edge Betweenness algorithm is a community detection method that focuses on identifying and removing edges that serve as bridges between different communities in a network. The central concept involves calculating the "betweenness centrality" for all edges, which measures the number of shortest paths passing through each edge. In practice, edges with high betweenness centrality are critical for connecting different parts of the network, indicating potential community boundaries. The algorithm iteratively removes edges with the highest betweenness scores, which progressively fragments the network into distinct communities. This process continues until no edges remain or a predefined number of communities is reached. The Edge Betweenness algorithm is especially valuable for uncovering natural divisions within complex networks but can be computationally intensive for larger networks.

The Infomap algorithm is a network community detection method that leverages information theory to uncover the modular structure of networks. It is based on the idea of random walks and utilizes the flow of these walks to reveal community structures. Specifically, Infomap minimizes the expected description length of a random walker's path through the network by finding a compact map description that best compresses the information in the network. This is achieved by partitioning the network into clusters (or communities) such that the random walk is highly predictable within clusters but less so between them. Infomap's effectiveness lies in its ability to detect multiple levels of communities in large and complex networks, making it a powerful tool for exploring the hierarchical organization of nodes based on the dynamics of flow within the network.

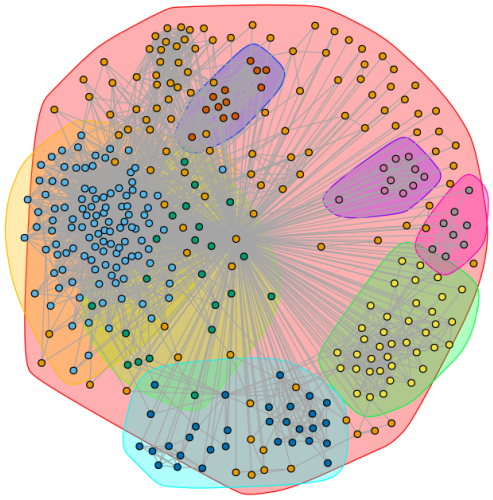
Node Id 1:

Modularity using Fast Greedy for Node ID 1: 0.413101

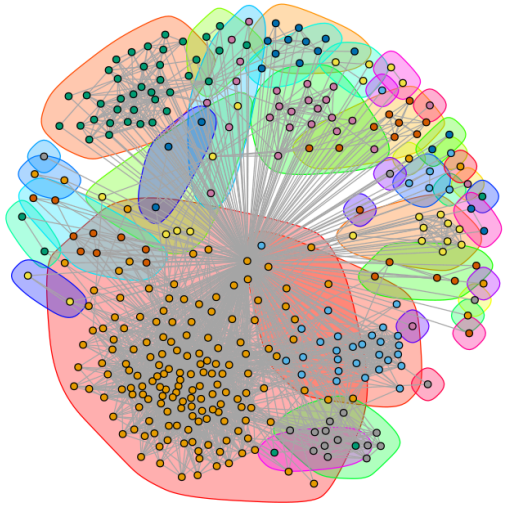
Modularity using Edge-Betweenness for Node ID 1: 0.353302

Modularity using Infomap for Node ID: 1: 0.389118

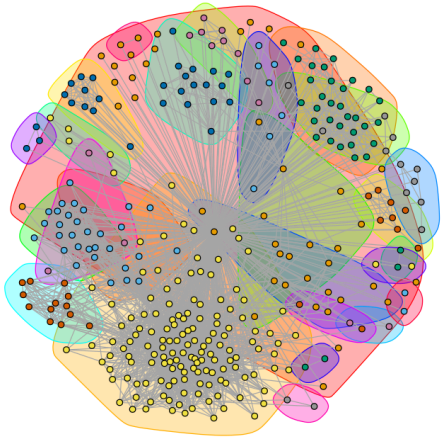
Community Structure using Fast Greedy for Node ID: 1



Community Structure using Edge-Betweenness for Node ID: 1



Community Structure using Infomap for Node ID: 1



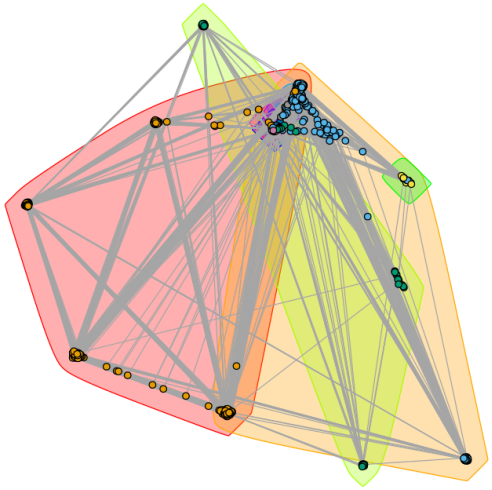
Node Id 108:

Modularity using Fast Greedy for Node ID 108: 0.435929

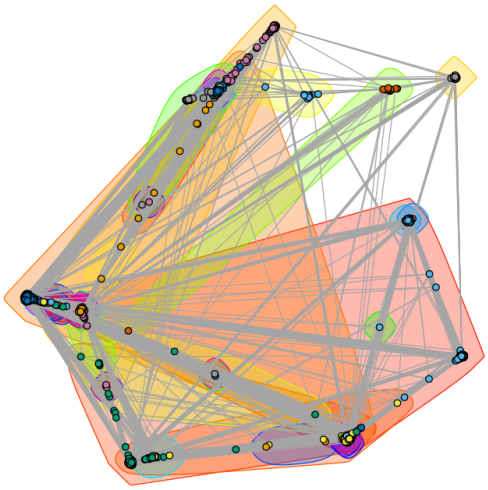
Modularity using Edge-Betweenness for Node ID 108: 0.506755

Modularity using Infomap for Node ID: 108: 0.508223

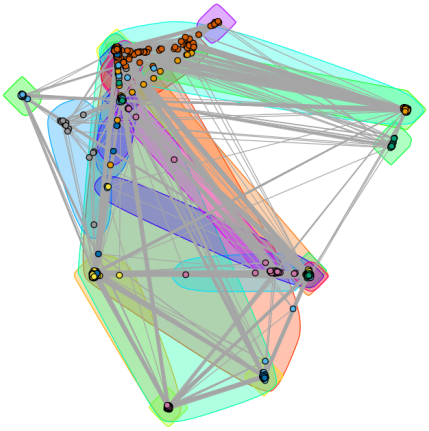
Community Structure using Fast Greedy for Node ID: 108



Community Structure using Edge-Betweenness for Node ID: 108

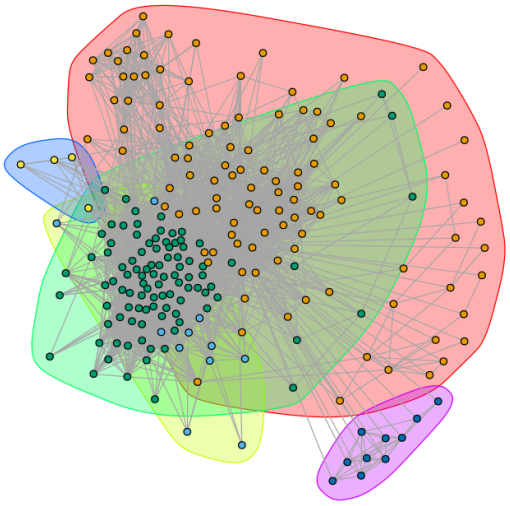


Community Structure using Infomap for Node ID: 108

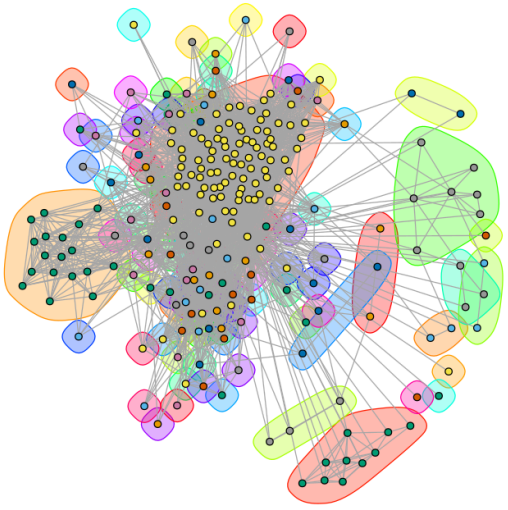


Node Id 349:
Modularity using Fast Greedy for Node ID 349: 0.251715
Modularity using Edge-Betweenness for Node ID 349: 0.133528
Modularity using Infomap for Node ID: 349: 0.095464

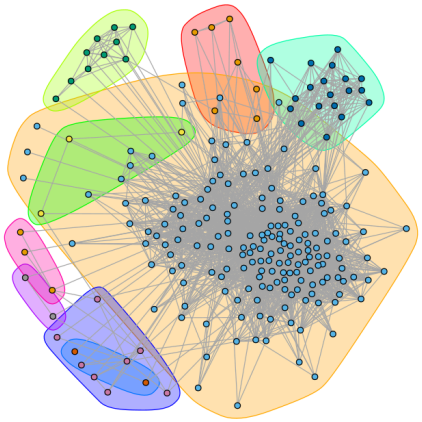
Community Structure using Fast Greedy for Node ID: 349



Community Structure using Edge-Betweenness for Node ID: 349

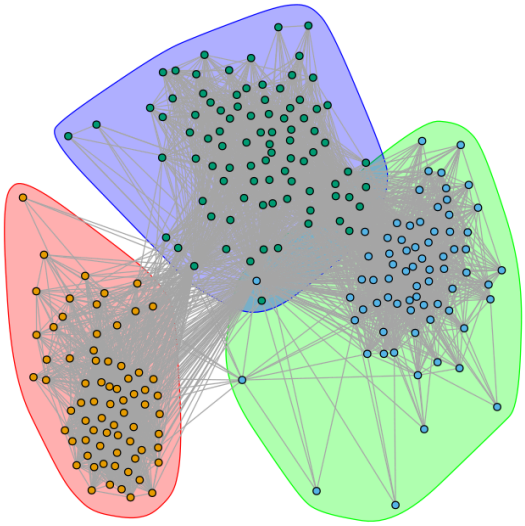


Community Structure using Infomap for Node ID: 349

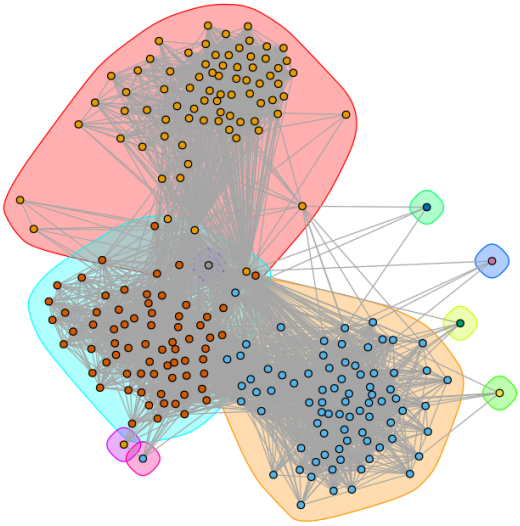


Node Id 484:
Modularity using Fast Greedy for Node ID 484: 0.507002
Modularity using Edge-Betweenness for Node ID 484: 0.489095
Modularity using Infomap for Node ID: 484: 0.515279

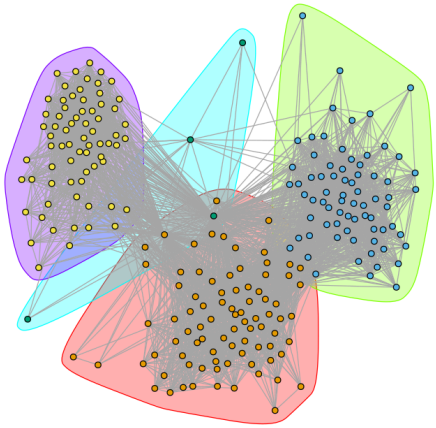
Community Structure using Fast Greedy for Node ID: 484



Community Structure using Edge-Betweenness for Node ID: 484

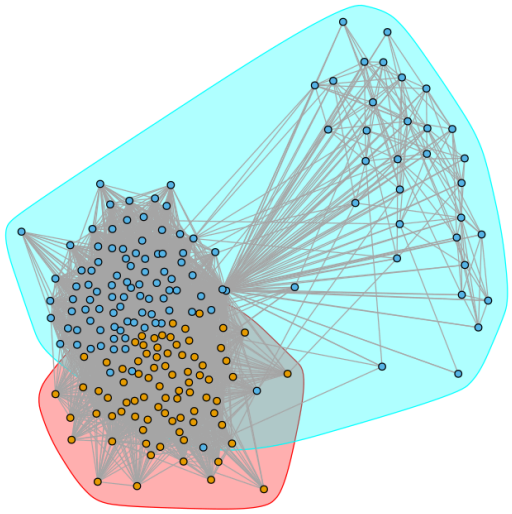


Community Structure using Infomap for Node ID: 484

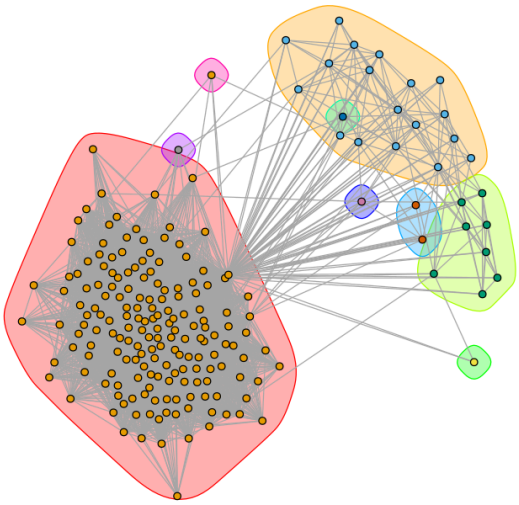


Node Id 1087:
Modularity using Fast Greedy for Node ID 1087: 0.145531
Modularity using Edge-Betweenness for Node ID 1087: 0.027624
Modularity using Infomap for Node ID: 1087: 0.026907

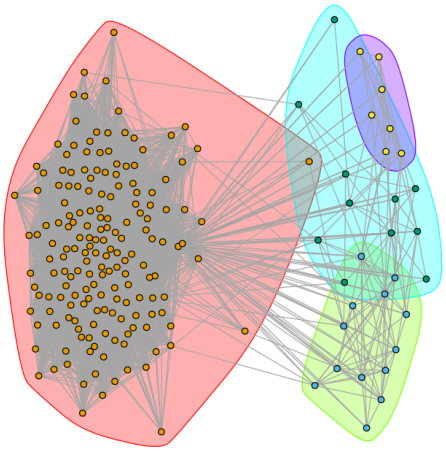
Community Structure using Fast Greedy for Node ID: 1087



Community Structure using Edge-Betweenness for Node ID: 1087



Community Structure using Infomap for Node ID: 1087

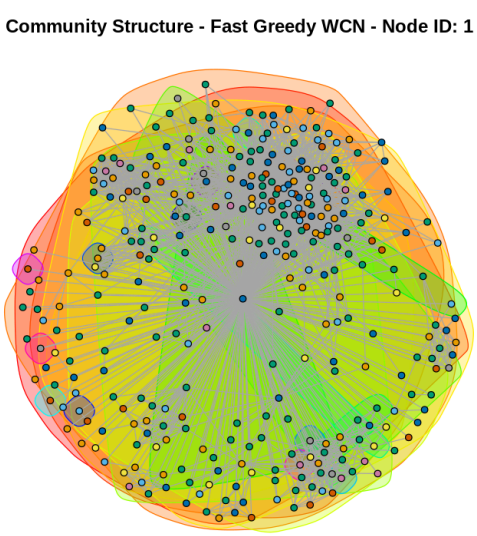


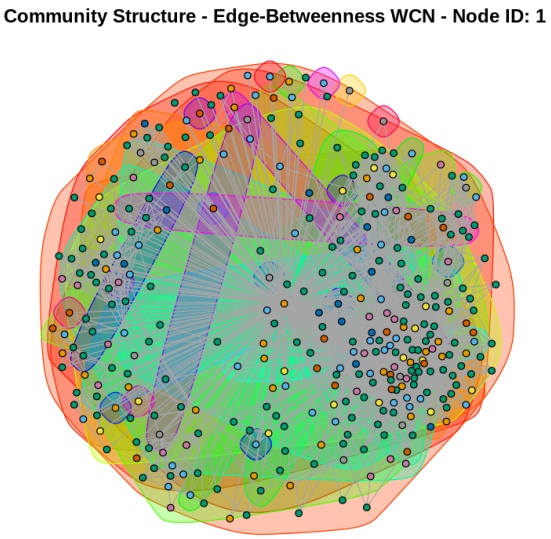
QUESTION 10:

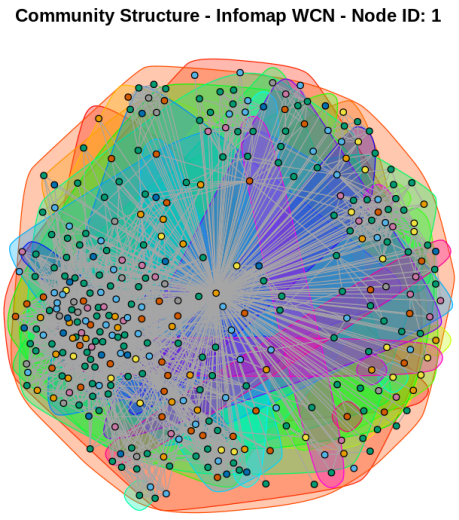
For each of the core node's personalized network (use same core nodes as Question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as Question 9. Compare the modularity score of the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

Node Id 1:

Modularity using Fast Greedy for Node ID (without core node): 1: 0.441853
Modularity using Edge-Betweenness for Node ID (without core node): 1: 0.416146
Modularity using Infomap for Node ID (without core node): 1: 0.418008







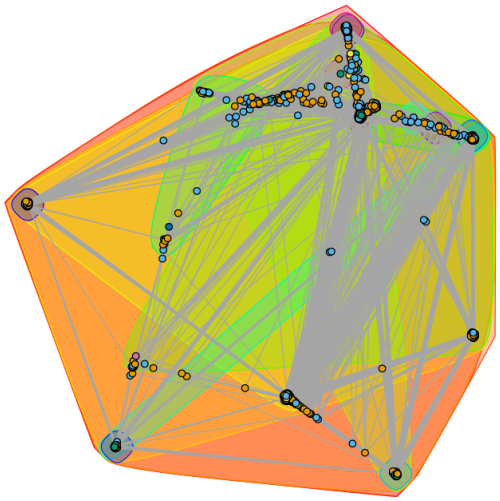
Node Id 108:

Modularity using Fast Greedy for Node ID (without core node): 108: 0.458127

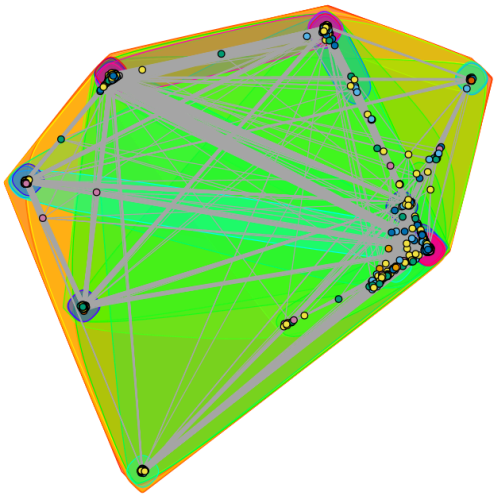
Modularity using Edge-Betweenness for Node ID (without core node): 108: 0.521322

Modularity using Infomap for Node ID (without core node): 108: 0.520517

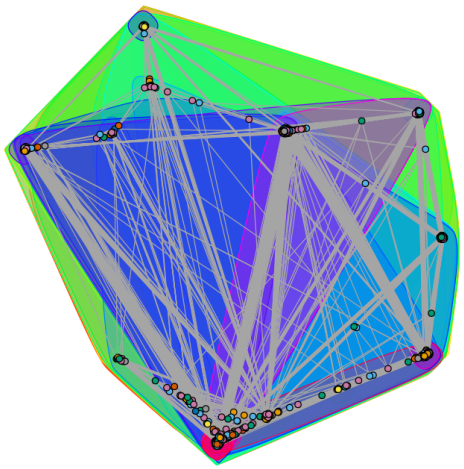
Community Structure - Fast Greedy WCN - Node ID: 108



Community Structure - Edge-Betweenness WCN - Node ID: 108



Community Structure - Infomap WCN - Node ID: 108



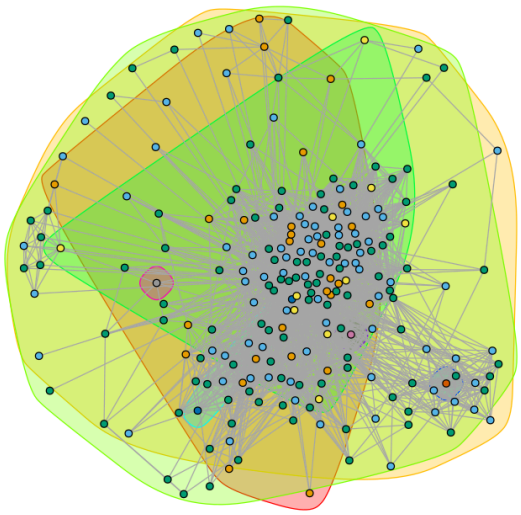
Node Id 349:

Modularity using Fast Greedy for Node ID (without core node): 349: 0.245692

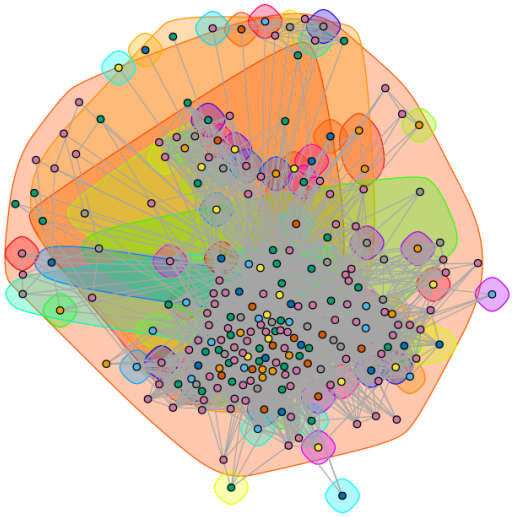
Modularity using Edge-Betweenness for Node ID (without core node): 349: 0.150566

Modularity using Infomap for Node ID (without core node): 349: 0.246578

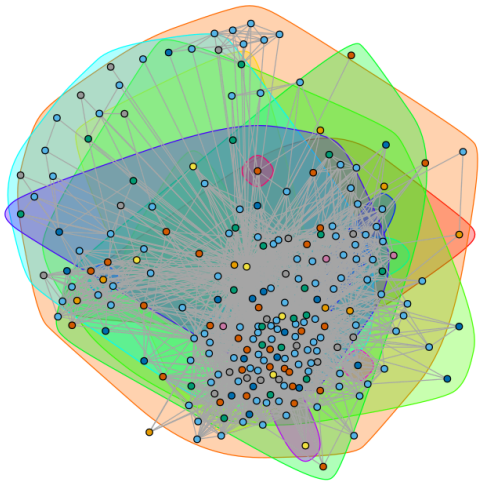
Community Structure - Fast Greedy WCN - Node ID: 349



Community Structure - Edge-Betweenness WCN - Node ID: 349



Community Structure - Infomap WCN - Node ID: 349



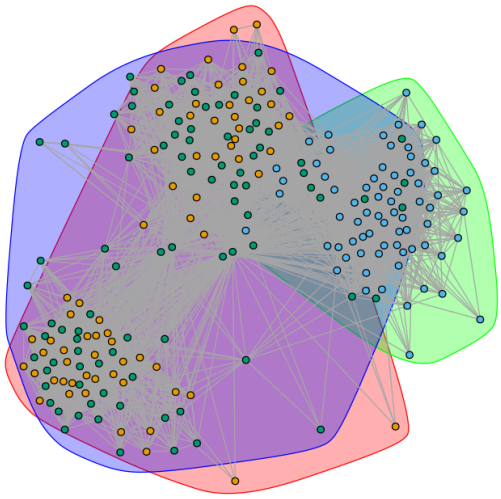
Node Id 484:

Modularity using Fast Greedy for Node ID (without core node): 484: 0.534214

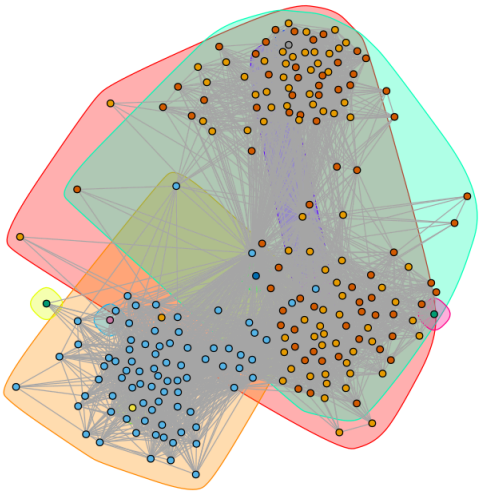
Modularity using Edge-Betweenness for Node ID (without core node): 484: 0.515441

Modularity using Infomap for Node ID (without core node): 484: 0.543444

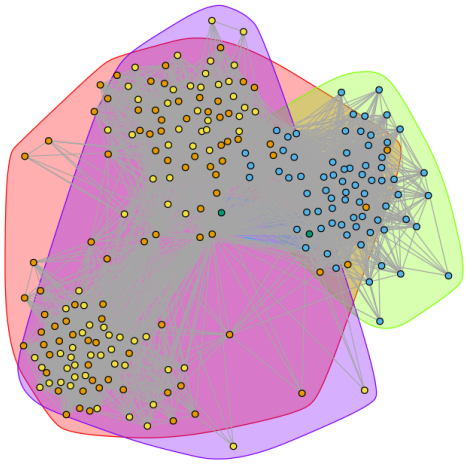
Community Structure - Fast Greedy WCN - Node ID: 484



Community Structure - Edge-Betweenness WCN - Node ID: 484



Community Structure - Infomap WCN - Node ID: 484



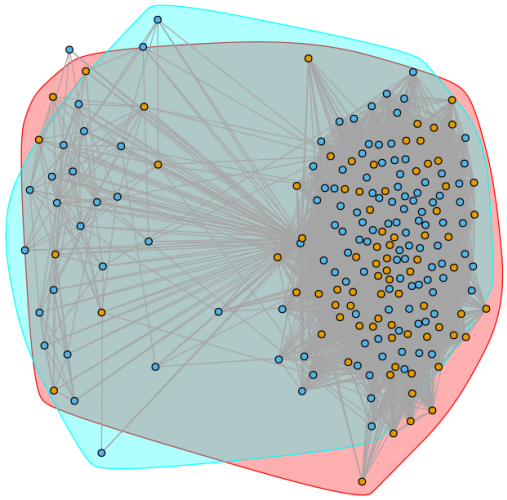
Node Id 1087:

Modularity using Fast Greedy for Node ID (without core node): 1087: 0.148196

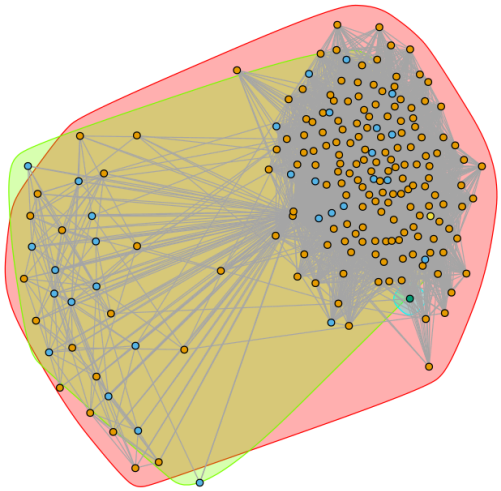
Modularity using Edge-Betweenness for Node ID (without core node): 1087: 0.032495

Modularity using Infomap for Node ID (without core node): 1087: 0.027372

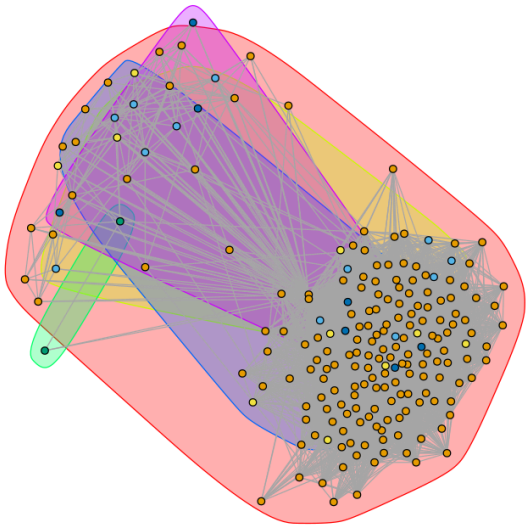
Community Structure - Fast Greedy WCN - Node ID: 1087



Community Structure - Edge-Betweenness WCN - Node ID: 1087



Community Structure - Infomap WCN - Node ID: 1087



We can see that the modularity scores have increased after removing the core node for all 5 personalized networks. This is because the core node acts as the bridge between all of the other nodes, which causes the network with the core node less capable of being partitioned into communities with strong intra-community connectedness and sparse inter-community connectedness compared to the network without the core node. With the presence of the core node, it is difficult to classify the core node into a single community as it is connected to every other node while also making it difficult to assign densely packed and sparsely inter-connected communities among the other nodes, resulting in a dense network with low modularity. In addition, with the core node present, the other nodes are not always directly connected to each other or the rest of the graph, resulting in many isolated communities (communities with a single node) for edge-betweenness. With the core node removed, the edges from the core node to all other nodes are removed, allowing the community partition algorithms to find densely packed areas of high connectedness with sparse inter-community edges. The probability of strong intra-community connections is greater for networks with core nodes removed, and likewise, the extent of sparsity among communities is amplified in such networks. We can also see that all the algorithms, especially edge-betweenness, have a decreased tendency of forming groups with isolated nodes. In addition, the community structures are still well-defined and well-partitioned, despite absence of the core nodes.

QUESTION 11:

Write an expression relating the Embeddedness between the core node and a non-core node to the degree of the non-core node in the personalized network of the core node.

Embeddedness in a network, specifically within the context of a core node and a non-core node in a personalized network, measures the number of shared neighbors between the two nodes. The embeddedness between the core node c and the non-core node v, denoted as E(c,v), can be defined as the number of common neighbors between c and v. The degree of the non-core node v in the personalized network, denoted as deg(v), is the total number of connections v has within this network. The embeddedness of a node is defined as the number of mutual vertices a given node shares with the core node. Embeddedness is
defined as the overlap in the social circle of two users.

$$\text{Embeddedness}^{P}_{v_i,v_c} = deg(v_i)^P - 1$$

$$\text{Embeddedness}_{v_i,v_c} = deg(v_i) - N_{v_i} - 1 = |deg(v_c) \cap deg(v_i)| - 1$$

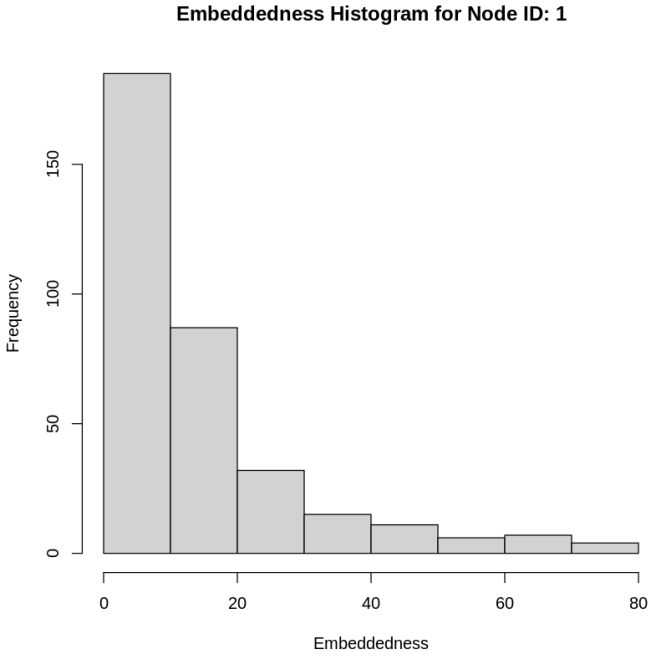
QUESTION 12:

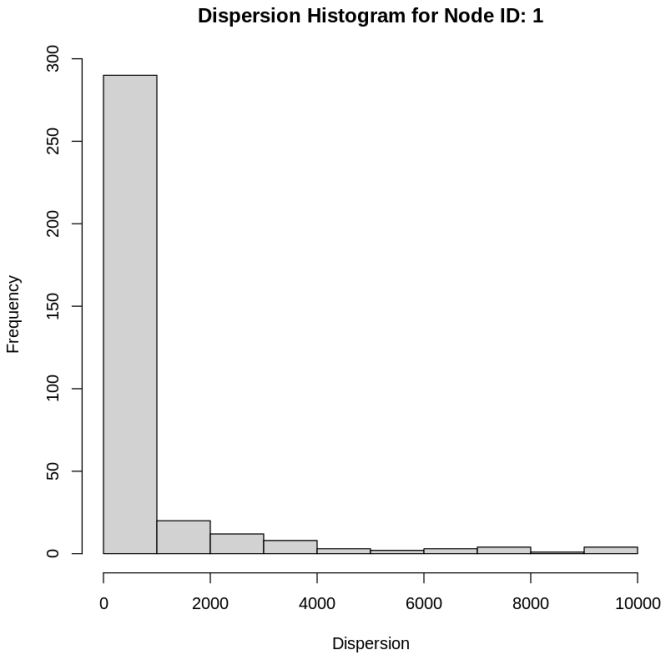
For each of the core node's personalized network (use the same core nodes as Question 9), plot the distribution histogram of embeddedness and dispersion. In this question, you will have 10 plots.
Hint Useful function(s): neighbors , intersection , distances

Dispersion of a node is defined as the sum of distances between every pair of the mutual vertices the node shares with the core node, calculated in a modified subgraph graph with the target node and core node removed.

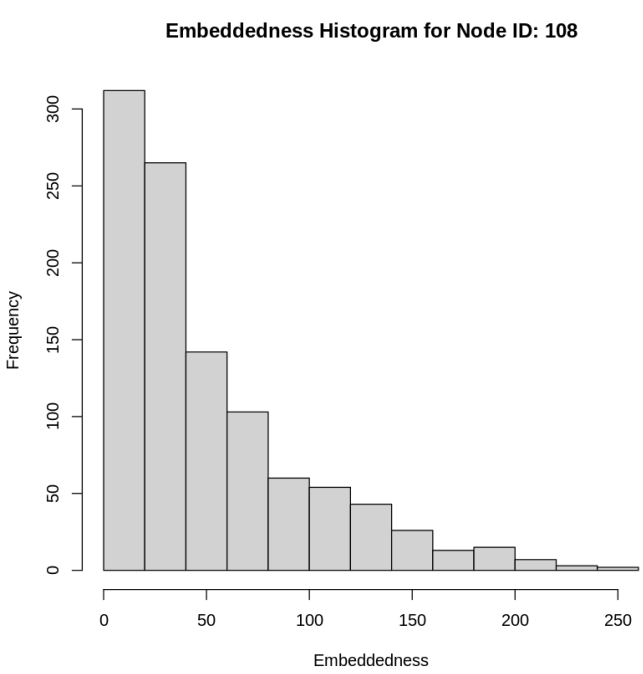
$$disp(u,v) = \sum_{s,t \in C_{uv}} d_v(s,t)$$

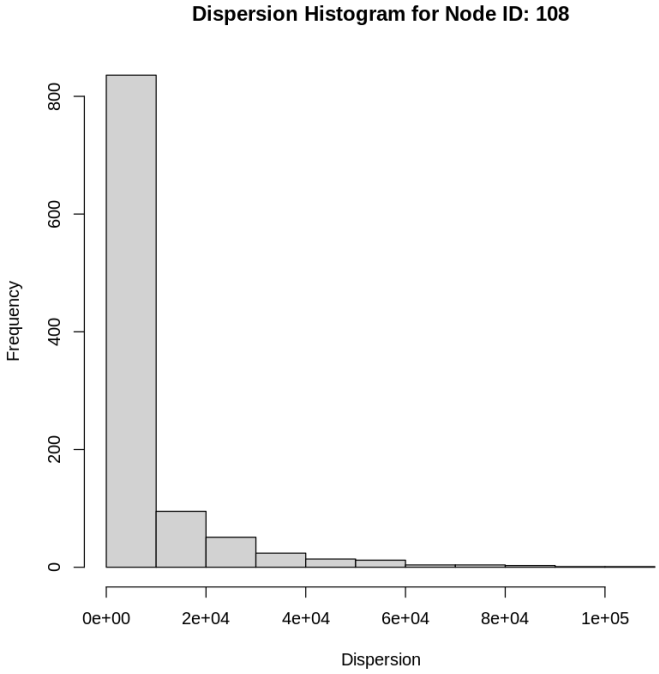
Node Id 1:



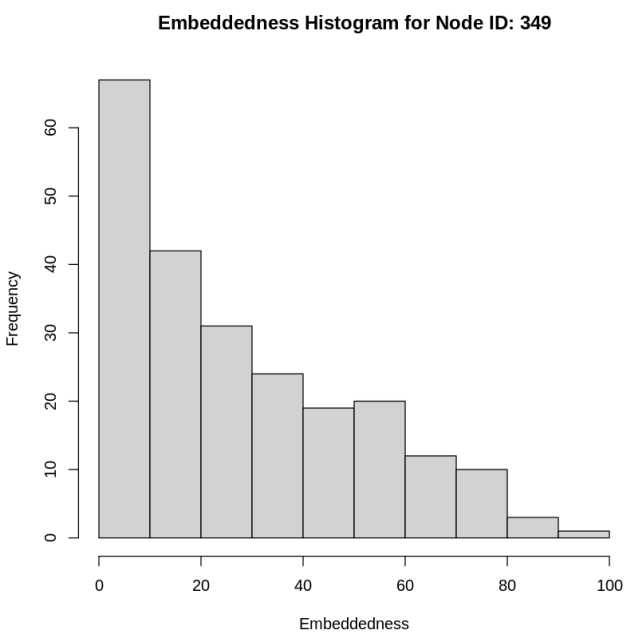


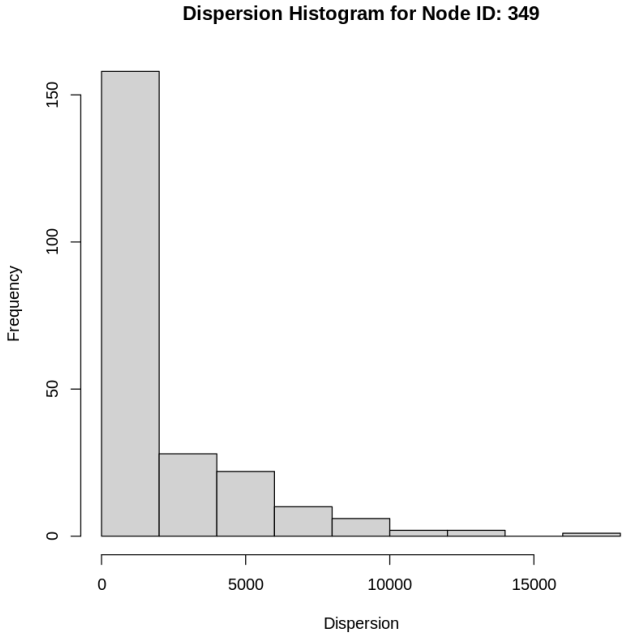
For Node Id 108:



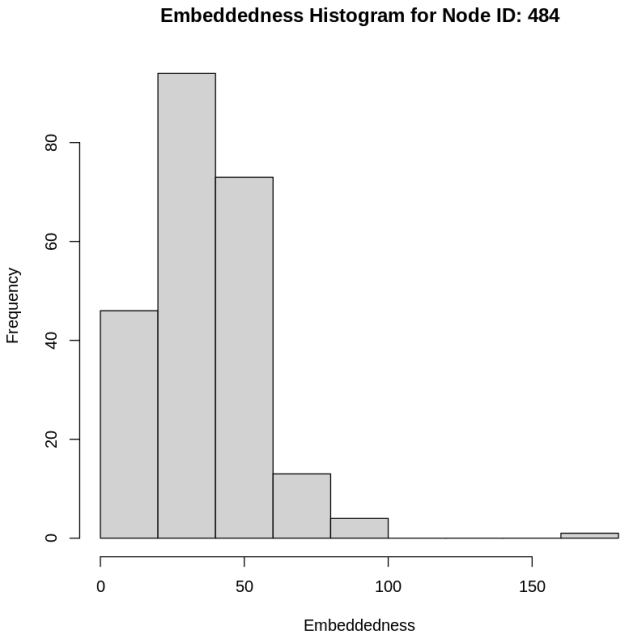


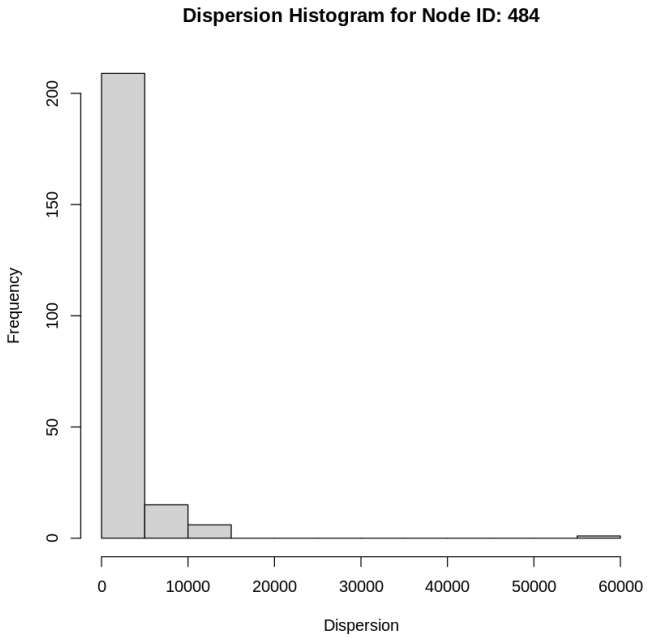
For Node Id 349:



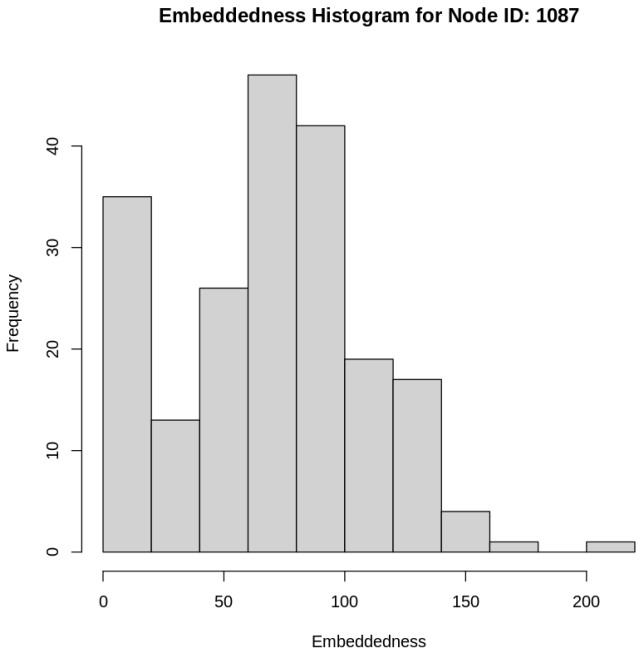


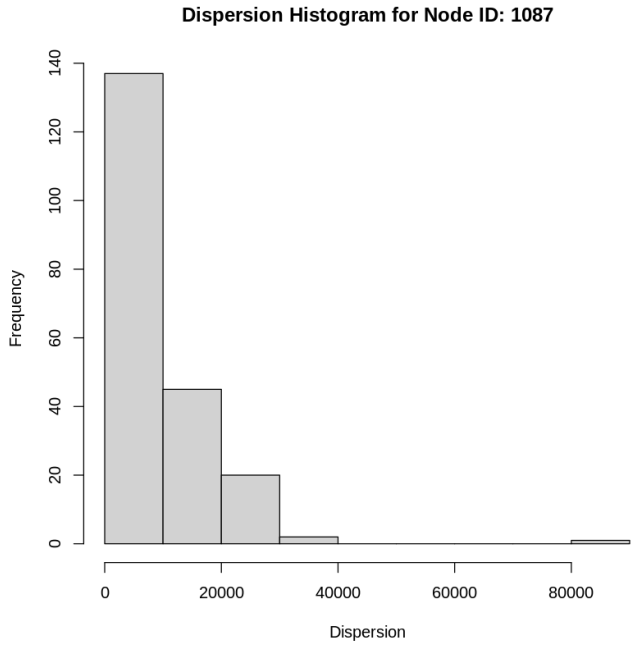
For Node Id 484:





For Node Id 1087:



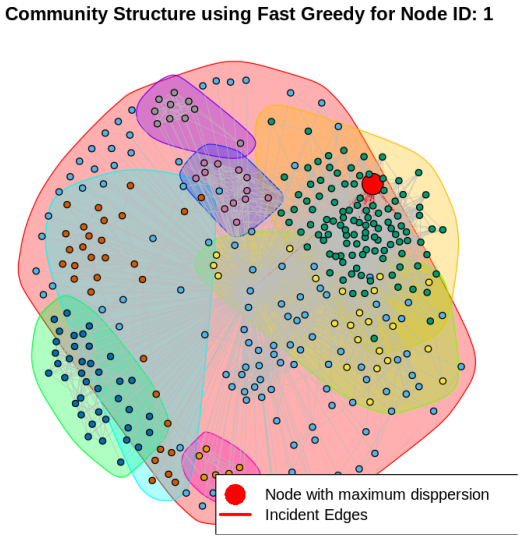


From the plots above, we can see that the range of dispersions depends on the number of edges (degree) of the core node in the personalized network, the higher the number of edges, the higher is the dispersion. In addition, the expected value of embeddedness is dependent on the number of edges in the personalized network. Both embeddedness and dispersion roughly follow the power-law distribution due to their inherent relation with the degree distribution of the personalized networks, which follow power-law distribution stemming from weak preferential attachment. Dispersion measures the extent to which two people's mutual friends are not themselves well-connected, in other words, the mutual nodes of two strongly connected nodes are not well connected and must display a dispersed structure. Embeddedness depends on the number of nodes in the personalized network. The higher the number of nodes and connectivity among them in the social circle, the greater is the embeddedness, even though embeddedness is not a good indicator of strong ties, as strongly tied nodes may have low degrees of embeddedness.

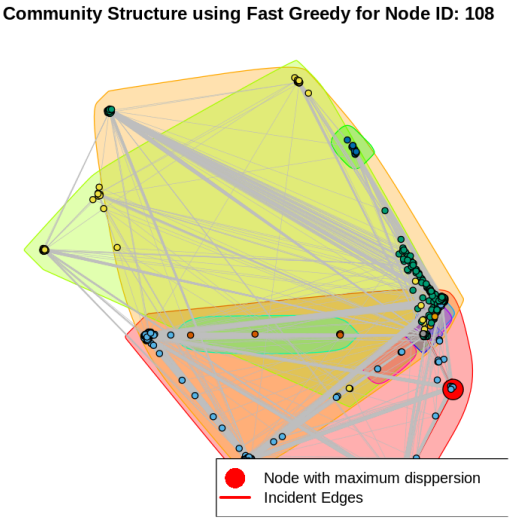
QUESTION 13:

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

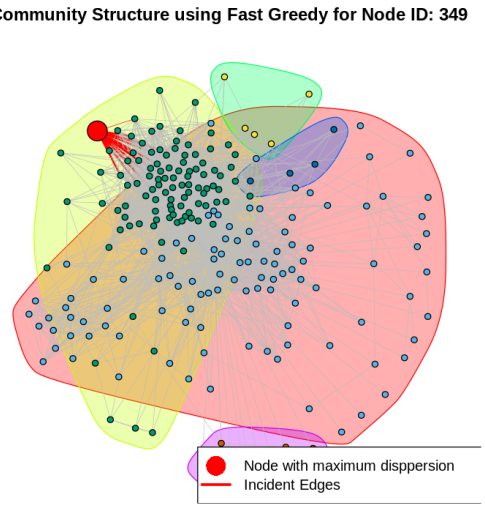
Node with Maximum Dispersion for Node Id 1 is : 49



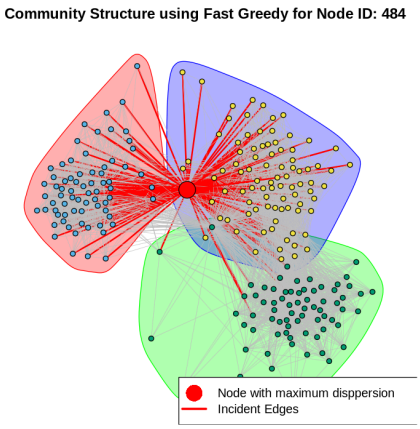
Node with Maximum Dispersion for Node Id 108 is : 977



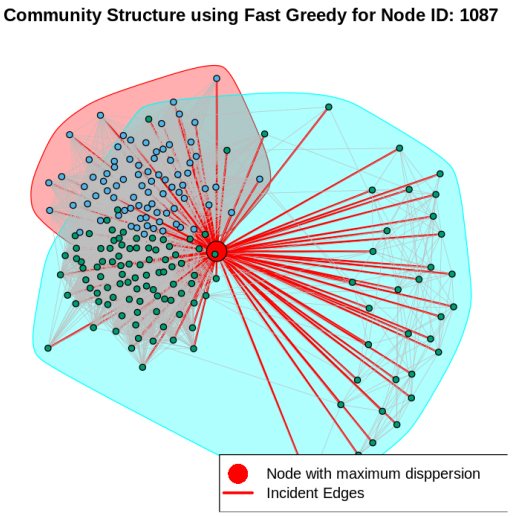
Node with Maximum Dispersion for Node Id 349 is : 31



Node with Maximum Dispersion for Node Id 484 is : 1



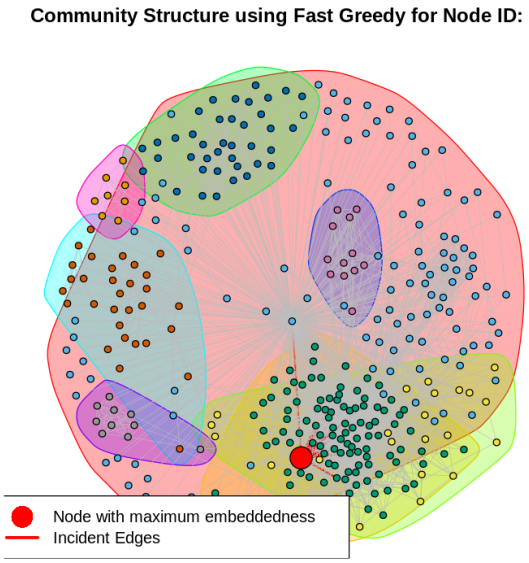
Node with Maximum Dispersion for Node Id 1087 is : 1



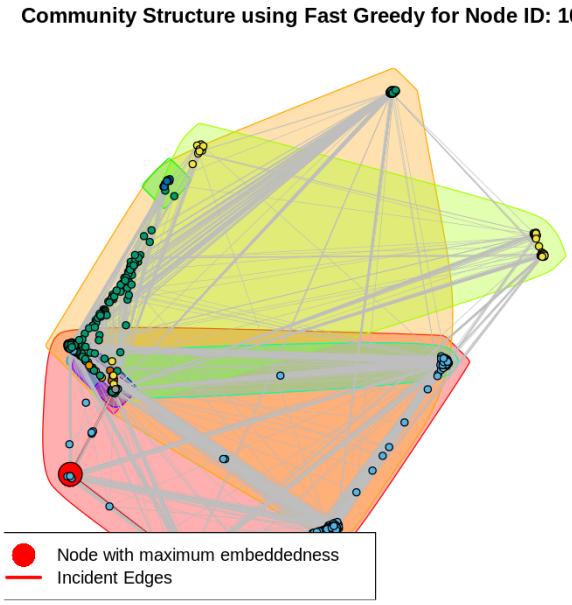
QUESTION 14:

Repeat Question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion/embeddedness (excluding the nodes having zero embeddedness if there are any). Also, highlight the edges incident to these nodes. Report the id of those nodes.

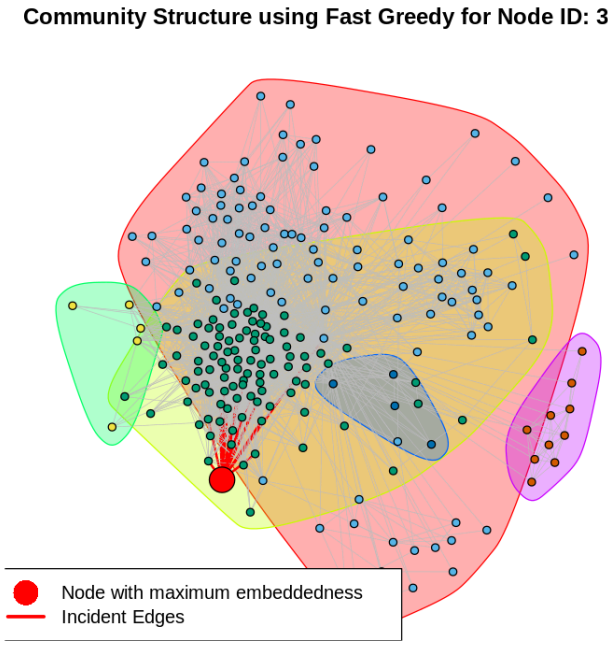
ID of node with maximum embeddedness for core node with ID 1 is: 49



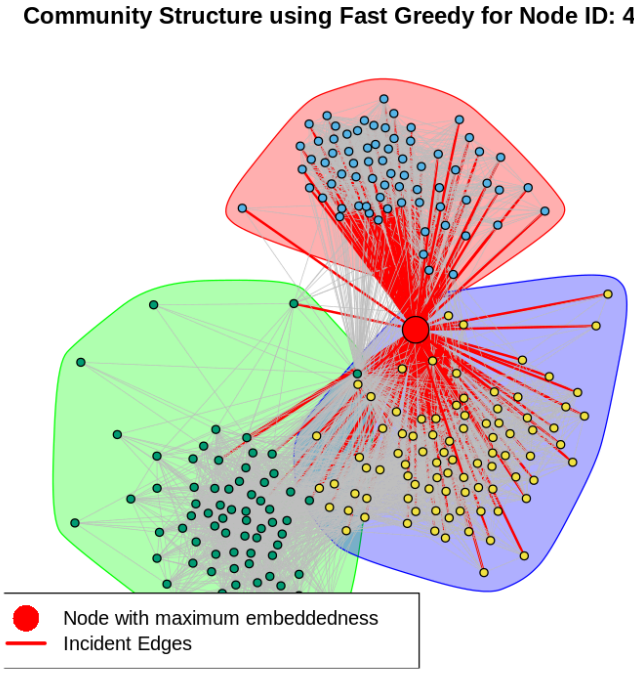
ID of node with maximum embeddedness for core node with ID 108 is: 977



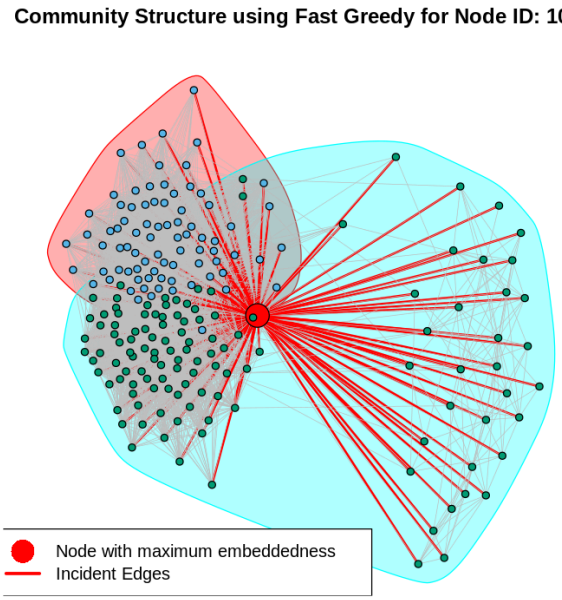
ID of node with maximum embeddedness for core node with ID 349 is: 31



ID of node with maximum embeddedness for core node with ID 484 is: 1

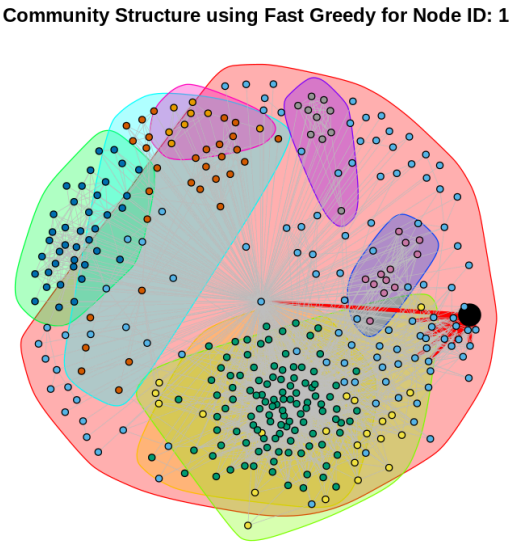


ID of node with maximum embeddedness for core node with ID 1087 is: 1

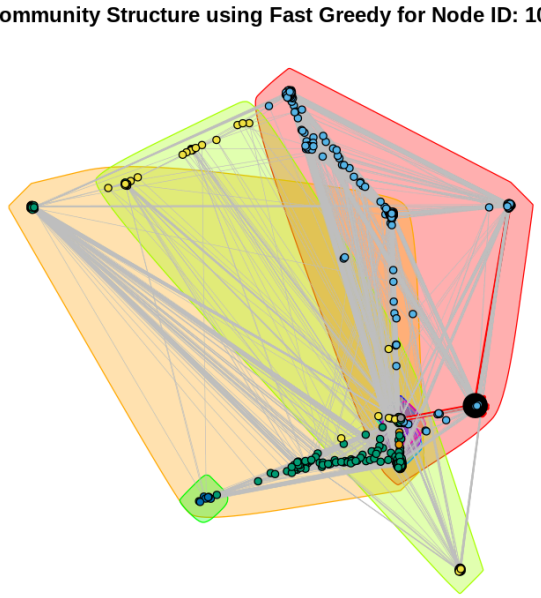


The nodes with the maximum embeddedness/dispersion is as below:

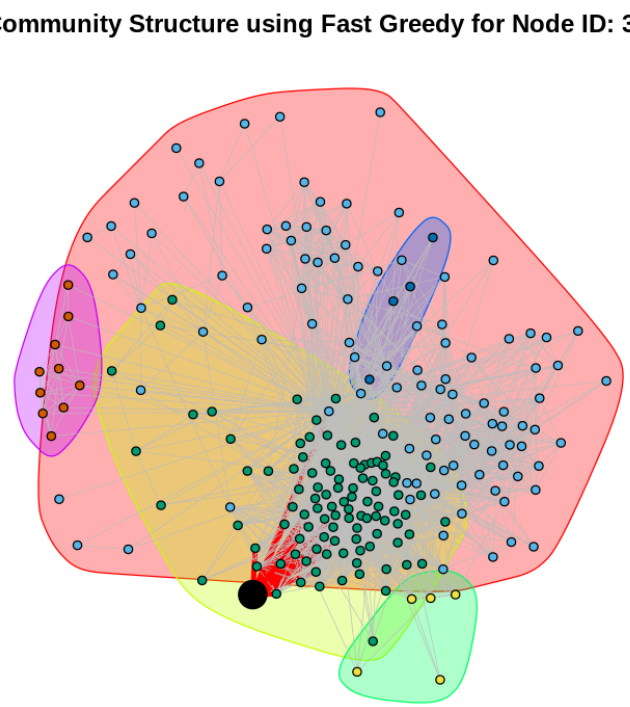
ID of node with maximum dispersion/embeddedness ratio for core node with ID 1 is 21



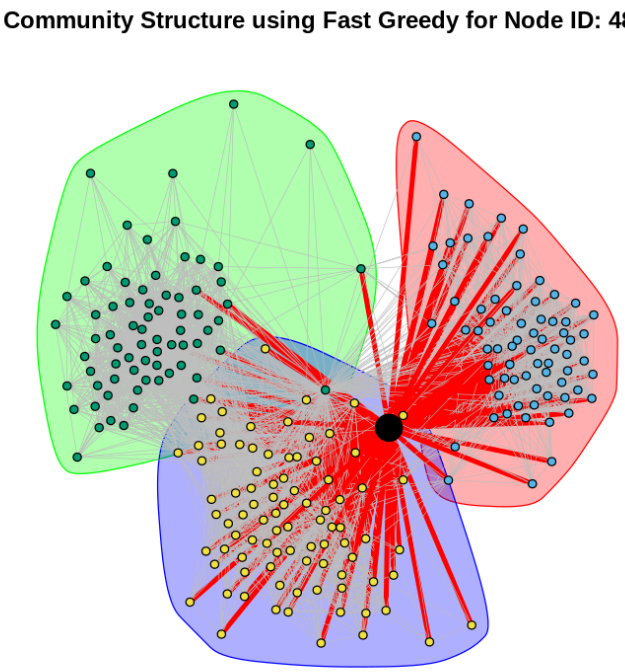
ID of node with maximum dispersion/embeddedness ratio for core node with ID 108 is 977



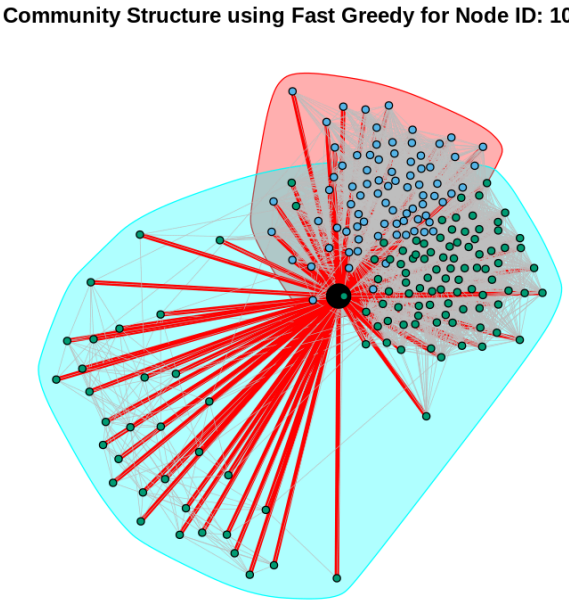
ID of node with maximum dispersion/embeddedness ratio for core node with ID 349 is 31



ID of node with maximum dispersion/embeddedness ratio for core node with ID 484 is 1



ID of node with maximum dispersion/embeddedness ratio for core node with ID 1087 is 1



QUESTION 15:

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

We defined embeddedness of a node as the number of mutual vertices a given node shares with the core node. Embeddedness is defined as the overlap in the social circle of two users. Embeddedness depends on the number of nodes in the personalized network. The higher the number of nodes and connectivity among them in the social circle, the greater is the embeddedness. The expected value of embeddedness is dependent on the number of edges in the personalized network. In addition, large communities are likely to have a node with a large value of embeddedness. In addition, compared to dispersion, the density of incident edges from the target node is much higher. This is by definition of embeddedness, which is proportional to the degree of the node. However, this is not a good measure of the strength of ties, as users with strong relationships are likely to be associated with users from various different communities.

We defined the dispersion of a node as the sum of distances between every pair of the mutual vertices the node shares with the core node, calculated in a modified subgraph graph with the target node and core node removed. The range (as well as expected value) of dispersions depends on the number of edges (degree) of the core node in the personalized network, the higher the number of edges, the higher is the dispersion of a target node. In addition, dispersion measures the extent to which two people's mutual friends are not themselves well-connected, in other words, the mutual nodes of two strongly connected nodes are not well connected and must display a dispersed structure. As a result, the density of incident edges from a node with maximum dispersion will likely be lower than a node with maximum embeddedness, since the network must display a dispersed structure with connectedness among users from various different communities. Dispersion is going to be higher for those nodes whose connections are farther apart (dispersed) than the nodes with dense connectedness. As a result, dispersion is a better measure of strength of ties or relationship over embeddedness.

The maximum value of dispersion/embeddedness (normalization of dispersion) occurs when dispersion of the node is very high while the embeddedness is very low. Such a node is likely to have mutual friends from different communities with stronger ties. The ratio is higher for smaller networks with dispersed structures, with friends that are themselves not strongly connected.

QUESTION 16:

What is |Nr|, i.e. the length of the list Nr?

The length of Nr is 11.

QUESTION 17:

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

- **Common Neighbors measure**
- **Jaccard measure**
- **Adamic Adar measure**

Based on the average accuracy values, which friend recommendation algorithm is the best? Hint Useful function(s): similarity

Common neighbors is based on the assumption that if two nodes have many common neighbors, then the probability of them being connected in the future is high as well. The likelihood of future connectedness between two users is directly proportional to the number of mutual friends the users have. The score is based on the notion that people with common neighbors will be introduced to each other by that mutual friend (closing a triangle).
Jaccard's coefficient is used to compute similarity of sample sets in statistics. In link prediction, all the friends of a node are denoted as a set and the prediction is done by ranking the similarity of the neighbor set for each pair of nodes. It is based on the notion that two users may have many mutual friends, but not all of them are due to strong ties when compared to the overall number of neighbors of each user. The coefficient ranges from 0 to 1. It takes into account the relative number of common neighbors, adjusted for degree of nodes.
Adamic Adar measure is based on the notion that if a mutual friend of two users has lots of friends, then it is less likely that the mutual friend will introduce the two people to each other compared to the case when the mutual friend had fewer neighbors. The more friends a node has, the lower the score, weighing neighbors with fewer friends more heavily. In other words, someone with few friends are more likely to introduce his friends to each other than someone with a lot of friends. The mutual friend of a pair of users with few neighbors contributes more to the Adamic Adar score.

The average accuracy of the friend recommendation algorithm using common measure is: 0.880303
The average accuracy of the friend recommendation algorithm using jaccard measure is: 0.850000
The average accuracy of the friend recommendation algorithm using adamic measure is: 0.879924

We see that the common neighbors algorithm performs the best, followed by Adamic Adar and Jaccard.
Note that the accuracies are very close for all three networks (within 4% of each other). In addition, Adamic Adar and common neighbors perform very similarly, with < 1% difference in accuracy.
Common neighbors versus Jaccard: Jaccard's coefficient is based on the assumption that only strongly tied nodes contribute to future friend recommendation (intersection over union). In other words, if the degree of the target nodes is high, then the Jaccard's coefficient will be low, with the implicit assumption that two users with too many friends may not have strong ties within their communities. However, if the nodes within a network are themselves not strongly tied in a network, then the Jaccard's coefficient will perform slightly worse than just taking the intersection over mutual friends (common neighbors). This is particularly prominent in small networks, where the probability of having strong ties is lower than that of larger networks.
Common neighbors versus Adamic Adar: Theoretically, Adamic Adar should perform better than common neighbors, as it gives more importance to nodes with few neighbors (rare neighbors), particularly the cases where . However, for such a small network, it may well be the case that the occurrence of such cases is low (explained by the low Jaccard score), and most nodes have a large number of neighbors in common. This is why common neighbors and Adamic Adar perform very similarly to each other.
The accuracies of the algorithms are within 4% of each other. This is mainly because we are operating on a small personalized network with relatively fewer nodes and edges compared to large social networks. In addition, we are recommending friends to users with a limited degree. The reach of the network is not all-inclusive of implicit assumptions made by each of the measures and instead adhere to the most common

assumptions, which does not yield significant differences in link prediction algorithm performance. In other words, we are computing local features over global similarity features, which does not truly reflect how social network friendship recommendations work on a large scale.

GOOGLE+ NETWORK

QUESTION 18:

How many personal networks are there?

The total number of Directed Personal Networks for users with more than 2 circles are: 57

QUESTION 19:

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

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

To answer this question we employed the following:

1. Plotting In-Degree and Out-Degree Distributions:

For each of the three personal networks with the given node IDs, we create histograms to visualize the distribution of in-degrees and out-degrees. In total, we generate six plots: three for in-degree distributions and three for out-degree distributions.

2. Comparing In-Degree and Out-Degree Distributions:

After plotting the distributions, we examine whether the personal networks exhibit similar patterns in their in-degree and out-degree distributions. We look for similarities or differences in the shapes, central tendencies, and spreads of the distributions across the networks.

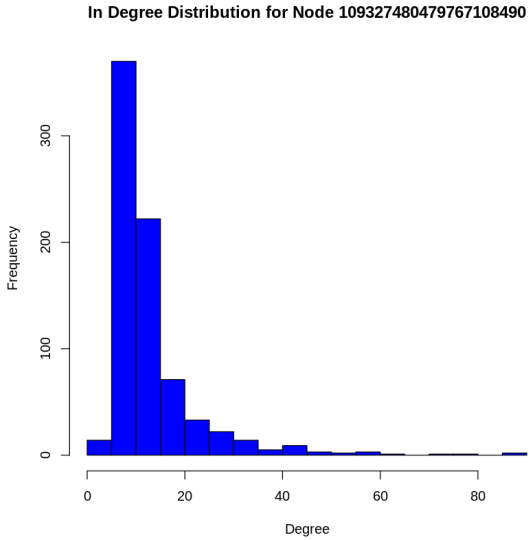
3. Interpretation:

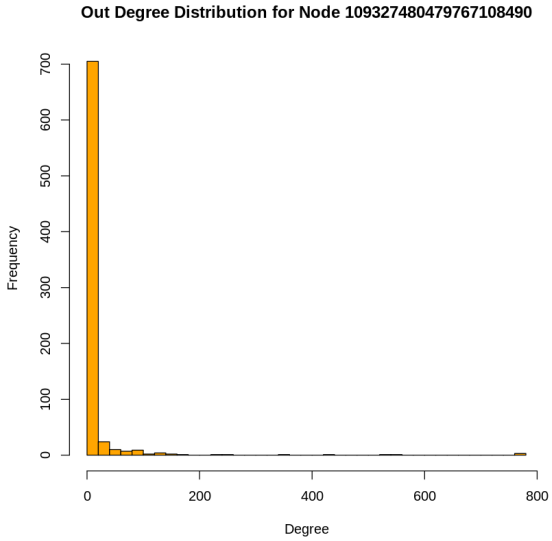
If the histograms for in-degree and out-degree distributions of the three personal networks show similar shapes, peaks, and spreads, it suggests that the networks have comparable patterns of connectivity. Conversely, if there are notable differences between the distributions, it indicates variations in the way nodes within each network receive and send connections.

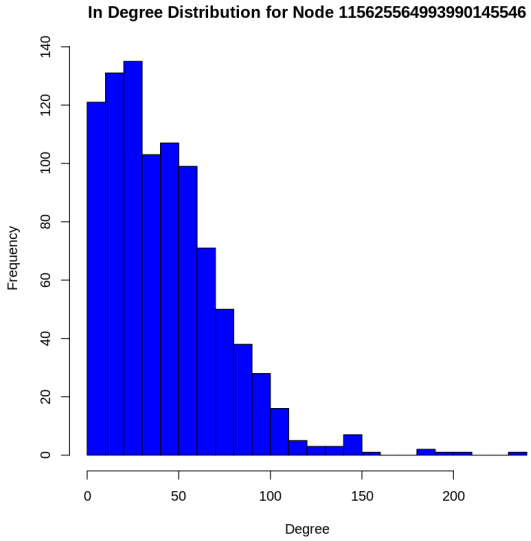
```
Node ID: 109327480479767108490
In-degree:
Mean     =    14.06202
Variance =    274.8241
Out-degree:
Mean     =    14.06202
Variance =    4026.402

Node ID: 115625564993990145546
In-degree:
Mean     =    43.63961
Variance =    1206.213
Out-degree:
Mean     =    43.63961
Variance =    8701.524

Node ID: 101373961279443806744
In-degree:
Mean     =    298.1182
Variance =    87059.55
Out-degree:
Mean     =    298.1182
Variance =    163623.3
```

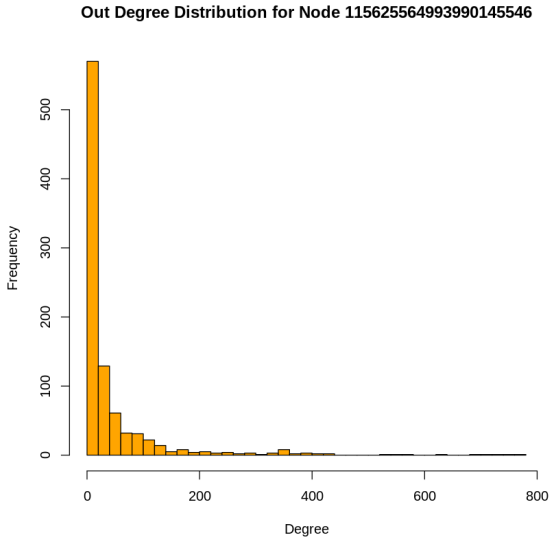
The figures here illustrate the central tendencies and the spreads of the distributions across the network.

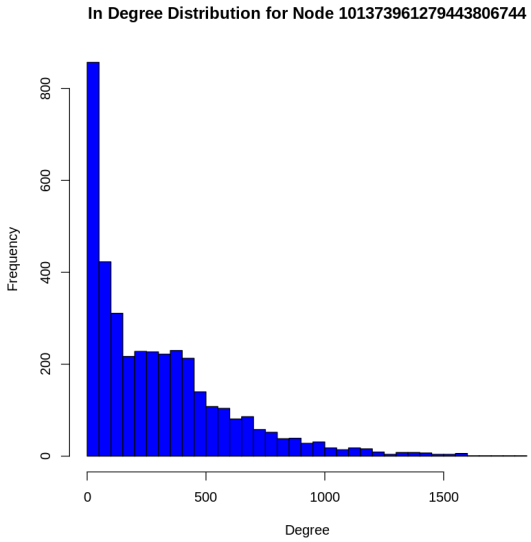
We have plotted 6 histograms to depict the in degrees and the out degrees of the network.

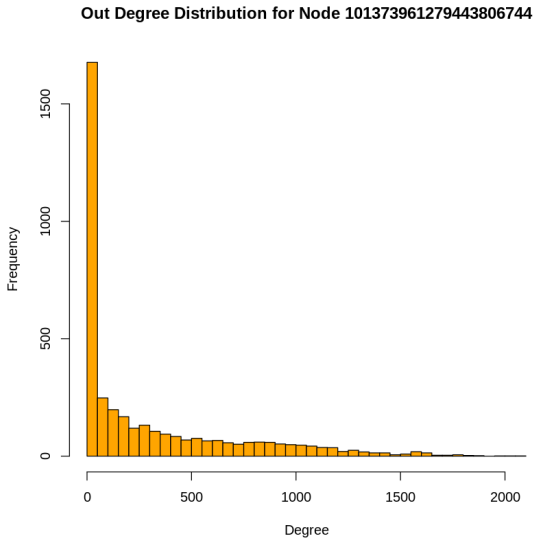

In Degree Distribution for Node 109327480479767108490
Frequency
0
100
200
300
0
20
40
60
80
Degree


Out Degree Distribution for Node 109327480479767108490
Frequency
0
100
200
300
400
500
600
700
0
200
400
600
800
Degree


In Degree Distribution for Node 115625564993990145546
Frequency
0
20
40
60
80
100
120
140
0
50
100
150
200
Degree


Out Degree Distribution for Node 115625564993990145546
Frequency
0
100
200
300
400
500
0
200
400
600
800
Degree


In Degree Distribution for Node 101373961279443806744
Frequency
0
200
400
600
800
0
500
1000
1500
Degree


Out Degree Distribution for Node 101373961279443806744
Frequency
0
500
1000
1500
0
500
1000
1500
2000
Degree

1. Overall Similarity in Distributions:

The in-degree and out-degree distributions for each node ID across the three personal networks exhibit notable similarities. Both the in-degree and the out-degree follow the Power-law behaviour. At the same time based on the central tendencies we observe that they have a very close values. We also see that the last node index in the graphs tends to have very high out-degree values and approximate zero in-degree values. This phenomenon arises because each node is following all other nodes in the network, resulting in a high out-degree but minimal in-degree for these nodes.

2. Differences Across Networks:

While the distributions may appear similar at a high level, closer examination reveals distinct characteristics for each network:

- **Network Complexity:** The third network demonstrates the highest in-degree and out-degree values overall, indicating the strongest connections between nodes. Conversely, the first network exhibits lower in-degree and out-degree values compared to the other two networks, suggesting simpler connectivity patterns.
- **Strength of Connections:** The second network displays stronger connections between nodes compared to the first network, as evidenced by higher in-degree and out-degree values. Moreover, the third network stands out for its particularly strong connections, indicating a higher level of complexity and interconnectivity.

3. Variation in In-Degree and Out-Degree Distributions:

- **Out-Degree Distribution**: All three networks exhibit similar out-degree distributions, following a power-law distribution. However, slight variations exist, with the out-degree distribution of the first network skewed slightly more to the right.
- **In-Degree Distribution**: While the in-degree distributions also demonstrate power-law behavior, there are noticeable differences among the networks. For instance, the in-degree distribution of the second network appears roughly linear, indicating a distinct connectivity pattern compared to the other networks.
- **Interpreting Network Characteristics**: The differences in in-degree and out-degree distributions offer insights into network characteristics. For example, the network with node 115625564993990145546 forms communities with strong internal connections and sparse inter-community connections, whereas the network with node 101373961279443806744 lacks clear community structures and exhibits low modularity, as indicated by the slow roll-off in the out-degree distribution.

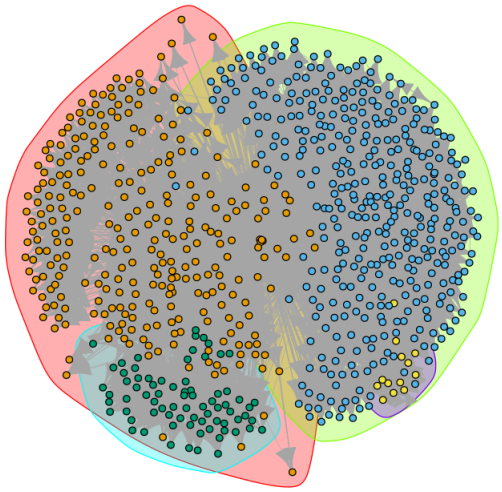
In summary, while the in-degree and out-degree distributions demonstrate some similarities across the three personal networks, there are notable differences in network complexity, strength of connections, and distribution characteristics. These differences reflect unique structural and connectivity patterns within each network, highlighting the diverse nature of personal networks and their underlying dynamics.

QUESTION 20:

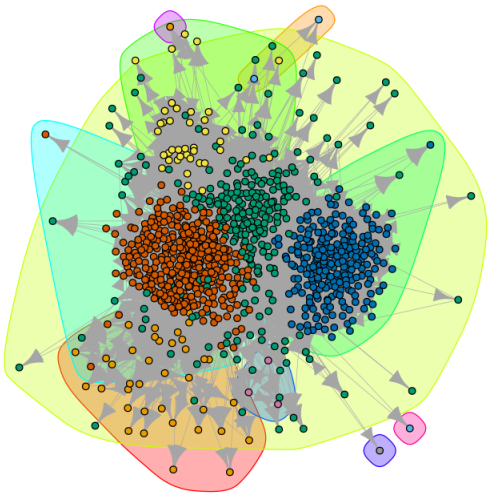
For the 3 personal networks picked in Question 19, extract the community structure of each personal network using Walktrap community detection algorithm. Report the modularity scores and plot the communities using colors. Are the modularity scores similar? In this question, you should have 3 plots.

The following figures show the community structure of each personal network extracted using the Walktrap community detection algorithm. Each communities are plotted using colors:

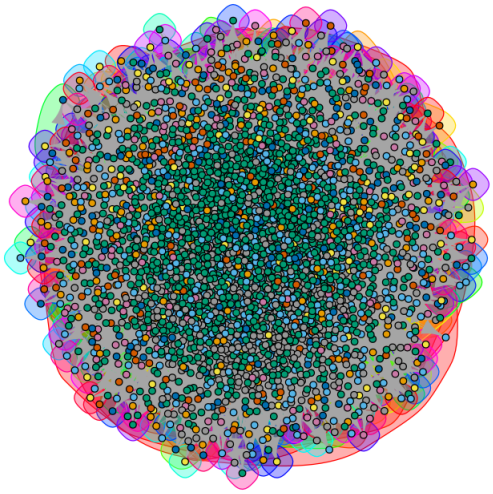
Walktrap Community Structure for Node = 109327480479767108490
Modularity Score: 0.25276535939251



Walktrap Community Structure for Node = 115625564993990145546
Modularity Score: 0.319472554647349



Community Structure, GPN, Node = 101373961279443806744



We also recorded the Modularity score for each of the Node IDs:

Node ID	Modularity Score (rounded to 5 decimal places)
109327480479767108490	0.25278
115625564993990145546	0.31947
101373961279443806744	0.19109

The modularity scores and community structures for the three nodes provide insights into their network characteristics and connectivity patterns.

Node 101373961279443806744 has the lowest modularity score of 0.19109, indicating that the network is less capable of being divided into densely packed modules with strong connectedness and sparse interconnections among communities. In the above figure most nodes in this network are grouped in a single chunk in the center, suggesting weak community structures and extensive interconnectivity. This network also exhibits the highest number of edges and nodes among all networks, which may contribute to the loss of global sparsity among communities, weakening the community structures.

We also observed that the personalized network for node 101373961279443806744 has the highest number of edges m and nodes among all networks. Intuitively, a higher value of indicates that an incoming node is connected to a larger number of older nodes. While this should result in strong intra-community connectedness, the global sparsity among different communities is lost due to the connectedness requirement brought on by high values of m, resulting in edges being formed among otherwise distinct clusters and hence weakening the community structures. Mathematically, it is given as:

$$Q(P) = \sum \frac{1}{2m} \sum_{i,j \in C_i} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

If the number of edges m in the network (not to be confused with the actual m we are talking about, which is the number of old nodes the incoming node connects with) increases, then Q(P) drops. As a result, less clusters are formed in the overall graph.

On the other hand, Node 115625564993990145546 boasts the highest modularity score of 0.31947. Its figure illustrates dense communities or clusters with sparse interconnectivity among them. This high modularity score signifies stronger connections within communities and weaker connections between communities.

For Node 109327480479767108490, the modularity score is 0.25278, indicating intermediate community structure. While not as strong as Node 115625564993990145546, it exhibits stronger community structures compared to Node 101373961279443806744.

In summary, the second node demonstrates the highest modularity score, suggesting well-defined community clustering with strong intra-community connections and weak inter-community connections. Conversely, the third node exhibits complex connections, resulting in strong inter-community connections and a lower modularity score.

One thing to note is that, from the previous graphs that the three personal networks have quite similar indegree and outdegree distribution for each node ID (the last node index in the graphs) of the personal network. The node ID has a very high out-degree and an approximate zero in-degree. This is because for each node ID of that personal network, the node is following all the other nodes in the graph, so the outdegree is equal to the number of nodes in that network. The modularity scores and community structures highlight the varying degrees of community organization and connectivity patterns across the three nodes, providing valuable insights into their network dynamics and structures.

QUESTION 21:

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

Homogeneity (h) measures the extent to which the communities (K) are able to capture the diversity of circles (C). It quantifies how well the communities represent the circles. A high homogeneity score indicates that the communities effectively capture the diversity of circles, meaning that individuals within the same community tend to belong to similar circles. Conversely, a low homogeneity score suggests that the communities poorly represent the diversity of circles, indicating that individuals within the same community may belong to a wide range of circles. In other words, homogeneity evaluates the consistency of circle memberships within communities and measures the purity of the community structure. When the community is composed of nodes coming from the same circle, the homogeneity reaches a higher score.

Completeness (c) measures the extent to which the circles (C) are able to capture the diversity of communities (K). It quantifies how well the circles represent the communities. A high completeness score indicates that the circles effectively capture the diversity of communities, meaning that individuals within the same circle tend to belong to similar communities. On the other hand, a low completeness score suggests that the circles poorly represent the diversity of communities, indicating that individuals within the same circle may belong to a wide range of communities. In essence, completeness evaluates the consistency of community memberships within circles and measures the purity of the circle. When the circle assigns nodes to the same community, the completeness scores become higher.

Mathematically:

Homogeneity: A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class or circle. Mathematically, homogeneity h is defined in terms of conditional entropy of labels or circles C given cluster assignments or circles K and is independent of absolute value of ground truth labels.

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

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}\cdot\log\left(\frac{n_{c,k}}{n_k}\right)$$

$$H(X) = -\sum_{h=1}^{|H|}\frac{n_h}{n}\cdot\log\left(\frac{n_h}{n}\right)$$

$$n_{c,k} = n_k \cap n_c$$

H(X) denotes the entropy of partition X, where X denotes non-overlapping groups of sample points. The information entropy is maximized when the sizes of the partitions are equal and minimized when some group within the partition takes up all the data points. The homogeneity score is maximized when each cluster Ki contains samples only from Ci , i.e. . $H(C|K) = 0$

Completeness: A clustering result satisfies completeness if all the data points that are members of a given circle are elements of the same community. Mathematically, completeness is defined in terms of conditional entropy of clusters K given ground truth labels C.

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

Completeness if maximized when each ground truth class Ci is part of some cluster Ki . In the ideal case (c = 1), a single cluster should encompass all members of a circle.

QUESTION 22:

Compute the h and c values for the community structures of the 3 personal networks (same nodes as Question 19). Interpret the values and provide a detailed explanation. Are there negative values? Why?

We get the following h (homogeneity) and c (completeness) values:

Node ID	Homogeneity h	Completeness c
109327480479767108490	**0.86400411**	**0.34569232**
115625564993990145546	**0.44294454**	**-3.37571773**
101373961279443806744	**0.00183925**	**-1.60926257**

```
Node ID: 109327480479767108490
[1] "Entropy H(C) = 0.45634767"
[1] "Entropy H(K) = 0.43733067"
[1] "Conditional Entropy H(C|K) = 0.06206141"
[1] "Conditional Entropy H(K|C) = 0.28614882"
[1] "Homogeneity h = 0.86400411"
[1] "Completeness c = 0.34569232"
[1] "V-measure V = 0.49380915"


Node ID: 115625564993990145546
[1] "Entropy H(C) = 3.67636649"
[1] "Entropy H(K) = 0.45911464"
[1] "Conditional Entropy H(C|K) = 2.04794004"
[1] "Conditional Entropy H(K|C) = 2.00895608"
[1] "Homogeneity h = 0.44294454"
[1] "Completeness c = -3.37571773"
[1] "V-measure V = 1.01968725"


Node ID: 101373961279443806744
[1] "Entropy H(C) = 0.16690804"
[1] "Entropy H(K) = 0.23234634"
[1] "Conditional Entropy H(C|K) = 0.16660105"
[1] "Conditional Entropy H(K|C) = 0.60625261"
[1] "Homogeneity h = 0.00183925"
[1] "Completeness c = -1.60926257"
[1] "V-measure V = 0.00368271"
```

From the above values we make the following observations:

1. The network associated with node ID 109327480479767108490 exhibits a notably high homogeneity score. This suggests that within each community, the majority of users are derived from the same circle. Although the completeness score is lower, it remains positive, implying that some users from one circle may have been placed in different communities. This is visible in Figure from Q20, where some orange dots are assigned to communities other than red. Despite this, the completeness score is comparatively higher than the other networks, indicating a relatively well-segregated arrangement of members into distinct communities with minimal overlap.

2. In contrast, the network linked to node ID 115625564993990145546 displays a significantly lower homogeneity score, almost half of that observed in the preceding network. This suggests a greater number of users from various circles have been incorrectly grouped into a single community. This misclassification is evident from the community structure plot in Figure from Question 20. Additionally, the completeness score is low and negative, indicating potential issues such as some circles remaining unassigned to any community, resulting in communities with few circles and a considerable mismatch between the number of circles and communities.

3. Finally, for the network associated with node ID 101373961279443806744, the homogeneity score is the lowest among the three networks, indicating a prevalent misclassification of users from different circles into a single community. This aligns with the low modularity scores, indicating difficulties in forming dense communities with sparse inter-community connections, suggesting insufficient community separation. While the completeness score remains negative, it surpasses that of the network with node 115625564993990145546, pointing to a higher degree of misclassification, likely stemming from similar issues discussed previously.

To answer the latter part of the question: yes, we observed negative values for the second and the third node. A negative completeness score can arise due to several factors:

1. Misclassification of Nodes: If nodes from the same circle are scattered across multiple communities or if nodes from different circles are grouped together within a single community, it can lead to a negative completeness score. This indicates that the clustering algorithm has failed to accurately capture the community structure of the network.

2. Sparse Community Structure: In networks where the community structure is not well-defined or where communities are sparse and poorly separated, the algorithm may struggle to assign nodes to appropriate communities. This can result in incomplete community assignments and, consequently, a negative completeness score.

3. Imbalanced Community Sizes: When the number of communities is much smaller than the number of circles, or when some communities contain significantly fewer nodes than others, it can lead to incomplete coverage of circles within communities. This imbalance can contribute to a negative completeness score.

4. Algorithmic Limitations: Certain community detection algorithms may have inherent limitations that prevent them from accurately capturing the true community structure of a network. For example, algorithms that rely on local information or heuristic approaches may produce suboptimal results in networks with complex or overlapping communities.

5. Data Quality Issues: Inaccurate or incomplete data, such as missing or erroneous node attributes, can adversely affect the performance of community detection algorithms and result in incomplete or incorrect community assignments.

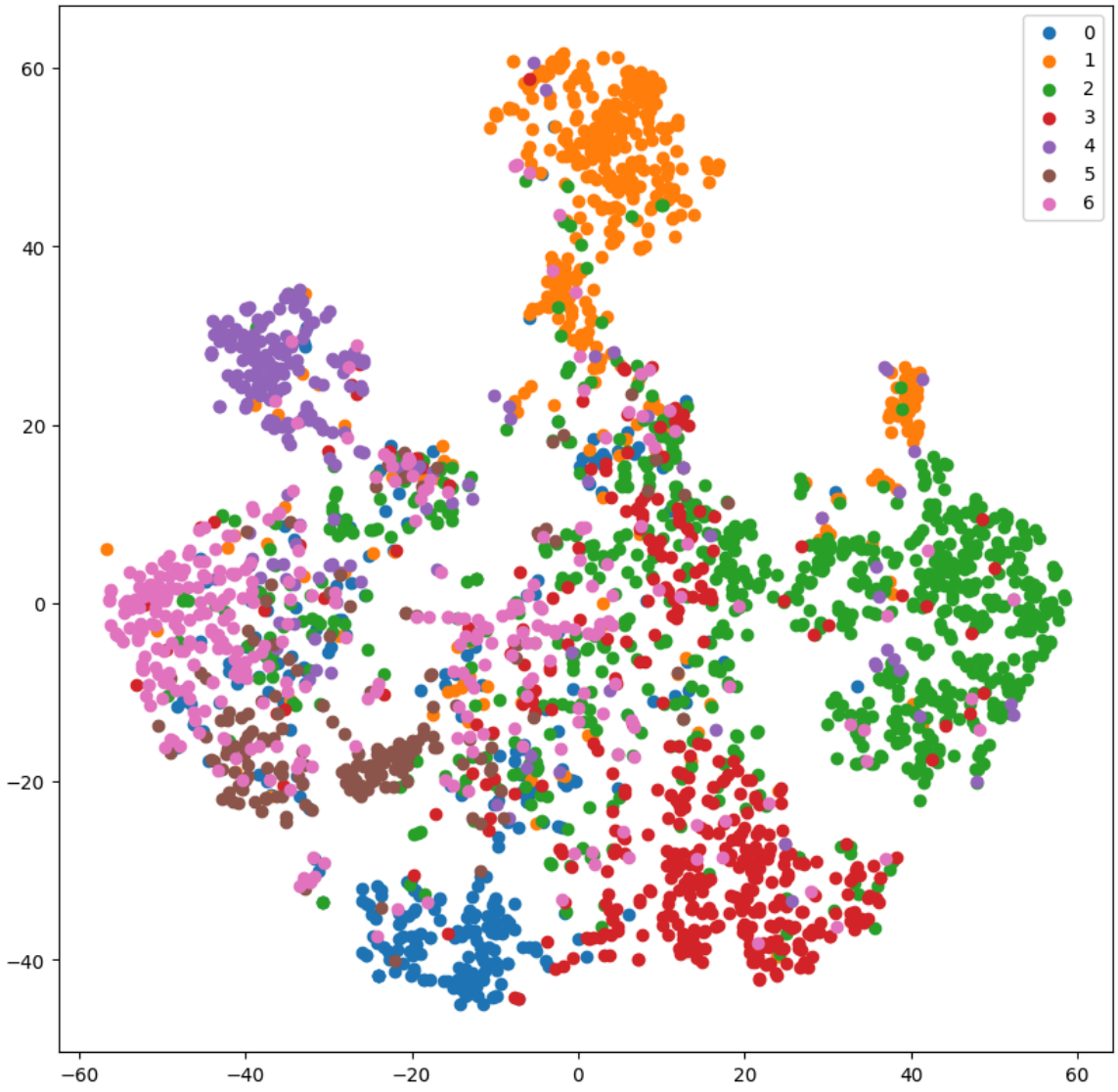
Overall, a negative completeness score indicates a mismatch between the true community structure of the network and the communities identified by the clustering algorithm, highlighting the need for further investigation and potentially more robust algorithmic approaches.

CORA DATASET

QUESTION 23: Idea 1

Use Graph Convolutional Networks [1]. What hyperparameters do you choose to get the optimal performance? How many layers did you choose?

With Hyperparameter Grid-Search:-



We performed grid search to find the best hyperparameters for a Graph Convolutional Network (GCN) applied to the Cora dataset. The best combination of hyperparameters indicates that a smaller number of channels (16), moderate dropout rate (0.3), moderate L2 regularization (0.01), and a moderate learning rate (0.01) result in the best validation accuracy.

- Best validation accuracy: 0.7979999780654907
- Best hyperparameters: {'channels': 16, 'dropout': 0.3, 'l2_reg': 0.01, 'learning_rate': 0.01}

In our model we use two layers of GCN with ReLU activation for the first layer and softmax activation for the final layer.

QUESTION 24: Idea 2

Extract structure-based node features using Node2Vec [2]. Briefly describe how Node2Vec finds node features. Choose your desired classifier (one of SVM, Neural Network, or Random Forest) and classify the documents using only Node2Vec (graph structure) features. Now classify the documents using only the 1433-dimensional text features. Which one outperforms? Why do you think this is the case? Combine the Node2Vec and text features and train your classifier on the combined features. What is the best classification accuracy you get (in terms of the percentage of test documents correctly classified)?

Briefly describe how Node2Vec finds node features:-

Node2Vec is an algorithm for feature learning on graphs, particularly useful for capturing structural information in the form of node embeddings. The algorithm is based on the idea of representing nodes as low-dimensional vectors while preserving the graph's structural properties. Here's a brief overview of how Node2Vec works:

- Random Walks: Node2Vec starts by generating random walks on the graph. A random walk is a sequence of nodes traversed by starting from a given node and repeatedly choosing a neighboring node to visit next. These random walks capture local neighborhood information around each node.

- Feature Learning: After generating random walks, Node2Vec employs a skip-gram model, similar to Word2Vec, to learn embeddings for each node in the graph. The skip-gram model predicts the context nodes (nodes visited in the random walk) given a target node (starting node of the random walk). By doing so, it learns node embeddings that encode structural similarities between nodes.

- Optimization: Node2Vec optimizes the embeddings to maximize the likelihood of observing the context nodes given the target node across all random walks.

- Embedding Extraction: Once the optimization process is complete, Node2Vec extracts the learned embeddings for each node in the graph. These embeddings represent low-dimensional vectors that encode structural information about the nodes and their relationships in the graph.

Which one outperforms?

- **Results using only Node2Vec**

    - **Classifier Training - SVC**

```
Mean Accuracy: 0.8428571428571429
Test Accuracy: 0.7526652452025586
```

    - **Classifier Training - Neural Network**

```
Mean Accuracy: 0.8285714285714285
Test Accuracy: 0.7164179104477612
```

- **Classifier Training - Random Forest**

```
Mean Accuracy: 0.7642857142857143
Test Accuracy: 0.7168443496801705
```

- **Results using only 1433-dimensional text features**
  - **Classifier Training - SVC**

```
Mean Accuracy: 0.06428571428571428
Test Accuracy: 0.12281449893390192
```

  - **Classifier Training - Neural Network**

```
Mean Accuracy: 0.11428571428571428
Test Accuracy: 0.13432835820895522
```

  - **Classifier Training - Random Forest**

```
Mean Accuracy: 0.1357142857142857
Test Accuracy: 0.13603411513859276
```

- **Results using Combination**
  - **Classifier Training - SVC**

```
SVM Accuracy (Combined Features): 0.733049040511727
```

  - **Classifier Training - Neural Network**

```
Neural Network Accuracy (Combined Features): 0.6396588486140725
```

  - **Classifier Training - Random Forest**

```
Random Forest Accuracy (Combined Features): 0.679317697228145
```

The results confirm that Node2Vec effectively captures structural information from the graph, leading to superior performance compared to using text features alone. The algorithm's ability to learn node embeddings that encode relationships between nodes in the graph contributes significantly to its success in classifying the documents.

What is the best classification accuracy you get (in terms of the percentage of test documents correctly classified)?

The best classification accuracy achieved using only Node2Vec features is approximately 75.26%. This indicates that the structural information captured by Node2Vec embeddings is effective in classifying the documents, outperforming the classification based solely on text features.

QUESTION 25: Idea 3

We can find the personalized PageRank of each document in seven different runs, one per class. In each run, select one of the classes and take the 20 seed documents of that class. Then, perform a random walk with the following customized properties: (a) teleportation takes the random walker to one of the seed documents of that class (with a uniform probability of 1/20 per seed document). Vary the teleportation probability in {0, 0.1, 0.2}. (b) the probability of transitioning to neighbors is not uniform among the neighbors. Rather, it is proportional to the cosine similarity between the text features of the current node and the next neighboring node. Particularly, assume we are currently visiting a document x0 which has neighbors x1,x2,x3.

Then the probability of transitioning to each neighbor is:

$$p_i = \frac{\exp(x_0 \cdot x_i)}{\exp(x_0 \cdot x_1) + \exp(x_0 \cdot x_2) + \exp(x_0 \cdot x_3)}; \text{ for } i = 1, 2, 3.$$

Repeat part b for every teleportation probability in part a. Run the PageRank only on the GCC. for each seed node, do 1000 random walks. Maintain a class-wise visited frequency count for every unlabeled node. The predicted class for that unlabeled node is the class which lead to maximum visits to that node. Report accuracy and f1 scores. For example if node 'n' was visited by 7 random walks from class A, 6 random walks from class B... 1 random walk from class G, then the predicted label of node of 'n' is class A.

All the experiments are run for 1000 random samples and for 100 walk steps.

Part (a):

1. **For Transition Probability = 0**

```
unvisited =  0
              precision    recall  f1-score   support

           0       0.33      0.53      0.41       285
           1       0.50      0.56      0.53       406
           2       0.53      0.30      0.38       726
           3       0.60      0.52      0.56       379
           4       0.09      0.12      0.11       214
           5       0.10      0.23      0.14       131
           6       0.27      0.22      0.24       344

    accuracy                           0.37      2485
   macro avg       0.35      0.35      0.34      2485
weighted avg       0.42      0.37      0.38      2485

0.37183098591549296
```

- **Accuracy Score: 0.3718**
- **F1 Score: 0.338571**

2. **For Transition Probability = 0.1**

```
unvisited =  7
              precision    recall  f1-score   support

           0       0.70      0.78      0.74       285
           1       0.83      0.93      0.88       406
           2       0.88      0.63      0.73       726
           3       0.78      0.84      0.81       379
           4       0.61      0.79      0.68       214
           5       0.48      0.91      0.63       131
           6       0.78      0.58      0.66       344

    accuracy                           0.75      2485
   macro avg       0.72      0.78      0.73      2485
weighted avg       0.78      0.75      0.75      2485

0.7492957746478873
```

- **Accuracy Score: 0.7493**
- **F1 Score: 0.7328**

3. **For Transition Probability = 0.2**

```
unvisited =  57
              precision    recall  f1-score   support

           0       0.60      0.78      0.68       285
           1       0.86      0.89      0.87       406
           2       0.84      0.60      0.70       726
           3       0.75      0.76      0.76       379
           4       0.67      0.85      0.74       214
           5       0.46      0.87      0.60       131
           6       0.68      0.54      0.60       344

    accuracy                           0.72      2485
   macro avg       0.69      0.76      0.71      2485
weighted avg       0.75      0.72      0.72      2485

0.7203219315895373
```

- **Accuracy Score: 0.7203**
- **F1 Score: 0.70714**

Part (b):

1. **For Transition Probability = 0**

```
unvisited =  0
              precision    recall  f1-score   support

           0       0.34      0.53      0.42       285
           1       0.53      0.64      0.58       406
           2       0.54      0.28      0.37       726
           3       0.57      0.56      0.57       379
           4       0.14      0.18      0.16       214
           5       0.10      0.22      0.14       131
           6       0.32      0.26      0.28       344

    accuracy                           0.39      2485
   macro avg       0.37      0.38      0.36      2485
weighted avg       0.44      0.39      0.40      2485

0.39476861167002014
```

- **Accuracy Score: 0.3948**
- **F1 Score: 0.36**

2. **For Transition Probability = 0.1**

```
unvisited =  15
              precision    recall  f1-score   support

           0       0.72      0.79      0.76       285
           1       0.85      0.93      0.89       406
           2       0.88      0.63      0.73       726
           3       0.76      0.85      0.80       379
           4       0.68      0.83      0.75       214
           5       0.52      0.92      0.67       131
           6       0.66      0.56      0.61       344

    accuracy                           0.76      2485
   macro avg       0.73      0.79      0.74      2485
weighted avg       0.77      0.76      0.75      2485

0.7553319919517103
```

- **Accuracy Score: 0.7553**
- **F1 Score: 0.7442**

3. For Transition Probability = 0.2

```
unvisited =  61
              precision    recall  f1-score   support

           0       0.60      0.74      0.66       285
           1       0.84      0.90      0.87       406
           2       0.86      0.56      0.68       726
           3       0.78      0.80      0.79       379
           4       0.67      0.83      0.74       214
           5       0.46      0.89      0.60       131
           6       0.64      0.58      0.61       344

    accuracy                           0.72      2485
   macro avg       0.69      0.76      0.71      2485
weighted avg       0.75      0.72      0.72      2485

0.717907444668008
```

- **Accuracy Score: 0.7179**
- **F1 Score: 0.7071**