

**NAME: VEDASHREE BHALERAO**

**ROLL NO- 2213191**

**BATCH- B**

### **Big Data Analytics Experiment no. 09**

**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 everyone is 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

The Yelp data is very large, so it will take a very long time to extract data from the json file.

**Network Graph** — Let's make two graphs comparing users that joined in 2005 and 2015 using the igraph package in R. What will a difference in 10 years make?

It takes a very long time to make network graphs, so we will limit our subset to 100k nodes and create subgraphs of the user with the maximum number of degrees.

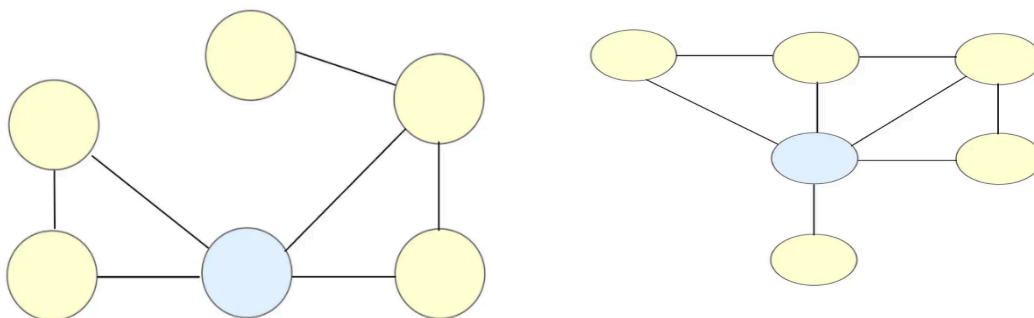
The graph on the left is more dense and this could be due to the fact that Yelpers that have been on the platform longer have had more time to build their reputation on Yelp and establish themselves in communities. On the other hand, Yelpers that joined in 2015 have less dense communities, less edges → less connections. An interesting insight from the 2015 community is the dense region of orange dots concentrated near the bottom of the network, implying that there is a large community of users that have similar traits.

From our subgraphs of communities, we can detect **cliques**:

We can use this to take a deeper look into a popular Yelper's network to visualize their sphere of influence.

The size of each node (user) indicates their connections (number of friends). Each edge (link) shows connections between nodes (user's friends). Michelle and Bryant appear twice because it shows that the *location* of their influence occurs in more than one group.

In the clique above, we found that Paige has the highest **betweenness centrality** — a measure of how many times a node (user) acts as a bridge between two nodes.



Bridges are important because they connect two different groups in a social network; they are useful in bridging the gap between different communities! An example of a bridge is someone that is able to communicate and interpret data to both tech and non-tech team members. Another example is when your friend introduces you to their new friend, with your friend acting as the bridge.

Conclusion: We Successfully studied and implemented Social Network Analysis using R.

## CODE:

```
install.packages("igraph")
```

```
install.packages("sna")
```

```
library(igraph)
```

```
library(network)
```

```
library(sna)
```

```
plot(make_full_graph(8, directed=FALSE))
```

```
Ring_Graph <- make_ring(12, directed = FALSE, mutual =FALSE, circular = TRUE)
```

```
# try for False
```

```
plot(Ring_Graph)
```

```
Star_Graph <- make_star(12, center = 4)
```

```
plot(Star_Graph)
```

```
gnp_Graph <- sample_gnp(20, 0.5, directed = FALSE, loops =FALSE)
```

```
plot(gnp_Graph)
```

```
gnp_Graph1 <- sample_gnp(7, 0.4, directed = FALSE, loops = FALSE)
```

```
plot(gnp_Graph1)
```

```
node_degrees <- degree(gnp_Graph)
```

```
print(node_degrees)
```

```
sample_graph <- sample_gnp(10, 0.3, directed = FALSE)
```

```
plot(sample_graph)
```

```
sample_density <- edge_density(sample_graph, loops = FALSE)
```

```
sample_density
```

```
sample_graph <- sample_gnp(20, 0.3, directed = FALSE, loops= FALSE)
```

```
plot(sample_graph)
```

```
clique_num(sample_graph)
```

```
components(sample_graph)
```

```
setwd("D:\\MIT ADT\\LY - Sem 1\\BDA Lab\\Amreen Mam\\Assign 9")
```

```
data <- read.csv("socialnetworkdata.csv")
```

```
y <- data.frame(data$first, data$second)
```

```
net <- graph.data.frame(y, directed=T)
```

```
V(net)
```

```
E(net)
```

```
plot(net)
```

```
hist(degree(net), col='purple', main = "histogram of node degree",ylab = 'freq', xlab =  
'vertices degree')
```

```
set.seed(222)
```

```
plot(net,
```

```
    vertex.color = 'cyan',
```

```
    vertex.size = 2,
```

```
    edge.arrow.size = 0.1,
```

```
    vertex.label.cex = 0.8)
```

```
plot(net,
```

```
    vertex.color = rainbow(vcount(net)), # Use the number of vertices to generate colors
```

```
    vertex.size = degree(net) * 0.4,    # Calculate vertex degrees and scale their size
```

```
    edge.arrow.size = 0.1,              # Keep arrow size for edges
```

```
    layout = layout.fruchterman.reingold)
```

```
hs <- hub_score(net)$vector
```

```
hs
```

```
as <- authority_score(net)$vector
```

```
as
```

```
set.seed(123)

plot(net,
      vertex.size=hs*30,
      main = 'Hubs',
      vertex.color = rainbow(52),
      edge.arrow.size=0.1,
      layout = layout.kamada.kawai)

undirected_net <- as.undirected(net, mode = "collapse")

community <- cluster_louvain(undirected_net)

plot(undirected_net,
     vertex.color = membership(community), # Color vertices by community membership
     vertex.size = degree(undirected_net) * 0.4, # Size vertices by degree
     edge.arrow.size = 0.1, # Edge arrow size
     main = "Community Detection using Louvain Method")
```

**OUTPUT:**

