SNA Basics VII

Centrality and Power

@ Sean F. Everton

Place a header at the top of your script that tells you what you called it, what it accomplishes, etc.

```
##################################################
# What: Centrality and Power in R
# File: snab7.R
# Created: 02.28.29
# Revised: 07.02.18
##################################################
```

Data

Centrality is one of SNA's oldest concepts. A central actor can be someone who has numerous ties to other actors (degree), someone who is closer (in terms of path distance) to all other actors (closeness), someone who lies on the shortest path (geodesic) between any two actors (betweenness), or someone who has ties to other highly central actors (eigenvector). In some networks, the same actors will score high on all four measures. In others, they won't. There are, of course, more than four measures of centrality. The current version of UCINET includes more than 20, and it doesn't include them all. Pajek includes several as well.

For this exercise, we'll once again use the Anabaptist Leadership network. If you recall, the dataset includes 67 actors, 55 who were sixteenth century Anabaptist leaders and 12 who were prominent Protestant Reformation leaders who had contact with and influenced some of the Anabaptist leaders included in this dataset. These network data build upon a smaller dataset (Matthews et al., 2013) that did not include some leading Anabaptist leaders, such as Menno Simons, who is generally seen as the "founder" of the Amish and Mennonites.

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.

```
setwd("~/Dropbox/Networks and Religion (Book)/Website/Labs/SNA Basics 7")
```

Centrality and Power in *statnet*

We need to first load the libraries we plan to use. The *sna* and *network* libraries are part of the *statnet* package. Because *igraph* and *sna* conflict with one another, we can't load them at the same time. Well, we can, but it only causes unnecessary trouble. There is a workaround, but it's just as easy to attach and detach the two libraries.

```
library(sna)
library(network)
```

Import the Anabaptist Leadership network. Here, we'll read in the csv file, but don't forget that we can read in Pajek and other types of files.

```r
anabaptist.mat <- as.matrix(read.csv(("Anabaptist Leaders.csv"),header=TRUE,
                            row.names=1,check.names=FALSE))
anabaptist.net <- as.network(anabaptist.mat,directed=FALSE)
anabaptist.net
##  Network attributes:
##   vertices = 67
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 183
##     missing edges= 0
##     non-missing edges= 183
##
##  Vertex attribute names:
##     vertex.names
##
## No edge attributes
```

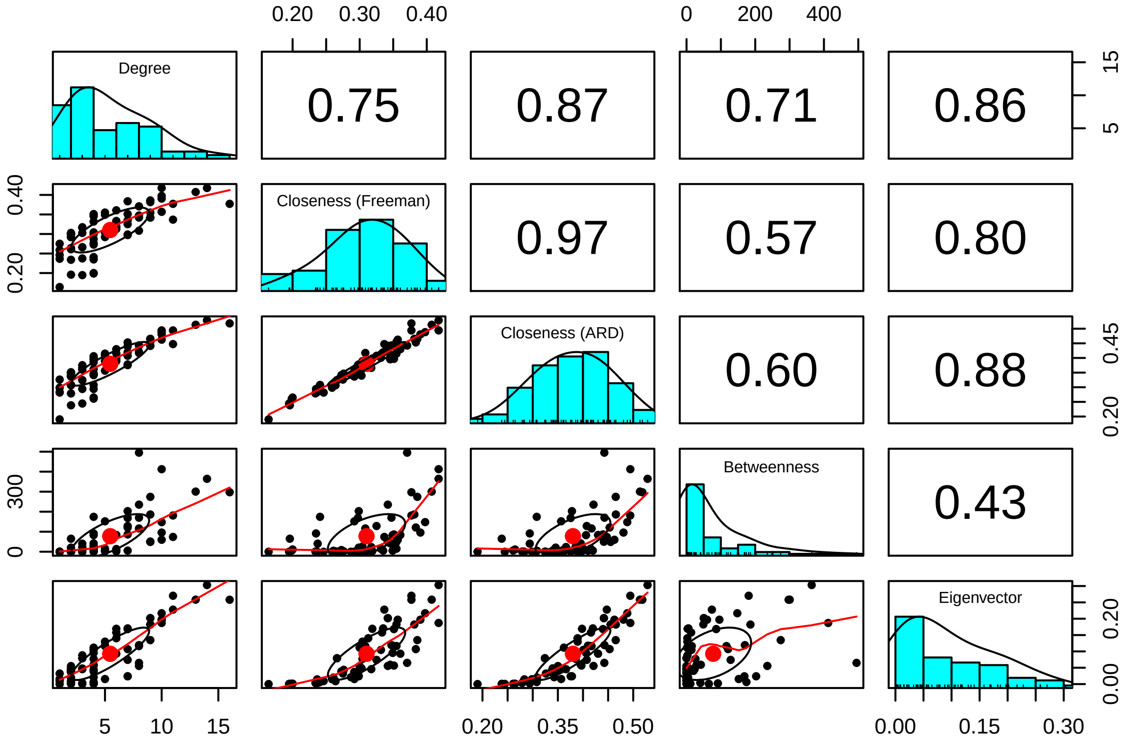
Calculate the four primary measures of centrality. Note that there are two closeness commands. The first is the standard measure of closeness (Freeman). Unfortunately, it doesn't hand infinite distances, so we show how to calculate an alternative (ARD) that does. The final command includes the five sets of measures into a data frame that we can use to estimate the correlation between the various measures

```r
anabaptist.deg <- degree(anabaptist.net,gmode="graph",ignore.eval=FALSE)
anabaptist.clo <- closeness(anabaptist.net,gmode="graph")
anabaptist.ard <- closeness(anabaptist.net,gmode="graph",cmode="suminvundir")
anabaptist.bet <- betweenness(anabaptist.net,gmode="graph")
anabaptist.eig <- evcent(anabaptist.net,use.eigen=TRUE)

names <- network.vertex.names(anabaptist.net)
centrality <- data.frame(names,anabaptist.deg,anabaptist.clo,anabaptist.ard,anabaptist.bet,
                            anabaptist.eig,row.names = TRUE)
colnames(centrality) <- c("Degree","Closeness (Freeman)","Closeness (ARD)",
                          "Betweenness","Eigenvector")
cor(centrality)
##                        Degree Closeness (Freeman) Closeness (ARD)
## Degree              1.0000000           0.7529908       0.8694941
## Closeness (Freeman) 0.7529908           1.0000000       0.9709355
## Closeness (ARD)     0.8694941           0.9709355       1.0000000
## Betweenness         0.7091429           0.5698427       0.6032354
## Eigenvector         0.8601345           0.8014529       0.8790569
##                     Betweenness Eigenvector
## Degree                0.7091429   0.8601345
## Closeness (Freeman)   0.5698427   0.8014529
## Closeness (ARD)       0.6032354   0.8790569
## Betweenness           1.0000000   0.4274049
## Eigenvector           0.4274049   1.0000000
```

Note that, for the most part, the centrality measures correlate very highly with degree, less so with the other measures. The two closeness measures correlate very high with each other, which is a good sign that they're tapping into the same phenomenon. Betweenness seems to be the most "independent" of the measures, as it tends to correlate with the other measures at lower levels.

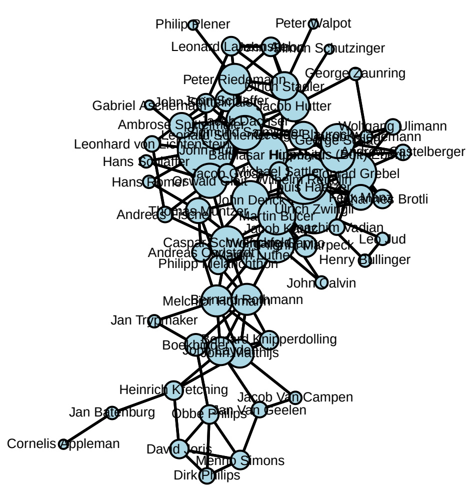
Here's a really nice correlation function associated with the *psych* library

```r
library(psych)
pairs.panels(centrality)
```



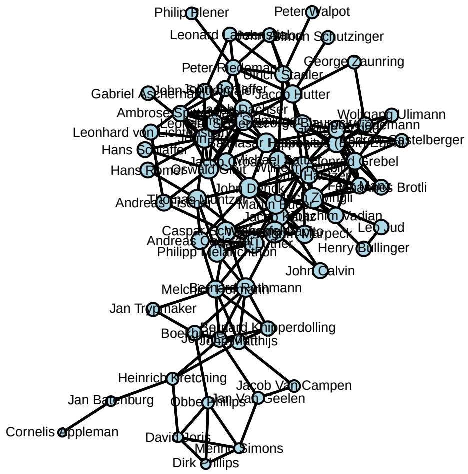
Let's plot the network where we vary node size by the centrality measures; note that I've rescaled them so that the nodes don't get overwhelmingly big or way too small. Also, so that some nodes don't disappear, I have added a small amount to the centrality score. For example, with degree centrality, I've divided the scores by 4 and then added a small amount (.5) to them.

```r
# Degree
coords <- gplot(anabaptist.net,label=network.vertex.names(anabaptist.net),label.pos = 5,
                vertex.col="Light Blue",label.col="black",label.cex=0.4,
                vertex.cex=((anabaptist.deg/4)+.5),usearrows=FALSE,mode="kamadakawai")
```

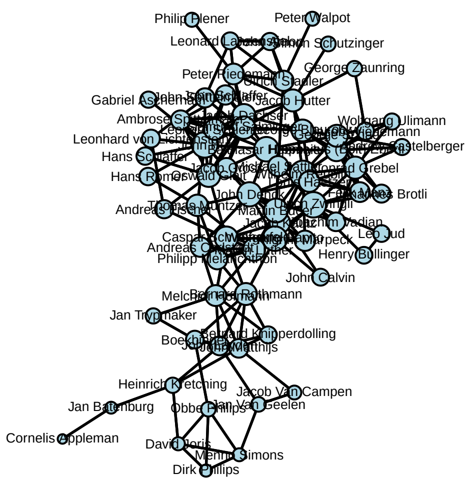



```r
# Closeness (Freeman)
```

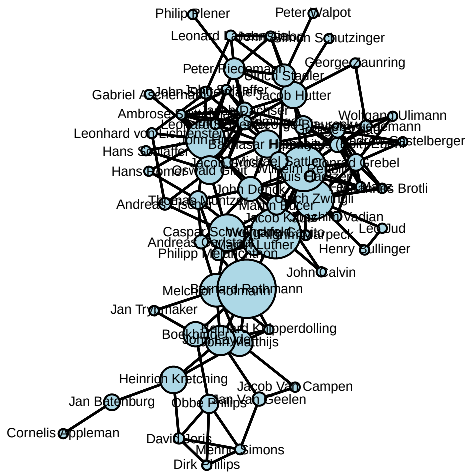
```r
gplot(anabaptist.net,label=network.vertex.names(anabaptist.net),label.pos = 5,
                vertex.col="Light Blue",label.col="black",label.cex=0.4,
                vertex.cex=anabaptist.clo*4,usearrows=FALSE,coord=coords)
```



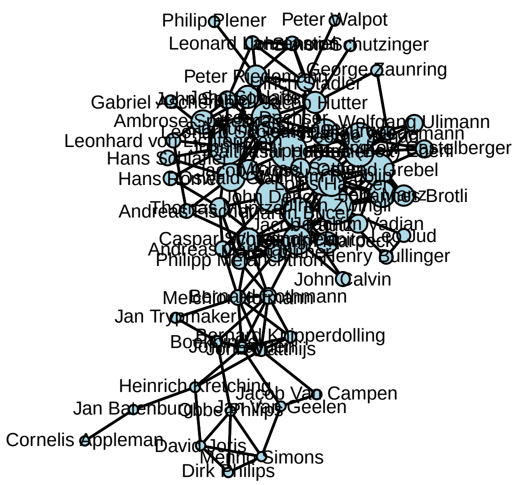

```r
# Closeness (ARD)
gplot(anabaptist.net,label=network.vertex.names(anabaptist.net),label.pos = 5,
                vertex.col="Light Blue",label.col="black",label.cex=0.4,
                vertex.cex=anabaptist.ard*4,usearrows=FALSE,coord=coords)
```




```r
# Betweenness
gplot(anabaptist.net,label=network.vertex.names(anabaptist.net),label.pos = 5,
                vertex.col="Light Blue",label.col="black",label.cex=0.4,
                vertex.cex=((anabaptist.bet/125)+.75),usearrows=FALSE,coord=coords)
```




```r
# Eigenvector
gplot(anabaptist.net,label=network.vertex.names(anabaptist.net),label.pos = 5,
                 vertex.col="Light Blue",label.col="black",label.cex=0.6,
                 vertex.cex=((anabaptist.eig*8)+.75),usearrows=FALSE,coord=coords)
```



Centrality and Power in *igraph*

Let's move to *igraph* and see how to do things there. We need to load *igraph* and detach *sna*

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

Read in the data in from the csv file

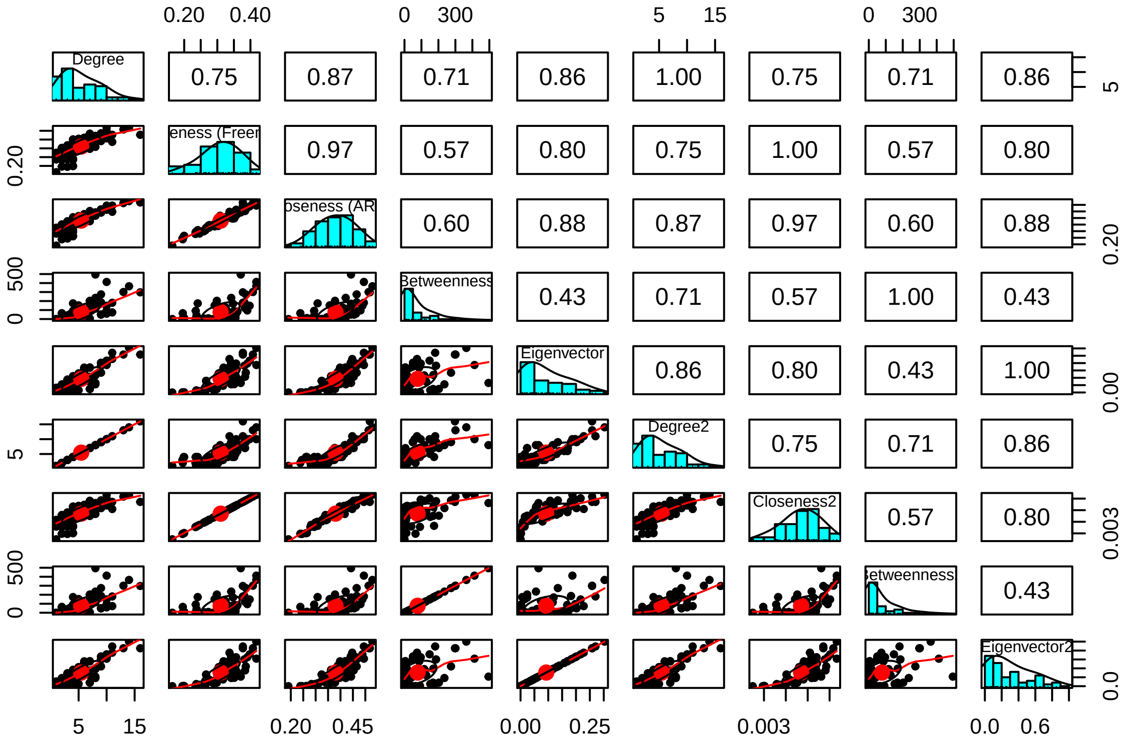
```r
anabaptist.dat <-read.csv("Anabaptist Leaders.csv",header=TRUE,row.names=1,check.names=FALSE)
anabaptist.mat <- as.matrix(anabaptist.dat)
anabaptist.ig <- graph.adjacency(anabaptist.mat,mode="undirected",weighted=NULL)
anabaptist.ig
## IGRAPH 837ef44 UN-- 67 183 --
## + attr: name (v/c)
## + edges from 837ef44 (vertex names):
##  [1] Martin Luther --Ulrich Zwingli
##  [2] Martin Luther --Thomas Muntzer
```

```
##  [3] Martin Luther --Andreas Carlstadt
##  [4] Martin Luther --Caspar Schwenckfeld
##  [5] Martin Luther --Melchior Hofmann
##  [6] Martin Luther --Philipp Melanchthon
##  [7] Martin Luther --Martin Bucer
##  [8] John Calvin   --Wolfgang Capito
## + ... omitted several edges
```

Estimate the four centrality scores and compare them to the *statnet* calculations. Note that the eigenvector function creates more than one object, so we have to extract the vector including the scores in a second step. The centrality scores correlate 100% with there *statnet* counterparts (as they should)

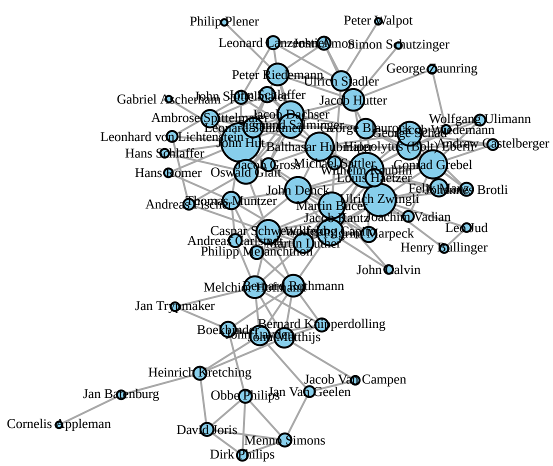
```
# Calculate centrality
anabaptist.deg2 <- degree(anabaptist.ig)
anabaptist.clo2 <- closeness(anabaptist.ig)
anabaptist.bet2 <- betweenness(anabaptist.ig)
anabaptist.eigen <- evcent(anabaptist.ig)
anabaptist.eig2 <- anabaptist.eigen$vector

# Add scores to dataframe
centrality$Degree2 <- anabaptist.deg2
centrality$Closeness2 <- anabaptist.clo2
centrality$Betweenness2 <- anabaptist.bet2
centrality$Eigenvector2 <- anabaptist.eig2

# Correlations
cor(centrality)
##                        Degree Closeness (Freeman) Closeness (ARD)
## Degree              1.0000000           0.7529908       0.8694941
## Closeness (Freeman) 0.7529908           1.0000000       0.9709355
## Closeness (ARD)     0.8694941           0.9709355       1.0000000
## Betweenness         0.7091429           0.5698427       0.6032354
## Eigenvector         0.8601345           0.8014529       0.8790569
## Degree2             1.0000000           0.7529908       0.8694941
## Closeness2          0.7529908           1.0000000       0.9709355
## Betweenness2        0.7091429           0.5698427       0.6032354
## Eigenvector2        0.8601345           0.8014529       0.8790569
##                     Betweenness Eigenvector   Degree2 Closeness2
## Degree                0.7091429   0.8601345 1.0000000  0.7529908
## Closeness (Freeman)   0.5698427   0.8014529 0.7529908  1.0000000
## Closeness (ARD)       0.6032354   0.8790569 0.8694941  0.9709355
## Betweenness           1.0000000   0.4274049 0.7091429  0.5698427
## Eigenvector           0.4274049   1.0000000 0.8601345  0.8014529
## Degree2               0.7091429   0.8601345 1.0000000  0.7529908
## Closeness2            0.5698427   0.8014529 0.7529908  1.0000000
## Betweenness2          1.0000000   0.4274049 0.7091429  0.5698427
## Eigenvector2          0.4274049   1.0000000 0.8601345  0.8014529
##                     Betweenness2 Eigenvector2
## Degree                 0.7091429    0.8601345
## Closeness (Freeman)    0.5698427    0.8014529
## Closeness (ARD)        0.6032354    0.8790569
## Betweenness            1.0000000    0.4274049
## Eigenvector            0.4274049    1.0000000
## Degree2                0.7091429    0.8601345
```

```
## Closeness2             0.5698427    0.8014529
## Betweenness2           1.0000000    0.4274049
## Eigenvector2           0.4274049    1.0000000
pairs.panels(centrality)
```

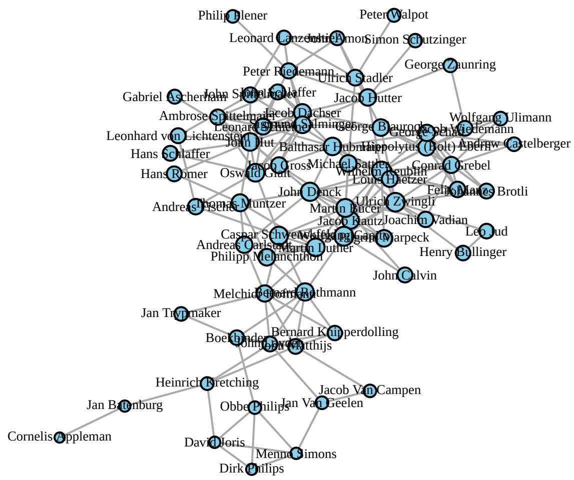


And a few plots; we'll use the coordinates from before

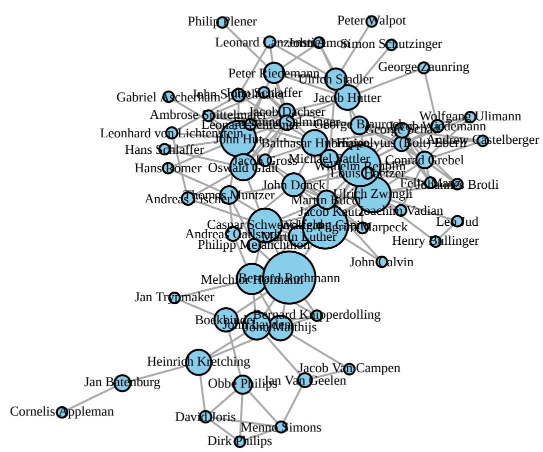
```
plot(anabaptist.ig,layout=coords,vertex.size=((anabaptist.deg2)+2),vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue")
```



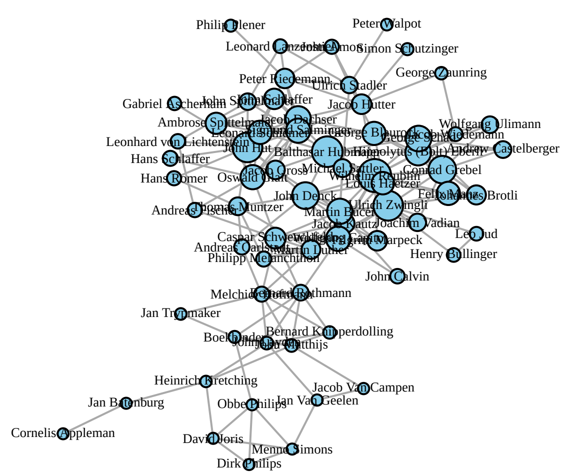

```
plot(anabaptist.ig,layout=coords,,vertex.size=((anabaptist.clo2*1000)+2),vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue")
```




```r
plot(anabaptist.ig,layout=coords,vertex.size=((anabaptist.bet2/25)+5),vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue")
```




```r
plot(anabaptist.ig,layout=coords,vertex.size=((anabaptist.eig2*10)+5),vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue")
```



That's all for now