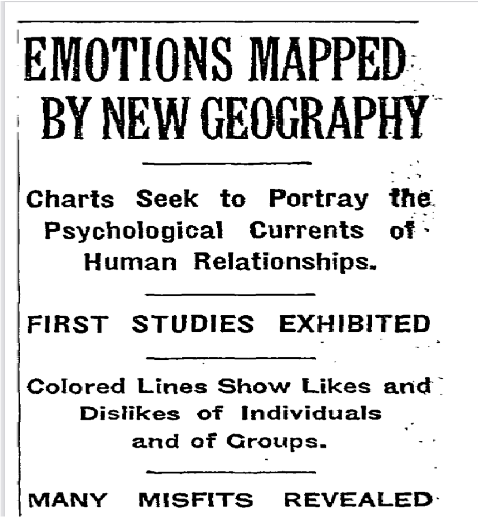
# Analyzing Networks

Brandon Bolte

Penn State University

November 18, 2020

# Sociometry and Network History



**EMOTIONS MAPPED BY NEW GEOGRAPHY**

Charts Seek to Portray the Psychological Currents of Human Relationships.

FIRST STUDIES EXHIBITED

Colored Lines Show Likes and Dislikes of Individuals and of Groups.

MANY MISFITS REVEALED

# Moreno...

“If we ever get to the point of charting a whole city or a whole nation, we would have an intricate maze of psychological reactions which would present a picture of a vast solar system of intangible structures, powerfully influencing conduct, as gravitation does bodies in space. Such an invisible structure underlies society and has its influence in determining the conduct of society as a whole...”

# Obligatory "Big" Definitions

A **Network** is a representation of a set of actors or units and the relations or information flows between them.

**Network Analysis** as an approach views (Wasserman and Faust 1994):

- Actors or actions as interdependent rather than autonomous
- Relational ties between actors as channels for the flow of information or resources
- Individual units are embedded in a set of relations among others, which provide opportunities for or constraints upon individual actions
- Network models as a conceptualization of structure as a lasting pattern of relations.

# Some Initial Terminology

**Components of a Network**

- **Node** (aka actor, vertex): discrete units between which relations can occur (isolates, dyads, triads, etc.)
- **Edge** (aka relation, tie, link): connection or individual relation between nodes (Directed vs Non-directed, Weights, Strong vs Weak)

# Some Initial Terminology

**Components of a Network**

- **Node** (aka actor, vertex): discrete units between which relations can occur (isolates, dyads, triads, etc.)
- **Edge** (aka relation, tie, link): connection or individual relation between nodes (Directed vs Non-directed, Weights, Strong vs Weak)

**Network Characteristics, etc.**

- **Size**: number of nodes
- **Density**: proportion of edges to potential edges
- **Average Path Length**: Largest shortest path between any two nodes in a network
- **Geodesic Distance**: shortest path between two nodes

# Some Initial Terminology

**Components of a Network**

- **Node** (aka actor, vertex): discrete units between which relations can occur (isolates, dyads, triads, etc.)
- **Edge** (aka relation, tie, link): connection or individual relation between nodes (Directed vs Non-directed, Weights, Strong vs Weak)

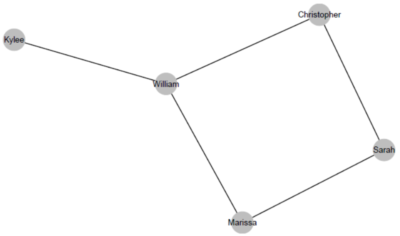
**Network Characteristics, etc.**

- **Size**: number of nodes
- **Density**: proportion of edges to potential edges
- **Average Path Length**: Largest shortest path between any two nodes in a network
- **Geodesic Distance**: shortest path between two nodes

**Network Types**

- Directional, Multi-modal (i.e. bipartite), Dynamic, etc.

# Really Basic Network



```
> N5[1:5,1:5]
            Christopher Kylee Marissa William Sarah
Christopher           0     0       0       1     1
Kylee                 0     0       0       1     0
Marissa               0     0       0       1     1
William               1     1       1       0     0
Sarah                 1     0       1       0     0
```

# Describing Networks

Like any dataset, we want to be able to summarize features of a network statistically.

- Node-level, e.g. centrality, local transitivity
- Network-level, e.g. reciprocity, global transitivity

# Centrality as a Concept

What does it mean to be a **Central** node?

- Importance / Influence
- Connectedness
- Vulnerability to contagion
- "Anchor point?"

# Centrality as a Concept

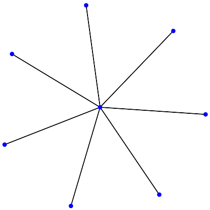
What does it mean to be a **Central** node?

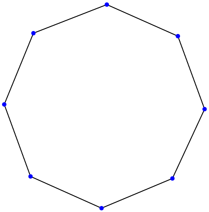
- Importance / Influence
- Connectedness
- Vulnerability to contagion
- “Anchor point?”

Dimensions of Centrality

- Node Positioning
  - Radial: Paths terminate or originate at a node
  - Medial: Paths that pass through a node
- Properties of Paths
  - Volume: Paths contribute according to their numbers
  - Length: Paths contribute according to lengths

# Centrality as a Concept





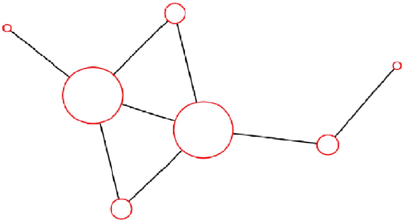
# Degree Centrality (Volume/Radial)

Degree centrality is formally defined as $c_d(\mathbf{G}) = \mathbf{G1}$

where G is an *nxn* adjacency matrix and $\mathbf{1}$ is an $nx1$ column of 1's.

A.k.a. the number of edges connecting that node to other nodes

In directed networks, *indegree* and *outdegree*

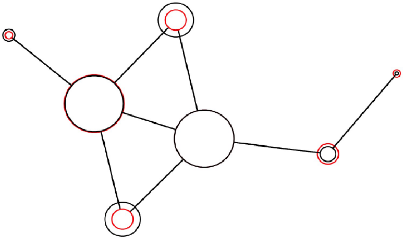


# Eigenvector Centrality (Volume/Radial)

Eigenvector centrality is given by $c_e(\mathbf{G}) = \lambda^{-1}\mathbf{G}c_e(\mathbf{G})$

where $c_e(\mathbf{G})$ is the eigenvector of adjacency matrix $\mathbf{G}$ and $\lambda$ is the largest eigenvalue.

Some connections are more important than others because of those connected nodes' centrality scores.



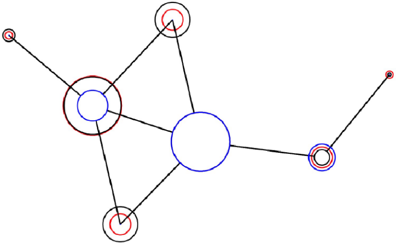
# Betweenness Centrality (Volume/Medial)

Betweenness centrality is given by

$$c_b(\mathbf{A}, v) = \sum_{i,j:i\neq j,i\neq v,j\neq v} \frac{g_{ivj}}{g_{ij}}$$

where $g_{ij}$ are the minimal paths from $i$ to $j$, $g_{ivj}$ is the number through $v$.

How often is the shortest path between two nodes through a given node?

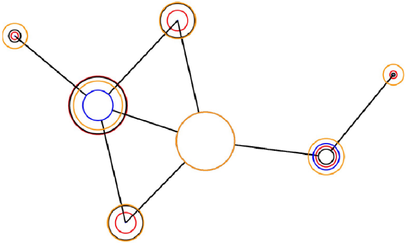


# Closeness Centrality (Length/Radial)

Closeness centrality is given by $c_c(\mathbf{A}, v) = \frac{n-1}{\sum\limits_{i:i \neq v} d(v,i)}$

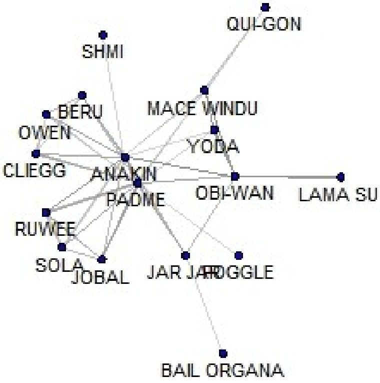
where $d(v, i)$ is the length of the shortest path from $v$ to $i$.

Only factors in shortest paths.



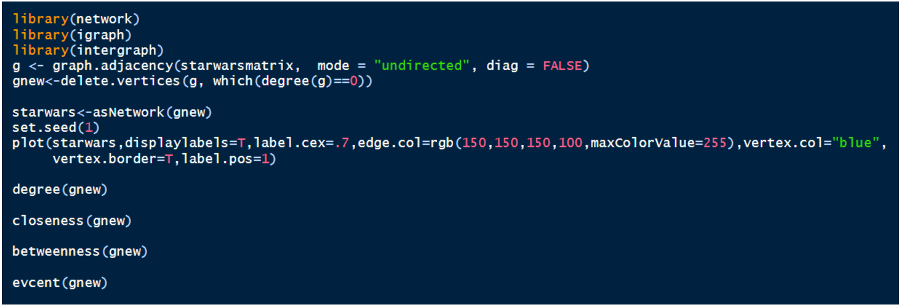
# Star Wars Episode II

“It really is the story of the tragedy of Darth Vader, and it starts when he’s nine, and it ends when he’s dead.” - George Lucas



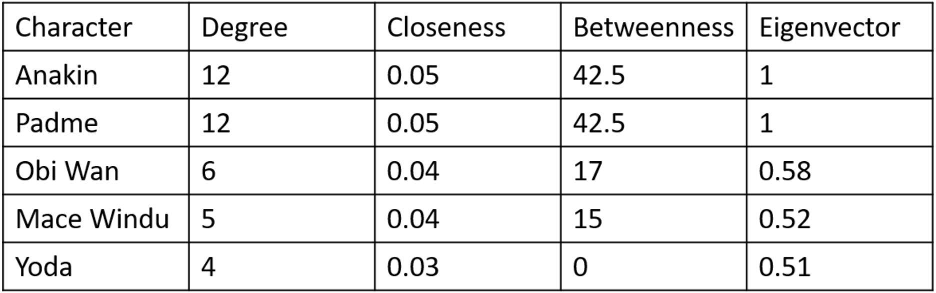
(example courtesy of Evelina Gabasova: http://evelinag.com/blog/2015/12-15-star-wars-social-network/)

# Star Wars Episode II



```
library(network)
library(igraph)
library(intergraph)
g <- graph.adjacency(starwarsmatrix,  mode = "undirected", diag = FALSE)
gnew<-delete.vertices(g, which(degree(g)==0))

starwars<-asNetwork(gnew)
set.seed(1)
plot(starwars,displaylabels=T,label.cex=.7,edge.col=rgb(150,150,150,100,maxColorValue=255),vertex.col="blue",
     vertex.border=T,label.pos=1)

degree(gnew)

closeness(gnew)

betweenness(gnew)

evcent(gnew)
```

# Star Wars Episode II



| Character | Degree | Closeness | Betweenness | Eigenvector |
|---|---|---|---|---|
| Anakin | 12 | 0.05 | 42.5 | 1 |
| Padme | 12 | 0.05 | 42.5 | 1 |
| Obi Wan | 6 | 0.04 | 17 | 0.58 |
| Mace Windu | 5 | 0.04 | 15 | 0.52 |
| Yoda | 4 | 0.03 | 0 | 0.51 |

# Centrality: Practical Things

- Choice of Measure:
    - Degree: basic “connectedness,” popularity
    - Eigenvector: embeddedness
    - Betweenness: information flows, loss if node removed
    - Closeness: proximity, influencers
- Can serve as weights/variables in regression analyses

# Reciprocity

The Garlaschelli and Loffredo (2004) measure for reciprocity is:

$$\rho = \frac{r - \overline{a}}{1 - \overline{a}}$$

where $r$ is the ratio of the number of edges pointing both directions to total number of edges, $\overline{a} = \frac{L}{N(N-1)}$, and $L$ is the total number of edges.

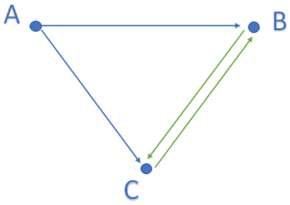
- $\rho > 1$ means reciprocal, $\rho < 1$ means antireciprocal, $\rho = 1$ means all links have mutual pairs, and $\rho = 0$ means neutral
- Obviously only for directed networks
- Assess "loopiness" of network, retaliation, directional redundancies
- Why not just use $r$?

# Transitivity I

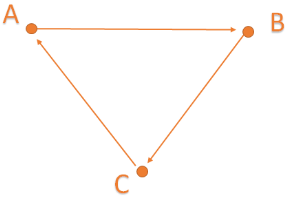
**Transitivity** is the tendency of the last two nodes of a two-path (a path that traverses three nodes) to receive an edge from the first node.

Undirected: triadic closure, clustering, formation of triangles

Directed: directionality matters, distinguishes from **cycles**



a) Transitive Triad



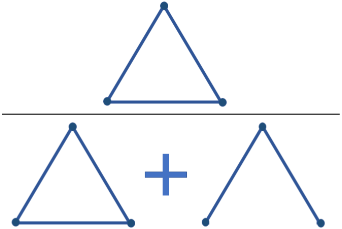
b) Cyclic Triple

# Transitivity II

**Node Level**: Proportion of a node's neighbors that are tied (a.k.a. "clustering coefficient").

**Network Level** (weak transitivity- most common):

$$\frac{\text{transitive and non-vacuous ordered triples}}{\text{ordered triples that are 1 or 0 edges short of non-vacuous transitive}}$$



# Transitivity III

Theoretically

- Hierarchies
- Cohesiveness, clustering, strong ties
- Polarization (between clusters)
- Information Redundancies

# Transitivity III

Theoretically

- Hierarchies
- Cohesiveness, clustering, strong ties
- Polarization (between clusters)
- Information Redundancies

Practical Notes:

- Lots of different measures
- igraph package only does undirected, sna package will do directed
- Can overburden network models... more on that later

# Assortative Mixing (Homophily/Heterophily) I

Qualitative/Discrete

- $e_{ij}$= fraction of edges connecting node of type $i$ to type $j$ (in undirected, $e_{ij} = e_{ji}$)
- $a_i$= nodes of a particular type sent
- $b_i$= nodes of a particular type received

Then,

$$r = \frac{\sum_{i=1}^{k} e_{ii} - \sum_{i=1}^{k} a_i b_i}{1 - \sum_{i=1}^{k} a_i b_i}$$

- $r = 0$ when no assortative mixing ($e_{ij} = a_i b_i$)
- $r = 1$ if "perfect" assortativity
- $r < 0$ if "dissortative" (most edges connect nodes of different types)

# Assortative Mixing (Homophily/Heterophily) II

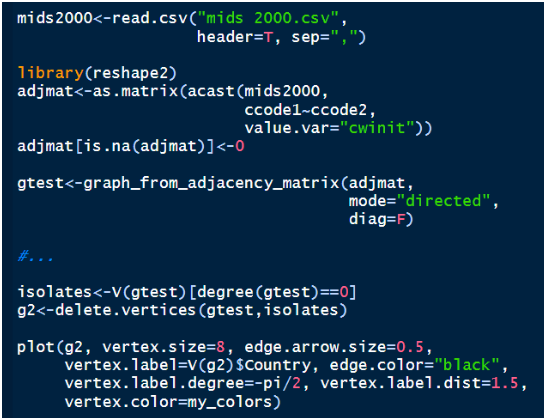
Quantitative/Scalar

- let (x,y) be the values of $z$ observed between senders and receivers of at least one edge
- redefine $r$ as a Pearson's correlation

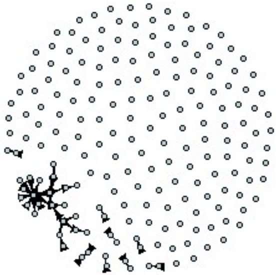
$$r = \frac{\sum_{x,y} xy(e_{xy} - a_x b_y)}{\sigma_a \sigma_b}$$

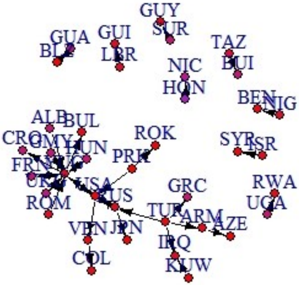
- $[-1:1]$, heterophilous to homophilous

# MIDs 2000

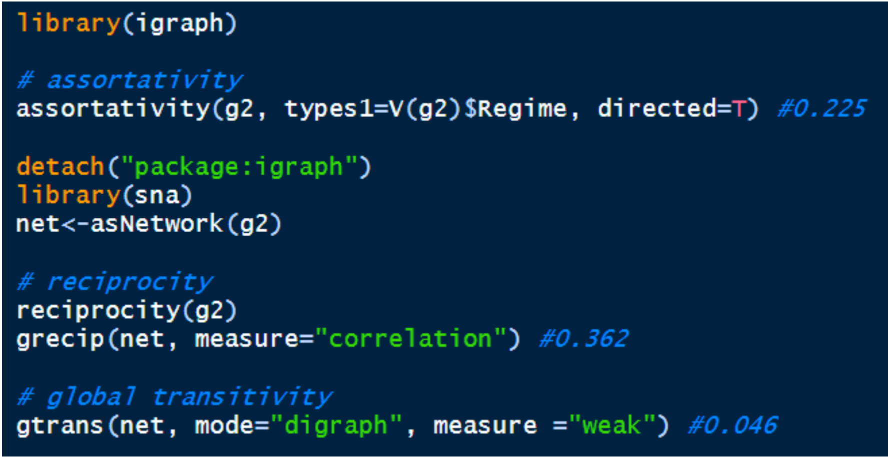


```
mids2000<-read.csv("mids 2000.csv",
                   header=T, sep=",")

library(reshape2)
adjmat<-as.matrix(acast(mids2000,
                        ccode1~ccode2,
                        value.var="cwinit"))
adjmat[is.na(adjmat)]<-0

gtest<-graph_from_adjacency_matrix(adjmat,
                                   mode="directed",
                                   diag=F)


#...

isolates<-V(gtest)[degree(gtest)==0]
g2<-delete.vertices(gtest,isolates)

plot(g2, vertex.size=8, edge.arrow.size=0.5,
     vertex.label=V(g2)$Country, edge.color="black",
     vertex.label.degree=-pi/2, vertex.label.dist=1.5,
     vertex.color=my_colors)
```





# MIDs Stats



```r
library(igraph)

# assortativity
assortativity(g2, types1=V(g2)$Regime, directed=T) #0.225

detach("package:igraph")
library(sna)
net<-asNetwork(g2)

# reciprocity
reciprocity(g2)
grecip(net, measure="correlation") #0.362

# global transitivity
gtrans(net, mode="digraph", measure ="weak") #0.046
```

# Network Modeling I

Why Model Networks?

- Test hypotheses about network structures or predict edges
- Account for network structure
- Deal with higher order dependencies
- Dyads... Conditional Independence Assumption...

# THE WAR(S)!

# Parametric Likelihood Framework for Inference

We observe x, from the random variable (population of networks) X:

$$X \sim f(X, \theta)$$

where $f$ is a family of probability distributions, $\theta$ is an unknown parameter vector to be estimated, and X is the DV (an adjacency matrix).

$$\hat{\theta}_{MLE} = argmax[f(x, \theta)]$$

# Exponential Random Graph Models (ERGMs) I

Assumptions

1. Equal probability of observing any two networks with the same values for included statistics (i.e. correctly specified)
2. Observed network exhibits the average value of these statistics over the networks that could be observed (i.e. like regression, average relationships are representative of the population)

# Exponential Random Graph Models (ERGMs) II

The probability of observing network $N$ is:

$$P(N,\theta) = \frac{exp(\theta' \mathbf{h}(N))}{\sum_{N^* \in \mathcal{N}} exp(\theta' \mathbf{h}(N^*))}$$

- $\mathbf{h}(N)$: Network Statistics
- $\theta$: Effects
- $exp(\theta' \mathbf{h}(N))$: Weight
- $\sum_{N^* \in \mathcal{N}} exp(\theta' \mathbf{h}(N^*))$: Normalizer

**h** can capture almost any form of interdependence among edges and covariates!

# The **h**

Node or Dyadic Covariates:

- $h_D(N) = \sum_{i \neq j} X_{ij} N_{ij}$

Endogenous Network Stats, e.g.:

- Reciprocity $= h_R(N) = \sum_{i<j} N_{ij} N_{ji}$
- Transitivity $= h_T(N) = \sum_{i,j \neq i,k} N_{ij} N_{ik} N_{jk}$

# ERGM Estimation I

The normalizer makes estimation extremely computationally difficult using traditional MLE...

Maximum Pseudolikelihood Estimation (MPLE)

$$P_\theta \approx \prod_{ij} P_\theta(g_{ij}|G_{-ij})$$

Reduces our likelihood to:

$$P_\theta(g_{ij} = 1|G_{-ij}) = logit^{-1}(\theta^T \delta(h(G)))$$

Computationally efficient but downward biased CIs in cross-sectional networks

# ERGM Estimation II

Markov Chain Monte Carlo MLE (MCMC-MLE) - normalizer is a sum over a population of networks, so approximate with a random sample of networks.

1. Start with an initial vector *theta* (determined by MPLE)
2. Choose a dyad at random, randomly determine if there should be a tie, weighted by model
3. Repeat for many steps (MCMC interval)
4. Take the network from the last step and calculate network statistics
5. Repeat this many times (MCMC sample size)
6. Calculate the average for each statistic in teh sample and compare to observed statistic
7. Update parameter estimates
8. Repeat until convergence (no statistical difference between MCMC sample average and observed statistics)

# ERGM Interpretation I

**Node**: A change in a value for a node-level attribute is associated with some change in some change in the log-odds of that node forming an edge with some other node.

- Or change in the log-odds of *sending* an edge
- Or change in the log-odds of *receiving* an edge

**Edge**: $P(N_{ij} = 1 | N_{-ij}, \theta) = logit^{-1}(\sum_{r=1}^{k} \theta_r \delta_r^{(ij)}(N))$

- $\delta_r^{(ij)}(N)$ is the change in $h_r$ when $N_{ij}$ is changed from zero to one.

**Network**: The relative likelihood of observing $N^{j+}$ to observing $N^{j}$ is $exp(\theta_j)$, where

- $\theta_j$ is the estimate of the parameter that corresponds to statistic $j$
- $N^{j+}$ is one unit greater than $N^{j}$ on statistic $j$ (e.g. one more edge, one more triangle, etc.)

# ERGM Interpretation II

Let's say we fit a model with just two network statistics: edges and triangles, where the coefficients are -1.68 and 0.169, respectively. (Note that the "edges" term is analogous to a constant because it captures the density of the network.)

- The conditional log-odds of two nodes forming an edge is $-1.68 * (\text{change in number of edges}) + 0.169 * (\text{change in number of triangles})$
- If a particular new edge would not add any triangles to the network, the log-odds of two nodes forming a tie is -1.68
- If a particular new edge will add one triangle to the network, the log-odds of edge formation is $-1.68 + 0.169 = -1.51$

# Degeneracy

**Degeneracy** occurs when most of the probability mass is concentrated on a few networks of some network population (usually the completely full or completely empty network). This is **bad**.

# Degeneracy

**Degeneracy** occurs when most of the probability mass is concentrated on a few networks of some network population (usually the completely full or completely empty network). This is **bad**.

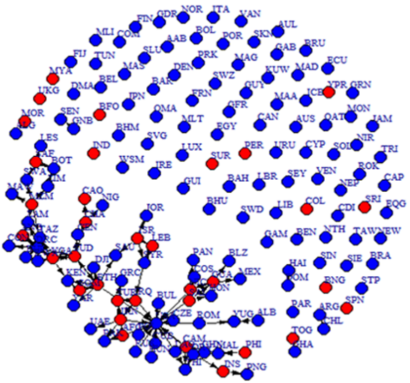
Why might this occur?

- Poor model specification...
- too many higher order dependency functions lead to too many combinatorial complexities

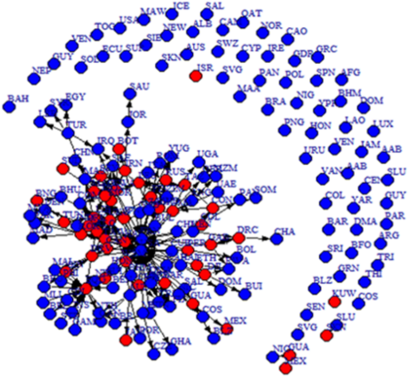
What to do:

- Check that MCMC quantiles are around 0
- Use network stats that down-weight repeated structures that involve the same edge...
  - Transitivity (number of triangles): $\sum_{i<j<k} N_{ij} N_{ik} N_{jk}$
  - Replace with Geometrically Weighted Edgewise Shared Partners (GWESP)...

# ERGM Example: Refugee Networks



Refugee Flows 1984-1988



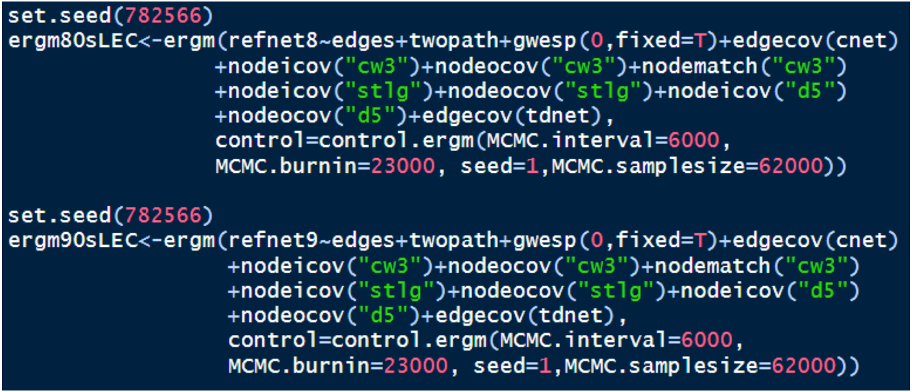
Refugee Flows 1994-1998

# ERGM Specification

- Network Statistics
  - Edges
  - Two-path
  - GWESP
- Edge-Level
  - Contiguity
  - Trade Dependency
  - Civil War Homophily
- Node-Level
  - Civil War (Sender, Receiver)
  - logGDP (Sender, Receiver)
  - Democracy (Sender, Receiver)

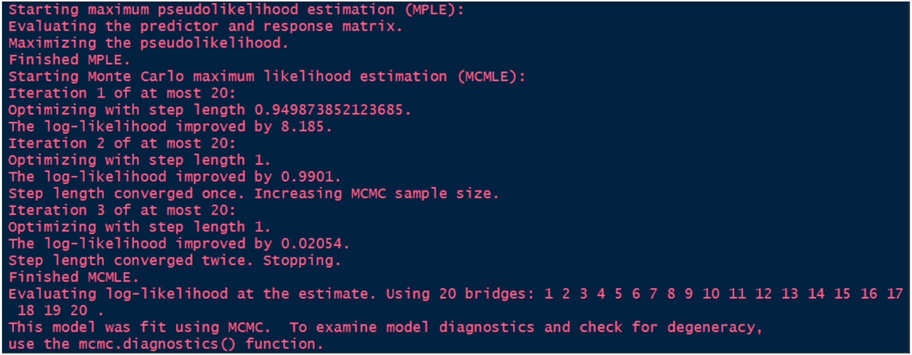
- ERGM MCMC Controls
  - MCMC.interval = 6000
  - MCMC.burnin = 23000
  - MCMC.samplesize = 62000

# ERGM Code



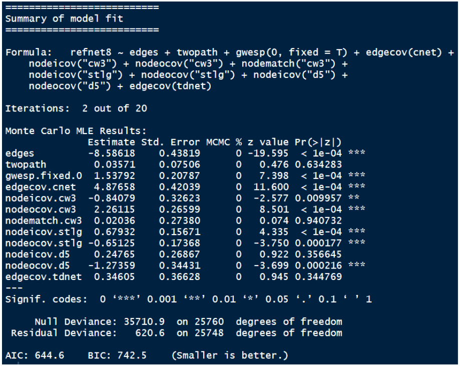
```
set.seed(782566)
ergm80sLEC<-ergm(refnet8~edges+twopath+gwesp(0,fixed=T)+edgecov(cnet)
                +nodeicov("cw3")+nodeocov("cw3")+nodematch("cw3")
                +nodeicov("stlg")+nodeocov("stlg")+nodeicov("d5")
                +nodeocov("d5")+edgecov(tdnet),
                control=control.ergm(MCMC.interval=6000,
                MCMC.burnin=23000, seed=1,MCMC.samplesize=62000))

set.seed(782566)
ergm90sLEC<-ergm(refnet9~edges+twopath+gwesp(0,fixed=T)+edgecov(cnet)
                 +nodeicov("cw3")+nodeocov("cw3")+nodematch("cw3")
                 +nodeicov("stlg")+nodeocov("stlg")+nodeicov("d5")
                 +nodeocov("d5")+edgecov(tdnet),
                 control=control.ergm(MCMC.interval=6000,
                 MCMC.burnin=23000, seed=1,MCMC.samplesize=62000))
```

# Estimation...

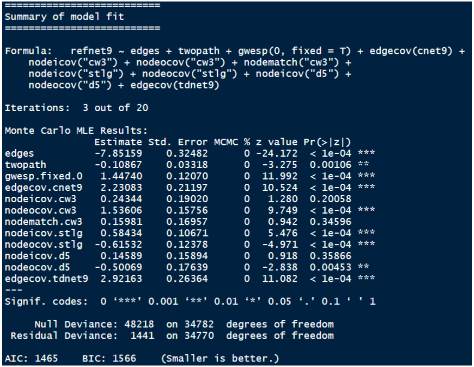


```
Starting maximum pseudolikelihood estimation (MPLE):
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Starting Monte Carlo maximum likelihood estimation (MCMLE):
Iteration 1 of at most 20:
Optimizing with step length 0.949873852123685.
The log-likelihood improved by 8.185.
Iteration 2 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 0.9901.
Step length converged once. Increasing MCMC sample size.
Iteration 3 of at most 20:
Optimizing with step length 1.
The log-likelihood improved by 0.02054.
Step length converged twice. Stopping.
Finished MCMLE.
Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
 18 19 20 .
This model was fit using MCMC.  To examine model diagnostics and check for degeneracy,
use the mcmc.diagnostics() function.
```

# ERGM Results



```
==========================
Summary of model fit
==========================

Formula:   refnet8 ~ edges + twopath + gwesp(0, fixed = T) + edgecov(cnet) +
    nodeicov("cw3") + nodeocov("cw3") + nodematch("cw3") +
    nodeicov("stlg") + nodeocov("stlg") + nodeicov("d5") +
    nodeocov("d5") + edgecov(tdnet)

Iterations:  2 out of 20

Monte Carlo MLE Results:
              Estimate Std. Error MCMC % z value Pr(>|z|)
edges         -8.58618    0.43819      0 -19.595  < 1e-04 ***
twopath        0.03571    0.07506      0   0.476 0.634283
gwesp.fixed.0  1.53792    0.20787      0   7.398  < 1e-04 ***
edgecov.cnet   4.87658    0.42039      0  11.600  < 1e-04 ***
nodeicov.cw3  -0.84079    0.32623      0  -2.577 0.009957 **
nodeocov.cw3   2.26115    0.26599      0   8.501  < 1e-04 ***
nodematch.cw3  0.02036    0.27380      0   0.074 0.940732
nodeicov.stlg  0.67932    0.15671      0   4.335  < 1e-04 ***
nodeocov.stlg -0.65125    0.17368      0  -3.750 0.000177 ***
nodeicov.d5    0.24765    0.26867      0   0.922 0.356645
nodeocov.d5   -1.27359    0.34431      0  -3.699 0.000216 ***
edgecov.tdnet  0.34605    0.36628      0   0.945 0.344769
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 35710.9  on 25760  degrees of freedom
 Residual Deviance:   620.6  on 25748  degrees of freedom

AIC: 644.6    BIC: 742.5    (Smaller is better.)
```



```
==========================
Summary of model fit
==========================

Formula:   refnet9 ~ edges + twopath + gwesp(0, fixed = T) + edgecov(cnet9) +
    nodeicov("cw3") + nodeocov("cw3") + nodematch("cw3") +
    nodeicov("stlg") + nodeocov("stlg") + nodeicov("d5") +
    nodeocov("d5") + edgecov(tdnet9)

Iterations:  3 out of 20

Monte Carlo MLE Results:
               Estimate Std. Error MCMC % z value Pr(>|z|)
edges          -7.85159    0.32482      0 -24.172  < 1e-04 ***
twopath        -0.10867    0.03318      0  -3.275  0.00106 **
gwesp.fixed.0   1.44740    0.12070      0  11.992  < 1e-04 ***
edgecov.cnet9   2.23083    0.21197      0  10.524  < 1e-04 ***
nodeicov.cw3    0.24344    0.19020      0   1.280  0.20058
nodeocov.cw3    1.53606    0.15756      0   9.749  < 1e-04 ***
nodematch.cw3   0.15981    0.16957      0   0.942  0.34596
nodeicov.stlg   0.58434    0.10671      0   5.476  < 1e-04 ***
nodeocov.stlg  -0.61532    0.12378      0  -4.971  < 1e-04 ***
nodeicov.d5     0.14589    0.15894      0   0.918  0.35866
nodeocov.d5    -0.50069    0.17639      0  -2.838  0.00453 **
edgecov.tdnet9  2.92163    0.26364      0  11.082  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 48218  on 34782  degrees of freedom
 Residual Deviance:  1441  on 34770  degrees of freedom

AIC: 1465    BIC: 1566    (Smaller is better.)
```

# MCMC Diagnostics I

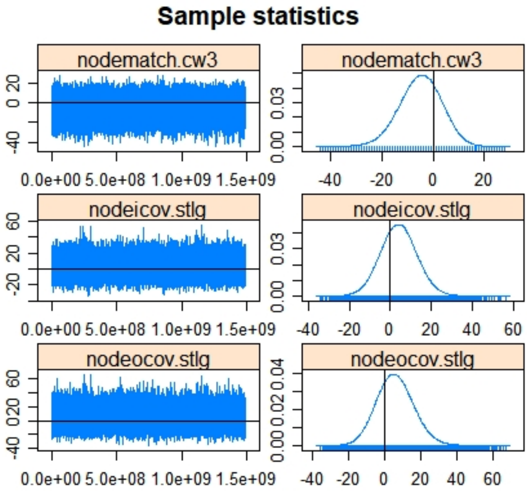
Things to look at (mcmc.diagnostics(model)):

- Trace plots- want random mixing around zero (stationarity).
- Autocorrelation sample statistics- correlation between sample statistics at different parts of chain; want small values.
- Geweke diagnostics- measures convergence by comparing mean statistics at different parts of the chain; want large p-values

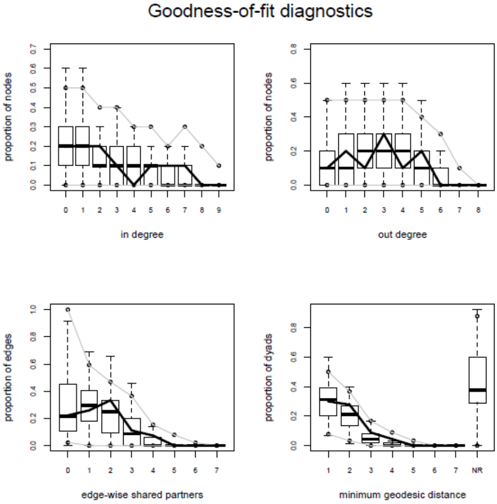
If:

- Trace plots or Geweke stats are bad, increase sample size and/or burnins
- If autocorrelation is bad (chain is mixing slowly), increase intervals

# MCMC Diagnostics II



# GOF



Goodness-of-fit diagnostics
proportion of nodes
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0 1 2 3 4 5 6 7 8 9
in degree
proportion of nodes
0.0
0.2
0.4
0.6
0.8
0 1 2 3 4 5 6 7 8
out degree
proportion of edges
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7
edge-wise shared partners
proportion of dyads
0.0
0.2
0.4
0.6
0.8
1 2 3 4 5 6 7 NR
minimum geodesic distance

# ERGM Summary

Advantages

- Tremendously flexible and inclusive: covariate, edge, network-level effects
- Does not require conditional independence assumption
- Use when the edges/relationships/ties are the outcome of interest

Disadvantages

- Too flexible?
- Degeneracy?

# Extensions

- Generalized ERGMs: extended to valued edges
- Temporal ERGMs (dynamics)
- Spatial networks
- Latent Space Models...

# Software Things

Packages

- sna
- igraph
- statnet suite (network, ergm, btergm, etc.)
- Many others, especially for plotting (GERGM, GGally, ggnet2, RSiena, keyplayer...)

**Note**: sna and igraph have many of the same command names, but implement them differently so be careful. Also, sna and igraph are not compatible... but yay intergraph!

# Stuff to Read

- Wasserman, Stanley and Katherine Faust. 1997. *Social Network Analysis*. New York: Cambridge University Press.
- Lusher, Dean, Johan Koskinen, and Garry Robins. 2012. *Exponential Random Graph Models for Social Networks*. New York: Cambridge University Press.
- Snijders, Tom A.B., Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. 2006. "New Specifications for Exponential Random Graph Models." *Sociological Methodology* 36(1): 99-153.
- Cranmer, Skyler J., Philip Leifeld, Scott D. McClurg, and Meredith Rolfe. 2017. "Navigating the Range of Statistical Tools for Inferential Network Analysis." *American Journal of Political Science* 61(1): 237-251.
- Leifeld, Philip, Skyler J. Cranmer, and Bruce A. Desmarais. 2018. "Temporal Exponential Random Graph Models with btergm." *Journal of Statistical Software* 83(6).
- Minhas, Shahryar, Peter D. Hoff, and Michael D. Ward. 2019. "Inferential Approaches for Network Analysis: AMEN for Latent Factor Models." *Political Analysis* 27(2): 208-222.