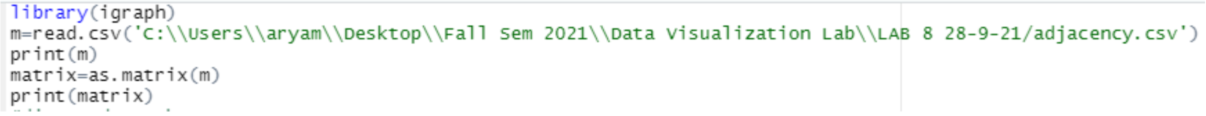
Aryaman Mishra

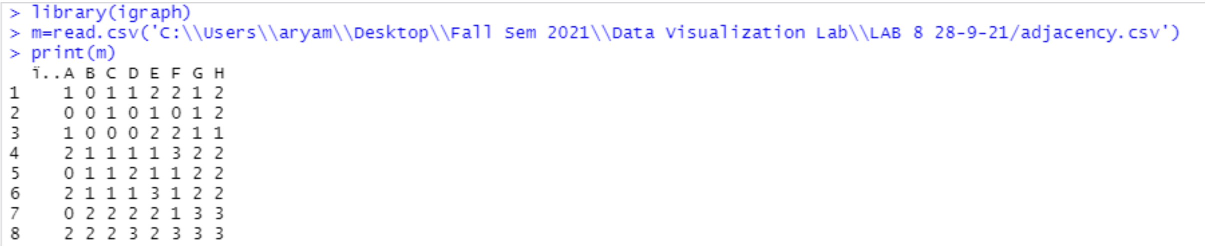
19BCE1027

LAB 8

1. **Read the given adjacency matrix into R (adjacency.csv)**
   library(igraph)
   m=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/adjacency.csv')
   print(m)
   matrix=as.matrix(m)
   print(matrix)

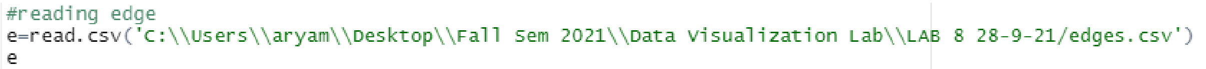


```
library(igraph)
m=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/adjacency.csv')
print(m)
matrix=as.matrix(m)
print(matrix)
```

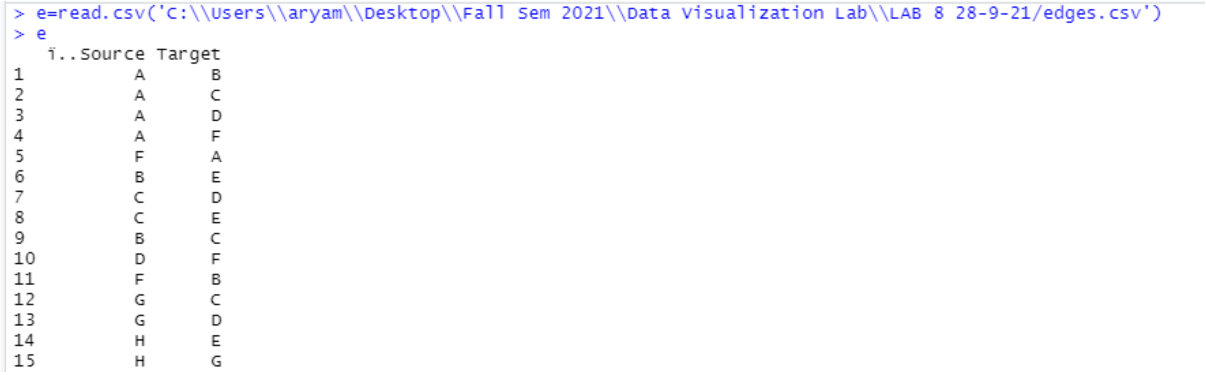


```
> library(igraph)
> m=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/adjacency.csv')
> print(m)
  ï..A B C D E F G H
1    1 0 1 1 2 2 1 2
2    0 0 1 0 1 0 1 2
3    1 0 0 0 2 2 1 1
4    2 1 1 1 1 3 2 2
5    0 1 1 2 1 1 2 2
6    2 1 1 1 3 1 2 2
7    0 2 2 2 2 1 3 3
8    2 2 2 3 2 3 3 3
```

2. **Read the given edge matrix into R(edges.csv)**
   #reading edge
   e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')
   e



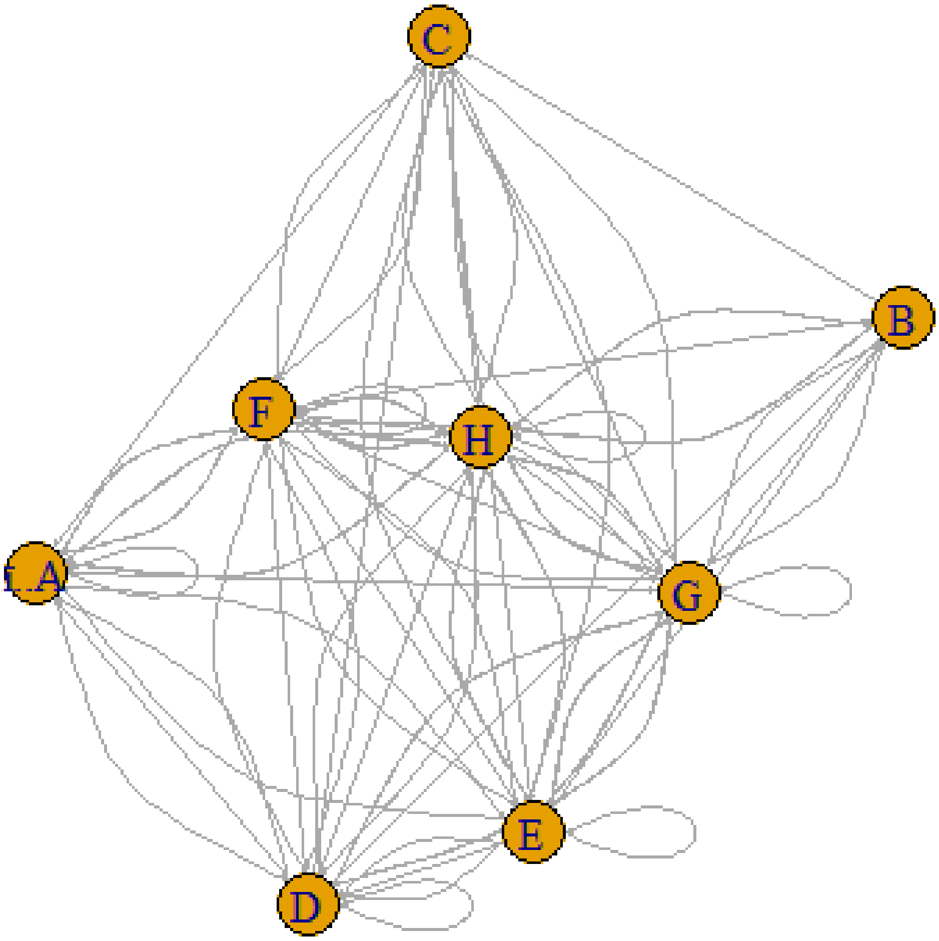
```
#reading edge
e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')
e
```



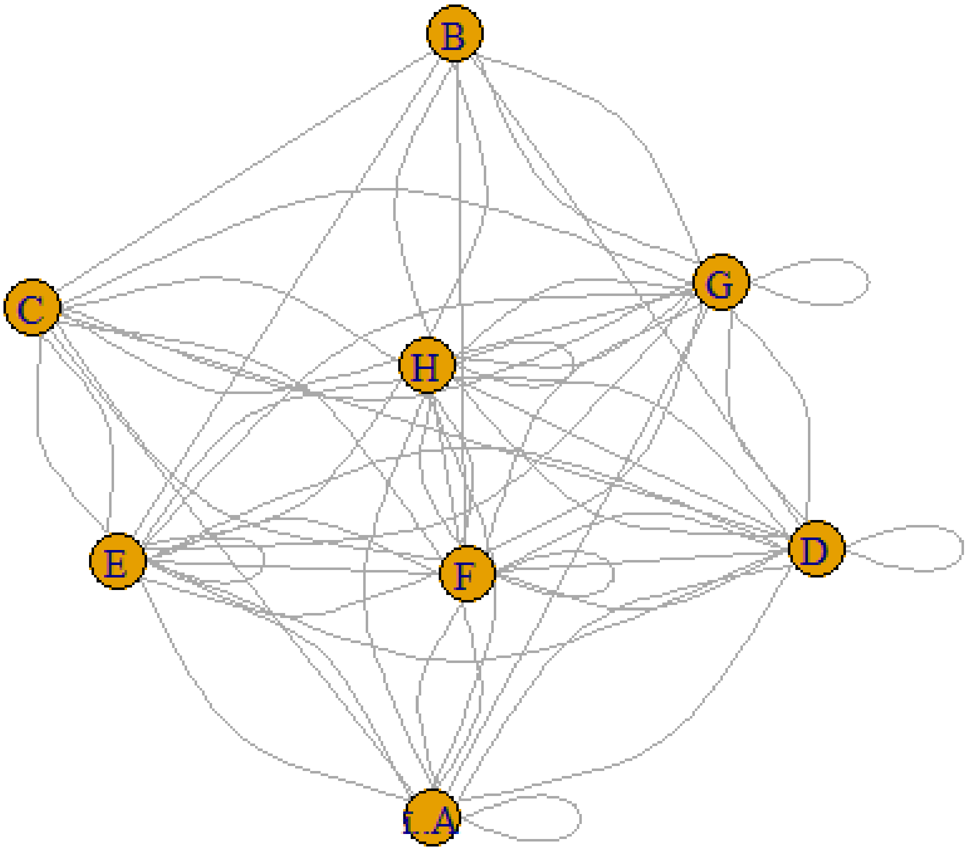
```
> e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')
> e
   ï..Source Target
1          A      B
2          A      C
3          A      D
4          A      F
5          F      A
6          B      E
7          C      D
8          C      E
9          B      C
10         D      F
11         F      B
12         G      C
13         G      D
14         H      E
15         H      G
```

3. Create and plot the graph from the adjacency matrix and edge matrix

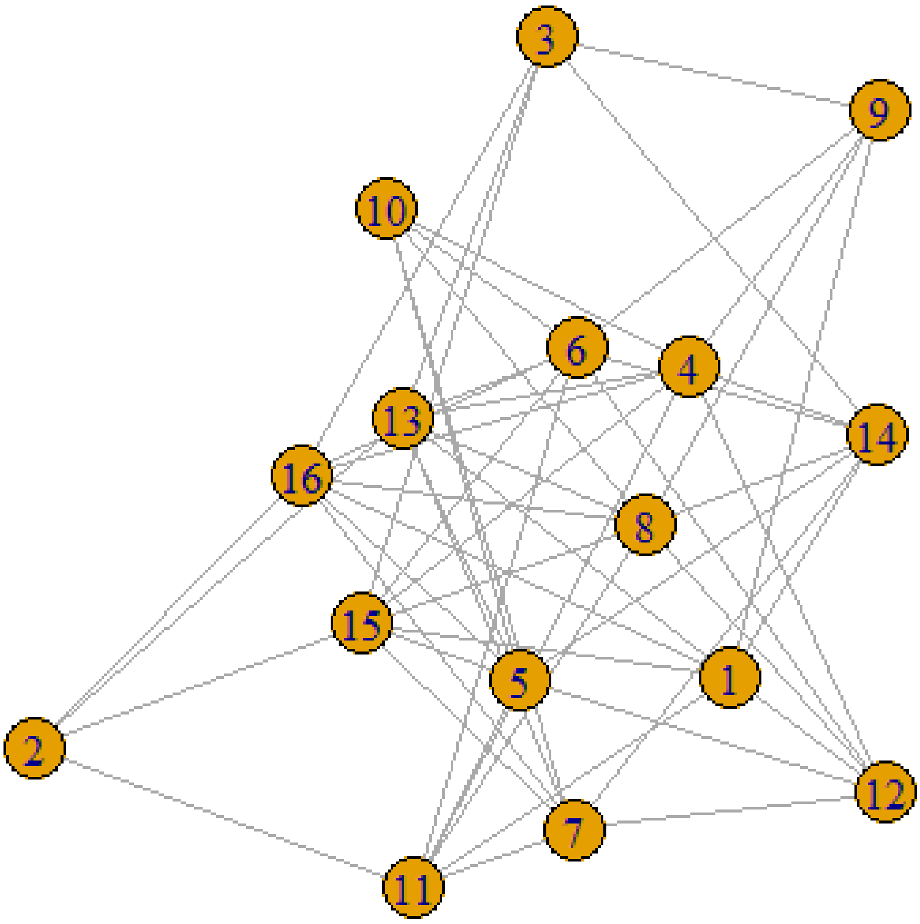
```
#directed graph

g=graph.adjacency(matrix,mode="directed",weighted=NULL)

plot.igraph(g,edge.arrow.size=0.1)

g
```



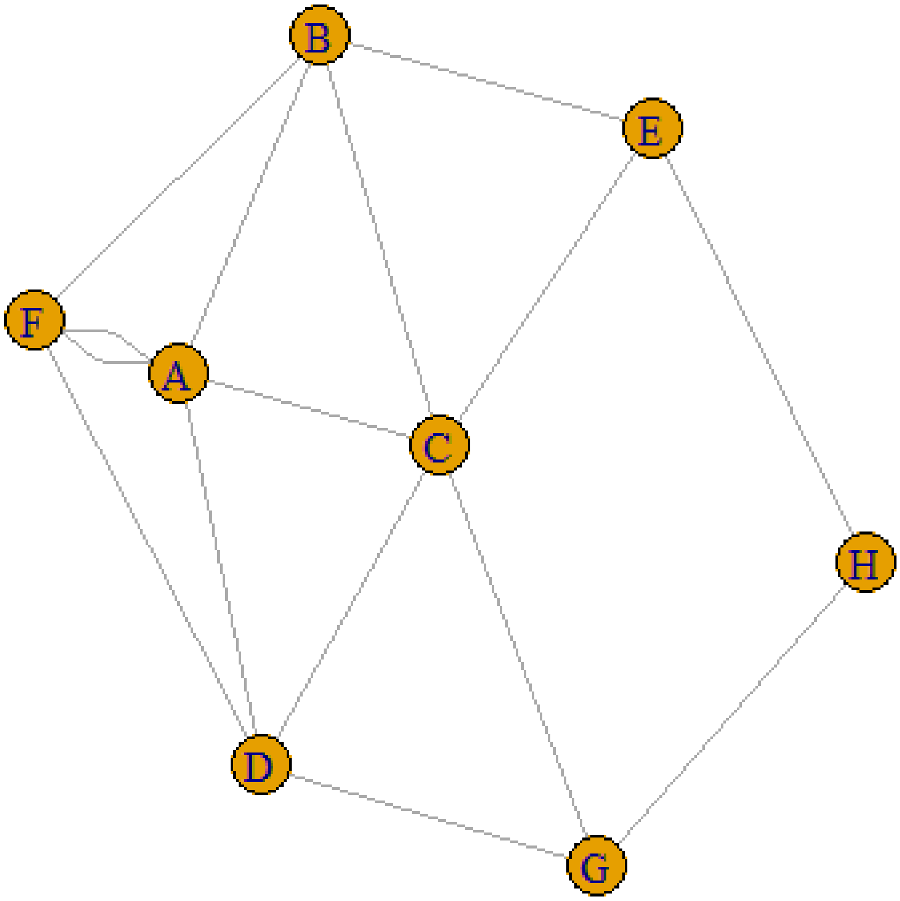
```
#undirected graph

g=graph.adjacency(matrix,mode="undirected",weighted=NULL)

plot.igraph(g,edge.arrow.size=0.1)

g
```



```
#create the network object

network=graph_from_incidence_matrix(matrix)

plot(network)
```



```
#reading edge

e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')

e

#create the network object

network=graph_from_data_frame(d=e,directed=F)

plot(network)
```



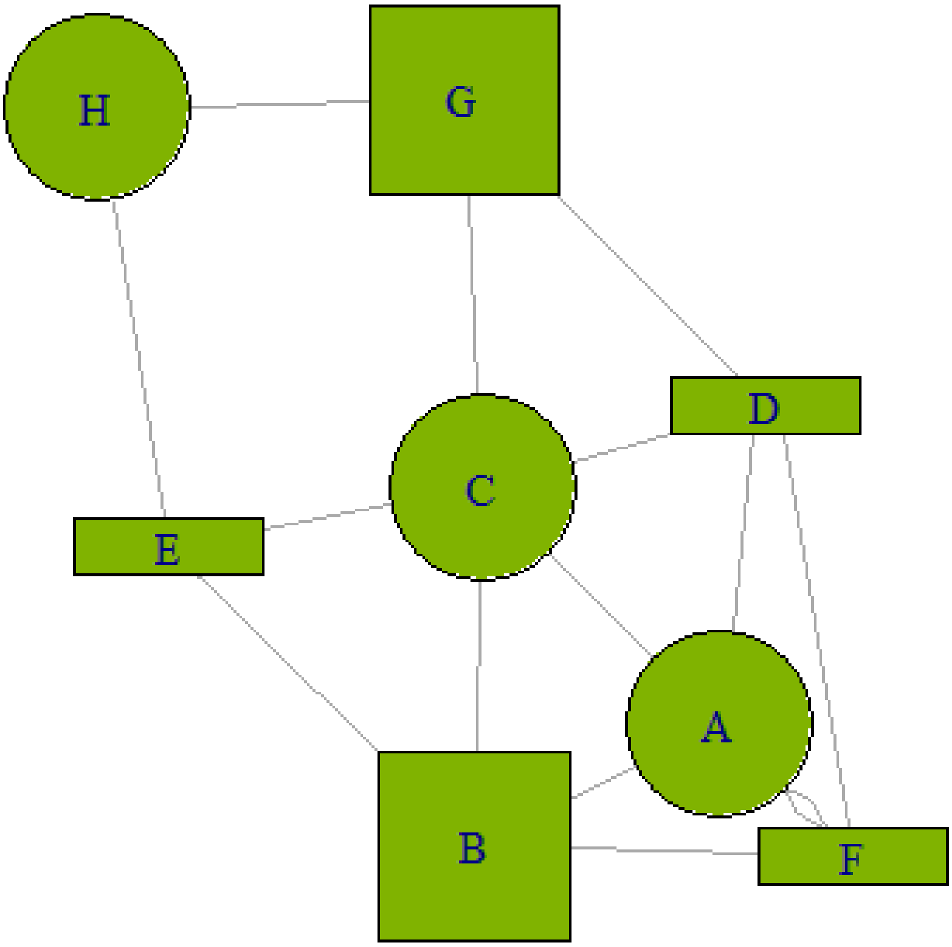
4. Create and plot the graph from the adjacency matrix and edge matrix

a. customize the vertex color, shape, size

#customize node features

#vertex customization

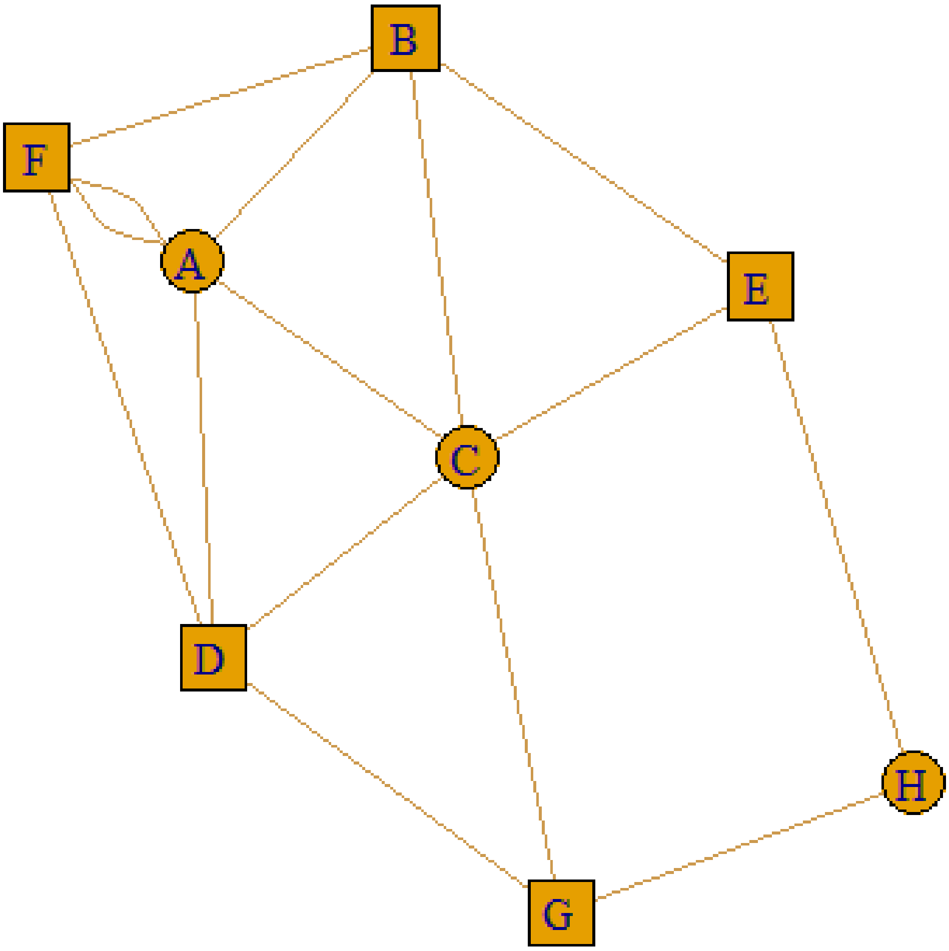
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(50))



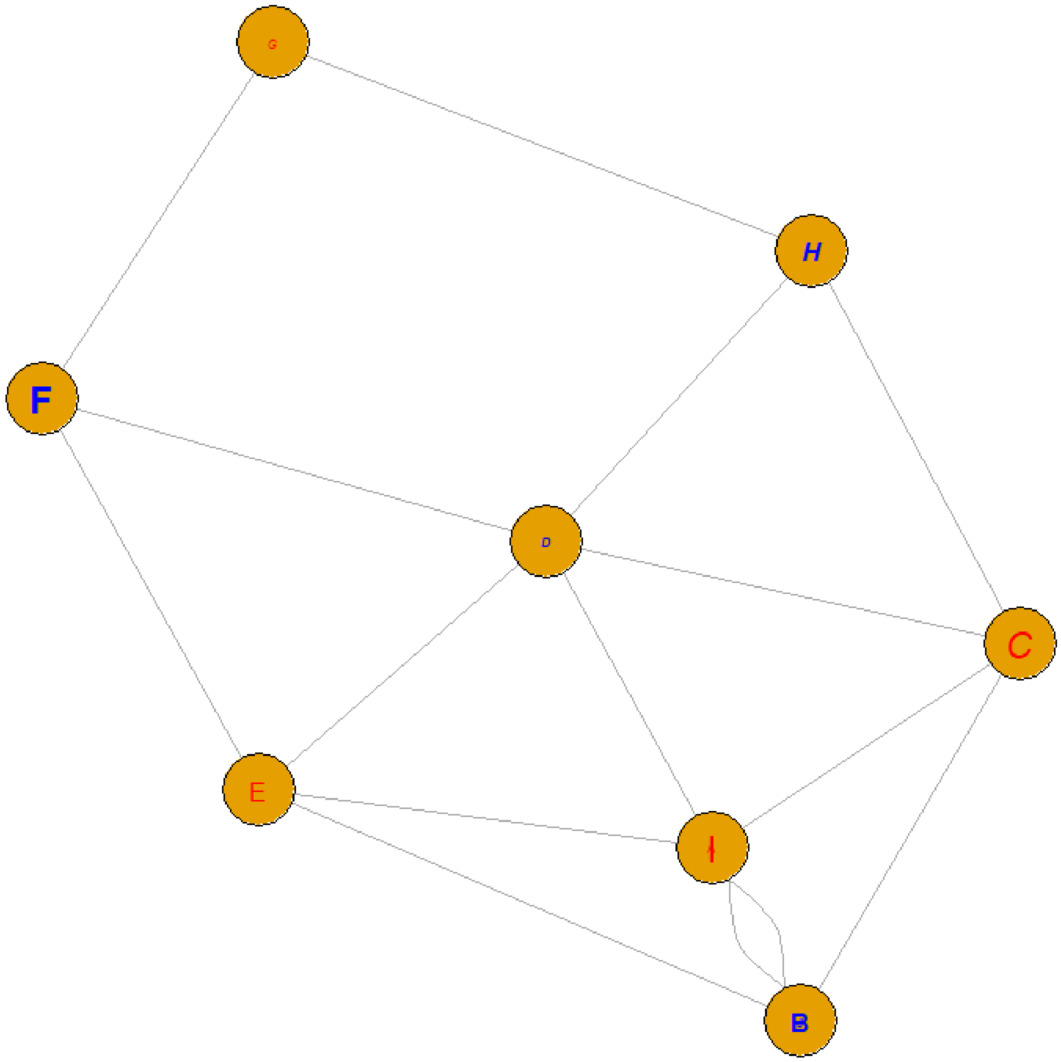
b. edge size, edge color, vertex frame and label

#edge customization

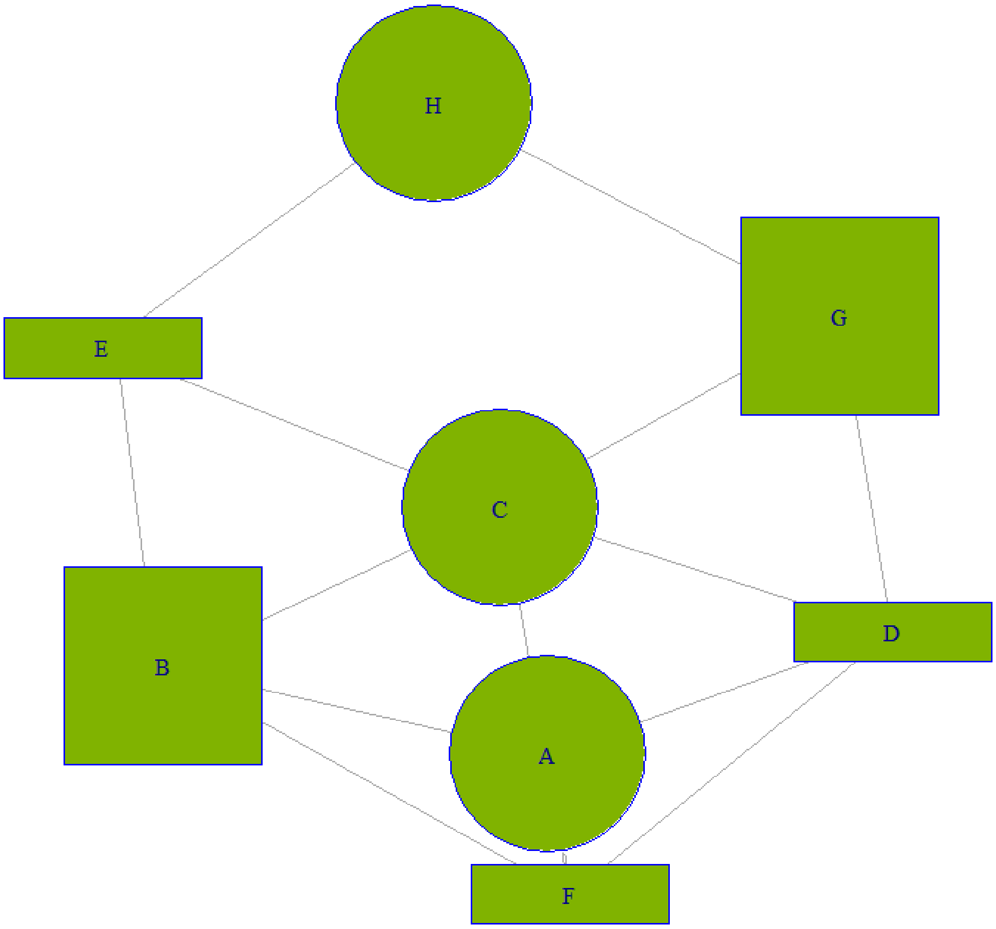
```
plot(network,edge.color=rgb(0.8,0.6,0.3),vertex.shape=c("circle","rectangle","square"),edge.size=c(8
0),edge.color="orange")
```



```
#label and frame customization
plot(network,
    vertex.label=LETTERS[1:10],              # Character vector used to label the nodes
    vertex.label.color=c("red","blue"),
    vertex.label.family="Times",             # Font family of the label (e.g.“Times”, “Helvetica”)
    vertex.label.font=c(1,2,3,4),            # Font: 1 plain, 2 bold, 3, italic, 4 bold italic, 5 symbol
    vertex.label.cex=c(0.5,1,1.5),           # Font size (multiplication factor, device-dependent)
    vertex.label.dist=0,                     # Distance between the label and the vertex
    vertex.label.degree=0 ,                  # The position of the label in relation to the vertex (use pi)
)
```

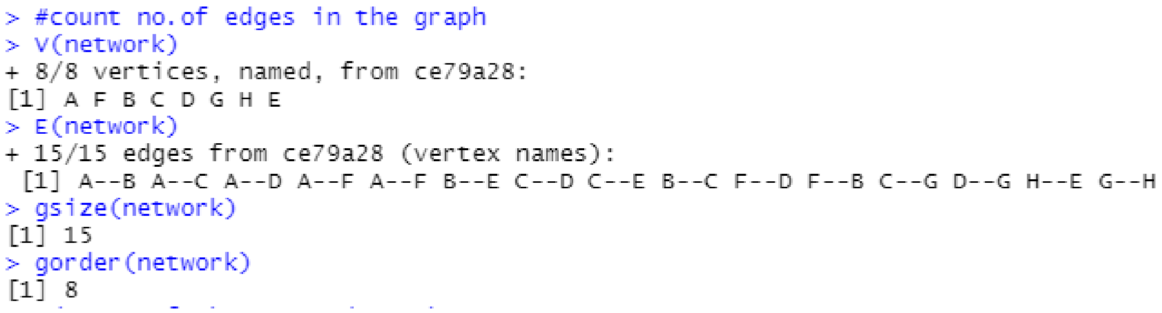


```
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(50),vertex.frame.color="blue")
```



5. Display the name of vertices, edges, no. of vertices and edges

```
V(network)

E(network)

gsize(network)

gorder(network)
```



```
> #count no.of edges in the graph
> V(network)
+ 8/8 vertices, named, from ce79a28:
[1] A F B C D G H E
> E(network)
+ 15/15 edges from ce79a28 (vertex names):
 [1] A--B A--C A--D A--F A--F B--E C--D C--E B--C F--D F--B C--G D--G H--E G--H
> gsize(network)
[1] 15
> gorder(network)
[1] 8
```

6. Find the degree of each vertex, min and max degree of the created graph

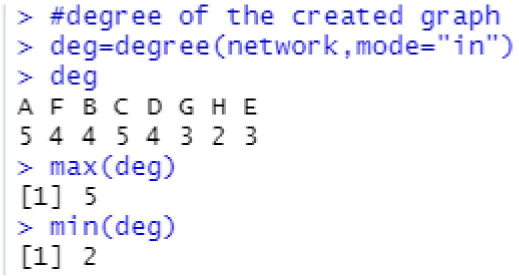
#degree of the created graph

deg=degree(network,mode="in")

deg

max(deg)

min(deg)



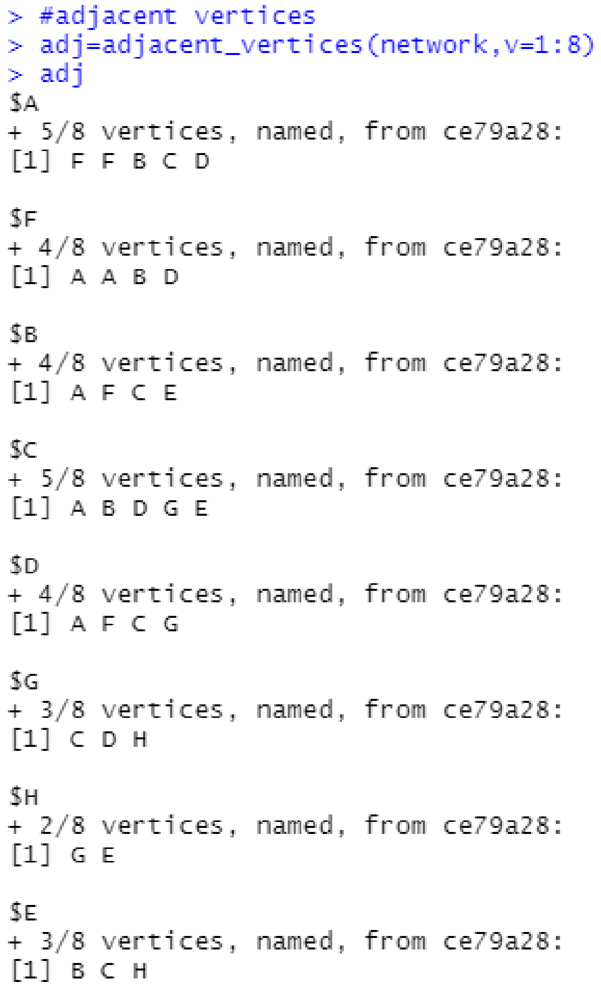
```
> #degree of the created graph
> deg=degree(network,mode="in")
> deg
A F B C D G H E 
5 4 4 5 4 3 2 3 
> max(deg)
[1] 5
> min(deg)
[1] 2
```

7. Display the adjacency vertices of each vertex(individual) in the created gap.

#adjacent vertices

adj=adjacent_vertices(network,v=1:6)

adj



```
> #adjacent vertices
> adj=adjacent_vertices(network,v=1:8)
> adj
$A
+ 5/8 vertices, named, from ce79a28:
[1] F F B C D

$F
+ 4/8 vertices, named, from ce79a28:
[1] A A B D

$B
+ 4/8 vertices, named, from ce79a28:
[1] A F C E

$C
+ 5/8 vertices, named, from ce79a28:
[1] A B D G E

$D
+ 4/8 vertices, named, from ce79a28:
[1] A F C G

$G
+ 3/8 vertices, named, from ce79a28:
[1] C D H

$H
+ 2/8 vertices, named, from ce79a28:
[1] G E

$E
+ 3/8 vertices, named, from ce79a28:
[1] B C H
```

ENTIRE R SCRIPT:

```
library(igraph)
m=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/adjacency.csv')
print(m)
matrix=as.matrix(m)
print(matrix)
#directed graph
g=graph.adjacency(matrix,mode="directed",weighted=NULL)
plot.igraph(g,edge.arrow.size=0.1)
g
#undirected graph
g=graph.adjacency(matrix,mode="undirected",weighted=NULL)
plot.igraph(g,edge.arrow.size=0.1)
g
#create the network object
network=graph_from_incidence_matrix(matrix)
plot(network)
#reading edge
e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')
e
#create the network object
network=graph_from_data_frame(d=e,directed=F)
plot(network)
#customize node features
#vertex customization
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(50))
#edge customization
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(50),edge.color="pink")
```

```
#label and frame customization
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(50),vertex.frame.color="blue")
#count no.of edges in the graph
V(network)
E(network)
gsize(network)
gorder(network)
#degree of the created graph
deg=degree(network,mode="in")
deg
max(deg)
min(deg)
#adjacent vertices
adj=adjacent_vertices(network,v=1:8)
adj
```

console:

```
> library(igraph)
> m=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/adjacency.csv')
> print(m)
  ï..A B C D E F G H
1    1 0 1 1 2 2 1 2
2    0 0 1 0 1 0 1 2
3    1 0 0 0 2 2 1 1
4    2 1 1 1 1 3 2 2
5    0 1 1 2 1 1 2 2
6    2 1 1 1 3 1 2 2
7    0 2 2 2 2 1 3 3
8    2 2 2 3 2 3 3 3
> matrix=as.matrix(m)
> print(matrix)
```

```
    ï..A B C D E F G H
[1,]    1 0 1 1 2 2 1 2
[2,]    0 0 1 0 1 0 1 2
[3,]    1 0 0 0 2 2 1 1
[4,]    2 1 1 1 1 3 2 2
[5,]    0 1 1 2 1 1 2 2
[6,]    2 1 1 1 3 1 2 2
[7,]    0 2 2 2 2 1 3 3
[8,]    2 2 2 3 2 3 3 3
> #directed graph
> g=graph.adjacency(matrix,mode="directed",weighted=NULL)
> plot.igraph(g,edge.arrow.size=0.1)
> g
IGRAPH ce532ee DN-- 8 93 --
+ attr: name (v/c)
+ edges from ce532ee (vertex names):
 [1] ï..A->ï..A ï..A->C    ï..A->D    ï..A->E    ï..A->E    ï..A->F    ï..A->F    ï..A->G    ï..A->H    ï..A->H    B   ->C
[12] B   ->E    B   ->G    B   ->H    B   ->H    C   ->ï..A C   ->E    C   ->E    C   ->F    C   ->F    C   ->G    C   ->H
[23] D   ->ï..A D   ->ï..A D   ->B    D   ->C    D   ->D    D   ->E    D   ->F    D   ->F    D   ->F    D   ->G    D   ->G
[34] D   ->H    D   ->H    E   ->B    E   ->C    E   ->D    E   ->D    E   ->E    E   ->F    E   ->G    E   ->G    E   ->H
[45] E   ->H    F   ->ï..A F   ->ï..A F   ->B    F   ->C    F   ->D    F   ->E    F   ->E    F   ->E    F   ->F    F   ->G
[56] F   ->G    F   ->H    F   ->H    G   ->B    G   ->B    G   ->C    G   ->C    G   ->D    G   ->D    G   ->E    G   ->E
[67] G   ->F    G   ->G    G   ->G    G   ->G    G   ->H    G   ->H    G   ->H    H   ->ï..A H   ->ï..A H   ->B    H   ->B
[78] H   ->C    H   ->C    H   ->D    H   ->D    H   ->D    H   ->E    H   ->E    H   ->F    H   ->F    H   ->F    H   ->G
+ ... omitted several edges
> #undirected graph
> g=graph.adjacency(matrix,mode="undirected",weighted=NULL)
> plot.igraph(g,edge.arrow.size=0.1)
> g
IGRAPH ce5ef3a UN-- 8 62 --
+ attr: name (v/c)
+ edges from ce5ef3a (vertex names):
 [1] ï..A--ï..A ï..A--C   ï..A--D   ï..A--D   ï..A--E   ï..A--E   ï..A--F   ï..A--F   ï..A--G   ï..A--H   ï..A--H
[12] B   --C   B   --D   B   --E   B   --F   B   --G   B   --G   B   --H   B   --H   C   --D   C   --E   C   --E
[23] C   --F   C   --F   C   --G   C   --G   C   --H   C   --H   D   --D   D   --E   D   --E   D   --F   D   --F
[34] D   --F   D   --G   D   --G   D   --H   D   --H   D   --H   E   --E   E   --F   E   --F   E   --F   E   --G
[45] E   --G   E   --H   E   --H   F   --F   F   --G   F   --G   F   --H   F   --H   F   --H   G   --G   G   --G
[56] G   --G   G   --H   G   --H   G   --H   H   --H   H   --H   H   --H
> #create the network object
> network=graph_from_incidence_matrix(matrix)
> plot(network)
> #reading edge
> e=read.csv('C:\\Users\\aryam\\Desktop\\Fall Sem 2021\\Data Visualization Lab\\LAB 8 28-9-21/edges.csv')
> e
  ï..Source Target
1         A      B
2         A      C
3         A      D
4         A      F
5         F      A
6         B      E
7         C      D
8         C      E
9         B      C
10        D      F
11        F      B
12        G      C
13        G      D
14        H      E
15        H      G
> #create the network object
> network=graph_from_data_frame(d=e,directed=F)
> plot(network)
> #customize node features
> #vertex customization
>
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(
50))
> #edge customization
>
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(
50),edge.color="pink")
> #label and frame customization
>
plot(network,vertex.color=rgb(0.5,0.7,0),vertex.shape=c("circle","rectangle","square"),vertex.size=c(
50),vertex.frame.color="blue")
> #count no.of edges in the graph
> V(network)
+ 8/8 vertices, named, from ce79a28:
[1] A F B C D G H E
> E(network)
+ 15/15 edges from ce79a28 (vertex names):
 [1] A--B A--C A--D A--F A--F B--E C--D C--E B--C F--D F--B C--G D--G H--E G--H
> gsize(network)
[1] 15
> gorder(network)
[1] 8
> #degree of the created graph
> deg=degree(network,mode="in")
> deg
A F B C D G H E
5 4 4 5 4 3 2 3
> max(deg)
[1] 5
```

```
> min(deg)
[1] 2
> #adjacent vertices
> adj=adjacent_vertices(network,v=1:6)
> adj
$A
+ 5/8 vertices, named, from ce79a28:
[1] F F B C D

$F
+ 4/8 vertices, named, from ce79a28:
[1] A A B D

$B
+ 4/8 vertices, named, from ce79a28:
[1] A F C E

$C
+ 5/8 vertices, named, from ce79a28:
[1] A B D G E

$D
+ 4/8 vertices, named, from ce79a28:
[1] A F C G

$G
+ 3/8 vertices, named, from ce79a28:
[1] C D H

> #adjacent vertices
> adj=adjacent_vertices(network,v=1:9)
```

```
Error in adjacent_vertices(network, v = 1:9) :
  At iterators.c:759 : Cannot create iterator, invalid vertex id, Invalid vertex id
> adj
$A
+ 5/8 vertices, named, from ce79a28:
[1] F F B C D

$F
+ 4/8 vertices, named, from ce79a28:
[1] A A B D

$B
+ 4/8 vertices, named, from ce79a28:
[1] A F C E

$C
+ 5/8 vertices, named, from ce79a28:
[1] A B D G E

$D
+ 4/8 vertices, named, from ce79a28:
[1] A F C G

$G
+ 3/8 vertices, named, from ce79a28:
[1] C D H

> #adjacent vertices
> adj=adjacent_vertices(network,v=1:8)
> adj
$A
```

```
+ 5/8 vertices, named, from ce79a28:
[1] F F B C D


$F
+ 4/8 vertices, named, from ce79a28:
[1] A A B D


$B
+ 4/8 vertices, named, from ce79a28:
[1] A F C E


$C
+ 5/8 vertices, named, from ce79a28:
[1] A B D G E


$D
+ 4/8 vertices, named, from ce79a28:
[1] A F C G


$G
+ 3/8 vertices, named, from ce79a28:
[1] C D H


$H
+ 2/8 vertices, named, from ce79a28:
[1] G E


$E
+ 3/8 vertices, named, from ce79a28:
[1] B C H
```

CONCLUSION:EXERCISE PROLEMS HAVE BEEN SUCCESFULLY EXECUTED.