

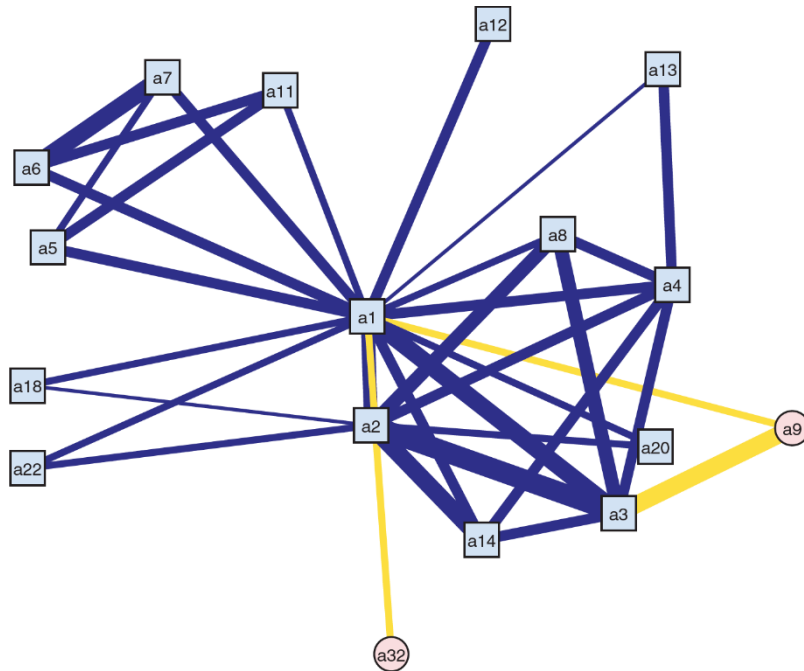
Network HW2

Chenrui Xu

Background and Short Introduction

We were asked to draw the network plot shown below. There are some differences we should pay attention to during the process of plotting compared with the first homework that, instead of drawing a simple plot without any non-default elements, we should set the node and edge attribute by ourselves. Those attributes include node color, node size, node shape, node label, edge color, edge width, edge type, etc. What is more, we can change the mode of plots by using different methods to improve the layout of the plot.

Before beginning the project, some principles we should know at first. An effective network graph should illustrate the essential information of the network. However, irrelevant information should be less highlighted in the graph (to avoid confusion). We should make sure that the network has minimized edge crossings, maximize the symmetry of the layout of the nodes and has the minimizes the number of lengths of edges, etc.



Part I: constructing the dataset of the network

The first step is constructing the dataset of the network. As the network in this case can be very complicated if we use adjacent matrix, so I decided to use edge list method instead to turn the dataset into network matrix (set directed equals to false).

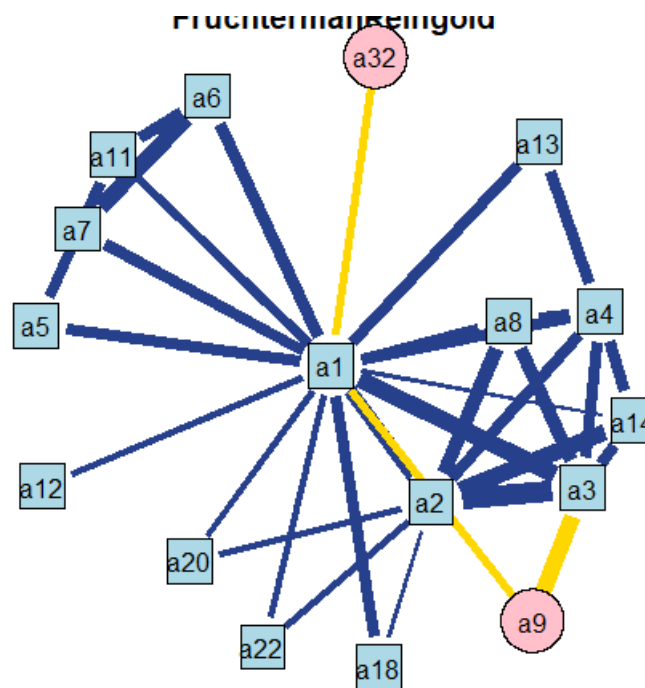
Part II: create an approximately similar graph

In this part, we were asked to create an approximately similar graph based on our network data set. At first, we should set some attributes mentioned in the introduction. We set the general node color to be light-blue and the special node color into pink. Then, I began to set the shape of nodes. It showed that in the plot that the regular shape should be square and the special nodes should be a circle. So, I set the numbers of edges to be 4 and 10000 (which can represent circle as the number of the edges can be seen as infinity). We do not need to consider the size of the node in

this part as we will hire the degree centrality method in the next part to adjust the size of nodes.

In terms of edge, we can set the color to be royal-blue and gold. There is a total of 34 edges in the network and as the widths are different from each other, we should set one by one. I set six kinds of edge width in the case and they are 2,4,6,8,10,12. And the order should be the same as the edge list (10,12,8,8,8,8,6,6,6,4,8,2,8,4,4,4,16,8,10,12,2,4,4,8,10,12,8,8,8,8,4,8,12,8).

After setting those attributes, we can extract them from the network and name after them. Then, we are good to draw the plot.



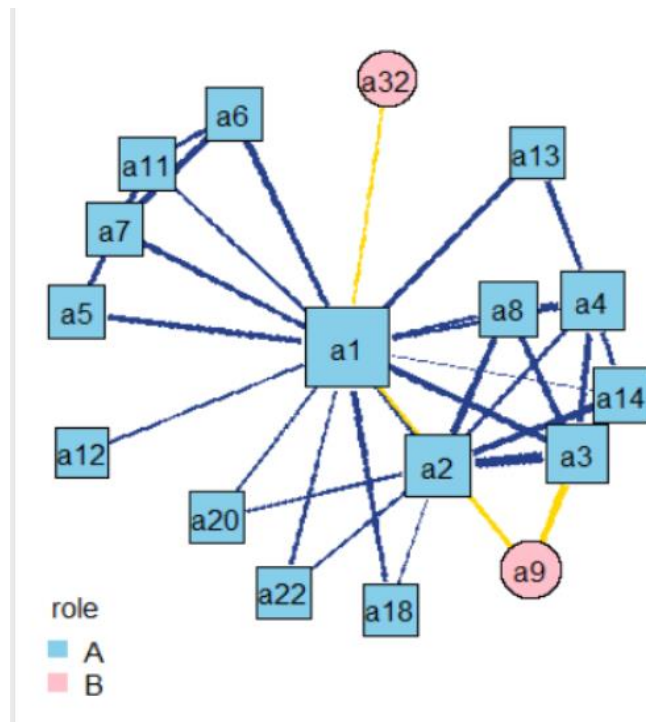
Here is the plot we draw via R. The algorithm we used is “fruchtermanreingold”, which is a good choice to make plot looks great. And there are some other settings shown below:

- the mode is graph
- the size of the vertex is 2.5
- after setting the shape of the nodes, I used vertex.rot = 45 to make sure the square won't be regarded as a diamond.

Part III Some Improvements

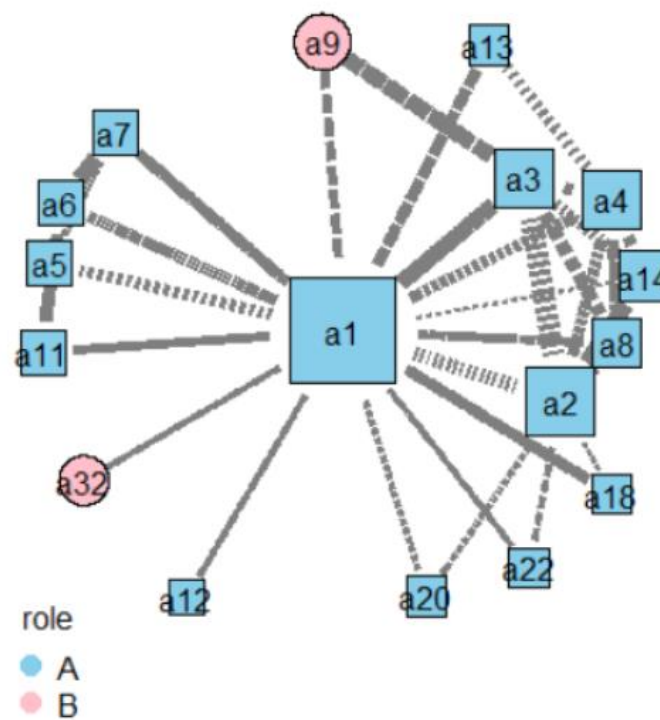
In this part, we were asked to use degree centrality to adjust the size of nodes appropriately and try to improve the graph using different edge type (solid, dashed, dotted or dot-dashed).

At first, we should use the degree function to get the size of each node. Then, a rescale function was used. As we want to use different types of edges, we randomly generated 34 numbers to represent 4 kinds of edges. All other settings are similar to the last part.



The plot is similar with the part two but we got different sizes of nodes. What is more, I set legend at the left bottom to make sure that the plot can be illustrated well.

Aiming to show different edges, we should use gplot function instead of plot function.



There are four kinds of edges in this plot.

Appendix

```
title: "Network_HW2"
author: "Chenrui Xu"
date: "2021/2/13"
output: html_document
```

```
```{r}
library(statnet)
library(UserNetR)
library(RColorBrewer)
set.seed(123)
```
```

Part 1 setting matrix

```
```{r}
netmat=rbind(c(1,2), #1
 c(1,3), #2
 c(1,4), #3
 c(1,5), #4
 c(1,6), #5
 c(1,7), #6
 c(1,8), #7
 c(1,9), #8
 c(1,10),#9
 c(1,11),#10
 c(1,12),#11
 c(1,13),#12
 c(1,14),#13
 c(1,15),#14
 c(1,16),#15
 c(1,17),#16
 #c(2,1), #17 #1
 c(2,3), #18
 c(2,4), #19
 c(2,8), #20
 c(2,13),#21
 c(2,14),#22
 c(2,15),#23
 c(2,16),#24
 #c(3,1), #25 #2
 #c(3,2), #26 #18
 c(3,4), #27
 c(3,8), #28
 c(3,9), #29
 c(3,13),#30
 #c(4,1), #31 #3
 #c(4,2), #32 #19
```



```
e','lightblue',"pink",'lightblue','lightblue','lightblue','lightblue','lightblue','lightblue','lightblue','pink'))
...

```

Give nodes the role attributes and set nodes colors.

```
```{r}
#set.vertex.attribute(net,"shape",c(4,4,4,4,4,4,4,10000,4,4,4,4,4,4,10000))
set.vertex.attribute(net,"shape",c(1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2))
...

```

Give nodes the shape attributes as 4 is the square and circle we can view it as infinite of edges shape square.

```
```{r}
set.edge.attribute(net,"ecolor",c('royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','gold1','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','gold1','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','gold1','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4','royalblue4'))
...

```

in terms of edges, we can set color to be royalblue and gold

The setting of the width, we put it into gplot() and according to the plot shown in the Powerpoint, we can set the numbers.

```
```{r}
set.seed(123)
rolelab=net%v%"role"
shape=net%v%"shape"
my_pal <- c('lightblue','pink')
ecolor=net%e%"ecolor"
op <- par(mar=c(0,0,0,0))
gplot(net,gmode="graph",
      mode="fruchtermanreingold",
      vertex.cex=2.5,
      main="FruchtermanReingold",
      vertex.col =rolelab,
      displaylabels=T,
      label.pos=5,
      vertex.sides =c(4,10000)[shape],vertex.rot=45,#let square shape rotate 45
      edge.lwd=c(10,12,8,8,8,8,6,6,6,4,8,2,8,4,4,4,16,8,10,12,2,4,4,8,10,12,8,8,8,8,4,8,12,8),
              # 1  2  3 4 5 6 7 8 9 0 1 2 3 4 5 6 7  8 9  0  1 2 3 4 5  6  7 8 9 0 1 2 3  4
              # 0                                1                                2                                3
      edge.col = ecolor
    )
par(op)
...

```

Part 3

```
```{r}
deg=degree(net,gmode = "graph")
deg

```



```

legend("bottomleft",
 legend=c("A","B"),
 col=my_pal,pch=19,
 pt.cex=1.5,
 bty="n",
 title="role"
)
par(op)
```

```{r}
op=par(mar=c(0,0,0,0),mfrow=c(1,2))
gplot(net,
 mode="fruchtermanreingold",
 usearrows=F,
 vertex.cex=rescale(deg,2,6),
 main="Adjusted node sizes with rescale function",
 displaylabels=T,
 edge.lty=c(2,3,4,5)[edge_cat],
 vertex.col=my_pal[rolecat],
 label.pos=5,
 vertex.sides =c(4,10000)[shape],vertex.rot=45,#let square shape rotate 45
 edge.lwd=c(10,12,8,8,8,8,6,6,6,4,8,2,8,4,4,4,16,8,10,12,2,4,4,8,10,12,8,8,8,8,4,8,12,8),
 edge.col = "gray50"
)
legend("bottomleft",
 legend=c("A","B"),
 col=my_pal,pch=19,
 pt.cex=1.5,
 bty="n",
 title="role"
)
par(op)
```

```