

**UNIVERSIDAD TÉCNICA DE MACHALA**

Maestría en Software

**Asignatura:**  
Base de datos NoSQL  
  
  
**Tema:**

**Workshop Graphs using R Studio**

**Docente:**

Ing. Nelson Piedra

**Estudiante:**

Ing. Jimmy Fernando Castillo Crespín

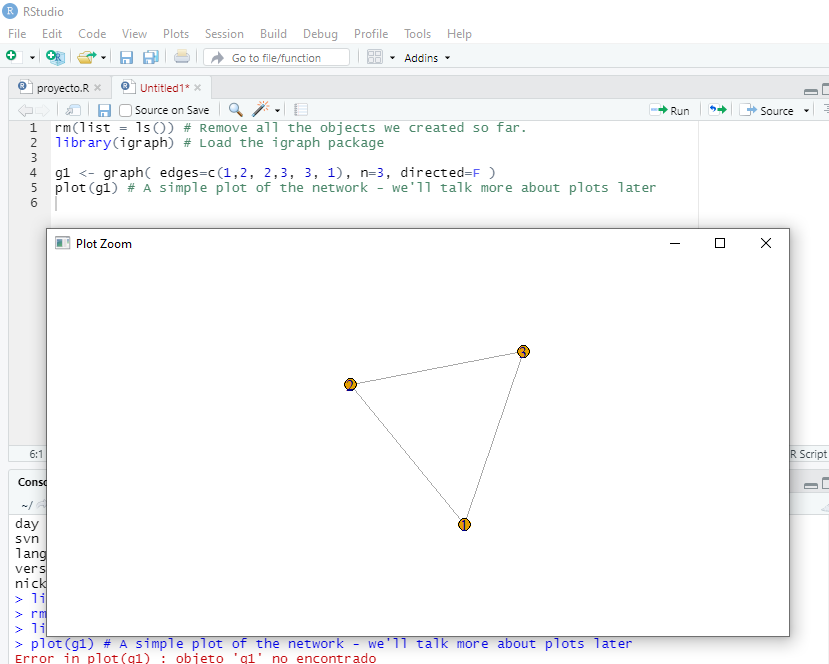
2021-2022

**Capítulo 2 - Networks in igraph**

**Create networks**

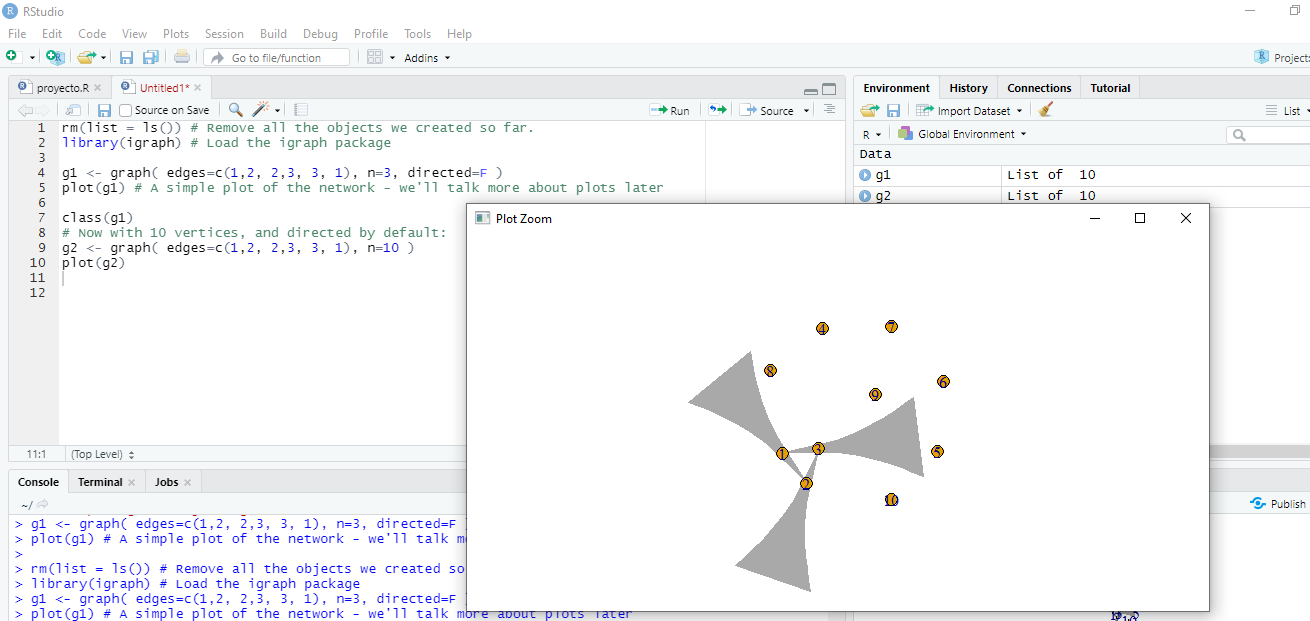
g1 <- graph( edges=c(1,2, 2,3, 3, 1), n=3, directed=F )

plot(g1) # A simple plot of the network - we'll talk more about plots later



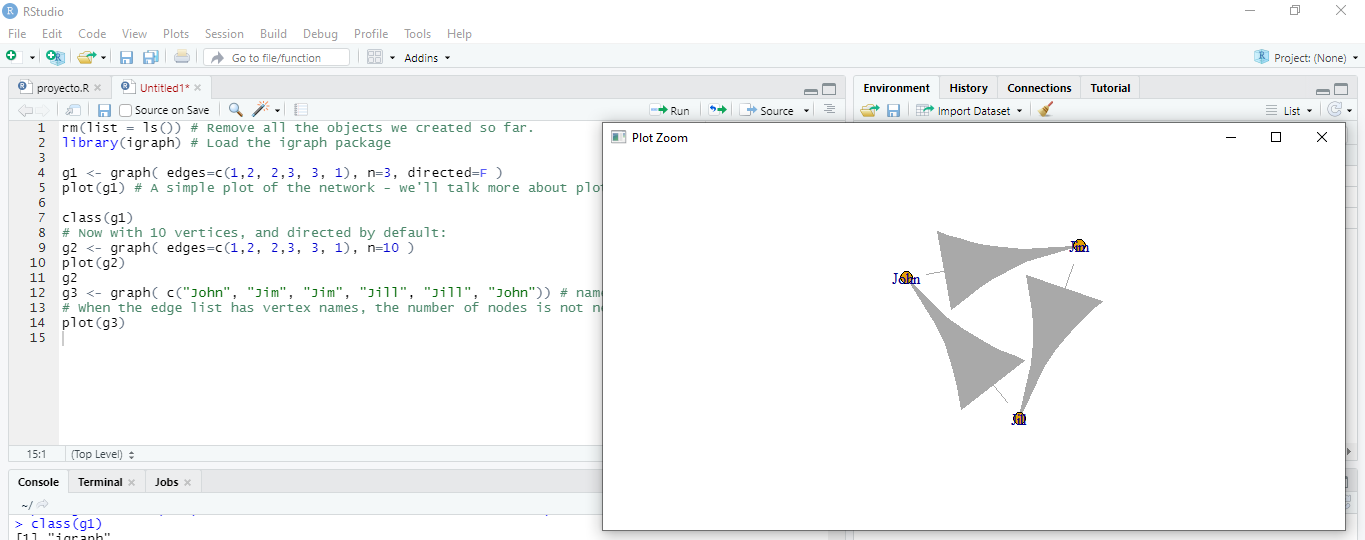
g2 <- graph( edges=c(1,2, 2,3, 3, 1), n=10 )

plot(g2)



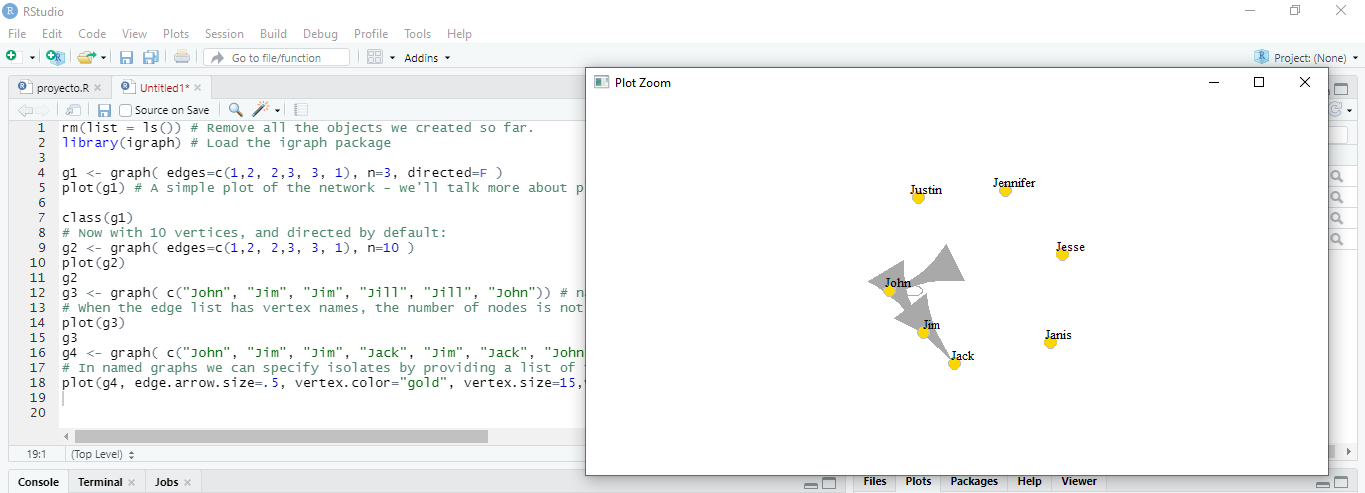
g3 <- graph( c("John", "Jim", "Jim", "Jill", "Jill", "John"))

plot(g3)

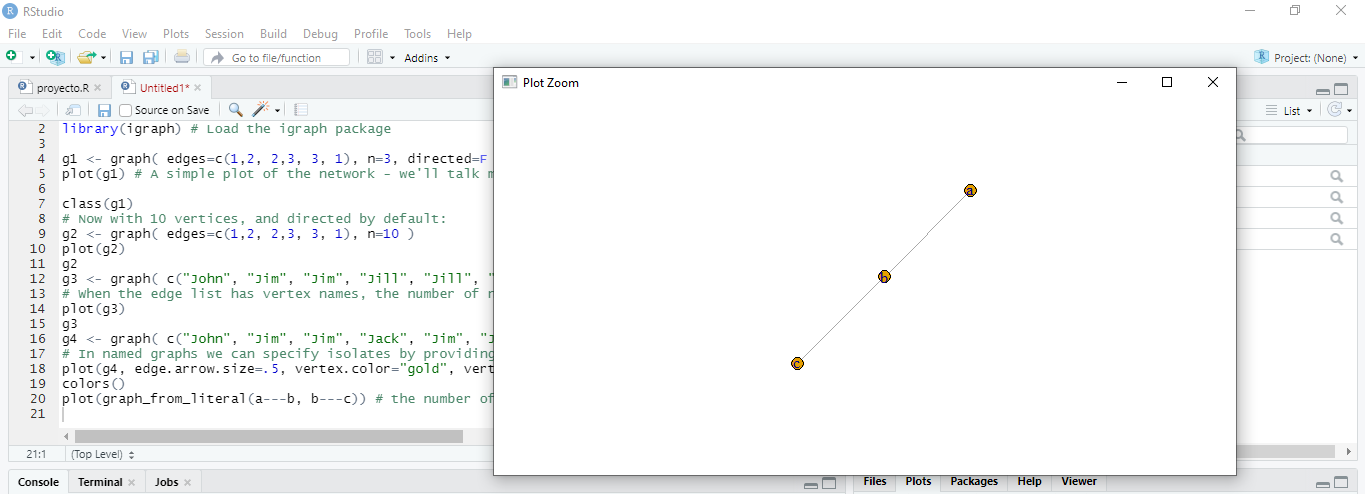


g4 <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"), isolates=c("Jesse", "Janis", "Jennifer", "Justin") )

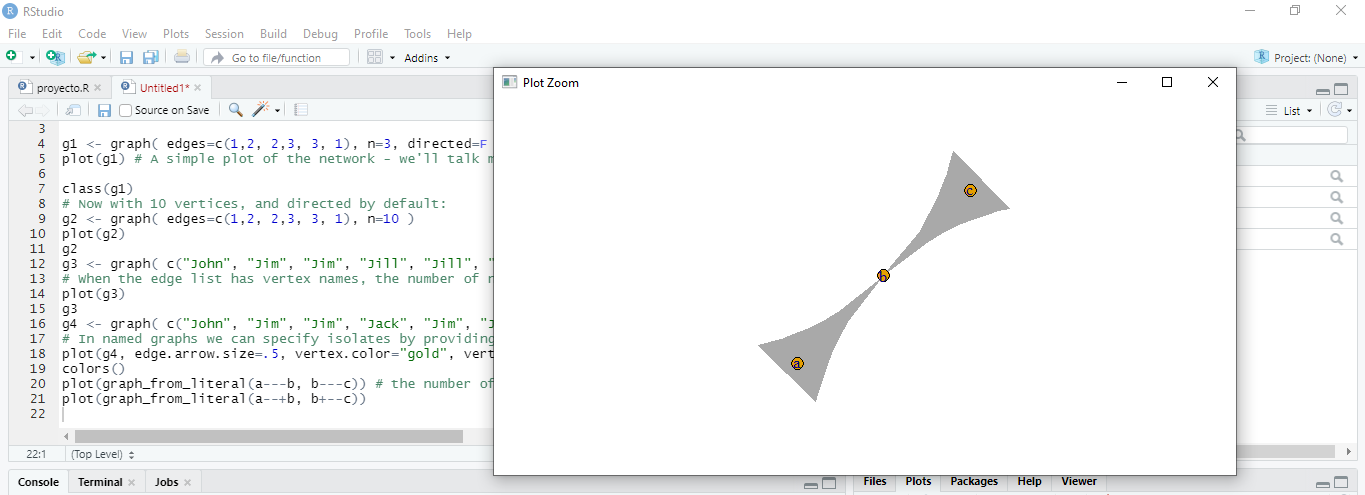
plot(g4, edge.arrow.size=.5, vertex.color="gold", vertex.size=15,vertex.frame.color="gray", vertex.label.color="black",vertex.label.cex=0.8, vertex.label.dist=2, edge.curved=0.2)



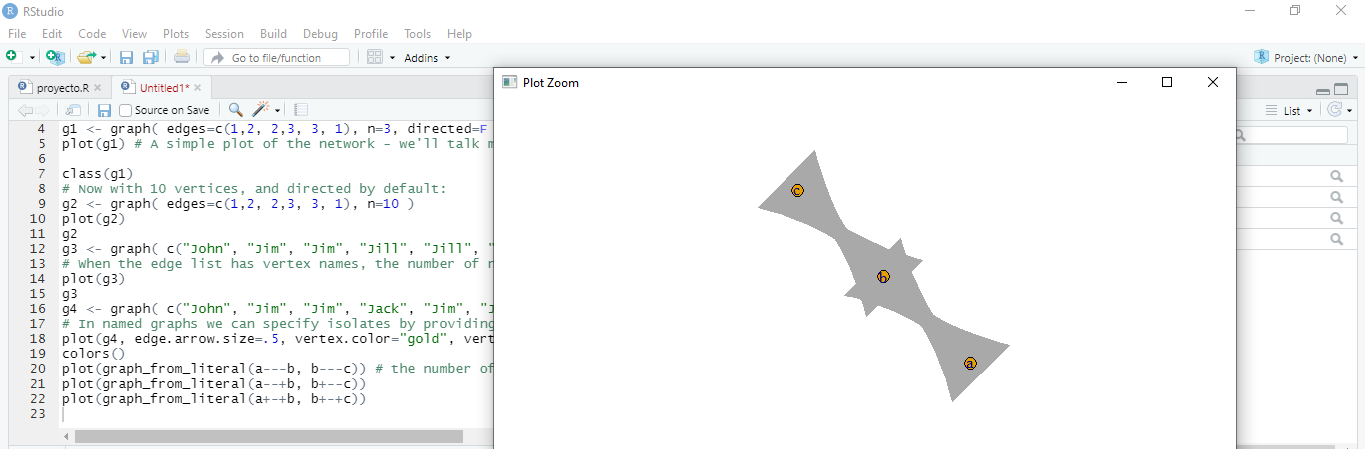
plot(graph\_from\_literal(a---b, b---c))



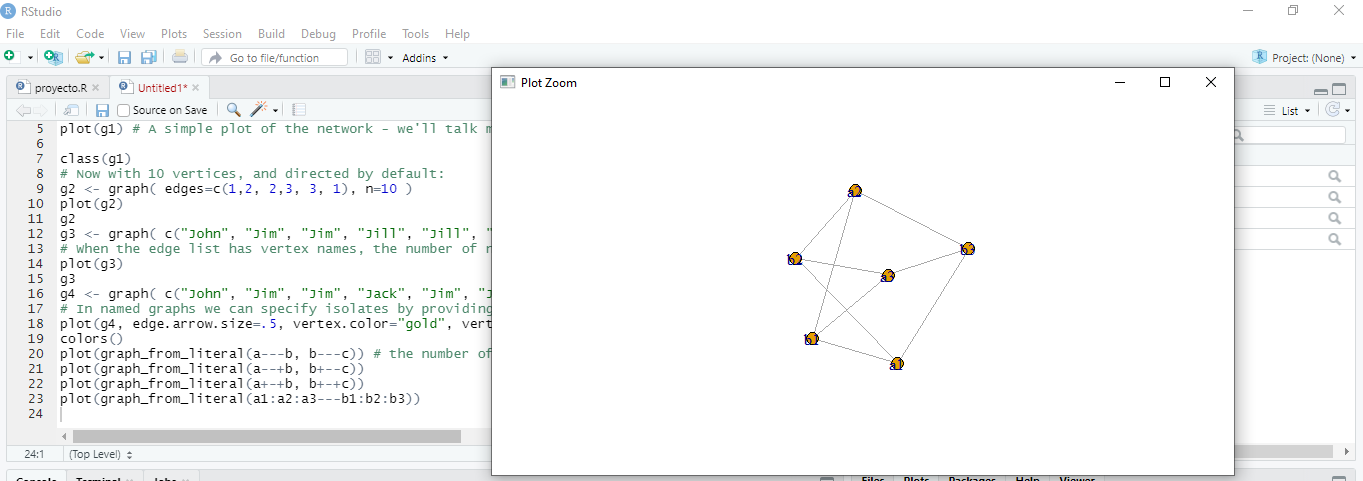
plot(graph\_from\_literal(a--+b, b+--c))



plot(graph\_from\_literal(a+-+b, b+-+c))

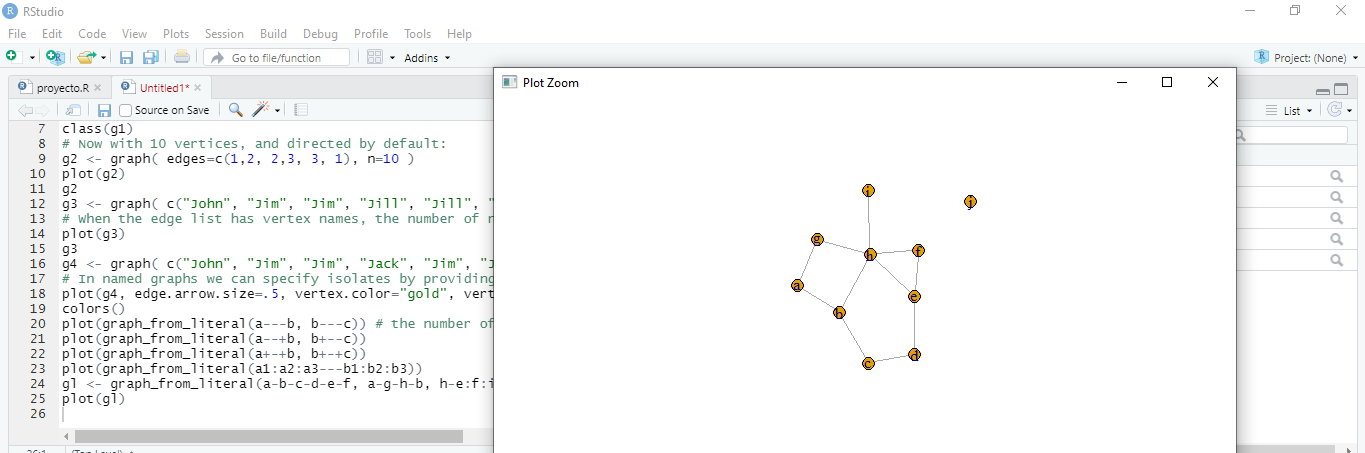


plot(graph\_from\_literal(a1:a2:a3---b1:b2:b3))



gl <- graph\_from\_literal(a-b-c-d-e-f, a-g-h-b, h-e:f:i, j)

plot(gl)



**Edge, vertex, and network attributes**

E(g4) # The edges of the object

V(g4) # The vertices of the object

g4[]

g4[1,]

V(g4)$name

V(g4)$gender <- c("male", "male", "male", "male", "female", "female", "male")

E(g4)$type <- "email" # Edge attribute, assign "email" to all edges

E(g4)$weight <- 10

edge\_attr(g4)

vertex\_attr(g4)

graph\_attr(g4)

g4 <- set\_graph\_attr(g4, "name", "Email Network")

g4 <- set\_graph\_attr(g4, "something", "A thing")

graph\_attr\_names(g4)

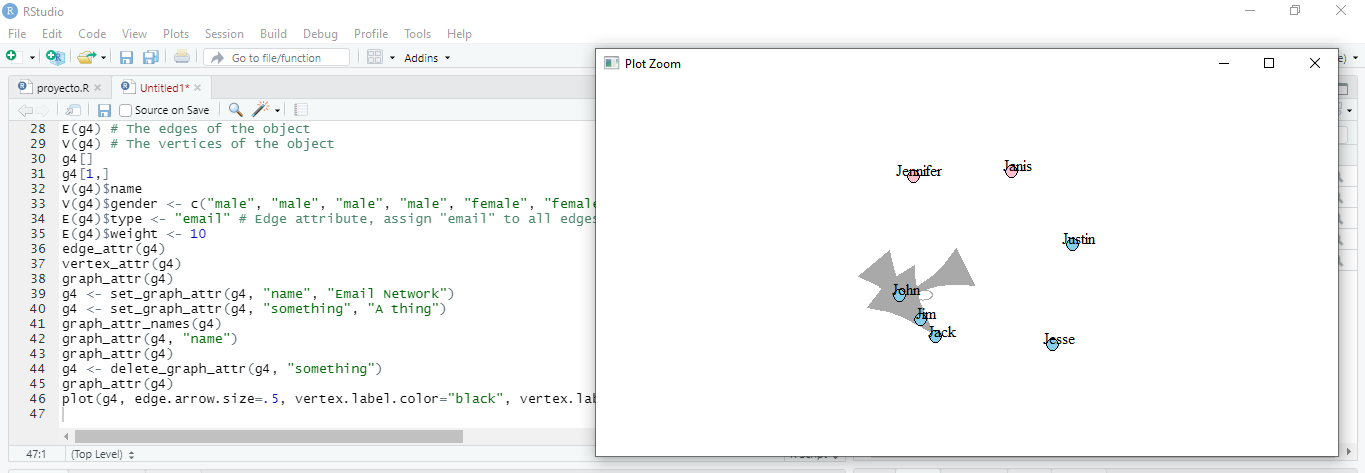
graph\_attr(g4, "name")

graph\_attr(g4)

g4 <- delete\_graph\_attr(g4, "something")

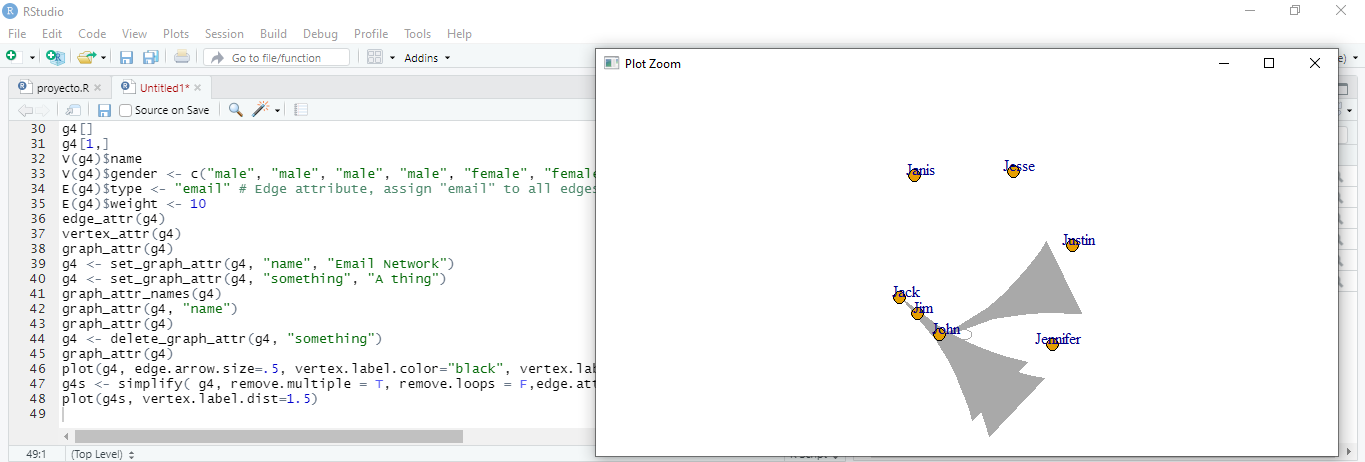
graph\_attr(g4)

plot(g4, edge.arrow.size=.5, vertex.label.color="black", vertex.label.dist=1.5,vertex.color=c( "pink", "skyblue")[1+(V(g4)$gender=="male")] )



g4s <- simplify( g4, remove.multiple = T, remove.loops = F,edge.attr.comb=c(weight="sum", type="ignore") )

plot(g4s, vertex.label.dist=1.5)

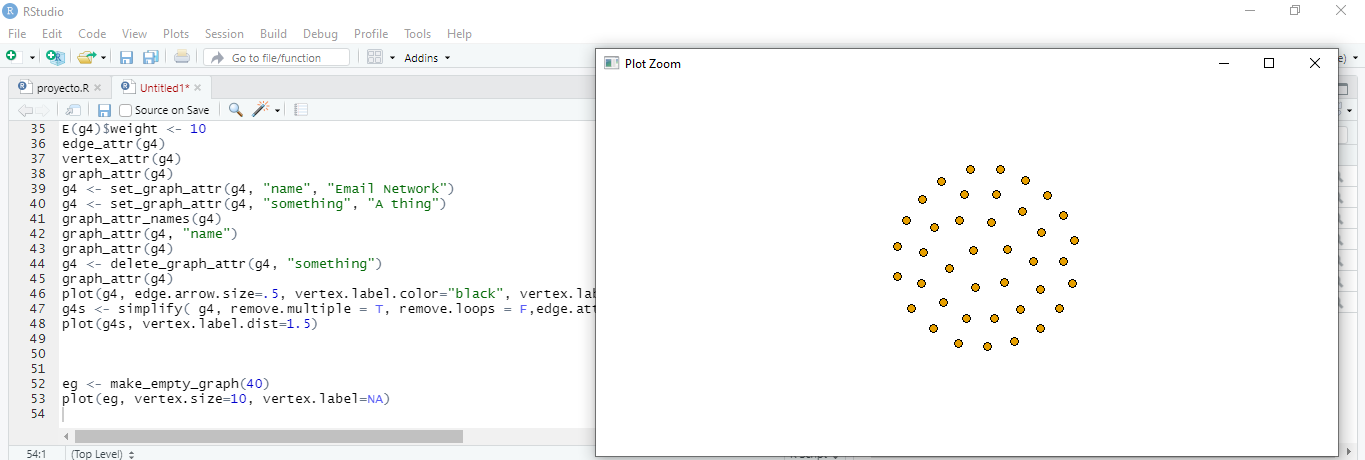


**Specific graphs and graph models**

**Empty graph**

eg <- make\_empty\_graph(40)

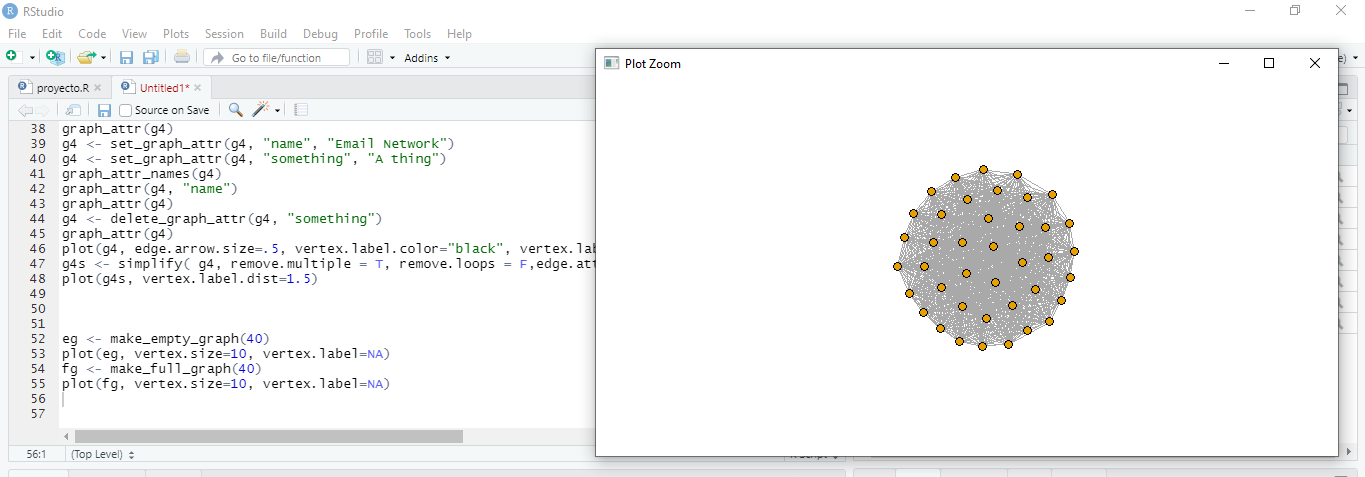
plot(eg, vertex.size=10, vertex.label=NA)



**Full graph**

fg <- make\_full\_graph(40)

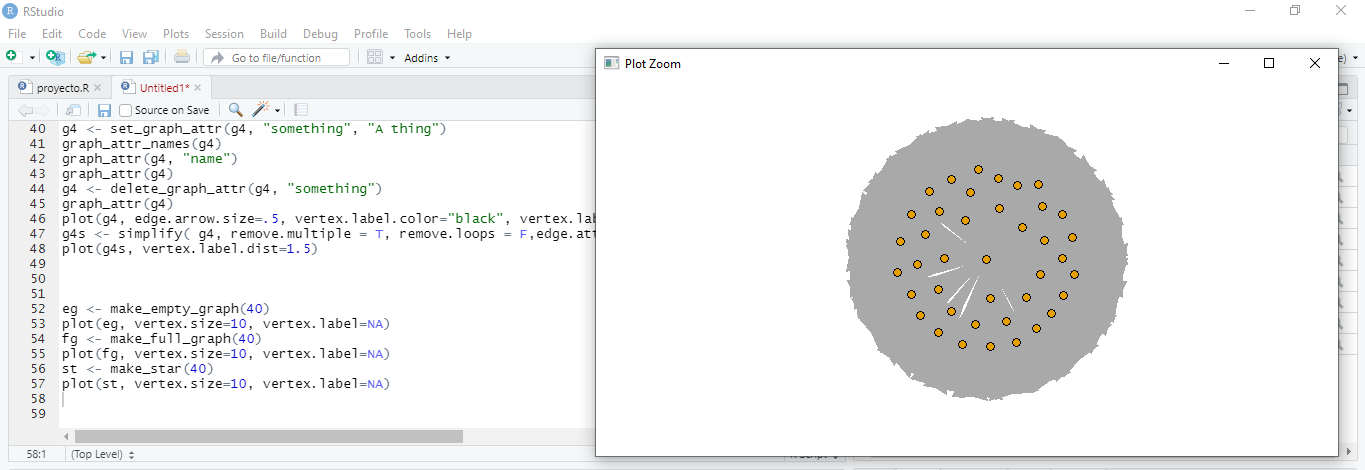
plot(fg, vertex.size=10, vertex.label=NA)



**Simple star graph**

st <- make\_star(40)

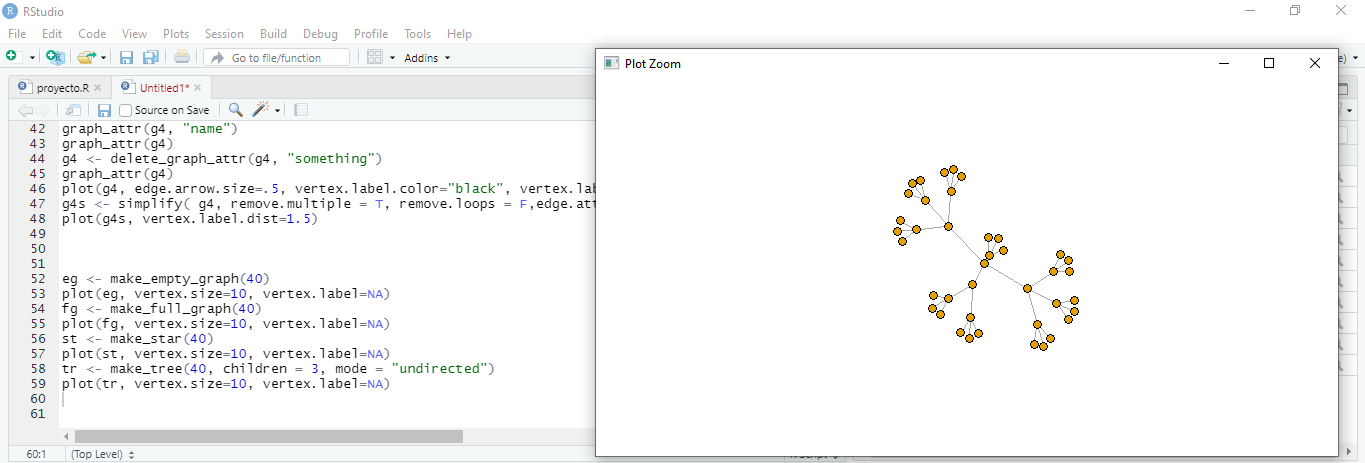
plot(st, vertex.size=10, vertex.label=NA)



**Tree graph**

tr <- make\_tree(40, children = 3, mode = "undirected”)

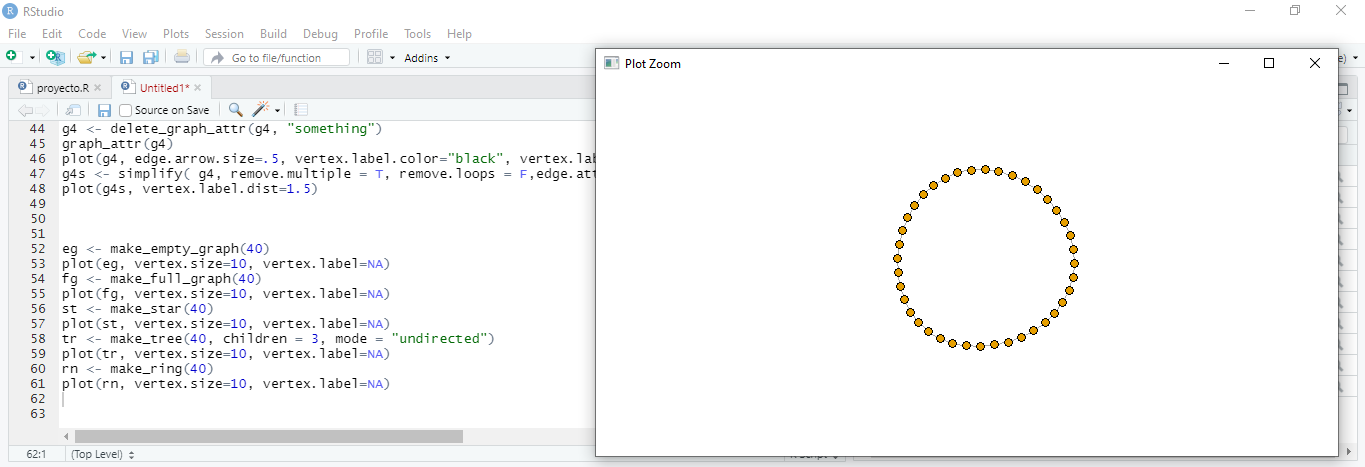
plot(tr, vertex.size=10, vertex.label=NA)



**Ring graph**

rn <- make\_ring(40)

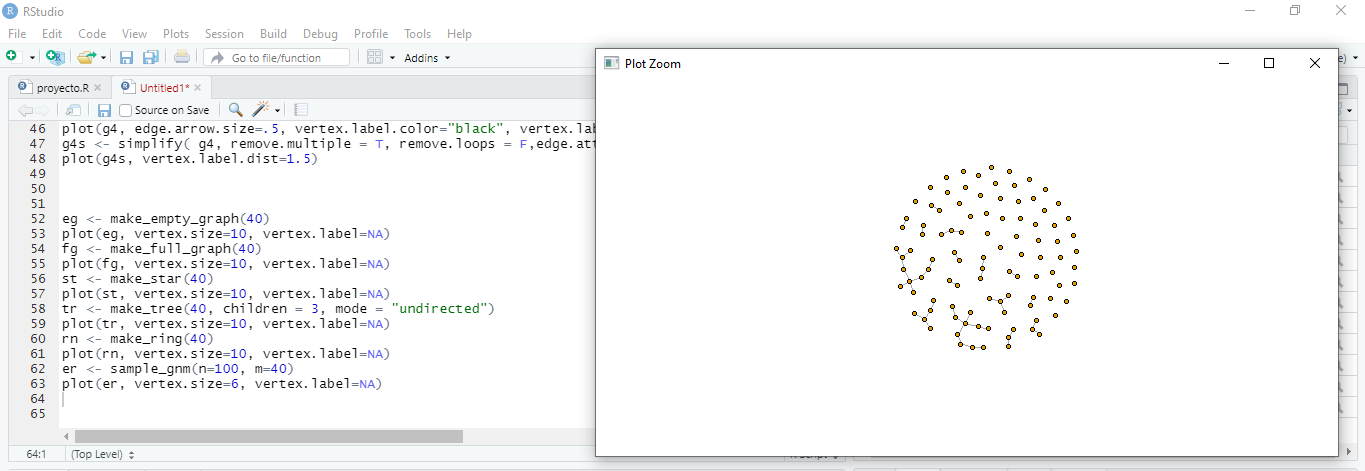
plot(rn, vertex.size=10, vertex.label=NA)



**Erdos-Renyi random graph model**

er <- sample\_gnm(n=100, m=40)

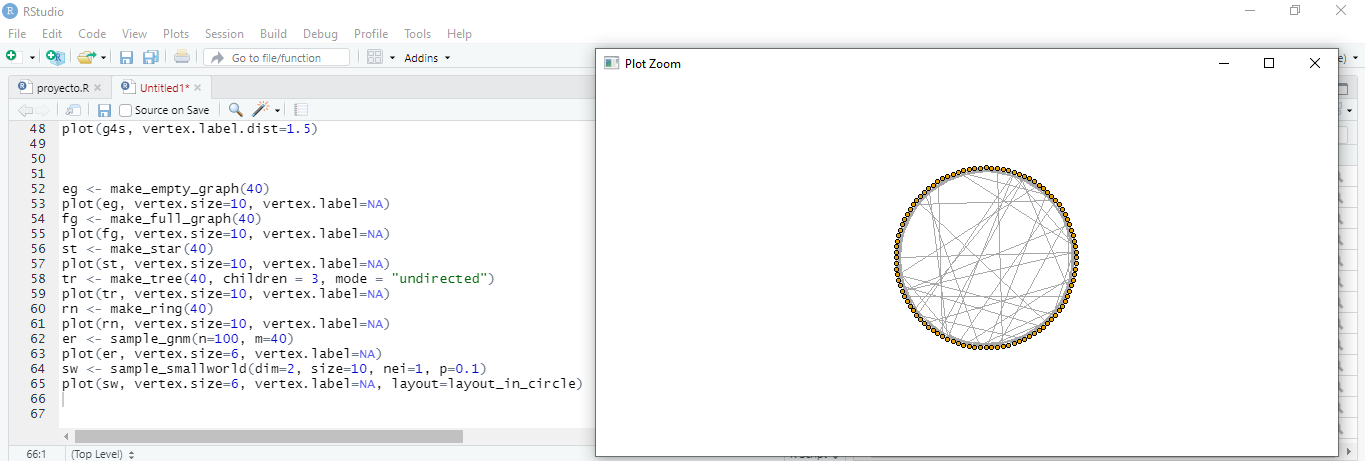
plot(er, vertex.size=6, vertex.label=NA)



**Watts-Strogatz small-world model**

sw <- sample\_smallworld(dim=2, size=10, nei=1, p=0.1)

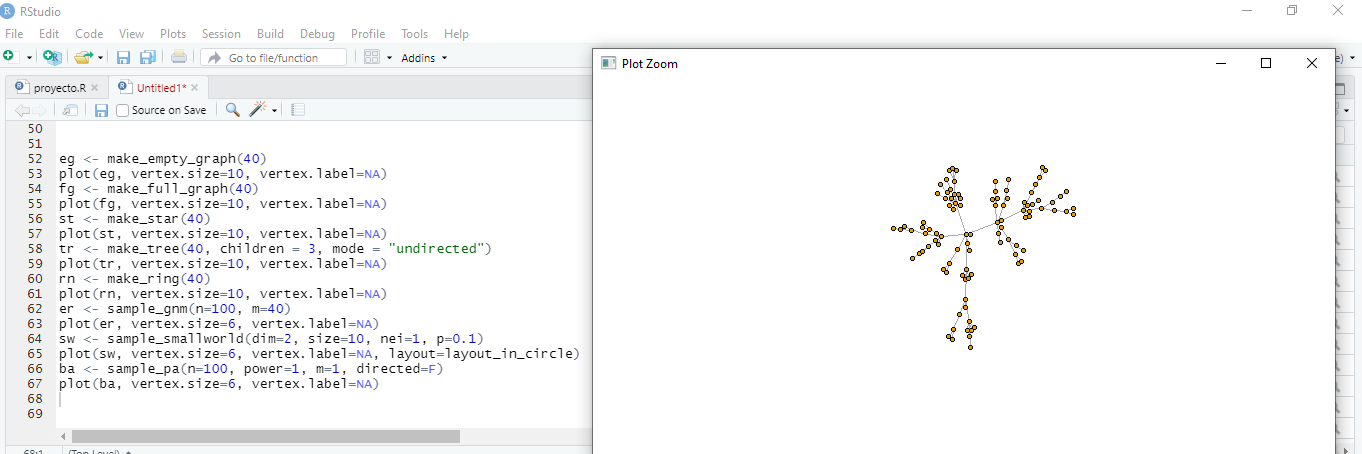
plot(sw, vertex.size=6, vertex.label=NA, layout=layout\_in\_circle)



**Barabasi-Albert preferential attachment model for scale-free graphs**

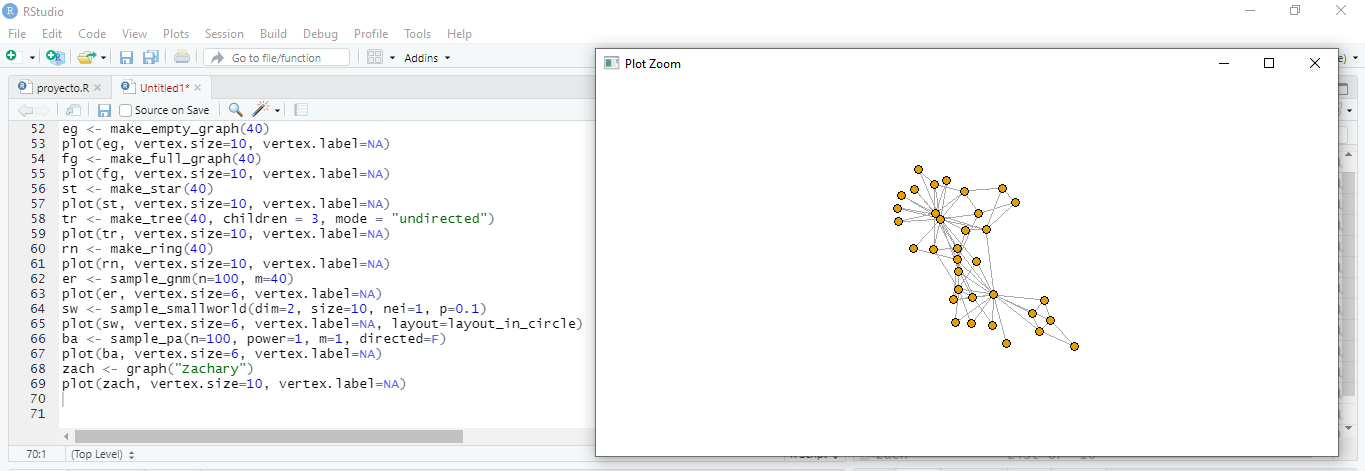
ba <- sample\_pa(n=100, power=1, m=1, directed=F)

plot(ba, vertex.size=6, vertex.label=NA)



zach <- graph("Zachary")

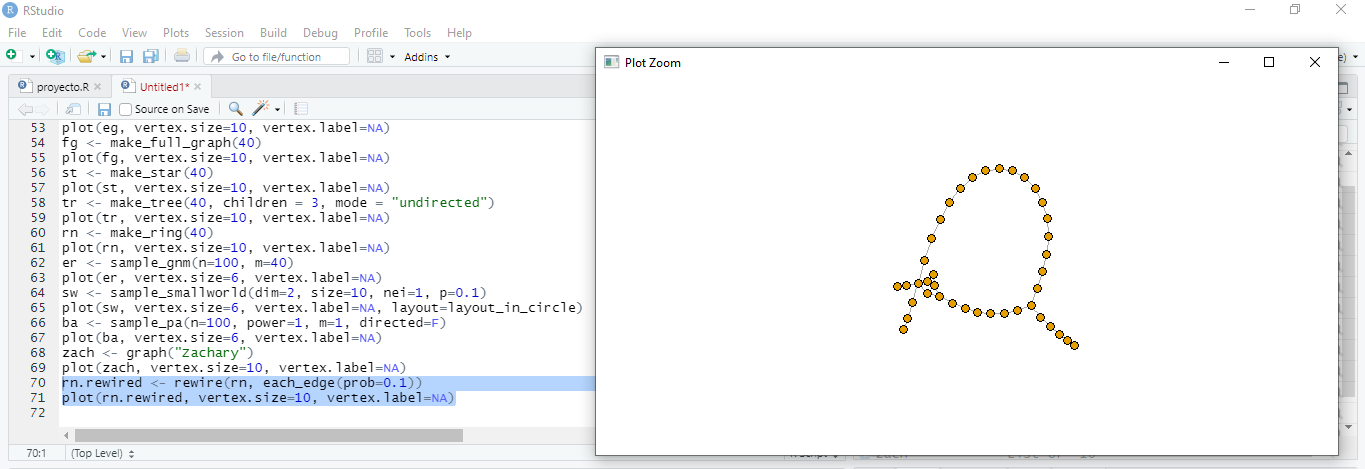
plot(zach, vertex.size=10, vertex.label=NA)



**Rewiring a graph**

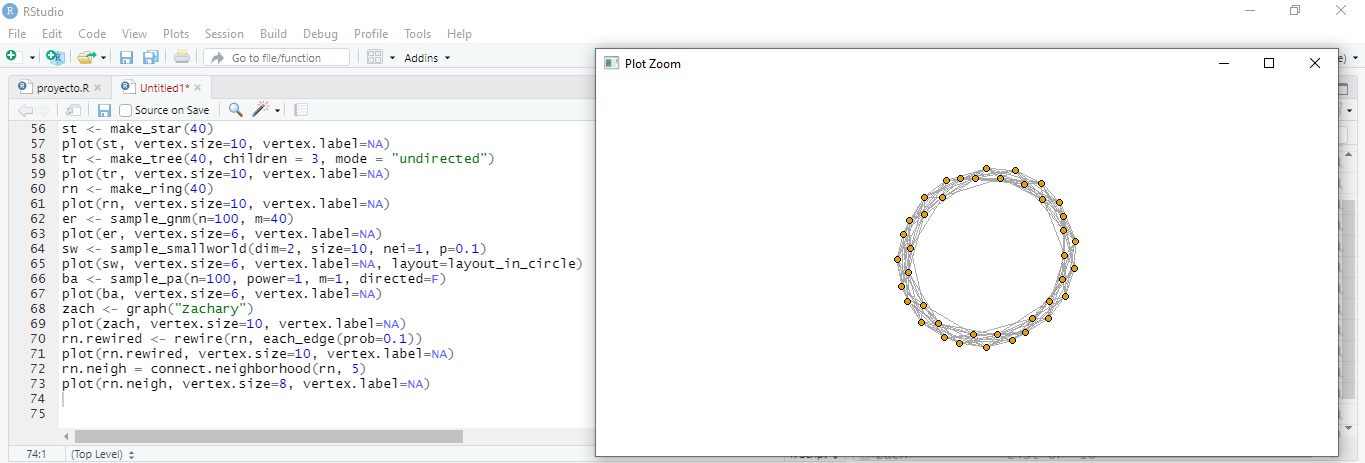
rn.rewired <- rewire(rn, each\_edge(prob=0.1))

plot(rn.rewired, vertex.size=10, vertex.label=NA)

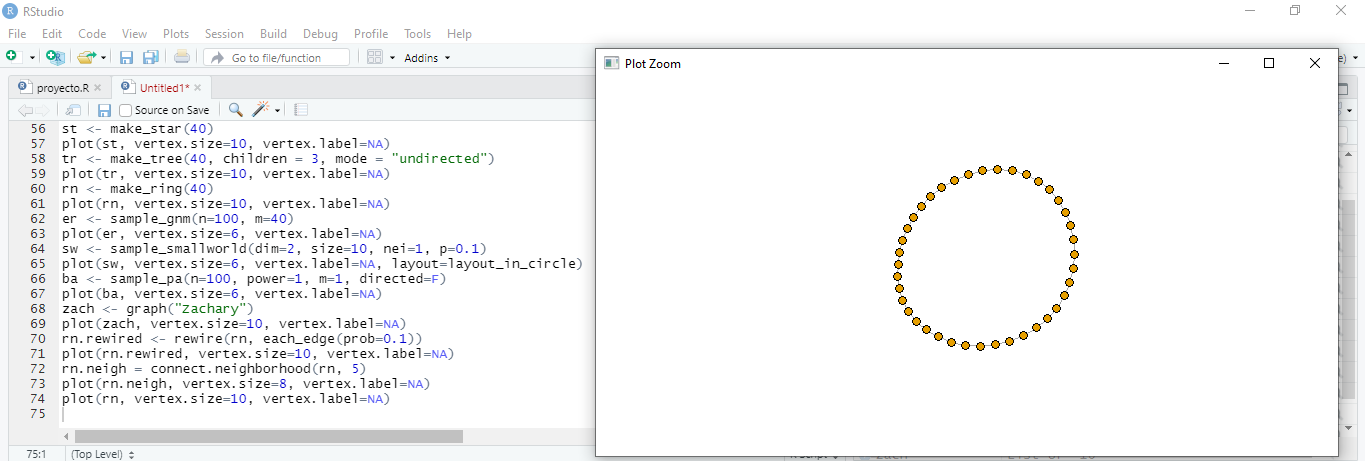


rn.neigh = connect.neighborhood(rn, 5)

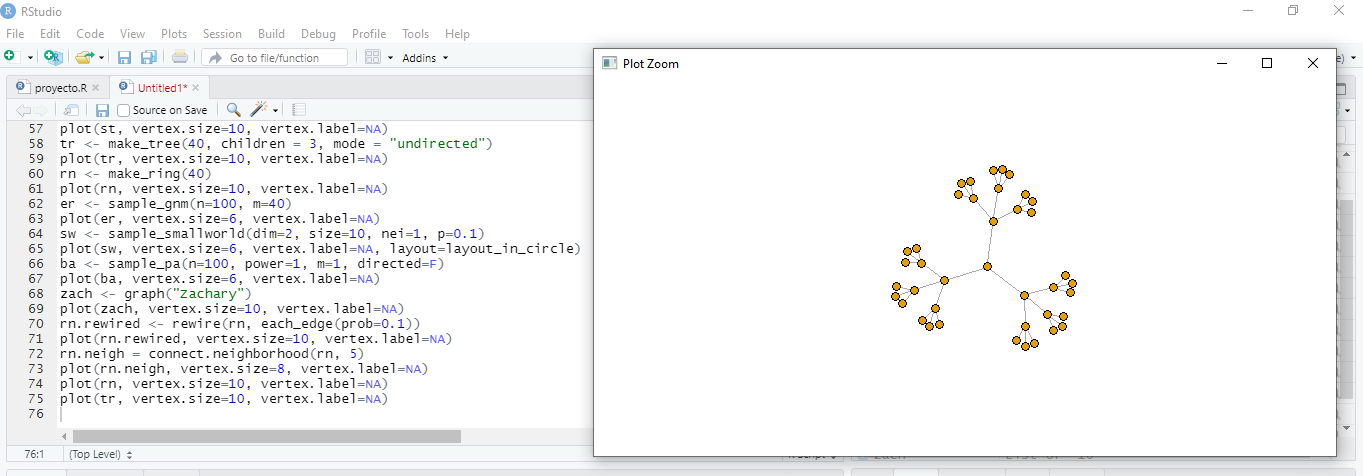
plot(rn.neigh, vertex.size=8, vertex.label=NA)



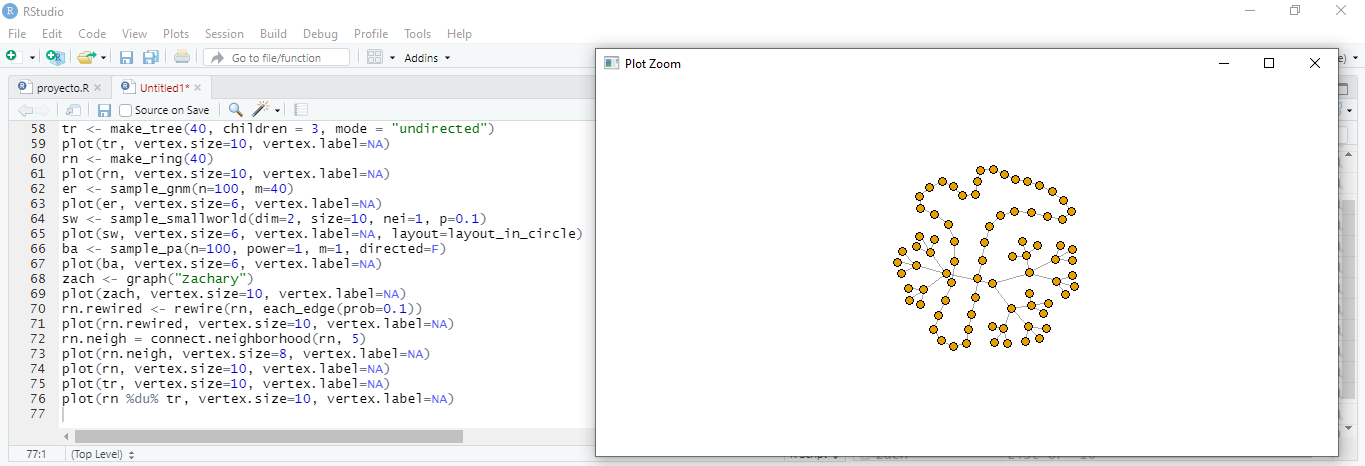
plot(rn, vertex.size=10, vertex.label=NA)



plot(tr, vertex.size=10, vertex.label=NA)



plot(rn %du% tr, vertex.size=10, vertex.label=NA)



**Capítulo 3 - Reading network data**

nodes <- read.csv("D:\\Maestria de Sofware\\Maestria-de-Sofware\\Base de datos no sql\\R\\Data files\\Dataset1-Media-Example-NODES.csv", header=T, as.is=T)

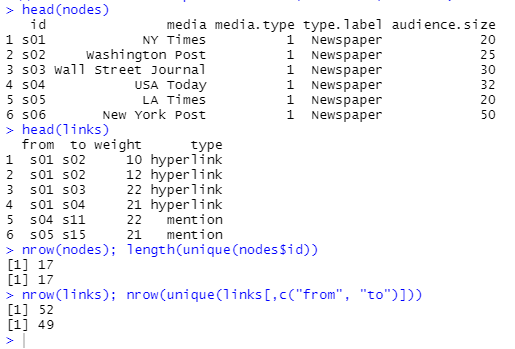
links <- read.csv("D:\\Maestria de Sofware\\Maestria-de-Sofware\\Base de datos no sql\\R\\Data files\\Dataset1-Media-Example-EDGES.csv", header=T, as.is=T)

head(nodes)

head(links)

nrow(nodes); length(unique(nodes$id))

nrow(links); nrow(unique(links[,c("from", "to")]))

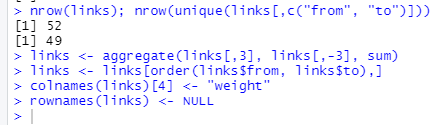


links <- aggregate(links[,3], links[,-3], sum)

links <- links[order(links$from, links$to),]

colnames(links)[4] <- "weight"

rownames(links) <- NULL



library(igraph)

net <- graph\_from\_data\_frame(d=links, vertices=nodes, directed=T)

class(net)

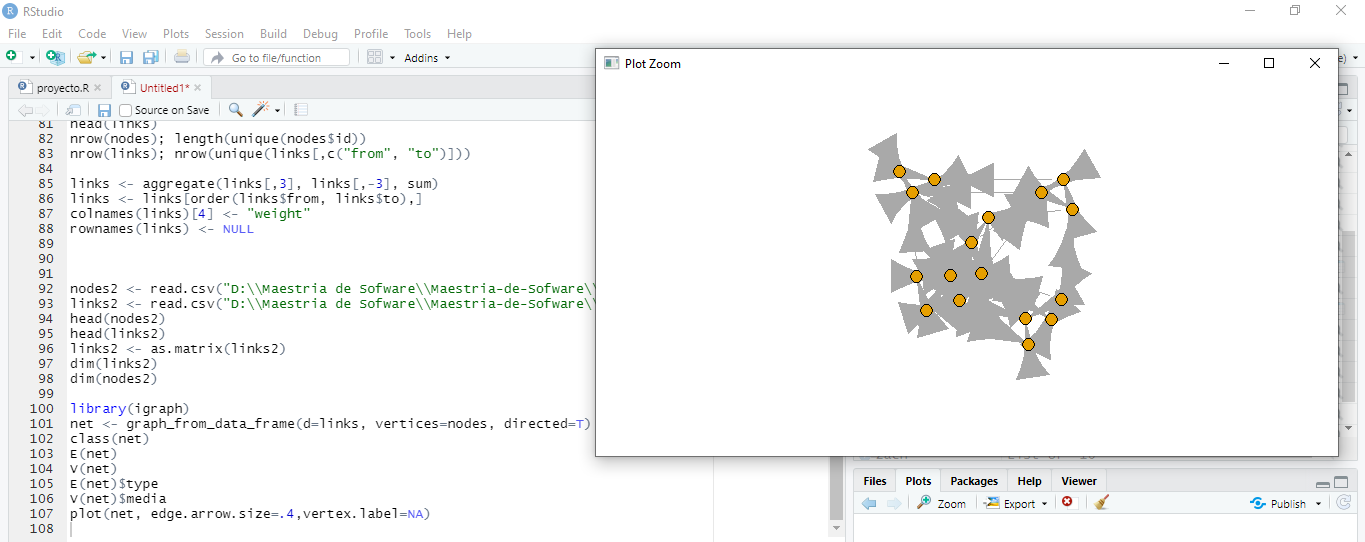
E(net)

V(net)

E(net)$type

V(net)$media

plot(net, edge.arrow.size=.4,vertex.label=NA)



nodes2 <- read.csv("D:\\Maestria de Sofware\\Maestria-de-Sofware\\Base de datos no sql\\R\\Data files\\Dataset1-Media-Example-NODES.csv", header=T, as.is=T)

links2 <- read.csv("D:\\Maestria de Sofware\\Maestria-de-Sofware\\Base de datos no sql\\R\\Data files\\Dataset1-Media-Example-EDGES.csv", header=T, as.is=T)

head(links2)

head(links2)

net2 <- graph\_from\_incidence\_matrix(links2)

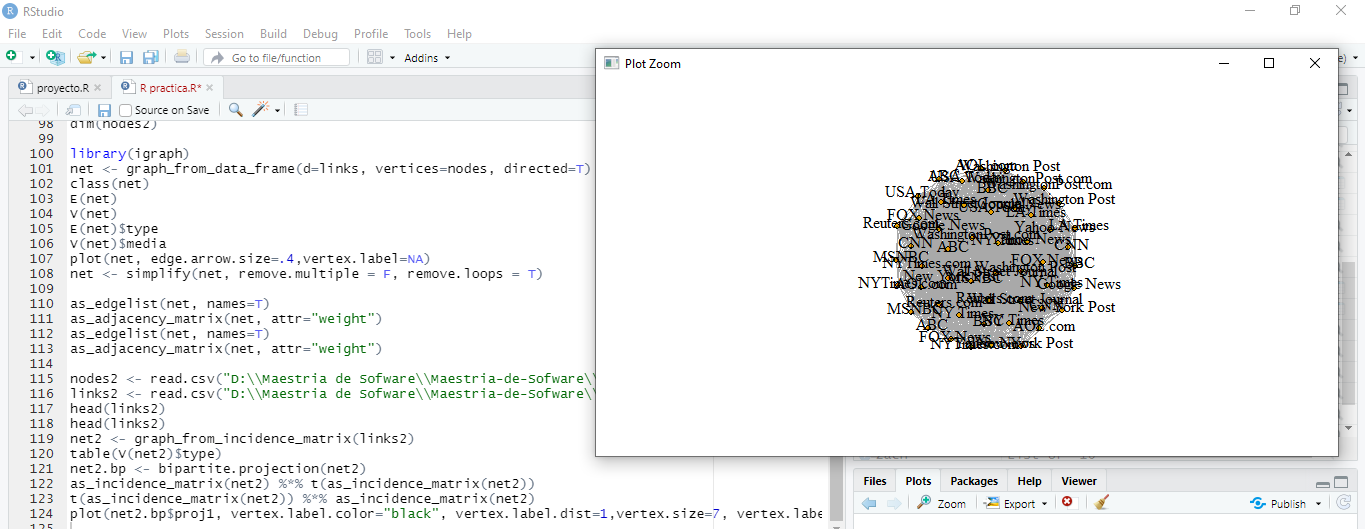
table(V(net2)$type)

net2.bp <- bipartite.projection(net2)

as\_incidence\_matrix(net2) %\*% t(as\_incidence\_matrix(net2))

t(as\_incidence\_matrix(net2)) %\*% as\_incidence\_matrix(net2)

plot(net2.bp$proj1, vertex.label.color="black", vertex.label.dist=1,vertex.size=7, vertex.label=nodes2$media[!is.na(nodes2$media.type)])



plot(net2.bp$proj2, vertex.label.color="black", vertex.label.dist=1,vertex.size=7, vertex.label=nodes2$media[ is.na(nodes2$media.type)])

