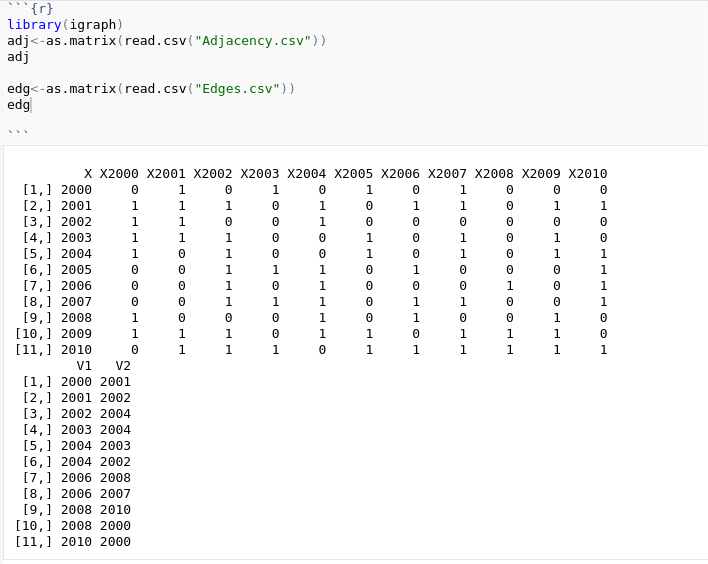
1.

2.



3.

dat=read.csv(file.choose(),header=TRUE,sep=',', row.names=1,check.names=FALSE)

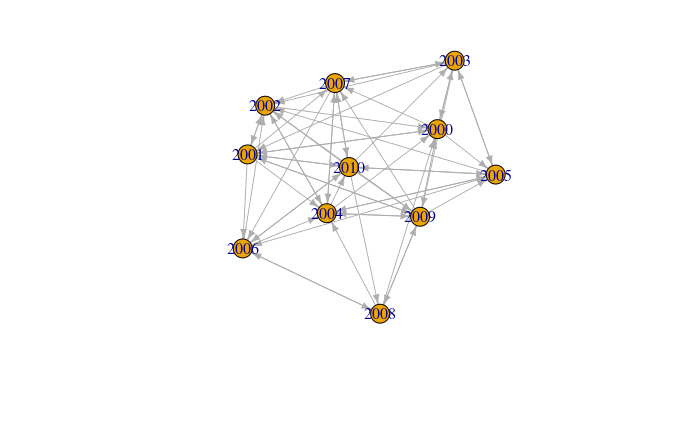
m=as.matrix(dat)

net=graph.adjacency(m,mode="directed",weighted=TRUE,diag=FALSE)

plot.igraph(net,vertex.label=V(net)$name,layout=layout.fruchterman.reingold,

edge.arrow.size=0.5)

net



IGRAPH c8bf0fb DNW- 11 59 --

+ attr: name (v/c), weight (e/n)

+ edges from c8bf0fb (vertex names):

[1] 2000->2001 2000->2003 2000->2005 2000->2007 2001->2000 2001->2002 2001->2004 2001->2006 2001->2007 2001->2009 2001->2010

[12] 2002->2000 2002->2001 2002->2004 2003->2000 2003->2001 2003->2002 2003->2005 2003->2007 2003->2009 2004->2000 2004->2002

[23] 2004->2005 2004->2007 2004->2009 2004->2010 2005->2002 2005->2003 2005->2004 2005->2006 2005->2010 2006->2002 2006->2004

[34] 2006->2008 2006->2010 2007->2002 2007->2003 2007->2004 2007->2006 2007->2010 2008->2000 2008->2004 2008->2006 2008->2009

[45] 2009->2000 2009->2001 2009->2002 2009->2004 2009->2005 2009->2007 2009->2008 2010->2001 2010->2002 2010->2003 2010->2005

[56] 2010->2006 2010->2007 2010->2008 2010->2009

4.

> V(net)

+ 11/11 vertices, named, from ad2dcbc:

[1] 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010

> E(net)

+ 43/43 edges from ad2dcbc (vertex names):

[1] 2000--2001 2000--2002 2000--2003 2000--2004 2000--2005 2000--2007 2000--2008 2000--2009 2001--2002 2001--2003 2001--2004

[12] 2001--2006 2001--2007 2001--2009 2001--2010 2002--2003 2002--2004 2002--2005 2002--2006 2002--2007 2002--2009 2002--2010

[23] 2003--2005 2003--2007 2003--2009 2003--2010 2004--2005 2004--2006 2004--2007 2004--2008 2004--2009 2004--2010 2005--2006

[34] 2005--2009 2005--2010 2006--2007 2006--2008 2006--2010 2007--2009 2007--2010 2008--2009 2008--2010 2009--2010

>

5. Find the count of vertices and edges of the created graph

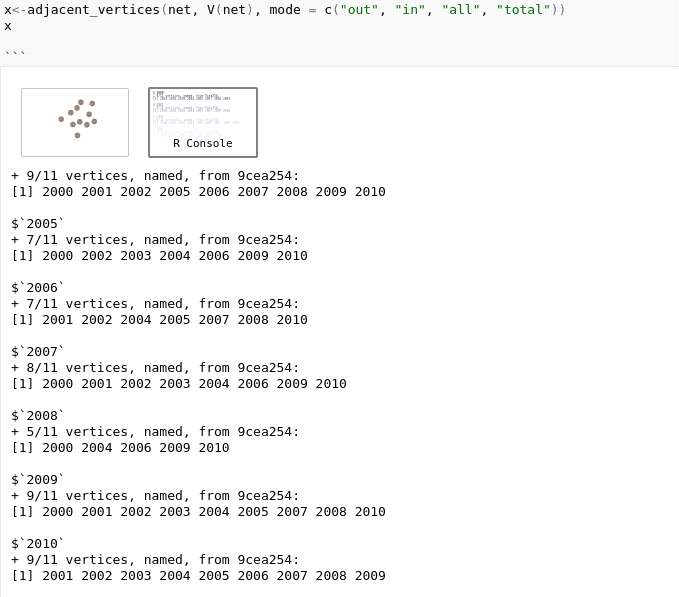
> vcount(net)

[1] 11

> ecount(net)

[1] 43

6.Display the adjacency vertices of each vertex(individual) in the created graph



7.

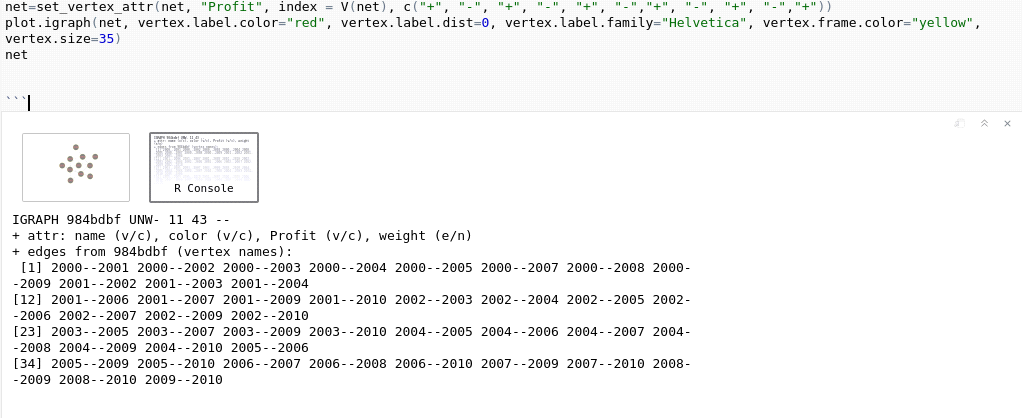
degree(net, v = V(net), mode = c("all", "out", "in", "total"),

loops = TRUE, normalized = FALSE)

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010

8 8 9 7 9 7 7 8 5 9 9

8.



9.



10.

e=set\_edge\_attr(net, "Weight", index=V(net), c(1,1,1,1,1,1,1,1,1,5,1))

plot.igraph(net, vertex.label.color="red", vertex.label.dist=0, vertex.label.family="Helvetica",

vertex.frame.color="yellow", vertex.size=35)

e

IGRAPH e0bcb71 UNW- 11 43 --

+ attr: name (v/c), Type (v/c), weight (e/n), Weight (e/n)

+ edges from e0bcb71 (vertex names):

[1] 2000--2001 2000--2002 2000--2003 2000--2004 2000--2005 2000--2007

[7] 2000--2008 2000--2009 2001--2002 2001--2003 2001--2004 2001--2006

[13] 2001--2007 2001--2009 2001--2010 2002--2003 2002--2004 2002--2005

[19] 2002--2006 2002--2007 2002--2009 2002--2010 2003--2005 2003--2007

[25] 2003--2009 2003--2010 2004--2005 2004--2006 2004--2007 2004--2008

[31] 2004--2009 2004--2010 2005--2006 2005--2009 2005--2010 2006--2007

[37] 2006--2008 2006--2010 2007--2009 2007--2010 2008--2009 2008--2010

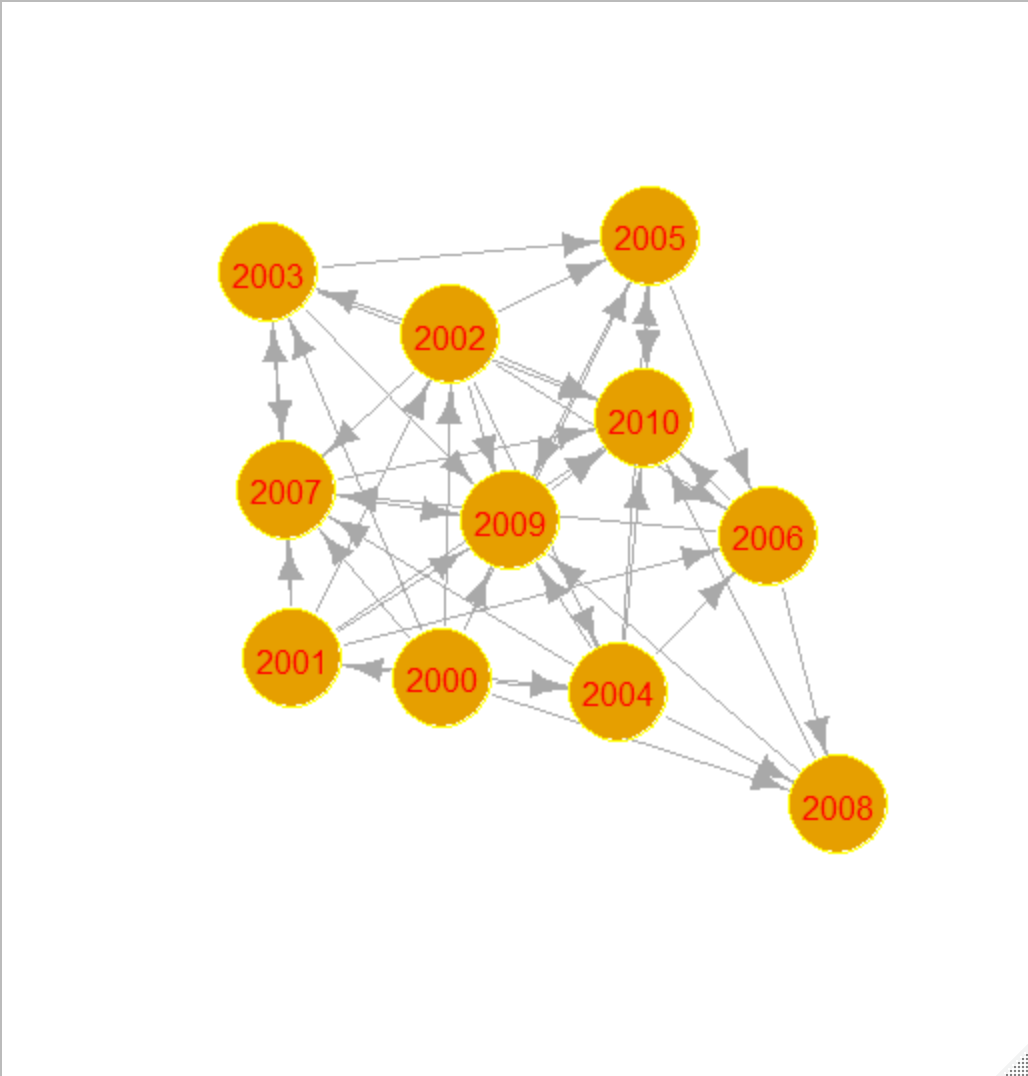
[43] 2009--2010

11.

newnet=as.directed(net, "arbitrary")

plot.igraph(newnet, vertex.label.color="red", vertex.label.dist=0, vertex.label.family="Helvetica",

vertex.frame.color="yellow", vertex.size=35)



12.

get.adjacency(newnet)

11 x 11 sparse Matrix of class "dgCMatrix"

[[ suppressing 11 column names ‘2000’, ‘2001’, ‘2002’ ... ]]

2000 . 1 1 1 1 1 . 1 1 1 .

2001 . . 1 1 1 . 1 1 . 1 1

2002 . . . 1 1 1 1 1 . 1 1

2003 . . . . . 1 . 1 . 1 1

2004 . . . . . 1 1 1 1 1 1

2005 . . . . . . 1 . . 1 1

2006 . . . . . . . 1 1 . 1

2007 . . . . . . . . . 1 1

2008 . . . . . . . . . 1 1

2009 . . . . . . . . . . 1

2010 . . . . . . . . . . .

13.

d=data.frame(

In=dgr(newnet, "indegree", undirected = FALSE),

Out=dgr(newnet, "outdegree", undirected = FALSE))