Digital Assingment-1

****

Course Code: CSE3021 **Course Title:** Social and Information Networks

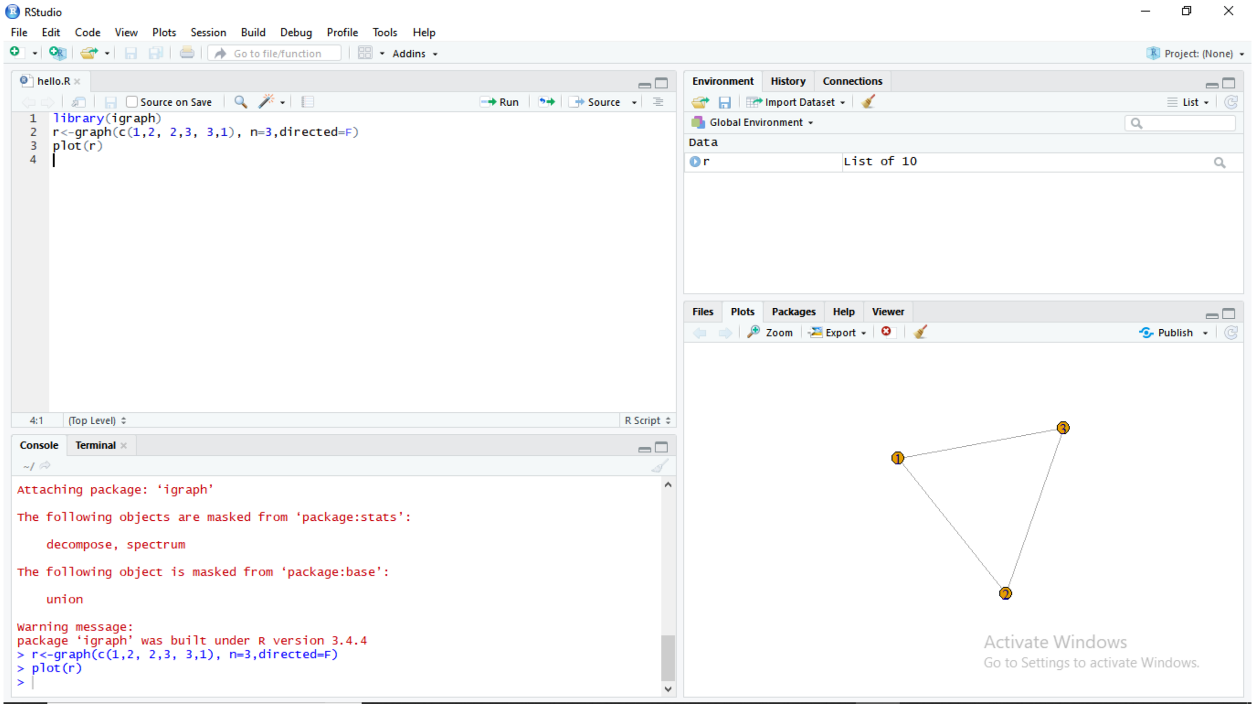
****

Reg. No.: 16BCE0880 Name: Rahul Trivedi

****

Question a): Construction of a directed graph and undirected graph with nodes (plot function).

Undirected:

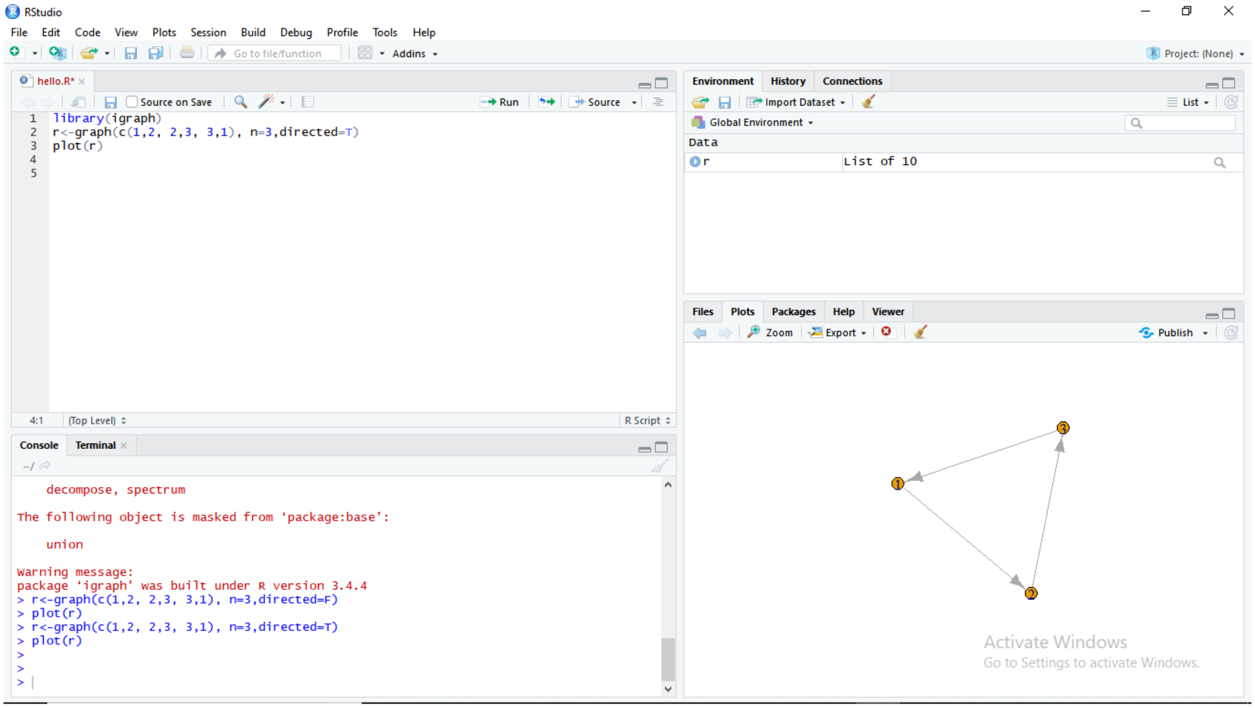


```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=F)
plot(r)
```

```
Data
r                     List of 10
```

```
Attaching package: ‘igraph’

The following objects are masked from ‘package:stats’:

    decompose, spectrum

The following object is masked from ‘package:base’:

    union

Warning message:
package ‘igraph’ was built under R version 3.4.4
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=F)
> plot(r)
>
```

Directed:

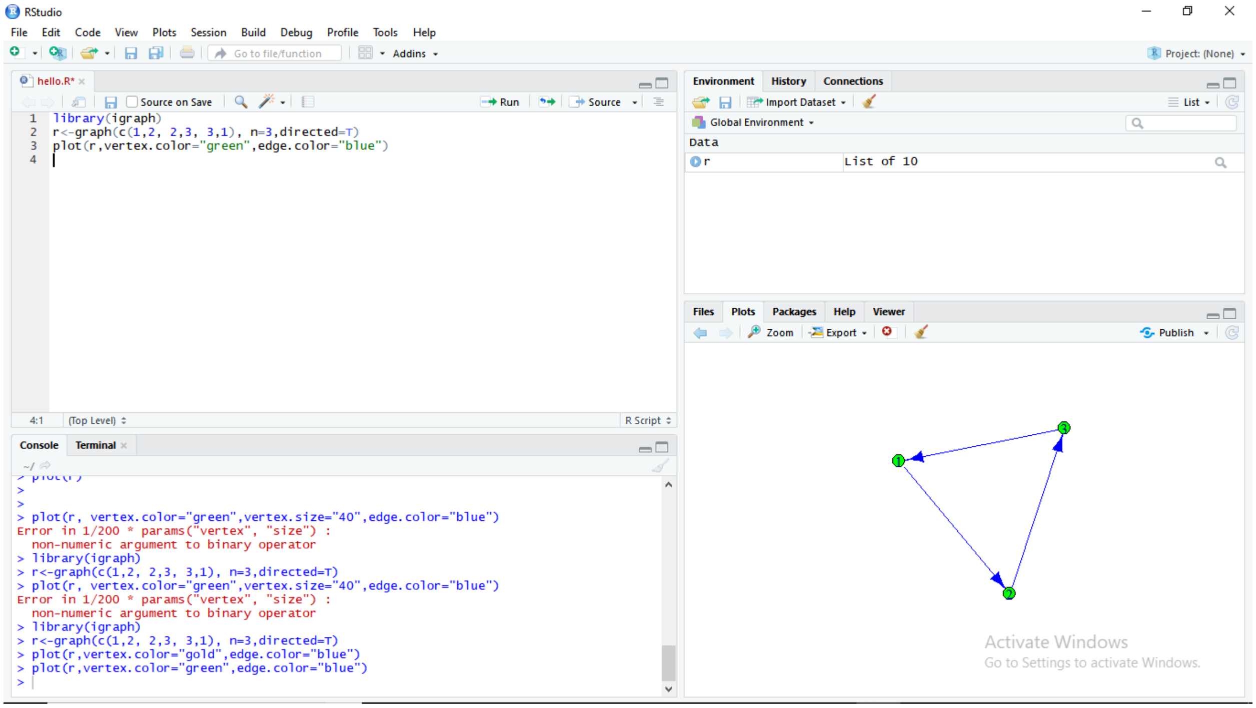


```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r)
```

```
Data
r                     List of 10
```

```
    decompose, spectrum

The following object is masked from ‘package:base’:

    union

Warning message:
package ‘igraph’ was built under R version 3.4.4
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=F)
> plot(r)
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r)
>
>
>
```

Question b) Colour the edges and nodes(plot).



```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
```

Data

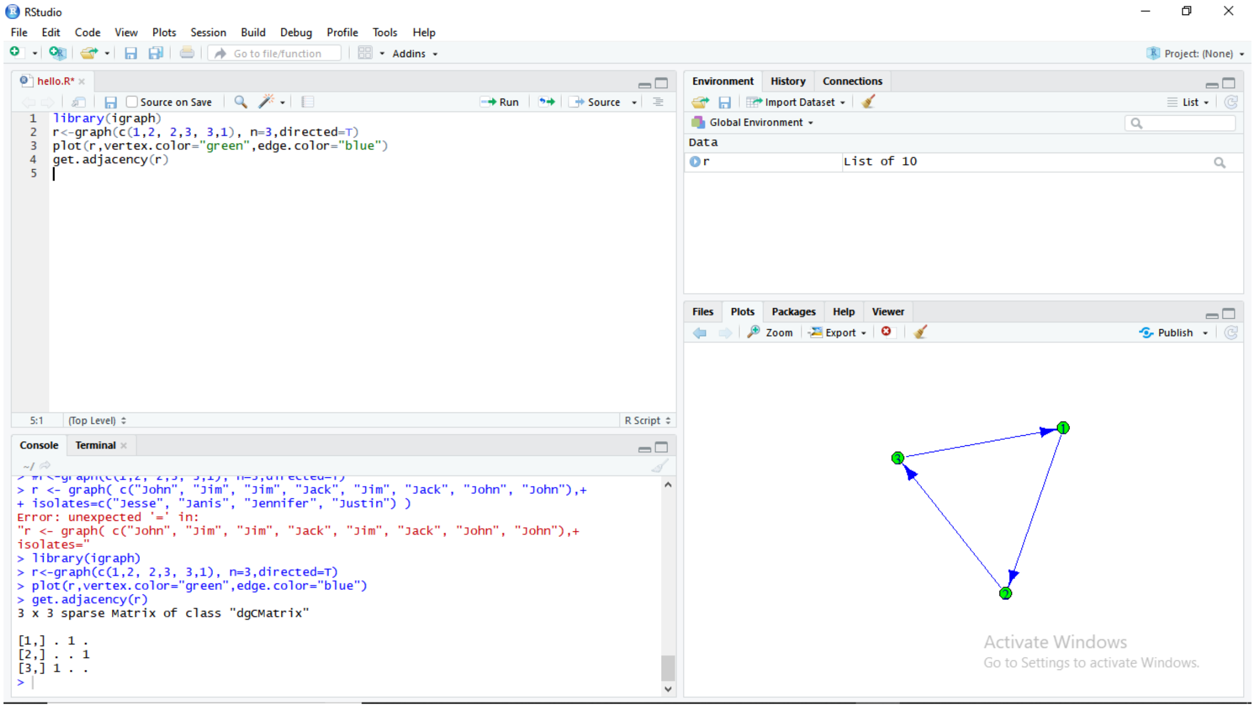
r List of 10

```
> plot(r)
>
>
> plot(r, vertex.color="green",vertex.size="40",edge.color="blue")
Error in 1/200 * params("vertex", "size") :
  non-numeric argument to binary operator
> library(igraph)
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r, vertex.color="green",vertex.size="40",edge.color="blue")
Error in 1/200 * params("vertex", "size") :
  non-numeric argument to binary operator
> library(igraph)
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r,vertex.color="gold",edge.color="blue")
> plot(r,vertex.color="green",edge.color="blue")
> 
```

Question c) Name the nodes.

The graph is already a named graph.

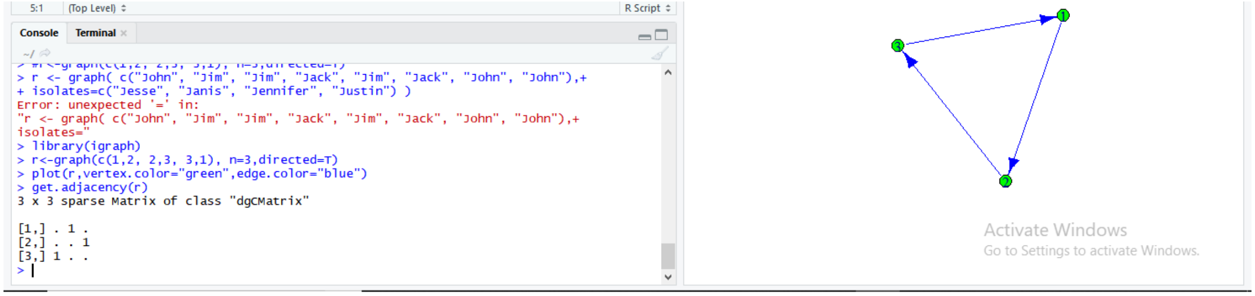
Question d) Print adjacency matrix of undirected graph.



```r
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
```

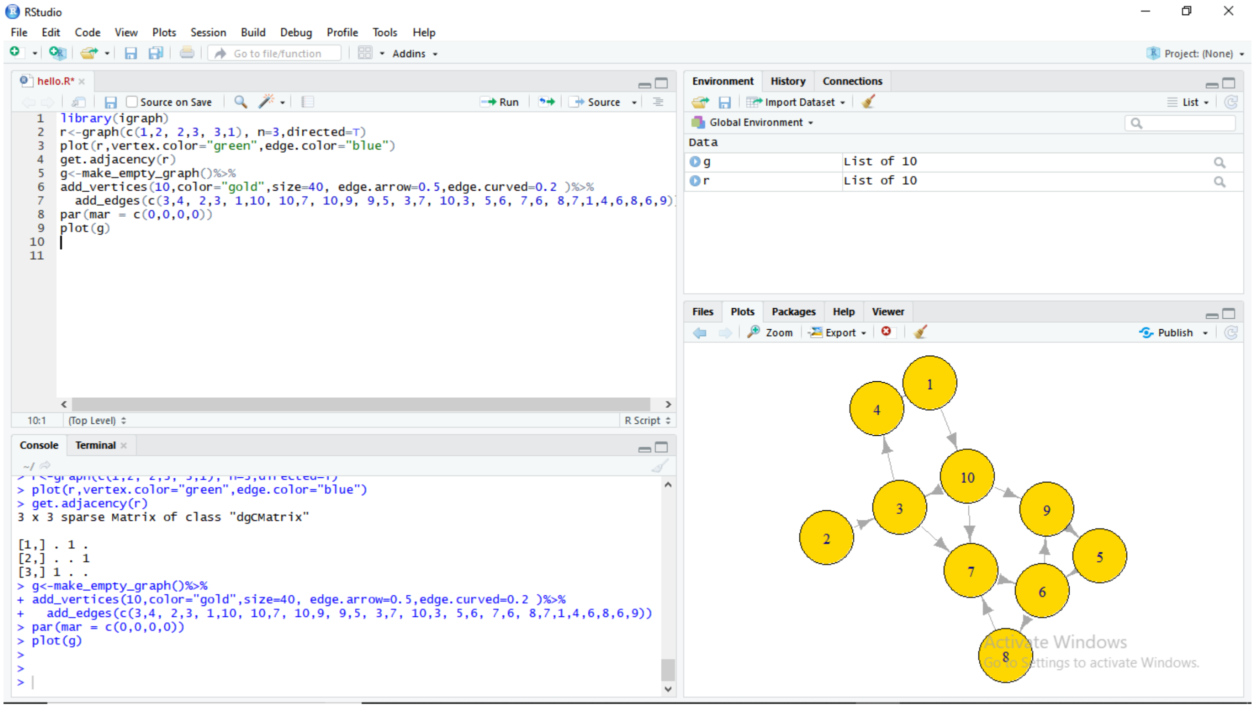
```
r                         List of 10
```

```
> #r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> r <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"),+
+ isolates=c("Jesse", "Janis", "Jennifer", "Justin") )
Error: unexpected '=' in:
"r <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"),+
isolates="
> library(igraph)
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r,vertex.color="green",edge.color="blue")
> get.adjacency(r)
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] . 1 .
[2,] . . 1
[3,] 1 . .
> 
```



```
> #r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> r <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"),+
+ isolates=c("Jesse", "Janis", "Jennifer", "Justin") )
Error: unexpected '=' in:
"r <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"),+
isolates="
> library(igraph)
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r,vertex.color="green",edge.color="blue")
> get.adjacency(r)
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] . 1 .
[2,] . . 1
[3,] 1 . .
> 
```

Question e) Add few extra nodes to the network and name them as well.



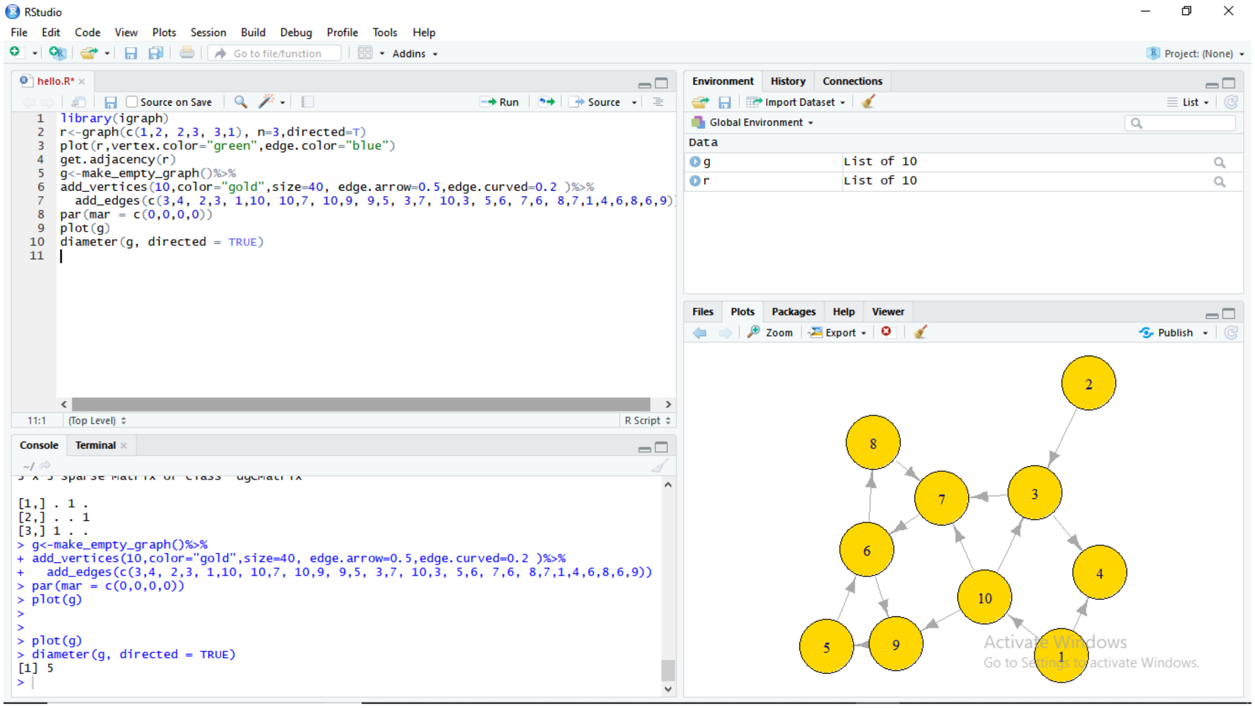
```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
g<-make_empty_graph()%>%
add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9)
par(mar = c(0,0,0,0))
plot(g)
```

Data

```
g    List of 10
r    List of 10
```

```
> r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
> plot(r,vertex.color="green",edge.color="blue")
> get.adjacency(r)
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] . 1 .
[2,] . . 1
[3,] 1 . .
> g<-make_empty_graph()%>%
+ add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
+   add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9))
> par(mar = c(0,0,0,0))
> plot(g)
>
>
> |
```

Question f) Print diameter of graph solution.

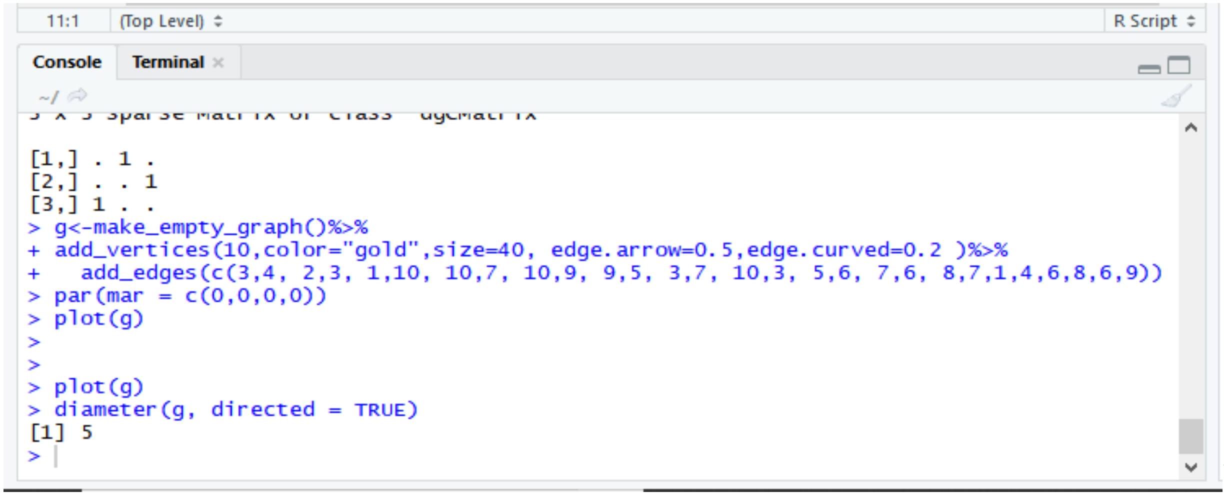


```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
g<-make_empty_graph()%>%
add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9)
par(mar = c(0,0,0,0))
plot(g)
diameter(g, directed = TRUE)
```

Data

```
g    List of 10
r    List of 10
```

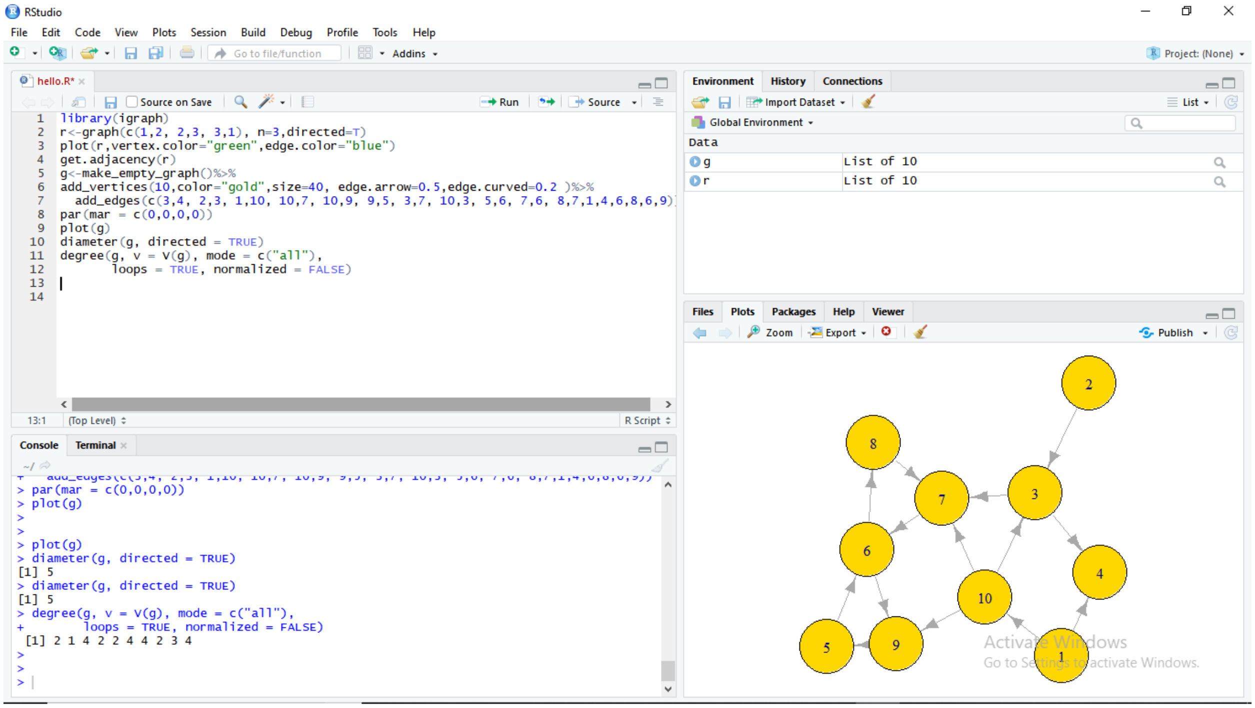
```
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] . 1 .
[2,] . . 1
[3,] 1 . .
> g<-make_empty_graph()%>%
+ add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
+   add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9))
> par(mar = c(0,0,0,0))
> plot(g)
>
>
> plot(g)
> diameter(g, directed = TRUE)
[1] 5
> |
```



```
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] . 1 .
[2,] . . 1
[3,] 1 . .
> g<-make_empty_graph()%>%
+ add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
+   add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9))
> par(mar = c(0,0,0,0))
> plot(g)
>
>
> plot(g)
> diameter(g, directed = TRUE)
[1] 5
> 
```

\

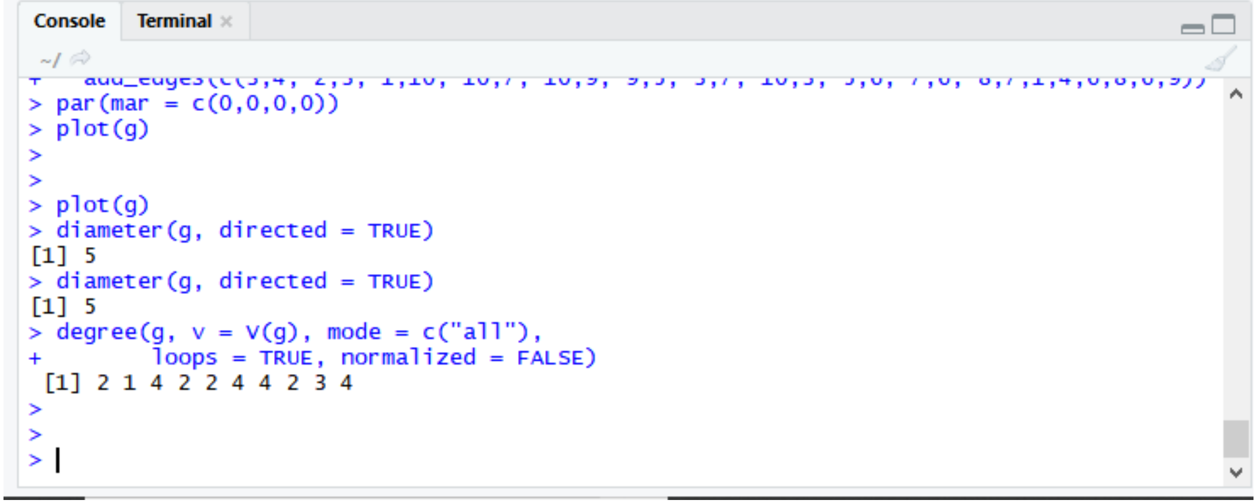
Question g) Find degree of all nodes.



```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
g<-make_empty_graph()%>%
add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9)
par(mar = c(0,0,0,0))
plot(g)
diameter(g, directed = TRUE)
degree(g, v = V(g), mode = c("all"),
       loops = TRUE, normalized = FALSE)
```

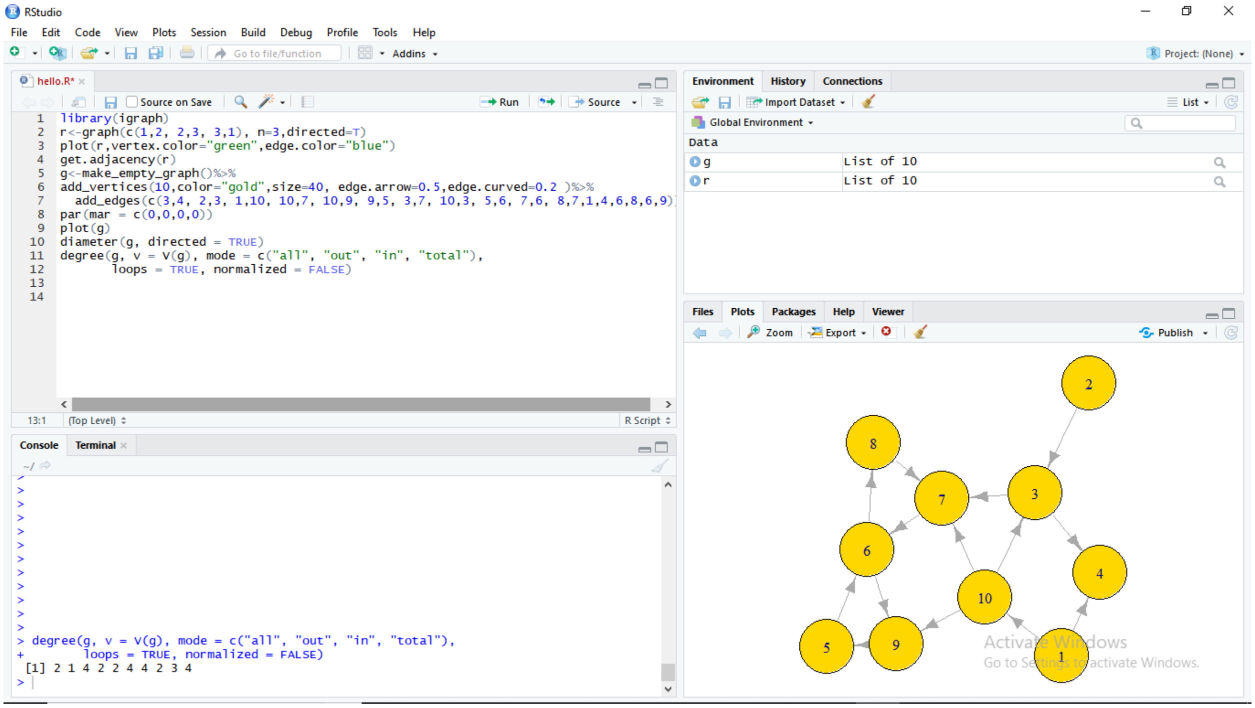
```
Data
g    List of 10
r    List of 10
```

```
+   add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9))
> par(mar = c(0,0,0,0))
> plot(g)
>
>
> plot(g)
> diameter(g, directed = TRUE)
[1] 5
> diameter(g, directed = TRUE)
[1] 5
> degree(g, v = V(g), mode = c("all"),
+        loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
>
>
> 
```



```
+   add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9))
> par(mar = c(0,0,0,0))
> plot(g)
>
>
> plot(g)
> diameter(g, directed = TRUE)
[1] 5
> diameter(g, directed = TRUE)
[1] 5
> degree(g, v = V(g), mode = c("all"),
+         loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
>
>
> |
```

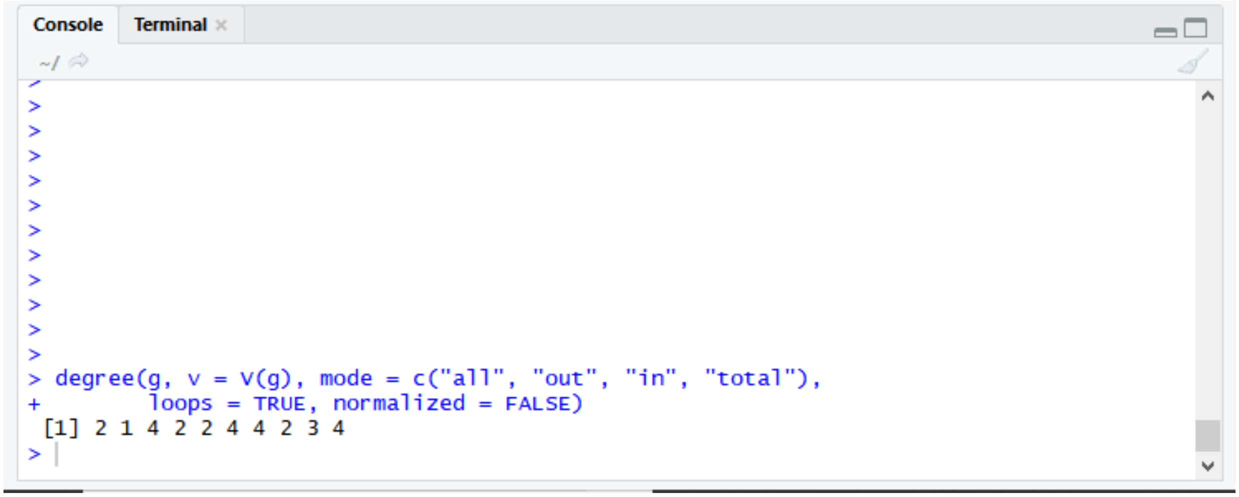
Question h) Find in-degrees of all nodes and out degrees of all nodes.



```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
g<-make_empty_graph()%>%
add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9)
par(mar = c(0,0,0,0))
plot(g)
diameter(g, directed = TRUE)
degree(g, v = V(g), mode = c("all", "out", "in", "total"),
        loops = TRUE, normalized = FALSE)
```

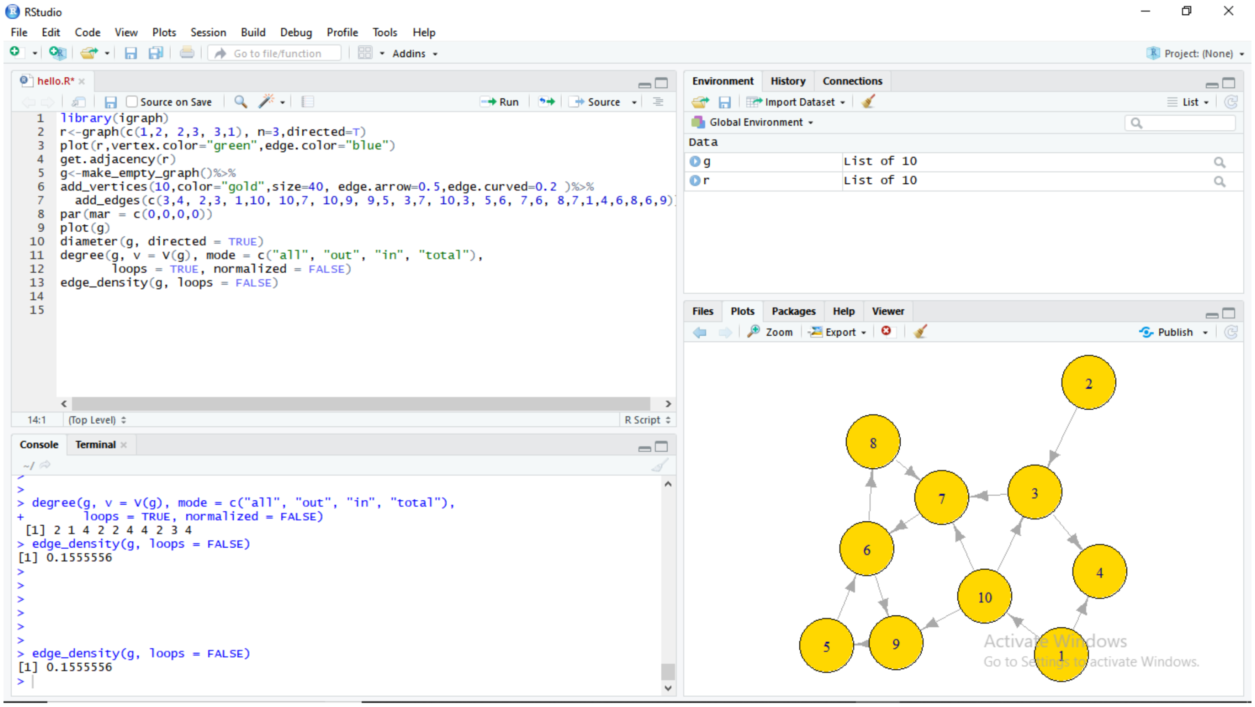
```
Data
g    List of 10
r    List of 10
```

```
>
>
>
>
>
>
>
>
>
>
>
> degree(g, v = V(g), mode = c("all", "out", "in", "total"),
+         loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
> |
```



```
>
>
>
>
>
>
>
>
>
>
>
>
> degree(g, v = V(g), mode = c("all", "out", "in", "total"),
+         loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
> 
```

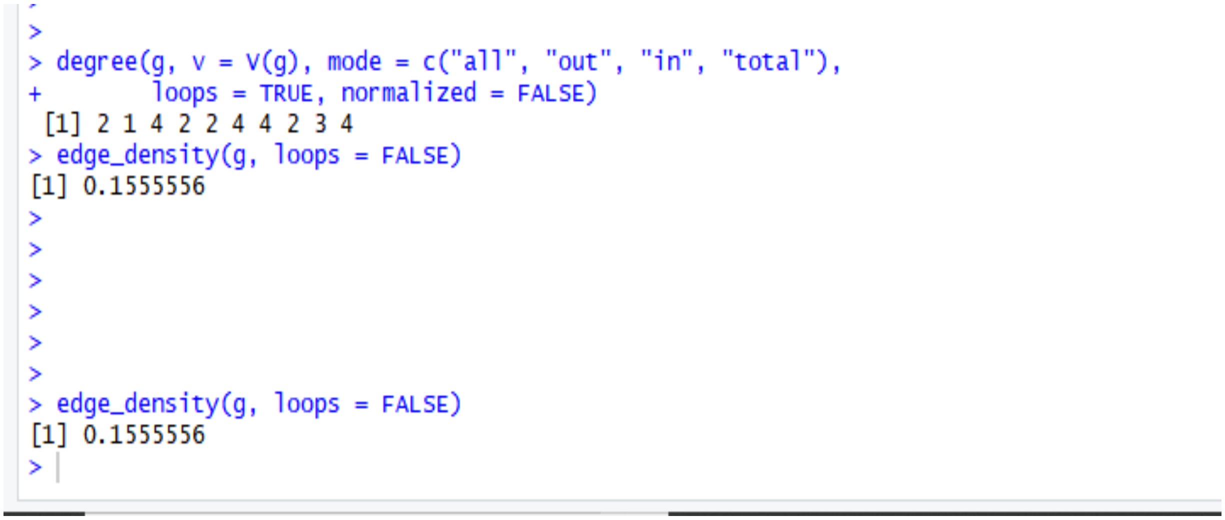
Question i) Find density of any nodes.



```
library(igraph)
r<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(r,vertex.color="green",edge.color="blue")
get.adjacency(r)
g<-make_empty_graph()%>%
add_vertices(10,color="gold",size=40, edge.arrow=0.5,edge.curved=0.2 )%>%
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6, 7,6, 8,7,1,4,6,8,6,9)
par(mar = c(0,0,0,0))
plot(g)
diameter(g, directed = TRUE)
degree(g, v = V(g), mode = c("all", "out", "in", "total"),
       loops = TRUE, normalized = FALSE)
edge_density(g, loops = FALSE)
```

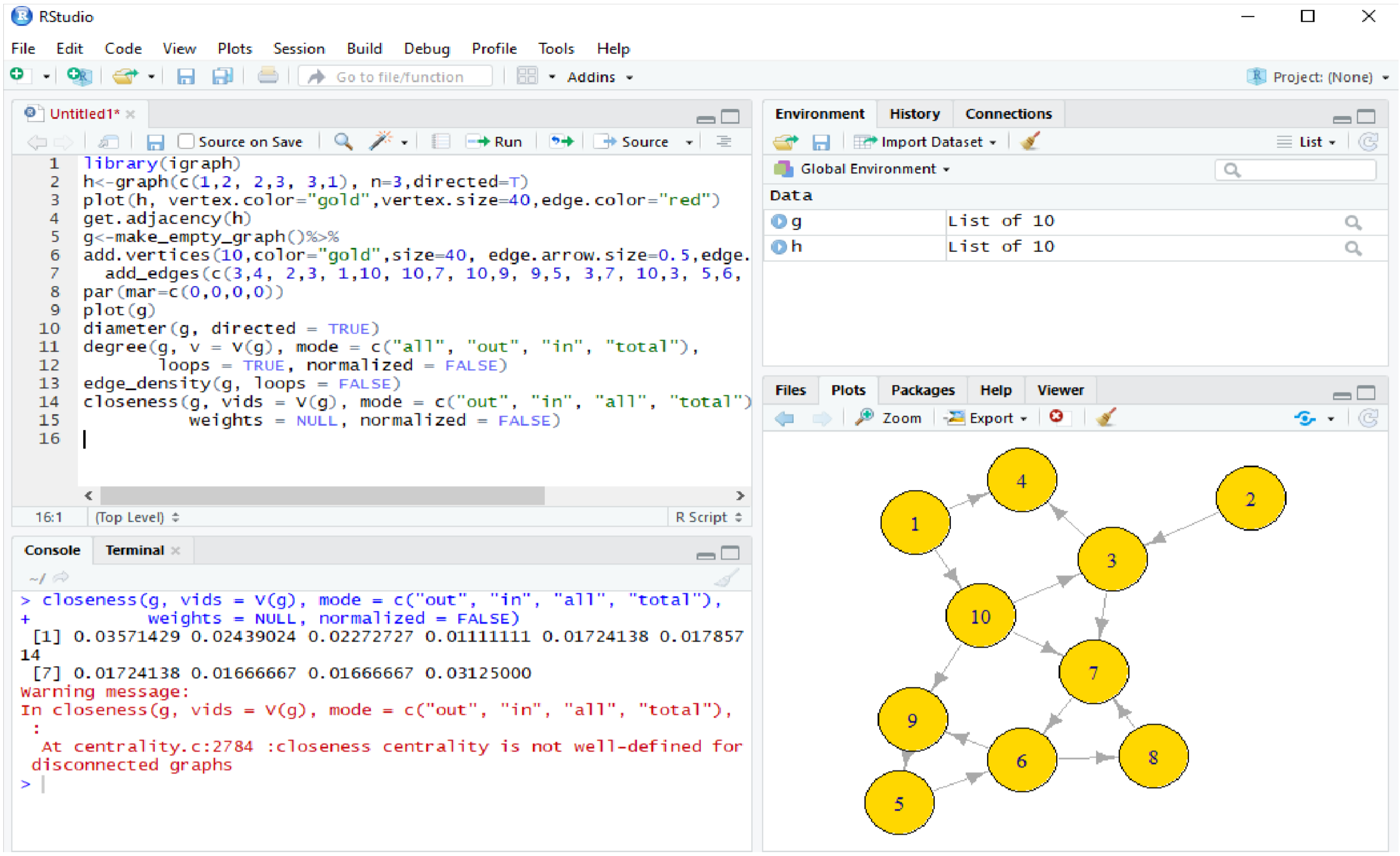
```
Data
g                    List of 10
r                    List of 10
```

```
>
> degree(g, v = V(g), mode = c("all", "out", "in", "total"),
+         loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
> edge_density(g, loops = FALSE)
[1] 0.1555556
>
>
>
>
>
>
> edge_density(g, loops = FALSE)
[1] 0.1555556
> 
```



```
>
> degree(g, v = V(g), mode = c("all", "out", "in", "total"),
+         loops = TRUE, normalized = FALSE)
 [1] 2 1 4 2 2 4 4 2 3 4
> edge_density(g, loops = FALSE)
[1] 0.1555556
>
>
>
>
>
>
> edge_density(g, loops = FALSE)
[1] 0.1555556
> 
```

Question j) Find closeness centrality of all nodes.

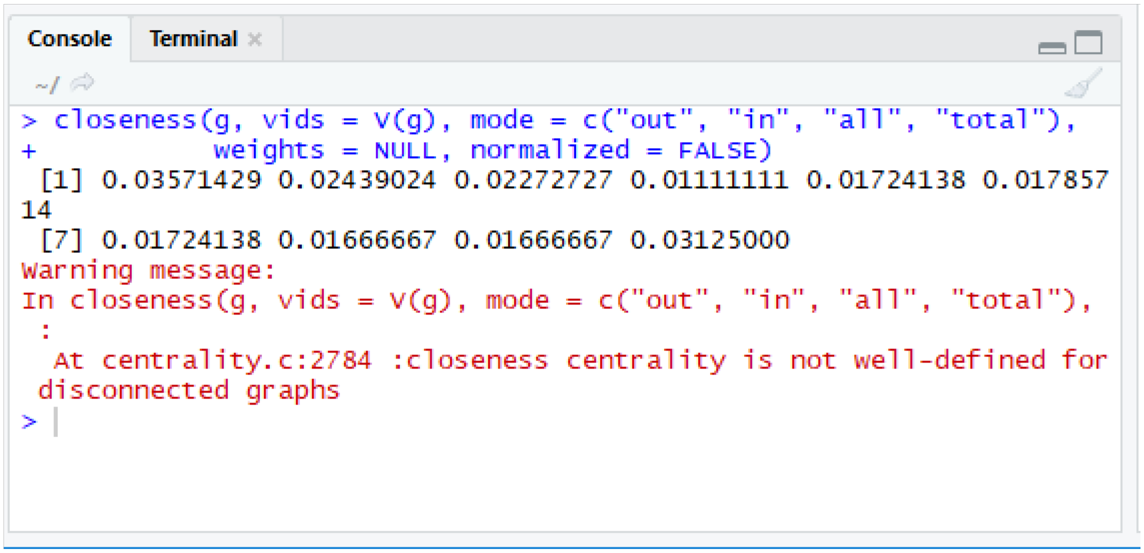


```r
library(igraph)
h<-graph(c(1,2, 2,3, 3,1), n=3,directed=T)
plot(h, vertex.color="gold",vertex.size=40,edge.color="red")
get.adjacency(h)
g<-make_empty_graph()%>%
add.vertices(10,color="gold",size=40, edge.arrow.size=0.5,edge.
  add_edges(c(3,4, 2,3, 1,10, 10,7, 10,9, 9,5, 3,7, 10,3, 5,6,
par(mar=c(0,0,0,0))
plot(g)
diameter(g, directed = TRUE)
degree(g, v = V(g), mode = c("all", "out", "in", "total"),
          loops = TRUE, normalized = FALSE)
edge_density(g, loops = FALSE)
closeness(g, vids = V(g), mode = c("out", "in", "all", "total")
              weights = NULL, normalized = FALSE)
```

Data

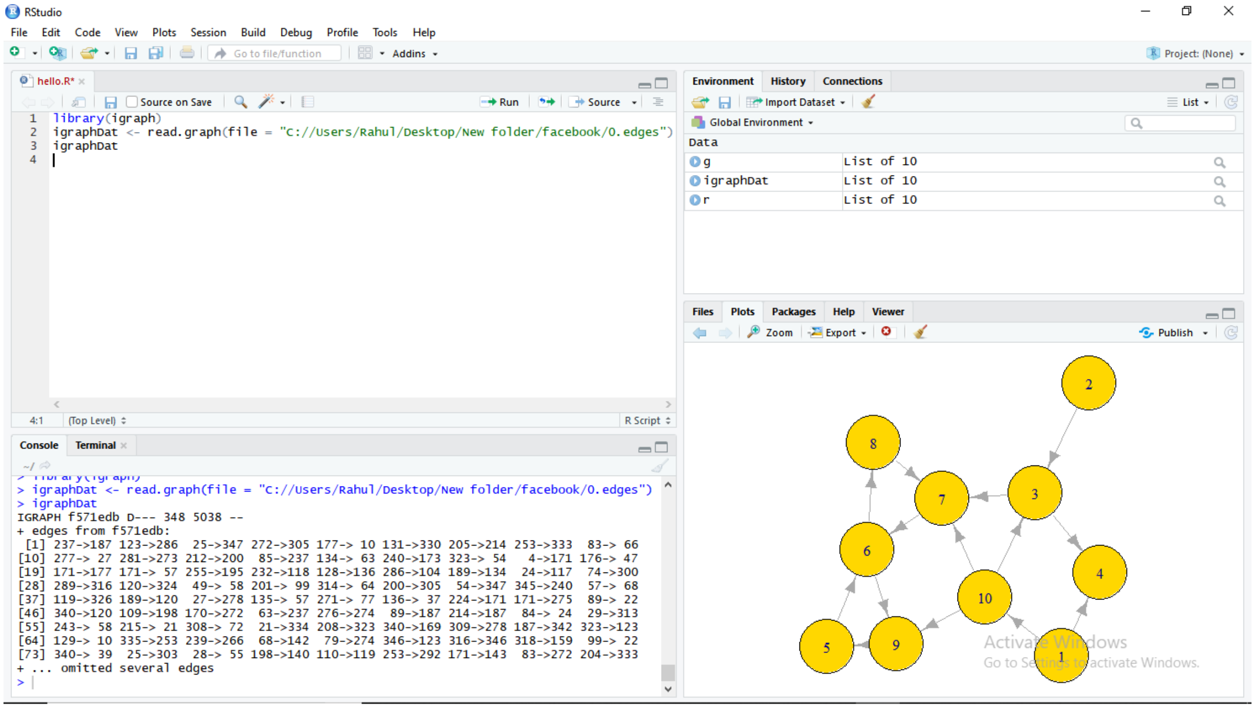
g	List of 10
h	List of 10

```
~/
> closeness(g, vids = V(g), mode = c("out", "in", "all", "total"),
+               weights = NULL, normalized = FALSE)
 [1] 0.03571429 0.02439024 0.02272727 0.01111111 0.01724138 0.017857
14
 [7] 0.01724138 0.01666667 0.01666667 0.03125000
Warning message:
In closeness(g, vids = V(g), mode = c("out", "in", "all", "total"),
 :
  At centrality.c:2784 :closeness centrality is not well-defined for
 disconnected graphs
> 
```



```
~/
> closeness(g, vids = V(g), mode = c("out", "in", "all", "total"),
+               weights = NULL, normalized = FALSE)
 [1] 0.03571429 0.02439024 0.02272727 0.01111111 0.01724138 0.017857
14
 [7] 0.01724138 0.01666667 0.01666667 0.03125000
Warning message:
In closeness(g, vids = V(g), mode = c("out", "in", "all", "total"),
 :
  At centrality.c:2784 :closeness centrality is not well-defined for
 disconnected graphs
> 
```

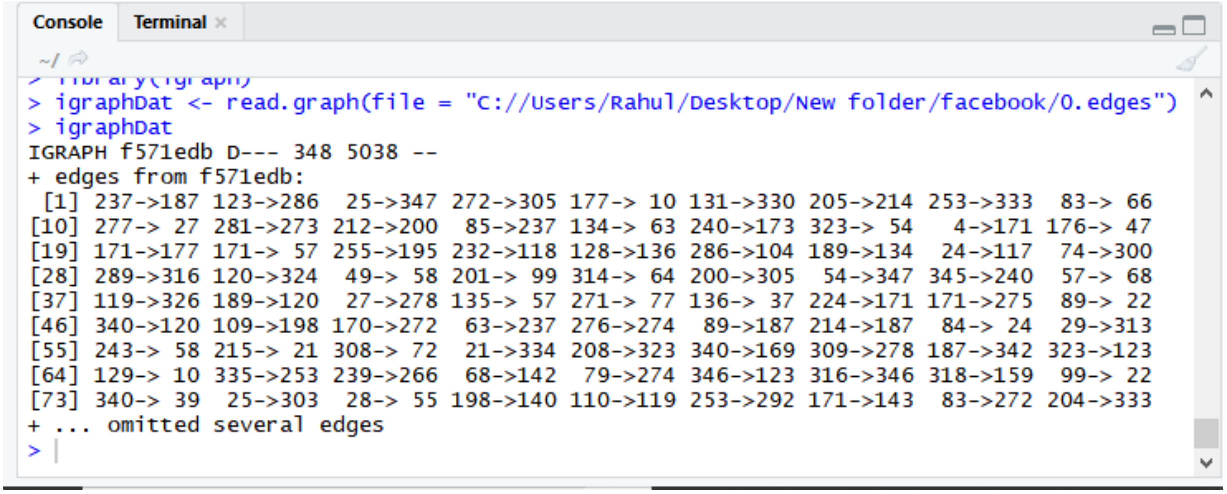
Question k) Create network from a given data set. You can choose any one of the data sets from the following link. https://snap.stanford.edu/data/



```
library(igraph)
igraphDat <- read.graph(file = "C://Users/Rahul/Desktop/New folder/facebook/0.edges")
igraphDat
```

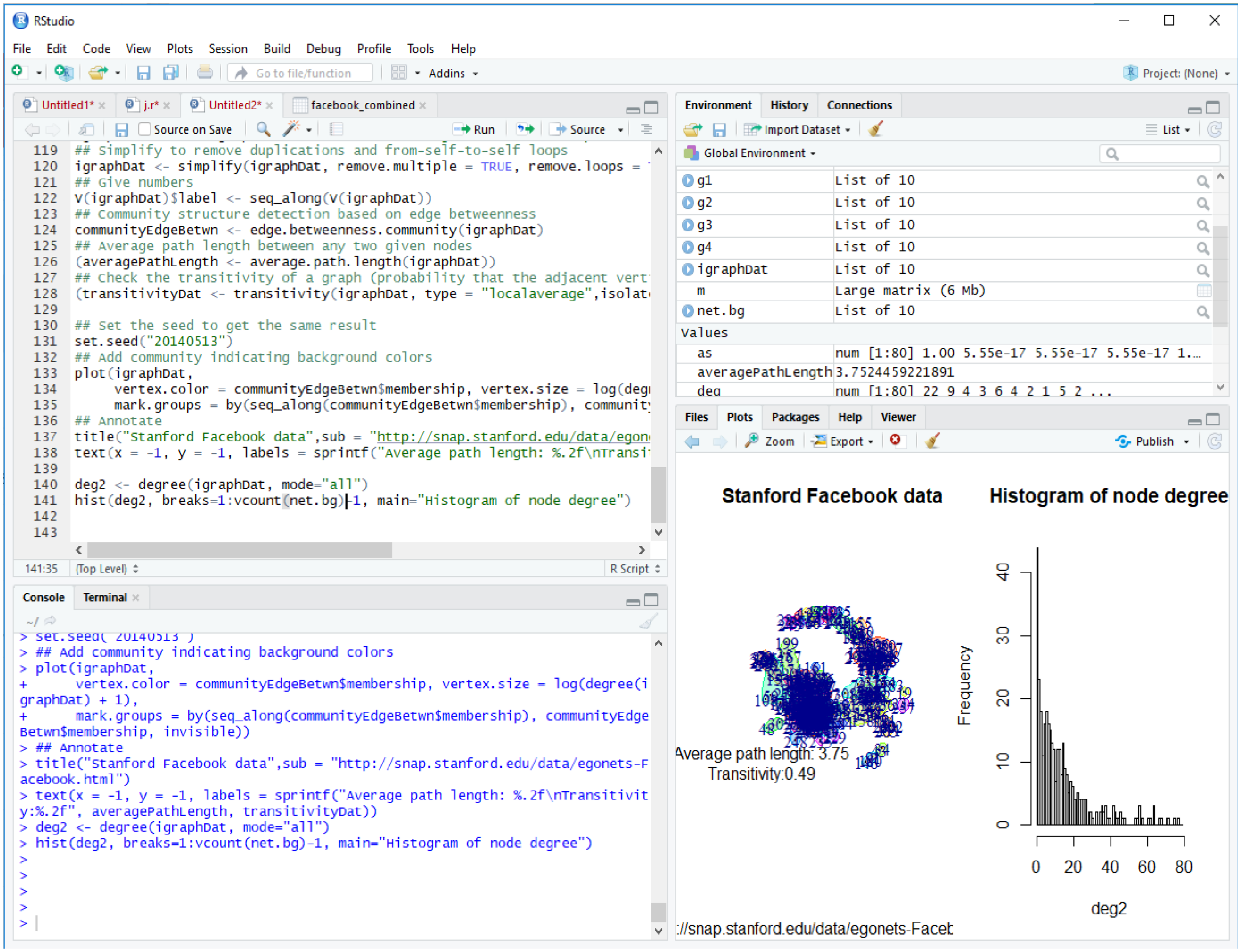
Data	
g	List of 10
igraphDat	List of 10
r	List of 10

```
> library(igraph)
> igraphDat <- read.graph(file = "C://Users/Rahul/Desktop/New folder/facebook/0.edges")
> igraphDat
IGRAPH f571edb D--- 348 5038 --
+ edges from f571edb:
 [1] 237->187 123->286  25->347 272->305 177-> 10 131->330 205->214 253->333  83-> 66
[10] 277-> 27 281->273 212->200  85->237 134-> 63 240->173 323-> 54   4->171 176-> 47
[19] 171->177 171-> 57 255->195 232->118 128->136 286->104 189->134  24->117  74->300
[28] 289->316 120->324  49-> 58 201-> 99 314-> 64 200->305  54->347 345->240  57-> 68
[37] 119->326 189->120  27->278 135-> 57 271-> 77 136-> 37 224->171 171->275  89-> 22
[46] 340->120 109->198 170->272  63->237 276->274  89->187 214->187  84-> 24  29->313
[55] 243-> 58 215-> 21 308-> 72  21->334 208->323 340->169 309->278 187->342 323->123
[64] 129-> 10 335->253 239->266  68->142  79->274 346->123 316->346 318->159  99-> 22
[73] 340-> 39  25->303  28-> 55 198->140 110->119 253->292 171->143  83->272 204->333
+ ... omitted several edges
> |
```



```
> library(igraph)
> igraphDat <- read.graph(file = "C://Users/Rahul/Desktop/New folder/facebook/0.edges")
> igraphDat
IGRAPH f571edb D--- 348 5038 --
+ edges from f571edb:
 [1] 237->187 123->286  25->347 272->305 177-> 10 131->330 205->214 253->333  83-> 66
[10] 277-> 27 281->273 212->200  85->237 134-> 63 240->173 323-> 54   4->171 176-> 47
[19] 171->177 171-> 57 255->195 232->118 128->136 286->104 189->134  24->117  74->300
[28] 289->316 120->324  49-> 58 201-> 99 314-> 64 200->305  54->347 345->240  57-> 68
[37] 119->326 189->120  27->278 135-> 57 271-> 77 136-> 37 224->171 171->275  89-> 22
[46] 340->120 109->198 170->272  63->237 276->274  89->187 214->187  84-> 24  29->313
[55] 243-> 58 215-> 21 308-> 72  21->334 208->323 340->169 309->278 187->342 323->123
[64] 129-> 10 335->253 239->266  68->142  79->274 346->123 316->346 318->159  99-> 22
[73] 340-> 39  25->303  28-> 55 198->140 110->119 253->292 171->143  83->272 204->333
+ ... omitted several edges
> |
```

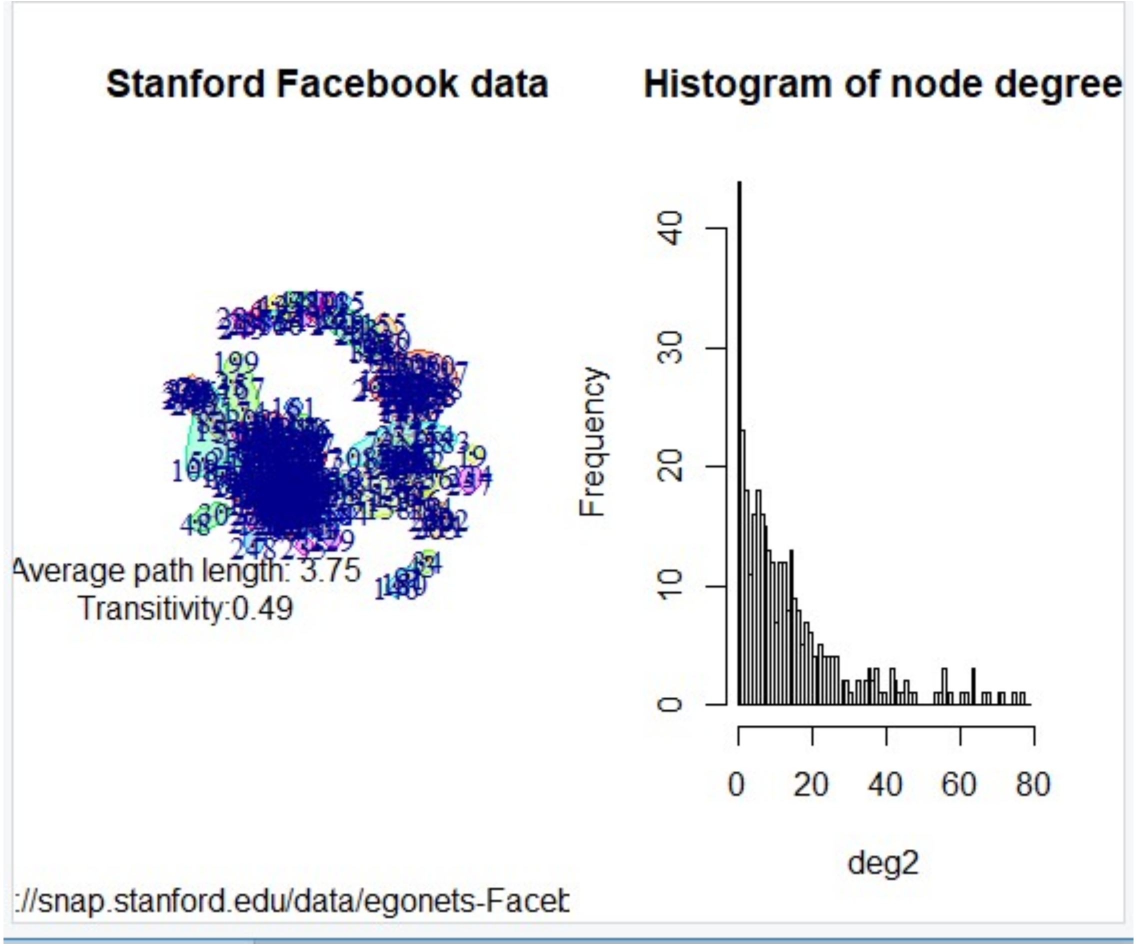
Question l) Prepare a histogram of ‘Frequency' vs 'Degree of Vertices'.



```
## Simplify to remove duplications and from-self-to-self loops
igraphDat <- simplify(igraphDat, remove.multiple = TRUE, remove.loops = 
## Give numbers
V(igraphDat)$label <- seq_along(V(igraphDat))
## Community structure detection based on edge betweenness
communityEdgeBetwn <- edge.betweenness.community(igraphDat)
## Average path length between any two given nodes
(averagePathLength <- average.path.length(igraphDat))
## Check the transitivity of a graph (probability that the adjacent vert
(transitivityDat <- transitivity(igraphDat, type = "localaverage",isolat

## Set the seed to get the same result
set.seed("20140513")
## Add community indicating background colors
plot(igraphDat,
     vertex.color = communityEdgeBetwn$membership, vertex.size = log(deg
     mark.groups = by(seq_along(communityEdgeBetwn$membership), communit
## Annotate
title("Stanford Facebook data",sub = "http://snap.stanford.edu/data/egon
text(x = -1, y = -1, labels = sprintf("Average path length: %.2f\nTransi

deg2 <- degree(igraphDat, mode="all")
hist(deg2, breaks=1:vcount(net.bg)-1, main="Histogram of node degree")
```

```
> set.seed("20140513")
> ## Add community indicating background colors
> plot(igraphDat,
+      vertex.color = communityEdgeBetwn$membership, vertex.size = log(degree(i
graphDat) + 1),
+      mark.groups = by(seq_along(communityEdgeBetwn$membership), communityEdge
Betwn$membership, invisible))
> ## Annotate
> title("Stanford Facebook data",sub = "http://snap.stanford.edu/data/egonets-F
acebook.html")
> text(x = -1, y = -1, labels = sprintf("Average path length: %.2f\nTransitivit
y:%.2f", averagePathLength, transitivityDat))
> deg2 <- degree(igraphDat, mode="all")
> hist(deg2, breaks=1:vcount(net.bg)-1, main="Histogram of node degree")
>
>
>
>
> 
```

g1	List of 10
g2	List of 10
g3	List of 10
g4	List of 10
igraphDat	List of 10
m	Large matrix (6 Mb)
net.bg	List of 10
Values	
as	num [1:80] 1.00 5.55e-17 5.55e-17 5.55e-17 1....
averagePathLength	3.7524459221891
deg	num [1:80] 22 9 4 3 6 4 2 1 5 2 ...


Stanford Facebook data
Histogram of node degree
Average path length: 3.75
Transitivity:0.49
://snap.stanford.edu/data/egonets-Facet
Frequency
0
10
20
30
40
0
20
40
60
80
deg2