Project 1 Report

Aozhu Chen 004773895
Ruoxi Zhang 404753334
Ning Xin 704775693

In this project, we will study a social network and graphs of users' personal friendship network. We will explore community structures in the friendship network and their interpretation and applications. All datasets are available online.

Question 1

Firstly, we are going to load the data from facebook_combined.txt file. We used the is.connected() method which is already exist in the igraph library. By doing this step we can get the connectivity of our network. The result shows true. Thus, we know the network is connected.

Secondly, we used diameter() method to measure the diameter of the network. It takes two parameters, one is the network, another one is the directed equals to false. The result shows eight.

Thus, we know the diameter of the network is 8.

```
> is.connected(g)
[1] TRUE
> diameter(g, directed = FALSE)
[1] 8
```

Figure 1.1 Result of connectivity and diameter of network

Next, we use hist() and plot() to plot the histogram of degree of facebook graph and degree distribution of facebook graph respectively.

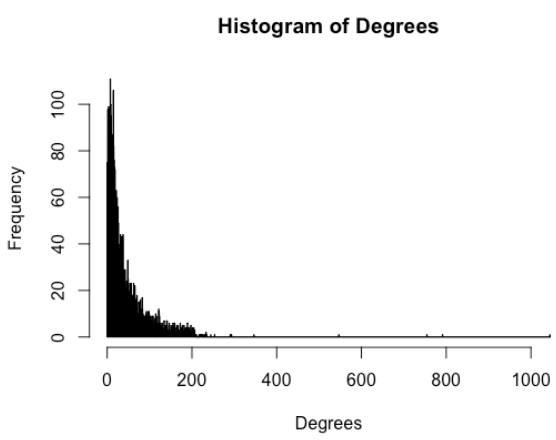


Figure 1.2 Histogram of degrees of facebook

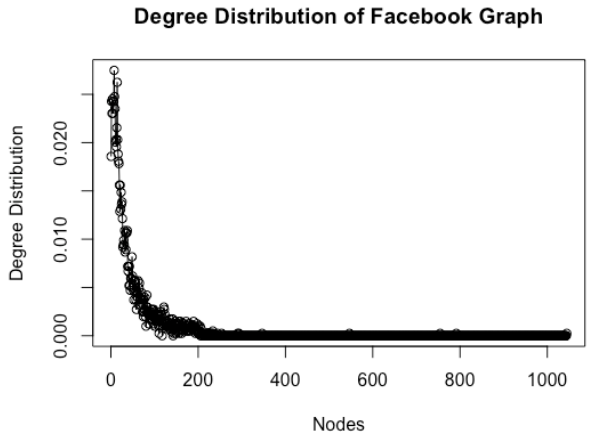


Figure 1.3 Degree distribution of facebook

Finally, we need to find out the best fitted curve. We installed the package of ggplot2 and fit.models which can be used to help us find the best fitted curve. The models we are going to use are 1/x, log(x), (1/x*a) + b*x, (a + b*log(x)), (1/x*a) + b and (exp(1)^(a + b * x)). After running our program of each model, we can get the plot as follows:

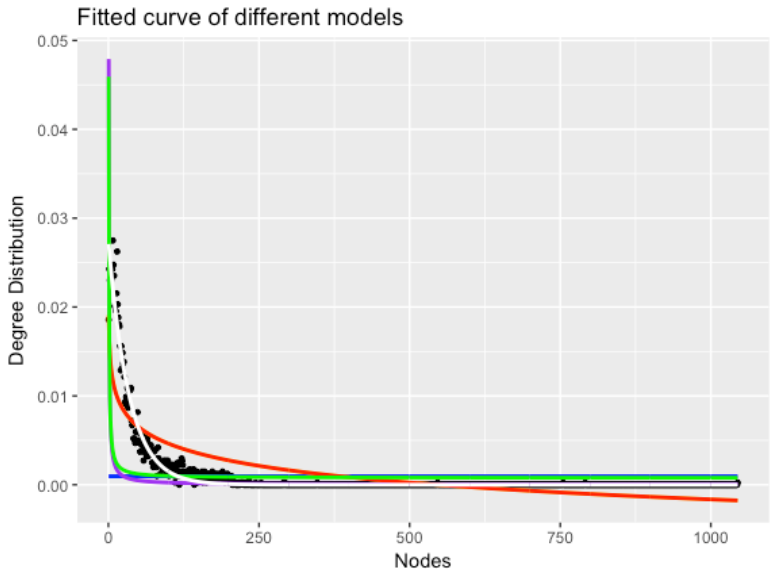


Figure 1.4 Fitted curve of different models

It is clearly to figure out that the white line is the best fitted curve for our degree distribution.

The white line curve model is (exp^(a + b * x)). The mean squared error is 7.817e-07. The average degree is 43.69101.

```
> mean(degree(g))
[1] 43.69101
>
```

Figure 1.5 Result of average degree

Question 2

We are going to use neighborhood() method which take the first node in the graph (the node whose ID is 1) and find its neighbors. Next, we used induced_subgraph() and cat() function. We will call this the personal network of node 1. We find the result of number of nodes is 348 and the number of edges is 2866.

```
> cat("Number of nodes :" , vcount(pn1))
Number of nodes : 348
> cat("Number of edges :", ecount(pn1))
Number of edges : 2866
```

Figure 2.1 Result of number of edges and nodes

The graph created for our network is as follows, it consists of node 1 which used blue to represent. The green nodes represent its neighbors. The grey line shows the edge between two nodes.

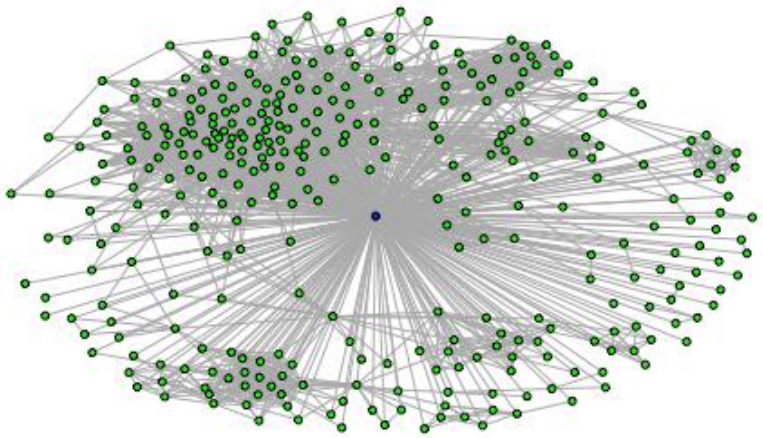


Figure 2.2 Graph of personal network

Question 3

By running our program, we get the result as follows:

```
There are 40 core nodes
```

Figure3.1 Number of core node

```
The average degree of these core nodes is 279.375
```

Figure3.2 Average degree of these core nodes

The number of core nodes we find in the network is 40.

As for the average degree of theses core nodes, it is 279.375

Next, we get the community structure of the core's personal network. The personal network of node ID-483 is as follows:

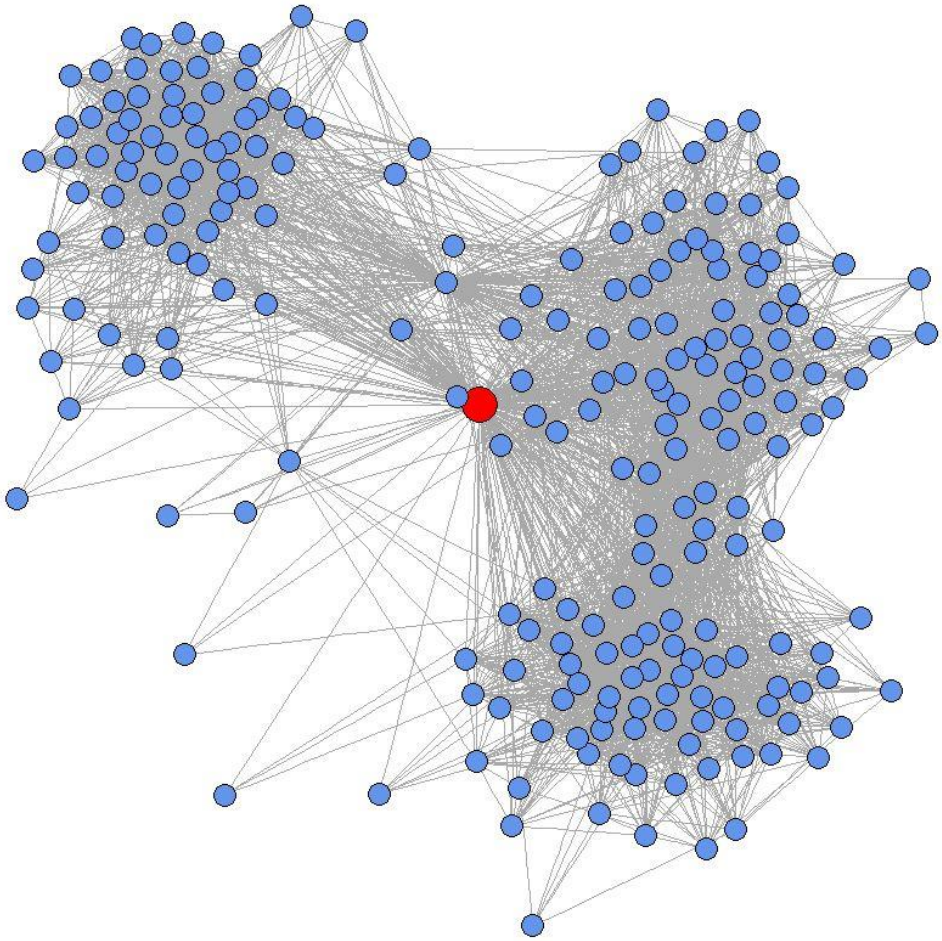


Figure3.3 Personal network of node ID-483

Fast-Greedy detection algorithm:

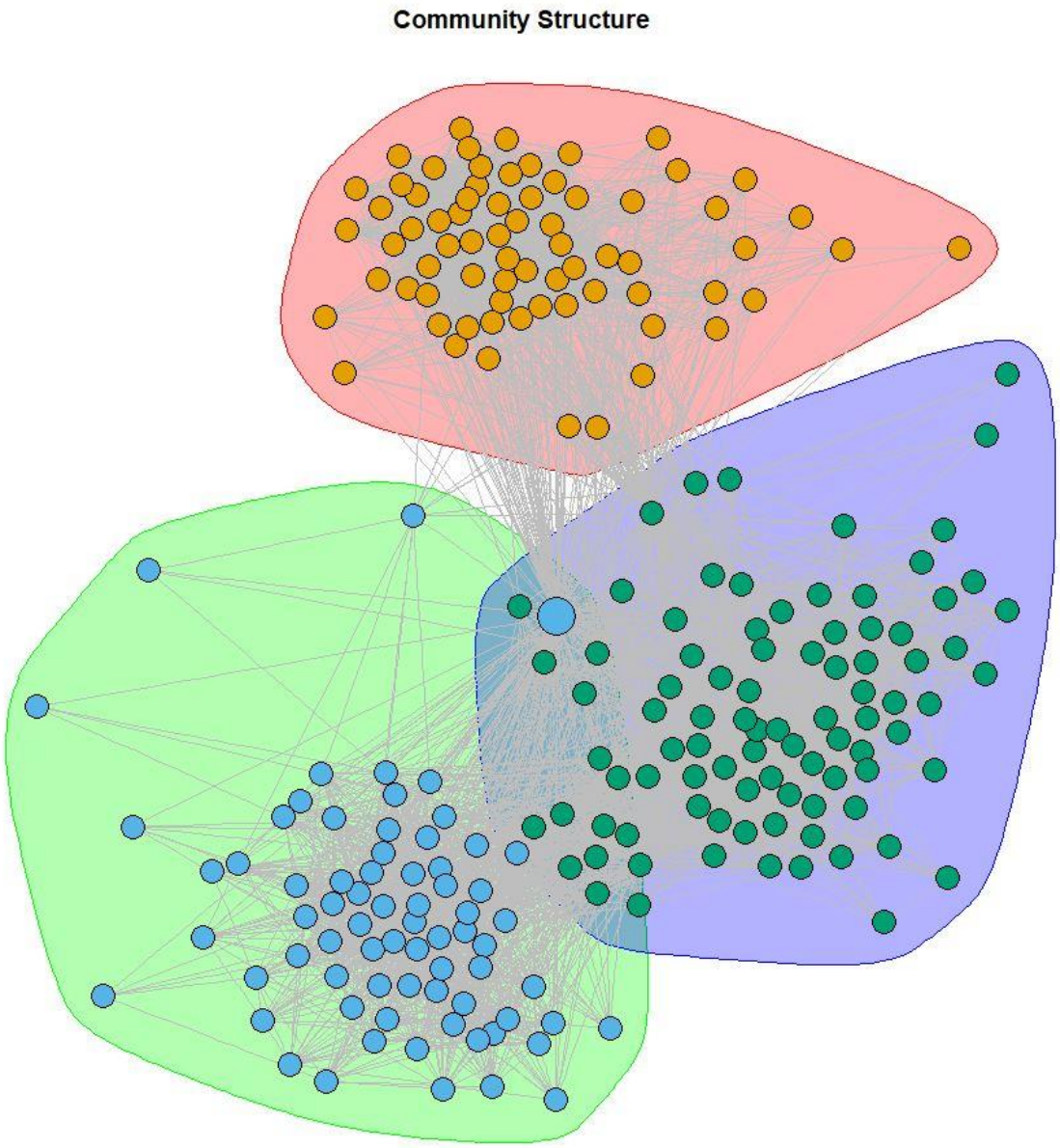


Figure3.4 Personal network of Fast Greedy Algorithm

```
Community sizes
 1  2  3
71 72 89
```

Figure3.5 Community structure of Fast Greedy Algorithm

Edge-Betweenness detection algorithm:

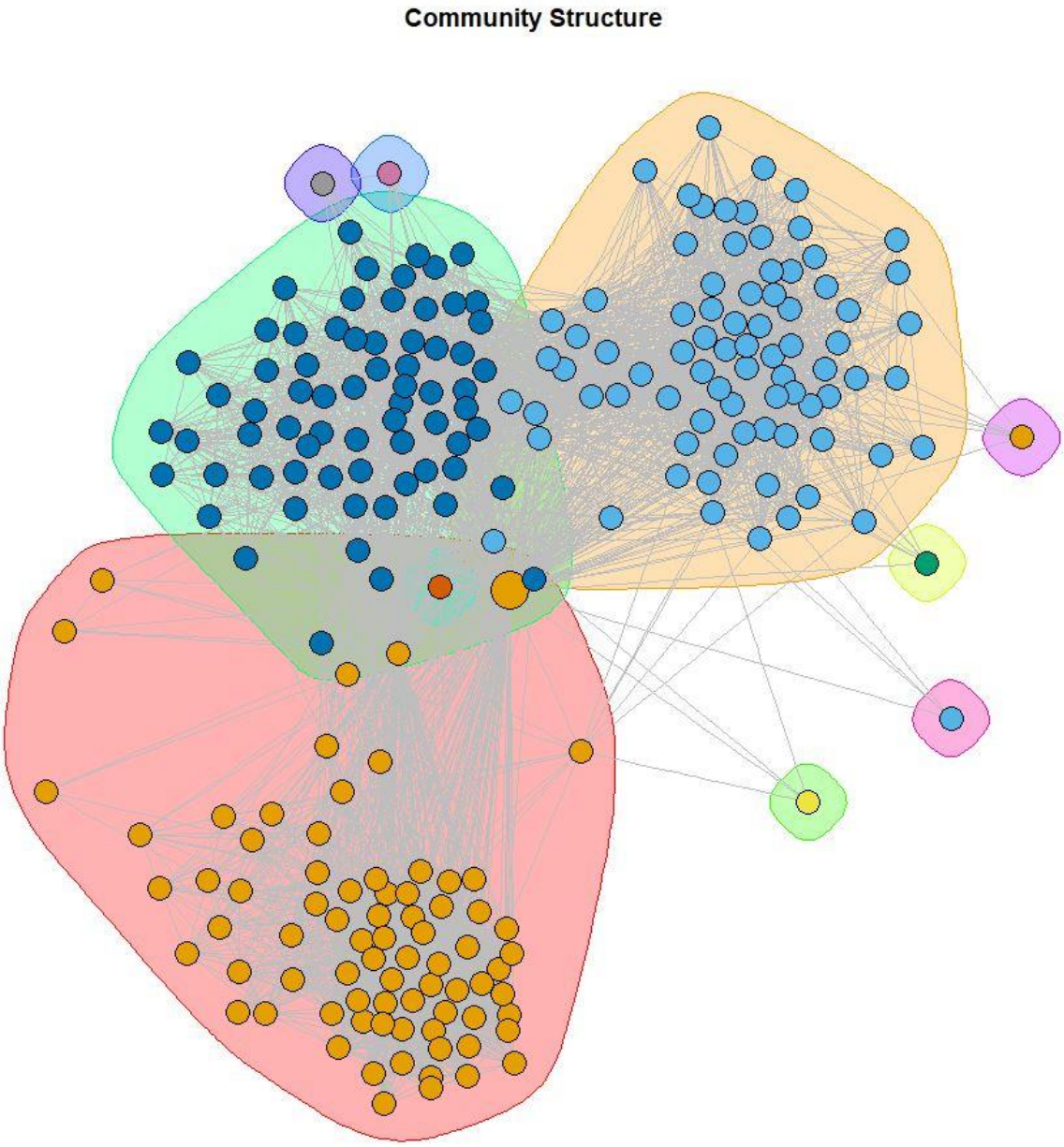


Figure3.6 Personal network of Edge-Betweenness Algorithm

```
Community sizes
 1  2  3  4  5  6  7  8  9 10
77 79  1  1 69  1  1  1  1  1
```

Figure3.7 Community structure of Edge-Betweenness Algorithm

InfoMap detection algorithm:

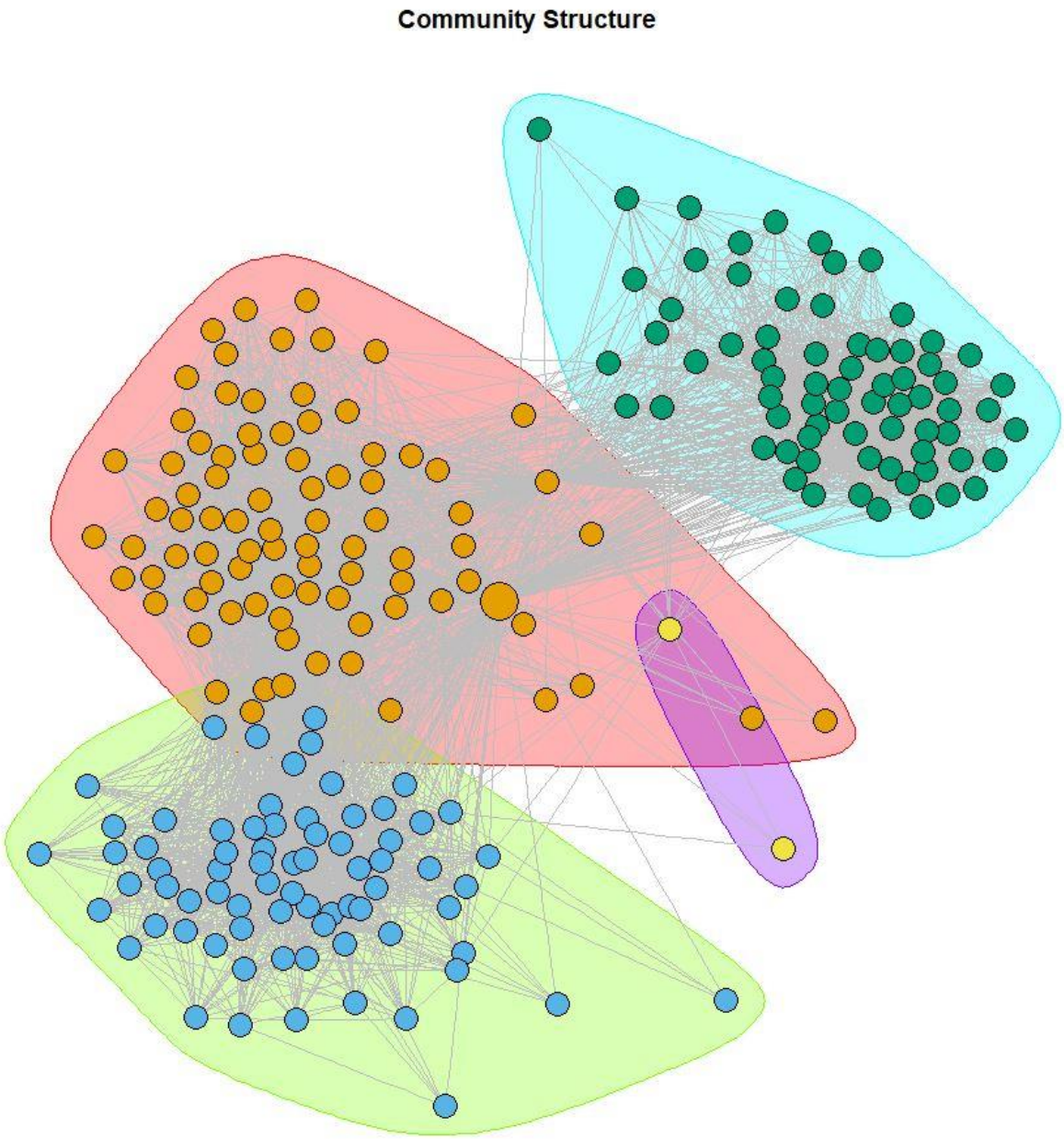


Figure3.8 Personal network of InfoMap Algorithm

```
Community sizes
 1  2  3  4
86 73 71  2
```

Figure3.9 Community structure of InfoMap Algorithm

Comparison of each detection algorithm results:

By analyzing our results, we can find that the Fast-Greedy algorithm can separate each community more equally. The Edge-Betweenness algorithm have more communities than other two algorithms. The number of communities can influence the modularity of the network. In other words, the Edge-Betweenness algorithm can separate those communities in more detail. In addition, we can find that the node ID-483 which located in the maximum number of nodes community.

Question 4

Firstly, by running our program, we get the result of personal network graph without node:

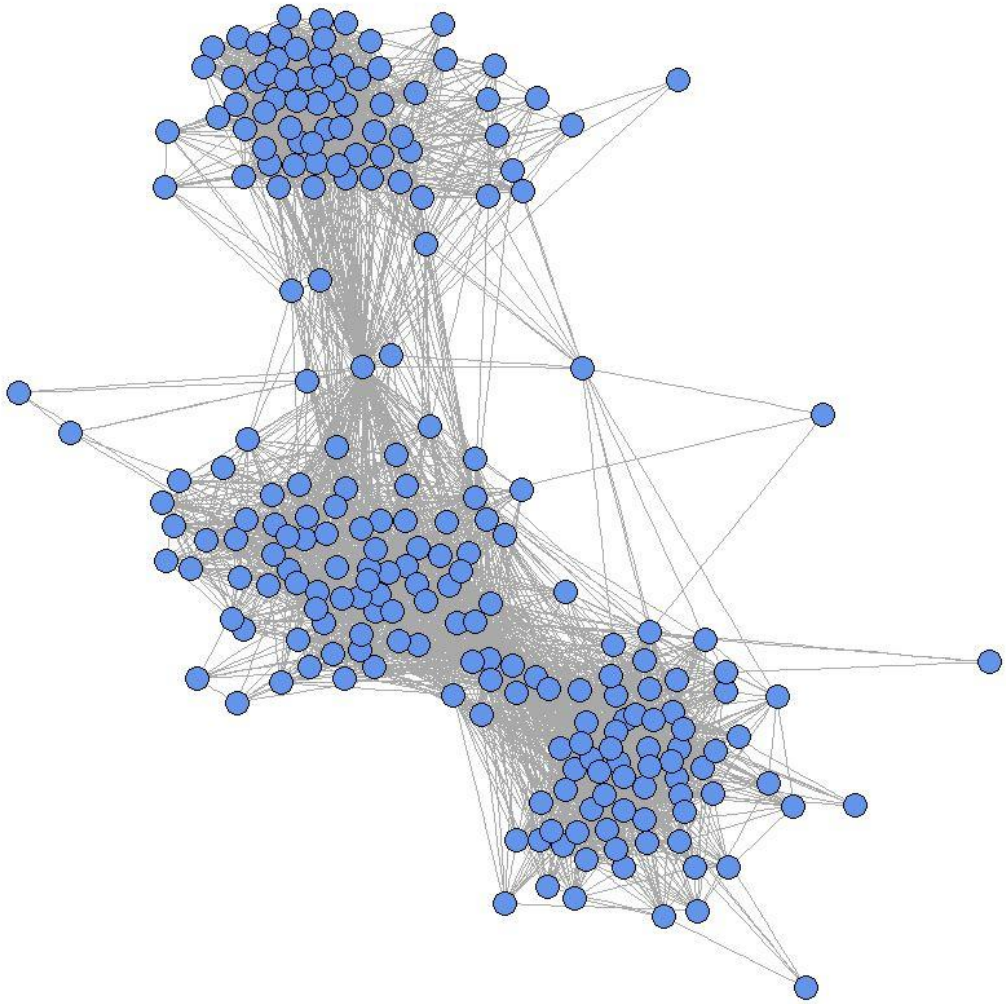


Figure4.1 Personal network without node

Fast-Greedy detection algorithm:

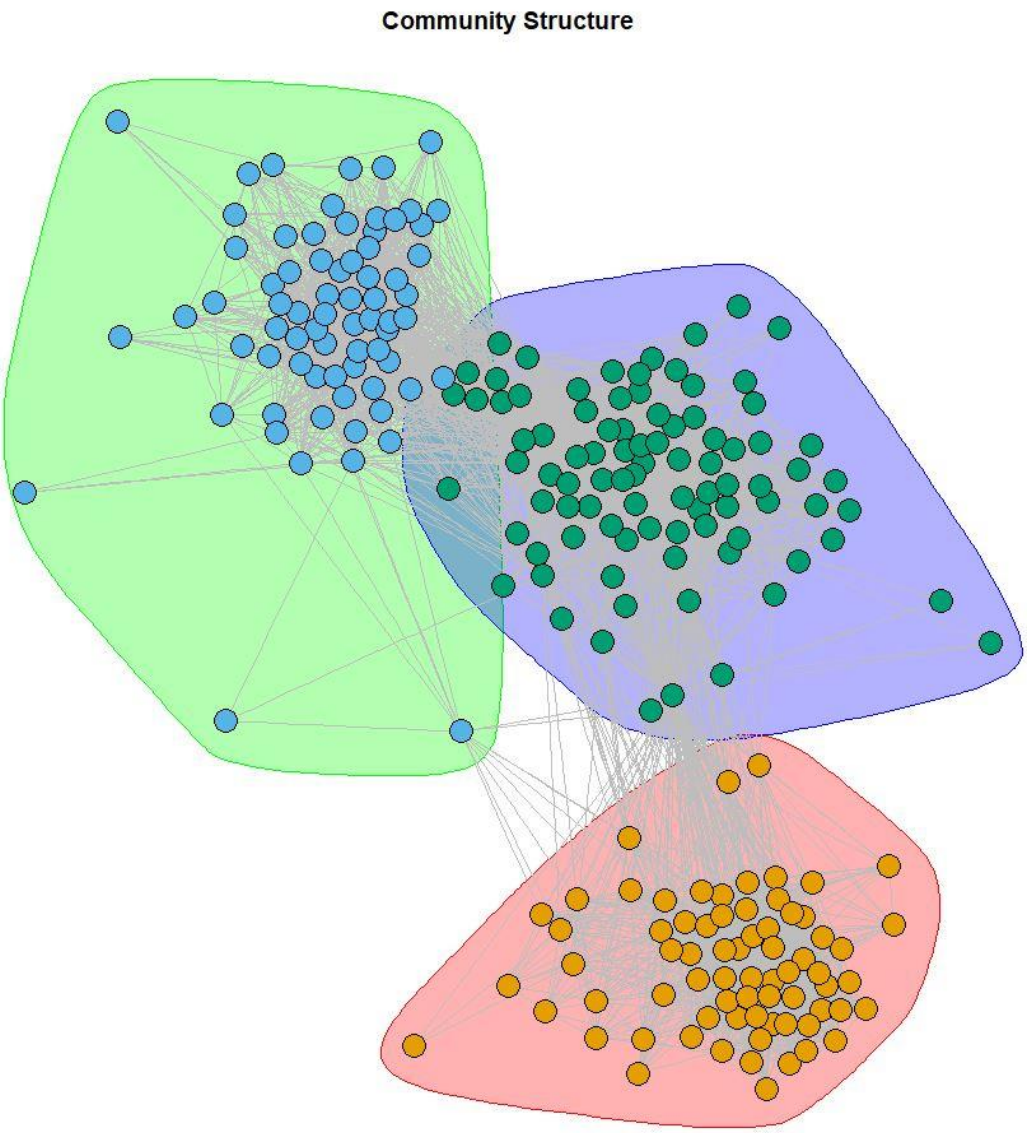


Figure4.2 Personal network without node of Fast Greedy Algorithm

```
Community sizes
 1  2  3
71 71 89
```

Figure4.3 Community structure without node of Fast Greedy Algorithm

Edge-Betweenness detection algorithm:

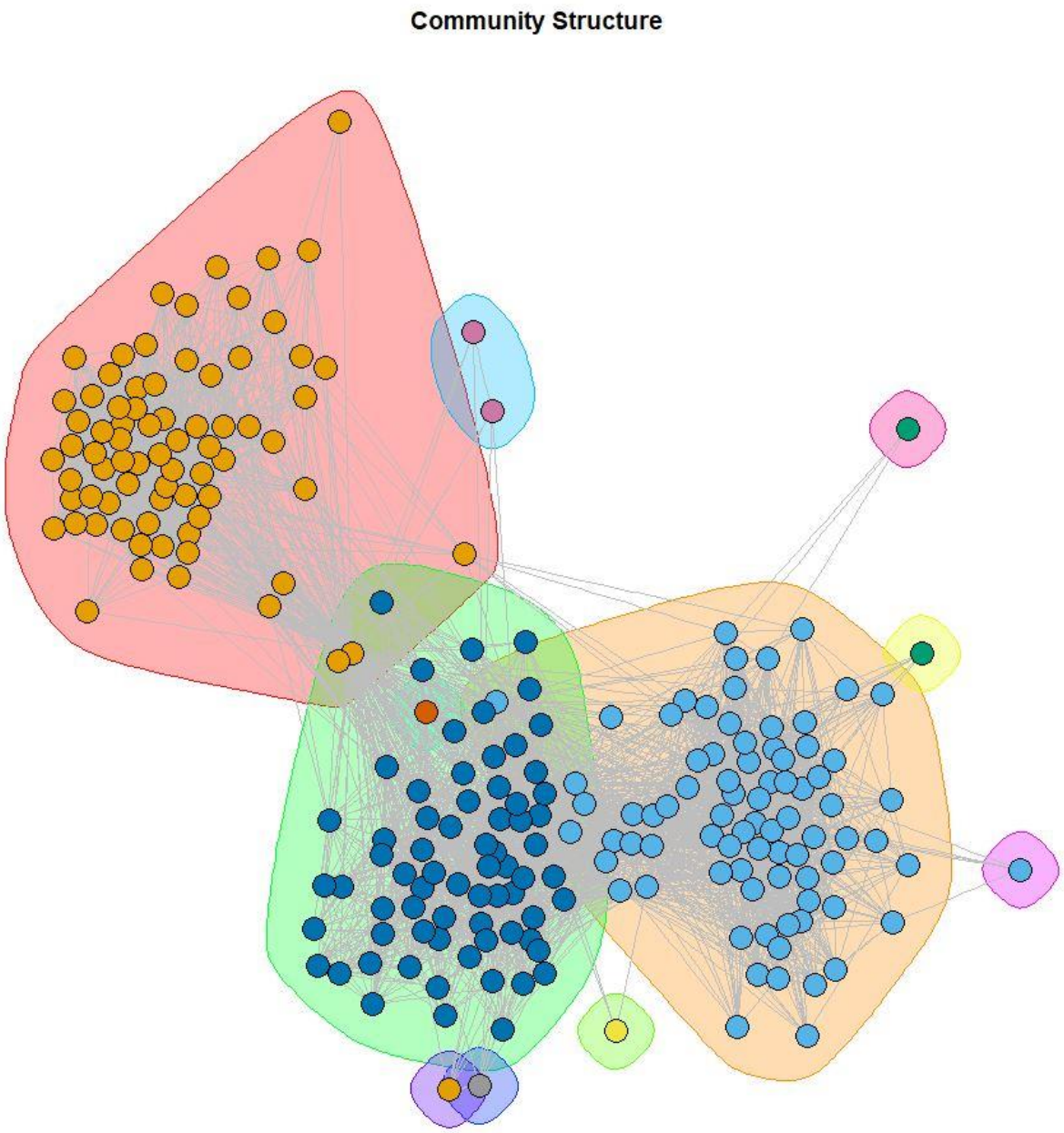


Figure4.4 Personal network without node of Edge-Betweenness Algorithm

```
Community sizes
 1  2  3  4  5  6  7  8  9 10 11
74 79  1  1 69  1  2  1  1  1  1
```

Figure4.5 Community structure without node of Edge-Betweenness Algorithm

InfoMap detection algorithm:

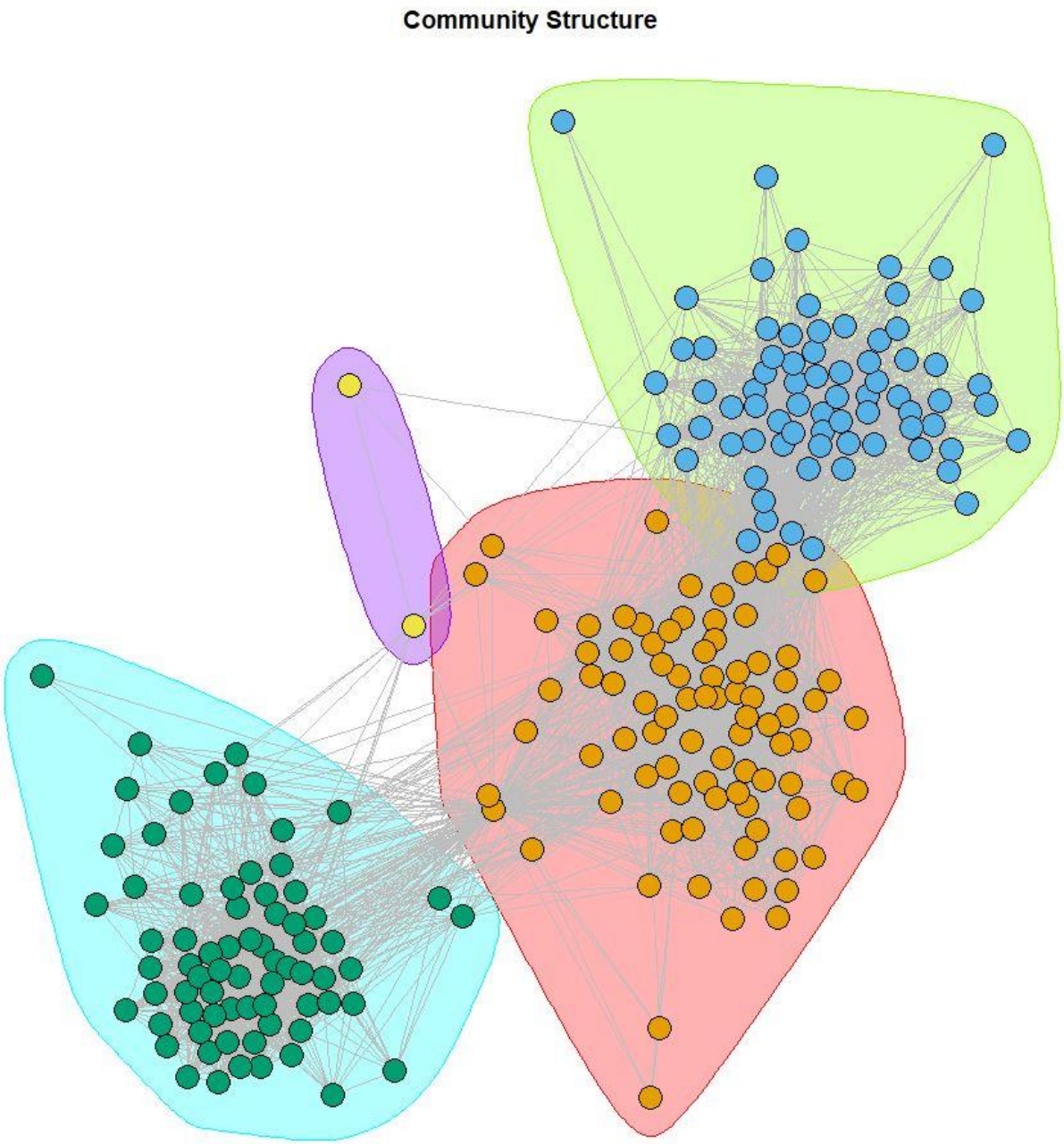


Figure4.6 Personal network without node of InfoMap Algorithm

```
Community sizes
 1  2  3  4
85 73 71  2
```

Figure4.7 Personal network without node of InfoMap Algorithm

Comparison of each detection algorithm results:

By comparing our results, we can find that the number of communities in Fast-Greedy and InforMap are remain same to Questions 3. The number of communities of Edge-Betweenness Algorithm increases to 11 without node. More specifically, the size of each community in Fast-Greedy and InforMap are remain same to Questions 3.

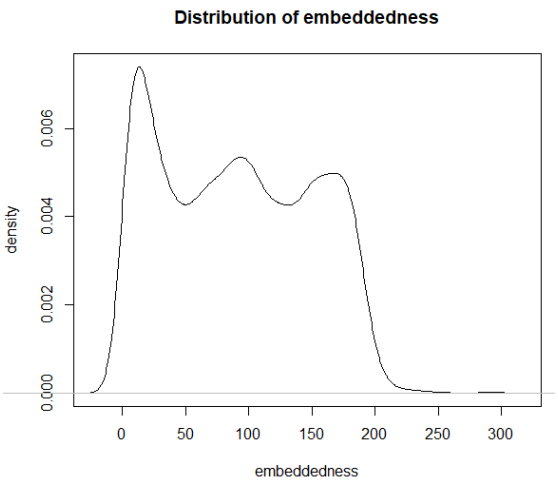
Question 5

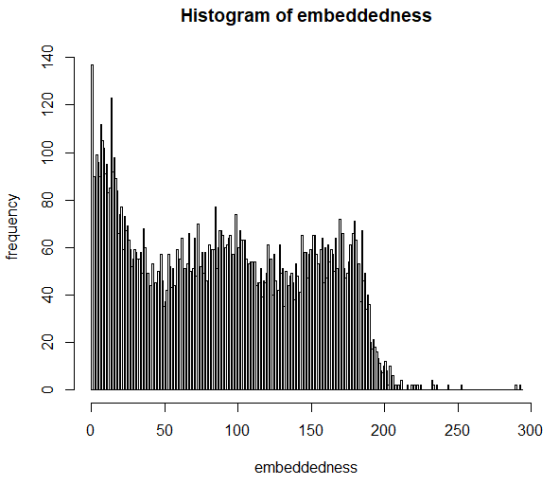
This question calculates 2 important features, dispersion and embeddedness.

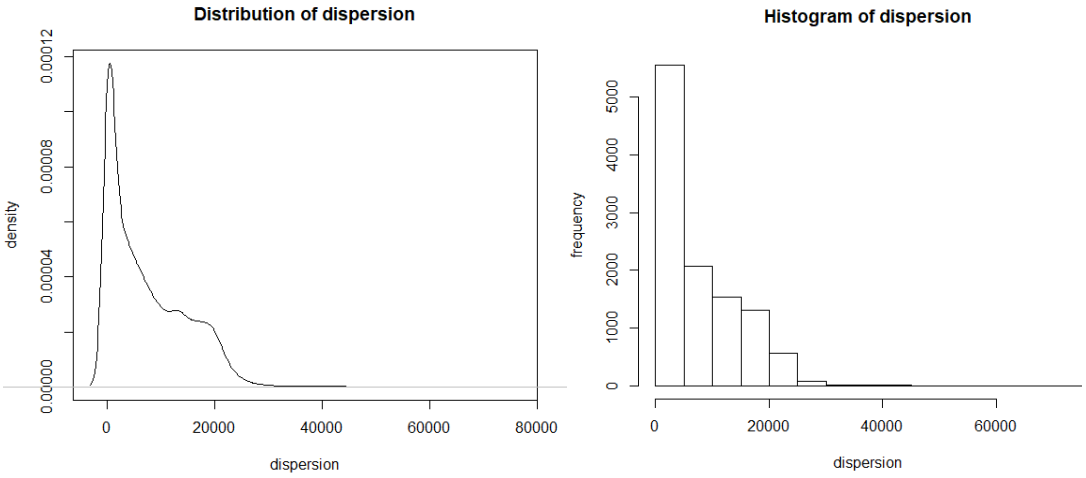
- Embeddedness is the number of mutual friends a node shares with the core node. We calculate the neighbor of core node and each node and find the intersection of them. The intersection is their mutual friends.
- Dispersion is the sum of the distances between every pair of the mutual friends a node shares with core node. In this section, we use distance () function to calculate the shortest distance between two nodes. We calculate the shortest distance between each pair of mutual friends and sum them up. The smaller the dispersion is, the more likely the mutual friends can know each other.

5.1 Distribution of embeddedness and dispersion

There are total 40 core nodes that meet the requirement. We calculate the embeddedness and dispersion for all nodes in the personal network of all core nodes. The distribution is as follow.







5.2 Visualization of Personal Networks structures

In this section, we find the nodes with largest embeddedness, dispersion and embeddedness over dispersion in the personal network. The core node we selected for this part are node 0, node 348 and node 483.

5.2.1 Embeddedness

In this part, we calculate the embeddedness of each nodes in the personal networks of 3 core nodes separately. We also used the fastgreedy algorithm to find the communities structure of the personal network. The communities are separated in different color. The big white node is the core node. The big pink node is the node with largest embeddedness value. The edges incident to the node with largest embeddedness is highlighted with thick red lines. Following are the visualization of the personal networks.

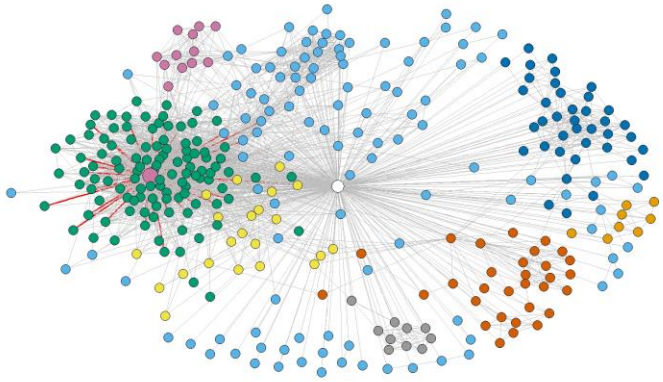


Figure 5.2.1.1 personal network for core node 0

`	1	2	3	4	5	6	7	8
Community size	114	112	22	39	31	12	10	8

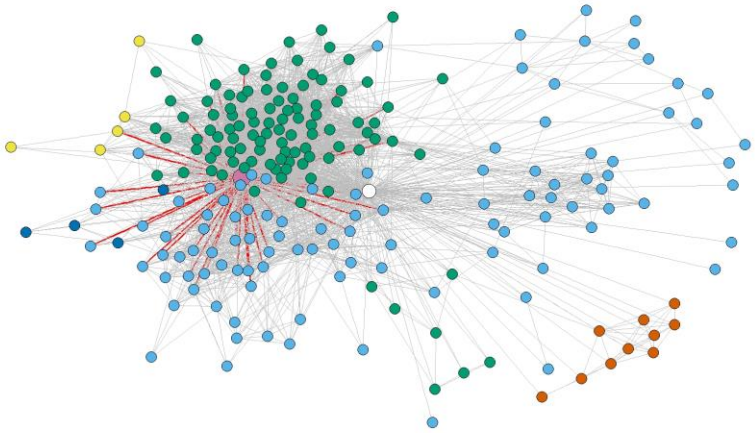


Figure 5.2.1.2 personal network for core node 348

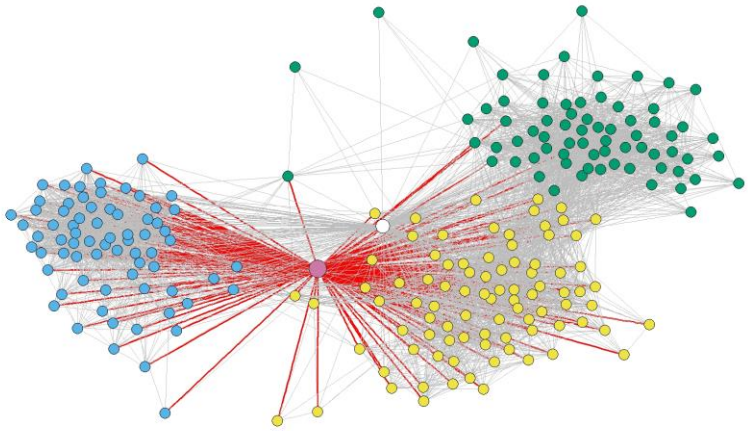


Figure 5.2.1.2 personal network for core node 483

	1	2	3
Community size	71	72	89

5.2.2 Dispersion

In this part, we calculate the Dispersion of each nodes in the personal networks of 3 core nodes separately. We also used the fastgreedy algorithm to find the communities structure of the personal network. The communities are separated in different color. Since the dispersion is calculated by removing the core node and the node is computing, some node may be disconnected with the rest of the nodes. Therefore, we will skip those pairs of mutual friends in the dispersion calculation.

The big white node is the core node. The big pink node is the node with largest dispersion value. The edges incident to the node with largest dispersion is highlighted with thick red lines. Following are the visualization of the personal networks.

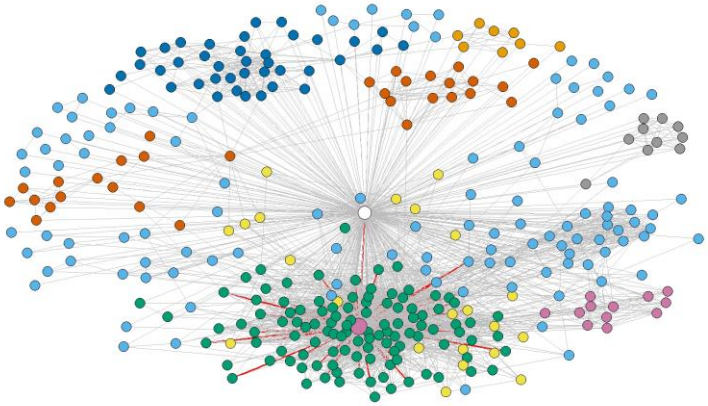


Figure 5.2.1.1 personal network for core node 0

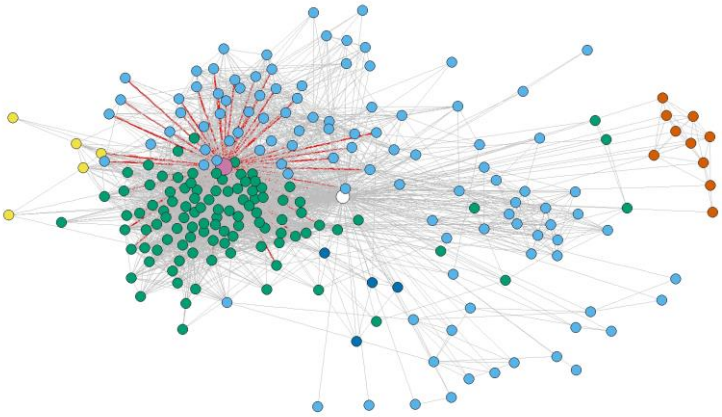


Figure 5.2.1.2 personal network for core node 348

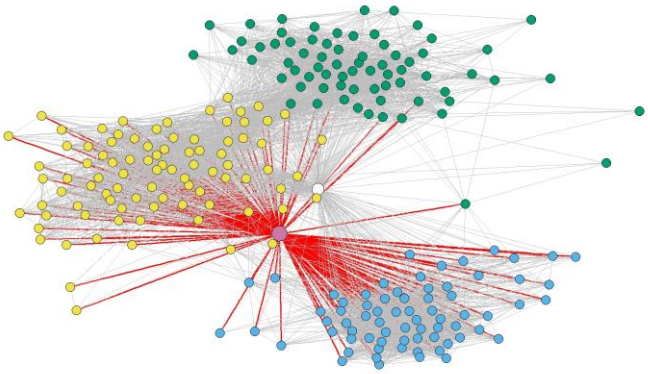


Figure 5.2.1.2 personal network for core node 483

5.2.3 $\frac{Dispersion}{Embeddedness}$

In this part, we calculate the $\frac{Dispersion}{Embeddedness}$ of each node in the personal networks of 3 core nodes separately. We also used the fastgreedy algorithm to find the communities structure of the personal network. The communities are separated in different color. Since the dispersion is calculated by removing the core node and the node is computing, some node may be disconnected with the rest of the nodes. Therefore, we will skip those pairs of mutual friends in the $\frac{Dispersion}{Embeddedness}$ calculation.

The big white node is the core node. The big pink node is the node with largest $\frac{Dispersion}{Embeddedness}$ value. The edges incident to the node with largest $\frac{Dispersion}{Embeddedness}$ is highlighted with thick red lines. Following are the visualization of the personal networks.

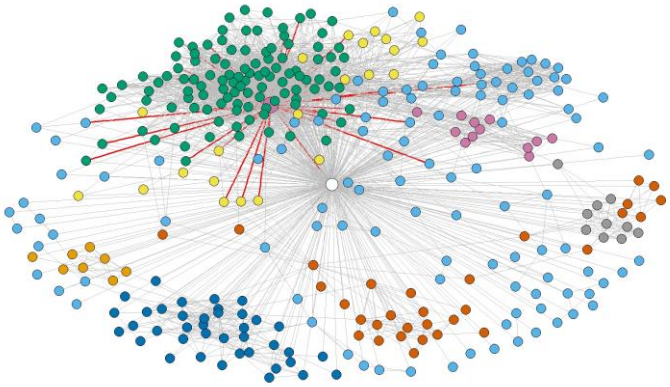


Figure 5.2.1.1 personal network for core node 0

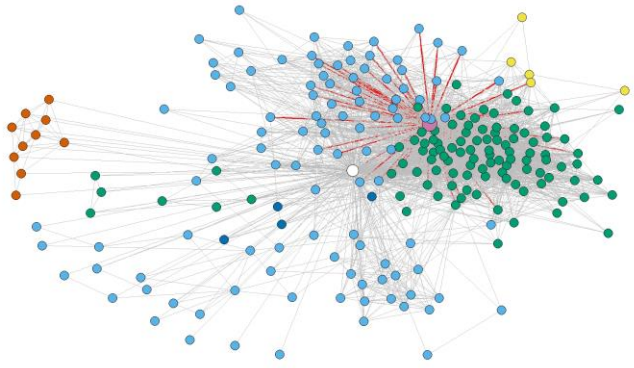


Figure 5.2.1.2 personal network for core node 348

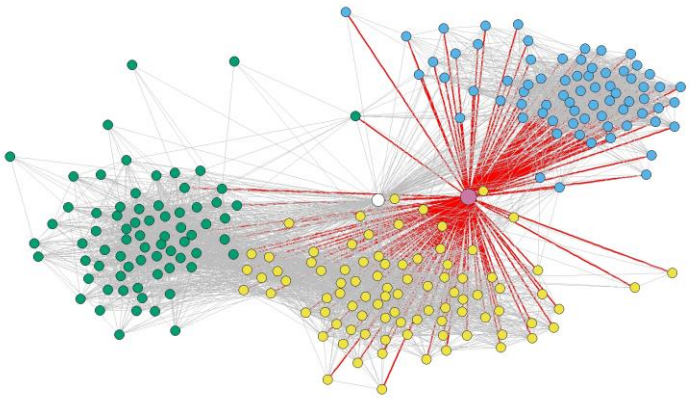


Figure 5.2.1.2 personal network for core node 483

5.2.4 Conclusion

Embeddedness:

The embeddedness value of a node will tell how strong is the node connected to the core node. Which means this node have most number of common friends with the core node. This is usually happened when the community size is large. From the figures, we can see, all the high lightened node belongs to the large group. Therefore, the larger the community size is, the larger probability that the node in the community has the maximum embeddedness.

Dispersion:

The node with largest dispersion in the graph shows that mutual friends of core node and the pink node are not connected to each other very well. That means the core node's community is separated from the testing node's community, however they have some mutual friends.

Dispersion/Embeddedness:

The high dispersion to embeddedness ratio shows that the core node and the pink node how likely to be in romantic relationship. The node with the highest ratio which is called normalized dispersion can predict the romantic relationship between two nodes correctly.

Question 6

In this question, we are going to focus on the characteristic of single person's personal networks and try to find the similarity among them. These similar personal network can be colleagues, family members and classmates. The features we use to characterize the closeness are: **clustering coefficient density of nodes, and community sizes.**

6.1 Features of Community

Before we calculate the values of feature in the data set, first we should take a look at the definition of each feature.

1. Clustering coefficient: It is a measure of the degree to which nodes in a graph tend to cluster together. It measures the closeness among neighbors of a node.

$$\text{Clustering coefficient} = \frac{2(number\ of\ direct\ links\ between\ neighbors\ of\ vertex\ V)}{(degree\ of\ node\ V) * (degree\ of\ node\ V - 1)}$$

2. Density: It is the ratio of the number of edges and the number of possible edges for certain node.

3. Community size: It is a primitive value companied with community calculation, which give us the size of each community within each personal network.

6.2 Analysis of Similarity among Communities

In the analysis process, we want to distinguish similar communities according to the value of the three above mentioned features. In order to classify them into 2 different categories, namely Type 1 and Type 2, we examined the maximum and minimum value of each feature. In the whole process, we only consider communities with more than 10 nodes within it.

The maximum values of each features indicate some reasonable intuition, such as "they are closely connected" and "they may share the same type". If all the communities with similar value of clustering coefficient share more closeness with each other, and such community maybe type of classmates, business partners and so on. Large density values may indicate possible future relationships. There are 40 personal network in the date set. The result obtained for each personal community is given in below tables.

Index	Clustering Coefficient	Density	Community Size
1	0.0073 **0.0229** 0.0008 0.0030 0.0014 0.0009 0.0006	0.0671 0.2177 0.2094 0.2371 0.1814 0.7179 **0.7090**	**114** 112 22 39 31 12 10 8
2	0.0157 **0.0319** 0.001	0.0794 0.1485 **0.2519**	464 **484** 70
3	0.0300, 0.0008, **0.0665**, 0.0011	0.1358, 0.3484, 0.3582, **0.5636**	**107** 11 98 10
4	0.0469, 0.0429, **0.0539**	**0.4874**, 0.4345, 0.3580	71 72 **89**
5	0.0983, **0.1319**	**0.6193**, 0.3503	81 **125**
6	0.1332, **0.1547**	0.5207, **0.6050**	**109 109**
7	0.0028, 0.1159, **0.1212**	**0.6583**, 0.6518, 0.4406	15 98 **122**
8	**0.1787**, 0.1055	0.4833 **0.6491**	**133** 88
9	0.1034, **0.1641**	**0.6577**, 0.4336	83 **129**
10	0.0108, 0.1022, **0.1417**	**0.6984**, 0.6437 0.5867	25 81 **100**
11	**0.1743**, 0.1019	0.4928, **0.5899**	**139** 97
12	**0.0214**, 0.0032, 0.0178, 0.0004	0.0974, 0.0735, 0.2183, **0.7157**	**371** 165 226 19
13	0.1056, **0.1379**	**0.5316**, 0.4995	100 **118**
14	**0.2029**, 0.1064	**0.6071**, 0.5647	**116** 87
15	**0.1512**, 0.1077	**0.5586**, 0.4825	**108** 98
16	0.0019, **0.1684**, 0.0787	**0.6263**, 0.5325, 0.5057	13 **137** 96
17	**0.2086**, 0.0800	**0.5682**, 0.5502	**121** 76
18	0.0562, 0.0493, **0.0976**	**0.6511**, 0.6205, 0.5136	74 71 **110**
19	**0.0568**, 0.03706, 0.0035, 0.0050	**0.5889**, 0.2133, 0.1090 0.5498	234 **314** 136 72

20	**0.1893**, 0.1691	**0.8264**, 0.5962	106 **118**
21	0.0401, 0.0087, **0.2951**	0.4144, 0.7898, **0.8271**	69 23 **133**
22	**0.2269**, 0.1680	**0.7761**, 0.7557	**109** 95
23	**0.2163**, 0.0450	**0.6230**, 0.2768	**120** 82
24	0.0118, **0.1611**, 0.1509	0.8166, **0.8332**, 0.7304	24 89 **92**
25	0.0020, **0.1892**, 0.1701	0.7454, **0.7998**, 0.7493	10 **98** 96
26	0.0987, **0.2656**	0.5049, **0.8201**	97 **125**
27	**0.1763**, 0.1442, 0.0084	0.7223, **0.8270**, 0.8008	**103** 87 21
28	**0.3114**, 0.1076	**0.8138**, 0.6944	**126** 80
29	**0.2474**, 0.1362	**0.8189**, 0.6372	**113** 95
30	0.1359, **0.2123**	0.5773, **0.7676**	107 **116**
31	**0.1659**, 0.0116, 0.1458	**0.8144**, 0.7800, 0.7487	**90** 24 88
32	0.0836, **0.2466**	0.4608, **0.8145**	99 **128**
33	**0.2866**, 0.1019	**0.7593**, 0.6559	**178** 114
34	0.1295, **0.2602**	0.6600, **0.8037**	91 **117**
35	0.0041, **0.1971**, 0.1566	0.7904, 0.7767, **0.8120**	14 **101** 88
36	0.1731, **0.2271**	0.7321, **0.8201**	97 **105**
37	**0.2240**, 0.1259	**0.7781**, 0.5657	**157** 138
38	0.0031, 0.1481, **0.1823**	0.7948, 0.6530, **0.8039**	12 **95 95**
39	0.1158, **0.2660**	0.5771, **0.8359**	92 **116**
40	0.0007, 0.0086, **0.0161**, 0.0012, 0.0009, 0.0009, 0.004	0.4761, 0.1351, 0.1105, 0.2137 **0.6736**, 0.1769, 0.2047	20 137 **208** 42 19 39 76

Question 7

In this question, we are going to analysis the network in another social network – Google + ego network. What's different in Google + networks is that it contains directed network, which means having someone in your list doesn't necessarily mean that you are also in his/her list. We defined circle as the tags we put on the relationships. Having the Google+ dataset, we can analysis the community structure and compare the overlap between community and circles.

7.1 Data Overview

In the dataset, each ego node has its unique node ID, which is represented by the filename. There are **132 different ego nodes** in the dataset. For each unique ID, there's a file with extension ".circle" indicating the circles of that ego node. If we only consider nodes with more than 2 circles, we can find **57 ego nodes** satisfying this condition.

7.2 Community Structure

In the dataset, there's another group of file with extension of ".edges" which contains the directed edges that pointing to other nodes. We add this file then we got a personal network of

certain ego node. In this project, we used two different algorithm to find the community structure, which are *walktrap.community* and *infomap.community*.

Here we run the algorithm to detect the structure of ego node ID = 100535338638690515335.

7.2.1 Walktrap Community

The community membership structure is showed below:

Index	Frequency
1	58
2	627
3	388
4	114
5	1

Table 7-1 Community Structure of Walktrap algorithm

7.2.2 Infomap Community

The community membership structure is showed below:

Index	Nodes	Index	Nodes
1	1109	10	3
2	26	11	2
3	5	12	3
4	9	13	2
5	6	14	2
6	2	15	2
7	5	16	2
8	5	17	2
9	3		

Table 7-2 Community Structure of Infomap algorithm

7.3 Overlap between Circle and Community

The overlap is represented as the intersection of the nodes in a circle and the nodes in the community. In equation, it can be expressed as:

$$Overlap = \frac{number\ of\ common\ nodes}{number\ of\ nodes\ in\ the\ circle}$$

We need to loop through each user and its circles to check the comparison. For analysis purpose, we chose 3 ego nodes. Then we use a confusion matrix to show the degree of overlap. In the figures

below, the darker color indicates that the overall overlap between community and circle is more intense.

7.3.1 Walktrap Community

The overlap calculated by walktrap community is presented in the following plots,

For Node 1:

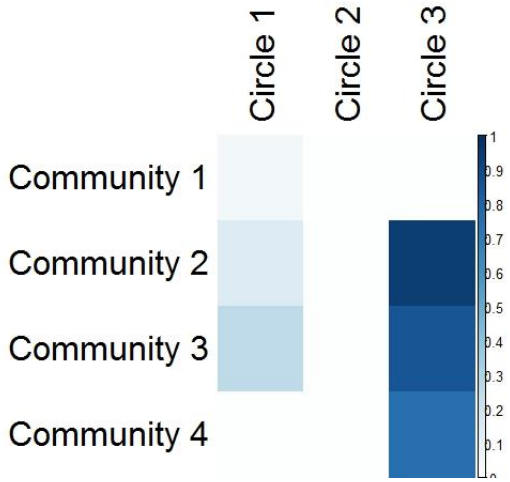


Figure 7-1 Walktrap Community Overlap Matrix (Node 1)

For Node 2:

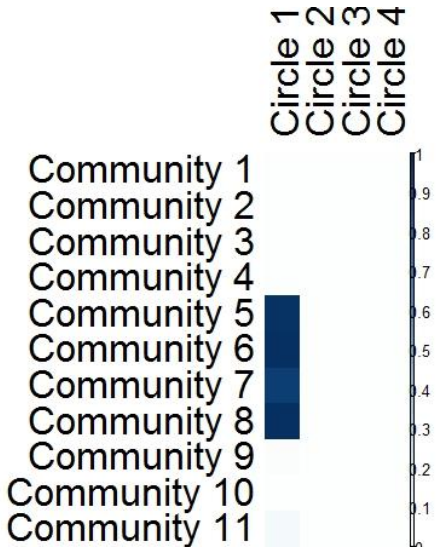


Figure 7-2 Walktrap Community Overlap Matrix (Node 2)

For Node 3:

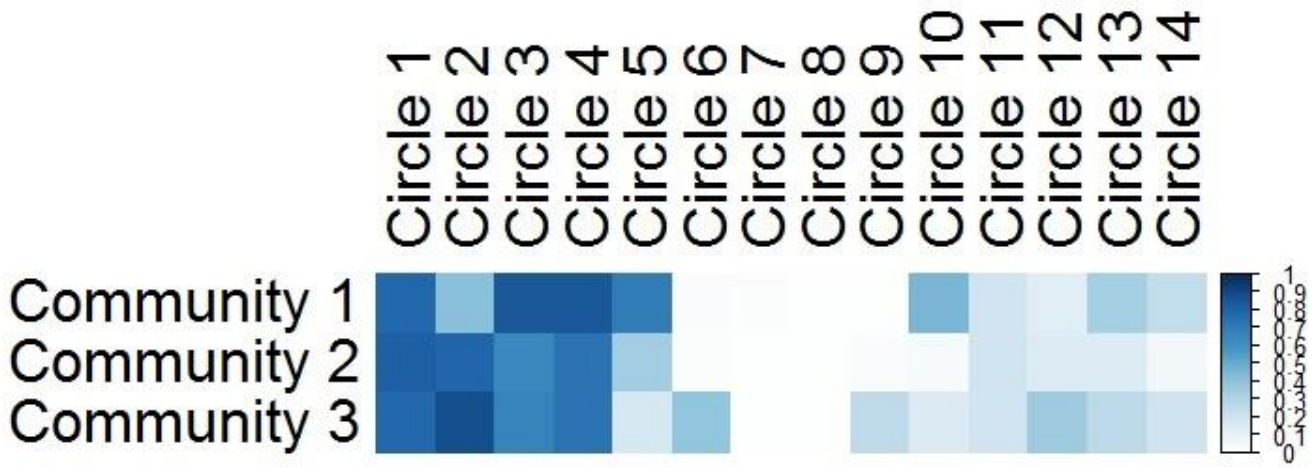


Figure 7-3 Walktrap Community Overlap Matrix (Node 3)

7.3.2 Infomap Community

For Node 1:

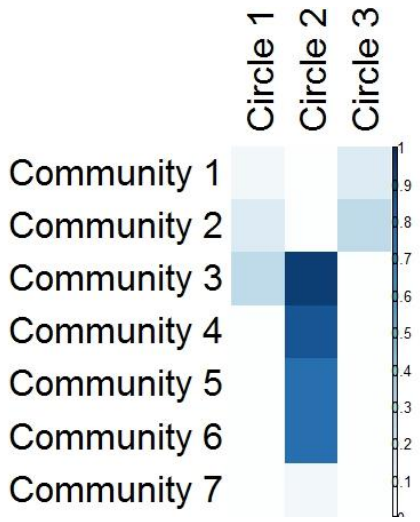


Figure 7-4 Infomap Community Overlap Matrix (Node 1)

For Node 2:

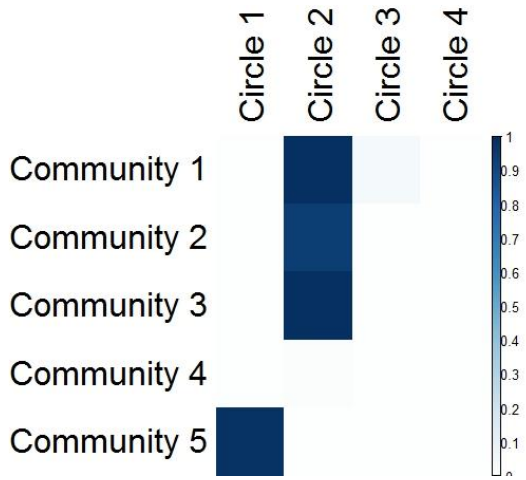


Figure 7-5 Infomap Community Overlap Matrix (Node 2)

For Node 3:

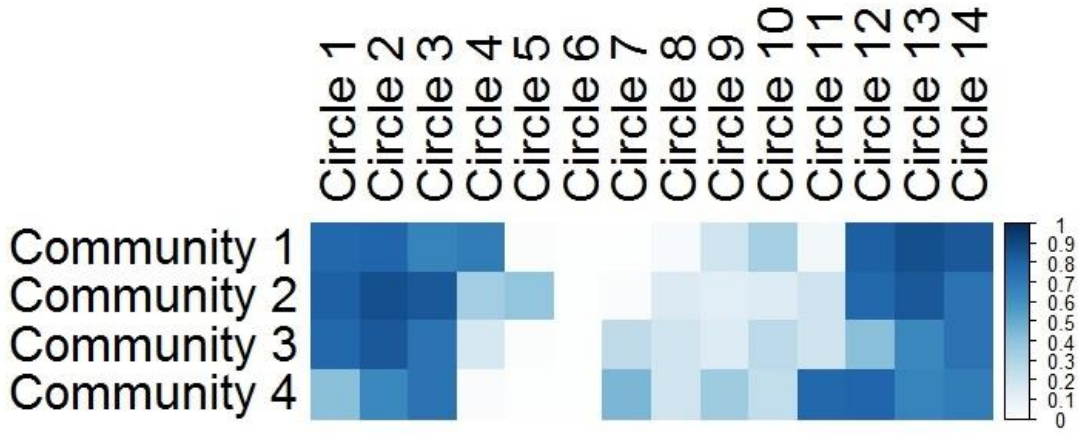


Figure 7-6 Infomap Community Overlap Matrix (Node 3)

7.3.3 Result Analysis

Some intuitive deduction can be found from the above figures.

We can observe that the communities of ego-nodes which have fewer number of circles seem to be concentrate within these in default circles. The relationships are assigned by default by Google+. This is depicted in the plots of Nodes 1 & 2. For user's with larger number of circles, community nodes are shown to be distributed among the circles and not a defined structure can be seen.

Besides, as the number of circles increases users tend to choose their relationship with people and choose their own circles for them, for example business, colleagues etc. In situations like these people related to the ego-node might belong to multiple circles. For example a person can be in the Family as well as Friends circle.