HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

INDEX

SR. NO.	Practicals	Date	Signature
1.	Write a program to compute the following for a given a network:(i) number of edges, (ii) number of nodes; (iii) degree of node; (iv)node with lowest degree; (v) the adjacency list; (vi) matrix of the graph.	03/07/24	
2.	Perform following tasks: (i) View data collection forms and/or import one-mode/two-mode datasets; (ii) Basic Networks matrices transformations	10/07/24	
3.	Compute the following node level measures: (i) Density; (ii) Degree; (iii) Reciprocity; (iv) Transitivity; (v) Centralization; (vi) Clustering.	28/07/24	
4.	For a given network find the following: (i)Length of the shortest path from a given node to another node; (ii) the density of the graph; (iii) Draw egocentric network of node G with chosen configuration parameters.	1/08/24	
5.	Write a program to distinguish between a network as a matrix, a network as an edge list, and a network as a sociogram (or “network graph”) using 3 distinct networks representatives of each.	12/08/24	
6.	Write a program to exhibit structural equivalence, automorphic equivalence, and regular equivalence from a network.	26/08/24	
7.	Perform SVD analysis of a network.	3/09/24	
8.	Displaying Bipartite network in the graph format.	17/09/24	
9.	Hamming distance.	05/10/24	

HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 1	
Aim: Write a program to compute the following for a given a network:(i) number of edges, (ii) number of nodes; (iii) degree of node; (iv)node with lowest degree; (v) the adjacency list; (vi) matrix of the graph.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 03/07/24	Sign:

Code:

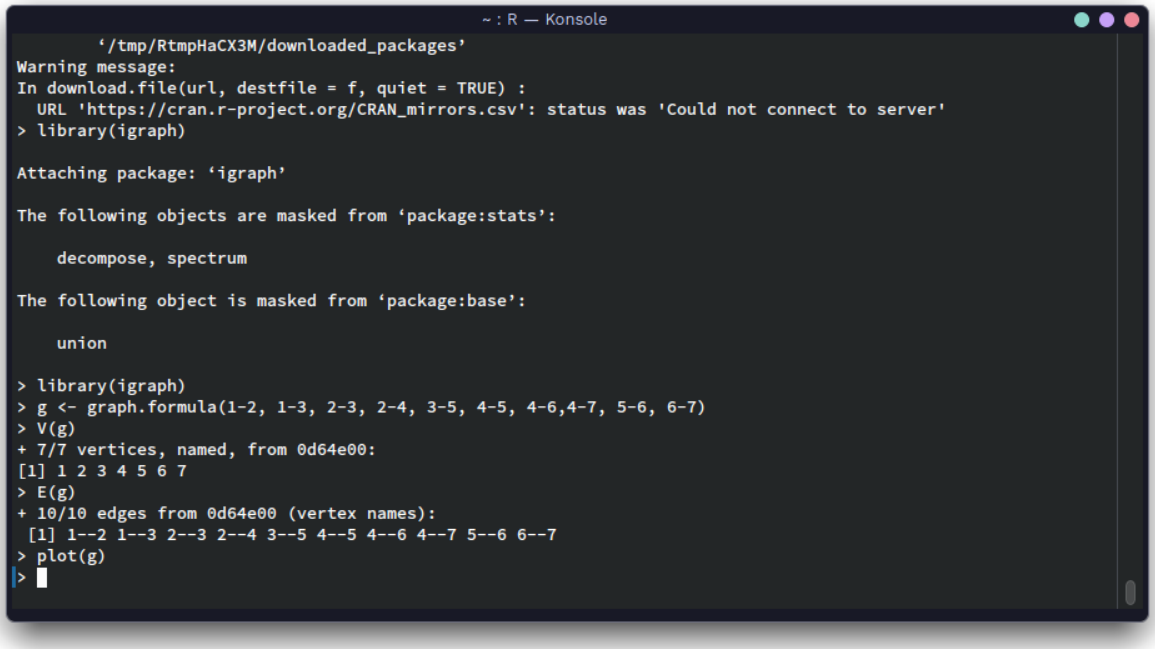
```
>library(igraph)
>g <- graph.formula(1-2, 1-3, 2-3, 2-4, 3-5, 4-5, 4-6, 4-7, 5-6, 6-7)
```

Name of Edges & Nodes

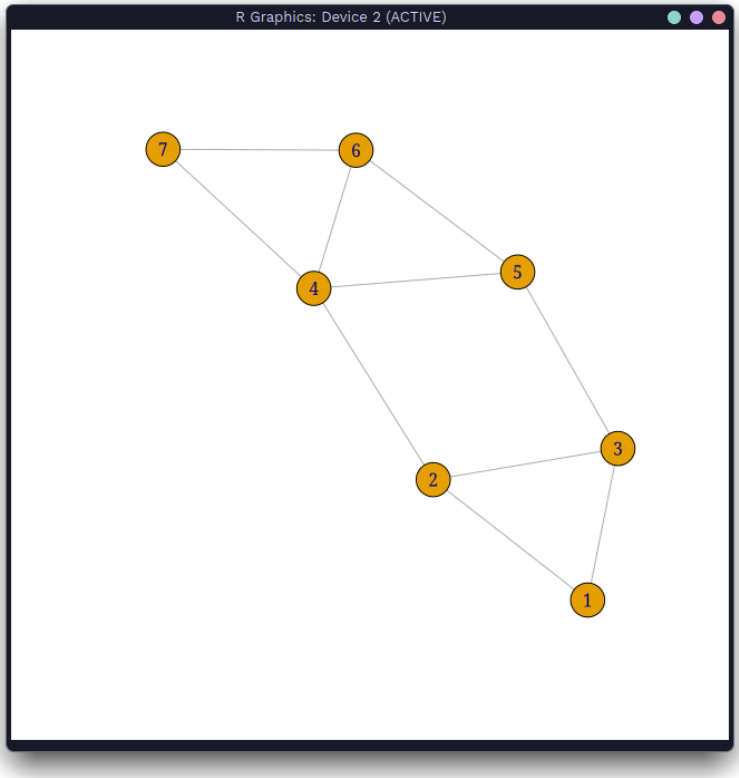
```
> V(g)
> E(g)
```

Plotting the graph

```
> plot(g)
```

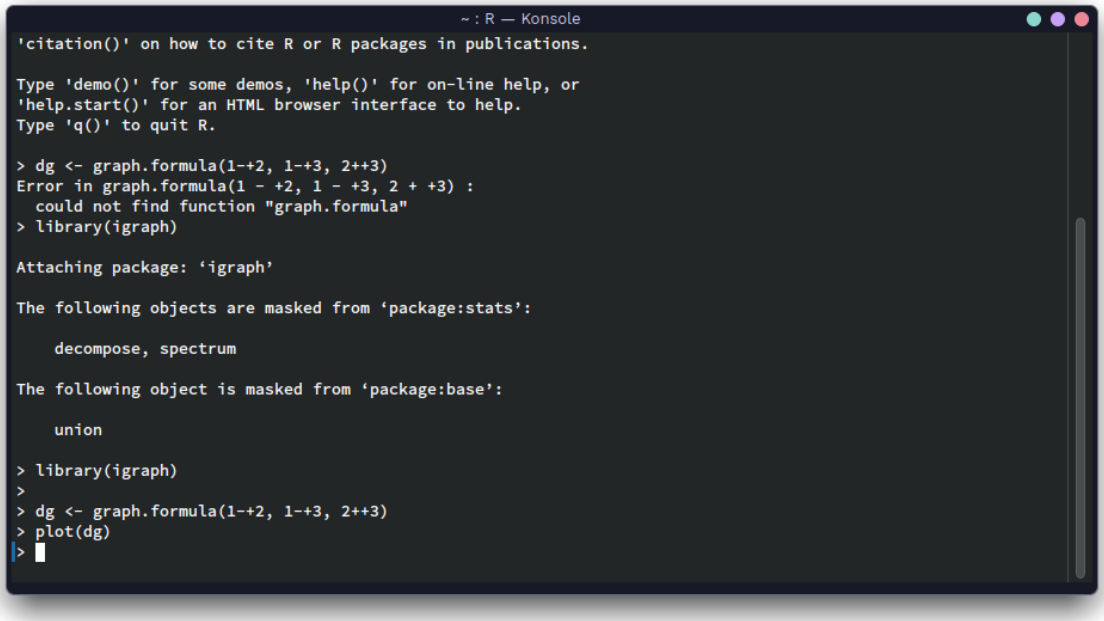


HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)



Directed graph

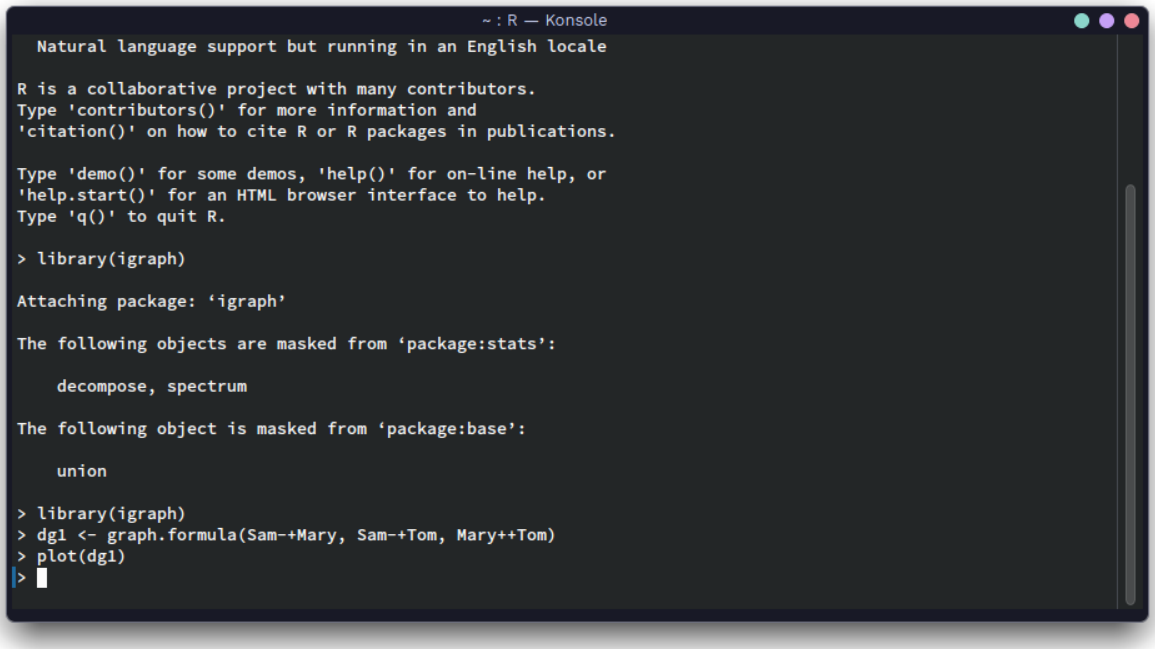
> dg <- graph.formula(1-+2, 1-+3, 2++3)

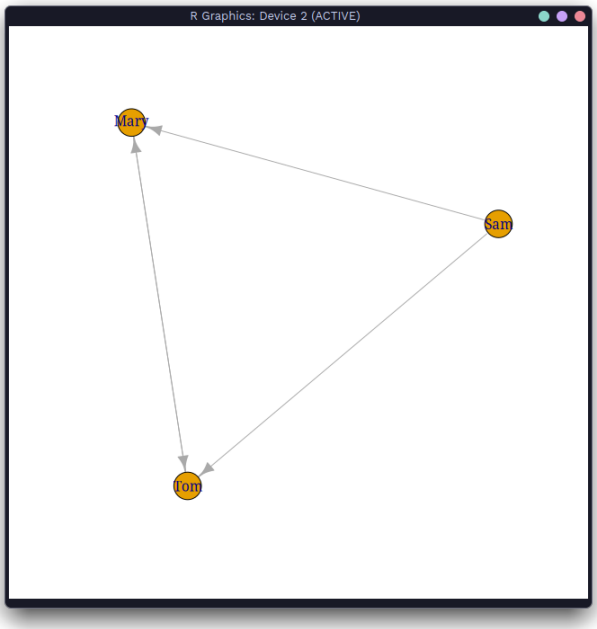


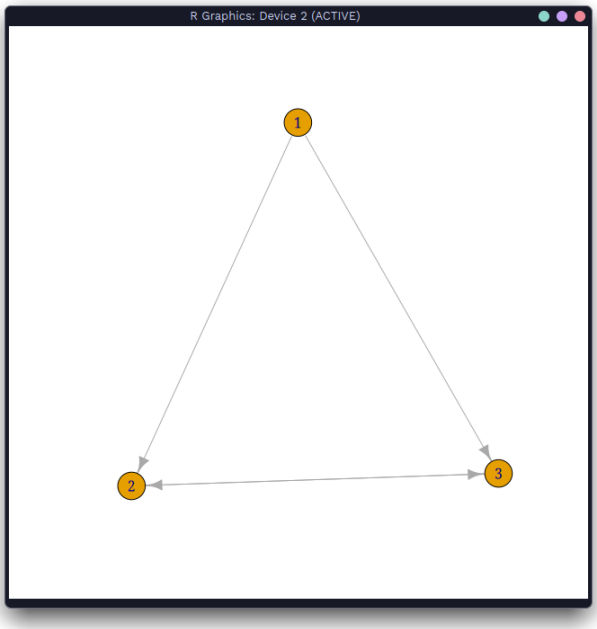
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Graph with names

> dg1 <- graph.formula(Sam-+Mary, Sam-+Tom, Mary++Tom)
> plot(dg1)





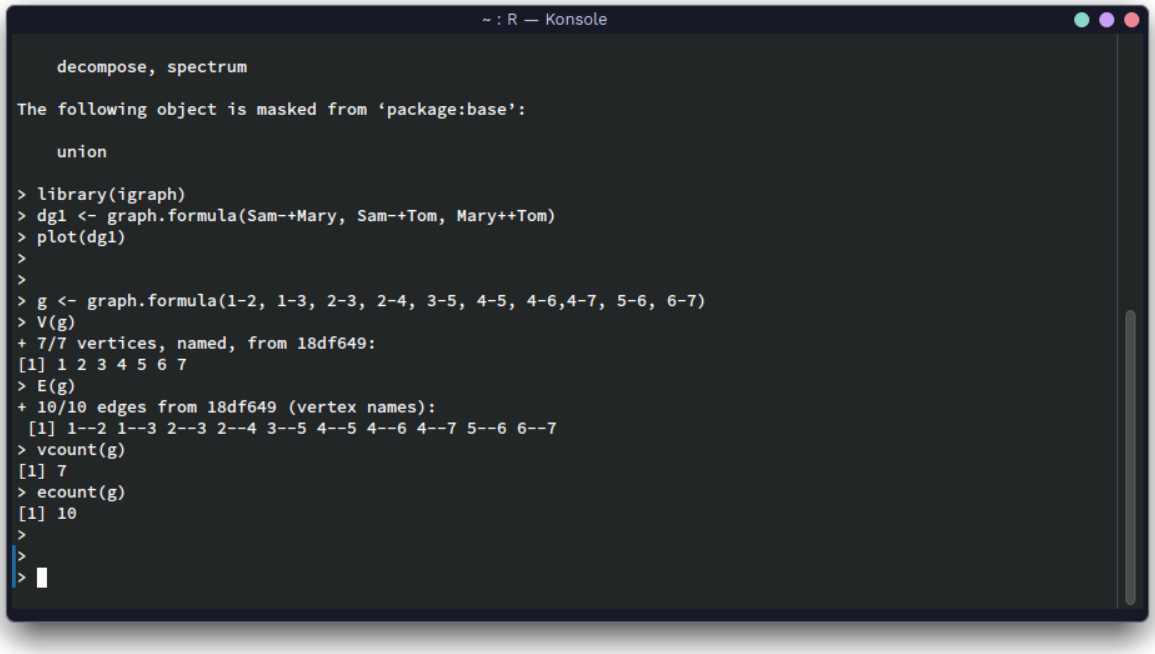


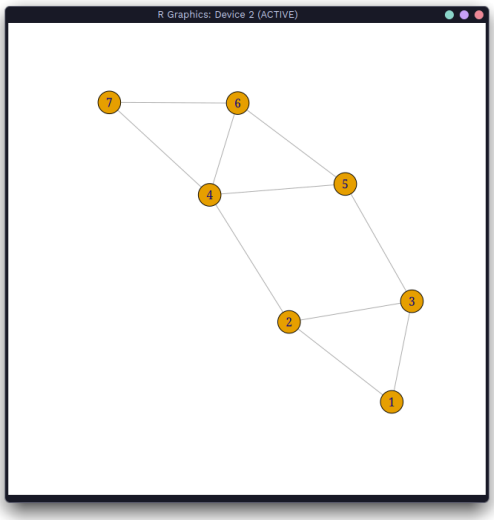
Number of vertices/node:

> vcount(g)

Number of edges/dyad/ties:

> ecount(g)





Degree of a Node

> degree(g)

In-degree

> degree(dg, mode="in")

Out-degree

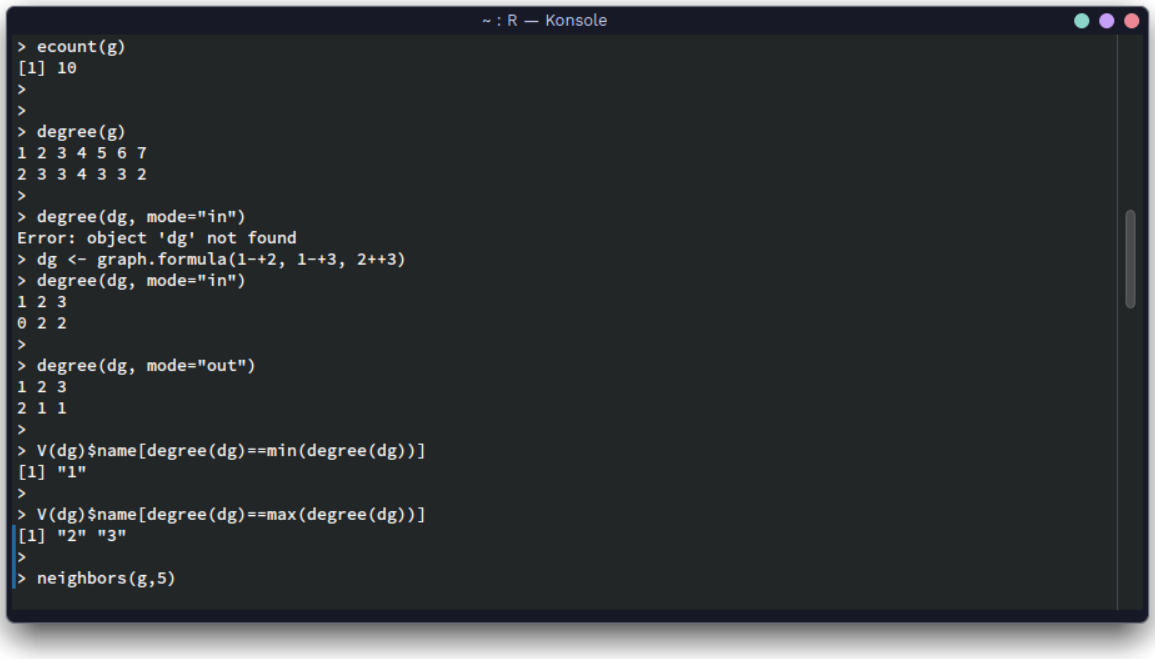
> degree(dg, mode="out")

Node with lowest degree

>V(dg)$name[degree(dg)==min(degree(dg))]

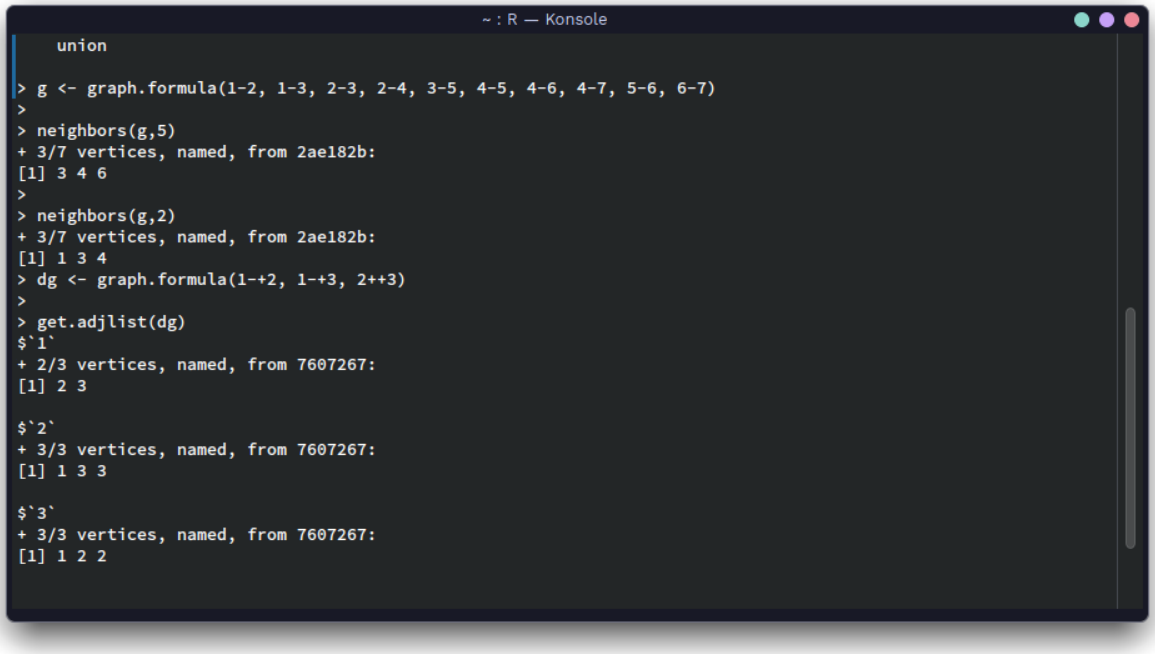
Node with highest degree

> V(dg)$name[degree(dg)==max(degree(dg))]



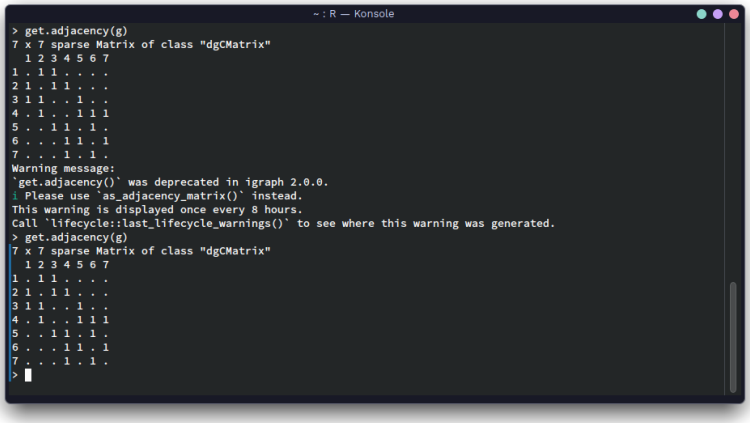
To find neighbours / adjacency list:

> neighbors(g,5)
> neighbors(g,2)
> get.adjlist(dg)
> get.adjacency(g)



Adjacency Matrix

> get.adjacency(g)



HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 2	
Aim: Perform following tasks: **(i) View data collection forms and/or import one-mode/two-mode datasets;** **(ii) Basic Networks matrices transformations**	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 10/07/24	Sign:

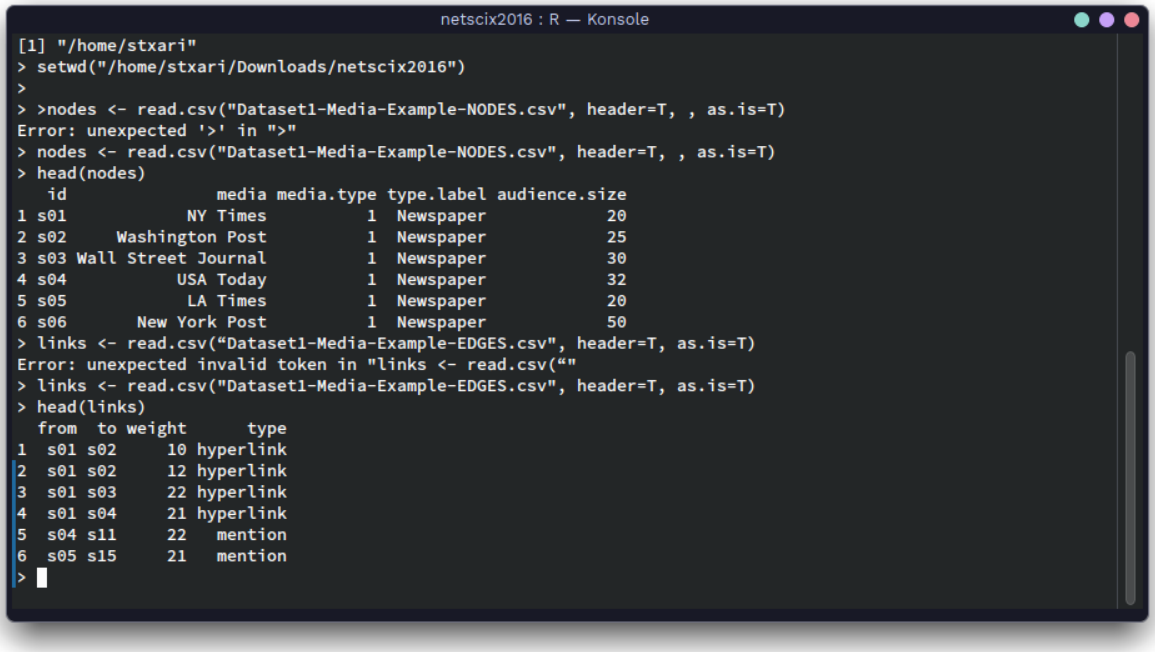
Note:

Where your working directory is set at this moment
> getwd()

R now knows exactly in which folder you're working.
> setwd("<location of your dataset>")

Reading data from a csv file

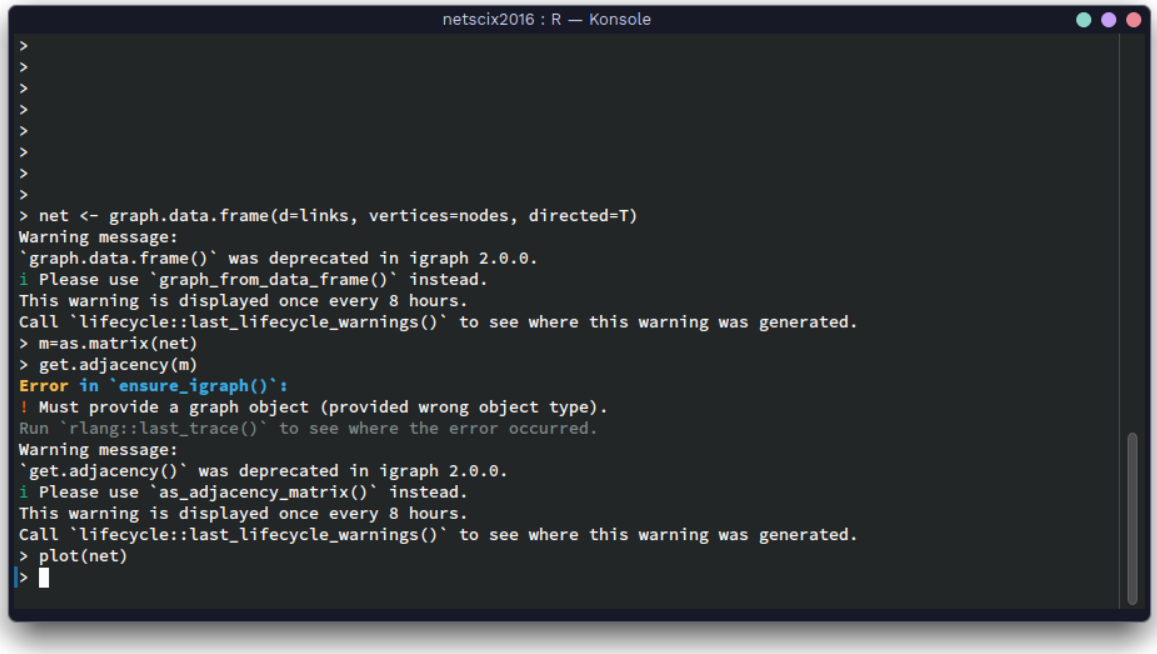
> nodes <- read.csv("Dataset1-Media-Example-NODES.csv", header=T, , as.is=T)
> head(nodes)
> links <- read.csv(“Dataset1-Media-Example-EDGES.csv", header=T, as.is=T)
> head(links)

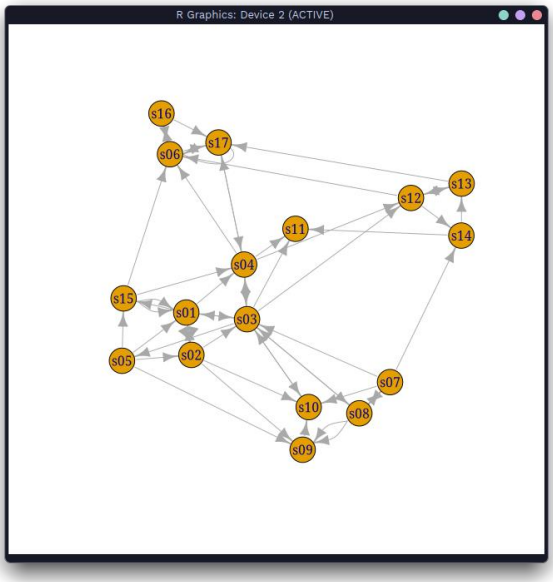


HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Basic Networks matrices transformations

> net <- graph.data.frame(d=links, vertices=nodes, directed=T)
> m=as.matrix(net)
>get.adjacency(m)
>plot(net)





HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

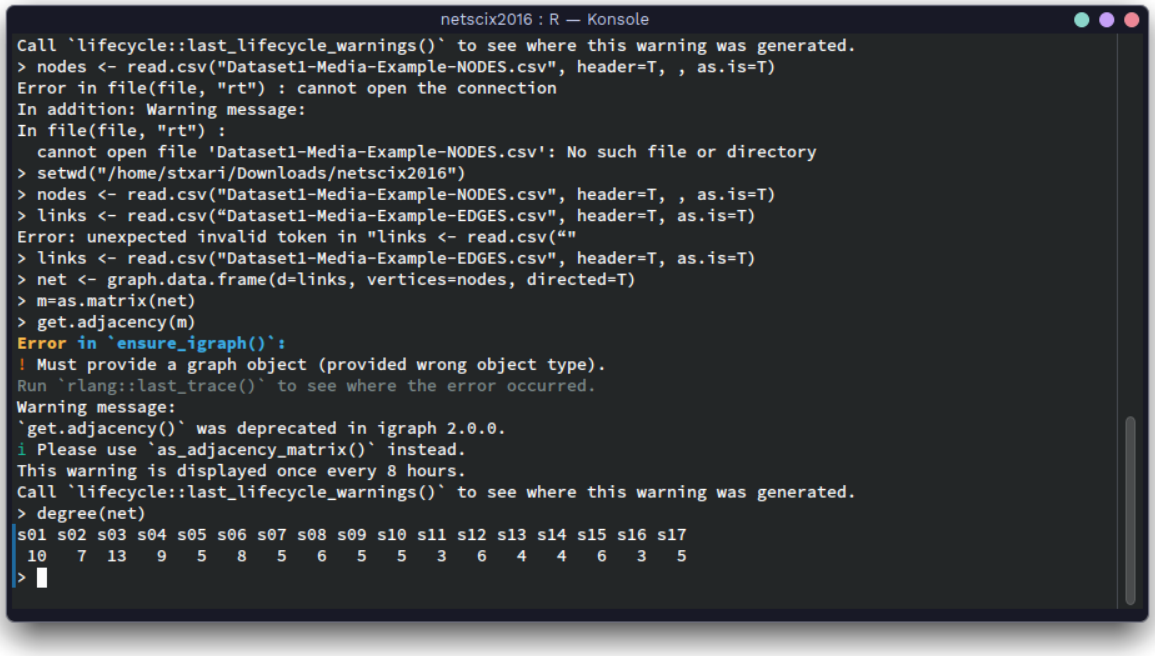
Practical No. 3	
Aim: Compute the following node level measures: **(i) Density; (ii) Degree; (iii) Reciprocity; (iv) Transitivity; (v) Centralization; (vi) Clustering.**	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 28/07/24	Sign:

Density

> vcount(g)
> ecount(g)
> ecount(g)/(vcount(g)*(vcount(g)-1))

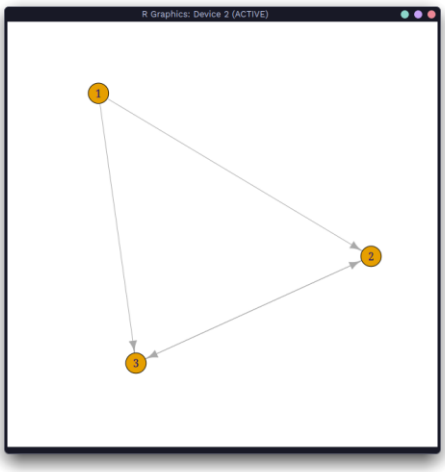
Degree

degree(net)



Reciprocity

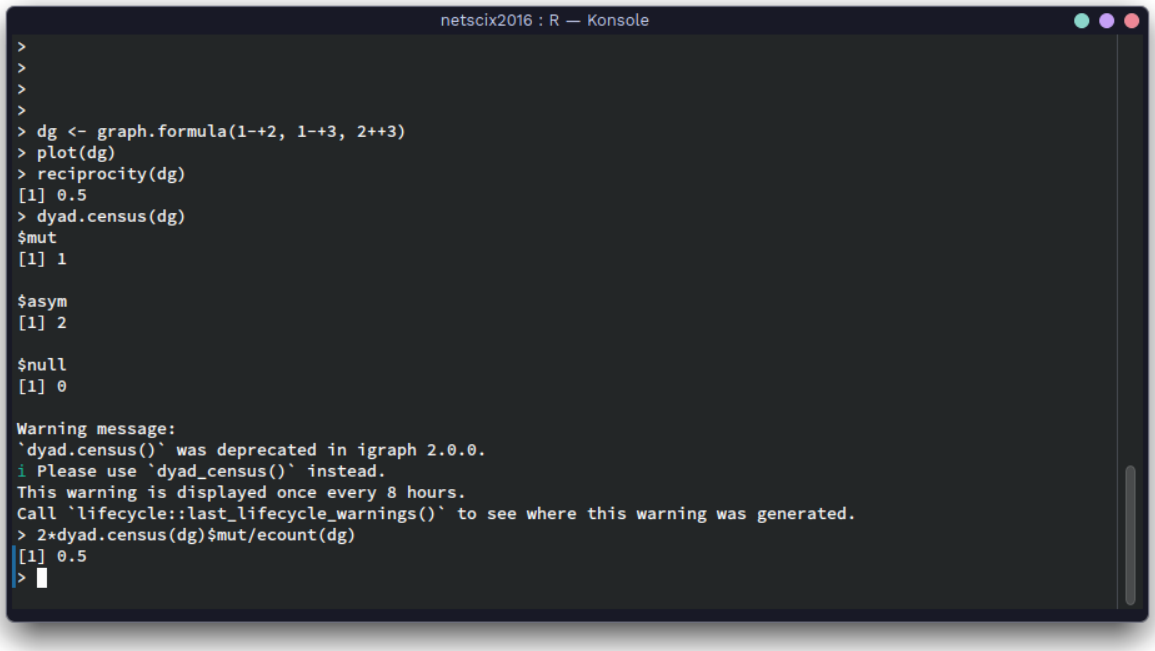
> dg <- graph.formula(1-+2, 1-+3, 2++3)
> plot(dg)
> reciprocity(dg)



HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

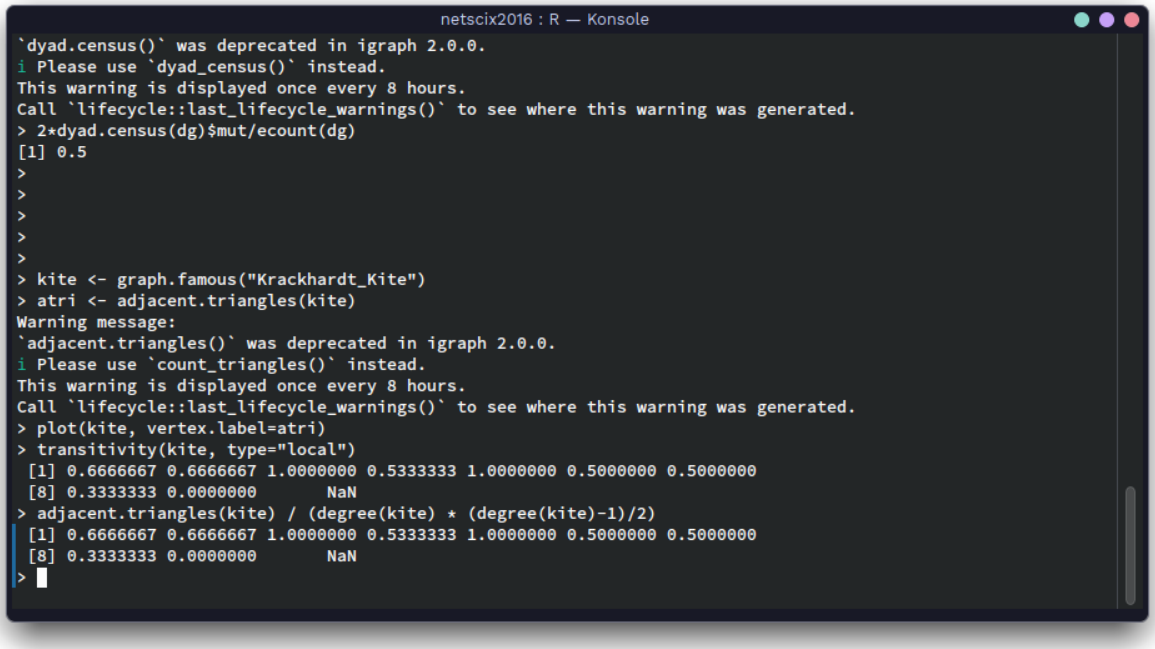
Formula as per text book

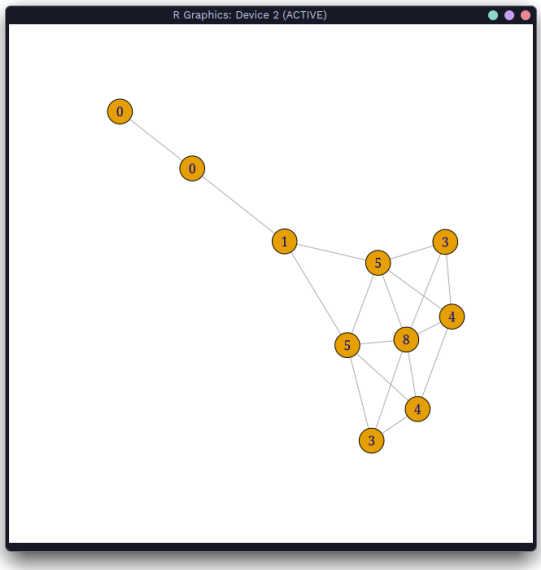
> dyad.census(dg)
> 2*dyad.census(dg)$mut/ecount(dg)



Transitivity

```
> kite <- graph.famous("Krackhardt_Kite")
> atri <- adjacent.triangles(kite)
> plot(kite, vertex.label=atri)
> transitivity(kite, type="local")
> adjacent.triangles(kite) / (degree(kite) * (degree(kite)-1)/2)
```





Centralization

Degree of centrality

```
> centralization.degree(net, mode="in", normalized=T)
```

Closeness Centralization

```
> closeness(net, mode="all", weights=NA)
> centralization.closeness(net, mode="all",normalized=T)
```

Betweeness Centrality

```
> betweenness(net, directed=T, weights=NA)
> edge.betweenness(net, directed=T, weights=NA)
> centralization.betweenness(net, directed=T, normalized=T)
```

Eigenvector centrality

```
> centralization.evcent(net, directed=T, normalized=T)
```

Clustering

```
> plot(kite)
> get.adjedgelist(kite, mode = c("all", "out", "in", "total"))
```

```
netscix2016 : R — Konsole
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> centralization.betweenness(net, directed=T, normalized=T)

$res
 [1]  26.857143   6.238095 126.511905  92.642857  13.000000  20.333333
 [7]   1.750000  21.000000   1.000000  15.000000   0.000000  33.500000
[13]  20.000000   4.000000   5.666667   0.000000  58.500000

$centralization
[1] 0.4439329

$theoretical_max
[1] 3840

Warning message:
`centralization.betweenness()` was deprecated in igraph 2.0.0.
i Please use `centr_betw()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> centralization.evcent(net, directed=T, normalized=T)

$vector
 [1] 0.7694528 0.5623895 1.0000000 0.8569443 0.3049992 0.9285033 0.1025656
 [8] 0.3362816 0.4696841 0.6510633 0.6361813 0.6479337 0.2674341 0.2289017
[15] 0.3277070 0.2831928 0.7125008
```

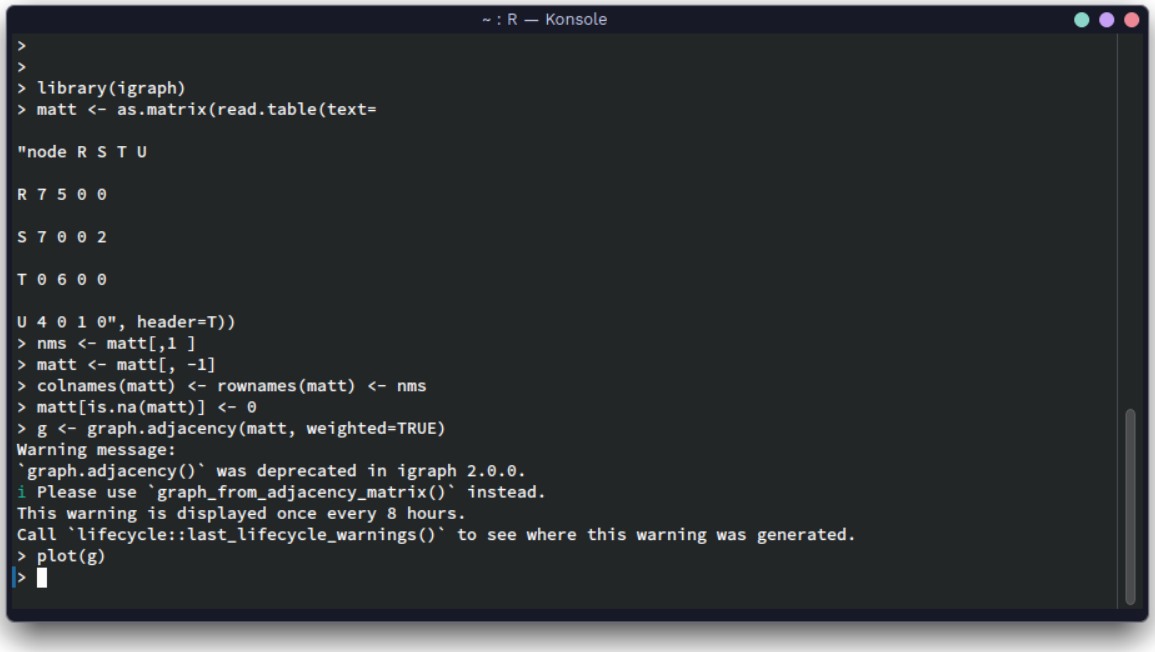
```
netscix2016 : R — Konsole

Warning message:
`centralization.evcent()` was deprecated in igraph 2.0.0.
i Please use `centr_eigen()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> plot(kite)

> get.adjedgelist(kite, mode = c("all", "out", "in", "total"))

[[1]]
+ 4/18 edges from dc8536a:
[1] 1--2 1--3 1--4 1--6

[[2]]
+ 4/18 edges from dc8536a:
[1] 1--2 2--4 2--5 2--7

[[3]]
+ 3/18 edges from dc8536a:
[1] 1--3 3--4 3--6

[[4]]
+ 6/18 edges from dc8536a:
[1] 1--4 2--4 3--4 4--5 4--6 4--7
```

HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

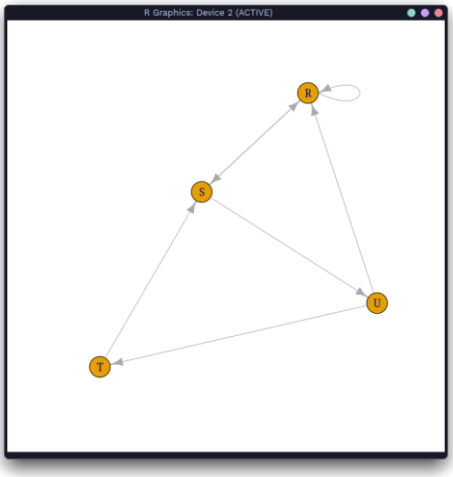
Practical No. 4	
Aim: For a given network find the following: **(i)Length of the shortest path from a given node to another node;** **(ii) the density of the graph;** **(iii) Draw egocentric network of node G with chosen configuration parameters.**	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 1/08/24	Sign:

Length of the shortest path from a given node to another node.

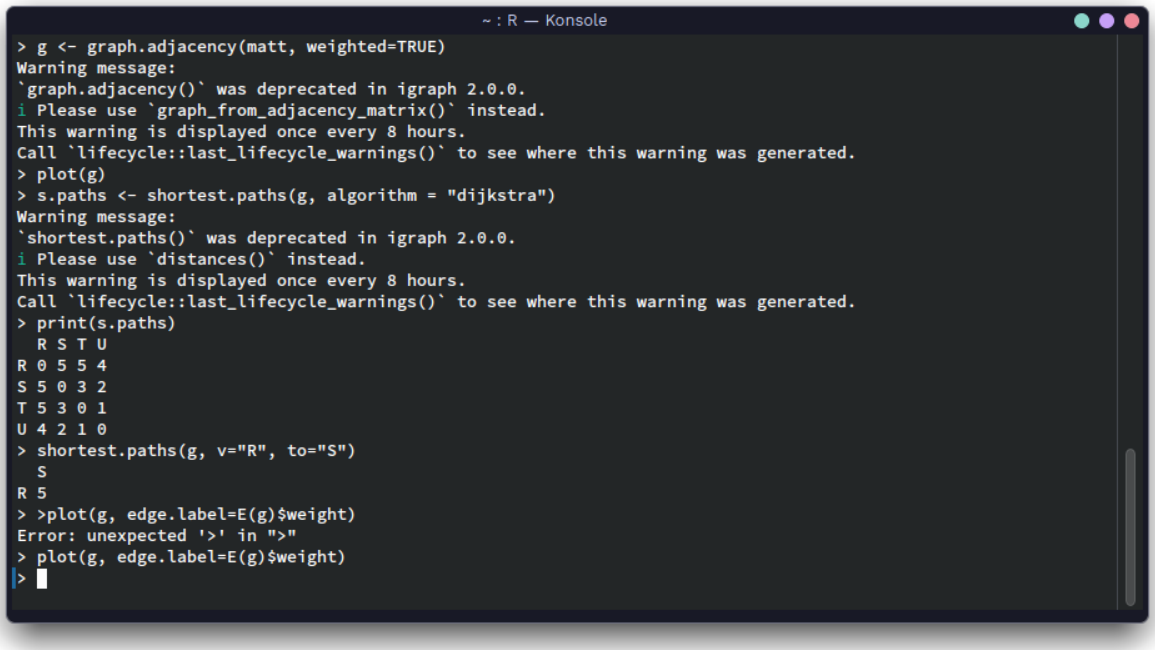
```
> library(igraph)
> matt <- as.matrix(read.table(text= "node R S T U
R 7 5 0 0
S 7 0 0 2
T 0 6 0 0
U 4 0 1 0", header=T))
> nms <- matt[,1 ]
> matt <- matt[, -1]
> colnames(matt) <- rownames(matt) <- nms
> matt[is.na(matt)] <- 0
> g <- graph.adjacency(matt, weighted=TRUE)
> plot(g)
```



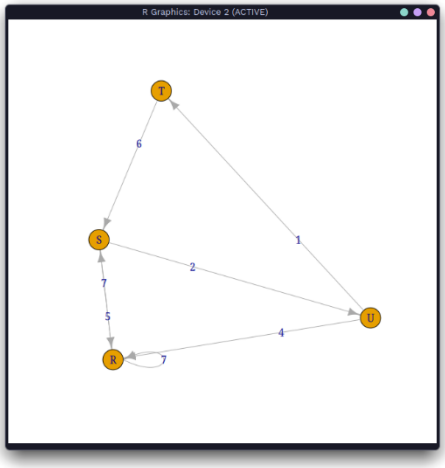
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)




```
> s.paths <- shortest.paths(g, algorithm = "dijkstra")
> print(s.paths)
> shortest.paths(g, v="R", to="S")
>plot(g, edge.label=E(g)$weight)
```



HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)



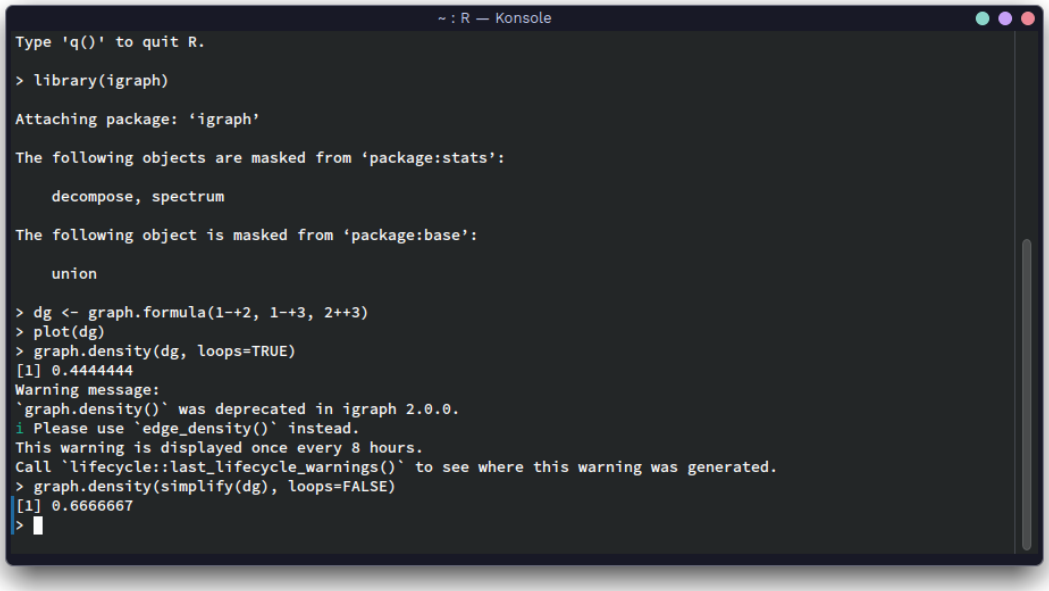
The density of the graph

- The density of a graph is the ratio of the number of edges and the number of possible edges.

```
> library(igraph)
> dg <- graph.formula(1-+2, 1-+3, 2++3)
> plot(dg)
> graph.density(dg, loops=TRUE)
```

- Without considering loops

```
> graph.density(simplify(dg), loops=FALSE)
```

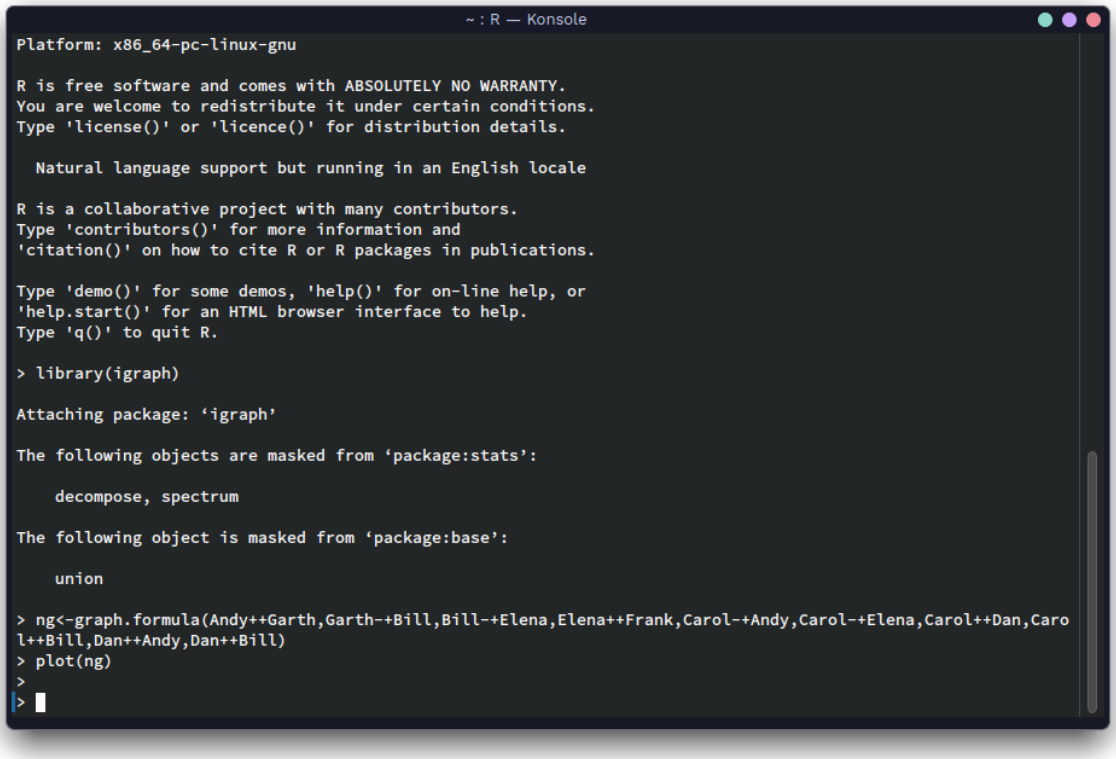


HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

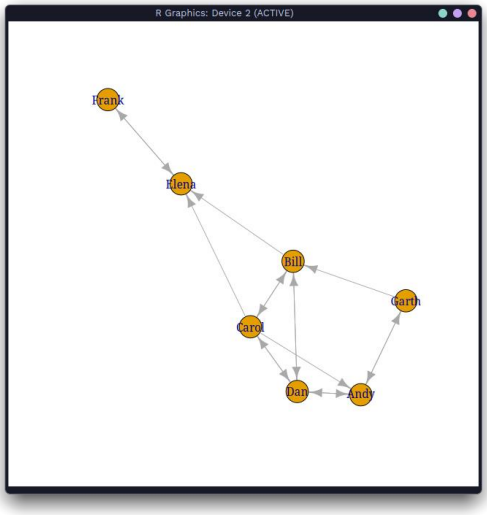
Practical No. 5	
Aim: Write a program to distinguish between a network as a matrix, a network as an edge list, and a network as a sociogram (or "network graph") using 3 distinct networks representatives of each.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 12/08/24	Sign:

A network as a graph

```
> library(igraph)
> ng<-graph.formula(Andy++Garth,Garth-+Bill,Bill-+Elena,Elena++Frank,Carol-+Andy,Carol-+Elena,Carol++Dan,Carol++Bill,Dan++Andy,Dan++Bill)
> plot(ng)
```

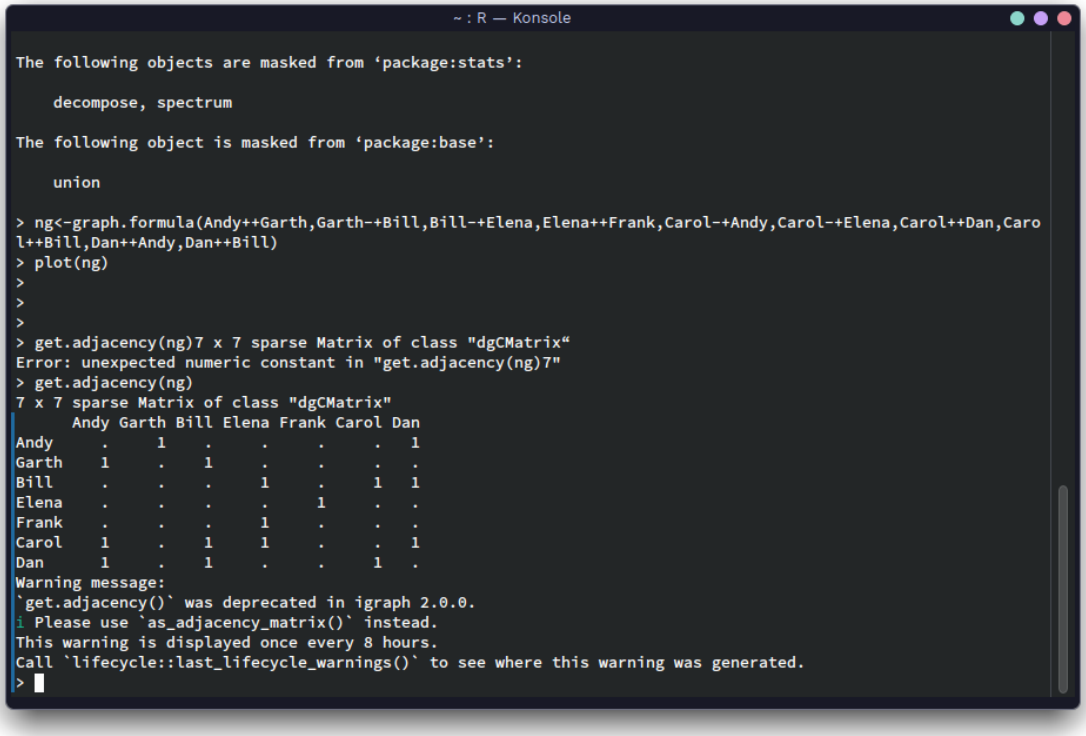


HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

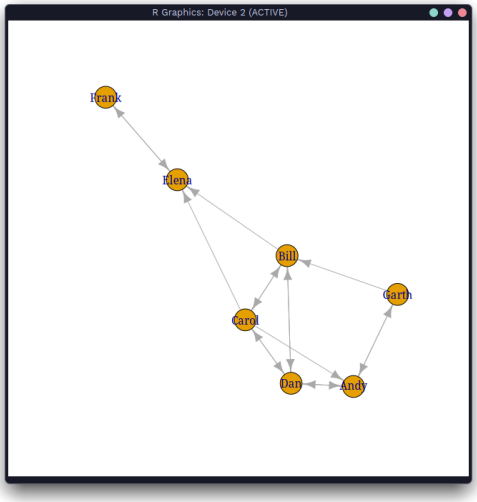


A network as a matrix

> get.adjacency(ng)



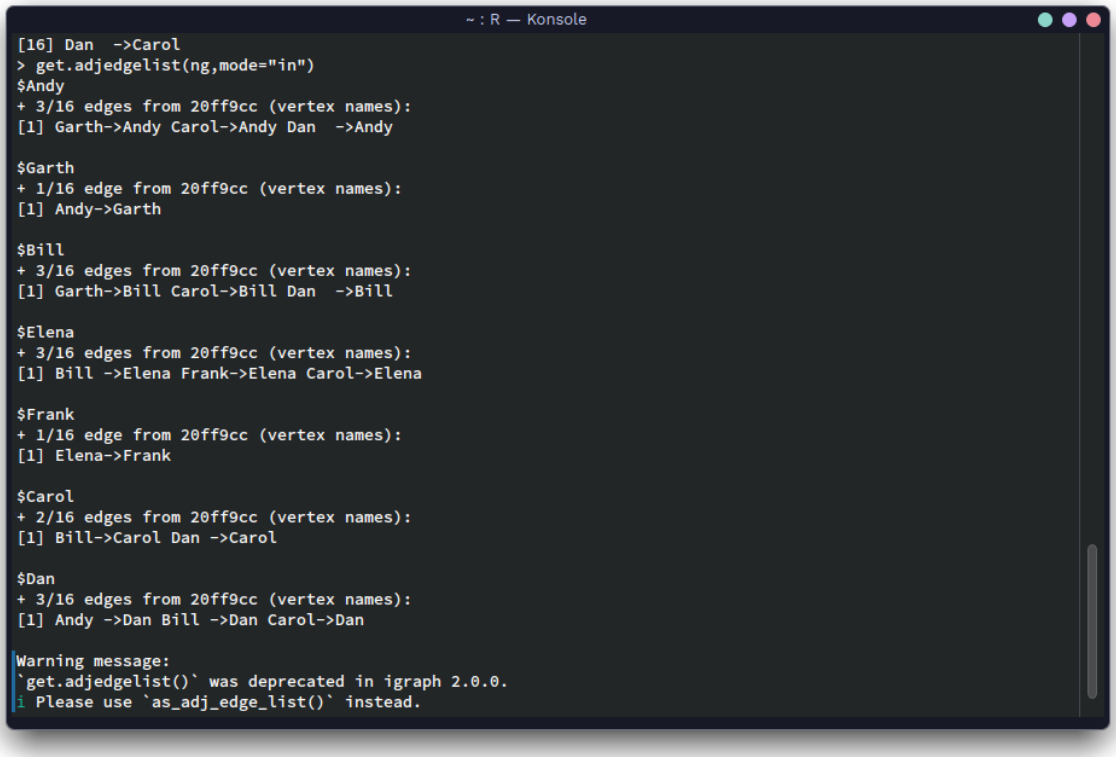
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)



A network as an edge list

> E(ng)

```
~ : R — Konsole

    union

> ng<-graph.formula(Andy++Garth,Garth-+Bill,Bill-+Elena,Elena++Frank,Carol-+Andy,Carol-+Elena,Carol++Dan,Caro
l++Bill,Dan++Andy,Dan++Bill)
> plot(ng)
>
>
>
> get.adjacency(ng)7 x 7 sparse Matrix of class "dgCMatrix“
Error: unexpected numeric constant in "get.adjacency(ng)7"
> get.adjacency(ng)
7 x 7 sparse Matrix of class "dgCMatrix"
      Andy Garth Bill Elena Frank Carol Dan
Andy     .     1    .     .     .     .   1
Garth    1     .    1     .     .     .   .
Bill     .     .    .     1     .     1   1
Elena    .     .    .     .     1     .   .
Frank    .     .    .     1     .     .   .
Carol    1     .    1     1     .     .   1
Dan      1     .    1     .     .     1   .
Warning message:
`get.adjacency()` was deprecated in igraph 2.0.0.
i Please use `as_adjacency_matrix()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> E(ng)
+ 16/16 edges from 20ff9cc (vertex names):
 [1] Andy ->Garth Andy ->Dan   Garth->Andy  Garth->Bill  Bill ->Elena
 [6] Bill ->Carol Bill ->Dan   Elena->Frank Frank->Elena Carol->Andy
[11] Carol->Bill  Carol->Elena Carol->Dan   Dan  ->Andy  Dan  ->Bill
[16] Dan  ->Carol
>
```

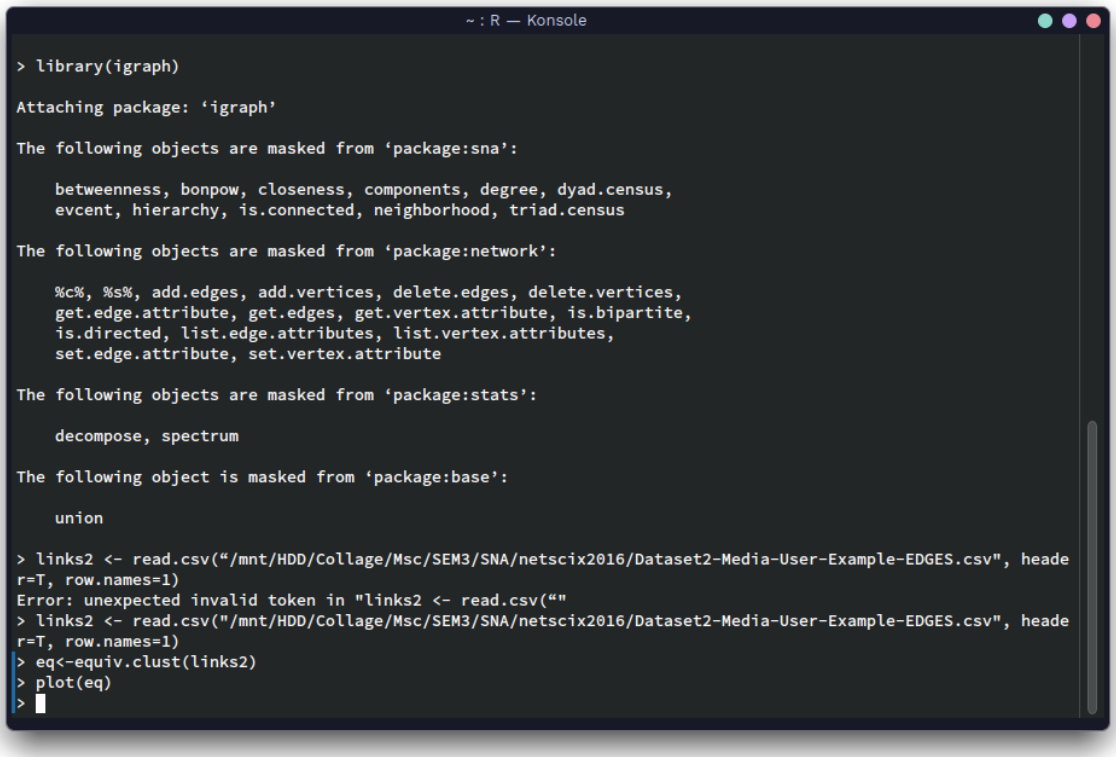

~ : R — Konsole
[16] Dan ->Carol
> get.adjedgelist(ng,mode="in")
$Andy
+ 3/16 edges from 20ff9cc (vertex names):
[1] Garth->Andy Carol->Andy Dan ->Andy
$Garth
+ 1/16 edge from 20ff9cc (vertex names):
[1] Andy->Garth
$Bill
+ 3/16 edges from 20ff9cc (vertex names):
[1] Garth->Bill Carol->Bill Dan ->Bill
$Elena
+ 3/16 edges from 20ff9cc (vertex names):
[1] Bill ->Elena Frank->Elena Carol->Elena
$Frank
+ 1/16 edge from 20ff9cc (vertex names):
[1] Elena->Frank
$Carol
+ 2/16 edges from 20ff9cc (vertex names):
[1] Bill->Carol Dan ->Carol
$Dan
+ 3/16 edges from 20ff9cc (vertex names):
[1] Andy ->Dan Bill ->Dan Carol->Dan
Warning message:
`get.adjedgelist()` was deprecated in igraph 2.0.0.
i Please use `as_adj_edge_list()` instead.

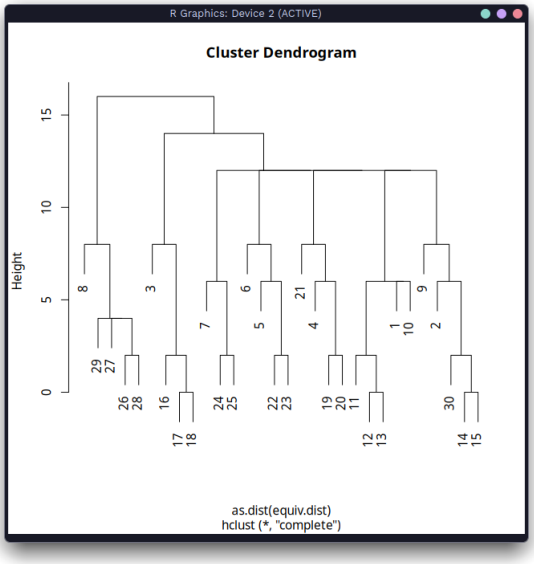
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 6	
Aim: Write a program to exhibit structural equivalence, automorphic equivalence, and regular equivalence from a network.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 26/08/24	Sign:

Code:

```
> library(sna)
> library(igraph)
> links2 <- read.csv("/mnt/HDD/Collage/Msc/SEM3/SNA/netscix2016/Dataset2-Media-User-Example-EDGES.csv", header=T, row.names=1)
> eq<-equiv.clust(links2)
> plot(eq)
```



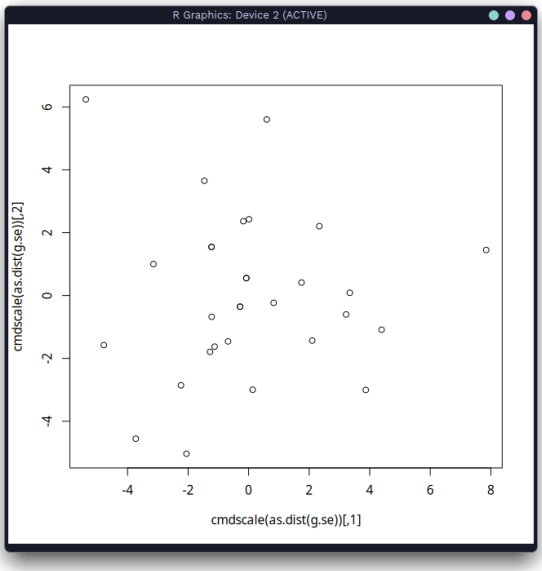


Get Structural Equivalence distances

>g.se<-sedist(links2)

Plot a metric MDS of vertex positions in two dimensions

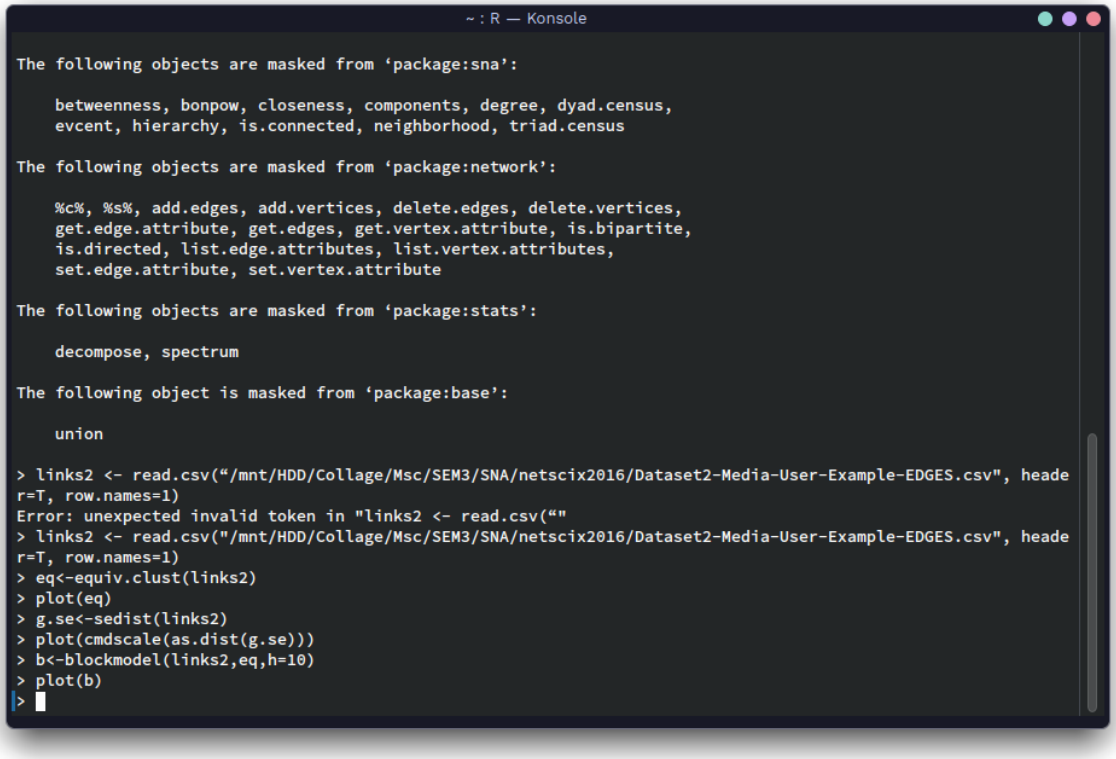
>plot(cmdscale(as.dist(g.se)))

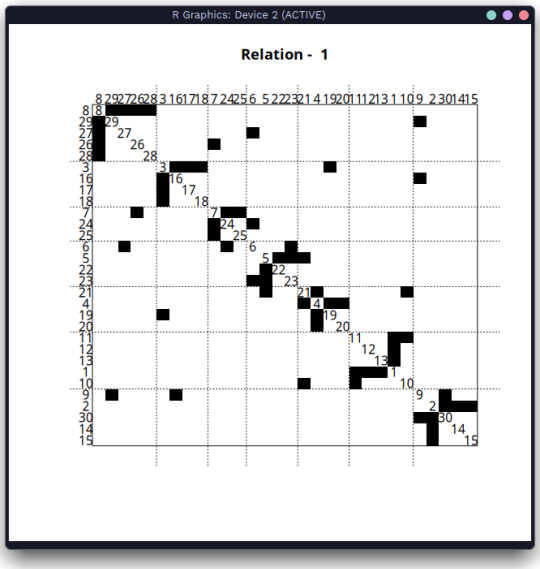


HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Blockmodeling

> b<-blockmodel(links2,eq,h=10)
> plot(b)



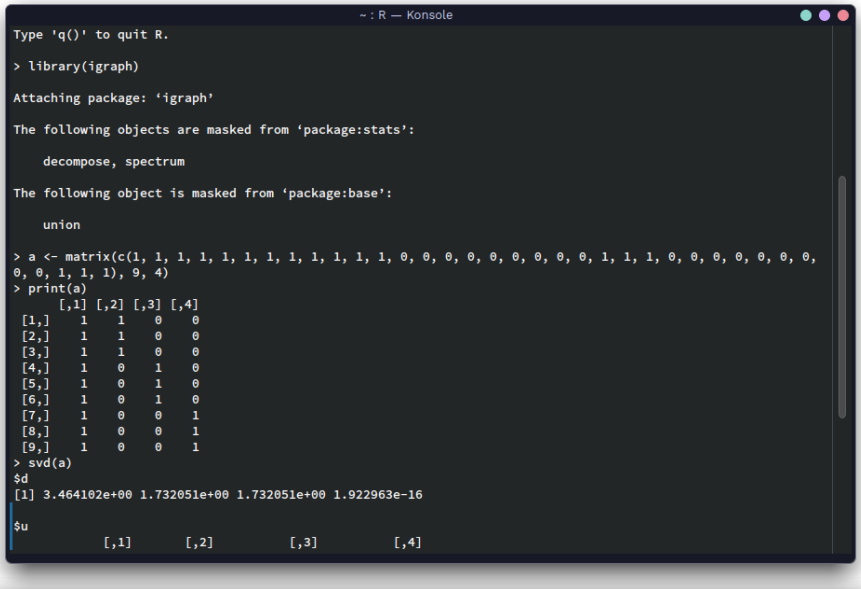


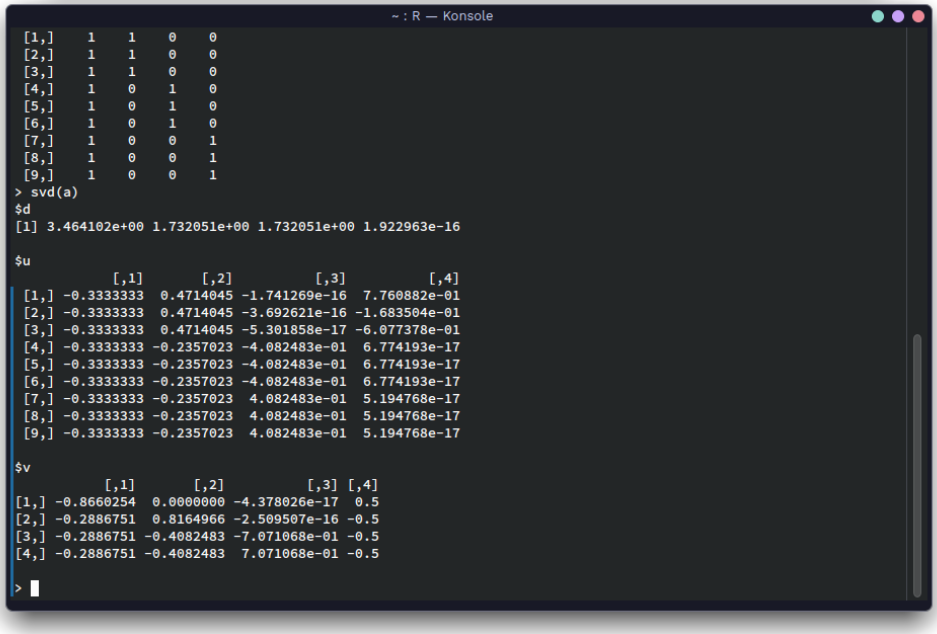
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 7	
Aim: Perform SVD analysis of a network.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 3/09/24	Sign:

Code:

```
> library(igraph)
> a <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, , 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1), 9, 4)
> print(a)
```

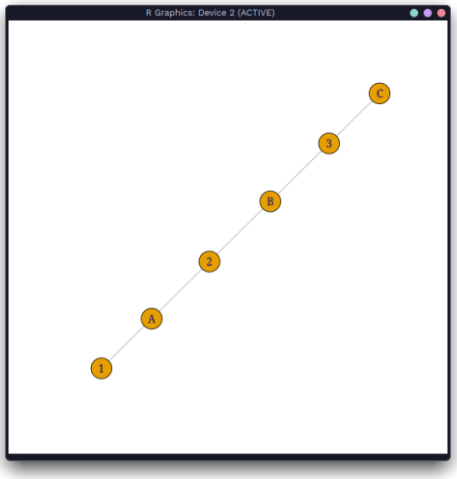




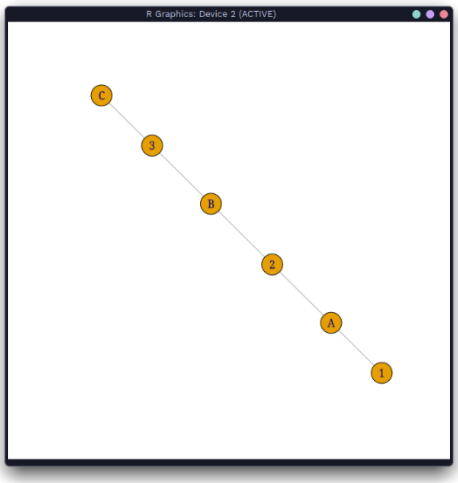
HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 8	
Aim: Displaying Bipartite network in the graph format.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 17/09/24	Sign:

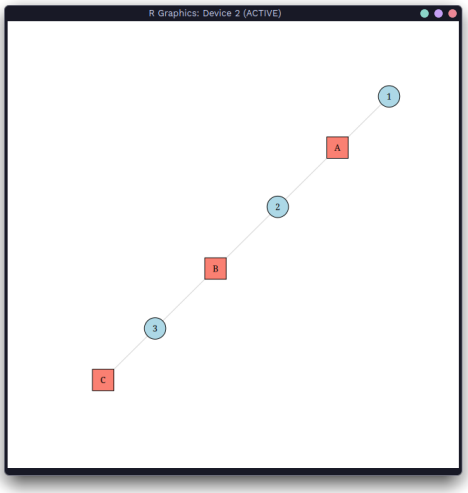
```
> library(igraph)
> davis <-read.csv("/mnt/HDD/Collage/Msc/SEM3/SNA/netscix2016/csv.csv")
> g <- graph.data.frame(davis, directed=FALSE)
> plot(g)
```



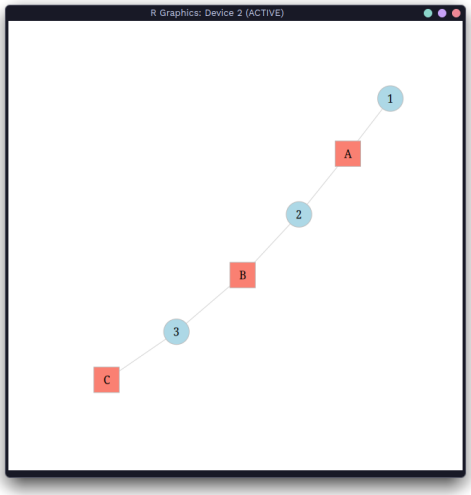

```
> bipartite.mapping(g)
> V(g)$type <- bipartite_mapping(g)$type
> plot(g)
```



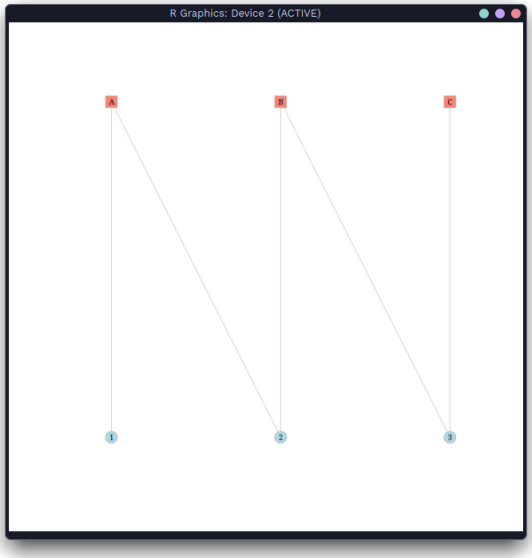
```
> plot(g, vertex.label.cex = 0.8, vertex.label.color ="black")
> V(g)$color <- ifelse(V(g)$type, "lightblue", "salmon")
> V(g)$shape <- ifelse(V(g)$type, "circle", "square")
> E(g)$color <- "lightgray"
> plot(g, vertex.label.cex = 0.8, vertex.label.color ="black")
```




```
> V(g)$label.color <- "black"
> V(g)$label.cex <- 1
> V(g)$frame.color <- "gray"
> V(g)$size <- 18
> plot(g, layout = layout_with_graphopt)

> plot(g, layout=layout.bipartite, vertex.size=7,
vertex.label.cex=0.6)
```



HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)



~ : R — Konsole

```
Warning message:
`graph.data.frame()` was deprecated in igraph 2.0.0.
i Please use `graph_from_data_frame()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> plot(g)
> bipartite.mapping(g)
$res
[1] TRUE

$type
    A     B     C     1     2     3
FALSE FALSE FALSE  TRUE  TRUE  TRUE

Warning message:
`bipartite.mapping()` was deprecated in igraph 2.0.0.
i Please use `bipartite_mapping()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> V(g)$type <- bipartite_mapping(g)$type
> plot(g)
> plot(g, vertex.label.cex = 0.8, vertex.label.color ="black")
> V(g)$color <- ifelse(V(g)$type, "lightblue", "salmon")
> V(g)$shape <- ifelse(V(g)$type, "circle", "square")
> E(g)$color <- "lightgray"
> plot(g, vertex.label.cex = 0.8, vertex.label.color ="black")
> V(g)$label.color <- "black"
> V(g)$label.cex <- 1
> V(g)$frame.color <- "gray"
> V(g)$size <- 18
> plot(g, layout = layout_with_graphopt)
> plot(g, layout=layout.bipartite, vertex.size=7,
vertex.label.cex=0.6)
```

HSNC UNIVERSITY, MUMBAI
KISHINCHAND CHELLARAM COLLEGE
M. Sc Computer Science Semester 3 (2024 – 25)

Practical No. 9	
Aim: Hamming distance.	
Name: Saail Chavan	Roll No.: KFPMSCCS016
Date: 05/10/24	Sign:

Code:

```
> library(e1071)
> x <- c(0, 0, 0, 0)
> y <- c(0, 1, 0, 1)
> z <- c(1, 0, 1, 1)
> w <- c(0, 1, 1, 1)
> hamming.distance(x, y)
> hamming.distance(y,z)
> hamming.distance(y,w)
> hamming.distance(z,w)
> hamming.distance(x, w)
> hamming.distance(x, z)
```