Assignment_11_Computer

Owen Ostermann: 7884571: B03

9/12/2022

Social Network of Friends in a Karate Club

The graph "karate" contains the network of friendships between the 34 members of a karate club at a US university, as described by Wayne Zachary in 1977. The data comes from W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977). The data was found online in a mtx file on https://networkrepository.com/soc-karate.php. I have converted the data into a csv file in which i have uploaded as well. The graph contains 34 nodes (the 34 members of the karate club) and 78 edges (the friendship connections between the members of the club). The graph is un-directed as it is assumed if one is a friend that it goes both ways between the members.

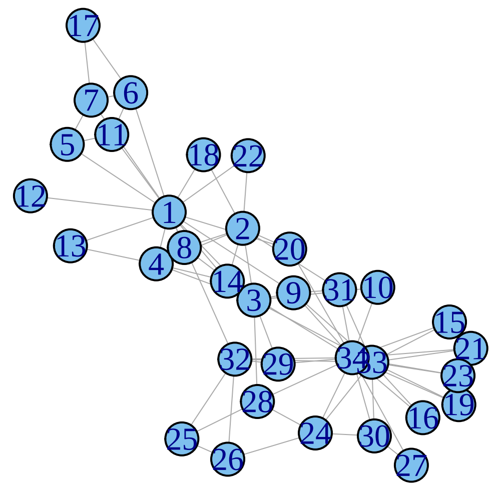
```
karate = read.csv("Karate.csv")
vertex_list = c()

for (i in 1:nrow(karate))
{
  vertex_list = append(vertex_list,as.numeric(karate[[i,1]]))
  vertex_list = append(vertex_list,as.numeric(karate[[i,2]]))
}
```

First lets create a simple igraph object

```
karate_graph = graph(vertex_list, directed = FALSE)
karate_graph = simplify(karate_graph)
plot(karate_graph, main = "Social Network of Friends in a Karate Club",
     vertex.color = "skyblue2",
     vertex.size = 15, edge.width = 0.5)
```

Social Network of Friends in a Karate Club



Some first observations from the graph. We can see that person 1,33,34 are very central in the graph. They appear to be popular ones in the karate club. Whereas person 17 or 12 are not very popular in the club. They have a small amount of edges corresponding to them.

Eccentrcities and Related Measures

We can observe the table that is the geodesic distances between the vertices in the graph. The geodesic distance is the length of the shortest path from one vertex in the graph to another. Since there are 34 vertices. The distance function produces a matrix of size 34 x 34. With the rows corresponding to vertices and then the column is the shortest path to the other vertices. For simplicity I am just showing the upper left 6x6 so we have a sense of how the geodisic table looks.

```r
dist = distances(karate_graph)
dist[1:6,1:6]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    2    1    1    1    1
## [2,]    2    0    1    1    3    3
## [3,]    1    1    0    1    2    2
## [4,]    1    1    1    0    2    2
## [5,]    1    3    2    2    0    2
## [6,]    1    3    2    2    2    0
```

From this table we can interpret the entries as connections. For example, in row 6, we observe that person 6 is directly friends with person 1 and the shortest path to get from vertex 6 to 1 is 1. But person 6 is not directly friends with person 2, they are friends with them through other person as the shortest path is 3 to get from vertex 6 to 2. The table does not give us information about who we have to go through to find connections between friendships, but this can be done with computing shortest paths between vertices.

We can calculate the eccentricities of the vertices. The eccentricity is the maximum distance from a vertex to any other vertex in the graph.

```
eccentricity = eccentricity(karate_graph)
vertex = 1:34
df = data.frame(vertex,eccentricity)
df
```

```
##    vertex eccentricity
## 1       1            3
## 2       2            4
## 3       3            3
## 4       4            3
## 5       5            4
## 6       6            4
## 7       7            4
## 8       8            4
## 9       9            3
## 10     10            4
## 11     11            4
## 12     12            4
## 13     13            4
## 14     14            3
## 15     15            5
## 16     16            5
## 17     17            5
## 18     18            4
## 19     19            5
## 20     20            3
## 21     21            5
## 22     22            4
## 23     23            5
## 24     24            5
## 25     25            4
## 26     26            4
## 27     27            5
## 28     28            4
## 29     29            4
## 30     30            5
## 31     31            4
## 32     32            3
## 33     33            4
## 34     34            4
```

```
radius(karate_graph)
```

```
## [1] 3
```

```
diameter(karate_graph)
```

```
## [1] 5
```

From this table we can see that the maximum of the eccentricities is 5 (diameter), this means that the graph is pretty closely connected, in other words, the karate club is pretty close.

We can compute the central points of the graph to see which people could be “leaders” or “popular” in the karate club.

```
central_points = which(radius(karate_graph) == df$eccentricity)
central_points
```

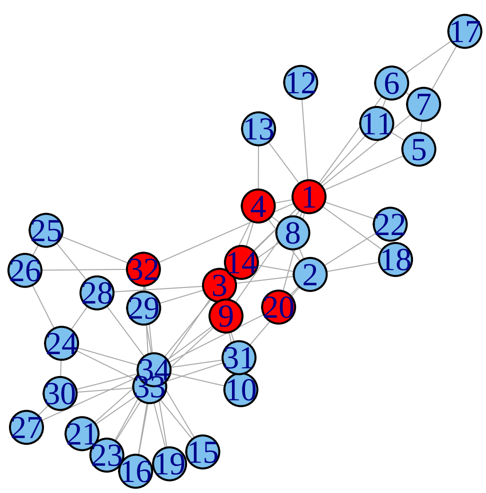
```
## [1]  1  3  4  9 14 20 32
```

This means that the above people are closely connected with other members of the club as the max distance to any other vertices from that person (the ones above) is 3. They are popular among the group.

Here we plot the set of central points in red. These are the people in the group who at maximum are friends with someone through two others, otherwise they are friends through one other person, or directly friends with a person.

```
V(karate_graph)[which(radius(karate_graph) == df$eccentricity)]$color = "red"
V(karate_graph)[which(radius(karate_graph) != df$eccentricity)]$color = "skyblue2"
plot(karate_graph,
     main = "Social Network of Friends in a Karate Club (Central Points in Red)",
     vertex.size = 15, edge.width = 0.5)
```

Social Network of Friends in a Karate Club (Central Points in Red)

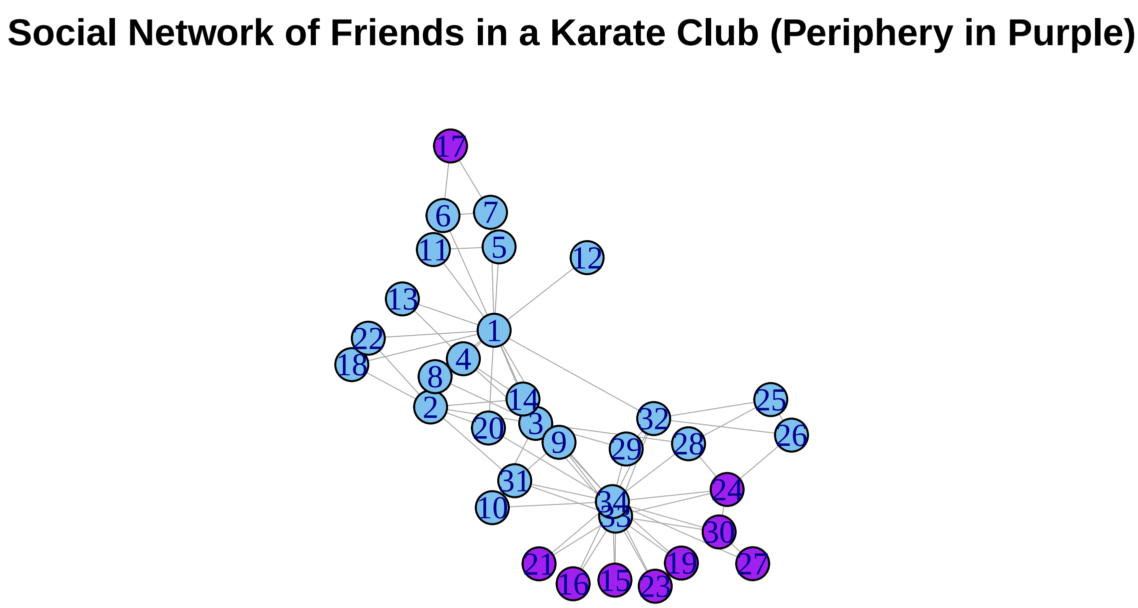


We can also plot the periphery points. Which is the set of vertices which achive the diameter.

```
periphery = which(diameter(karate_graph) == df$eccentricity)
periphery
```

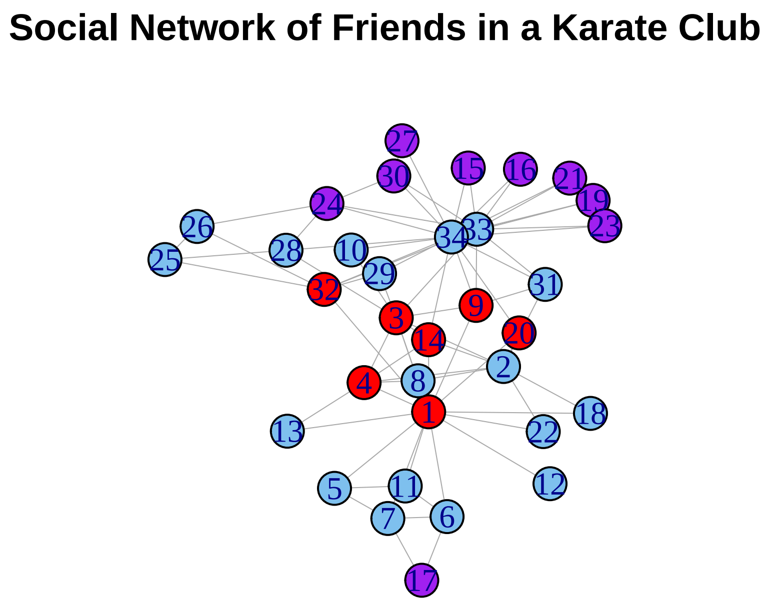
```
## [1] 15 16 17 19 21 23 24 27 30
```

```
V(karate_graph)[which(diameter(karate_graph) == df$eccentricity)]$color = "purple"
V(karate_graph)[which(diameter(karate_graph) != df$eccentricity)]$color = "skyblue2"
plot(karate_graph,
     main = "Social Network of Friends in a Karate Club (Periphery in Purple)",
     vertex.size = 15, edge.width = 0.5)
```



Putting the two graphs together to observe the central and periphery points of the network.

```
V(karate_graph)[which(radius(karate_graph) == df$eccentricity)]$color = "red"
V(karate_graph)[which(diameter(karate_graph) == df$eccentricity)]$color = "purple"
plot(karate_graph,
     main = "Social Network of Friends in a Karate Club",
     vertex.size = 15, edge.width = 0.5)
```



Girth

The girth is the total number of edges in the shortest cycle of a graph.

```
girth(karate_graph)
```

```
## $girth
## [1] 3
##
## $circle
## + 3/34 vertices, from 1cca5ad:
## [1] 3 1 4
```

The girth is 3 and it includes person 1, 2, and 3. This means that a possible shortest cycle is between those three people. They are a smaller group within the club that are connected and close friends.

The density of a graph is the fraction of maximum number of edges. If the graph has p edges and n vertices then its density is (2p/n(n-1)).

```
edge_density(karate_graph)*100
```

```
## [1] 13.72549
```

Here 13.7% of possible edges are present in the network. This is a small fraction and tells us that there is not a large amount of direct friendships within the karate club.

Connectedness

Lets check to see if the graph is connected or not.

```
is.connected(karate_graph)
```

```
## [1] TRUE
```

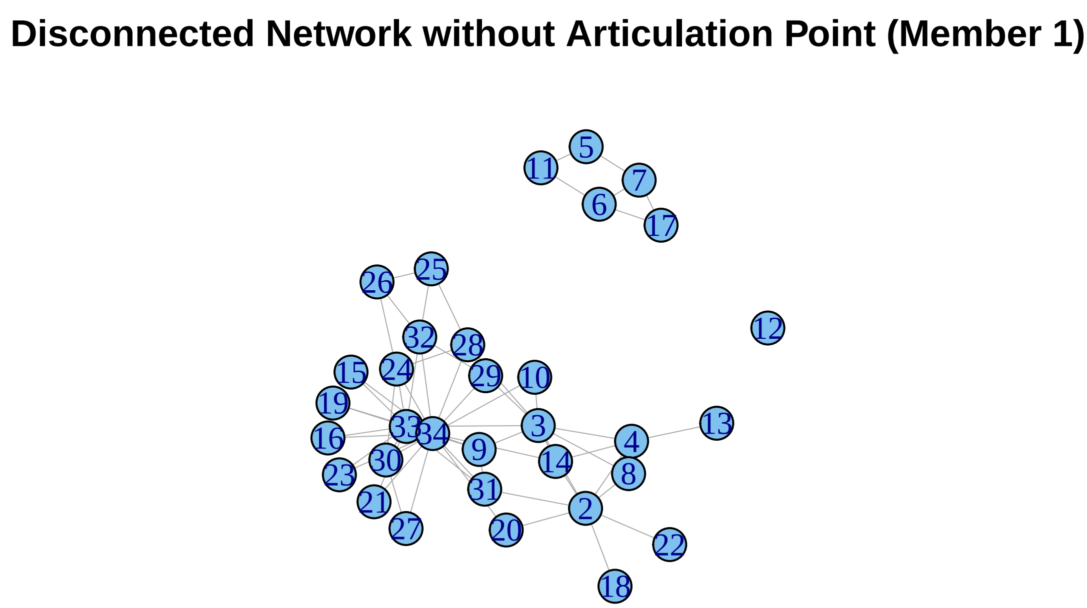
Thus the graph is connected. Lets see if there are any vertices that if we remove it makes the graph disconnected.

```
articulation_points(karate_graph)
```

```
## + 1/34 vertex, from 1cca5ad:
## [1] 1
```

We see that member number one, if removed would make the graph disconnected. Thus we could assume that member one has a large amount of friendships in the group and that if removed the group is not connected, that is not everyone has a mutual link anymore.

```
V(karate_graph)$name = V(karate_graph)
karate_graph_disconnected = delete_vertices(karate_graph, 1)
plot(karate_graph_disconnected,
     main = "Disconnected Network without Articulation Point (Member 1)",
     vertex.size = 15, edge.width = 0.5, vertex.color = "skyblue2")
```



```r
# connected?
is_connected(karate_graph_disconnected)
```

```
## [1] FALSE
```

Clearly the removal of member 1 creates 3 different social groups upon the karate club. Member 1 was member 12's only link to the bigger portion of the graph, as well member 1 was the group containing 17,7,6,5, and 11's only link to the larger portion of the group.

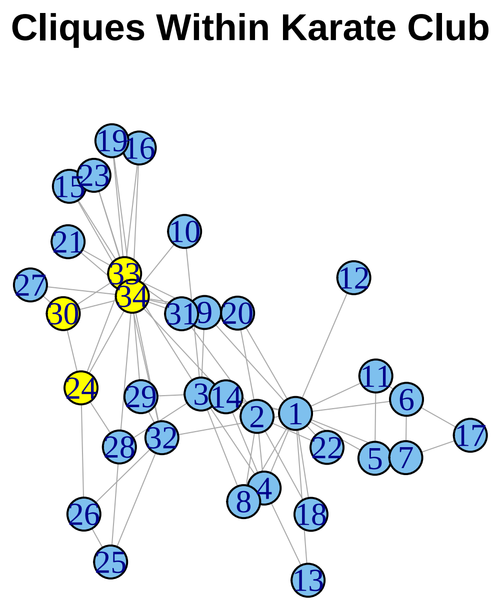
Cliques

A clique is a sub graph of the original graph in which all vertices in the sub-graph are adjacent. With regards to the karate club, this is what the name suggests. It shows social groups within the karate club. Below shows one of the largest cliques in the karate club.

```r
lc = largest_cliques(karate_graph)
clique = lc[[1]]
clique
```

```
## + 4/34 vertices, named, from 1cca5ad:
## [1] 34 33 30 24
```

```r
V(karate_graph)$color = "skyblue2"
V(karate_graph)[clique]$color = "yellow"
plot(karate_graph,
     main = "Cliques Within Karate Club",
     vertex.size = 15, edge.width = 0.5)
```



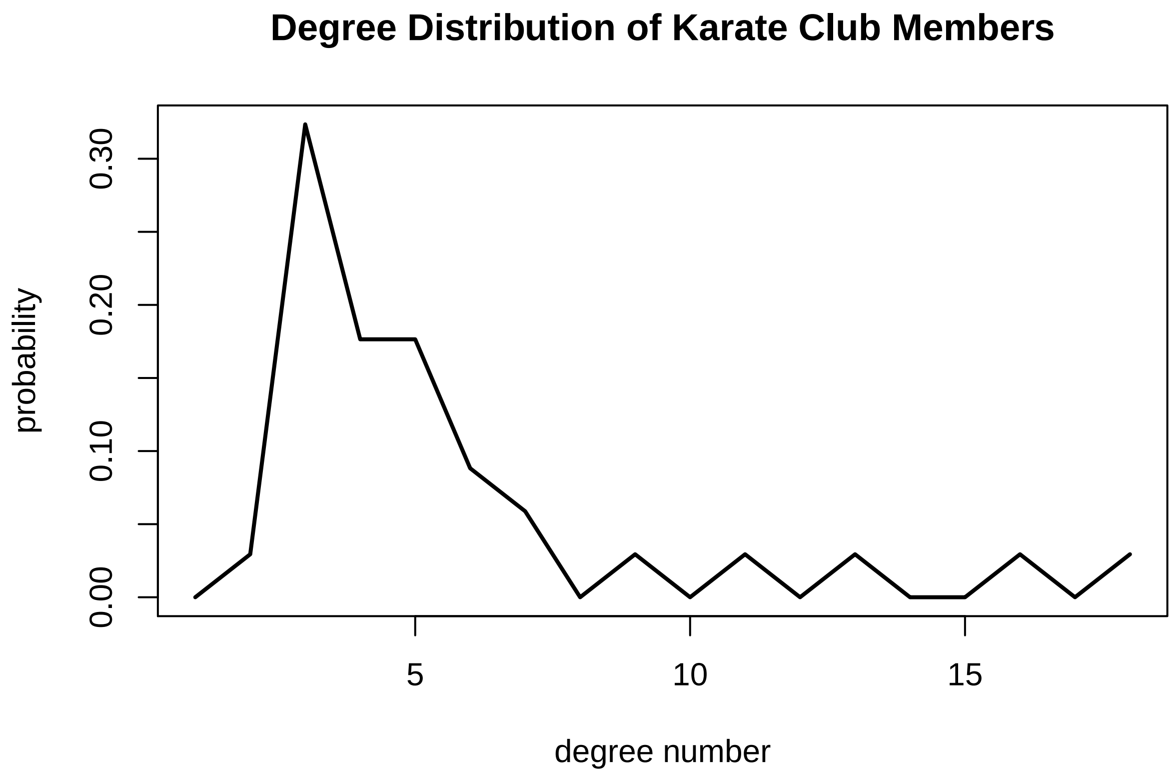
Centrality and Degrees

```
degree(karate_graph)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 15  8 10  6  3  4  4  4  5  2  3  1  2  5  2  2  2  2  2  3  2  2  2  5  3  3
## 27 28 29 30 31 32 33 34
##  2  4  3  4  4  6 12 17
```

We see that some of the members with the largest degrees are member 1,2,3,33, and 34. This means that these members have the most friendships in the karate club. Lets view the average distribution of degrees in the network.

```
distribution = degree_distribution(karate_graph)
y_max = max(distribution)
plot(distribution, type = "l", lwd = 2, col = "black",
     ylim = c(0,y_max), xlab = "degree number", ylab = "probability",
     main = "Degree Distribution of Karate Club Members")
```



The degree distribution is a probability distribution of the degrees throughout the graph. We see that on average from the distribution that most members have between 3-7 friends in the karate club (adjacent edges between vertices).

Lets look at the nearest neighbors in the graph. Below is the average of nearest neigbours for each member.

```
knn = knn(karate_graph)
knn$knn
```

```
##         1         2         3         4         5         6         7         8
##  4.000000  4.500000  6.400000  7.333333  7.333333  6.000000  6.000000  9.750000
##         9        10        11        12        13        14        15        16
## 11.600000 13.500000  7.333333 15.000000 10.500000 11.200000 14.500000 14.500000
##        17        18        19        20        21        22        23        24
##  4.000000 11.500000 14.500000 13.333333 14.500000 11.500000 14.500000  8.000000
##        25        26        27        28        29        30        31        32
##  4.333333  4.666667 10.500000  8.750000 11.000000  9.000000 10.500000  8.833333
##        33        34
##  5.083333  3.823529
```

The values above measure the average friendships of each member in the karate club.

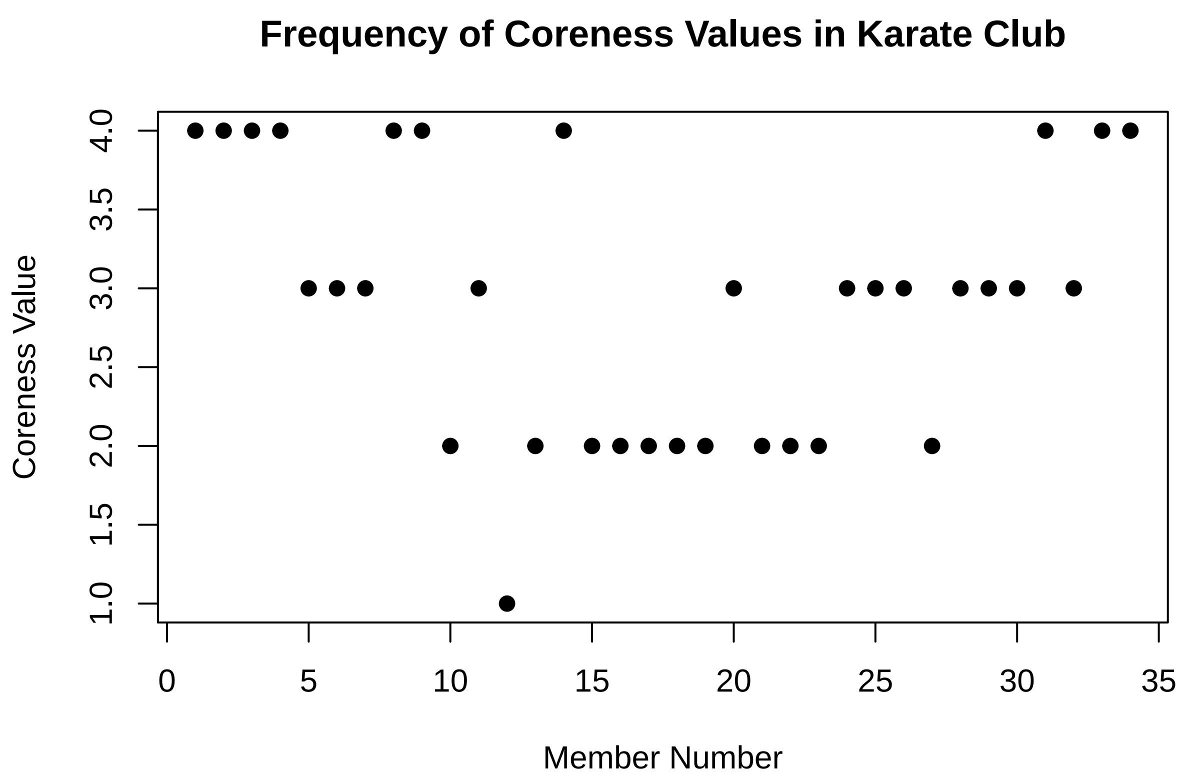
Coreness

The k-core of a graph is a maximal sub graph in which each vertex has degree at least K.

```r
core = coreness(karate_graph)
core
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  4  4  4  4  3  3  3  4  4  2  3  1  2  4  2  2  2  2  2  3  2  2  2  3  3  3
## 27 28 29 30 31 32 33 34
##  2  3  3  3  4  3  4  4
```

```r
plot(core,
     main = "Frequency of Coreness Values in Karate Club",
     xlab = "Member Number", pch = 19, ylab = "Coreness Value")
```



This measures the sparsity of the graph. A sparse graph is one in which there are few edges. Above we can see that the coreness of the karate club network only has one k-core with a value of 1 (vertex 13). Thus the karate club network is relatively connected (rarely any member has only one friend within the club).

Betweenness

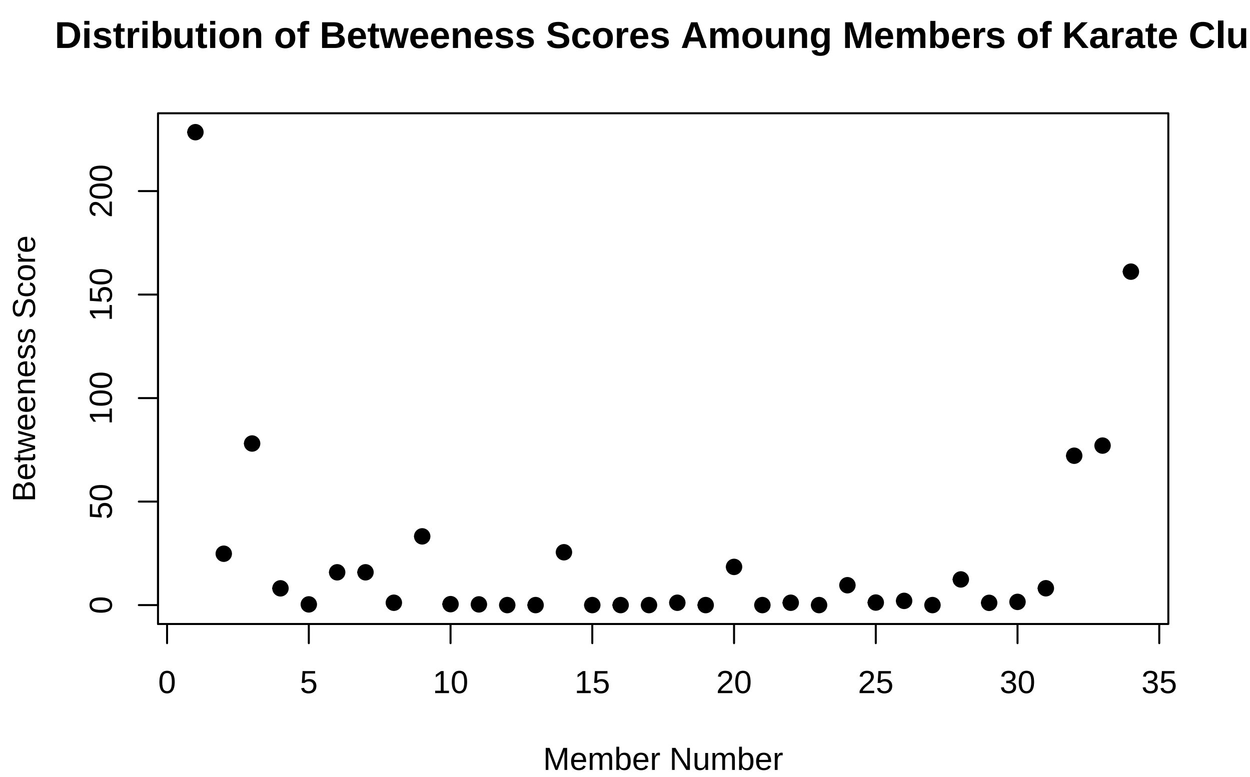
Betweeness is the measure of centrality in a graph based on the shortest paths. We can interpret this with respect to the karate club as the connectedness and closeness of the club members. A high betweenss means that member is central and a member that has a high influence in the network on mutual friendships (connections between members).

```r
between = betweenness(karate_graph)
between
```

```
##           1           2           3           4           5           6 
## 228.4598901  24.8119048  78.0482295   8.0816850   0.3333333  15.8333333 
##           7           8           9          10          11          12 
##  15.8333333   1.1269231  33.1960317   0.4476190   0.3333333   0.0000000 
##          13          14          15          16          17          18 
##   0.0000000  25.5197192   0.0000000   0.0000000   0.0000000   1.1269231 
##          19          20          21          22          23          24 
##   0.0000000  18.4506716   0.0000000   1.1269231   0.0000000   9.6000000 
##          25          26          27          28          29          30 
##   1.2166667   2.0277778   0.0000000  12.3920635   1.0745421   1.5428571 
##          31          32          33          34 
##   8.1300366  72.1595238  77.0443223 161.0823565
```

We observe that member 1, 3, 32, 33, and 34 have high betweeness. They create a lot of mutual friendships between other members within the karate club.

```r
plot(between, pch = 19,
     main = "Distribution of Betweeness Scores Amoung Members of Karate Club",
     ylab = "Betweeness Score", xlab = "Member Number")
```



Closeness

The closeness of a vertex is a measure of centrality in a network, it is calculated as the sum of the length of the shortest paths between the vertex and all other vertices in the graph. Thus the more central a vertex is, the closer it is to all other vertices. Ones with a high closeness score have the shortest distances to all other vertices. With respect to the karate club, these are the members with the most strong connection. In other words these are the people with friendships with the most amount of other members and through the smallest amount of other members, they have more direct friendships.

```
closeness_scores = closeness(karate_graph)
closeness_scores
```

```
##           1           2           3           4           5           6
## 0.016949153 0.012987013 0.016949153 0.014084507 0.011363636 0.011494253
##           7           8           9          10          11          12
## 0.011494253 0.013333333 0.015625000 0.013157895 0.011363636 0.010989011
##          13          14          15          16          17          18
## 0.011235955 0.015625000 0.011235955 0.011235955 0.008547009 0.011363636
##          19          20          21          22          23          24
## 0.011235955 0.015151515 0.011235955 0.011363636 0.011235955 0.011904762
##          25          26          27          28          29          30
## 0.011363636 0.011235955 0.010989011 0.013888889 0.013698630 0.011627907
##          31          32          33          34
## 0.013888889 0.016129032 0.015625000 0.016666667
```

```
max(closeness_scores)
```

```
## [1] 0.01694915
```

We see that member 1, as expected (the most central member to the network) has the highest closeness score.

PageRank Algorithim

The page rank algorithm concept is a way in which a social network vertex can be given an "importance score". With respect to our karate club social network each member is assigned an "importance score" in which we display below. The sum of all of the scores is 1, which is also show below.

```
page_rank = page_rank(karate_graph)
page_rank$vector
```

```
##           1           2           3           4           5           6
## 0.093273946 0.048436267 0.057820050 0.036473234 0.022267776 0.029464867
##           7           8           9          10          11          12
## 0.029464867 0.024925387 0.030084309 0.014406392 0.022267776 0.009697288
##          13          14          15          16          17          18
## 0.014864330 0.030005311 0.014600978 0.014600978 0.016934333 0.014843642
##          19          20          21          22          23          24
## 0.014600978 0.019923565 0.014600978 0.014843642 0.014600978 0.031670875
##          25          26          27          28          29          30
## 0.021176344 0.021101217 0.015102446 0.025790405 0.019711832 0.026403566
##          31          32          33          34
## 0.024861664 0.037450163 0.072131153 0.101598464
```

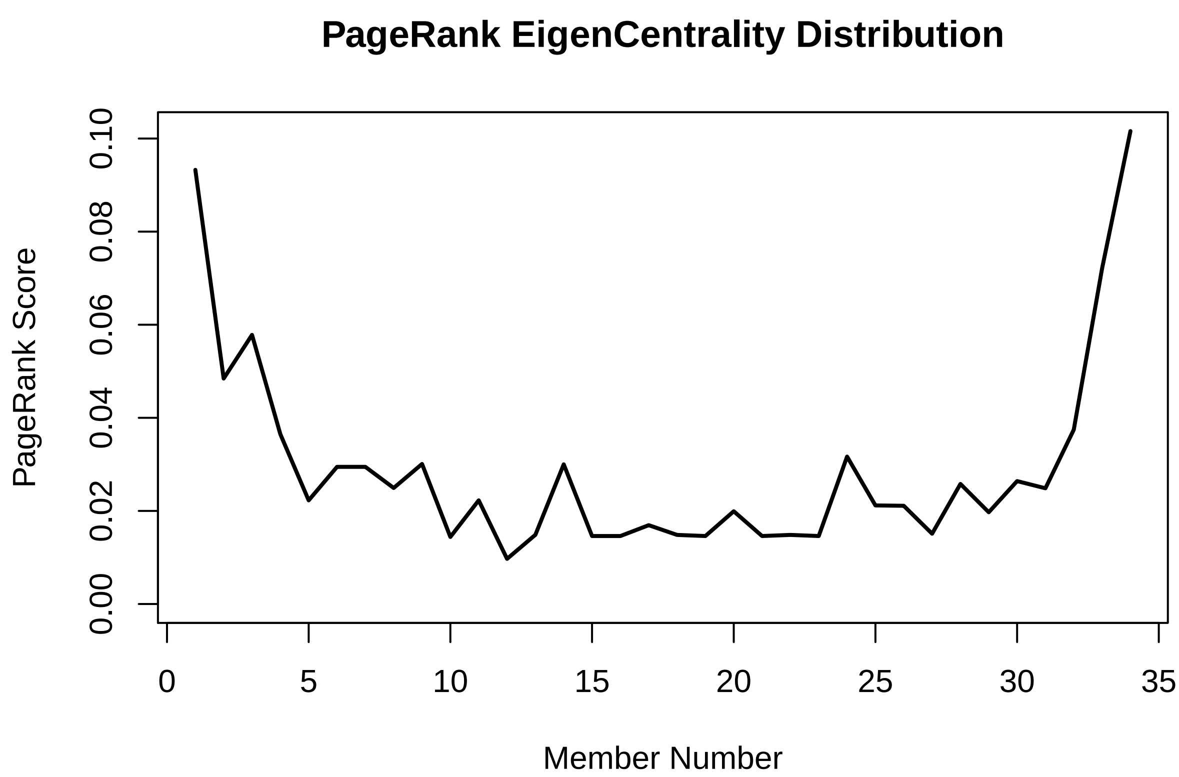
```
sum(page_rank$vector)
```

```
## [1] 1
```

```
max(page_rank$vector)
```

```
## [1] 0.1015985
```

```
y_max = max(page_rank$vector)
plot(page_rank$vector, type = "l", lwd = 2, col = "black",
     ylim = c(0,y_max), xlab = "Member Number",
     ylab = "PageRank Score",
     main = "PageRank EigenCentrality Distribution")
```



From the following distribution we can see that members 1 and 34 have the highest pageRank score. We can say that those members are important in the network with regards to centrality. Members 1 to 5 and 30 to 34 have the highest pageRank scores.

Dijkstra's Algorithim

Another final interesting way to see the paths between members of the karate club is to use Dijkstra's algorithm. We can see the friendship links between members of the group.

How is member 1 friends with member 25?

```r
one_to_twofive = shortest_paths(karate_graph, from = V(karate_graph)[1],
                                to = V(karate_graph)[25], weights = NULL)
one_to_twofive$vpath
```

```
## [[1]]
## + 3/34 vertices, named, from 1cca5ad:
## [1] 1  32 25
```

Member 1 is mutual friends with member 25 through member 32.

How is member 1 friends with member 27?

```r
one_to_twoseven = shortest_paths(karate_graph, from = V(karate_graph)[1],
                                 to = V(karate_graph)[27], weights = NULL)
one_to_twoseven$vpath
```

```
## [[1]]
## + 4/34 vertices, named, from 1cca5ad:
## [1] 1  9  34 27
```

Member 1 is friends with member 27 through members 9 and 34.

With the shortest paths function in R, we can replace the from and to parameters to get the links of frindships in the karate club between any two vertices in the social network.