



FOSSEE Summer Internship Report

on

R Studio

submitted by

Arnab Karmakar (Vellore Institute of Technology, Chennai)

under the guidance of

Prof. Mohana N

Department of Mathematics VIT, Chennai

July 23rd - August 29th , 2021

Acknowledgement

I want to express my sincere gratitude to Prof. Mohana N, Department of Mathematics VIT,Chennai, for creating the FOSSEE Internship programme and providing students an opportunity to participate in it. I would also like to thank for her immense support, patience, motivation, knowledge & influence throughout this internship and for helping me on various statistical models. I would also like to thank the other fellows who got selected along with me, namely M. S Naman for his support, intellectual discussions and enthusiasm. I am very grateful to be given such a fantastic opportunity to work on this exciting project.

Contents

1 Introduction

2 Keywords

3 Social Network Analysis

Abstract

Methodology

Data Collection

 Data Exploration

 Data Analysis

 Visualizing Social Networks

 Centrality Measures

 Finding Key Actors

 Residuals

 Plot Eigenvector Centrality on

 Betweenness

4 Results

5 Conclusion

6 References

Introduction

Every kind of social aggregation can be represented in terms of units composing this aggregation and relations between these units. This kind of representation of a social structure is called “Social Network”. In a social network, every unit, usually called “social actor” (a person, a group, an organization, a nation, a blog and so on), is represented as a node. A relation is represented as a linkage or a flow between these units. The set of possible relations is potentially infinite; the term relation can have many different meaning: acquaintance, kinship, evaluation of another person, the need of a commercial exchange, physical connections, the presence in a web-page of a link to another page and so on. Therefore, the objects under observation are not individuals and their attributes, but the relationships between individuals and their structure. The advantage of such a representation is that it permits the analysis of social processes as a product of the relationships among social entities. In Social Network Analysis we can study two different kinds of variables: structural and composition. Variables of the first type are the most important in this field because they represent the different kinds of ties between social actors (friendship, trust and so on).

Keywords

Social Network Analysis, Social Networking, Social Networking References, demographic indicators, hyperlinking networks, algebraic roles, Computer Mediated Communication, Social Networking, Tcl / Tk Network Graph, Medium Steps, Plot genvector Centrality on Betweenness, Finding Key Players.

Social Network Analysis

Abstract

Social Network Analysis is a widely used method of psychology, such as social science, economics, and other fields. What is different about this view that's it doesn't focus to individuals or to others sectors of society but by the relationship between them. In this paper my purpose is to give a general overview of this concept, providing a description of the key resources and topics covered in Network Analysis. Initially, I will focus on the methodological and systematic analysis of the analysis. In the last section, I shall show the latest lessons on the Internet Network and its relationships with Information Technologies, especially online. Lastly, I shall represent you how this method can be helpful in learning some of the web materials.

Methodology

Data Exploration

Loading Social Network Data:- Loading An Edgelist

- directed vs. undirected
- SNA data also can also be represented as a sociomatrix

```
# Social Network Analysis
```

```
library(igraph)
```

```
g <- graph(c(1,2,2,3,3,4,4,1),  
          directed = F,  
          n=7)
```

```
plot(g,  
     vertex.color = "green",  
     vertex.size = 40,  
     edge.color = 'red')
```

```
g[]
```

```
g1 <- graph(c("Amy", "Ram", "Ram", "Li", "Li", "Amy",  
             "Amy", "Li", "Kate", "Li"),  
           directed=T)
```

```
plot(g1,  
     vertex.color = "green",  
     vertex.size = 40,  
     edge.color = 'red')
```

```
g1
```

```
# Network measures
```

```
degree(g1, mode='all')
```

```
degree(g1, mode='in')
```

```
degree(g1, mode='out')
```

```
diameter(g1, directed=F, weights = NA)
```

```
edge_density(g1, loops = F)
```

```
ecount(g1)/(vcount(g1)*(vcount(g1)-1))
```

```
reciprocity(g1)
```

```
closeness(g1, mode='all', weights = NA)
```

```
betweenness(g1, directed=T, weights=NA)
```

```
edge_betweenness(g1)
```

```

*Untitled - Notepad
File Edit Format View Help
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.fruchterman.reingold)
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.graphopt)
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.kamada.kawai)

# Hub and officials
hs <- hub_score(net) $ vector
such as <- authority_score(net) $ vector
par(mfrow = c(1,2))
set.seed(123)
structure(net,
  vertex.size = hs * 30,
  main = 'Hubs',
  vertex.color = rainbow(52),
  edge.arrow.size = 0.1,
  layout = layout.kamada.kawai)
set.seed(123)
structure(net,
  vertex.size=as*30,
  main = 'Authorities',
  vertex.color = rainbow(52),
  edge.arrow.size=0.1,
  layout = layout.kamada.kawai)
par(mfrow=c(1,1))

# Community detection
net <- graph.data.frame(y, directed = F)
cnet <- cluster_edge_betweenness(net)
plot(cnet,
  net,
  vertex.size = 10,
  vertex.label.cex = 0.8)

```

```

*Untitled - Notepad
File Edit Format View Help
# Read the data file
data <- read.csv(file.choose(), header = T)
y <- data.frame(data $ first, data $ second)

# Create a network
inetha <- graph.data.frame(y, directed = T)
V(net)
E(net)
V(net) $ label <- V(net) $ name
V(net) $ degree <- degree(net)

# Histogram node degree
hist(V(net) $ degree,
  icol = 'green',
  main = 'Histogram for Node Degree',
  ylab = 'Frequency',
  xlab = 'Viewing qualifications')

# Drawing of a network
set.seed(222)
structure(net,
  vertex.color = 'green',
  vertex.size = 2,
  edge.arrow.size = 0.1,
  vertex.label.cex = 0.8)

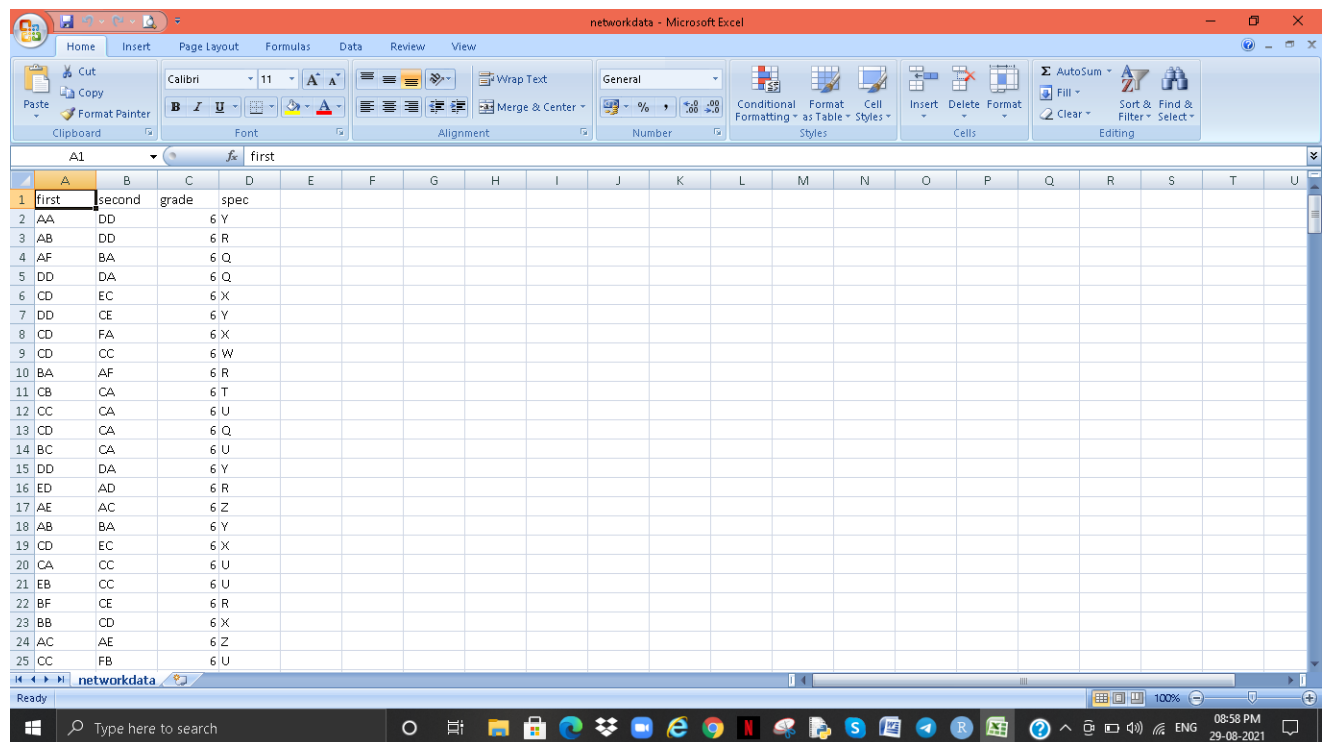
# Highlighting degrees & layouts
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.fruchterman.reingold)
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.graphopt)
plot(net,
  vertex.color = rainbow(52),
  vertex.size = V(net)$degree*0.4,
  edge.arrow.size = 0.1,
  layout=layout.kamada.kawai)

# Hub and officials

```

source target

1 2
1 10
2 1
2 10
3 7
4 7
4 209
5 132
6 150
7 3
7 4
7 9
8 106
8 115
9 1
9 2
9 7
10 1
10 2
11 133
11 218
12 88



The screenshot shows a Microsoft Excel window titled "networkdata - Microsoft Excel". The spreadsheet contains a table with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	first	second	grade	spec																	
2	AA	DD	6	Y																	
3	AB	DD	6	R																	
4	AF	BA	6	Q																	
5	DD	DA	6	Q																	
6	CD	EC	6	X																	
7	DD	CE	6	Y																	
8	CD	FA	6	X																	
9	CD	CC	6	W																	
10	BA	AF	6	R																	
11	CB	CA	6	T																	
12	CC	CA	6	U																	
13	CD	CA	6	Q																	
14	BC	CA	6	U																	
15	DD	DA	6	Y																	
16	ED	AD	6	R																	
17	AE	AC	6	Z																	
18	AB	BA	6	Y																	
19	CD	EC	6	X																	
20	CA	CC	6	U																	
21	EB	CC	6	U																	
22	BF	CE	6	R																	
23	BB	CD	6	X																	
24	AC	AE	6	Z																	
25	CC	FB	6	U																	

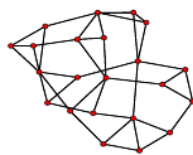
Data Analysis

Differ by Graph index

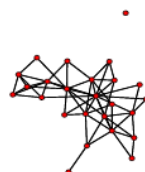
- Degree distribution
 - average node-to-node distance – average shortest path length
 - clustering coefficient – Global, local
-

network examples

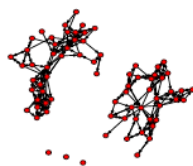
Taro Exchange



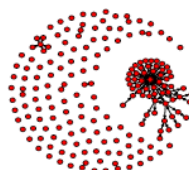
Texas SAR EMON



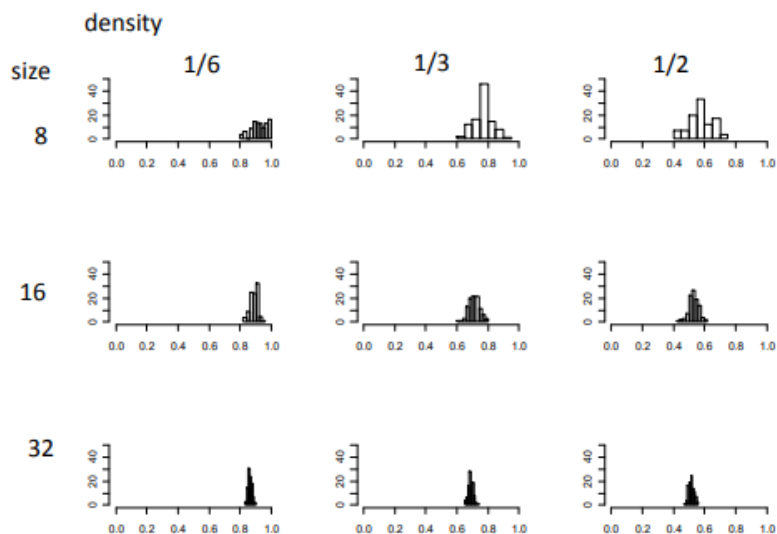
Coleman Friendship Network



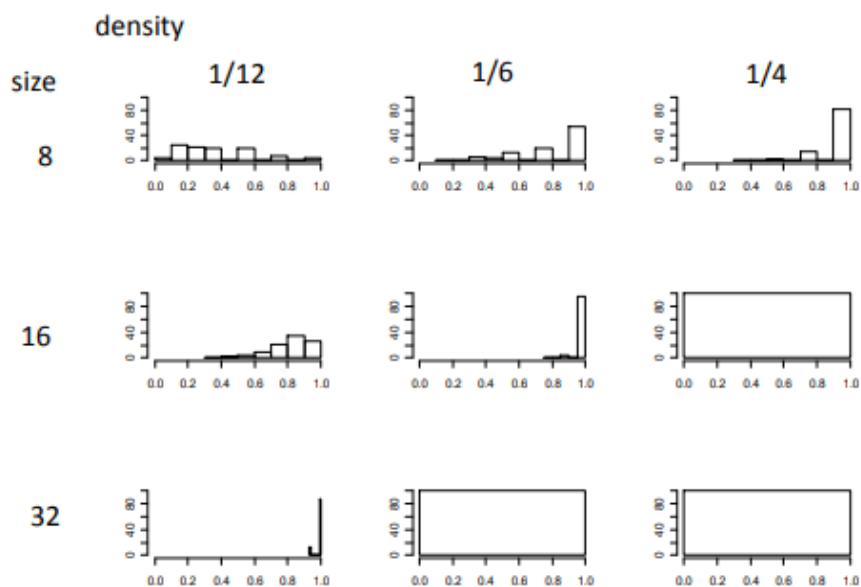
Year 2000 MIDs



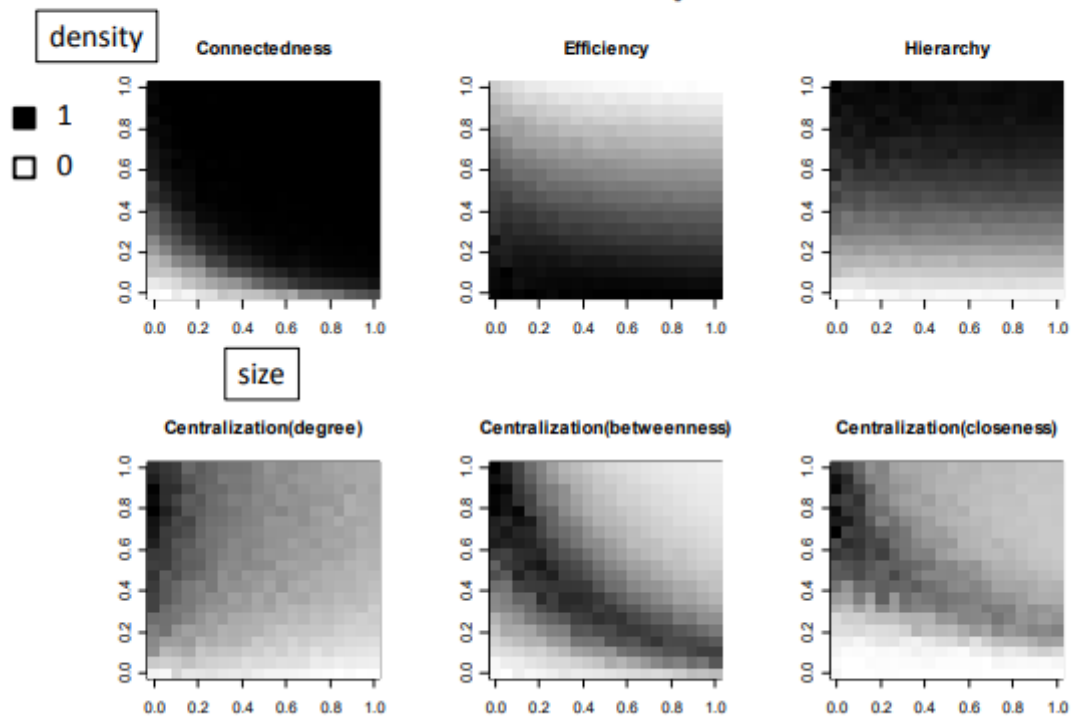
efficiency distribution by graph size and density



connectedness distribution by graph size and density

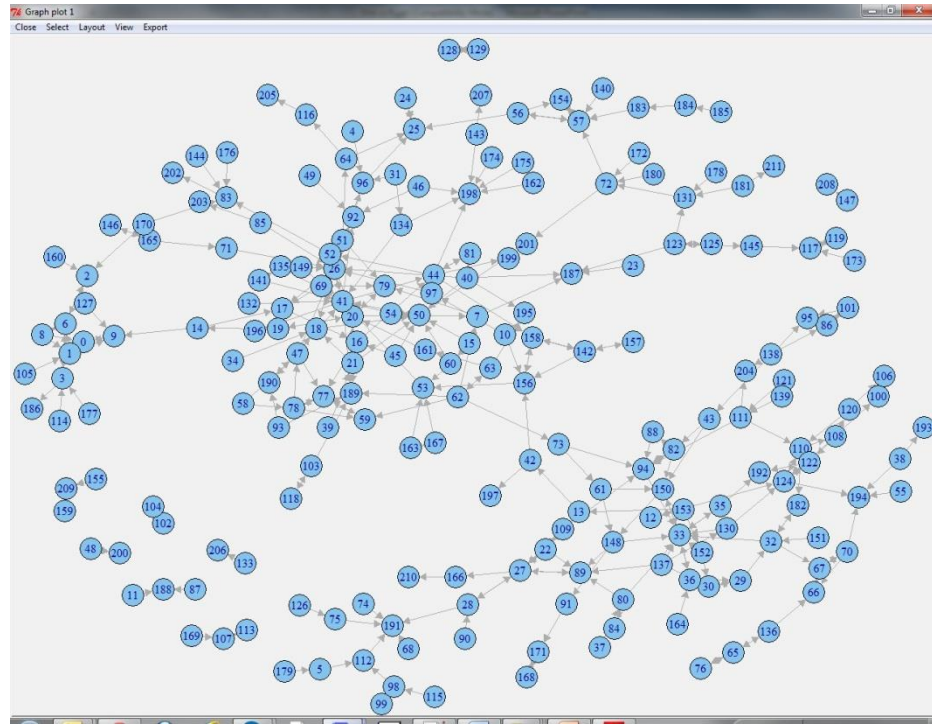


GLI map



1) Visualizing Social Networks (Tcl/Tk Network Graph)

```
tkplot(igraph,  
layout=layout.fruchterman.reingold)
```



2) Finding Key Actors

Centrality Measures

Medium or prominence is a well-known metric for determining a person's position within the overall structure of a social network. It can be computed using a variety of measures. Degree, betweenness, proximity, and eigenvector centrality are the most commonly utilised. The first three were proposed by Freeman (1978) and are best suited for low-bandwidth networks. Brin and Page (2012) recently came up with extensions for weight loss networks. Analog (1987) presented the fourth measure, eigenvector centrality, which is based entirely on a spectral graph approach. It gained a lot of traction once it was utilised as the foundation for the well-known Google PageRank algorithm, which we'll go over in the next part. Whereas other quantitative measures for individual degree are proposed in the literature, we will focus on defining median values in this section. These processes determine the individual's relative cost within the network, indicating how relationships are targeted on some persons and, as a result, providing information on the societal power.

Greater center levels were associated with much more prominent characters inside the group, as their central location grants them benefits such as easier and faster access to various personalities.

```

Metrics of actor centrality include:
• degree
  (number of connections)
• betweenness
  (number of shortest paths an actor is on)
• closeness
  (relative distance to all other actors)
• eigenvector centrality
  (leading eigenvector of sociomatrix)

Calculating Centrality Measures
metrics <- data.frame( deg=degree(igraph), # degree
bet=betweenness(igraph), # betweenness
clo=closeness(igraph), # closeness eig=evcent(igraph)$vector, # eig.cent. cor=graph.coreness(igraph) # coreness
)

```

3) Plot Eigenvector Centrality on Betweenness

This measure assesses how a protagonist is linked to certain other well-linked individuals and is predicated on the distribution of correlated values in each and every region. A first eigenvalues of the data structure is given this position. The basic premise of embedded is that a person's ability and status are continuously determined by the power and position of his switching. Figures who really are immediately linked to a certain personality, described to as the psyche, are called to as Changes over time. Alters is a term widely used in the examination of a narcissistic shared community. In other words, we can state that a single node's lengths are proportional to the combined of its nearby items.

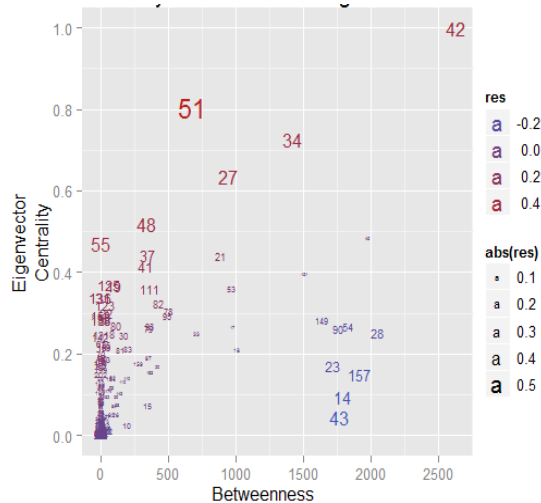
- Alternatively is to adjust / reverse eigenvector central ness during fossil testing.
- A character with a high concentration and low eigenvector centrality may be an important gate keeper for the middle character.
- A low-key character and a high middle eigenvector may have a unique access to mid-level players.

4) Visualizing with ggplot2

```

library(ggplot2) ggplot(
metrics,
aes(x=bet,y=eig,
label=rownames(metrics), colour=res,size=abs(res))
)+
xlab("Betweenness Centrality")+ ylab("Eigenvector Centrality")+ geom_text()+
opts(title="Key Actor Analysis for Hartford Drug Users")

```



5) Hierarchical clustering

It is a common approach for locating organisations because it does not require any estimations of membership organization, participation, or frequency. Agglomerative methods are based on a "excited formal organisation" (small groups in 's increasingly that are aggregated into larger groups), which is commonly depicted by a graph. The multilayer structure is revealed. Such characteristics are quite valuable.

In the available domains, there is less information regarding the network's plant communities. Additionally, these approaches have been shown to be particularly efficient in controlling fuzzy clustering challenges, giving particularly appealing for graph division and group recognition.

The classic cluster formation procedure, which is based on the criterion of strong affinity, is relatively simple. The decision of template matching, which is then used to determine exactly close those items were, is usually the initial step.

The items are defined by the a regional or global attribute. Semantic similarity, Weaving scale, Geometric or Midtown relationships, and humming distances are instances of these kind of behaviours. The best match among all sets of items is then calculated.

The purpose of cluster analysis in graphs is to group identical nodes together. The structural qualities of nodes in a network can be used to determine similarities / difference. For particular, the number of related neighbours among network cells can be regarded as a measure of resemblance, and nodes with the most common neighbours can be categorised into a single group (Stamm and Fast (1994)). Endpoints from same population are not aggregated into a single cluster when this criterion is used (Newman 2004). Different measurements and methodologies for capturing the plant communities of systems via clustering algorithm have been developed.

Results

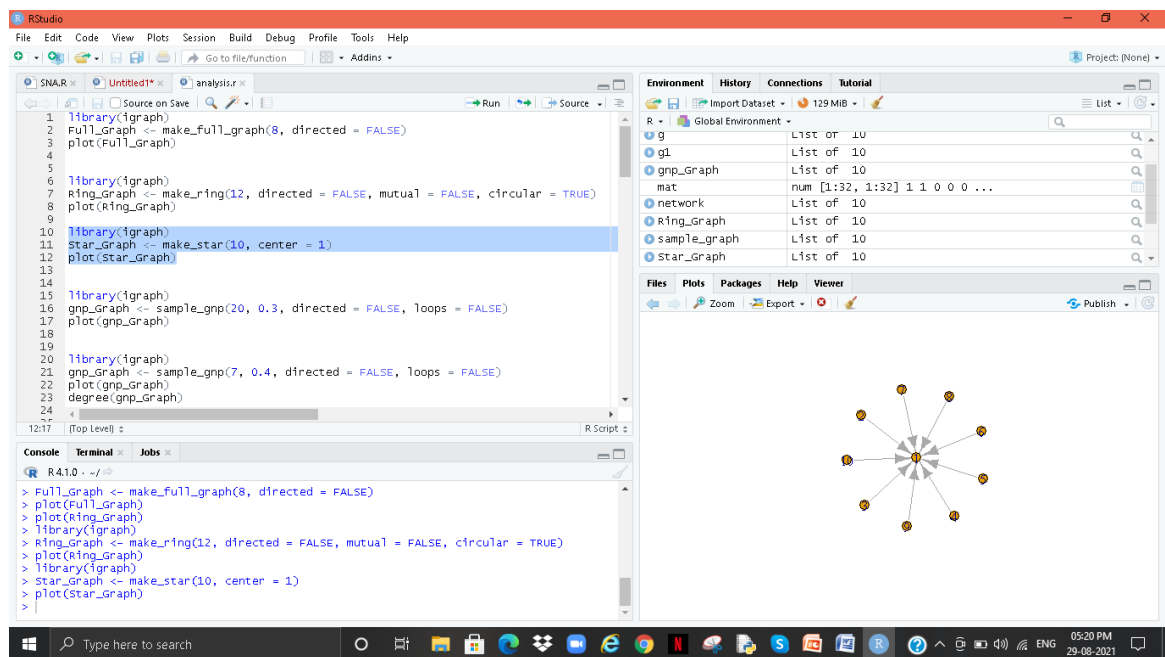
Graph Representations

Charts are forms of metadata architectures that have been addressed in the literature: category organization and divisional structure. These formats are appropriate for saving diagrams on a system so that technical solutions may examine these. Action listings & neighbouring listings are examples of collection architectures.

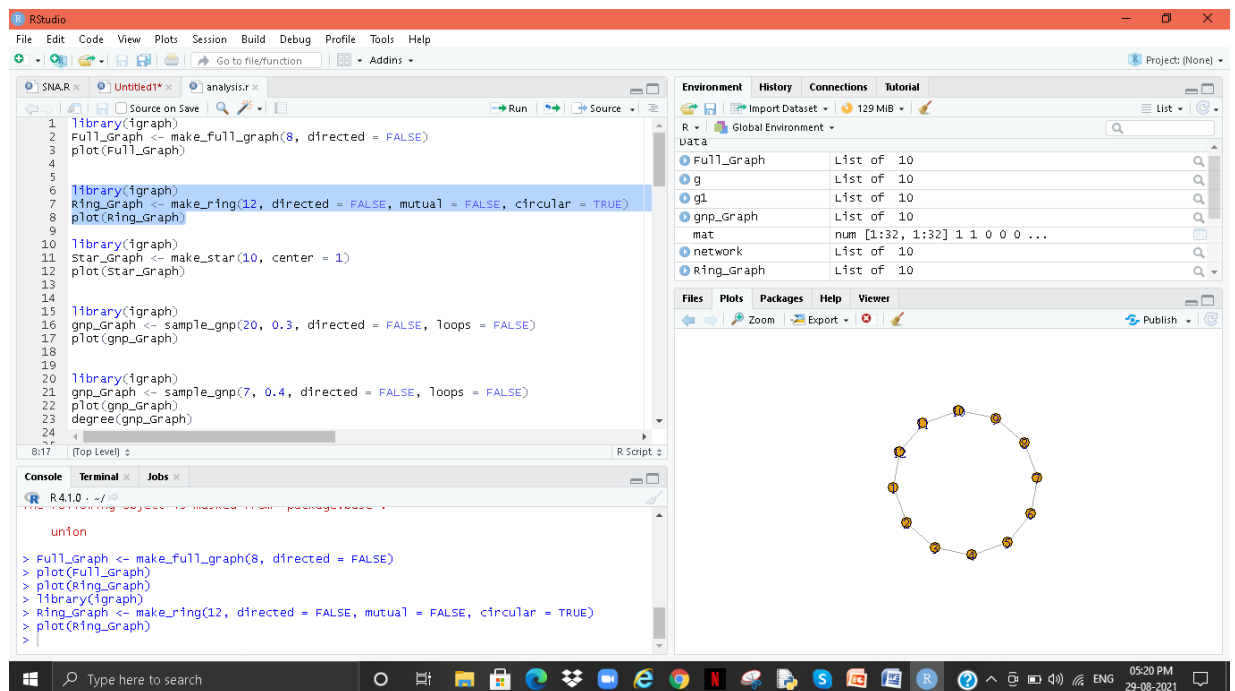
Tiny diagrams have less hard drive capacity, making them excellent for archiving.

On the other hand, matrix structures such as the Incidence Matrix, the Adjacency Matrix or the Social Matrix, the Laplacean matrices (which contain both adjacent and degree information) and the distant matrices (adjacent matrices with the same length as the matrix entries are shorter than the pairs of headers). Ways are suitable for referring to complete matrices. A variety of graphs can be used to model different types of social networks.

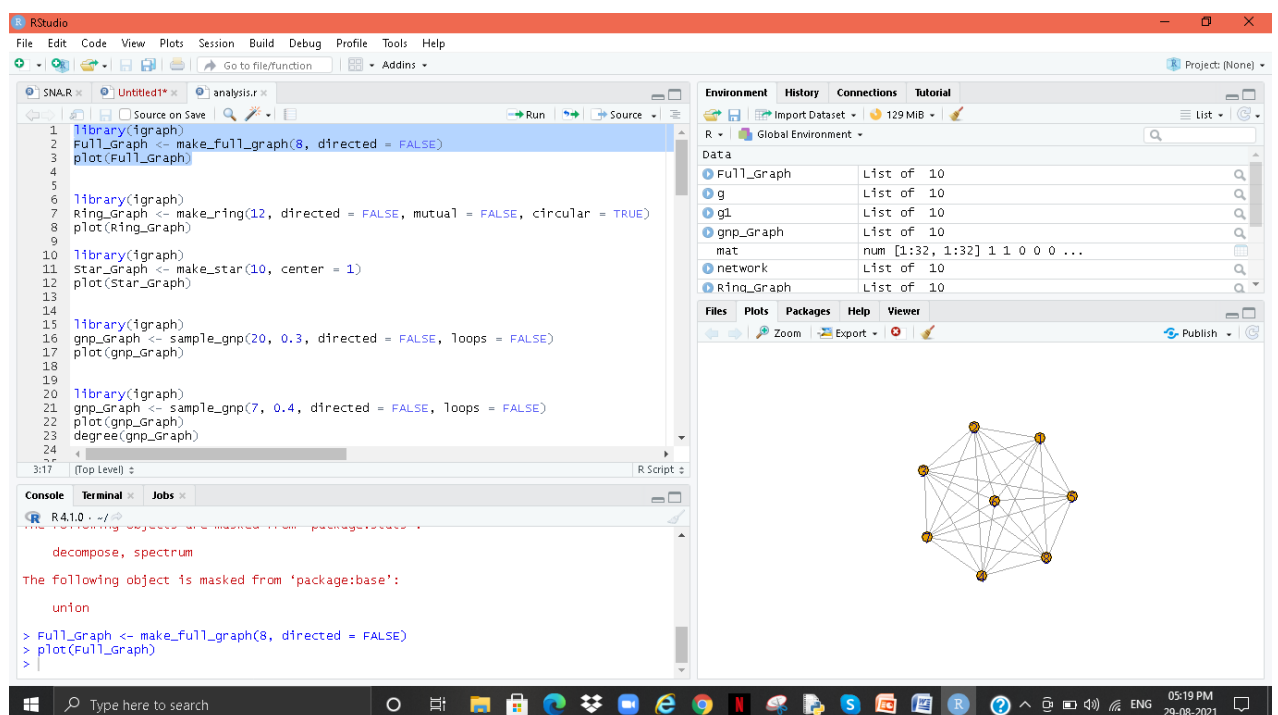
1. Star Graph



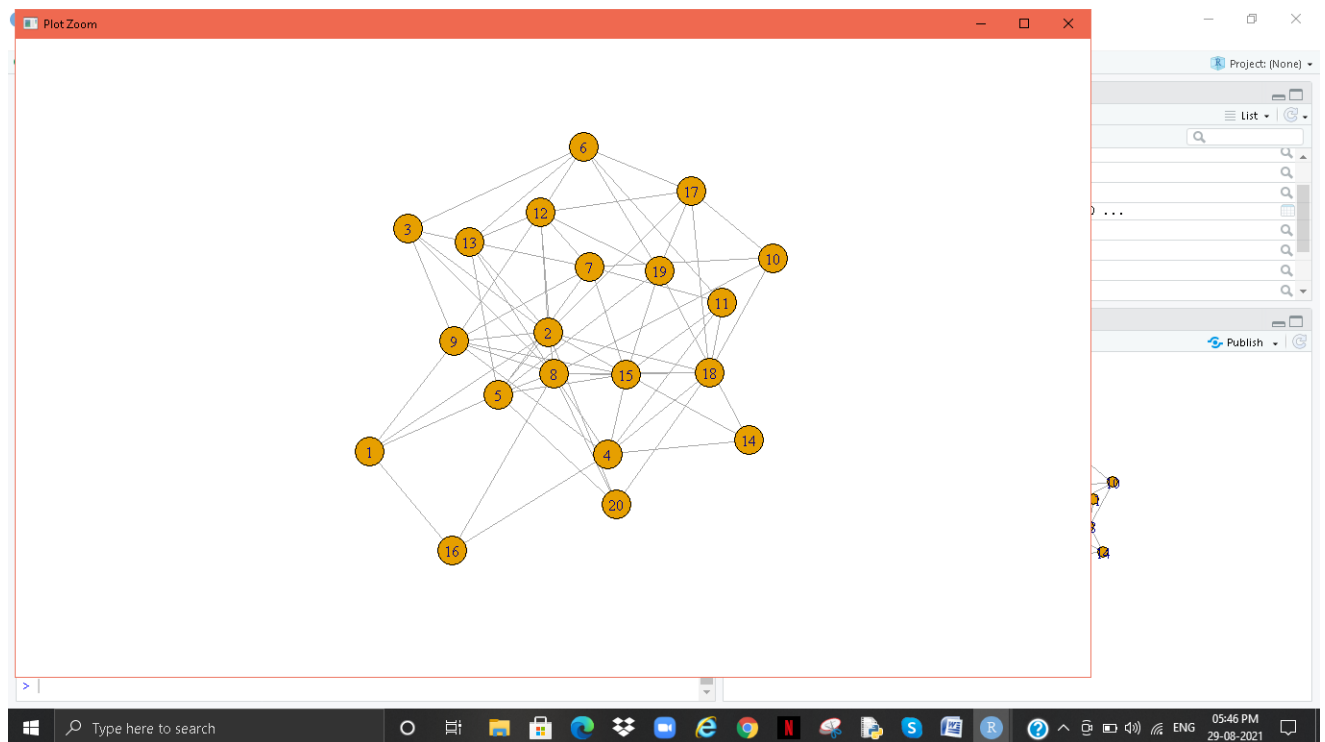
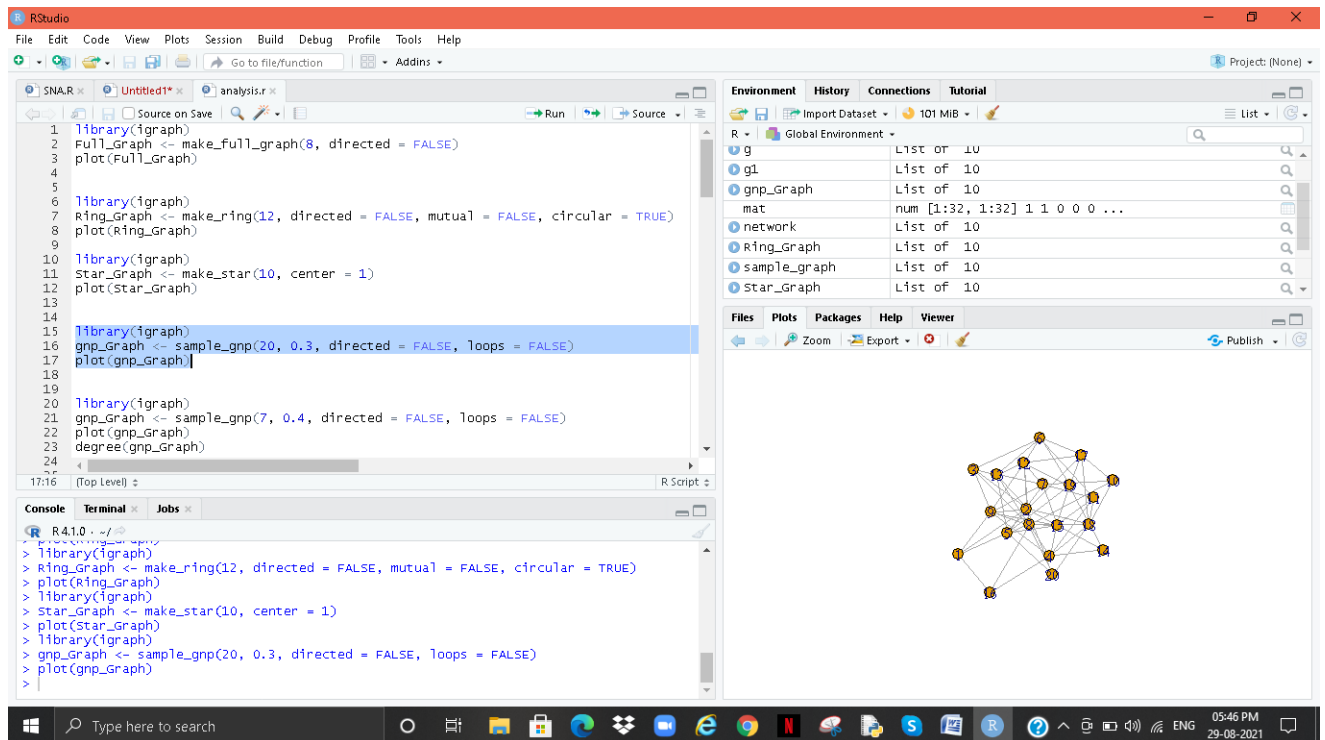
2. Ring Graph



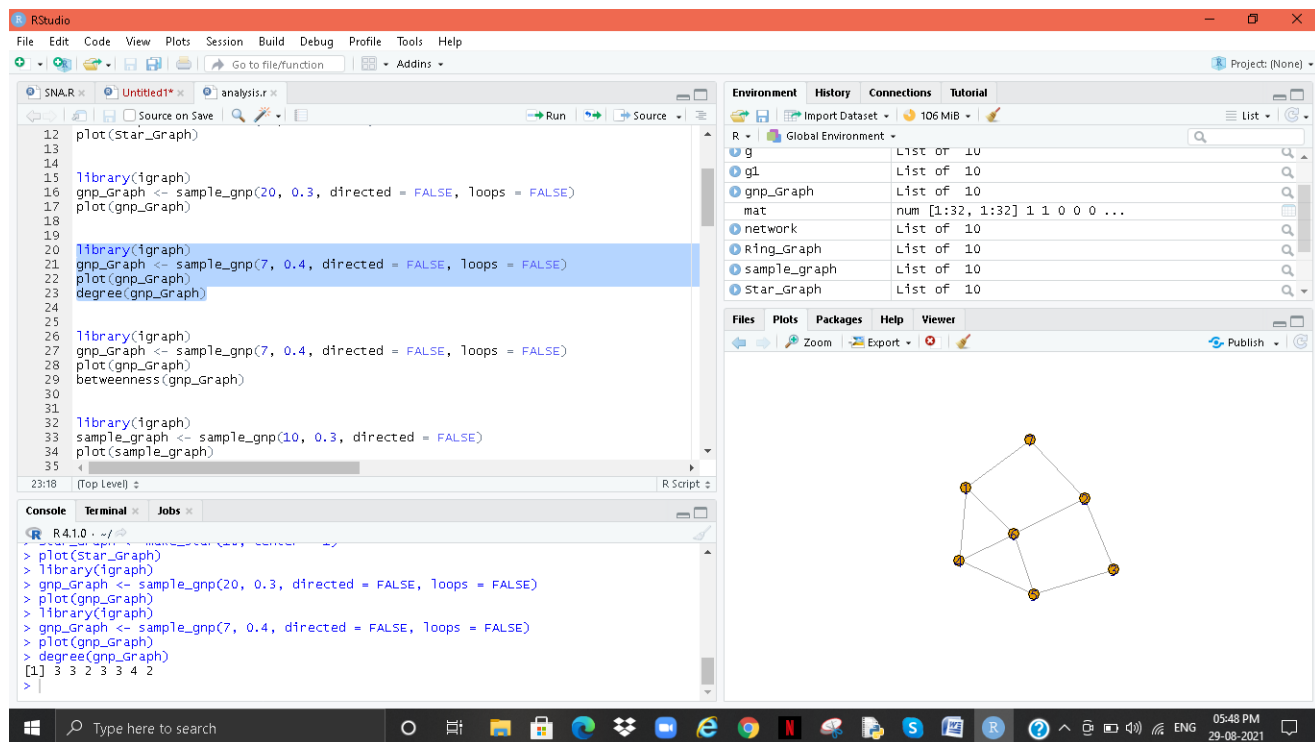
3. Full Graph



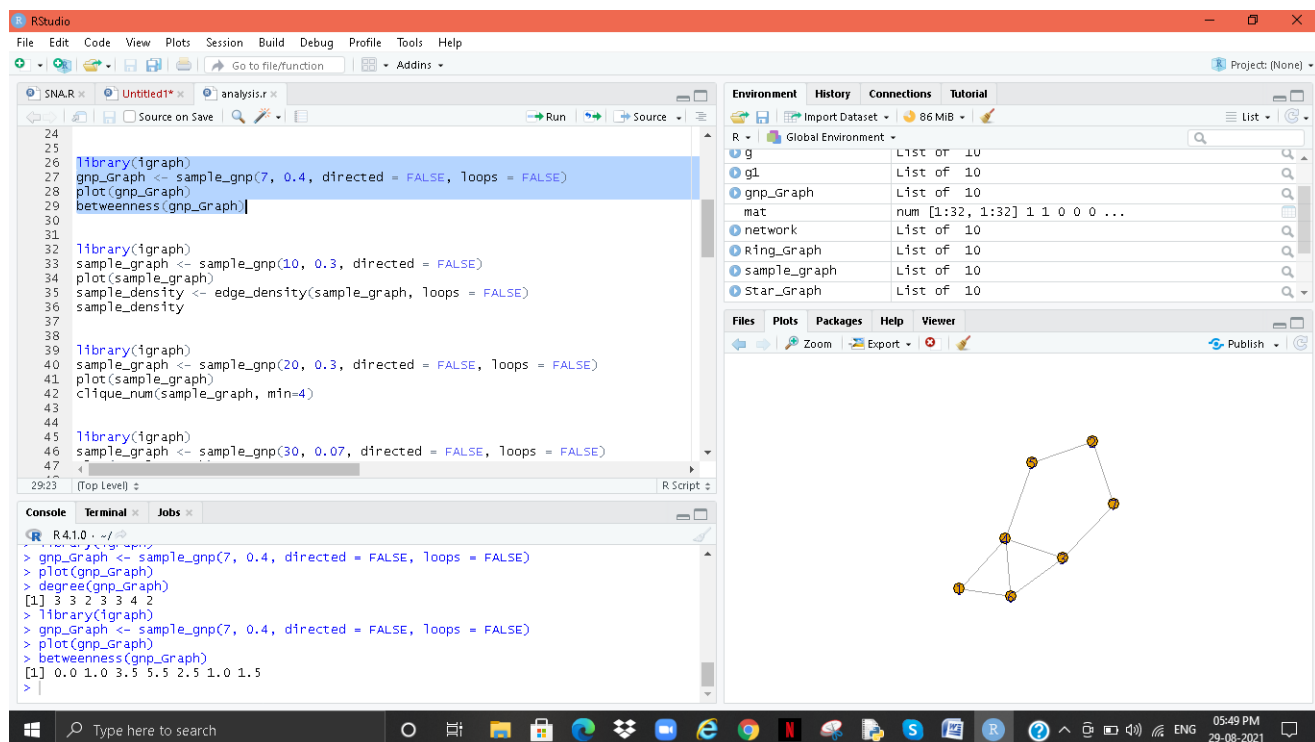
4. Gnp Graph



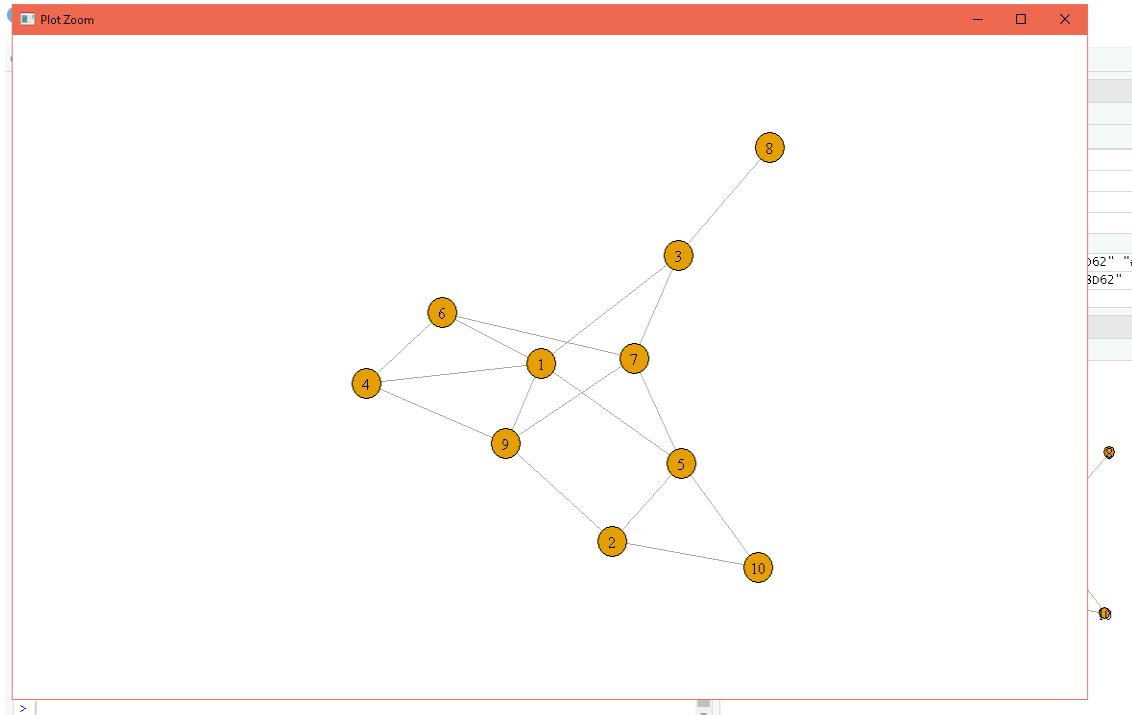
5. Gnp Degree Graph



6. Gnp Betweenness Graph



7. Density Graph



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Environment History Connections Tutorial

Global Environment

Object	Class	Attributes
network	igraph	list of 10
Ring_graph	igraph	list of 10
sample_graph	igraph	list of 10
Star_graph	igraph	list of 10

Values

Variable	Value
cou1	chr [1:3] "#66C2A5" "#FC8D62" "#8DA0CB"
my_color	chr [1:32] "#FC8D62" "#FC8D62" "#66C2A5" "#FC8D62"...
sample_density	0.3555555555555556

```

24
25
26 library(igraph)
27 gnp_graph <- sample_gnp(7, 0.4, directed = FALSE, loops = FALSE)
28 plot(gnp_graph)
29 betweenness(gnp_graph)
30
31
32 library(igraph)
33 sample_graph <- sample_gnp(10, 0.3, directed = FALSE)
34 plot(sample_graph)
35 sample_density <- edge_density(sample_graph, loops = FALSE)
36 sample_density
37
38
39 library(igraph)
40 sample_graph <- sample_gnp(20, 0.3, directed = FALSE, loops = FALSE)
41 plot(sample_graph)
42 clique_num(sample_graph, min=4)
43
44
45 library(igraph)
46 sample_graph <- sample_gnp(30, 0.07, directed = FALSE, loops = FALSE)
47
3615 (Top Level) R Script

```

Console

```

R 4.1.0 ~ /
> library(igraph)
> gnp_graph <- sample_gnp(7, 0.4, directed = FALSE, loops = FALSE)
> plot(gnp_graph)
> betweenness(gnp_graph)
[1] 0.0 1.0 3.5 5.5 2.5 1.0 1.5
> library(igraph)
> sample_graph <- sample_gnp(10, 0.3, directed = FALSE)
> plot(sample_graph)
> sample_density <- edge_density(sample_graph, loops = FALSE)
> sample_density
[1] 0.3555556
>

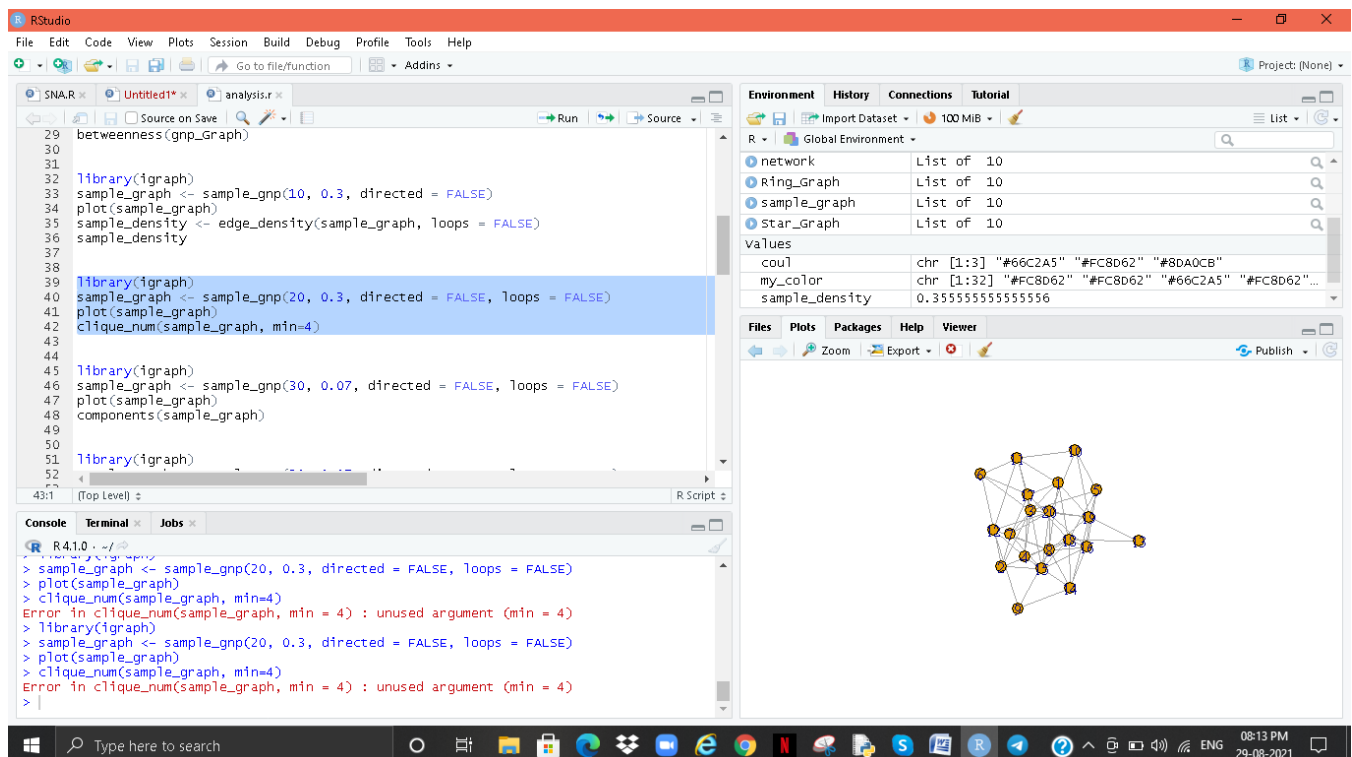
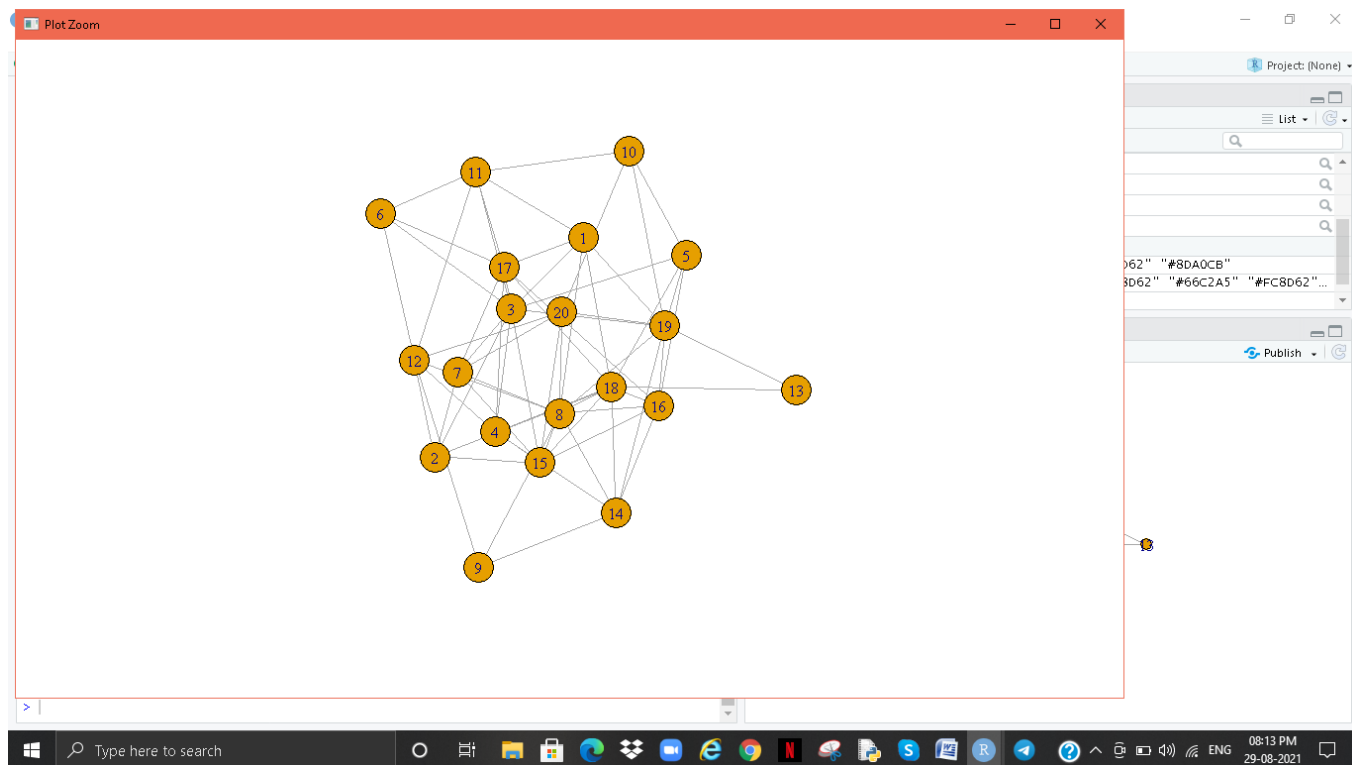
```

Files Plots Packages Help Viewer

Zoom Export Publish

Windows Taskbar: Type here to search, 05:52 PM, 29-08-2021

8. Clique Graph



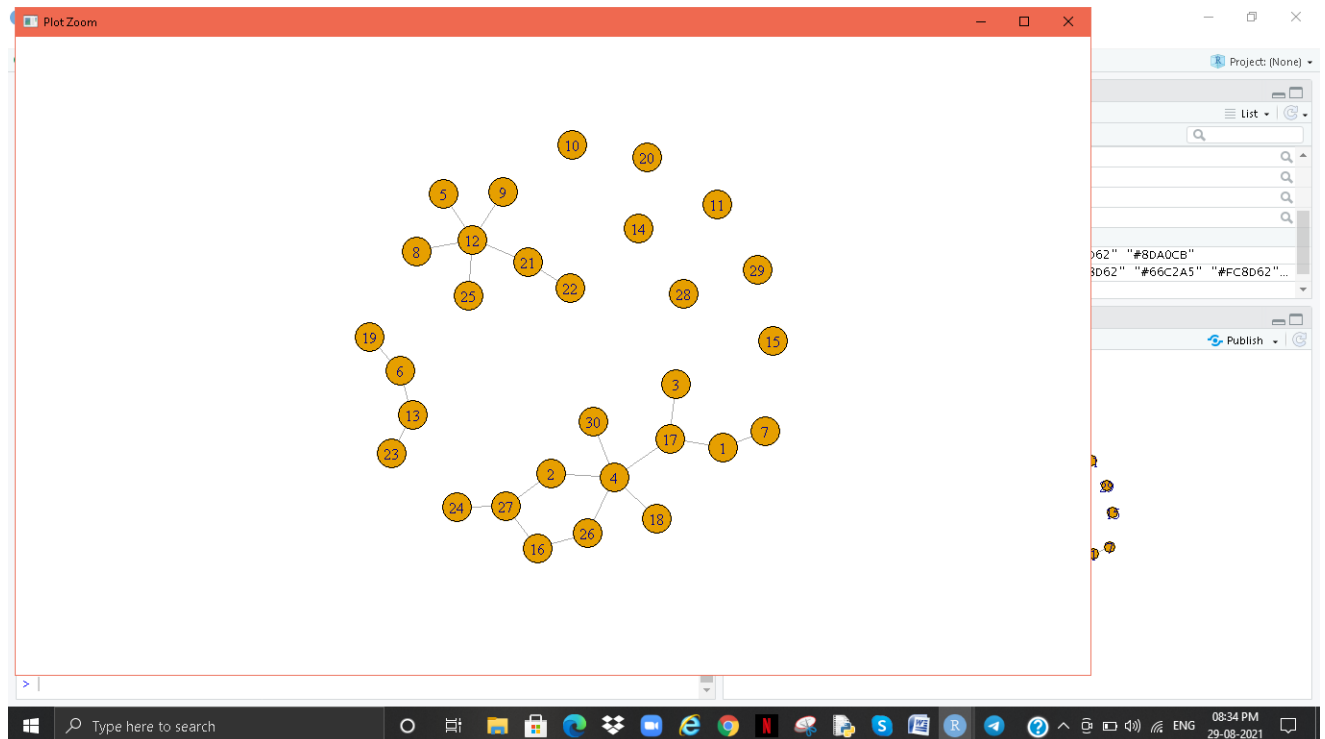
9. Component Graph

Libraries (iGraph)

```
Sample_graph <- sample_gnp (30, 0.07, direction = wrong, loop = wrong)
```

```
Plot (sample_graph)
```

```
Component (sample_graph)
```



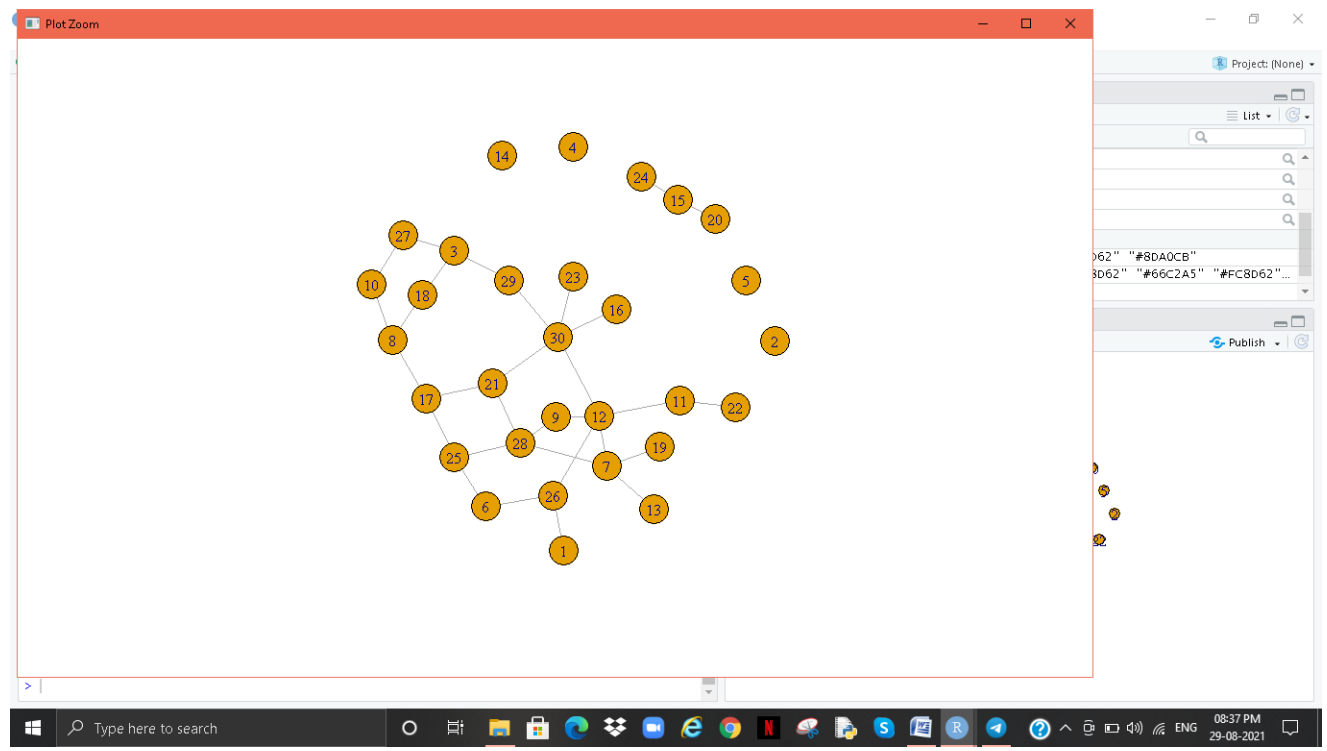
10. Graph Network

Libraries (iGraph)

```
Sample_graph <- sample_gnp (30, 0.07, direction = wrong, loop = wrong)
```

```
Plot (sample_graph)
```

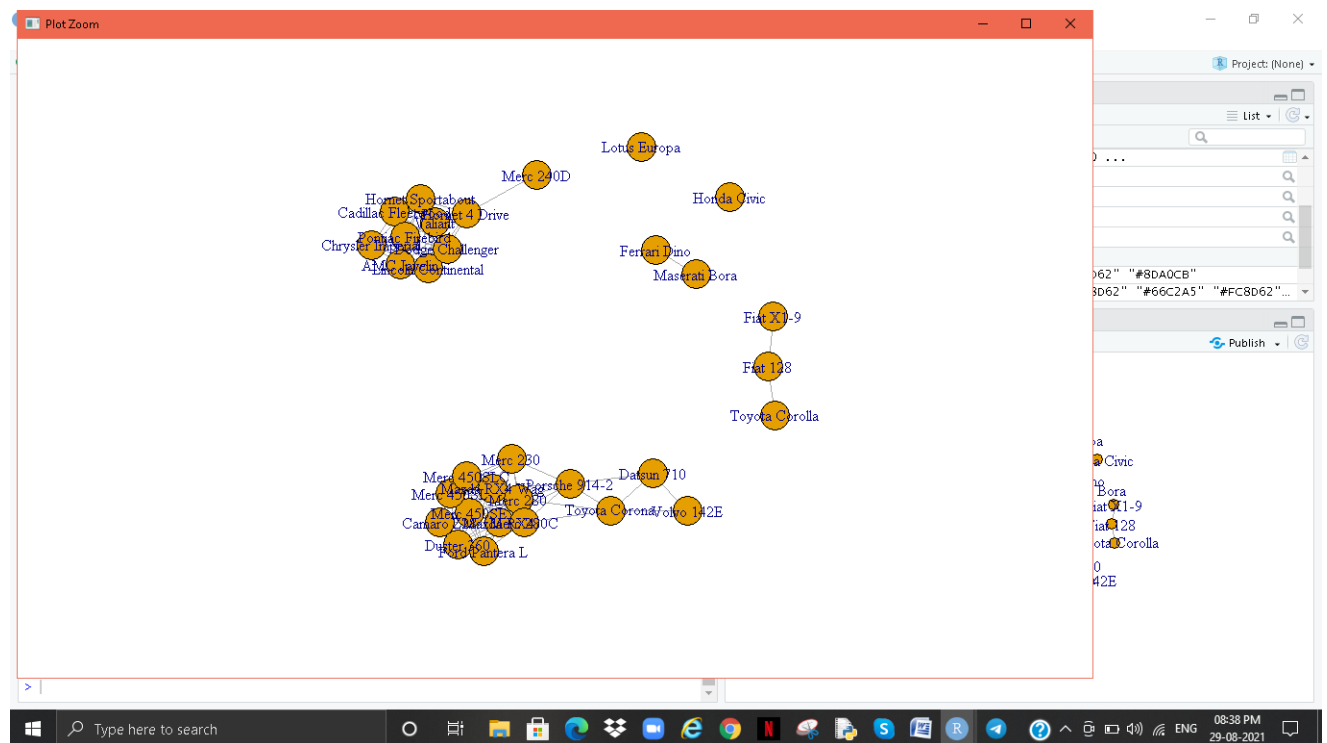
```
random_walk (sample_graph, 8, 10, difficulty = "return")
```



Libraries (iGraph)

```
Mat <- core (t (mCars [, c (1,3: 6)]))
```

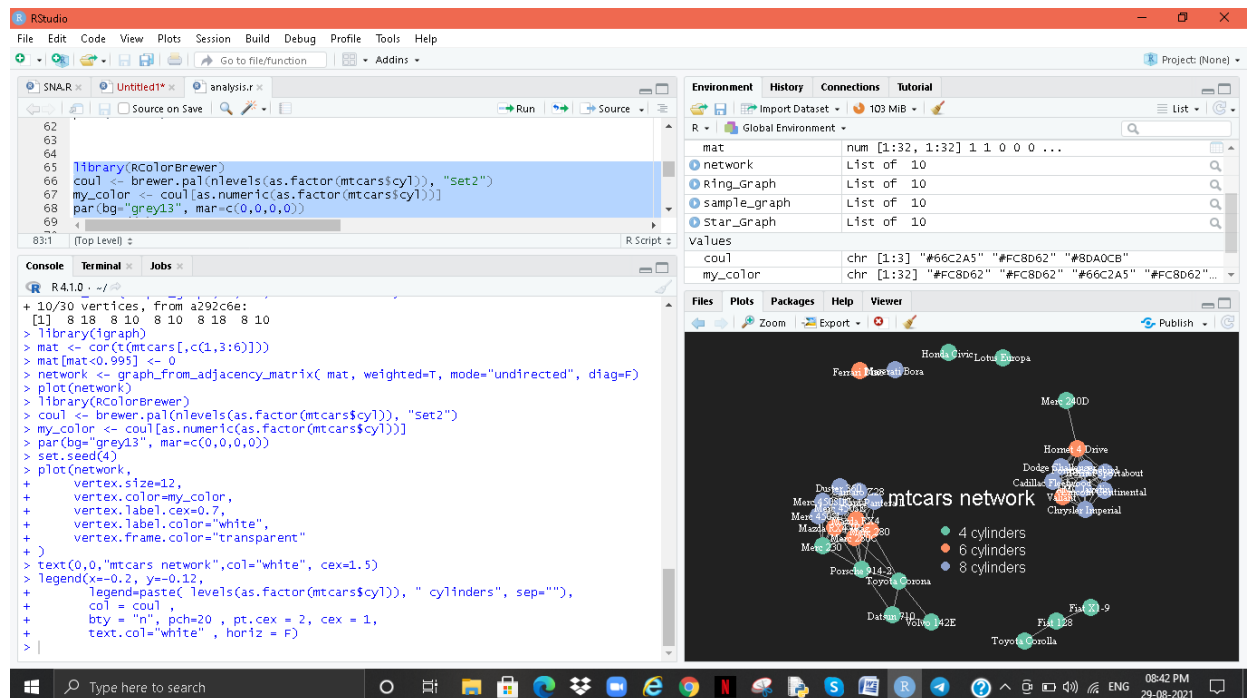
```
Mat [mat <0.995] <- 0
```



11. Final - car to Bluetooth device connections

```
*SNA - Notepad
File Edit Format View Help

Libraries (Arcolor Brewer)
coul <- brewer.pal (nlevels (as.factor (mtcars $ cyl)), "Set2")
my_color <- coul [as.numeric (as.factor (mtcars $ cyl))]
Equals (bg = "gray 13", mar = c (0,0,0,0))
set.seed (4)
Plot (network,
      Head size = 12,
      vertex.color = my_color,
      vertex.label.cex = 0.7,
      vertex.label.color = "white",
      vertex.frame.color = "Transparent"
    ,
Text (0,0, "mtcars network", col = "white", cex = 1.5)
Legend (x = -0.2, y = -0.12,
        legend = paste (level (as.factor (mtcars $ cyl)), "cylinder", sept = ""),
        Colonel = Kaul,
        bty = "n", pch = 20, pt.cex = 2, cex = 1,
        text.col = "white", horiz = F)
|
```



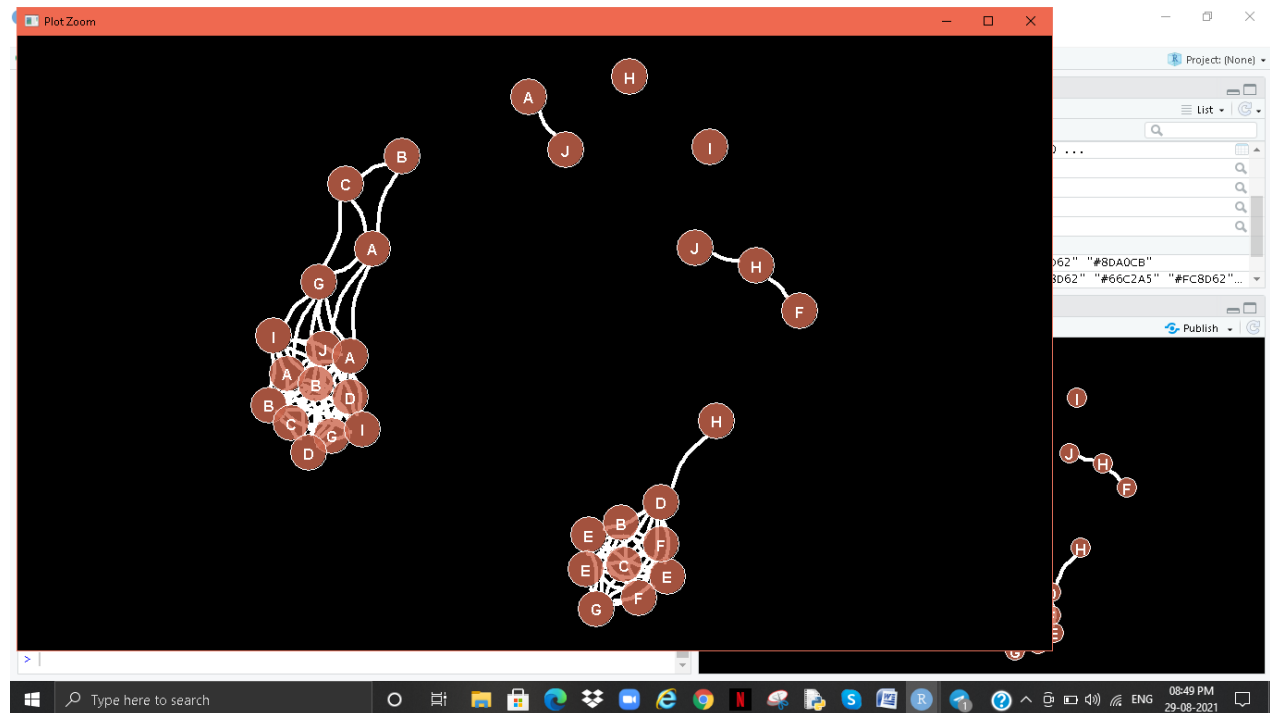

```
Plot (network,
      Edge.Color = representative (c ("red", "pink"), 5),
      Edge.width = seq (1,10),
      Edge.arrow.size = 1,
      Edge.arrow.width = 1,
      Edge.lty = c ("solid")
      (FALSE sets it to 0 and TRUE to 0.5)
      ,
      Equals (bg = "black")
)

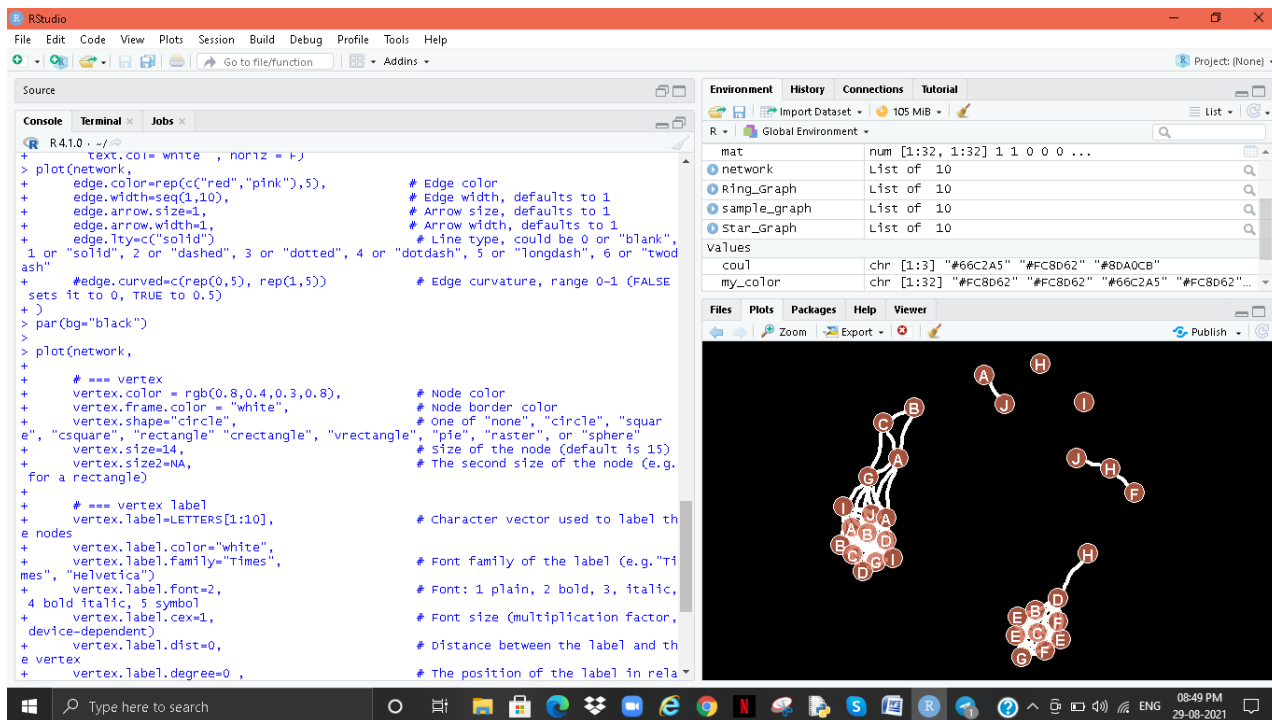
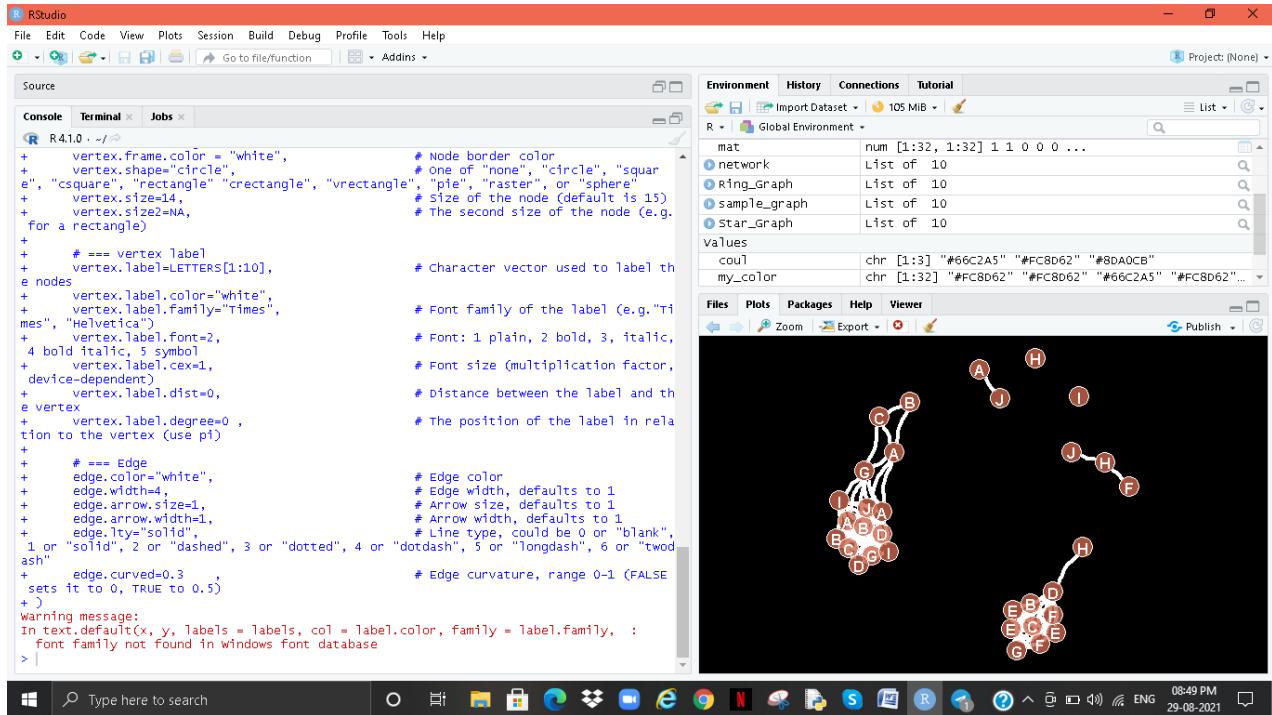
Plot (network,
      # === Top
      vertex.color = rgb (0.8,0.4,0.3,0.8),
      vertex.frame.color = "white",
      vertex.shape = "circle",
      Head.size = 14,
      vertex.size2 = NA,

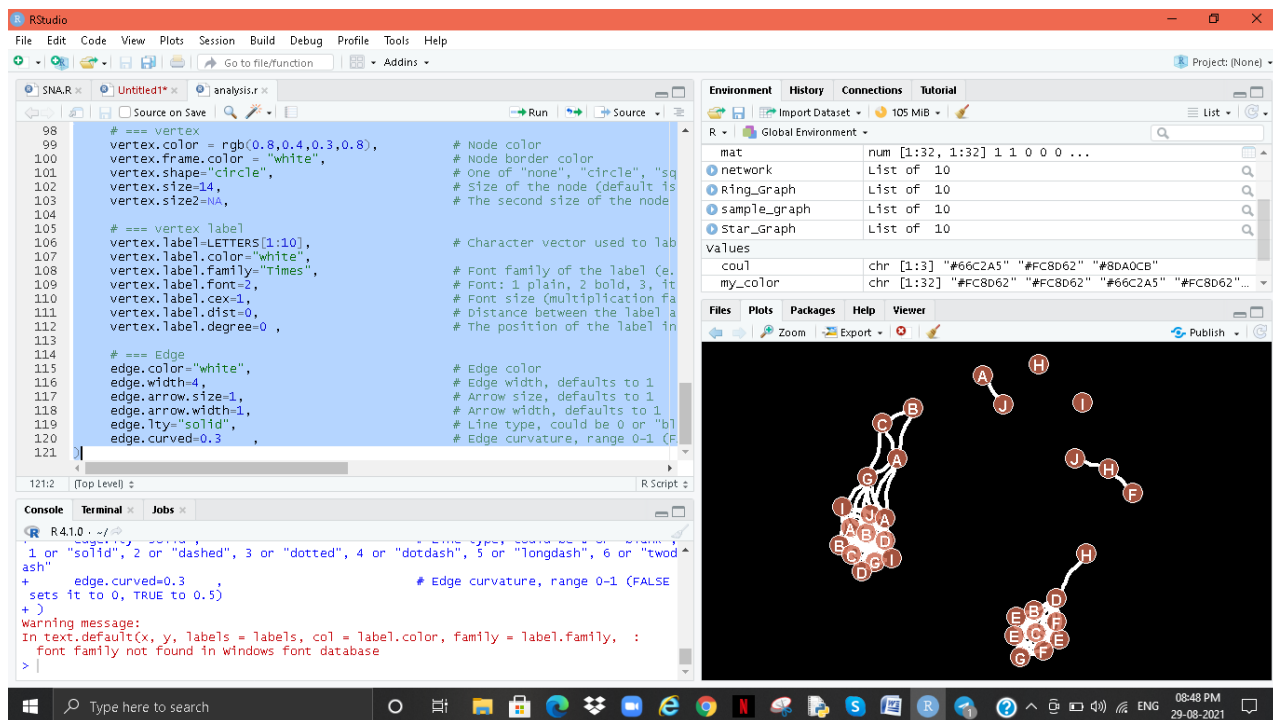
      # === Top label
      vertex.label = characters [1:10],
      vertex.label.color = "white",
      vertex.label.family = "Times",
      vertex.label.font = 2,
      vertex.label.cex = 1,
      vertex.label.dist = 0,
      vertex.label.degree = 0,

      # === Torrent
      Edge.Color = "white",
      Edge.Width = 4,
      Edge.arrow.size = 1,
      Edge.arrow.width = 1,
      Edge.lty = "solid",
      Edge.amp = 0.3,

```







Conclusion

SNA is seldom chastised for concentrating upon methodological issues rather than complicated & profound experimental identifying situations on clear and strong complex algorithms such as computational geometry and relatively non mathematics, as I just stated. This strategy. There are still numerous concerns to be handled in the paradigm, even though there are many empirical issues relating to the potential of researching social elements of an internet.

I've given a quick summary of link prediction methodology, aims, and applications in this presentation. Numerous SNA activities and related duties are carried out while bearing in mind various network types. Hierarchical clustering has become increasingly common today, thanks to the high quality of network troubleshooting and knowledge obtained from data supplied by a multitude of scenarios. As the volume of commitment to providing grows, so does the sophistication.

One of the primary problems right now is advancement with in analysis, management, and exploitation of significantly large connections. It is getting increasingly intensive and complicated as Internet 2.0, Rfid, and other technologies proliferate.

Informatics, Earth Science, Cognitive Science, Mobility Patterns, Recommendations, etc. Fraud detection, along with social media, gene expression, protein interactions, marketing, brainstorming prediction, etc. The recent growing demand for these applications on complex realities also requires progress in the world
Various network topologies such as multi-layered, heterogeneous and development networks. Therefore, this article paves the way for a basic understanding of the more complex issues associated with network analysis.

References

- Abraham, A., Hassanien, A.-E., Sná, V. et al. (2009) Computational social network analysis: Trends, tools and research advances. London: Springer Science & Business Media.
- Adamic, L. A. and Adar, E. (2003) Friends and neighbors on the web. Social networks, 25, 211–230.
- Aggarwal, C. C. (2009) Models for incomplete and probabilistic information. In Managing and Mining Uncertain Data, 1–34.
- https://www.itm-conferences.org/articles/itmconf/pdf/2016/01/itmconf_ics2016_03011.pdf
- Geek for Geeks.
- J.R. Turner and R. Muller, Int J Project Manage, 21, 1 (2003).
- T. Cooke-Davies, Int J Proj Manage, 20, 185 (2002).
- M.B. Pinto and J.K. Pinto, JPIM, 7, 200 (1990).
- P.S. Chinowsky, J. Diekmann and J. O'Brien, JCEM, 136, 452 (2009).
- P. Dietrich, IJPM, 37, 49 (2006).

- H. Kerzner, Advanced project management: Best practices on implementation (2004).
- H.J. Thamhaim and D. L. Wilemon, SLOAN MANAGE REV., 16,31 (1975).
- W.A. Reed, Doctoral dissertation (2006).
- D. Hinds and R. M. Lee, Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, IEEE, pp. 323-323 (2008).
- P. He, B. Li and Y. Huang, Second International Conference on CGC, pp. 418-423 (2012).
- R. Alsamadani, M. Hallowell and A. N. JavemickWill, CEM, 31, 568 (2013).
- T. Mueller-Prothmann and I. Finke, J. UCS, 10, 691 (2004).
- S.D. Pryke, Construct Manag Econ, 23,927 (2005).
- A. El-Sheikh and S.D. Pryke, Construct Manag Econ, 28, 1205 (2010).
- P. Chinowsky, J. Diekmann and V. Gaiotti, JCEM, 134, 804 (2008).
- P. Chinowsky and J.E. Taylor, EPOJ, 2, 15 (2012).
- H.R. Kerzner, Project management: a systems approach to planning, scheduling, and controlling (2013).