SNA Basics II

Two-Mode Networks in R

@ Sean F. Everton

Place a header at the top of your scripts that tell you what the script does, what its name is, etc.

```
##################################################
# What: Two-Mode Social Network Data in R
# File: snab2.R
# Created: 02.19.14
# Revised: 06.08.18
##################################################
```

Data

The data that we will use here is what is known as Davis's Southern Club Women. Davis and her colleagues recorded the observed attendance of 18 Southern women at 14 different social events.

Setup

Clear the workspace each time before beginning.

```
rm(list=ls())
```

Set your working directory to where the data are, so you don't have to include the entire path when importing and exporting data, files, etc. Of course, you'll need to set this to your own directory.

```
setwd("~/Dropbox/Networks and Religion (Book)/Website/Labs/SNA Basics 2")
```

Two-Mode Social Network Data in *statnet*

Next, we need to load the libraries we plan to use. The *sna* and *network* libraries are part of the *statnet* package; we'll use *igraph* later, but because *igraph* and *sna* conflict with one another, we can't load them at the same time.

```
# Load libraries
library(sna)
library(network)
```

Read the Southern Women (Davis) network into R from a Pajek network file.

```
davis.net <- as.network(read.paj("davis.net"),bipartite=TRUE,directed=FALSE)
```

The previous command reads the data in as a network object (graph). If we want to examine it as a matrix, we need to first transform it into a matrix and then view it.

```
davis.mat <- as.matrix(davis.net)
```

List attributes and actor (vertex) labels

```
list.vertex.attributes(davis.net)
## [1] "na"           "vertex.names" "x"            "y"
## [5] "z"
```

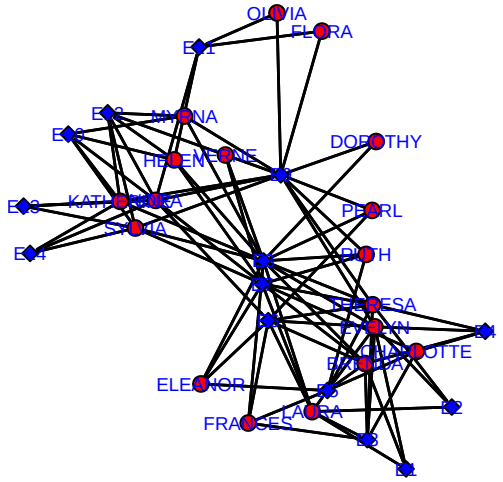
```
get.vertex.attribute(davis.net,"vertex.names")
##  [1] "EVELYN"    "LAURA"     "THERESA"   "BRENDA"    "CHARLOTTE"
##  [6] "FRANCES"   "ELEANOR"   "PEARL"     "RUTH"      "VERNE"
## [11] "MYRNA"     "KATHERINE" "SYLVIA"    "NORA"      "HELEN"
## [16] "DOROTHY"   "OLIVIA"    "FLORA"     "E1"        "E2"
## [21] "E3"        "E4"        "E5"        "E6"        "E7"
## [26] "E8"        "E9"        "E10"       "E11"       "E12"
## [31] "E13"       "E14"
```

Here's an easier way to get the vertex labels

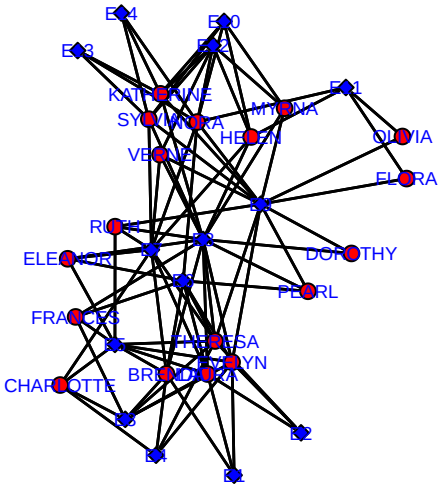
```
network.vertex.names(davis.net)
##  [1] "EVELYN"    "LAURA"     "THERESA"   "BRENDA"    "CHARLOTTE"
##  [6] "FRANCES"   "ELEANOR"   "PEARL"     "RUTH"      "VERNE"
## [11] "MYRNA"     "KATHERINE" "SYLVIA"    "NORA"      "HELEN"
## [16] "DOROTHY"   "OLIVIA"    "FLORA"     "E1"        "E2"
## [21] "E3"        "E4"        "E5"        "E6"        "E7"
## [26] "E8"        "E9"        "E10"       "E11"       "E12"
## [31] "E13"       "E14"
```

Time for a few plots of the data

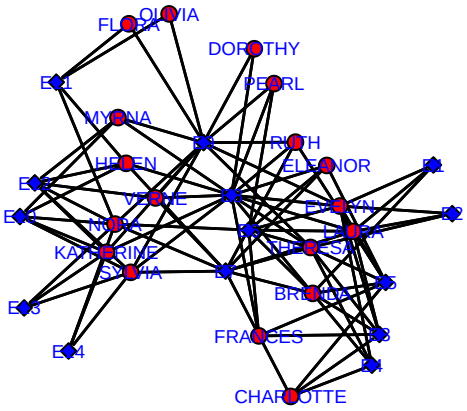
```
gplot(davis.net,gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,usearrows=FALSE)
```



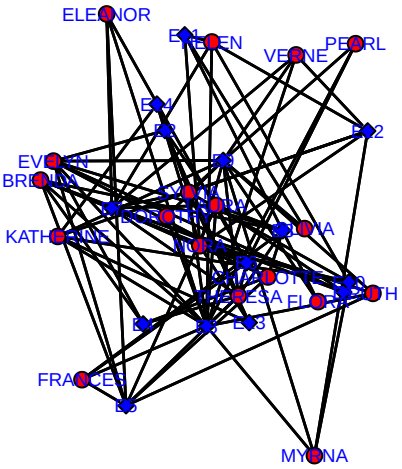

```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
      mode ="fruchtermanreingold",usearrows=FALSE)
```




```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
      mode="kamadakawai",usearrows=FALSE)
```

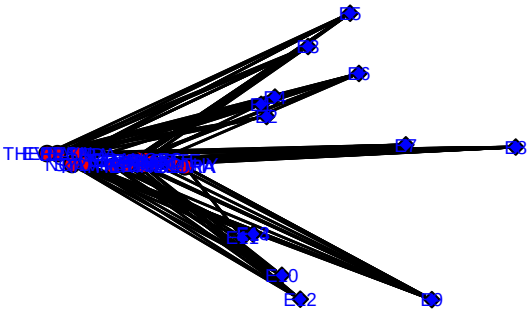



```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
      mode ="spring",usearrows=FALSE)
```

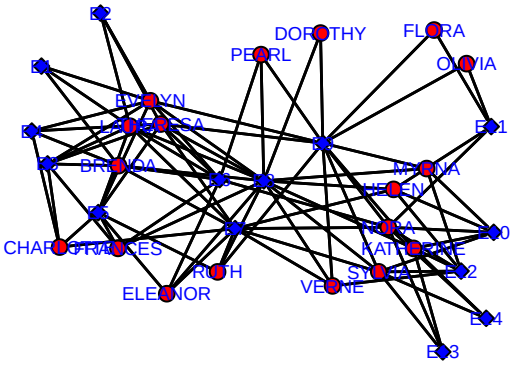



```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
```

```
      mode="mds",usearrows=FALSE)
```



```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,usearrows=FALSE)
```



Let's calculate two-mode degree centrality of the network and then assign the scores (a vector of scores) as actor attributes. Note the change in the output when we list the attributes of the network.

```
deg <- degree(davis.net,gmode="graph")
davis.net%v%"degree" <- deg
list.vertex.attributes(davis.net)
```

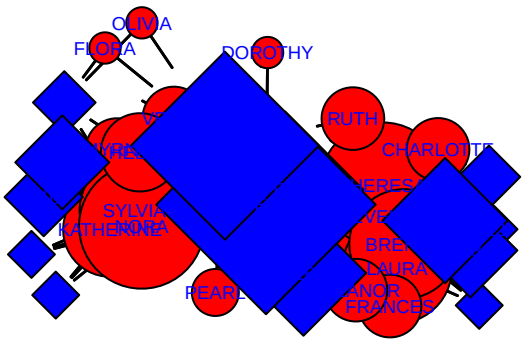
```
## [1] "degree"       "na"           "vertex.names" "x"
## [5] "y"            "z"
```

```
degree(davis.net)
```

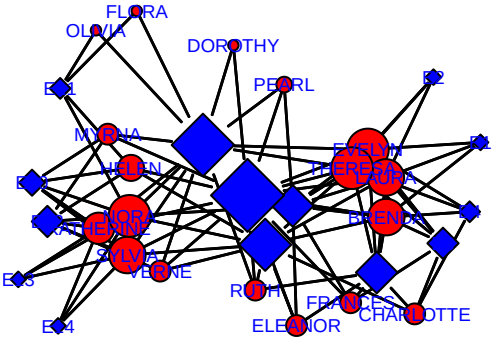
```
##  [1] 16 14 16 14  8  8  8  6  8  8  8 12 14 16 10  4  4  4  6  6 12  8 16
## [24] 16 20 28 24 10  8 12  6  6
```

Plot graph with node size reflecting two-mode degree centrality; the second graph rescales degree to make the graph easier to interpret. The size of the nodes in the first graph are a bit overwhelming.

```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,vertex.cex=deg,
      usearrows=FALSE)
```

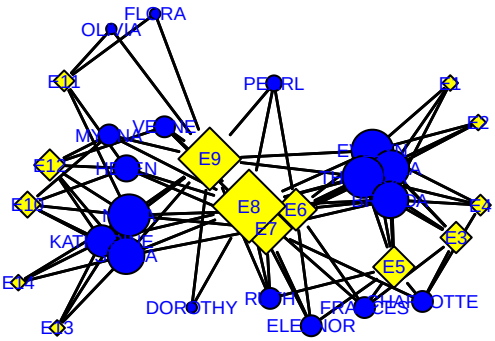



```
gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
      vertex.cex=deg/3,usearrows=FALSE)
```



The default colors for sna are blue and red, so if you want to assign different colors you have to create a separate "color" vector. Here's one way to do it. Set up a vector where the first set of colors is one color (in this case 18 blues for the 18 women), and the second set is the second color (in this case, 14 yellows for the 14 events)

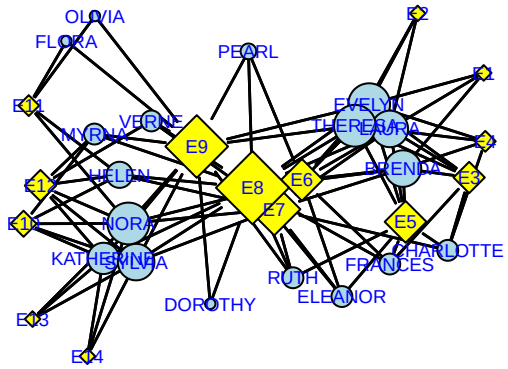
```
color <- c("blue","blue","blue","blue","blue","blue","blue","blue","blue","blue","blue",
           "blue","blue","blue","blue","blue","blue","blue","yellow","yellow","yellow",
           "yellow","yellow","yellow","yellow","yellow","yellow","yellow","yellow",
           "yellow","yellow","yellow")

gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,,label.pos=5,
      vertex.cex=deg/3,usearrows=FALSE,vertex.col=color)
```



That isn't terribly efficient, especially with large large networks. Here's a better way

```
m <- matrix(data="light blue",ncol=18) # Assign the women a light blue color
n <- matrix(data="yellow",ncol=14) # Assign the events a yellow color
```

```
color <- c(m,n) # combine into a single vector

gplot(davis.net, gmode="twomode",label=network.vertex.names(davis.net),
      label.col="blue",label.cex=0.6,label.pos=5,
      vertex.cex=deg/3,usearrows=FALSE,vertex.col=color)
```



We can transform the network into a one-mode network of the women by multiplying the matrix (not the graph) by it's transpose in order to to get a one-mode of the women (rows). To transform the network into a one-mode network of the events, multiply the transpose of the matrix by the matrix.

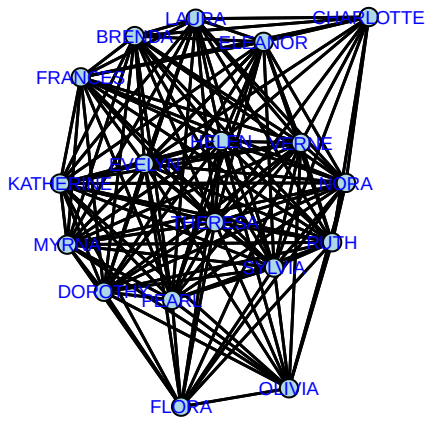
```
daviswomen.mat = davis.mat %*% t(davis.mat)
davisevents.mat = t(davis.mat) %*% davis.mat
```

Convert them to network objects

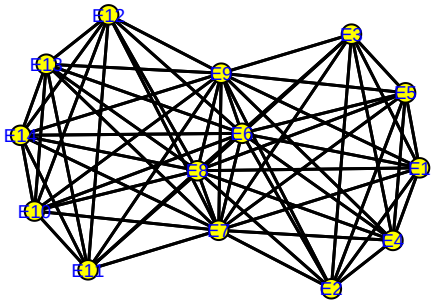
```
daviswomen.net <-as.network(daviswomen.mat)
davisevents.net <-as.network(davisevents.mat)
```

Now plot both of the new networks

```
gplot(daviswomen.net,gmode="onemode",label=network.vertex.names(daviswomen.net),
      label.col="blue",label.cex=0.6,label.pos=5,mode ="kamadakawai",
      usearrows=FALSE,vertex.col="light blue")
```




```
gplot(davisevents.net,gmode="onemode",label=network.vertex.names(davisevents.net),
      label.col="blue",label.cex=0.6,label.pos=5,mode ="kamadakawai",
      usearrows=FALSE,vertex.col="yellow")
```



Two-mode Social Network Data in *igraph*

Now, let's see how we can do all of this in *igraph*. First, we need to detach *sna* and then load *igraph*.

```
detach("package:sna", unload=TRUE)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:network':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges,
##     delete.vertices, get.edge.attribute, get.edges,
##     get.vertex.attribute, is.bipartite, is.directed,
##     list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

Read in the data

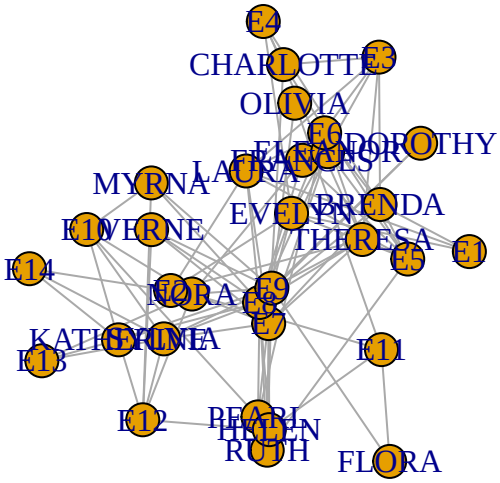
```
davis.ig <- read.graph("davis.net",format = c("pajek"))
```

Tell igraph that the ids (names) should be used as labels (rather than the vertex #s) and then create a few basic plots. As you can see the default settings for igraph plots are not aesthetically pleasing

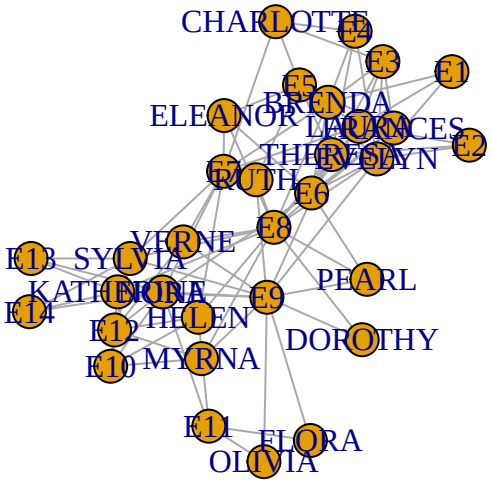
```
V(davis.ig)$id
```

```
##  [1] "EVELYN"    "LAURA"     "THERESA"   "BRENDA"    "CHARLOTTE"
##  [6] "FRANCES"   "ELEANOR"   "PEARL"     "RUTH"      "VERNE"
## [11] "MYRNA"     "KATHERINE" "SYLVIA"    "NORA"      "HELEN"
## [16] "DOROTHY"   "OLIVIA"    "FLORA"     "E1"        "E2"
## [21] "E3"        "E4"        "E5"        "E6"        "E7"
## [26] "E8"        "E9"        "E10"       "E11"       "E12"
## [31] "E13"       "E14"
```

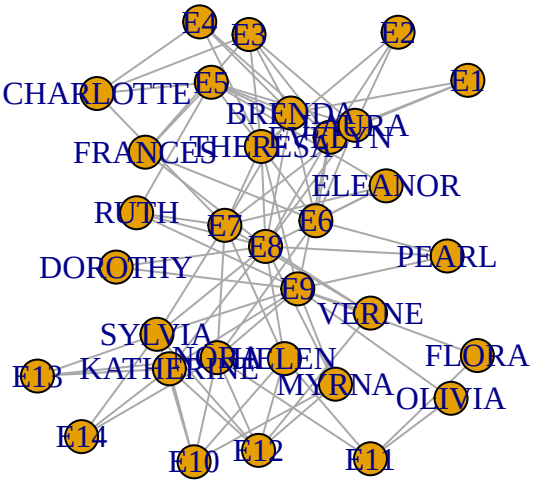
```
V(davis.ig)$label = V(davis.ig)$id

plot(davis.ig)
```




```
plot(davis.ig,layout=layout.fruchterman.reingold)
```

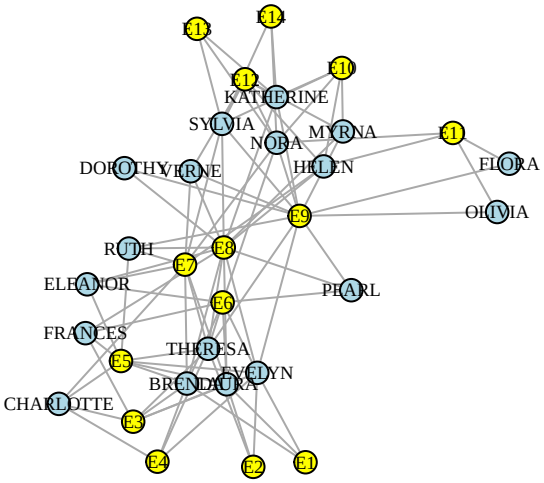



```
plot(davis.ig,layout=layout.kamada.kawai)
```

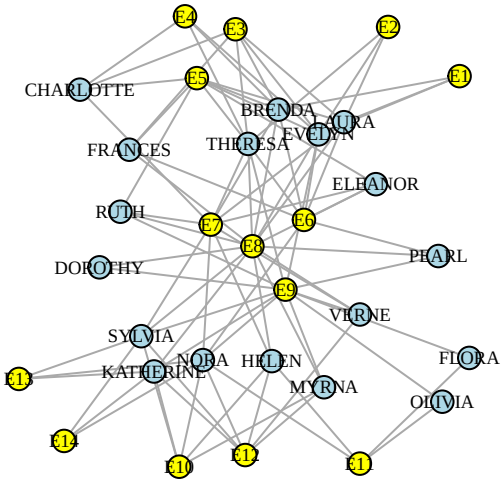


We can add a few options to make the graphs look a little better; we can also assign the same color vector that we created earlier

```
plot(davis.ig,layout=layout.fruchterman.reingold,vertex.size=10,
     vertex.label.cex=.6,vertex.label.color="black",vertex.color=color)
```

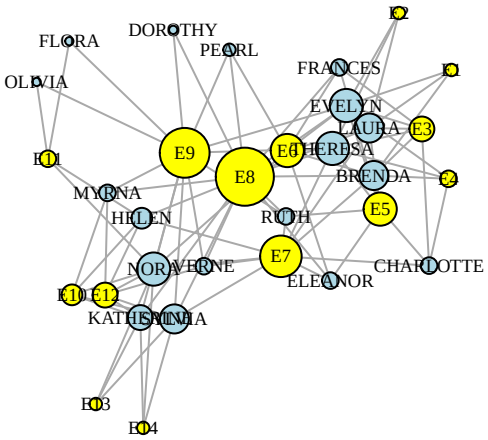



```
plot(davis.ig,layout=layout.kamada.kawai,vertex.size=10,
     vertex.label.cex=.6,vertex.label.color="black",vertex.color=color)
```



Now, let's calculate degree centrality and then plot the graphs again but adjust the node size to reflect degree, which we've rescaled in order to make the nodes more visible

```
deg <- degree(davis.ig)

plot(davis.ig, layout=layout.fruchterman.reingold,vertex.size=(deg*2),
     vertex.label.cex=.6,vertex.label.color="black",vertex.color=color)
```



To transform the network into two one-mode networks, we first convert the two-mode igraph to a matrix. The first command makes sure that the matrix has labels once it is transformed

```
V(davis.ig)$name = V(davis.ig)$id
davis.mat <- get.incidence(davis.ig)
```

Multiply the matrices by their transpose

```
daviswomen.mat = davis.mat %*% t(davis.mat)
davisevents.mat = t(davis.mat) %*% davis.mat
```

Convert both to igraph objects

```
daviswomen.ig <- graph.adjacency(daviswomen.mat, mode=c("undirected"),weighted=TRUE)
davisevents.ig <- graph.adjacency(davisevents.mat, mode=c("undirected"),weighted=TRUE)
```

Remove loops and multiple edges (i.e., simplify) the networks

```
daviswomen.ig <- simplify(daviswomen.ig)
davisevents.ig <- simplify(davisevents.ig)
```

Check attributes and assign actor (vertex) names

```
V(daviswomen.ig)$name
```

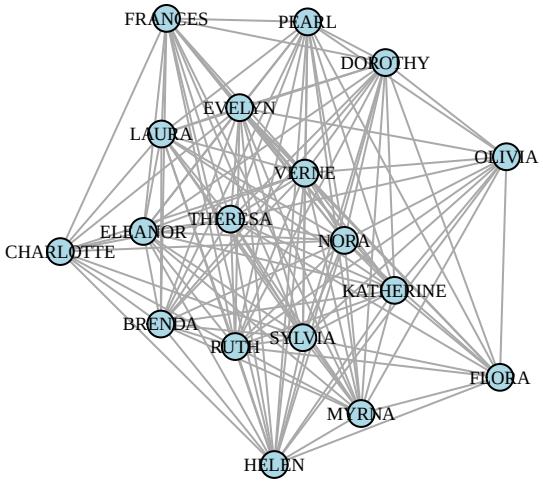
```
##  [1] "EVELYN"    "LAURA"     "THERESA"   "BRENDA"    "CHARLOTTE"
##  [6] "FRANCES"   "ELEANOR"   "PEARL"     "RUTH"      "VERNE"
## [11] "MYRNA"     "KATHERINE" "SYLVIA"    "NORA"      "HELEN"
## [16] "DOROTHY"   "OLIVIA"    "FLORA"
```

```
V(davisevents.ig)$name
```

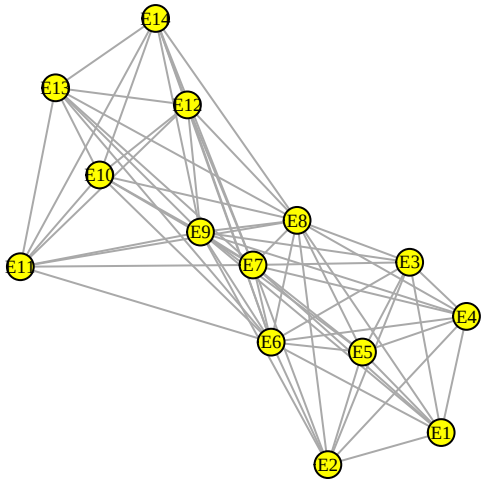
```
##  [1] "E1"  "E2"  "E3"  "E4"  "E5"  "E6"  "E7"  "E8"  "E9"  "E10" "E11"
## [12] "E12" "E13" "E14"
```

Plot the two new graphs

```
plot(daviswomen.ig, layout=layout.fruchterman.reingold,vertex.size=12,
     vertex.label.cex=.6,vertex.label.color="black",vertex.color="light blue")
```

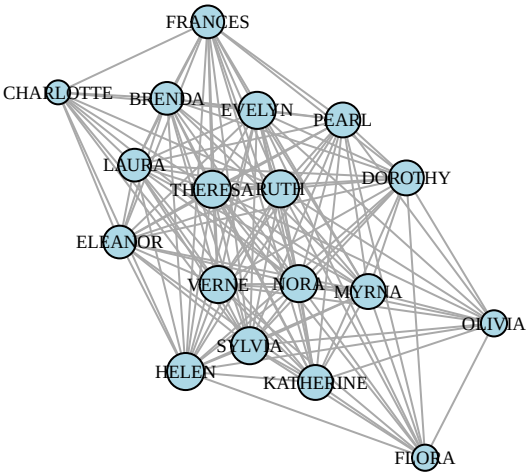



```
plot(davisevents.ig, layout=layout.fruchterman.reingold,vertex.size=12,
     vertex.label.cex=.6,vertex.label.color="black",vertex.color="yellow")
```

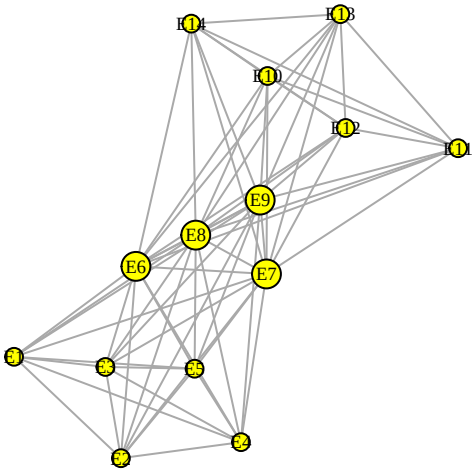


Get degree centrality and rolot the graphs where node size reflects degree

```
dwdeg <- degree(daviswomen.ig)
dedeg <- degree(davisevents.ig)

plot(daviswomen.ig,layout=layout.fruchterman.reingold,vertex.size=(dwdeg),
     vertex.label.cex=.6,vertex.label.color="black",vertex.color="light blue")
```




```
plot(davisevents.ig, layout=layout.fruchterman.reingold,vertex.size=(dedeg),
     vertex.label.cex=.6,vertex.label.color="black",vertex.color="yellow")
```



That's all for now