

# Analyzing Social Networks with R

**Michael D. Siciliano**  
**Department of Public Administration**  
**University of Illinois at Chicago**

**Chicago R User Group – October 3<sup>rd</sup>, 2012**

# Overview

- Software
- Three “Types” of network analysis and an example of each
  - Descriptive analysis and visualization
    - Data - network of IV drug users
  - Antecedent analysis
    - Data - organizational collaboration after an earthquake
  - Consequence analysis
    - Data - teacher networks in 21 schools
- Available online resources for more information

# Network Analysis Software

## Packages in R

- statnet
  - sna
  - network
  - ergm
- igraph

## Other programs

- UCINET
- ORA
- Pajek
- NodeXL
- libSNA
- C-IKNOW
- many more...

## Visualization tools

- Netdraw
- Visone
- Tulip
- Gephi

Real advantage with R is that network analysis, statistical analysis, and visualization are integrated into one program.

# Types of network analysis

- Descriptive
  - Visualization of the network along with summary statistics at the individual, dyadic, and whole network levels.
- Antecedent
  - Attempts to understand the micro-processes at work in forming the observed network.
- Consequence
  - Measures the implication of different network positions or structures.

# Descriptive Analysis

- For any system under study we may have two types of data: behavioral/relational and attribute.
- We can also construct nodal attributes based on the relational data, for example centrality measures.
- Two simple measures of centrality are:
  - Degree – the total number of connections.
  - Betweenness - the number of shortest paths from one node to another that pass through a given node.

Goal: identify key members in a needle sharing population to diffuse safe practices

Data = social network of IV drug users in Hartford, CT

[illegible]

# Load data as a network object

- Relational data = 293 actors and 337 ties.

```
library("statnet")  
drug=read.csv(file="drugnet.csv", header=TRUE, row.names=1)  
drug=as.matrix(drug)#to make certain the data is a matrix  
drugn=as.network(drug, matrix.type="adjacency", directed=TRUE) #create a  
network object
```

- Attribute data = gender and ethnicity; plus we will calculate some centrality measures

```
#read in the attribute data  
drugatt=read.csv(file="drugnet_att.csv", header=TRUE, row.names=1)  
drugatt=as.data.frame(drugatt)  
#calculate some additional attributes  
deg1=degree(drugn)  
bet1=betweenness(drugn)  
drugatt$bet=bet1#add the measures to our existing attribute data  
drugatt$deg=deg1
```

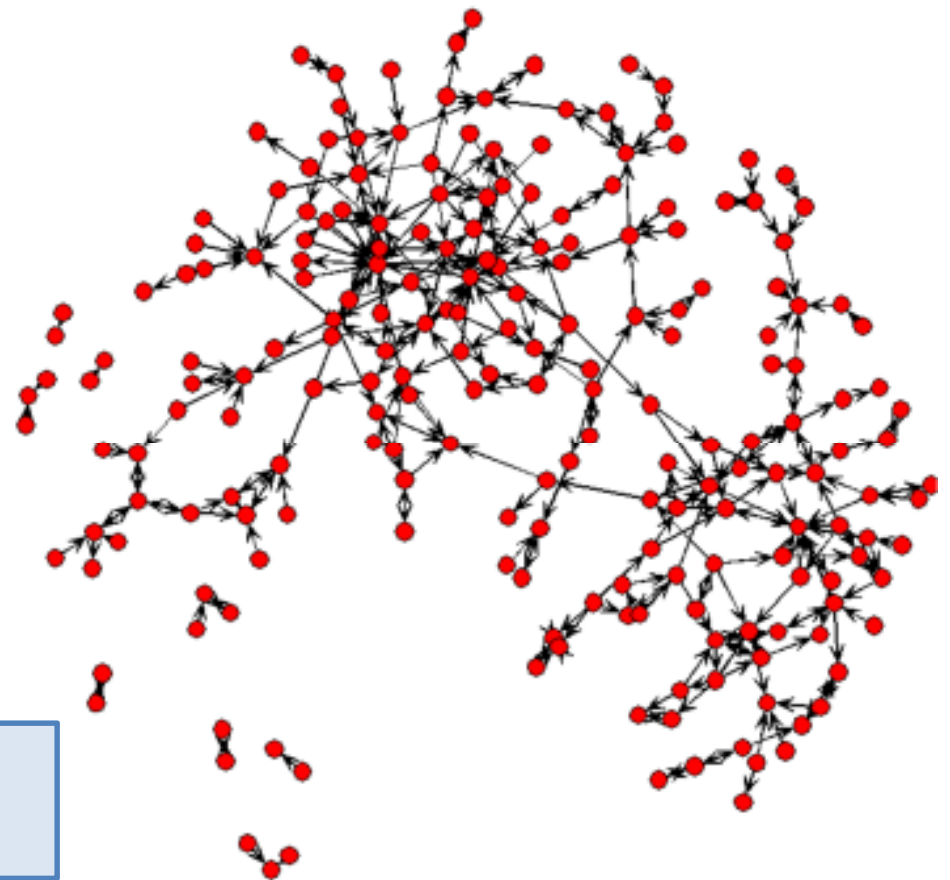
# IV Drug Users in Hartford, CT

## Attribute Data

```
head(drugatt)
```

	Ethnicity	Gender	HasTie	bet	deg
1	1	1	1	4	5
2	1	1	1	6	6
3	1	1	1	14	4
4	1	2	1	19	5
5	3	1	1	0	1
6	3	1	1	2	2

## Relational Data



```
#plot the IV users Network
```

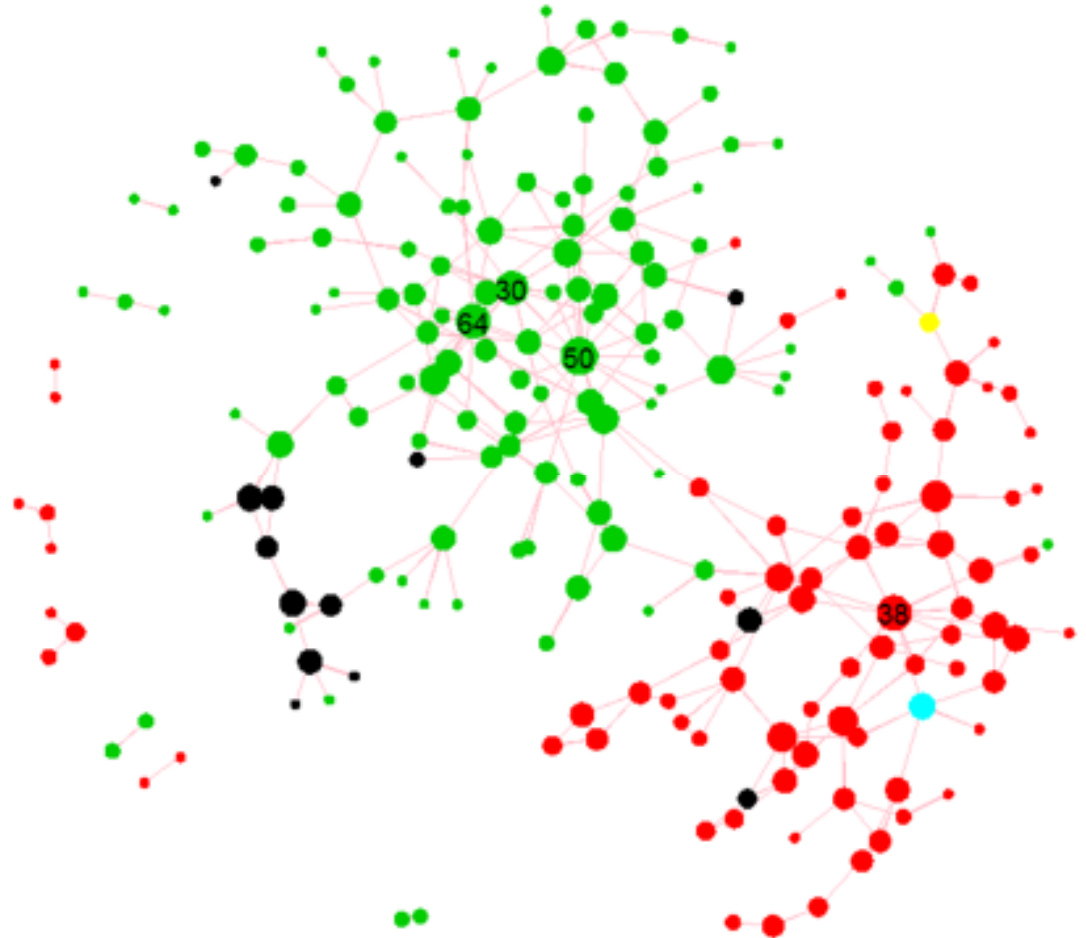
```
plot.network(drugn, mode = "fruchtermanreingold",  
displayisolates = FALSE)
```



# Scale nodes based on degree centrality

```
#Define a label set that is dependent on
degree centrality
label1=vector()
for (i in 1:length(deg1))
  label1[i]=ifelse
(deg1[i]>=10,row.names(drugatt[i,]),NA)

#Plot as network object, using high degree
nodes as labels
plot.network(drugn, mode =
"fruchtermanreingold",
vertex.col=drugatt$Ethnicity, label=label1,
displayisolates=FALSE,
label.cex=1.1, usearrows=FALSE,
edge.col="pink",
vertex.cex=log10(deg1+1), label.pos=5,
vertex.border=drugatt$Ethnicity,
edge.lwd=.25)
```



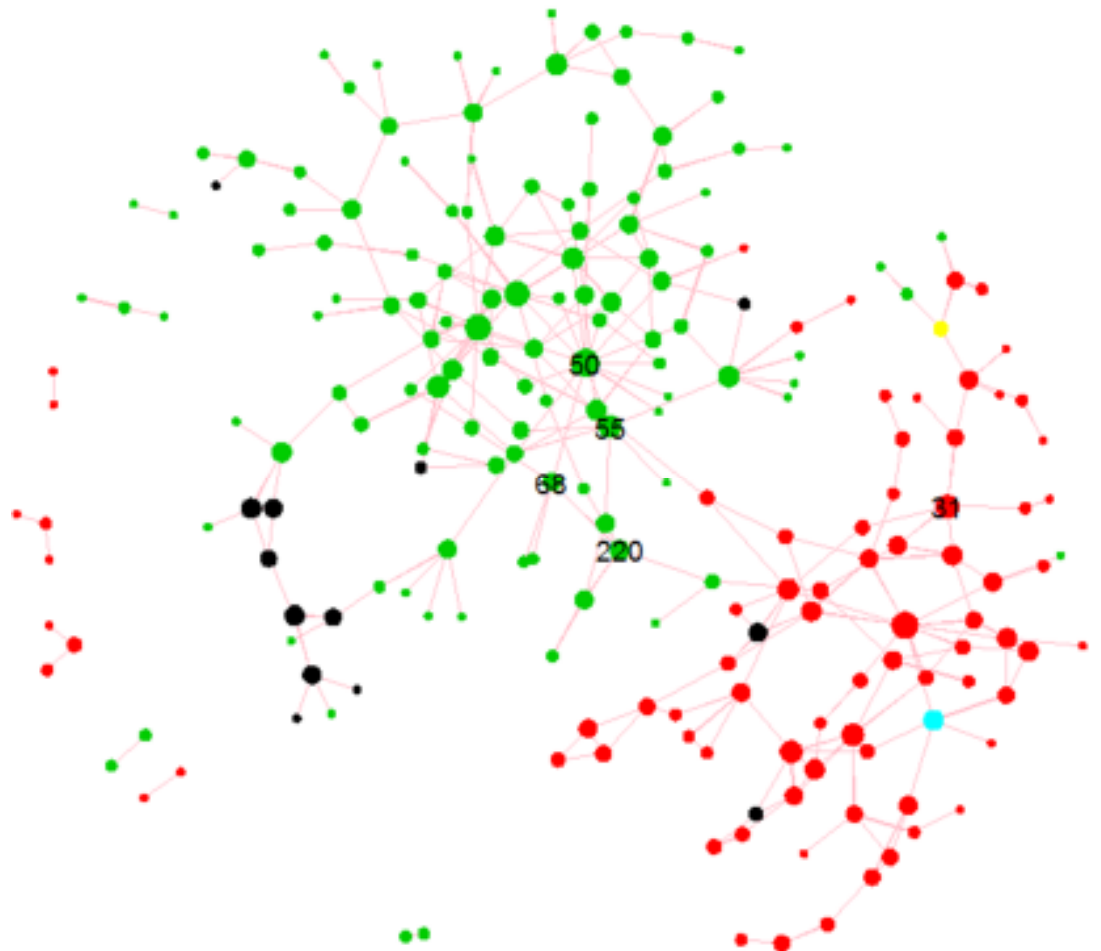
# Scale nodes based on betweenness

```
#Define a label set that is dependent on  
betweenness centrality
```

```
label2=vector()  
for (i in 1:length(bet1))  
  label2[i]=ifelse  
(bet1[i]>1800,row.names(drugatt[i,]),NA)
```

```
#plot as network object, using high between  
nodes as labels
```

```
plot.network(drugn, mode =  
"fruchtermanreingold",  
vertex.col=drugatt$Ethnicity,  
label=label2,displayisolates=FALSE,  
label.cex=1.1, usearrows=FALSE,  
edge.col="pink",  
vertex.cex=logb((deg1+1.1), base=7),  
label.pos=5,  
vertex.border=drugatt$Ethnicity,  
edge.lwd=.25)
```




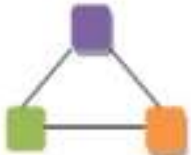






# Antecedent Analysis

- Many recent statistical and computational advances in network analysis have been in the area of exponential random graph models (ERGMs).
- These models allow a researcher to investigate the organizing principals that give rise to an observed network structure.
- Even though the network under study may be a single representation of an evolving system, the patterns of network ties act as the structural signature of the network and provide evidence from which we may infer something of the social processes that built the network (Robins, 2011, p. 484).

Goal: identify the processes and factors affecting organizational collaboration during disasters

# Examples of organizing principals in an undirected graph

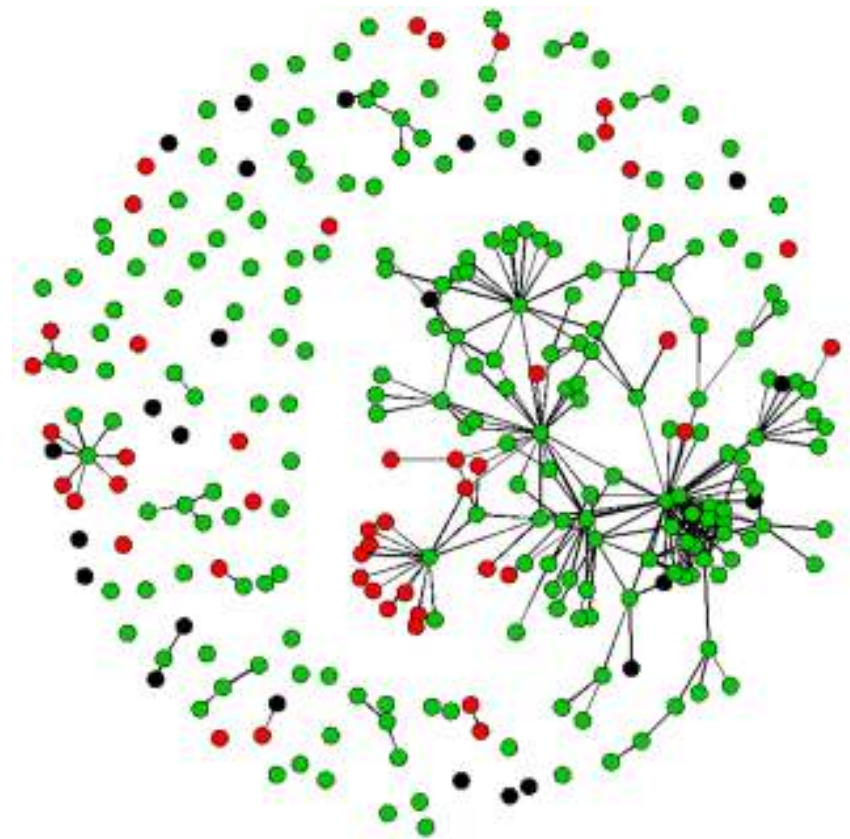
Parameter			Description
Edges			Overall tendency for ties to form in the network
Triangles/Triadic Closure			Tendency to form closed triangles in the network
Attribute based popularity			Main effect of categorical nodal attribute
Attribute based homophily			Propensity to form ties with those similar to you

# Response network to the 2009 Padang earthquake

## Attribute data

	Fund2	Juris2
1	1	1
2	1	2
3	2	2
4	1	3
5	3	1
6	1	3

## Relational Data



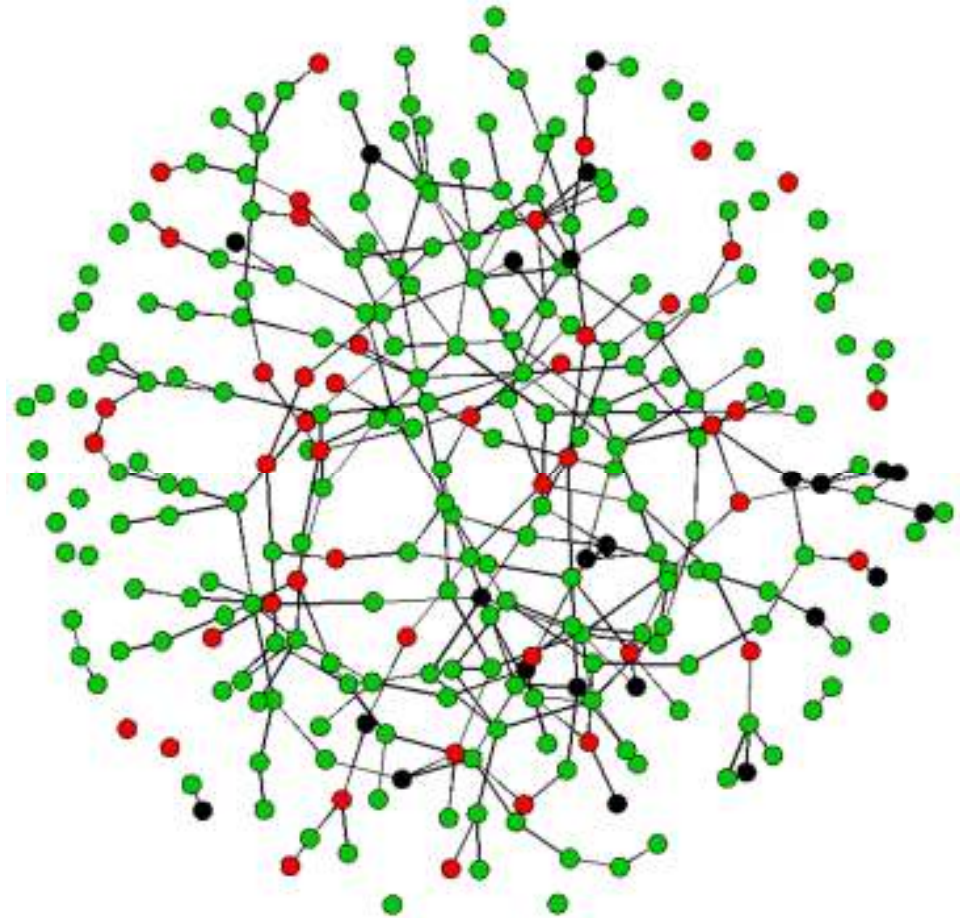
```
#read in a Pajek data file
padang=read.paj("Padang_Pajek.net")
#read in the attributes from a text file
att=read.delim("Padang_Attrib.txt", header=TRUE)
#create a network object and assign the attributes to
the nodes
padang2=as.network(padang2, vertex.attr=att,
directed=FALSE)
#plot the network, with vertex colors defined by
funding type
plot(padang2, vertex.col='Fund2')
```

# Assume a random process

```
# a completely random process
rmodel=ergm(padang2~edges)
summary(rmodel)
Monte Carlo MLE Results:
      Estimate Std. Error MCMC % p-value
edges -4.98450   0.06029    NA  <1e-04 ***
gden(padang2) #graph density of the observed
network (observed ties/total possible ties)
[1] 0.006796712

#simulate networks based on the random model
rmodel.sim=simulate(rmodel, nsim=10)
#take a look at a simulated network
plot(rmodel.sim$networks[[1]], vertex.col='Fund2')

#this ERG model is equivalent to generating a random
graph (function in R is rgraph) with the same tie
probability as found in the observed network
```





# A more realistic model

#in addition to an edge term, we can pose other factors as important in network formation. For example, the node factor term captures the propensity for nodes of a particular type to form ties.

```
model5=ergm(padang2~edges +  
nodefactor("Fund2") + nodefactor("Juris2") +  
nodematch("Fund2", diff=T) + nodematch("Juris2",  
diff=T)+ isolates + gwesp(0.30, fixed=T))
```

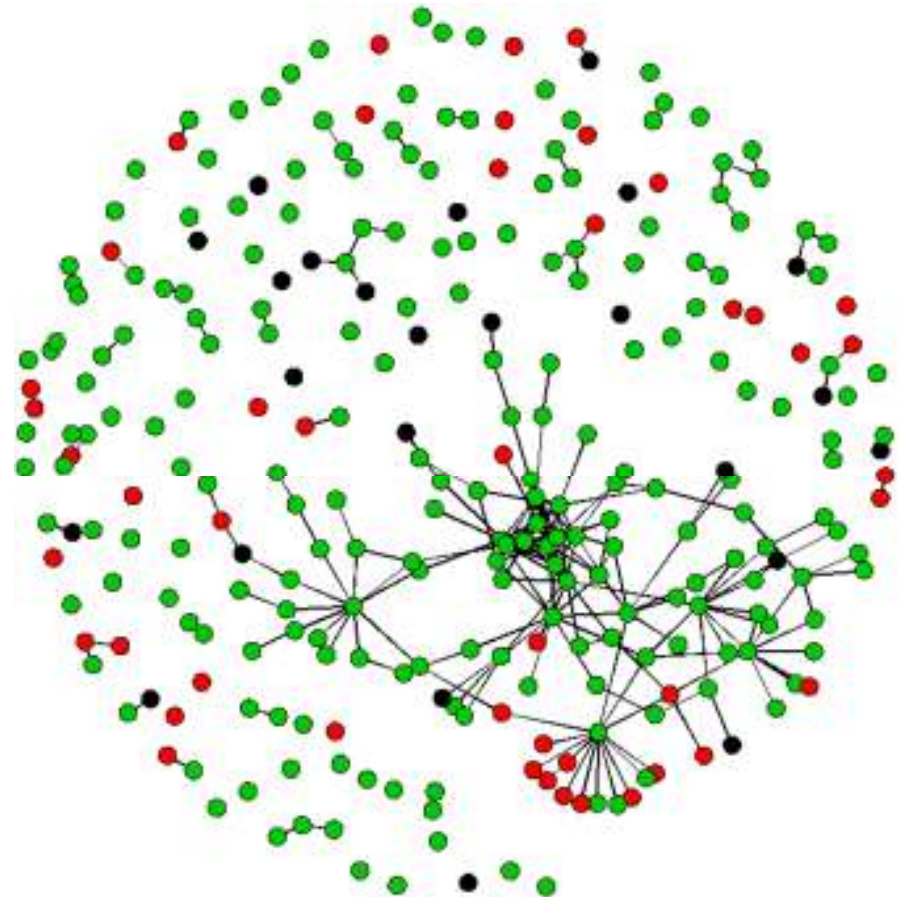
```
summary(model5)#if you wanted to see the specific  
parameter values and level of significance
```

```
#simulate some networks based on the model
```

```
model5.sim=simulate(model5, nsim=10)
```

```
#plot one of the simulated networks
```

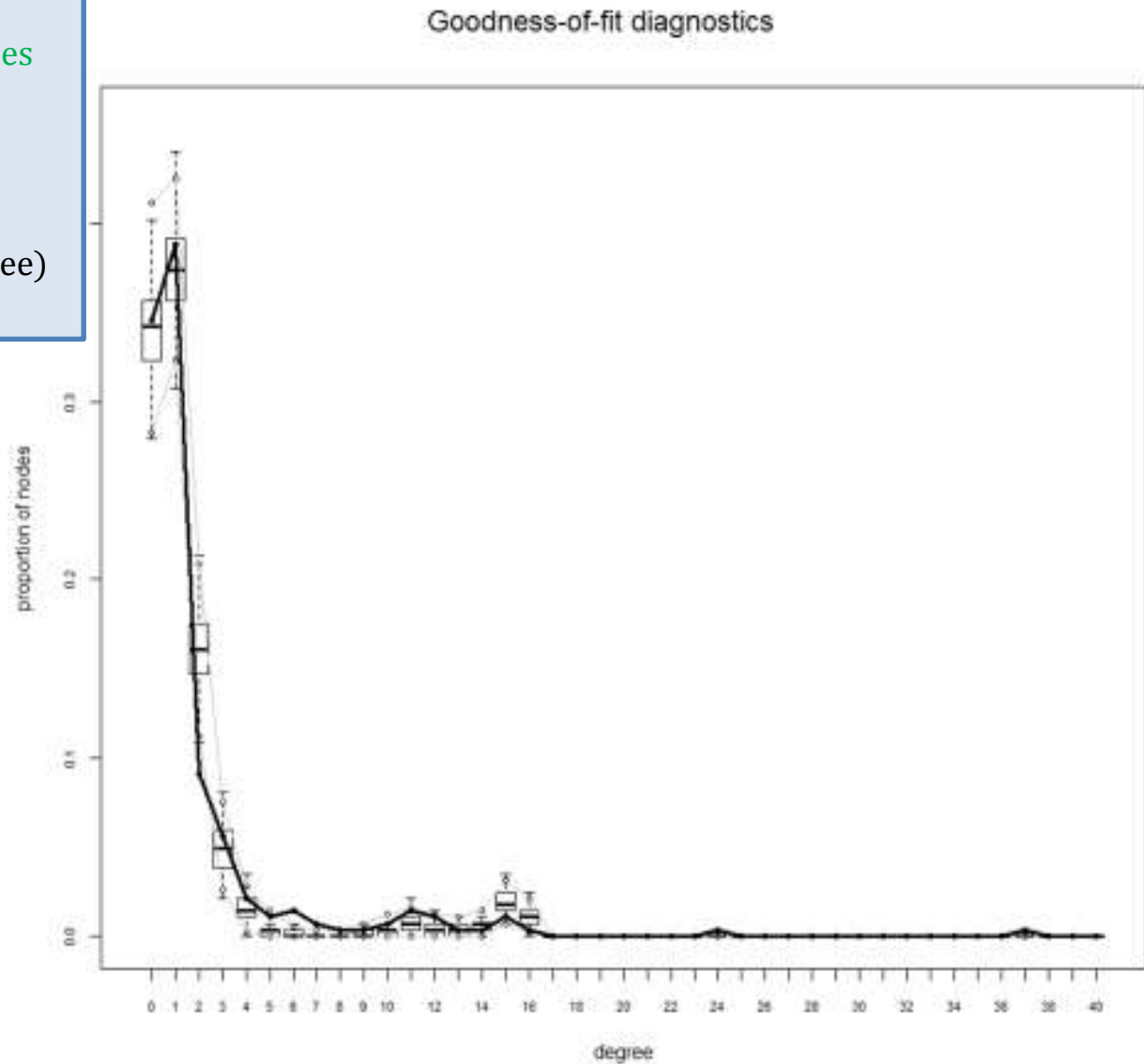
```
plot(model5.sim$networks[[1]],vertex.col='Fund2' )
```



#Goodness of fit to test our model parameters. We can test this by picking a network statistic that is not in the model, and comparing the value of this statistic in the observed network to the distribution of values we obtained in the simulated networks

#lets look at degree

```
model5.gofdeg=gof(model5~degree)  
plot(model5.gofdeg)
```





# Consequence Analysis

- Used to investigate the association between network variables and various group and/or individual level outcomes through the use of econometric models.
- Can also measure and estimate the social influence of alters' belief on ego through a class of models known as network autocorrelation models.
  - These models are predicated on the belief that individuals connected within a social network transmit interpersonal influence on one another.
  - Leenders (2002) states that an actor may form a belief based on his or her own intrinsic values as well as based on the beliefs of his or her alters.

$$y = \rho_1 W_1 y + \chi \beta + \varepsilon$$

Goal: identify the influence of peer self-efficacy on a teacher's self-efficacy

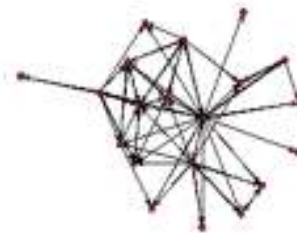
# Teacher social interactions in 21 schools

- Attribute data

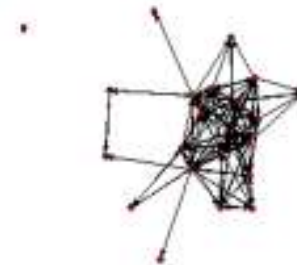
- Age
- Race
- Grade Level
- Education
- Years Teaching
- Efficacy
- Commitment
- Trust
- Reflective Dialogue
- Collective Responsibility
- Instructional Leadership
- Etc...

## Relational Data

Site 1 Friendship Network



Site 2 Friendship Network



Site 3 Friendship Network



Site 1 Advice Network



Site 2 Advice Network



Site 3 Advice Network



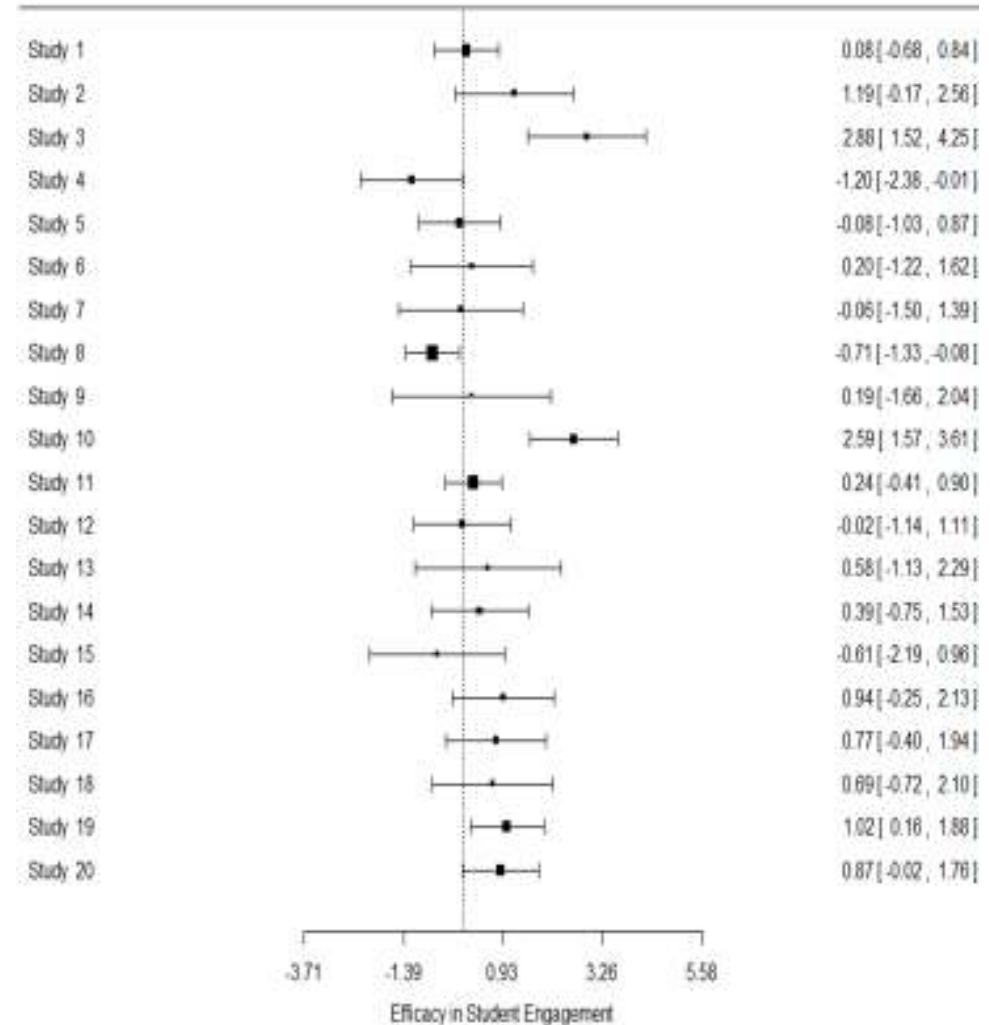
# Analyzing peer influence

## Influence from Advice Ties

```
#run a network autocorrelation model for each school;
the model uses a set of actor attributes along with a set of
advice and friendship relations to predict a teacher's sense of
self efficacy with regard to student engagement; both
attributes and networks are stored in lists where each school
is a component
lnam_se=list()
for (i in c(1:18,20,21))#site 19 dropped from study
  lnam_se[[i]]=lnam(SelfEfficacy [[i]],
cbind(IndVar1[[i]], IndVar2[[i]], IndVar3[[i]]),
W1=list(advicenet[[i]], friendnet[[i]]) )

lnam_se_sum=lapply(lnam_se, summary)#store the summary
results in a list

#pull out the influence effect of the advice network in each of
the models
rho1=vector()
for (i in c(1:18,20,21))
  rho1[i]=lnam_se_sum[[i]]$rho1[[1]]
#pull out the standard error of the influence effect
rho1se=vector()
for (i in c(1:18,20,21))
  rho1se[i]=lnam_se_sum[[i]]$rho1.se[[1]]
#create a simple dataset of model results
se_results=as.data.frame(cbind(rho1, rho1se))
#use a forest plot to display the influence effects in each of the
schools
library("metafor")
forest(se_results$rho1, sei=se_results$rho1se, xlab="Efficacy
in Student Engagement")
```



# Meta-analysis

- Move from network models to standard meta-analysis techniques to pool the results across the schools. Produce a population level effect.

	Advice Network		Friendship Network	
	$\rho$	$p$ -value	$\rho$	$p$ -value
<b>Efficacy</b>				
Student Engagement	0.3236	0.0000	-0.4608	0.0000

# Some online resources

- Steve Borgatti's webpage: <http://www.steveborgatti.com/>
- Dai Shizuka's homepage:  
<https://sites.google.com/site/daishizuka/toolkits>
- Stanford SNA labs in R: <http://sna.stanford.edu/rlabs.php>
- Network analysis with igraph:  
<http://igraph.sourceforge.net/igraphbook/>
- Statnet resources:  
<https://statnet.csde.washington.edu/trac>

# Contact Info

Michael D. Siciliano  
Department of Public Administration  
College of Urban Planning and Public Affairs  
University of Illinois at Chicago  
[sicilian@uic.edu](mailto:sicilian@uic.edu)