Network Analysis 2
Statistical Approaches and Modeling

Statistics

Vladimir Batagelj

IMFM Ljubljana, IAM UP Koper, and NRU HSE Moscow

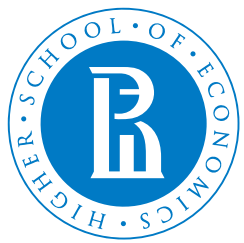
XIII Summer School of the ANR-Lab
Network Analysis and Contemporary Decision Sciences
International Laboratory for Applied Network Research
NRU HSE, Moscow, August 2022

Outline

1 Autocorrelation
2 Network statistics
3 CUG
4 QAP
5 What else?
6 Resources

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (September 6, 2022 at 00 : 36): slides PDF



Continuous Autocorrelation

Assume that each node has score on continuous variable, such as age or rank.

Positive autocorrelation exists when nodes of similar value of variable tend to be adjacent

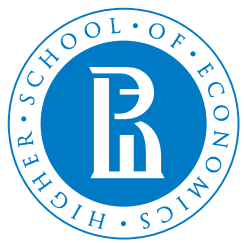
- Friendships tend to be homophilous wrt age
- Mentoring tends to be heterophilous wrt age

We can measure similarity via difference (Geary) or product (Moran).

Autocorrelation Measures

Two measures were “borrowed” from spatial statistics:

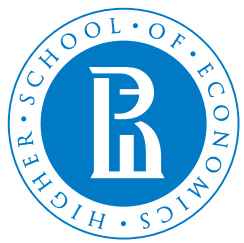
- Geary’s C
  - Also called Geary’s [Contiguity] Ratio
  - Most sensitive to local autocorrelation
- Moran’s I
  - Measures autocorrelation not only on variable values or location (adjacency), but rather on both simultaneously
  - More sensitive to global autocorrelatoin
- I is about covariation of pairs, C is about variation in variable values



Statnet `nacf`

Both measures are built in the statnet's `nacf`. It computes dependence statistics for the given vector on network structure, for neighborhoods of various orders. It produces a vector of measures for multiple steps out into the network, up to the theoretical maximum indicated by the order of the network.

For our purposes, we consider only the immediate neighborhood around each node, meaning that we are interested in autocorrelation between nodes that are just one step from one another. The vector goes from 0 steps to however many nodes are present in the network. Thus, the [2] in the scripts refers to the second measure in the vector (one step).



Geary's C

Let $a_{uv} > 0$ indicate adjacency of nodes u and v, and X_u indicate the score of node u on attribute X (e.g., age)

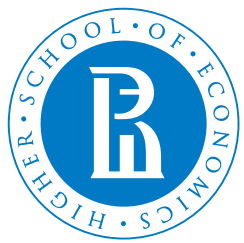
$$C = (n-1)\frac{\sum_{u,v} a_{uv}(x_u - x_v)^2}{2\sum_{u,v} a_{uv} \cdot \sum_u (x_u - \bar{x})^2}$$

Range of values: $0 \leq C \leq 2$

- $C = 1$ indicates independence;
- $C > 1$ indicates negative autocorrelation;
- $C < 1$ indicates positive autocorrelation (homophily)

For $\alpha \neq 0, \beta \in \mathbb{R}$ we have

$$C(\alpha X + \beta) = C(X)$$



Moran's I

$$I = n\frac{\sum_{u,v} a_{uv}(x_u - \bar{x})(x_v - \bar{x})}{\sum_{u,v} a_{uv} \cdot \sum_u (x_u - \bar{x})^2}$$

Ranges between -1 and +1

- Expected value under independence is $\frac{-1}{n-1}$
- $I \to +1$ when positive autocorrelation
- $I \to -1$ when negative autocorrelation

For $\alpha \neq 0, \beta \in \mathbb{R}$ we have

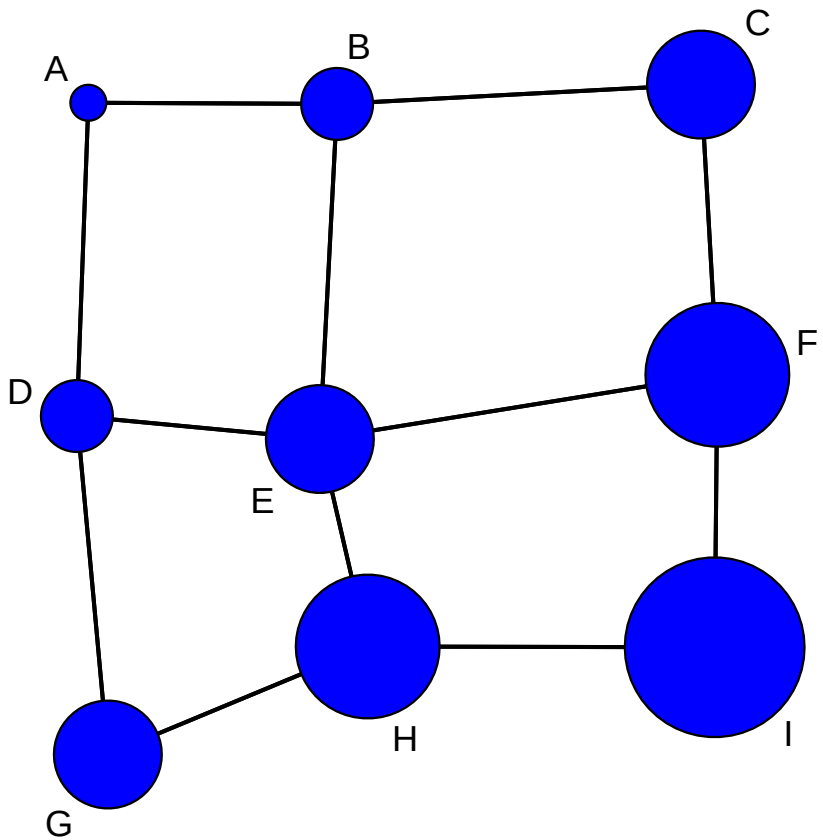
$$I(\alpha X + \beta) = I(X)$$

Simple example


```
library(statnet)
g <- c(
  0, 1, 0,  1, 0, 0,  0, 0, 0,
  0, 0, 1,  0, 1, 0,  0, 0, 0,
  0, 0, 0,  0, 0, 1,  0, 0, 0,
  0, 0, 0,  0, 1, 0,  1, 0, 0,
  0, 0, 0,  0, 0, 1,  0, 1, 0,
  0, 0, 0,  0, 0, 0,  0, 0, 1,
  0, 0, 0,  0, 0, 0,  0, 1, 0,
  0, 0, 0,  0, 0, 0,  0, 0, 1,
  0, 0, 0,  0, 0, 0,  0, 0, 0  )
G <- symmetrize(matrix(g,byrow=TRUE,nrow=9))
rownames(G) <- colnames(G) <- c("A","B","C","D","E","F","G","H",
plot.sociomatrix(G)
gplot(G,gmode="graph",displaylabels=TRUE)
a1 <- c( 1, 2, 3, 2, 3, 4, 3, 4, 5 )
gplot(G,gmode="graph",displaylabels=TRUE,vertex.cex=a1)
nacf(G,a1,type="geary",mode="graph")[2]
nacf(G,a1,type="moran",mode="graph")[2]
a2 <- c(3, 4, 3, 4, 3, 2, 1, 2, 5 )
nacf(G,a2,type="geary",mode="graph")[2]
nacf(G,a2,type="moran",mode="graph")[2]
a3 <- c(4, 1, 4, 2, 5, 2, 3, 3, 3 )
nacf(G,a3,type="geary",mode="graph")[2]
nacf(G,a3,type="moran",mode="graph")[2]
```

Positive Autocorrelation

Similars adjacent; Geary's $C < 1$, Moran's $I > -0.125$

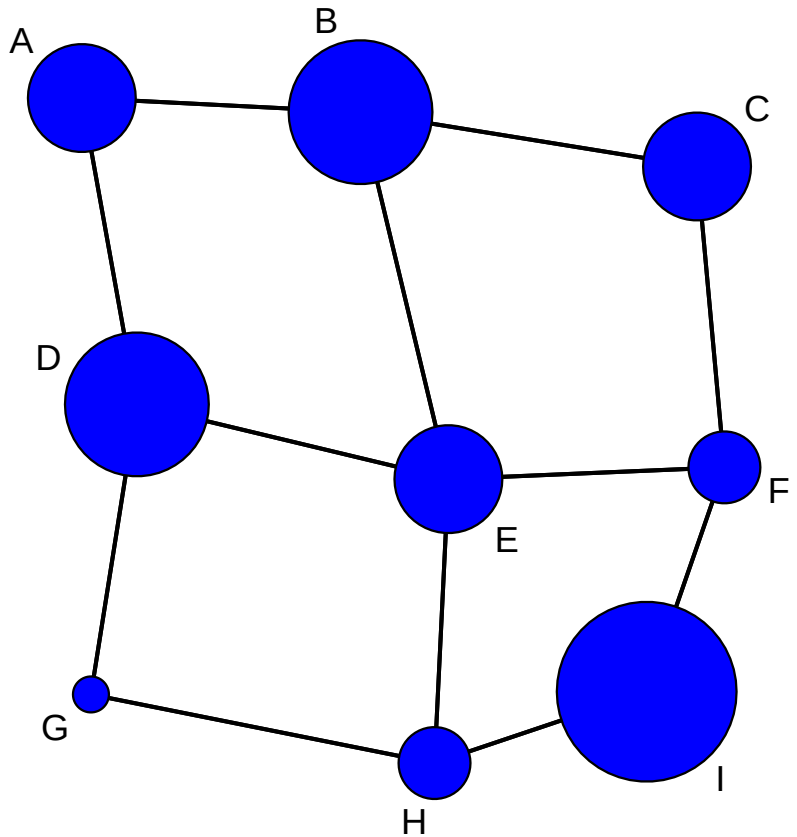


Node	Attrib
A	1
B	2
C	3
D	2
E	3
F	4
G	3
H	4
I	5

Geary's C: 0.333
Moran's I: 0.500

No Autocorrelation

Random pattern; Geary's $C \approx 1$, Moran's $I \approx -0.125$

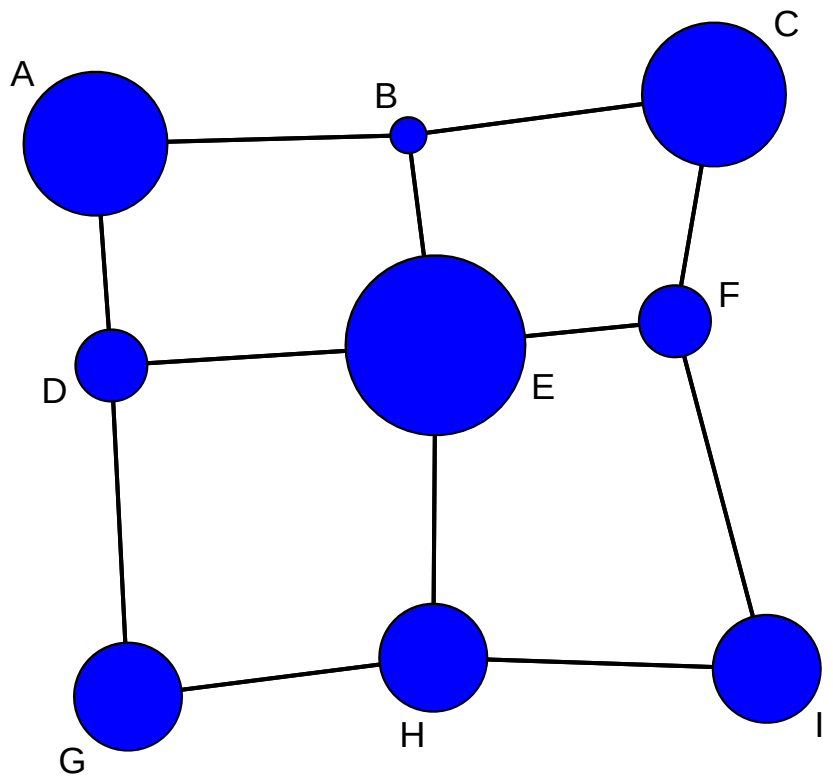


Node	Attrib
A	3
B	4
C	3
D	4
E	3
F	2
G	1
H	2
I	5

Geary's C: 1.000
Moran's I: -0.250

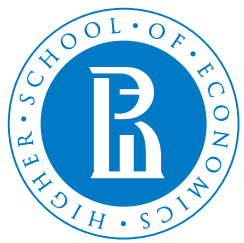
Negative Autocorrelation

Dissimilars adjacent; Geary's $C > 1$, Moran's $I < -0.125$



Node	Attrib
A	4
B	1
C	4
D	2
E	5
F	2
G	3
H	3
I	3

Geary's C: 1.833
Moran's I: -0.875



Interpreting Autocorrelation

Moran's I ranges from -1 to 1 and is interpreted like Pearson's Correlation Coefficient.

- A value near +1.0 indicates clustering (adjacency tends to accompany similarity along a dimension)
- A value near -1.0 indicates dispersion (adjacency tends to accompany dissimilarity along a dimension)
- a value near 0 indicates random distribution (independence)

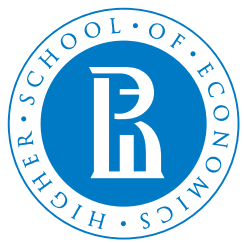
Geary's C ranges from 0 to 2. Just replace +1, -1 ond 0 in the above with 0, 2 and 1.

Florentine families

What is the dependance of the wealth of a family on marriage network?

```
> library(statnet)
> data(florentine)
> fw <- flomarriage %v% "wealth"
> gplot(flomarriage,displaylabels=TRUE,vertex.cex=0.025*fw,
+ gmode="graph",main='Florentine families')
> I <- nacf(flomarriage,fw,type="moran",mode="graph")
> I
           0           1           2           3           4
 1.00000000 -0.31073529  0.06531299 -0.06045322 -0.06267282  0.0
> C <- nacf(flomarriage,fw,type="geary",mode="graph")
> C
           0           1           2           3           4
0.000000000 1.683607336 0.811642725 0.899673648 0.826831324 0.00
> I[2]
          1
-0.3107353
> C[2]
        1
1.683607
```

We can see that the Florentine marriage network is (moderately, I=0.31) negatively (C = 1.68) autocorrelated with respect to the family wealth. Usually bride and spouse were not both equally rich.

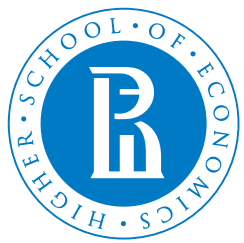


Problem with network statistics

Some issues with network statistics:

- Samples non-random
- Often work with populations
- Observations not independent
- Distributions unknown

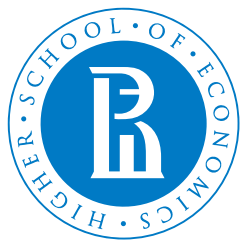
One crucial thing in SNA is that even when networks are completely 'random' they exhibit certain non-random network features. For example, the embeddedment in triads. When a random network is dense, nodes will have a higher chance to be embedded in triads by default. Hence, you cannot really say that in one network nodes are more clustered than in another if you do not consider this.



General strategy

Network inference assuming conditional independence:

1. Calculate on the “observed” network a network statistic (global measure) that you are interested in.
2. Think about the properties of the network that you want to conserve.
3. Generate many “random” networks that have the same properties as the “observed” network.
4. Calculate the network statistic on these “conditional random networks” and compare this baseline distribution against the actually observed network statistic in the “observed” network.



Conditional Uniform Graphs (CUG)

This strategy is implemented in the sna function `cug.test` – Conditional uniform graphs (CUGs).

It supports conditioning the simulated networks on three of the possible modes (`cmode`): *size*; *number of edges*; and the *distribution of dyads*. Select the conditioning mode according to what you suspect about your own network of interest. (Note: CUG can be made to condition on other properties.)

In the following example we will try for comparision all three options. For a statistics we will use the betweenness centralization. Centralization involves two items: the network being analyzed; and the name of the centrality measure being applied. Other global measures will, therefore, not require the FUN.arg=list(), argument.

Conditional Uniform Graphs (CUG)

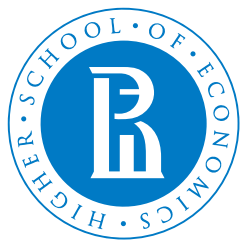

```
library(statnet)
data(florentine)
gplot(flobusiness,gmode="graph",displaylabels=TRUE)

rSize <- cug.test(flobusiness,centralization,
  FUN.arg=list(FUN=betweenness),mode="graph",cmode="size")
rEdges <- cug.test(flobusiness,centralization,
  FUN.arg=list(FUN=betweenness),mode="graph",cmode="edges")
rDyad <- cug.test(flobusiness,centralization,
  FUN.arg=list(FUN=betweenness),mode="graph",cmode="dyad.census"

# Aggregate results
Betweenness <- c(rSize$obs.stat,rEdges$obs.stat,rDyad$obs.stat)
PctGreater <- c(rSize$pgteobs,rEdges$pgteobs,rDyad$pgteobs)
PctLess <- c(rSize$plteobs,rEdges$plteobs,rDyad$plteobs)
report <- cbind(Betweenness, PctGreater, PctLess)
rownames(report) <- c("Size","Edges","Dyads")
report
```

Conditional Uniform Graphs (CUG)


```
       Betweenness PctGreater PctLess
Size     0.2057143       0.001   0.999
Edges    0.2057143       0.713   0.289
Dyads    0.2057143       0.738   0.263
```

Consider the visualization of the network. Given the network's structure, it is at least somewhat dominated by the Barbadori and Medici families (betweenness centralization = 0.21). But is that level of centralization special to the Florentine business network, or is this something that we would normally expect for a network this size? Is it something that we would normally expect for a network with this number of edges? Is it something that we would normally expect for a network with this distribution of dyads?

As we can see in the output above, this level of centralization is very uncommon in a network of this size. But it is not at all uncommon in a network with the same number of edges, or the same distribution of dyads.

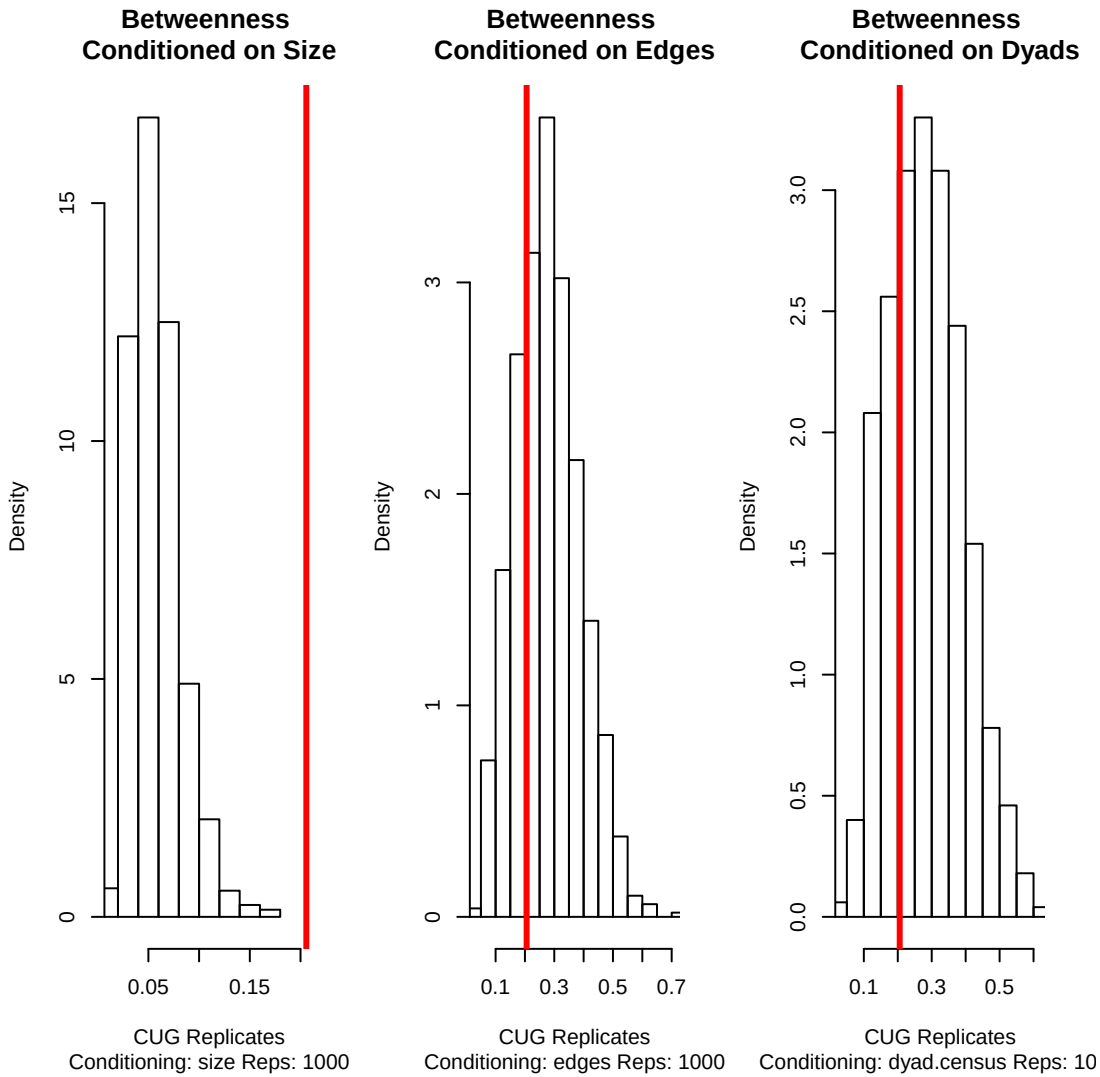


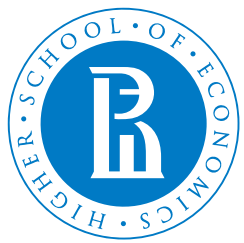
Conditional Uniform Graphs (CUG)

We can depict the same information graphically by displaying the distribution of betweeness centralization measures for the randomly generated networks, and indicating where the betweenness centralization measure of 0.21 lies in comparison to each distribution.

```
par(mfrow=c(1,3))
plot(rSize,main="Betweenness \nConditioned on Size")
plot(rEdges,main="Betweenness \nConditioned on Edges")
plot(rDyad,main="Betweenness \nConditioned on Dyads")
par(mfrow=c(1,1))
```

Conditional Uniform Graphs (CUG)

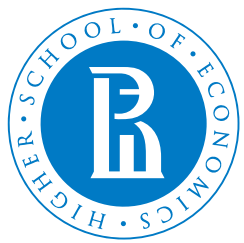




Quadratic Assignment Procedure (QAP)

Quadratic assignment procedure (QAP) is similar to CUG, in that it uses simulation in order to generate a distribution of hypothetical networks. But QAP controls for network structure, as compared with CUG, which controls for size, the number of edges, or dyad census. QAP is useful for running a variety of statistics. It is based on the *permutation test*

- Get observed test statistic
- Construct a distribution of test statistics under null hypothesis
  - Thousands of permutations of actual data
- Count proportion of statistics on permuted data that are as large as the observed
  - This is the p-value of the test



Quadratic Assignment Procedure (QAP) graph correlation

Did the Florentine families base their business dealings on marriage ties? (Or maybe their marriages are based on their business ties?)
Get the graph correlation value

```
> gcor(flobusiness,flomarriage)
[1] 0.3718679
```

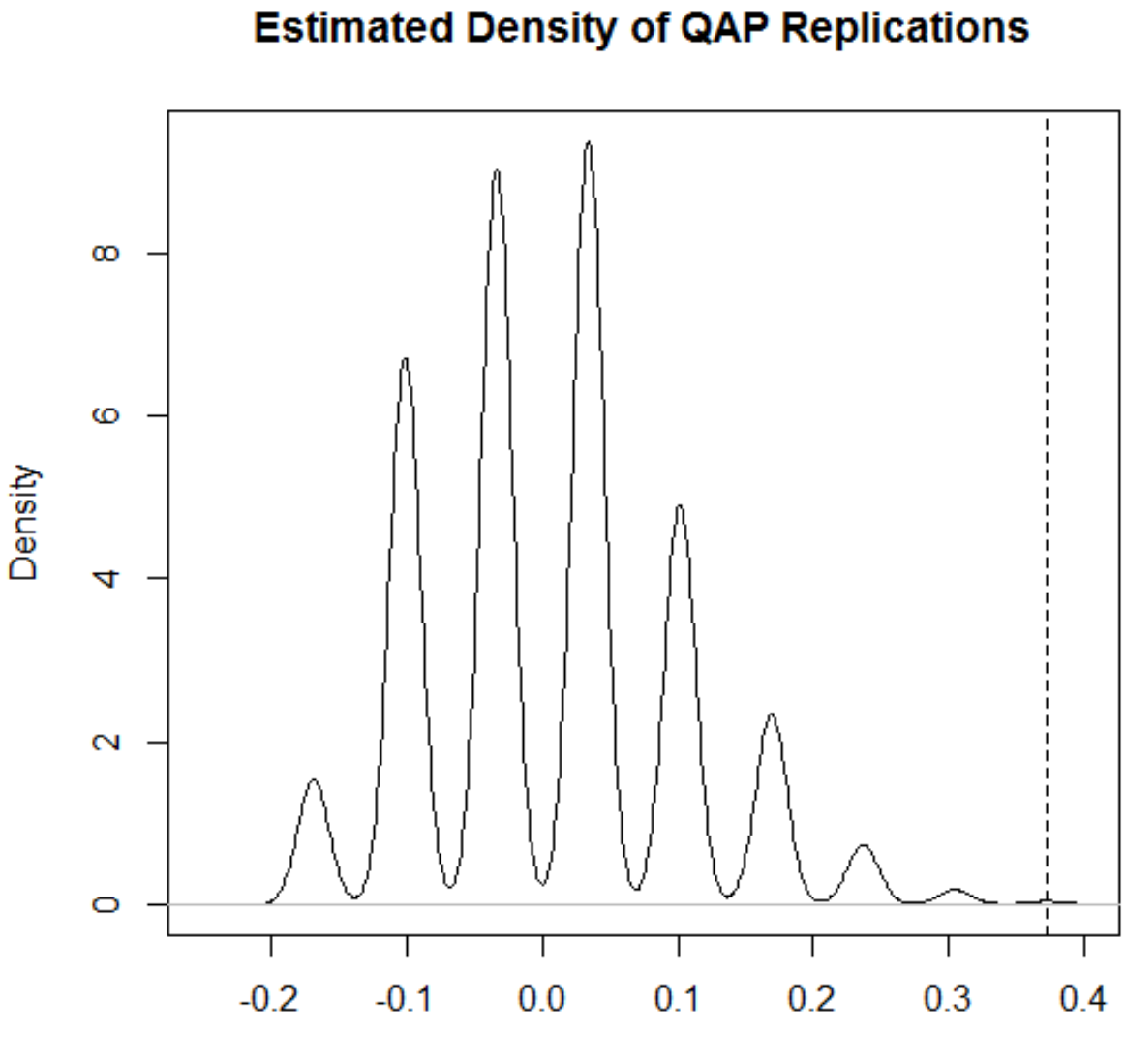
Marriage and business ties are moderately correlated (r=0.37).
Is it significant?

```
> (rCor <- qaptest(list(flobusiness,flomarriage),gcor,
+   g1=1,g2=2,reps=1000))
QAP Test Results

Estimated p-values:
        p(f(perm) >= f(d)): 0.001
        p(f(perm) <= f(d)): 1
```

The correlation is significant at the 0.05 alpha level. We know this because less than 5% of the permuted networks - or in this case, all of them - exhibited correlation coefficients that were less than that of the value we calculated for the marriage network and the business.

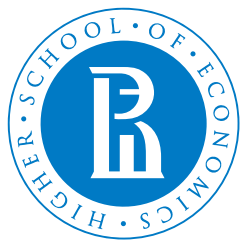
```
> plot(rCor, xlim=c(-0.25, 0.4))
```

Quadratic Assignment Procedure (QAP) graph correlation



What else?

- QAP and regression
- MRQAP
- ERGM – Exponential random graph models
- Stochastic actor-oriented models, Siena
- Stochastic blockmodeling



Resources I

Decker, Krackhardt, Snijders: Sensitivity of MRQAP Tests to Collinearity and Autocorrelation Conditions.

Grund, Thomas: Lectures, 2016.

Kilduff, Martin, Tsai, Wenpin: Social Networks and Organizations. SAGE publications, 2003.

Kolaczyk, Eric D., Csárdi, Gábor: Statistical Analysis of Network Data with R (Use R!). Springer, 2014.

Lusher, Dean, Koskinen, Johan, Robins, Garry : Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications. Cambridge University Press, 2013.

Murphy, Phil: Practicum 9 - Hypothesis Testing in Network Analysis. 2017.

Simpson, William: “QAP – the Quadratic Assignment Procedure” , Harvard Business School, 2001. page