Experiment No.6
Social Network Analysis using R (for example: Community Detection Algorithm)
Date of Performance:04/09/2023
Date of Submission:11/09/2023

Aim: Social Network Analysis using R (for example: Community Detection Algorithm)

Theory:
Online social platforms have enabled people around the world to interact with each other and build relationships with others they share common interests with. This can be observed in real life — naturally, we tend to develop and maintain relationships with others that are similar to us. People with similar interests tend to gravitate towards each other and become associated in communities — clusters or groups of people that share similar traits with each other. Since people tend to cluster with others similar to them, we can use community detection to identify users with a high number of degrees (connections) and see how far their reach can travel in the network.

User Data Extraction — Since we are only interested in user data, we will only extract the following variables:

User_id — Yelp user ID; this is needed to make nodes and edges
Name — user's first name
Review count — the number of reviews user has written
Yelping since — date user joined Yelp
Friends — a list containing all of the user's friends by user_id
Fans — number of fans user has
Elite — number of years the user has Elite status
Average stars — user's average rating of all reviews written

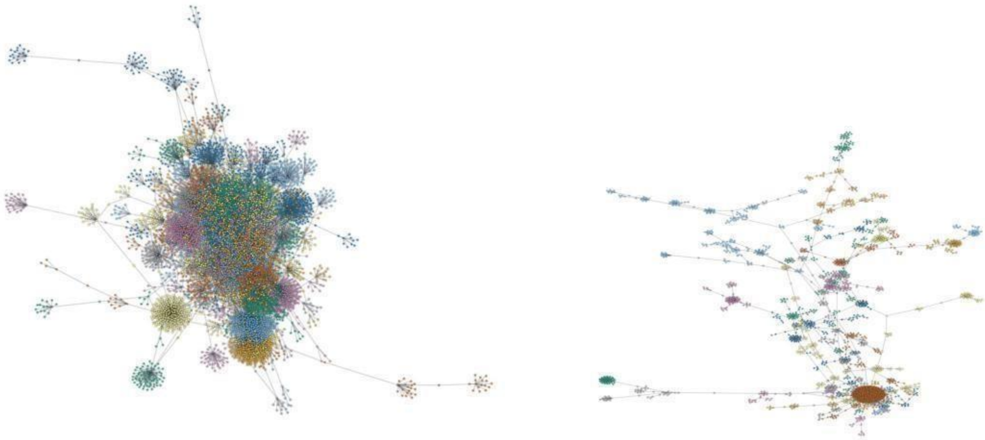
CODE:

```
#remove users with no friends

sample <- subset(user_df, friends != "None")

#make a subset; we only need to retain data of users with some social activity sub

<- subset(sample, year == 2005 & review_count >= 2 & no_of_friends >= 2)

#make links (nodes and edges) sample_friends <- sub %>%

select(user_id,          friends)          sample_users          <-

strsplit(sample_friends$friends, split = ",")

sample_dat <- data.frame(user_id = rep(sample_friends$user_id, sapply(sample_users, length)),
friends = unlist(sample_users))

#network is still too big, take a random sample of 100k nodes samp_net

<- sample_n(sample_dat, 100000)
```

```
#make network
network  <-  graph.data.frame(samp_net)
network_s <- simplify(network) net_deg
<-   degree(network_s)   all_degree   <-
degree(network, mode = 'all')
#graph user with max degrees
sub_all <- subcomponent(network_s, which(all_degree == max(all_degree)), 'all') g_sub
<- induced_subgraph(network_s, sub_all)
#communities
graph.com <- fastgreedy.community(as.undirected(g_sub))
V(g_sub)$color <- graph.com$membership + 1
#create pdf graph for high resolution (try zooming in!) pdf("communities2005.pdf",
10,10)
plot(g_sub,
    vertex.color                =
    V(g_sub)$color, vertex.size =
    1, vertex.label = NA,
    vertex.frame.color = adjustcolor("#41424c", alpha.f = 0.25),
    edge.arrow.size = 0.1, edge.color = adjustcolor("#41424c",
    alpha.f = 0.20), edge.width = 1.5, edge.arrow.mode=0,
    layout=layout_with_lgl,
    asp = 0.9,
    dpi=300
```

)

dev.off()

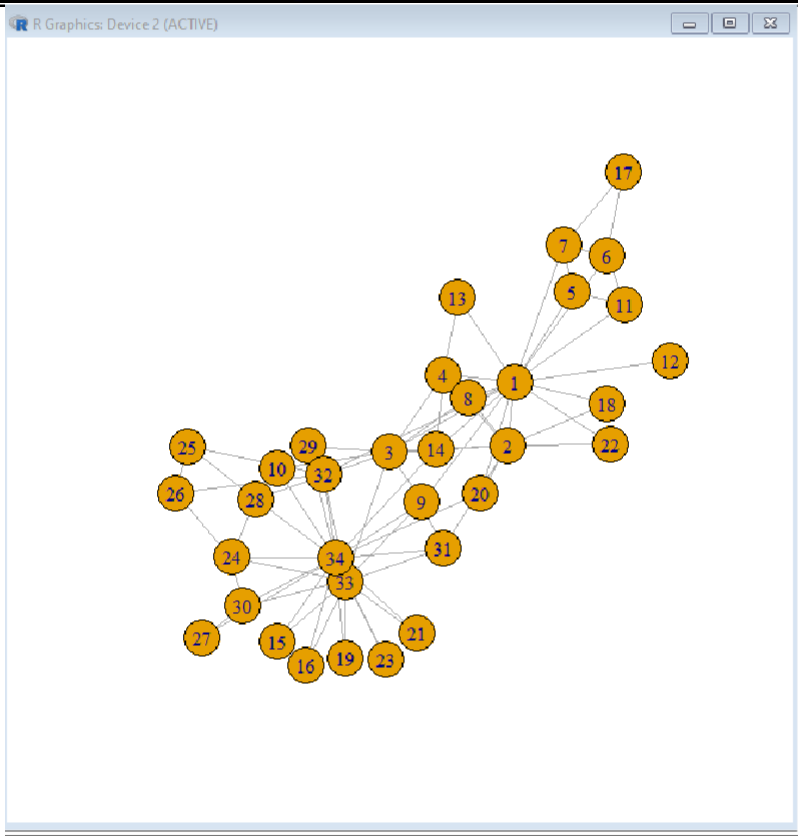


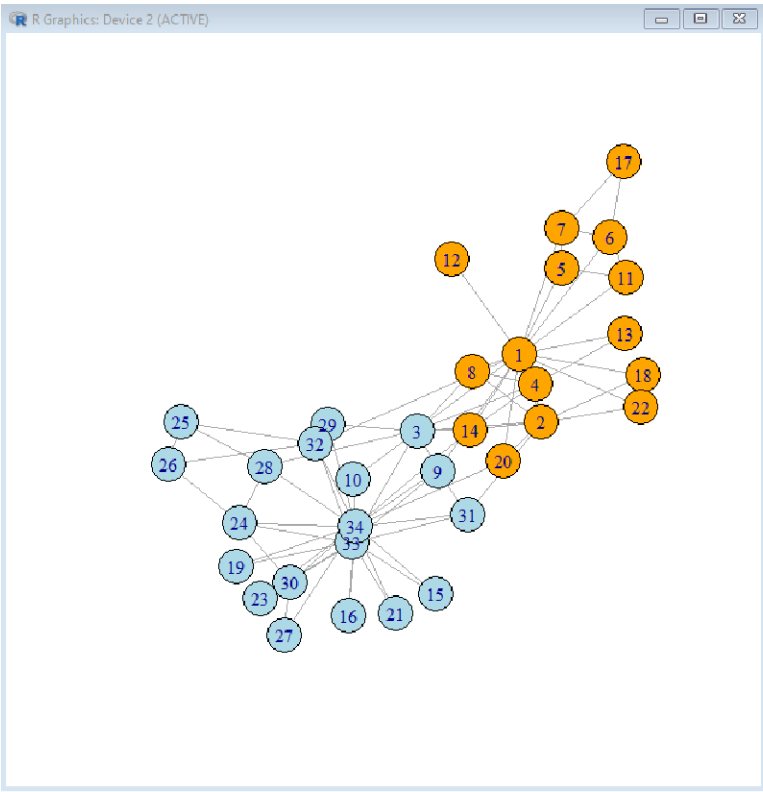
OUTPUT;

RGui (64-bit) - [C:\Users\admin\Desktop\CommunityDetection\algo.R - R Editor]

File Edit Packages Windows Help

```
library(igraph)

girvan <- function(G){
  c = decompose.graph(G)
  l = length(c)
  v <- vector()
  while(l==1){
    x <- E(G)
    y <- edge_betweenness(G)
    z <- which.max(y)
    edge <- x[z]
    a <- ends(G,z[1])[1]
    b <- ends(G,z[1])[2]
    v <- c(v,a,b)
    G <- delete_edges(G,edge)
    c = decompose.graph(G)
    l = length(c)
  }
  if(l==2){
    paths <- shortest.paths(G)
    for(i in 1:length(V(G))){
      if(paths[a,i]!=Inf){
        V(G)[i]$color = "lightblue"
      }
      else{
        V(G)[i]$color = "orange"
      }
    }
    G <- G + edge(v)
    plot(G)
  }
  return(c)
}


g <- read.graph("C:/Users/admin/Desktop/CommunityDetection/karate.gml",format = "gml")
plot(g)
c <- girvan(g)
```


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


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

CONCLUSION:

In our experiment on Social Network Analysis using R, focusing on Community Detection Algorithms, we discovered valuable insights into social network structures. We learned that selecting the right algorithm is crucial, as performance varies with network size and complexity. Visualizations like network graphs enhance interpretation. Social Network Analysis has practical applications in sociology, marketing, and epidemiology by revealing influential nodes and information spread. Future research could explore advanced algorithms and larger datasets. Overall, this experiment underscores the importance of Social Network Analysis in understanding complex social relationships and its potential to benefit various fields.