Assignment 5

CS 432

Spring 2017

Michelle Graham

Question 1

The “igraph” package in RStudio is very useful for analyzing the data from Zachary’s Karate Club. Before I began, I came across an excellent article about the study found here:

https://milesott.com/2016/08/26/karate-club-network-club/

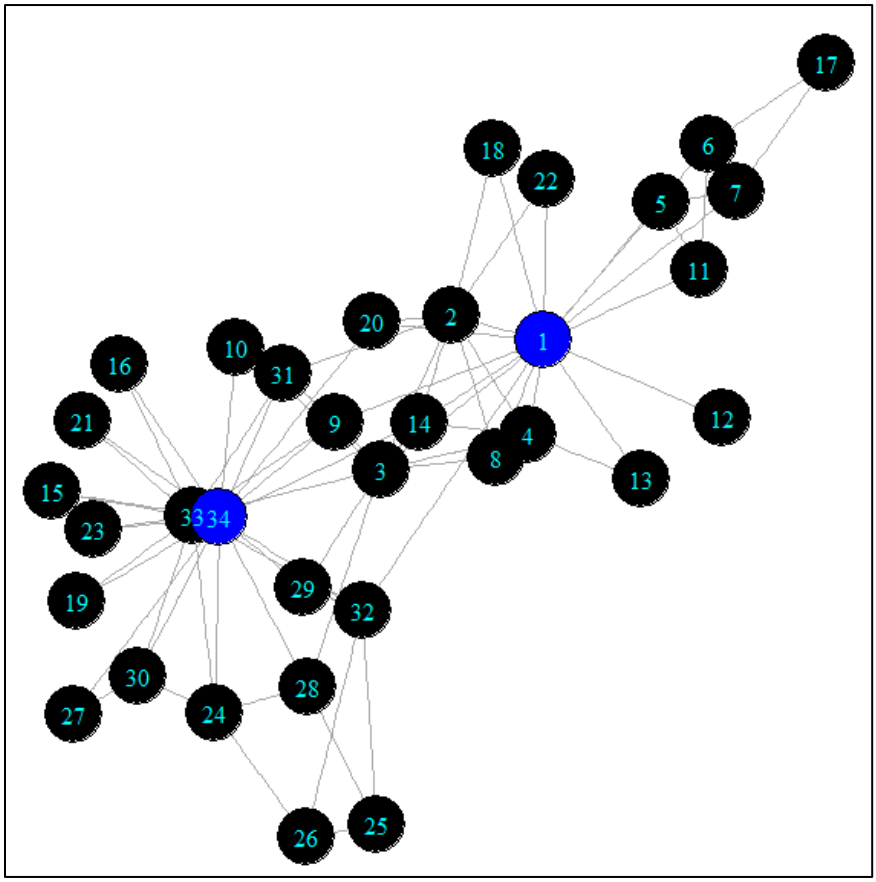
It provided a link to a dataset in gml format. I used this as a starting point for my analysis. To begin, I read the data into a graph in RStudio. Then, I created a variable to house the edge betweenness. I used most of the code in this **example**:

```
# Now we have the merges/splits and we need to calculate the modularity
# for each merge for this we'll use a function that for each edge
# removed will create a second graph, check for its membership and use
# that membership to calculate the modularity
mods <- sapply(0:ecount(g), function(i){
  g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])
  cl <- clusters(g2)$membership
# March 13, 2014 - compute modularity on the original graph g
# (Thank you to Augustin Luna for detecting this typo) and not on the induced one g2.
  modularity(g,cl)
})
```

Before the Split:

```
library(igraph)

#undirected graph
g <- read.graph("C:/CS432/A5/karate.gml", format="gml")

#find the edge betweenness
ebc <-edge.betweenness.community(g)

#color corresponds with membership
#V(g)$color <- ebc$membership + 5

V(g)$color[1] <- "blue"
V(g)$color[2:33] <- "black"
V(g)$color[34] <- "blue"

plot(g,  layout = layout.fruchterman.reingold, vertex.label.color = "cyan", vertex.size = 15)
```

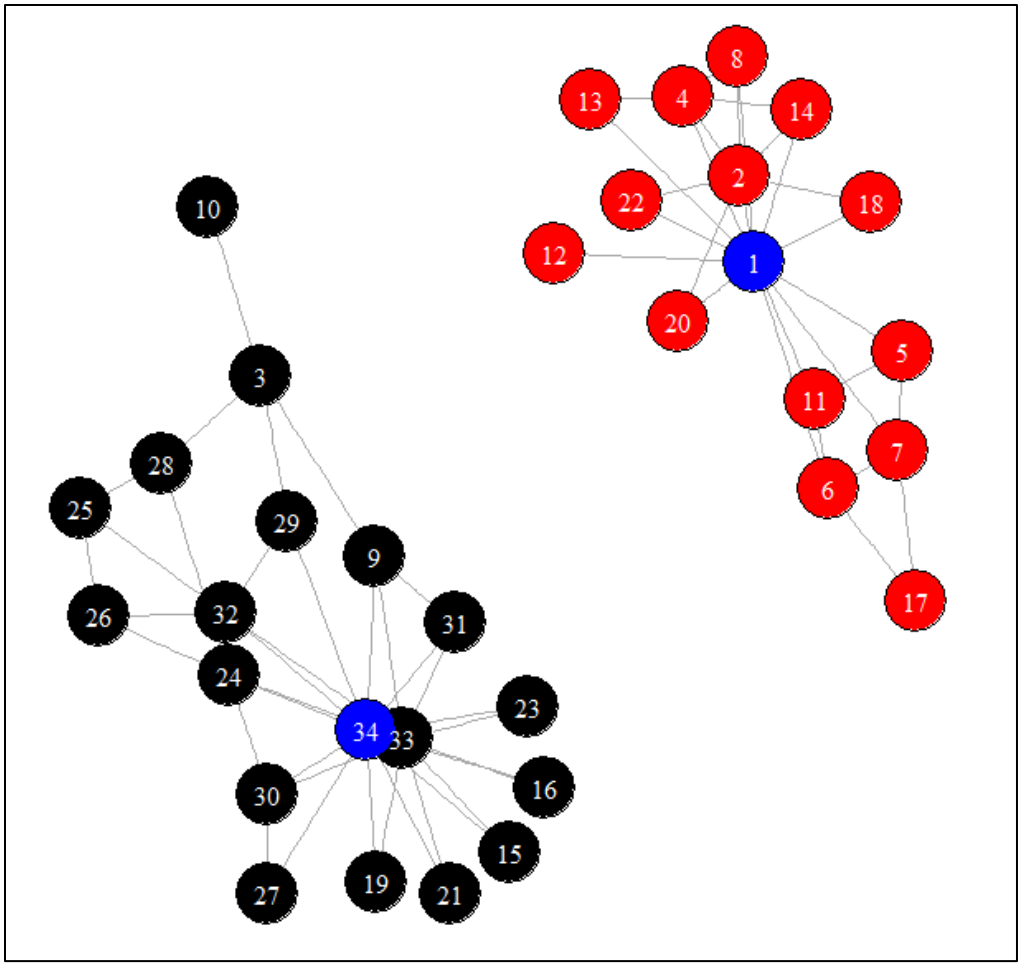
Note: I have indicated which nodes belong to “Mr. Hi” (1) and “John A.” (34).



Two Clusters:

```
library(igraph)

#undirected graph
g <- read.graph("C:/CS432/A5/karate.gml", format="gml")

#find the edge betweenness
ebc <-edge.betweenness.community(g)
```

```
v(g)$color[1] <- "blue"
v(g)$color[22] <- "red"
v(g)$color[20] <- "red"
v(g)$color[8] <- "red"
v(g)$color[2] <- "red"
v(g)$color[4] <- "red"
v(g)$color[11] <- "red"
v(g)$color[14] <- "red"
v(g)$color[13] <- "red"
v(g)$color[18] <- "red"
v(g)$color[12] <- "red"
v(g)$color[5] <- "red"
v(g)$color[6] <- "red"
v(g)$color[7] <- "red"
v(g)$color[17] <- "red"
v(g)$color[34] <- "blue"

#plot(g,  layout = layout.fruchterman.reingold, vertex.label.color = "cyan", vertex.size = 15)


#delete edges to find the remaining components that make up the communities
mods <- sapply(
0:ecount(g), function(i){
g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])
cl <- clusters(g2)$membership
if(no.clusters(g2)==2){
    plot(g2, layout = layout.fruchterman.reingold, vertex.label.color = "white", vertex.size = 15)
    }
  }
```



Results:

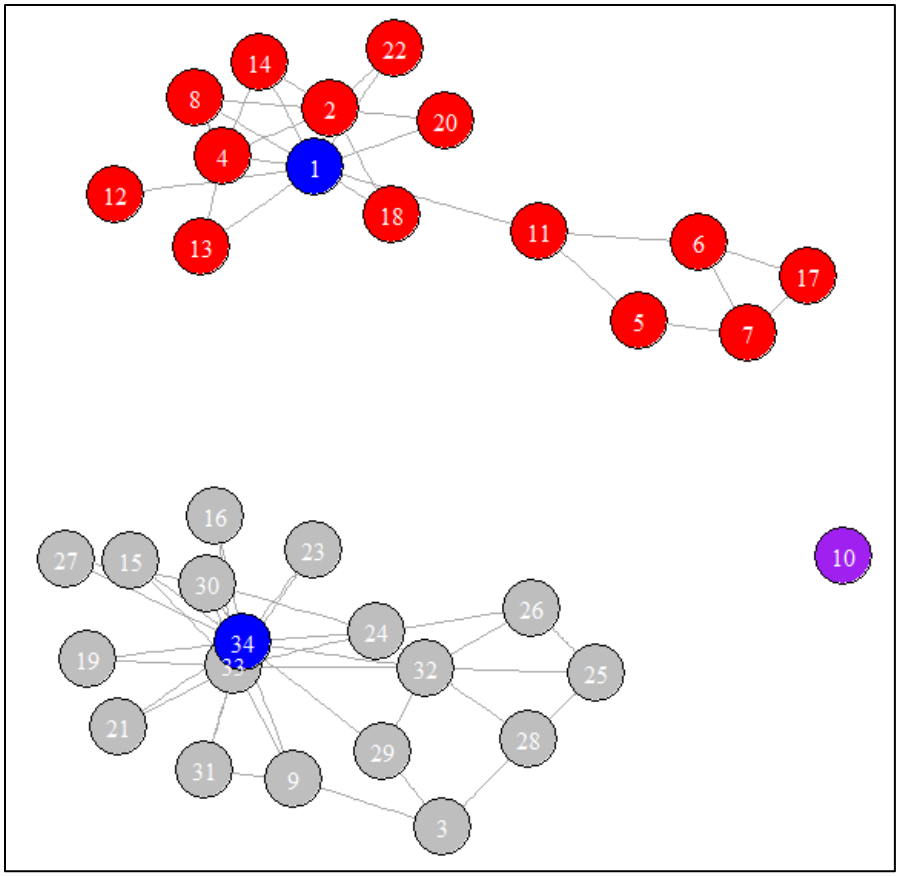
Group 1 consists of 15 members including- 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 20, and 22. Group 2 consists of 19 members including- 3, 9, 10, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, and 34. This is on par with Zachary's prediction- 97% of it was accurate, as the only deficiency results from individual 3.

Question 2

In order to obtain a cluster level of 3, 4, and 5, the “no.clusters” was set to each corresponding number. Some slight changes were made to help distinguish between each cluster for readability.

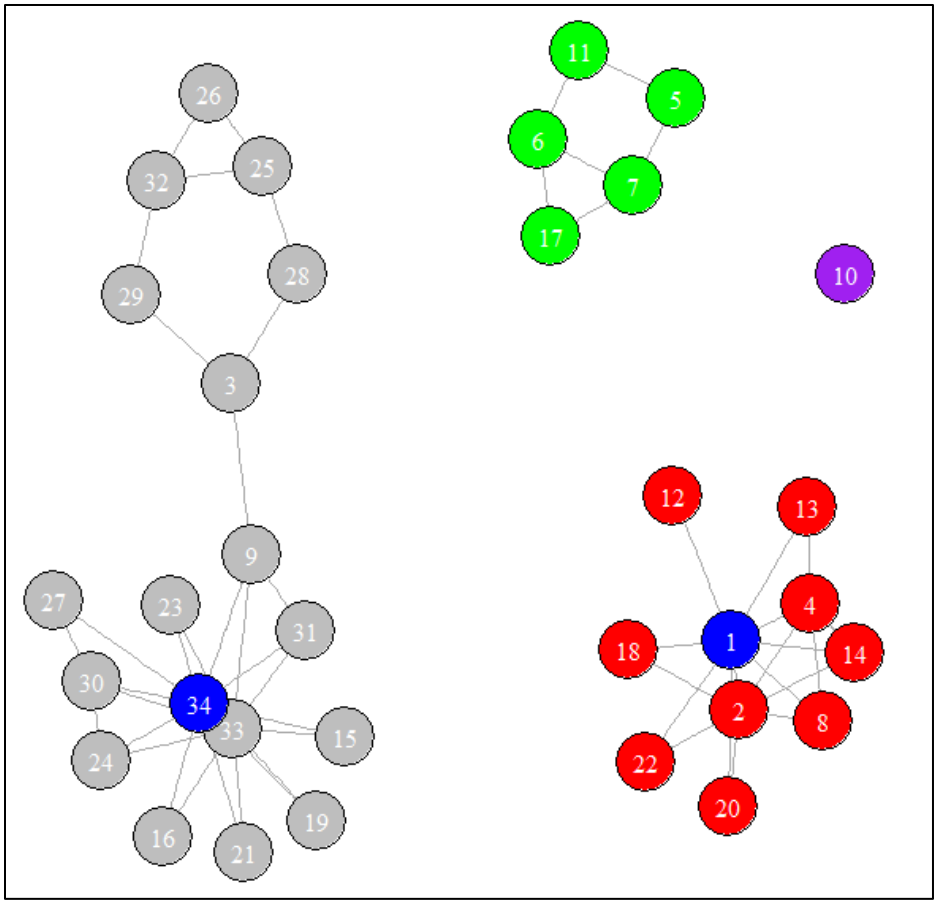
Three Clusters:

```
library(igraph)

#undirected graph
g <- read.graph("C:/CS432/A5/karate.gml", format="gml")

#find the edge betweenness
ebc <-edge.betweenness.community(g)

#color corresponds with membership; for 5 clusters
#V(g)$color <- ebc$membership + 5

V(g)$color <- "gray"
V(g)$color[1] <- "blue"
V(g)$color[34] <- "blue"


# for two clusters
# V(g)$color[22] <- "red"
# V(g)$color[20] <- "red"
# V(g)$color[8] <- "red"
# V(g)$color[2] <- "red"
# V(g)$color[4] <- "red"
# V(g)$color[11] <- "red"
# V(g)$color[14] <- "red"
# V(g)$color[13] <- "red"
# V(g)$color[18] <- "red"
# V(g)$color[12] <- "red"
# V(g)$color[5] <- "red"
# V(g)$color[6] <- "red"
# V(g)$color[7] <- "red"
# V(g)$color[17] <- "red"

# for three clusters
V(g)$color[10] <- "purple"

V(g)$color[2] <- "red"
V(g)$color[4] <- "red"
V(g)$color[5:8] <- "red"
V(g)$color[11:14] <- "red"
V(g)$color[17:18] <- "red"
V(g)$color[20] <- "red"
V(g)$color[22] <- "red"



#plot(g,  layout = layout.fruchterman.reingold, vertex.label.color = "cyan", vertex.size = 15)


#delete edges to find the remaining components that make up the communities
mods <- sapply(
0:ecount(g), function(i){
g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])
cl <- clusters(g2)$membership
if(no.clusters(g2)==3){
    plot(g2, layout = layout.fruchterman.reingold, vertex.label.color = "white", vertex.size = 15)
    }
  }
)
```


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

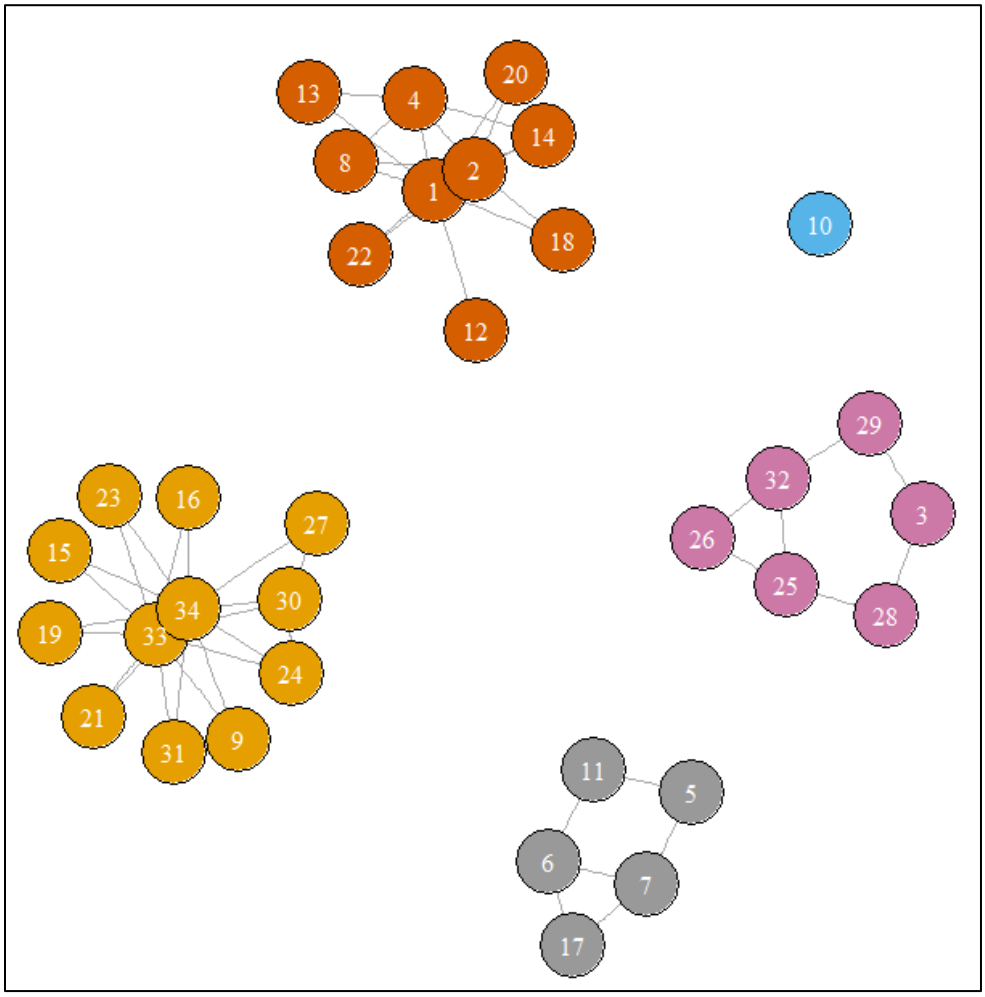
Four Clusters:

```
library(igraph)

#undirected graph
g <- read.graph("C:/CS432/A5/karate.gml", format="gml")

#find the edge betweenness
ebc <-edge.betweenness.community(g)

#color corresponds with membership; for 5 clusters
#V(g)$color <- ebc$membership + 5

V(g)$color <- "gray"
V(g)$color[1] <- "blue"
V(g)$color[34] <- "blue"

# for three clusters
V(g)$color[10] <- "purple"

V(g)$color[2] <- "red"
V(g)$color[4] <- "red"
V(g)$color[8] <- "red"
V(g)$color[12:14] <- "red"
V(g)$color[18] <- "red"
V(g)$color[20] <- "red"
V(g)$color[22] <- "red"

V(g)$color[5:7] <- "green"
V(g)$color[11] <- "green"
V(g)$color[17] <- "green"



#plot(g,  layout = layout.fruchterman.reingold, vertex.label.color = "cyan", vertex.size = 15)


#delete edges to find the remaining components that make up the communities
mods <- sapply(
0:ecount(g), function(i){
g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])
cl <- clusters(g2)$membership
if(no.clusters(g2)==4){
    plot(g2, layout = layout.fruchterman.reingold, vertex.label.color = "white", vertex.size = 15)
    }
  }
)
```


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

Five Clusters:

```
library(igraph)

#undirected graph
g <- read.graph("C:/CS432/A5/karate.gml", format="gml")

#find the edge betweenness
ebc <-edge.betweenness.community(g)

#color correspongs with membership
V(g)$color <- ebc$membership + 5

#delete edges to find the remaining components that make up the communities
mods <- sapply(
0:ecount(g), function(i){
g2 <- delete.edges(g, ebc$removed.edges[seq(length=i)])
cl <- clusters(g2)$membership
if(no.clusters(g2)==5){
    plot(g2, layout = layout.fruchterman.reingold, vertex.label.color = "white", vertex.size = 15)
    }
  }
)
```



Resources

http://aris.ss.uci.edu/~lin/76.pdf

http://www.sixhat.net/finding-communities-in-networks-with-r-and-igraph.html

http://stackoverflow.com/questions/9876267/r-igraph-community-detection-edge-betweenness-method-count-list-members-of-e

https://en.wikipedia.org/wiki/Girvan%E2%80%93Newman_algorithm

https://www.r-bloggers.com/going-viral-with-rs-igraph-package/

http://www.shizukalab.com/toolkits/sna/plotting-networks-pt-2

https://www.r-bloggers.com/going-viral-with-rs-igraph-package/

http://rstudio-pubs-static.s3.amazonaws.com/5014_4e3001382f7442629c0760f373cdadd4.html

http://stackoverflow.com/questions/31992685/colour-specific-node-in-igraph

http://stackoverflow.com/questions/14164887/change-the-font-and-colour-of-the-title-of-the-igraph