

SNA Basics I

One-Mode Networks in R

@ Sean F. Everton

It's generally a good idea to place a header at the top of your scripts that tell you what the script does, what its name is, etc.

```
#####  
# What: One-Mode Social Network Data in R  
# File: snab1.R  
# Created: 02.28.14  
# Revised: 05.05.18  
#####
```

Data

For this lab, we will use data collected by Stuart Koschade of the 17 individuals who participated in the first Bali bombing. Koschade (2006) recorded both the ties between the individuals, as well as the strength of the tie between them.

Setup

I like to clear the R workspace each time before beginning.

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

Typically, it's a good idea to 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 1")
```

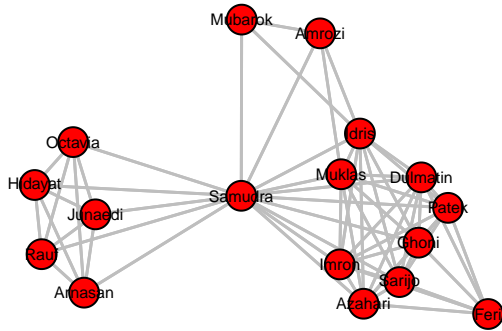
One-mode Social Network Data in *statnet*

Next, we need to load the libraries we plan to use. The *sna* and *network* libraries are part of the *statnet* package; *intergraph* allows us to convert a *network* object to an *igraph* object, which is an SNA library that we'll use later. However, because *igraph* and *sna* conflict with one another, we can't have them loaded at the same time.

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

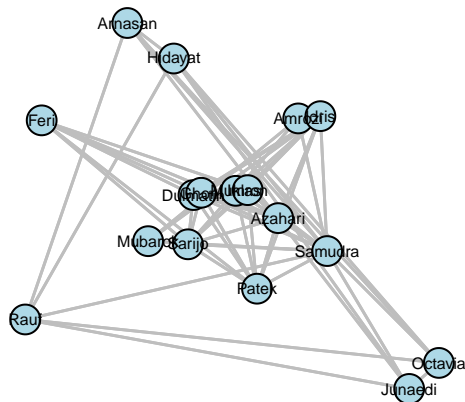
Import (read in) the Koschade data from the corresponding Pajek file. Notice that I saved the file with the *.net* extension. An extension isn't necessary, but I find it helpful to do this so that it's easy to know what type of object it is (in this case, a *network* object). Below, you will see that I save *igraph* objects with a *.ig* extension.

```
koschade.net <- as.network(read.paj("Koschade Bali.net"),directed=FALSE)
```

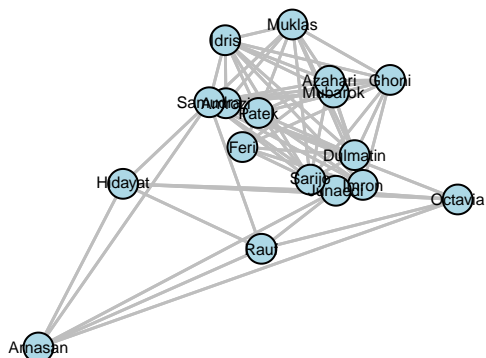



The remaining plots use different layouts available in the *sna* library.

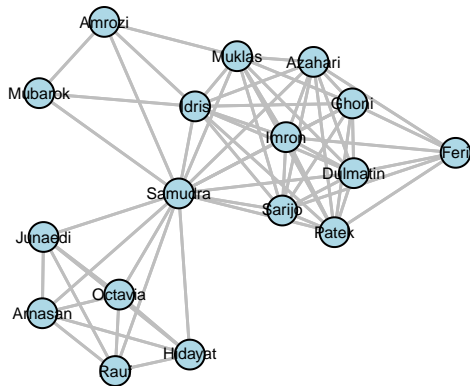
```
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net),
      label.col="black", label.cex=0.5, label.pos=5, mode="spring",
      edge.col="gray", usearrows=FALSE, vertex.col="light blue",
      vertex.cex=1.6)
```



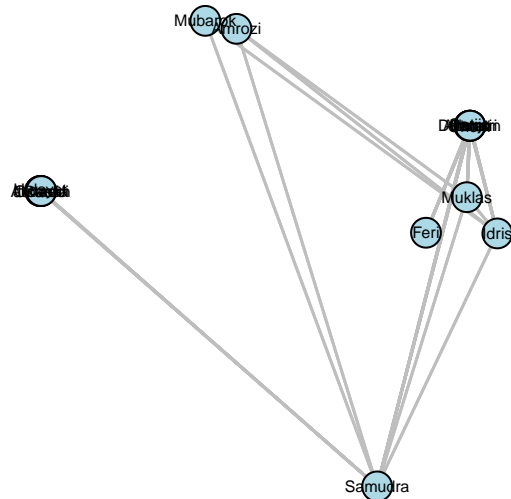
```
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net),
      label.col="black", label.cex=0.5, label.pos=5,
      mode="springrepulse", edge.col="gray",
      usearrows=FALSE, vertex.col="light blue", vertex.cex=1.6)
```



```
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net),
      label.col="black", label.cex=0.5, label.pos=5,
      mode="kamadakawai", edge.col="gray",
      usearrows=FALSE, vertex.col="light blue", vertex.cex=1.6)
```

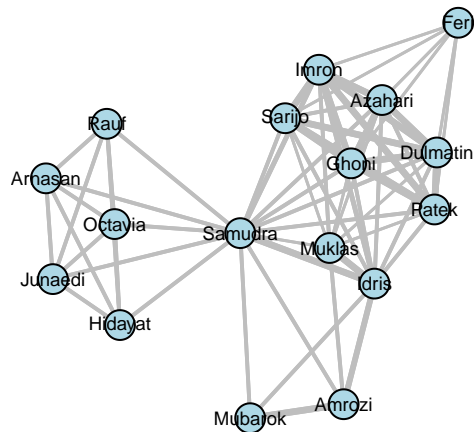


```
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net), label.col="black",
      label.cex=0.5, label.pos=5, mode="mds", edge.col="gray",
      usearrows=FALSE, vertex.col="light blue", vertex.cex=1.6)
```



Now, here's how we can graph the network with the width of lines reflecting strength of tie. This also shows how to save the coordinates so that the remaining plots will have the same layout.

```
coord <- gplot(koschade.net, gmode="graph", jitter=TRUE,
              label=network.vertex.names(koschade.net),
              edge.col="gray", label.col="black", edge.lwd=(koschade.mat),
              label.cex=0.6, label.pos=5, mode="kamadakawai",
              usearrows=FALSE, vertex.col="light blue",
              vertex.cex=1.6)
```



One way to save a plot is to use the *File>Save* command in the RStudio console or the Export feature in RStudio. An alternative way is to first you tell R what type of file to create (e.g., pdf, jpeg), then give the graphing command, and then turn the graphics device off with the *dev.off()* command. This last command is important because without it you'll get a partial plot or nothing at all.

As a pdf:

```
pdf(file = "koschade1.pdf")
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net), edge.col="gray",
      label.col="black", edge.lwd=(koschade.mat), label.cex=0.5,
      label.pos=5, usearrows=FALSE, vertex.col="light blue",
      coord=coord, vertex.cex=1.6)
dev.off()
```

As a jpeg:

```
jpeg(file = "koschade1.jpg")
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net), edge.col="gray",
      label.col="black", edge.lwd=(koschade.mat), label.cex=0.5,
      label.pos=5, usearrows=FALSE, vertex.col="light blue",
      coord=coord, vertex.cex=1.6)
dev.off()
```

As a tiff:

```
tiff(file = "koschade1.tif")
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net), edge.col="gray",
      label.col="black", edge.lwd=(koschade.mat), label.cex=0.5,
      label.pos=5, usearrows=FALSE, vertex.col="light blue",
      coord=coord, vertex.cex=1.6)
dev.off()
```

This alternative approach is preferable because it is easy to control the size and resolution of the resulting graph:

```
pdf(file = "koschade2.pdf", width = 8, height = 8)
gplot(koschade.net, gmode="graph", jitter=TRUE,
      label=network.vertex.names(koschade.net), edge.col="gray",
      label.col="black", edge.lwd=(koschade.mat), label.cex=0.5,
      label.pos=5, usearrows=FALSE, vertex.col="light blue",
```

```

    coord=coord,vertex.cex=1.6)
dev.off()

jpeg(file = "koschade2.jpg",width = 8,height = 8,units = 'in',res = 600)
gplot(koschade.net,gmode="graph",jitter=TRUE,
      label=network.vertex.names(koschade.net),edge.col="gray",
      label.col="black",edge.lwd=(koschade.mat),label.cex=0.5,
      label.pos=5,usearrows=FALSE,vertex.col="light blue",
      coord=coord,vertex.cex=1.6)
dev.off()

tiff(file = "koschade2.tif",width = 8,height = 8,units = 'in',res = 300)
gplot(koschade.net,gmode="graph",jitter=TRUE,
      label=network.vertex.names(koschade.net),edge.col="gray",
      label.col="black",edge.lwd=(koschade.mat),label.cex=0.5,
      label.pos=5,usearrows=FALSE,vertex.col="light blue",
      coord=coord,vertex.cex=1.6)
dev.off()

```

One-mode Social Network Data in *igraph*

Now let's see how to do this in "igraph." Like most SNA software, both *statnet* and *igraph* have their strengths and weaknesses. Thus, it's generally good to know both. We need to detach *sna* and load *igraph* because as was noted earlier, they sometimes conflict with one another when both are loaded.

```

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

```

Import data

One way to load the data in *igraph* is to use *intergraph* to convert it from a network object to an *igraph* object.

```

koschade.ig <- asIgraph(koschade.net)

```

View the data as a data frame. Note that *igraph* records the data as an edgelist although the edge weights disappear (all ties become 1 (Null//False)).

```

get.data.frame(koschade.ig)

```

```

##      from to X1    na
## 1      1  2  2 FALSE
## 2      1  3  2 FALSE
## 3      1  4  1 FALSE
## 4      1  5  1 FALSE
## 5      1  6  5 FALSE
## 6      1  8  1 FALSE
## 7      1  9  1 FALSE
## 8      1 15  1 FALSE
## 9      1 17  1 FALSE
## 10     2  4  2 FALSE
## 11     2  6  4 FALSE
## 12     2  7  5 FALSE
## 13     3  4  3 FALSE

```

```
## 14 3 5 5 FALSE
## 15 3 6 3 FALSE
## 16 3 8 5 FALSE
## 17 3 9 5 FALSE
## 18 3 15 5 FALSE
## 19 3 16 1 FALSE
## 20 3 17 5 FALSE
## 21 4 5 2 FALSE
## 22 4 6 5 FALSE
## 23 4 7 2 FALSE
## 24 4 8 2 FALSE
## 25 4 9 2 FALSE
## 26 4 10 2 FALSE
## 27 4 11 2 FALSE
## 28 4 12 2 FALSE
## 29 4 13 2 FALSE
## 30 4 14 2 FALSE
## 31 4 15 2 FALSE
## 32 4 17 2 FALSE
## 33 5 6 2 FALSE
## 34 5 8 5 FALSE
## 35 5 9 5 FALSE
## 36 5 15 5 FALSE
## 37 5 16 1 FALSE
## 38 5 17 5 FALSE
## 39 6 7 2 FALSE
## 40 6 8 2 FALSE
## 41 6 9 2 FALSE
## 42 6 15 2 FALSE
## 43 6 17 2 FALSE
## 44 8 9 5 FALSE
## 45 8 15 2 FALSE
## 46 8 16 1 FALSE
## 47 8 17 2 FALSE
## 48 9 15 5 FALSE
## 49 9 16 1 FALSE
## 50 9 17 5 FALSE
## 51 10 11 2 FALSE
## 52 10 12 2 FALSE
## 53 10 13 2 FALSE
## 54 10 14 2 FALSE
## 55 11 12 2 FALSE
## 56 11 13 2 FALSE
## 57 11 14 2 FALSE
## 58 12 13 2 FALSE
## 59 12 14 2 FALSE
## 60 13 14 2 FALSE
## 61 15 16 1 FALSE
## 62 15 17 5 FALSE
## 63 16 17 1 FALSE
```

Another way to bring the data into *igraph* is to import the data from the Pajek file.

```
koschade1.ig <- read.graph("Koschade Bali.net",format = c("pajek"))
```

View as data frame (note that igraph records the data as an edge list).

```
get.data.frame(koschade1.ig)
```

##	from	to	weight
## 1	Muklas	Amrozi	2
## 2	Muklas	Imron	2
## 3	Muklas	Samudra	1
## 4	Muklas	Dulmatin	1
## 5	Muklas	Idris	5
## 6	Muklas	Azahari	1
## 7	Muklas	Ghoni	1
## 8	Muklas	Patek	1
## 9	Muklas	Sarijo	1
## 10	Amrozi	Muklas	2
## 11	Amrozi	Samudra	2
## 12	Amrozi	Idris	4
## 13	Amrozi	Mubarok	5
## 14	Imron	Muklas	2
## 15	Imron	Samudra	3
## 16	Imron	Dulmatin	5
## 17	Imron	Idris	3
## 18	Imron	Azahari	5
## 19	Imron	Ghoni	5
## 20	Imron	Patek	5
## 21	Imron	Feri	1
## 22	Imron	Sarijo	5
## 23	Samudra	Muklas	1
## 24	Samudra	Amrozi	2
## 25	Samudra	Imron	3
## 26	Samudra	Dulmatin	2
## 27	Samudra	Idris	5
## 28	Samudra	Mubarok	2
## 29	Samudra	Azahari	2
## 30	Samudra	Ghoni	2
## 31	Samudra	Arnasan	2
## 32	Samudra	Rauf	2
## 33	Samudra	Octavia	2
## 34	Samudra	Hidayat	2
## 35	Samudra	Junaedi	2
## 36	Samudra	Patek	2
## 37	Samudra	Sarijo	2
## 38	Dulmatin	Muklas	1
## 39	Dulmatin	Imron	5
## 40	Dulmatin	Samudra	2
## 41	Dulmatin	Idris	2
## 42	Dulmatin	Azahari	5
## 43	Dulmatin	Ghoni	5
## 44	Dulmatin	Patek	5
## 45	Dulmatin	Feri	1
## 46	Dulmatin	Sarijo	5
## 47	Idris	Muklas	5

## 48	Idris	Amrozi	4
## 49	Idris	Imron	3
## 50	Idris	Samudra	5
## 51	Idris	Dulmatin	2
## 52	Idris	Mubarok	2
## 53	Idris	Azahari	2
## 54	Idris	Ghoni	2
## 55	Idris	Patek	2
## 56	Idris	Sarijo	2
## 57	Mubarok	Amrozi	5
## 58	Mubarok	Samudra	2
## 59	Mubarok	Idris	2
## 60	Azahari	Muklas	1
## 61	Azahari	Imron	5
## 62	Azahari	Samudra	2
## 63	Azahari	Dulmatin	5
## 64	Azahari	Idris	2
## 65	Azahari	Ghoni	5
## 66	Azahari	Patek	2
## 67	Azahari	Feri	1
## 68	Azahari	Sarijo	2
## 69	Ghoni	Muklas	1
## 70	Ghoni	Imron	5
## 71	Ghoni	Samudra	2
## 72	Ghoni	Dulmatin	5
## 73	Ghoni	Idris	2
## 74	Ghoni	Azahari	5
## 75	Ghoni	Patek	5
## 76	Ghoni	Feri	1
## 77	Ghoni	Sarijo	5
## 78	Arnasan	Samudra	2
## 79	Arnasan	Rauf	2
## 80	Arnasan	Octavia	2
## 81	Arnasan	Hidayat	2
## 82	Arnasan	Junaedi	2
## 83	Rauf	Samudra	2
## 84	Rauf	Arnasan	2
## 85	Rauf	Octavia	2
## 86	Rauf	Hidayat	2
## 87	Rauf	Junaedi	2
## 88	Octavia	Samudra	2
## 89	Octavia	Arnasan	2
## 90	Octavia	Rauf	2
## 91	Octavia	Hidayat	2
## 92	Octavia	Junaedi	2
## 93	Hidayat	Samudra	2
## 94	Hidayat	Arnasan	2
## 95	Hidayat	Rauf	2
## 96	Hidayat	Octavia	2
## 97	Hidayat	Junaedi	2
## 98	Junaedi	Samudra	2
## 99	Junaedi	Arnasan	2
## 100	Junaedi	Rauf	2
## 101	Junaedi	Octavia	2

```

## 102 Junaedi Hidayat 2
## 103 Patek Muklas 1
## 104 Patek Imron 5
## 105 Patek Samudra 2
## 106 Patek Dulmatin 5
## 107 Patek Idris 2
## 108 Patek Azahari 2
## 109 Patek Ghoni 5
## 110 Patek Feri 1
## 111 Patek Sarijo 5
## 112 Feri Imron 1
## 113 Feri Dulmatin 1
## 114 Feri Azahari 1
## 115 Feri Ghoni 1
## 116 Feri Patek 1
## 117 Feri Sarijo 1
## 118 Sarijo Muklas 1
## 119 Sarijo Imron 5
## 120 Sarijo Samudra 2
## 121 Sarijo Dulmatin 5
## 122 Sarijo Idris 2
## 123 Sarijo Azahari 2
## 124 Sarijo Ghoni 5
## 125 Sarijo Patek 5
## 126 Sarijo Feri 1

```

This command brings it in as a directed (assymmetric) network. We need to convert it from directed graph to undirected graph.

```

koschade1.ig <- as.undirected(koschade1.ig,edge.attr.comb=list(weight="mean"))

```

View as data frame: Note that the order of nodes may change (but all should be fine).

```

koschade1.edge <- get.data.frame(koschade1.ig)
koschade1.edge

```

```

##      from      to weight
## 1  Muklas  Amrozi     2
## 2  Muklas   Imron     2
## 3  Muklas Samudra     1
## 4  Amrozi Samudra     2
## 5   Imron Samudra     3
## 6  Muklas Dulmatin     1
## 7   Imron Dulmatin     5
## 8 Samudra Dulmatin     2
## 9  Muklas   Idris     5
## 10 Amrozi   Idris     4
## 11  Imron   Idris     3
## 12 Samudra   Idris     5
## 13 Dulmatin   Idris     2
## 14 Amrozi Mubarok     5
## 15 Samudra Mubarok     2
## 16   Idris Mubarok     2
## 17  Muklas Azahari     1
## 18  Imron Azahari     5
## 19 Samudra Azahari     2

```

```

## 20 Dulmatin Azahari 5
## 21 Idris Azahari 2
## 22 Muklas Ghoni 1
## 23 Imron Ghoni 5
## 24 Samudra Ghoni 2
## 25 Dulmatin Ghoni 5
## 26 Idris Ghoni 2
## 27 Azahari Ghoni 5
## 28 Samudra Arnasan 2
## 29 Samudra Rauf 2
## 30 Arnasan Rauf 2
## 31 Samudra Octavia 2
## 32 Arnasan Octavia 2
## 33 Rauf Octavia 2
## 34 Samudra Hidayat 2
## 35 Arnasan Hidayat 2
## 36 Rauf Hidayat 2
## 37 Octavia Hidayat 2
## 38 Samudra Junaedi 2
## 39 Arnasan Junaedi 2
## 40 Rauf Junaedi 2
## 41 Octavia Junaedi 2
## 42 Hidayat Junaedi 2
## 43 Muklas Patek 1
## 44 Imron Patek 5
## 45 Samudra Patek 2
## 46 Dulmatin Patek 5
## 47 Idris Patek 2
## 48 Azahari Patek 2
## 49 Ghoni Patek 5
## 50 Imron Feri 1
## 51 Dulmatin Feri 1
## 52 Azahari Feri 1
## 53 Ghoni Feri 1
## 54 Patek Feri 1
## 55 Muklas Sarijo 1
## 56 Imron Sarijo 5
## 57 Samudra Sarijo 2
## 58 Dulmatin Sarijo 5
## 59 Idris Sarijo 2
## 60 Azahari Sarijo 2
## 61 Ghoni Sarijo 5
## 62 Patek Sarijo 5
## 63 Feri Sarijo 1

```

Now, let's create a vector of edge weights.

```
edgeweight1 <- get.edge.attribute(koschade1.ig,"weight")
```

A final way is to read network in from the matrix that we used earlier.

```
koschade2.ig <- graph.adjacency(koschade.mat,mode="undirected",weighted=TRUE)
```

View as data frame: note that the order of nodes does not change.

```
koscahde2.edge <- get.data.frame(koschade2.ig)
koscahde2.edge
```

```
##      from      to weight
## 1  Muklas  Amrozi      2
## 2  Muklas  Imron      2
## 3  Muklas  Samudra     1
## 4  Muklas  Dulmatin     1
## 5  Muklas  Idris       5
## 6  Muklas  Azahari     1
## 7  Muklas  Ghoni       1
## 8  Muklas  Patek       1
## 9  Muklas  Sarijo      1
## 10 Amrozi  Samudra     2
## 11 Amrozi  Idris       4
## 12 Amrozi  Mubarak     5
## 13 Imron   Samudra     3
## 14 Imron   Dulmatin     5
## 15 Imron   Idris       3
## 16 Imron   Azahari     5
## 17 Imron   Ghoni       5
## 18 Imron   Patek       5
## 19 Imron   Feri        1
## 20 Imron   Sarijo      5
## 21 Samudra Dulmatin     2
## 22 Samudra Idris       5
## 23 Samudra Mubarak     2
## 24 Samudra Azahari     2
## 25 Samudra Ghoni       2
## 26 Samudra Arnasan     2
## 27 Samudra Rauf        2
## 28 Samudra Octavia     2
## 29 Samudra Hidayat     2
## 30 Samudra Junaedi     2
## 31 Samudra Patek       2
## 32 Samudra Sarijo      2
## 33 Dulmatin Idris       2
## 34 Dulmatin Azahari     5
## 35 Dulmatin Ghoni       5
## 36 Dulmatin Patek       5
## 37 Dulmatin Feri        1
## 38 Dulmatin Sarijo      5
## 39 Idris   Mubarak     2
## 40 Idris   Azahari     2
## 41 Idris   Ghoni       2
## 42 Idris   Patek       2
## 43 Idris   Sarijo      2
## 44 Azahari Ghoni       5
## 45 Azahari Patek       2
## 46 Azahari Feri        1
## 47 Azahari Sarijo      2
## 48 Ghoni   Patek       5
## 49 Ghoni   Feri        1
## 50 Ghoni   Sarijo      5
```

```
## 51 Arnasan Rauf 2
## 52 Arnasan Octavia 2
## 53 Arnasan Hidayat 2
## 54 Arnasan Junaedi 2
## 55 Rauf Octavia 2
## 56 Rauf Hidayat 2
## 57 Rauf Junaedi 2
## 58 Octavia Hidayat 2
## 59 Octavia Junaedi 2
## 60 Hidayat Junaedi 2
## 61 Patek Feri 1
## 62 Patek Sarijo 5
## 63 Feri Sarijo 1
```

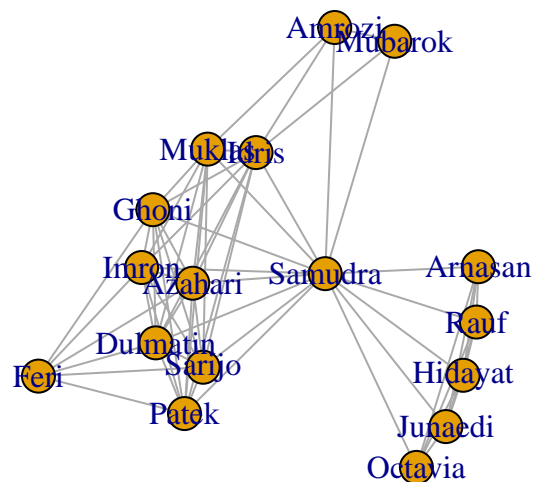
Once again, create a vector of edge weights.

```
edgweight2 <- get.edge.attribute(koschade2.ig, "weight")
```

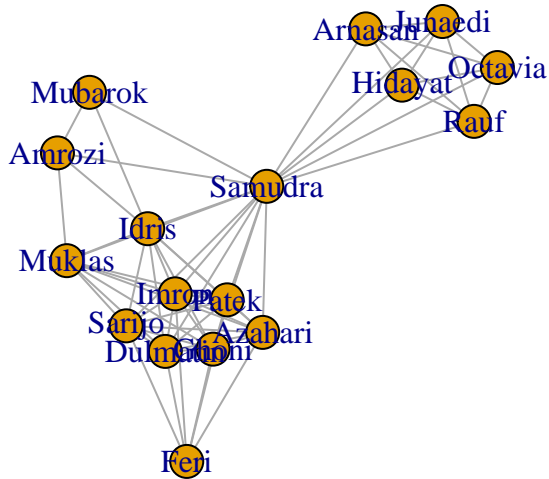
Graphing in igraph

Here's a simple plot using *igraph*; note that the vertex ids are numbers (compare)

```
plot(koschade1.ig)
```

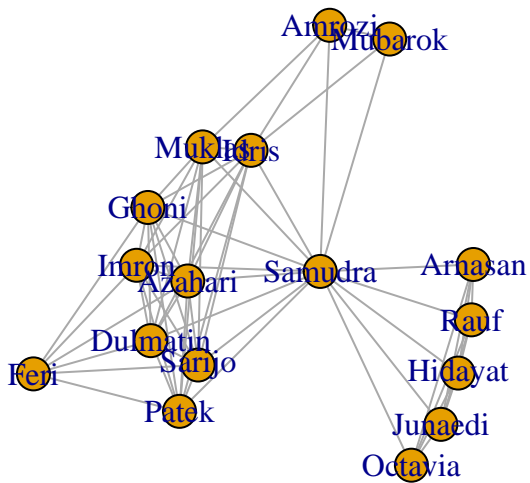


```
plot(koschade2.ig)
```

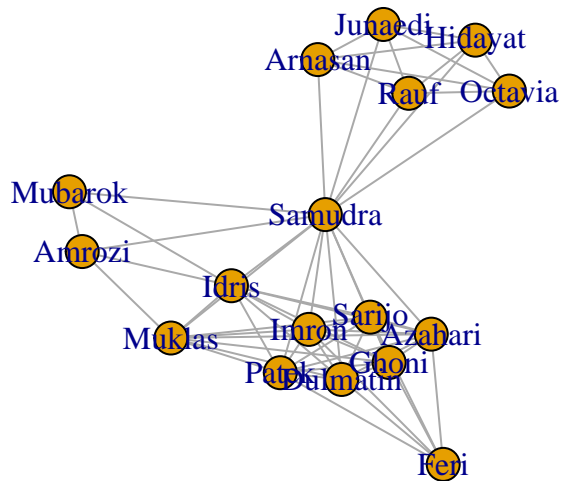


Now, ell igraph that it should use the vertex ids as labels for koschade1.igraph (don't need it for koschade2.ig), and then plot again with labels.

```
V(koschade1.ig)$label = V(koschade1.ig)$id
plot(koschade1.ig)
```

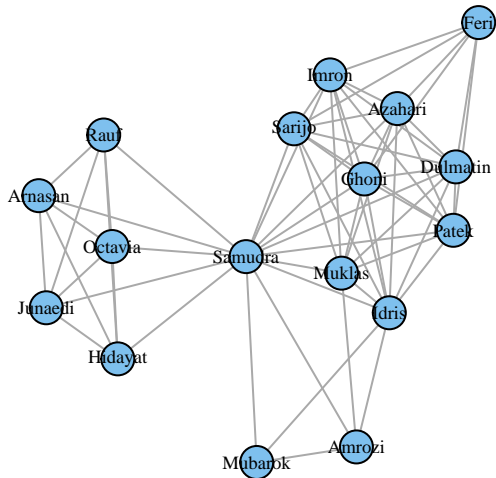


```
plot(koschade2.ig)
```

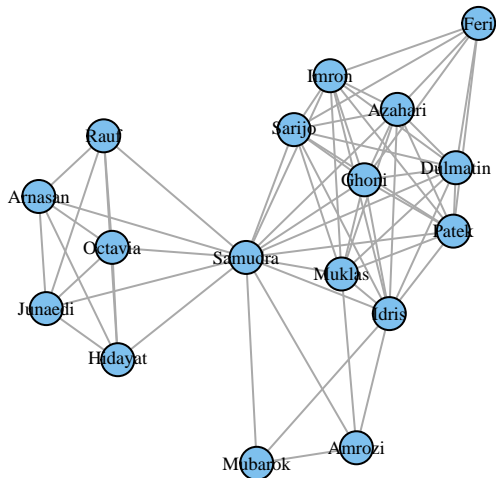


Let's try making some more sophisticated plots

```
plot(koschade1.ig,layout=coord,vertex.color="Skyblue2",  
     vertex.label.cex=.6,vertex.label.color="black")
```

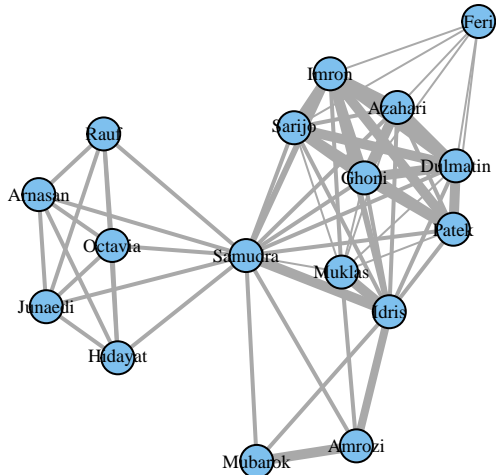


```
plot(koschade2.ig,layout=coord,vertex.color="Skyblue2",  
     vertex.label.cex=.6,vertex.label.color="black")
```

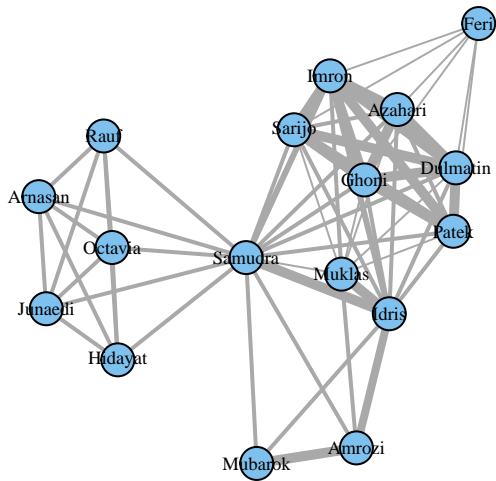


Size the edges by tie strength and plot again

```
plot(koschade1.ig,layout=coord,vertex.color="Skyblue2",  
     vertex.label.cex=.6,vertex.label.color="black",edge.width=edgeweight1)
```



```
plot(koschade2.ig,layout=coord,vertex.color="Skyblue2",
      vertex.label.cex=.6,vertex.label.color="black",edge.width=edgeweight2)
```



Save final plot in various formats

```
pdf(file = "koschade3.pdf",width = 8,height = 8)
plot(koschade2.ig,layout=coord,vertex.color="Skyblue2",
      vertex.label.cex=.6,vertex.label.color="black",edge.width=edgeweight2)
dev.off()

jpeg(file = "koschade3.jpg",width = 8,height = 8,units = 'in',res = 600)
plot(koschade2.ig,layout=coord,vertex.color="Skyblue2",
      vertex.label.cex=.6,vertex.label.color="black",edge.width=edgeweight2)
dev.off()

tiff(file = "koschade3.tif",width = 8,height = 8,units = 'in',res = 300)
plot(koschade2.ig,layout=coord,vertex.color="Skyblue2",
      vertex.label.cex=.6,vertex.label.color="black",edge.width=edgeweight2)
dev.off()
```

That's all for now