Assignment No. 9

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

Theory:

Social Network Analysis (SNA) involves studying the relationships and interactions among entities within a network, which can be individuals, organizations, or even websites. In this practical, **R** is used to perform SNA, particularly focusing on **Community Detection** algorithms, which identify groups of nodes (vertices) that are more densely connected to each other than to the rest of the network.

The analysis begins by generating different graph types, such as **Ring Graph**, **Star Graph**, and **Random Graph (GNP)**, using the igraph library. These graphs simulate different types of networks and allow us to explore their structural properties, such as node degrees, edge density, and cliques (subgraphs where every node is connected to every other node). The degree of a node indicates the number of edges (connections) it has, and clique analysis helps to identify closely-knit groups within the network.

Community detection, specifically the **Louvain method**, is employed to uncover subgroups or communities within the network. The **hub** and **authority** scores, calculated using the **HITS (Hyperlink-Induced Topic Search)** algorithm, are used to identify influential nodes (hubs) and authoritative nodes (authorities) in the network.

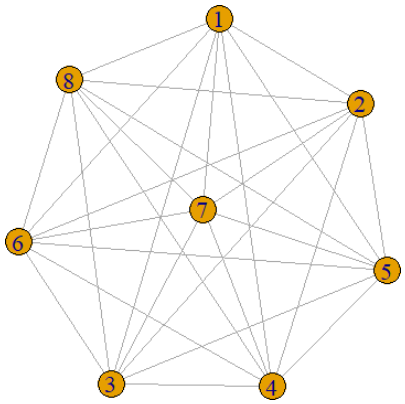
The dataset socialnetworkdata.csv is read into R, which represents relationships between pairs of entities (e.g., two people connected in a social network). Using this data, a graph is created to visualize connections, and community detection is performed to reveal subgroups or clusters within the network.

Code and output:

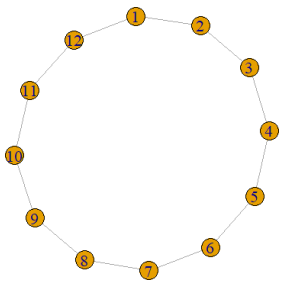
```
install.packages("igraph")
install.packages("sna")
```

```
library(igraph)
library(network)
```

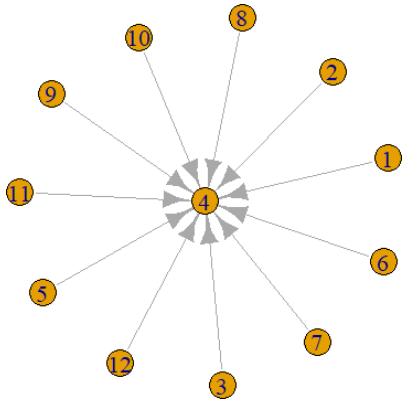
```
library(sna)

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



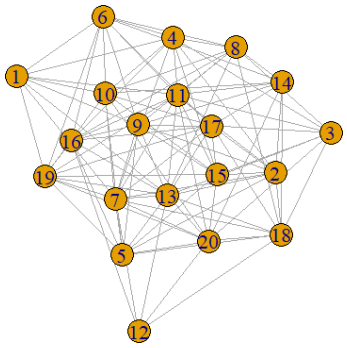

```
Ring_Graph <- make_ring(12, directed = FALSE, mutual =FALSE, circular = TRUE)
# try for False
plot(Ring_Graph)
```



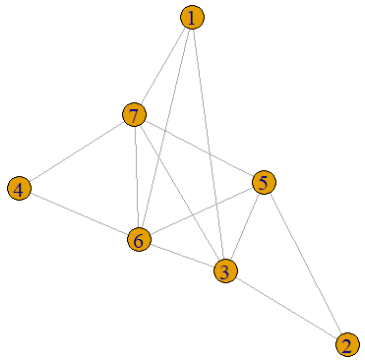

```
Star_Graph <- make_star(12, center = 4)
plot(Star_Graph)
```




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




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



degree(gnp_Graph)

betweenness(gnp_Graph)

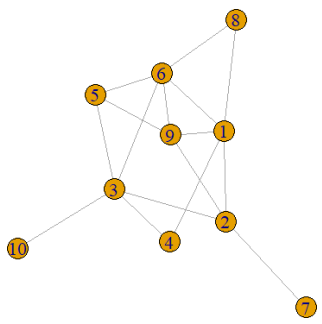
```
> degree(gnp_Graph)
[1] 3 2 5 2 4 5 5
> betweenness(gnp_Graph)
[1] 0.000000 0.000000 2.833333 0.000000 1.500000 2.333333 2.333333
```

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

plot(sample_graph)

sample_density <- edge_density(sample_graph, loops = FALSE)

sample_density

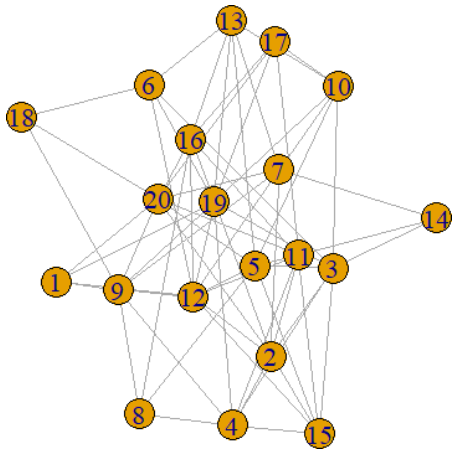


```
> sample_density <- edge_density(sample_graph, loops = FALSE)
> sample_density
[1] 0.3555556
```

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

plot(sample_graph)

clique_num(sample_graph)




```
> components(sample_graph)
$membership
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$csize
[1] 20

$no
[1] 1
```

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

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

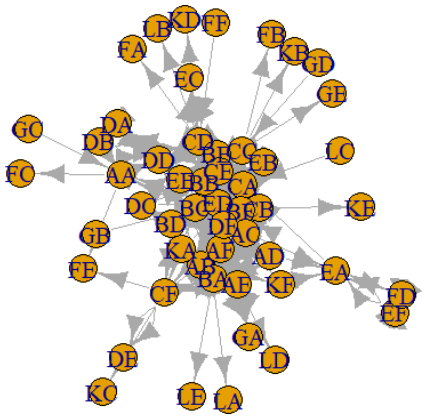
y <- data.frame(data$first, data$second)

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

V(net)

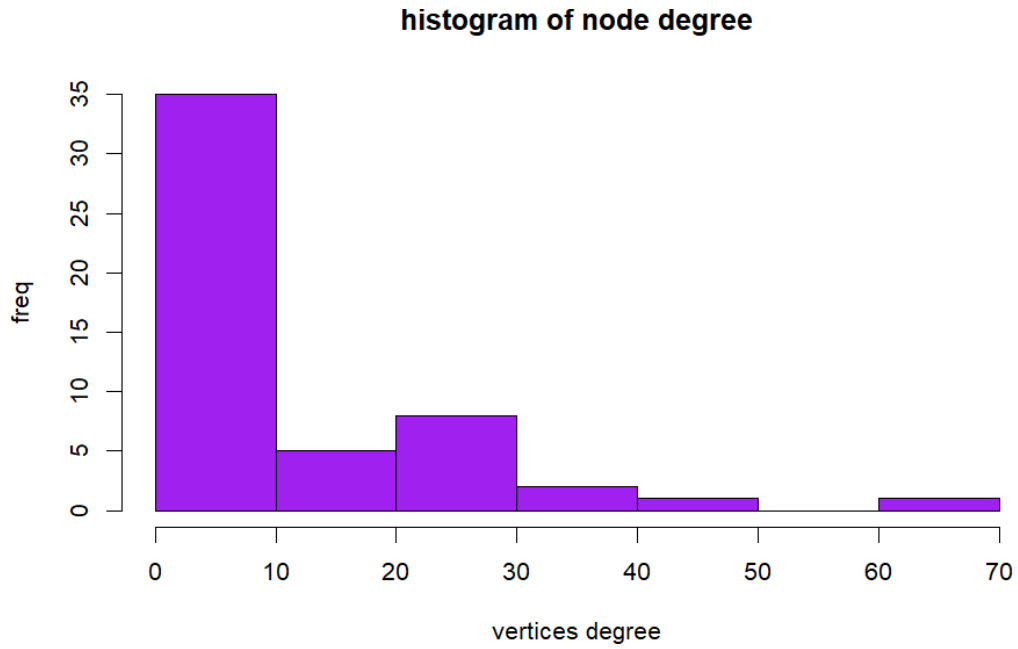
E(net)

plot(net)

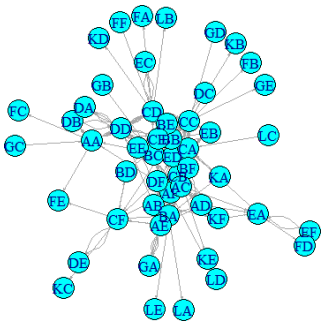


```
> V(net)
+ 52/52 vertices, named, from dc9ea19:
 [1] AA AB AF DD CD BA CB CC BC ED AE CA EB BF BB AC DC BD DB CF DF BE EA CE EE EF
[27] FF FD GB GC GD AD KA KF LC DA EC FA FB DE FC FE GA GE KB KC KD KE LB LA LD LE
> E(net)
+ 290/290 edges from dc9ea19 (vertex names):
  [1] AA->DD AB->DD AF->BA DD->DA CD->EC DD->CE CD->FA CD->CC BA->AF CB->CA CC->CA
 [12] CD->CA BC->CA DD->DA ED->AD AE->AC AB->BA CD->EC CA->CC EB->CC BF->CE BB->CD
 [23] AC->AE CC->FB DC->BB BD->CF DB->DA DD->DA DB->DD BC->AF CF->DE DF->BF CB->CA
 [34] BE->CA EA->CA CB->CA CB->CA CC->CA CD->CA BC->CA BF->CA CE->CA AC->AD BD->BE
 [45] AE->DF CB->DF AC->DF AA->DD AA->DD AA->DD CD->DD AA->DD EE->DD CD->DD DB->AA
 [56] AA->FC BE->CC EF->FD CF->FE BB->DD CD->DD BA->AB CD->EC BE->EE CE->CC CD->CC
 [67] ED->CC BB->CC BE->CE DD->CE AC->CD ED->CD FF->CD AC->CD DD->CD DD->CD AE->GA
 [78] AE->GA AE->GA AE->GA BA->ED BE->ED EB->ED CD->ED FD->EF FD->EF CD->BB BF->BB
 [89] BC->BB BB->CF AE->AC DD->DA BE->CA BE->CA CB->CA CB->CA CC->CA BE->CC BE->CC
[100] DB->DD BE->CD ED->CD ED->CD BB->CD BC->CD ED->AF BF->AF CF->AF BC->AF GB->BC
+ ... omitted several edges
> plot(net)
```

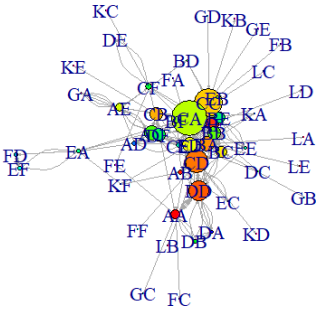
hist(degree(net), col='purple', main = "histogram of node degree",ylab = 'freq', xlab = 'vertices degree')



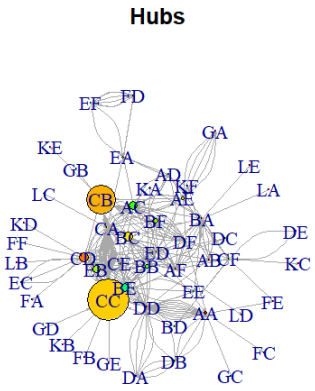

```
set.seed(222)
plot(net,
     vertex.color = 'cyan',
     vertext.size = 2,
     edge.arrow.size = 0.1,
     vertex.label.cex = 0.8)
```



```
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)$vector
as <- authority.score(net)$veetor
set.seed(123)
plot(net,     vertex.size=hs*30,     main = 'Hubs',     vertex.color = rainbow(52),
ge.arrow.size=0.1,     layout = layout.kamada.kawai)
```



Conclusion:

This practical showcases the use of R for Social Network Analysis, including community detection and network visualization. The Louvain method effectively identified communities, and hub/authority scores highlighted influential nodes within the network.