

STATNET WEB

THE EASY WAY TO LEARN (OR TEACH) STATISTICAL
MODELING OF NETWORK DATA WITH ERGMS

SUNBELT

June 26, 2018

Martina Morris, Ph.D.

Skye Bender-deMoll



What this workshop will cover

- Basic intro to statistical modeling for networks
 - Descriptive methods/EDA
 - Exponential-family Random Graph Models (ERGMs)
 - A bit of the underlying theory
 - Cross sectional network modeling
- Software: statnetWeb
 - A browser-based GUI for the statnet packages

Intro to statnet software family

Core statnet packages

For descriptive and statistical network analysis

network, sna, ergm (cross-sectional nets)
networkDynamic, tsna, tergm (temporal nets)



statnetWeb

Rshiny app for statnet with user-friendly
GUI for descriptive and statistical analysis

Currently limited to cross-sectional
methods

All of our software is

- Open source
- Published on CRAN
- Supported by online training materials
- The functionality is continuing to be improved and extended with new methods/packages
- And others are writing packages that extend our software
 - E.g., Xergm, Bergm, Hergm

Motivation

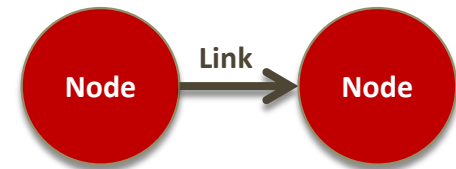
Statistical methods for network analysis are new-ish

- Older stuff: permutation and CUG tests
 - These are relatively easy to learn and use
- Modern methods: Exponential-family Random Graph Models
 - These are more complex
 - And the software can be intimidating

statnetWeb makes it easier to get started

Network basics (quick review, then we'll get started)

- **Node:** the entity of interest
 - often, nodes represent people; also called actors or vertices
- **Link:** the relationship of interest
 - also called a tie, an edge, or a line
- **Network:** a set of actors and the relations among them
 - Also called a graph



Types of nodes

■ Individual units

- Humans
- Animals
- Airports
- Computers
- Genes

■ Collectivities

- Countries, cities
- Families
- Species
- Organs, Sensory systems

In social networks, a focal node is called “ego”, and the nodes linked to this focal node are “alters”

Types of links (these are just examples)

■ Social

- **Affective** (like/dislike, trust/do not trust)
- **Kinship / social role** (mother of, brother of, boss of)
- **Exchange** (advice seeking, sexual intercourse, trade)
- **Cognitive** (knows/does not know)
- **Affiliation** (belongs to, is a member of)

■ Physical

- **Road**
- **Flight path**
- **Wire / Wireless**

■ Regulatory (as in gene expression)

Link properties

- Directed (e.g., likes)

- Mutual



- Asymmetric



- Null



Nodes are now classified as senders and/or receivers

- A directed graph is also called a di-graph

- A directed edge is also called an arc

- Undirected (e.g., talks with)



- Binary (0,1 on or off only)

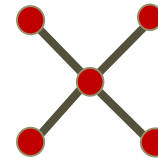
- Signed and/or Valued (... -2, -1, 0, 1, 2 ...)

Configurations

Dyads

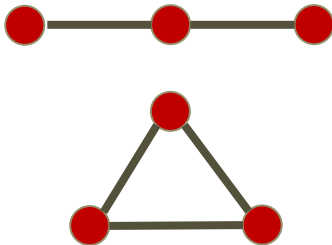


Stars

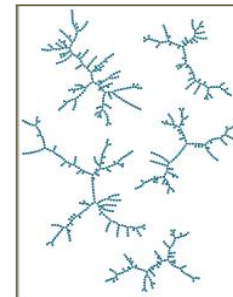


“4-star”

Triples & Triangles



Components



Any collection of nodes and links can be defined as a configuration

Types of networks

- Simplest form: 1-mode, undirected, binary ties, single relation
- 2-mode (aka *Bipartite*)
 - Two different types of nodes
 - Ties only allowed between groups

Examples: Online network groups and persons (an Affiliation network)

Heterosexual sex network
- Multiplex
 - More than one type of link possible

Example: Neighbor and running partner

Representing network data

■ Sociomatrix

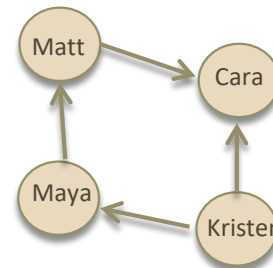
- aka adjacency matrix
- simple but inefficient for large sparse networks (order n^2)

	Matt	Cara	Kristen	Maya
Matt	0	1	0	0
Cara	1	0	0	1
Kristen	0	1	0	1
Maya	1	0	0	0

■ Edgelist

Matt	Cara
Cara	Matt
Cara	Maya
Kristen	Cara
Kristen	Maya
Maya	Matt

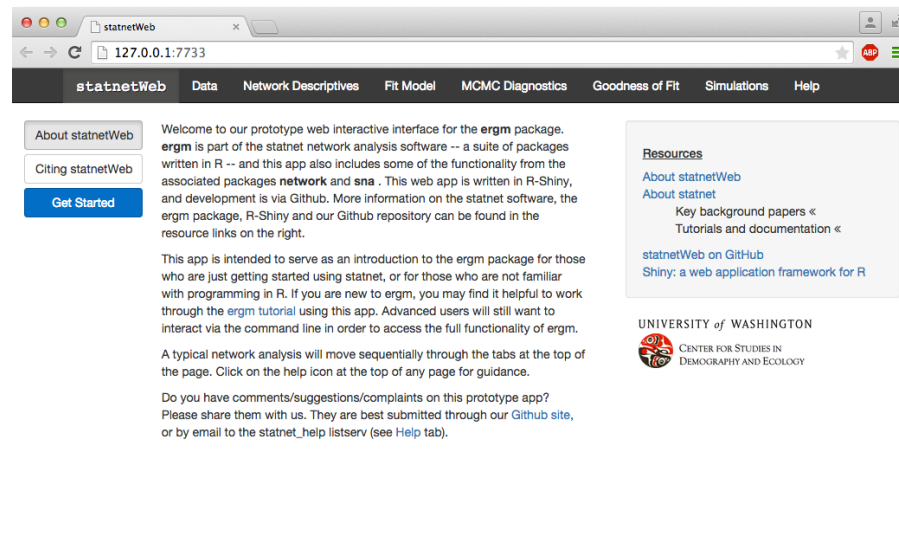
■ Graph



Intro to statnetWeb

statnetWeb is a GUI for the statnet suite of R packages

- Runs in a web browser
- Or in a pop-up window from R



Getting started: R and Rstudio

- Install R

<https://cran.r-project.org/>

R is the software that will run the statnetWeb package

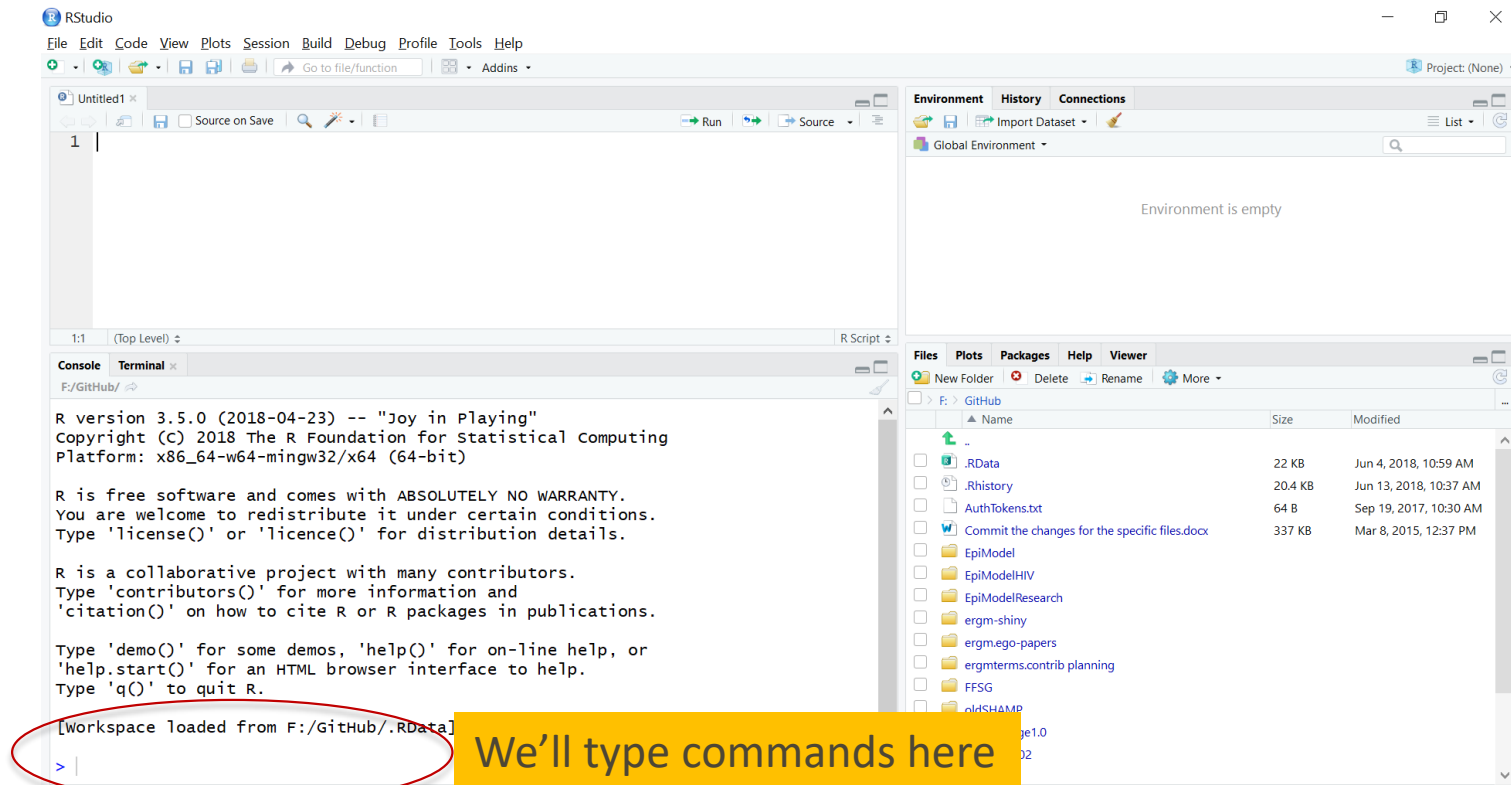
- Install Rstudio

<https://www.rstudio.com/products/rstudio/download/>

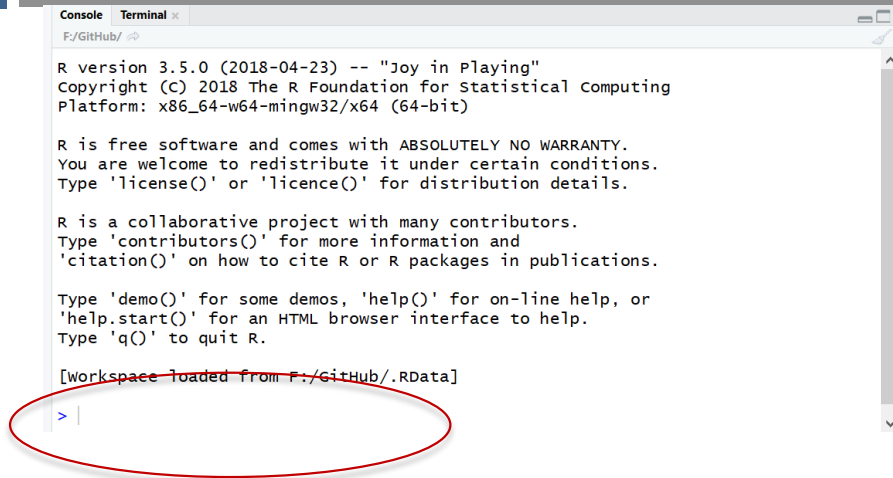
Rstudio is an interface that makes it easier to work with R

Getting started: Installing statnetWeb

■ First: Launch Rstudio



Getting started: Installing statnetWeb



```
Console Terminal x
F:/GitHub/

R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

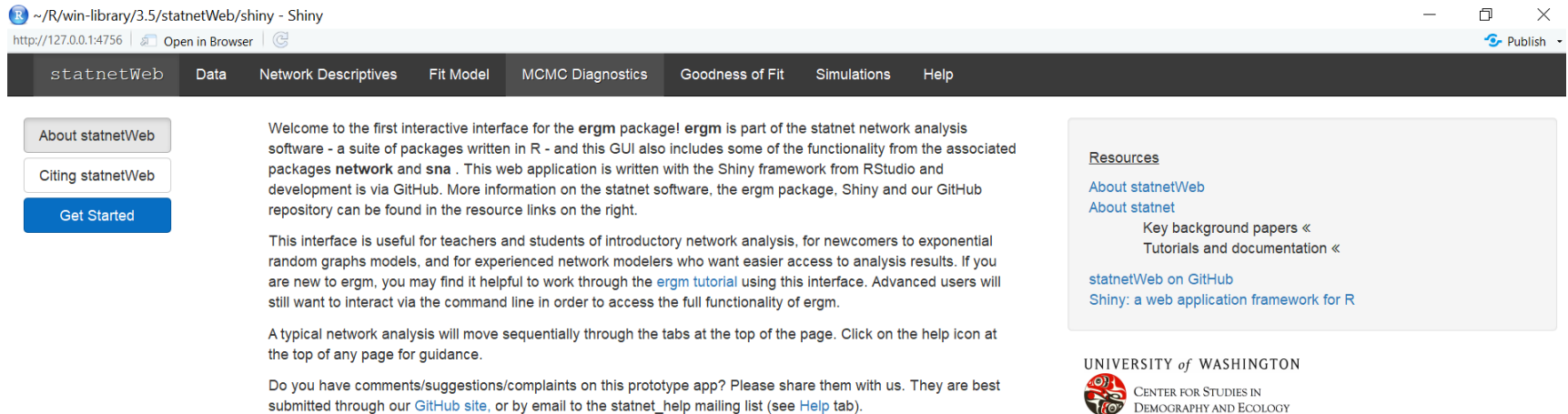
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from F:/GitHub/.RData]
> |
```

Then type the following commands at the > prompt

- `install.packages("devtools")`
- `devtools::install_github('statnet/statnetWeb')`
- `library(statnetWeb)`
- `run_sw()`

You should get a pop-up window



The screenshot shows a web browser window displaying the statnetWeb Shiny application. The browser's address bar shows the URL `http://127.0.0.1:4756`. The application has a dark navigation bar with tabs: `statnetWeb`, `Data`, `Network Descriptives`, `Fit Model`, `MCMC Diagnostics`, `Goodness of Fit`, `Simulations`, and `Help`. On the left side, there are three buttons: `About statnetWeb`, `Citing statnetWeb`, and `Get Started`. The main content area contains a welcome message, a description of the interface's purpose, and a list of resources. The resources section includes links to `About statnetWeb`, `About statnet`, `Key background papers`, `Tutorials and documentation`, `statnetWeb on GitHub`, and `Shiny: a web application framework for R`. The footer of the application displays the University of Washington logo and the text `CENTER FOR STUDIES IN DEMOGRAPHY AND ECOLOGY`.

~/R/win-library/3.5/statnetWeb/shiny - Shiny
http://127.0.0.1:4756 | Open in Browser | Publish

statnetWeb Data Network Descriptives Fit Model MCMC Diagnostics Goodness of Fit Simulations Help

About statnetWeb
Citing statnetWeb
Get Started

Welcome to the first interactive interface for the **ergm** package! **ergm** is part of the statnet network analysis software - a suite of packages written in R - and this GUI also includes some of the functionality from the associated packages **network** and **sna**. This web application is written with the Shiny framework from RStudio and development is via GitHub. More information on the statnet software, the ergm package, Shiny and our GitHub repository can be found in the resource links on the right.

This interface is useful for teachers and students of introductory network analysis, for newcomers to exponential random graphs models, and for experienced network modelers who want easier access to analysis results. If you are new to ergm, you may find it helpful to work through the [ergm tutorial](#) using this interface. Advanced users will still want to interact via the command line in order to access the full functionality of ergm.

A typical network analysis will move sequentially through the tabs at the top of the page. Click on the help icon at the top of any page for guidance.

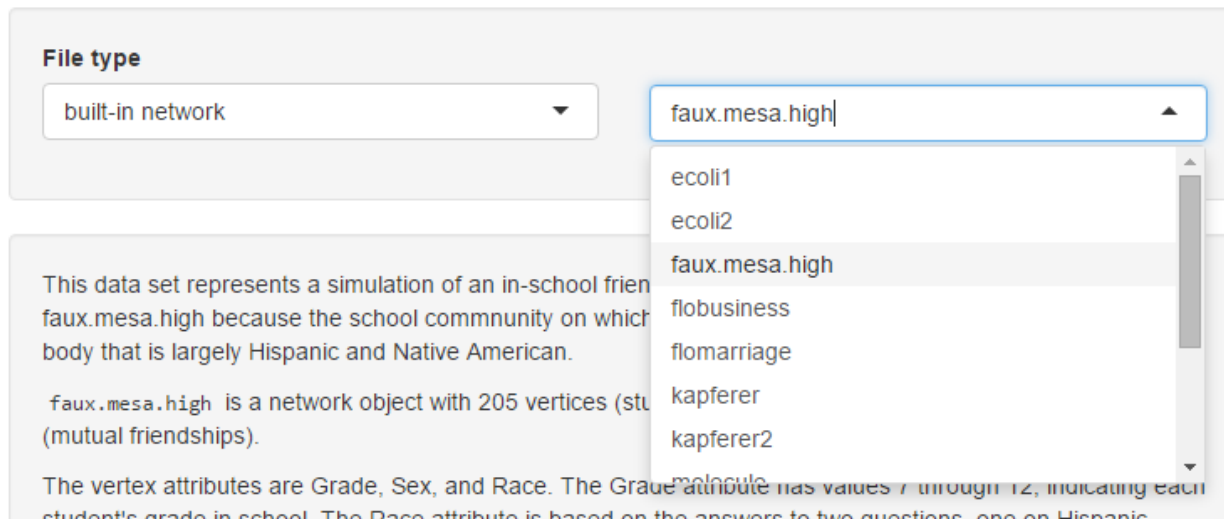
Do you have comments/suggestions/complaints on this prototype app? Please share them with us. They are best submitted through our [GitHub site](#), or by email to the statnet_help mailing list (see [Help tab](#)).

Resources
[About statnetWeb](#)
[About statnet](#)
Key background papers «
Tutorials and documentation «
[statnetWeb on GitHub](#)
[Shiny: a web application framework for R](#)

UNIVERSITY of WASHINGTON
CENTER FOR STUDIES IN
DEMOGRAPHY AND ECOLOGY

Data: for loading your network data

- Can upload your own
- Or use one of the built-in datasets (we'll do this)
 - Load the faux.mesa.high network

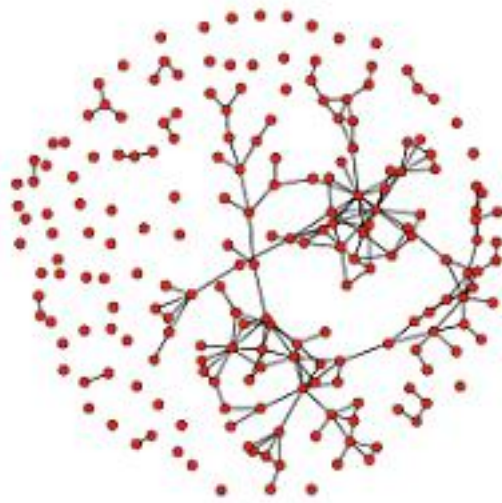


The screenshot shows a web interface for loading network data. On the left, a 'File type' dropdown menu is set to 'built-in network'. To its right, a search dropdown menu is open, displaying a list of built-in datasets. The dataset 'faux.mesa.high' is highlighted in the list. Below the search dropdown, there is a text area providing information about the selected dataset: 'This data set represents a simulation of an in-school friend network. faux.mesa.high because the school community on which it is based is largely Hispanic and Native American. faux.mesa.high is a network object with 205 vertices (students) and 1,200 edges (mutual friendships). The vertex attributes are Grade, Sex, and Race. The Grade attribute has values 7 through 12, indicating each student's grade in school. The Race attribute is based on the answers to two questions: one on Hispanic or Latino ethnicity and one on race or ethnicity.

Descriptive Statistics

- This tab in statnetWeb can be used to explore your data

Plots



Tables

Network Plot

Attributes

Degree Distribution

```
[1] Grade  
  
 7  8  9 10 11 12  
62 40 42 25 24 12  
[1] Race  
  
Black  Hisp NatAm Other White  
   6   109   68    4    18  
[1] Sex  
  
   F   M  
 99 106  
[1] Missing  
  
FALSE  
 205
```

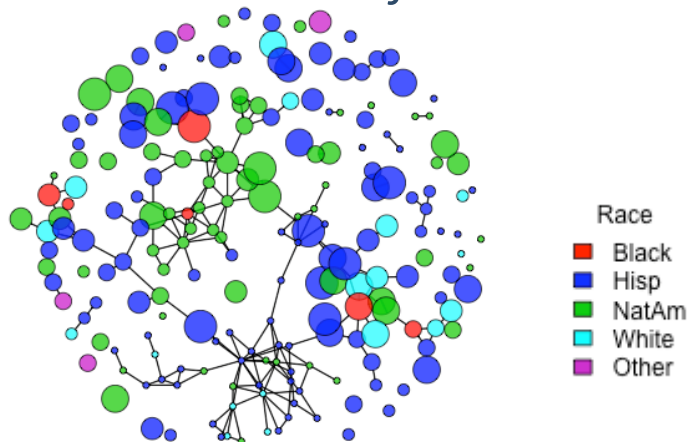
Levels of measurement in networks

As we look at ways of describing network data, keep in mind the different levels of measurement

- Node level: *attributes of each individual node*
 - Examples: age, sex, infection state, degree
- Dyad level: *attributes of pairs or edges*
 - Examples: type of relationship, duration
- Component level: *subgraph attributes and distributions*
 - Examples: size, density, degree and geodesic distributions ...
- Network level: *overall structural attributes and distributions*
 - Examples: density, degree, geodesic distribution ...

Nodal Attributes

- Nodes can have attributes like age, race, sex, etc.
- *Explore:*
 - *Click on a node in the network plot to see its name and attributes*
 - *Double click to highlight a node's neighbors*
 - *Color-code or size nodes with menu options on the right*
 - *Sort or search attributes in the interactive table*
 - *What can you say about the structure of the network after editing the plot?*

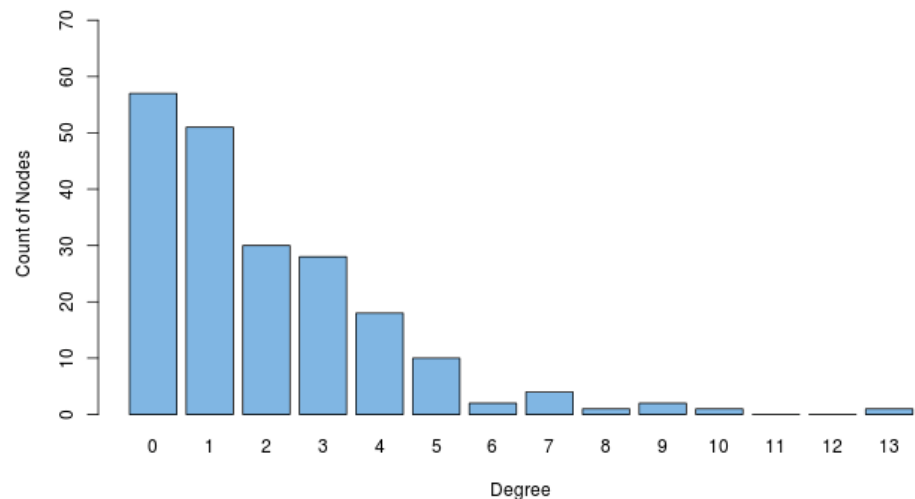


Measuring degree

- Node level: The number of edges adjacent to a node
 - Every node has a degree: $\deg(i)$
 - For di-graphs, in- and out- degrees: $\text{ideg}(i)$ and $\text{odeg}(i)$
 - Indegree: the number of arcs that terminate at n_i
 - Outdegree: the number of arcs that originate from n_i
- Network level: The degree distribution
 - Well-known parametric degree distributions
 - Uniform, Binomial, Poisson, Power-law
 - An empirical degree distribution may or may not resemble any of these

Degree distribution

- The degree distribution is a basic structural property
- To view it in statnetWeb:

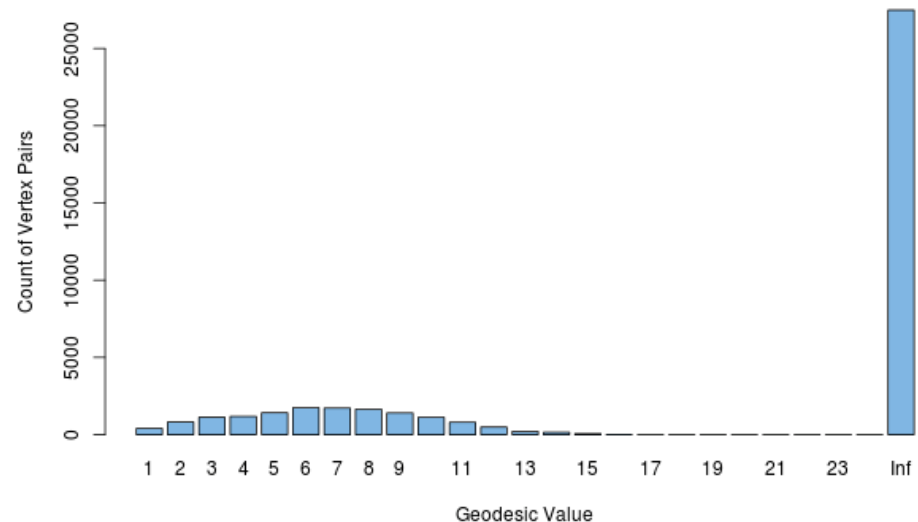


Connectivity measures: Geodesic

- Nodes are **reachable** if there is a path between them.
- A **geodesic** is the shortest path between two nodes
 - Two nodes have an infinite geodesic distance if they are unreachable

Geodesic distribution

- The geodesic distribution is another basic structural property of a network
- To view it in statnetWeb:



The last bar represents the node pairs with infinite geodesic distance

Description vs. Inference

- So far we have been using descriptive statistics to explore our data
- Next, we might want to compare these statistics to what we would expect by chance
 - What do we mean “by chance”?
 - Can we use inferential statistics to draw more general conclusions?
- Need to define a reference distribution to act as the “null model”
- Two common null model distributions
 - Conditional Uniform Graph (CUG): Same density as the observed net
 - Bernoulli Random Graph (BRG): Same tie probability as the observed net

Statistical Model: Testing

- Suppose kids have a tendency to become friends with their friends' friends
 - And this is the only generative process occurring.
- Presumably, this would mean that you would observe more triangles than expected by chance in the graph.
 - How would you test this in an empirical dataset?

A basic statistical test approach

- Begin by counting the # triangles in your network
 - Say this is “ T ”, your test statistic
- Then determine the probability of observing T or more triangles in this network ...
- And see if it is less than 5%

But how would you determine that probability?

What is this probability distribution?

Start with a network this size (size = # nodes)

- Enumerate all possible networks for a fixed number of nodes,
- Count the number of triangles in each network, and
- Construct the frequency distribution of the counts.

This is a
“permutation test”

for 4 nodes:

of dyads is $4*3/2 = 6$

of possible networks = $2^6 = 64$

for 10 nodes:

of dyads is $10*9/2 = 45$

of possible networks = $2^{45} \approx 35$ trillion

for 20 nodes:

of dyads is $20*19/2 = 190$

of possible networks = $2^{190} \approx 10^{57}$

A conditional null probability distribution

Condition on ***both the size and density*** of the network
This is the Conditional Uniform Graph test (CUG)

- Enumerate all possible networks for a fixed number of nodes and links,
- Count the number of triangles in each network, and
- Construct the frequency distribution of the counts.

Better (in terms of reducing the sample space)

but still a lot of graphs...

we use a sample of about 50 of these for our CUG tests

Another conditional null probability distribution

Condition on *the probability* of a link

The Bernoulli Random Graph model (BRG)

- Simulate networks by randomly selecting a dyad, and using a coin flip to update the tie status
- Count the number of triangles after each 1000 updates
- Construct the frequency distribution of the counts.

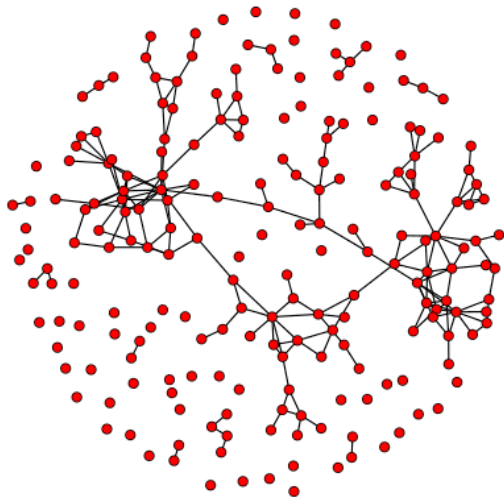
This is a stochastic approach, in comparison to the CUG.

- It doesn't enumerate the whole space, it just randomly wanders around it

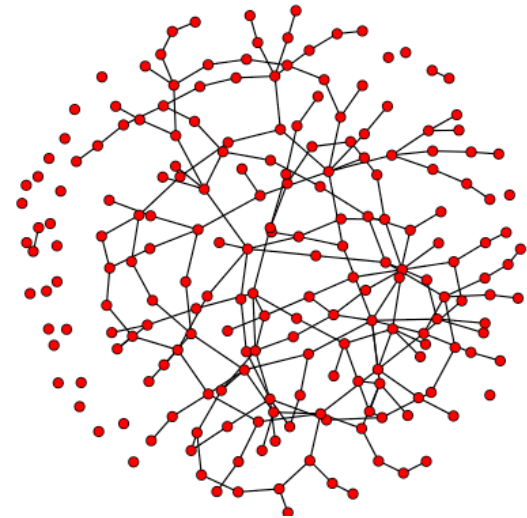
we use a sample of 50 for our BRG tests

Example of a BRG simulation

faux.mesa.high network



Simple random graph with the same tie prob

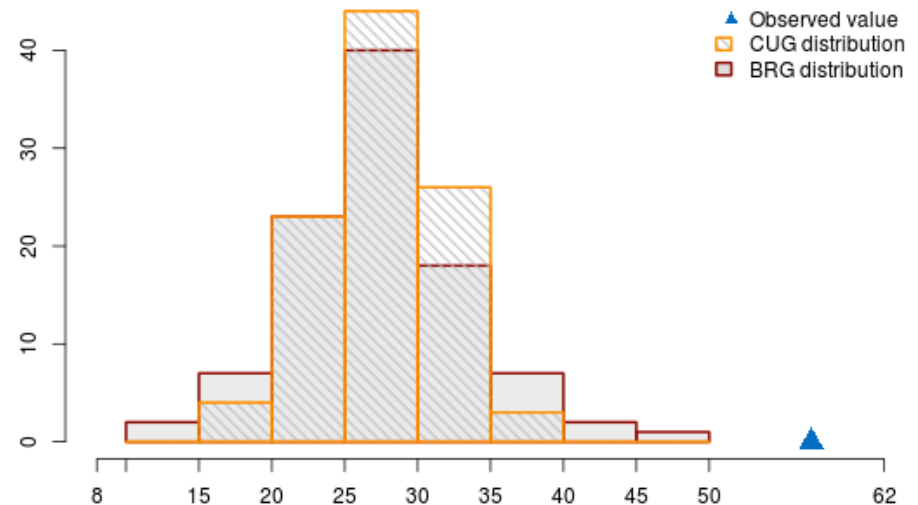


Using null models

- Select a summary measure for the observed data
- Compare it to the distribution simulated from a null model
- In statnetWeb:
 - We can conduct CUG and BRG tests for network summary measures
 - We can plot overlays on degree and geodesic distributions

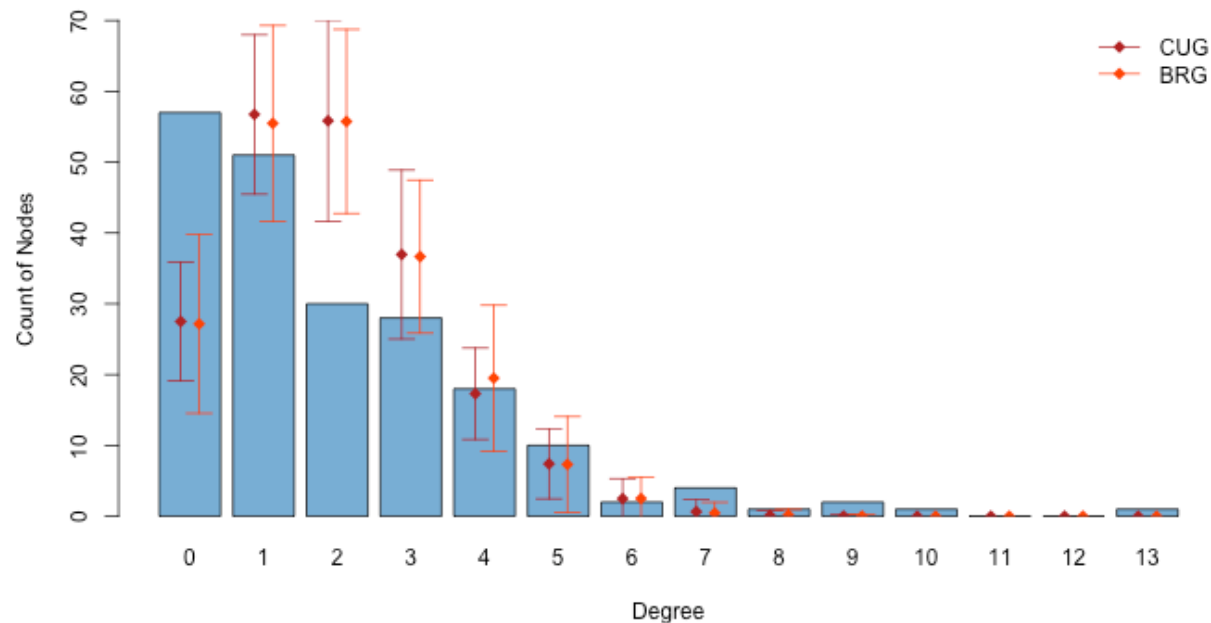
Conditional uniform graph tests

Compare the number of isolates in faux.mesa.high to what we would expect by chance



Degree distribution comparisons

In the degree distribution, add overlays for each null model



Mean and 95% confidence intervals from 50 random draws are plotted

Let's look at triangles

- Are there more triangles than expected in faux.mesa.high?
- In statnetWeb, go to the Conditional Uniform Graph Tests

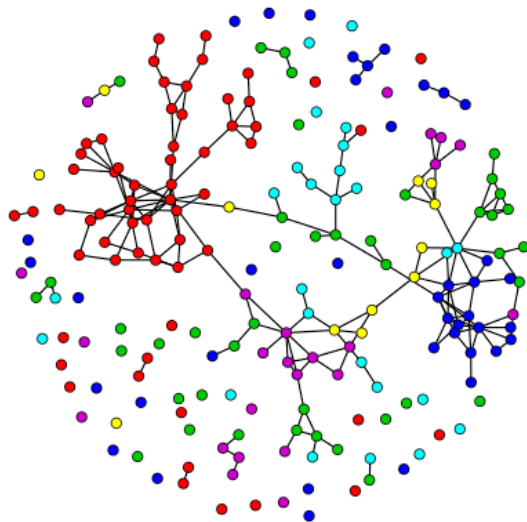


- Choose the triangle term and run 100 simulations to see how our network compares to random graphs
 - “CUG” and “BRG” versions

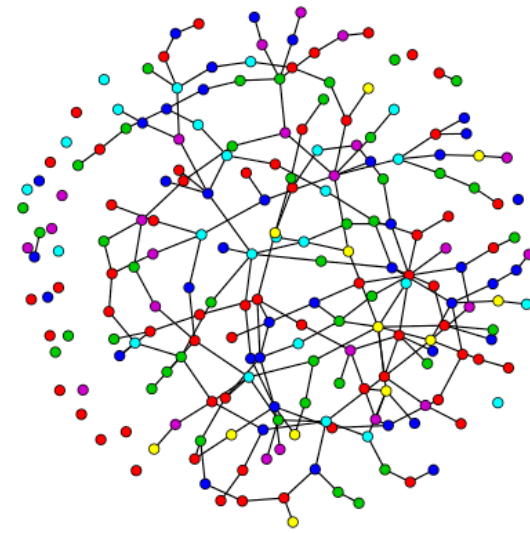
Limitations of these simple tests

- Why are there so many more triangles?
- What do you see when color-coding the nodes by their attributes?

faux.mesa.high network



Simple random graph with the same tie prob

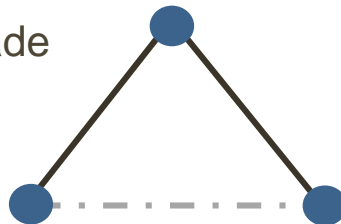


Friend of a friend, or birds of a feather?

(At least) two theories about the process that generates triangles:

1. Homophily: People tend to choose friends who are like them, in grade, race, etc. (“birds of a feather”), triad closure is a by-product
2. Transitivity: People who have friends in common tend to become friends (“friend of a friend”), closure is the key process

So, for three actors in the same grade



A cycle-closing tie may form because of transitivity but also because of homophily

Transitivity and homophily are confounded

But not completely. Any tie may be classified by whether it is:

<u>Triangle forming:</u>		
<u>Within Grade:</u>	<i>Yes</i>	<i>No</i>
<i>Yes</i>	Both	Homophily
<i>No</i>	Transitivity	<i>Neither</i>

The cells show which processes can influence that tie.

This suggests we should be able to disentangle the two processes statistically, by looking at the relative frequency in each cell

Statistical Model: Basic idea

- We want to model the probability of a tie as a function of:
 - Nodal attributes (that influence degree and mixing)
 - The propensity for certain “configurations” (like triangles)
- The tie-status of dyads may be dependent
 - Nodal attribute effects do not induce dyad dependence (homophily)
 - But triad closure effects do
- So we model the joint distribution directly

Exponential Random Graph Model (ERGM)

Probability of observing a graph (set of relationships) y on a fixed set of nodes:

$$P(Y = y | \theta) = \frac{\exp(\theta' g(y))}{k(\theta)}$$

where: $g(y)$ = vector of network statistics

θ = vector of model parameters

$k(\theta)$ = numerator summed over all possible networks on node set y

- Exponential family model
- Well understood statistical properties Besag (1974), Frank (1986)
- Very general and flexible

Exponential Random Graph Model (ERGM)

Probability of observing a graph (set of relationships) y on a fixed set of nodes:

$$P(Y = y | \theta) = \frac{\exp(\theta' g(y))}{k(\theta)}$$

If you're not familiar with this kind of compact vector notation, the numerator is just:

$$\exp(\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_3 x_3)$$

Kind of like a linear model, but a bit different (watch out for this later)

The conditional probability of a tie

$$P(Y = y | \theta) = \frac{\exp(\theta' g(y))}{k(\theta)} \quad \text{can be re-expressed as}$$

$$\begin{aligned} \text{logit}(P(Y_{ij} = 1 | \text{rest of the graph})) &= \log \left(\frac{P(Y_{ij} = 1 | \text{rest of the graph})}{P(Y_{ij} = 0 | \text{rest of the graph})} \right) \\ &= \theta' \partial(g(y)) \end{aligned}$$

The “change statistic”

the change in $g(y)$ when Y_{ij}
is toggled from 0 to 1

ERGM specification: $\theta' g(y)$

The $g(y)$ terms in the model represent “network statistics”

■ These are counts of network configurations, for example:

1. Edges: $\sum y_{ij}$
2. Within-group ties: $\sum y_{ij} I(i \in C, j \in C)$
3. 2-stars: $\sum y_{ij} y_{ik}$
4. 3-cycles: $\sum y_{ij} y_{ik} y_{jk}$

■ A key distinction in the types of terms:

- Dyad independent (1 & 2 are examples)
- Dyad dependent (3 & 4 are examples)

ERGM specification: $\theta' g(y)$

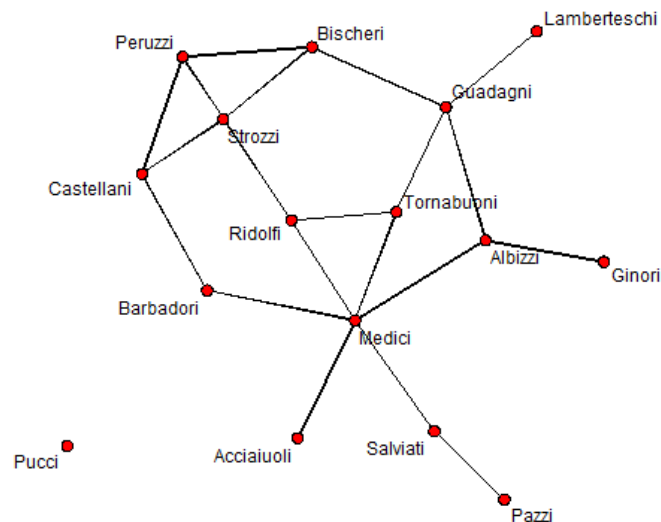
Model specification involves:

1. Choosing the set of network statistics $g(y)$
 - From minimal : # of edges
 - To saturated: one term for every dyad in the network
 - statnetWeb allows you to choose from the list of terms and retrieve documentation for each one
2. Choosing “homogeneity constraints” on θ . For example, with edges:
 - all homogeneous
 - group specific (e.g., sex or age specific)
 - dyad specific

Ok, back to statnetWeb

Flomarriage: Bernoulli Model

- Load the flomarriage network
 - Network of marriage ties between families in Renaissance Florence



- On the Fit Model page, look up the documentation on the edges term

Flomarriage: Bernoulli Model

- Add edges to the ergm formula

Step 1

The screenshot shows the Flomarriage web interface. On the left, under 'Network:', there is a text box containing 'flomarriage'. On the right, under 'ERGM terms:', there is a text box containing 'edges'. Below the 'ERGM terms:' box, there are two buttons: 'Add Term(s)' (highlighted in blue) and 'Reset Formula'.

- Fit the model

Step 2

The screenshot shows the Flomarriage web interface. On the left, under 'Current ergm formula:', there is a text box containing 'edges'. Below this, under 'Summary statistics:', there is a text box containing 'edges' and '203'. At the bottom, there are three buttons: 'Fit Model' (highlighted in blue), 'Save Current Model (0/5)', and 'Clear All Models'.

- Assumes homogeneous edge probability (the BRG)
 - Every tie is equally likely
 - Not a very interesting model

Flomarriage: Bernoulli Model

- How to interpret coefficients? The log-odds of any tie existing is

$$= -1.609 \times \text{change in \# ties}$$

$$= -1.609 \times 1$$

- Corresponding probability:

$$= \frac{\exp(-1.609)}{1 + \exp(-1.609)}$$

$$= 0.1667$$

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

Formula:   nw() ~ edges
<environment: 0x8cb6228>

Iterations: 5 out of 20

Monte Carlo MLE Results:
      Estimate Std. Error MCMC % p-value
edges  -1.6094    0.2449      0 <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      Null Deviance: 166.4 on 120 degrees of freedom
      Residual Deviance: 108.1 on 119 degrees of freedom

AIC: 110.1   BIC: 112.9   (Smaller is better.)
```

Flomarriage: Triad Formation

- The “triangle” term is one measure of clustering
- Read the documentation for the triangle term
- Fit the model edges + triangle
 - Hint: you can just add the triangle term if edges is already in your formula
 - Then click Fit Model
- Triangle is a dyad dependent term, so the estimation algorithm changes to MCMC

```
Monte Carlo MLE Results:
      Estimate Std. Error MCMC % p-value
edges      -1.6770     0.3508      0 <1e-04 ***
triangle    0.1568     0.5854      0  0.789
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Flomarriage: Triad Formation

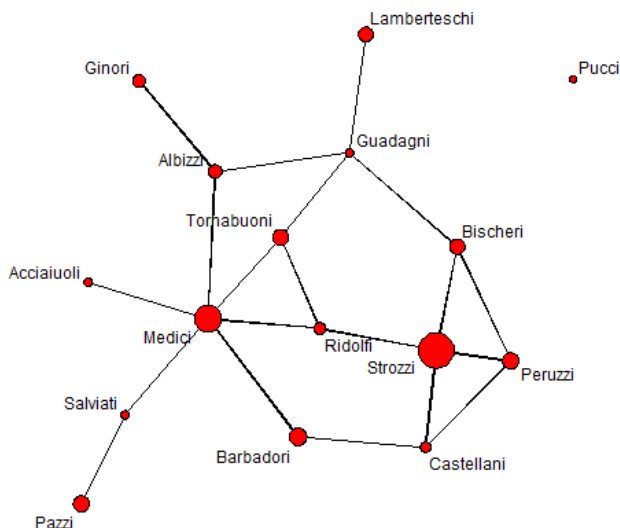
```
Monte Carlo MLE Results:
      Estimate Std. Error MCMC % p-value
edges      -1.6770     0.3508      0 <1e-04 ***
triangle    0.1568     0.5854      0  0.789
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note, not significant

- Now how to interpret results?
 - Conditional log-odds of two actors having a tie:
($-1.68 \times \text{change in the \# of ties}$) + ($0.16 \times \text{change in \# of triangles}$)
how many triangles can one tie change?
 - For a tie that will create zero triangles $-1.68 + 0 = -1.68$
 - One triangle $-1.68 + 0.16 = -1.52$
 - Two triangles $(-1.68 \times 1) + (0.16 \times 2) = -1.36$

Flomarriage: Nodal covariates

flomarriage sized by wealth



- What do you notice?
- We can test whether edge probabilities are a function of wealth
- This is a quantitative nodal attribute, so we use the ergm term “nodecov”

Flomarriage: Nodal covariates

- Reset the ergm formula and fit the following model:

Network:

ERGM terms:

- There is a significant positive wealth effect on the odds of a tie

```
Monte Carlo MLE Results:
              Estimate Std. Error MCMC % p-value
edges          -2.594929   0.536056     0 <1e-04 ***
nodecov.wealth   0.010546   0.004674     0  0.0259 *
```

- What does the positive coefficient mean?
 - Not that there is homophily by wealth
 - Just that wealthy nodes have more ties
 - Note that the wealth effect operates on both nodes in a dyad.

Flomarriage: Nodal covariates

- The conditional log-odds of a tie between two actors is:
 $-2.59 \times \text{change in \# ties} + 0.01 \times \text{wealth of node 1} + 0.01 \times \text{wealth of node 2}$
 - For a tie between two nodes with minimum wealth
 $-2.59 + 0.01 \times (3 + 3) = -2.53$
 - For a tie between two nodes with maximum wealth
 $-2.59 + 0.01 \times (146 + 146) = 0.33$
 - For a tie between nodes with maximum and minimum wealth
 $-2.59 + 0.01 \times (146 + 3) = -1.1$
- Note: To specify homophily on wealth, you would use the ergm-term ***absdiff***

Model Degeneracy

- Models with dyad dependent terms can behave differently than we expect
 - They look simple
 - But they represent effects that cascade through a network via a chain of dependence (**this is the “watch out” from earlier**)
- Homogeneous triangle and k-star terms turn out to be some of the worst offenders

Model Degeneracy

- Definition
 - When a model places almost all probability on a small number of uninteresting graphs
- Most common “uninteresting” graphs:
 - Complete (all links exist)
 - Empty
- Model degeneracy = misspecification

Model Degeneracy

- Switch back to the faux.mesa.high network
- Fit a model where the formula is edges + triangle
 - What happens?

```
Error: Number of edges in a simulated network exceeds that in the observed by a factor of more than 20. This is a strong indicator of model degeneracy or a very poor starting parameter configuration. If you are reasonably certain that neither of these is the case, increase the MCMLE.density.guard control.ergm() parameter.
```

- Trying to fit this model, the algorithm heads off into networks that are *much* more dense than the observed network.
- What does this mean? That this model would not have produced this network, for any combination of parameter estimates for the two terms
 - i.e., this is a model misspecification problem

Degeneracy Plot (for the 2 star model)

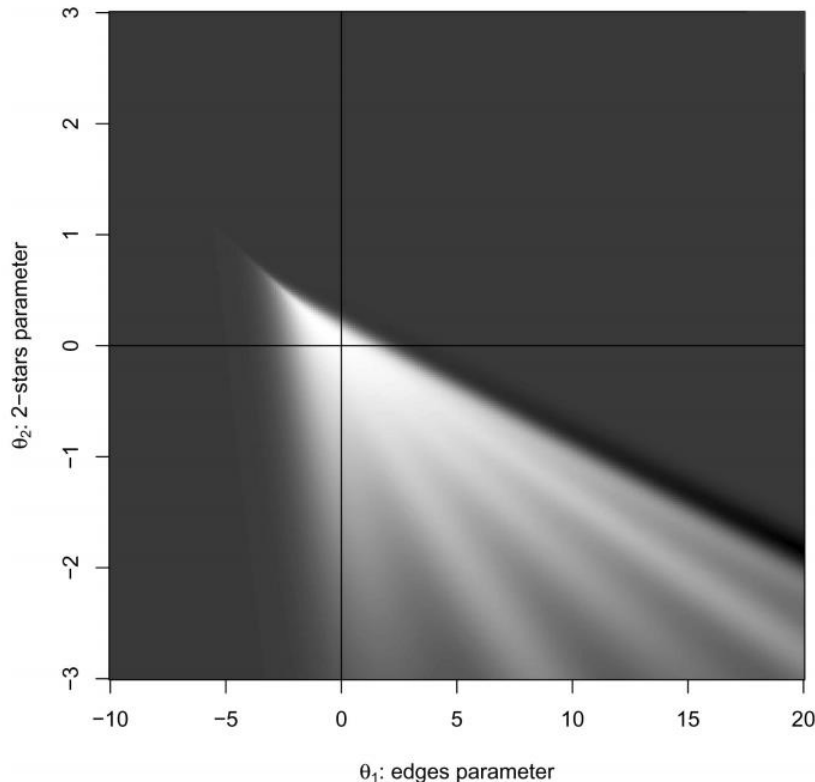


Figure 3: Cumulative Degeneracy Probabilities for graphs with 7 actors.

From Mark Handcock's 2003 tech report:

<https://www.csss.washington.edu/Papers/2003/wp39.pdf>

- Only the white area has networks with some interesting variation
- The dark areas are complete graphs, or empty graphs (+/- 1 or 2 edges)
- This model does not produce many useful networks

Solution: better network statistics

- Old statistic: # of triangles in the graph

$$g(y) = \sum y_{ij}y_{jk}y_{ki}$$

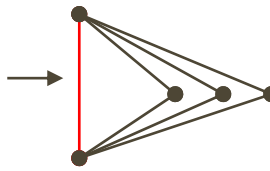
Here, every additional 3-cycle has the same impact, θ

- New statistic: GWESP

$$g(y) = e^{\alpha} \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} sp_i$$

- ***geometrically weighted edge-wise shared partners***
- Sets declining marginal returns for each additional 3-cycle involving the same edge
- The parameter that specifies the rate of decline in marginal returns is α
- The smaller the α , the more rapid the decline

Solution: better network statistics



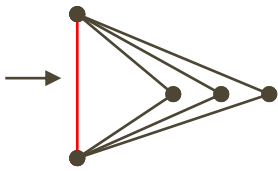
$$gwesp = e^{\alpha} \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} sp_i \quad sp_i = \# \text{ of edges with } i \text{ shared partners}$$

This configuration contains:

- 1 edge with 3 shared partners
- 6 edges with 1 shared partner
- 9 edges involved in the three triangles (one of these is counted 3 times)

α	GWESP(α)	
0	= 7	this is the number of edges involved in at least one triangle
0.5	= 7.55	
1	= 8.03	this is headed towards 9, the number of edges in the 3 triangles

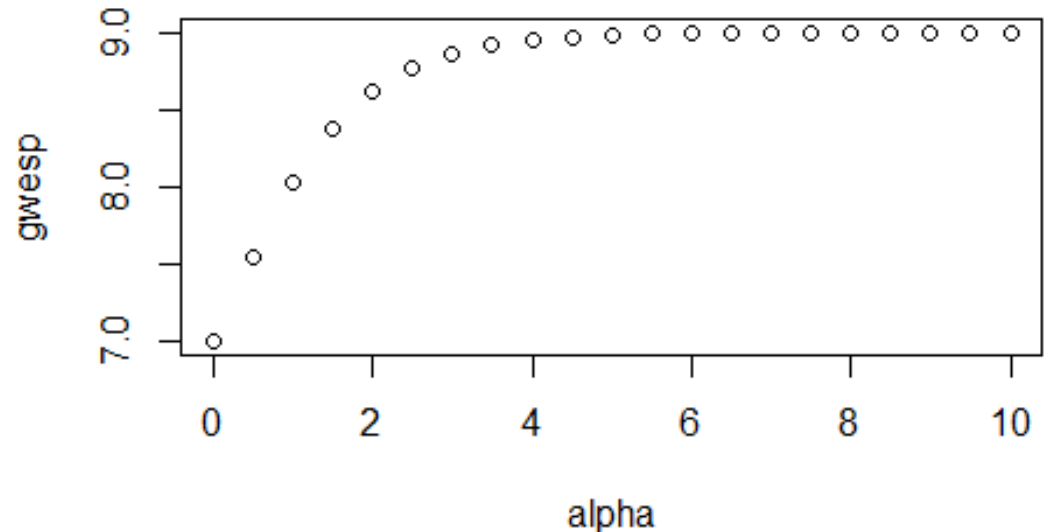
Solution: better network statistics


$$gwesp = e^{\alpha} \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} sp_i$$

$sp_i = \#$ of edges with i shared partners

A count of each edge
in each triangle
(i.e. # of triangles x 3)

A count of edges in at
least one triangle
(only the first triangle
counts)

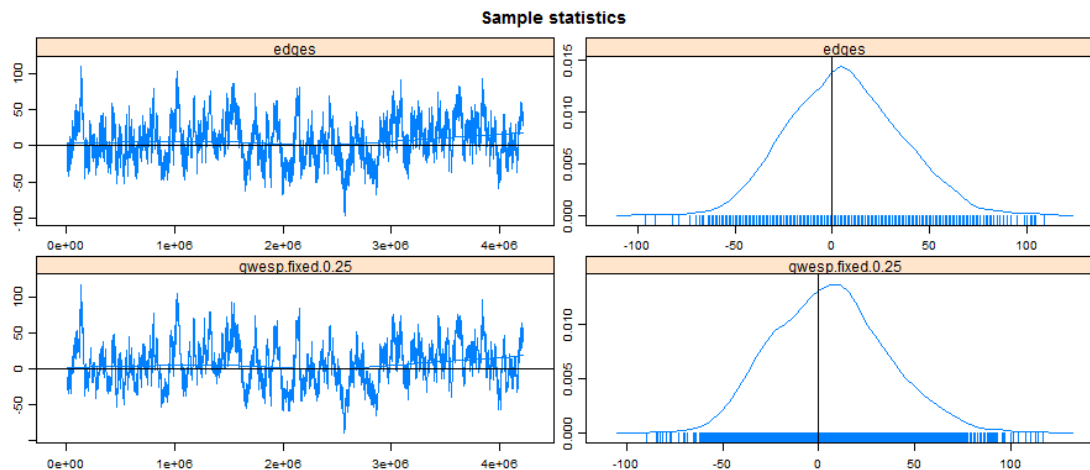


Fit the model with edges and gwesp

- Try edges + gwesp(0.25, fixed = TRUE)

MCMC Diagnostics

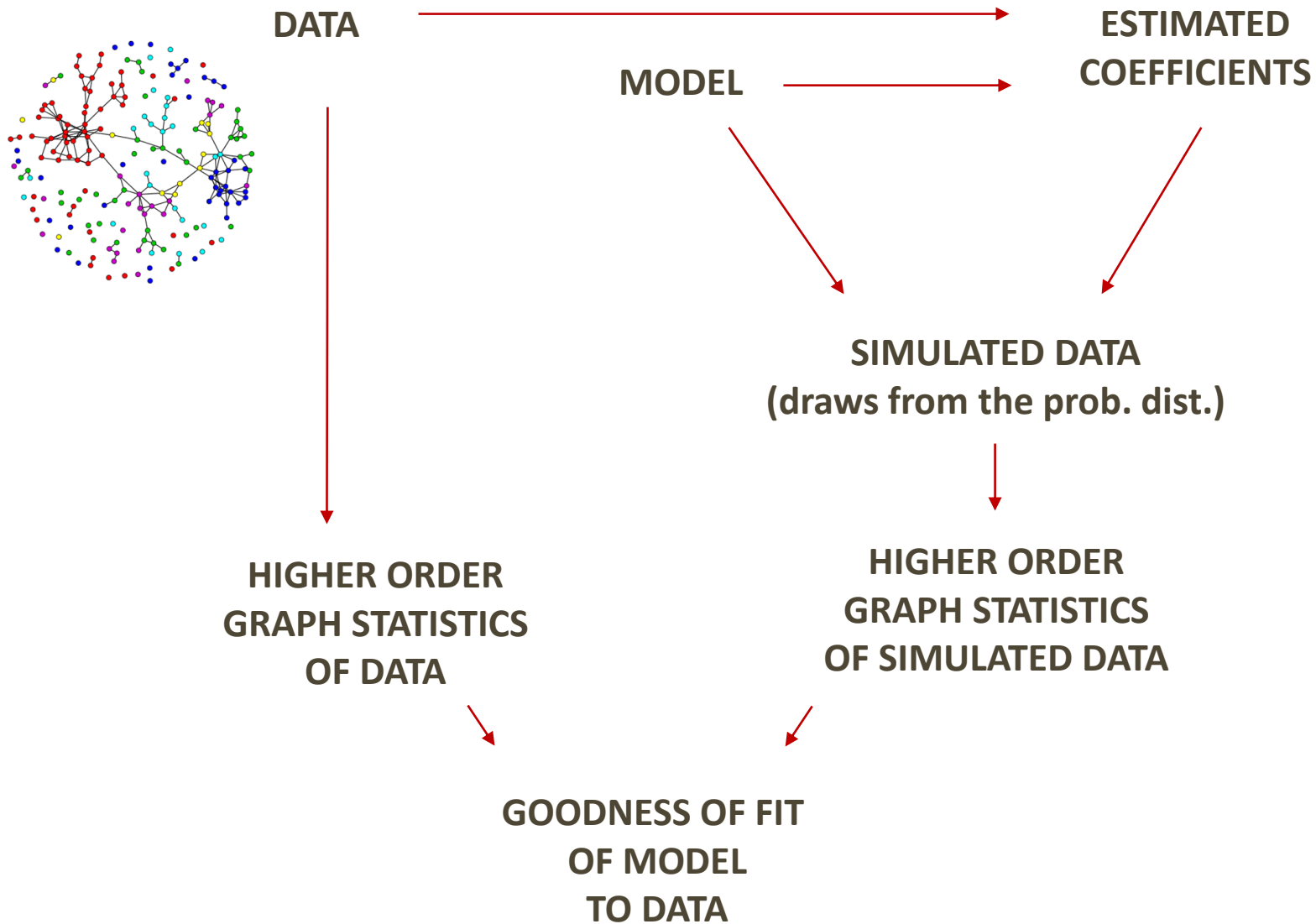
- Go to the MCMC Diagnostics tab
 - MCMC Diagnostics tell us if the estimation algorithm is mixing well



- But is it a good fit to the observed data? We need to check the goodness-of-fit diagnostics for that

Testing goodness of fit

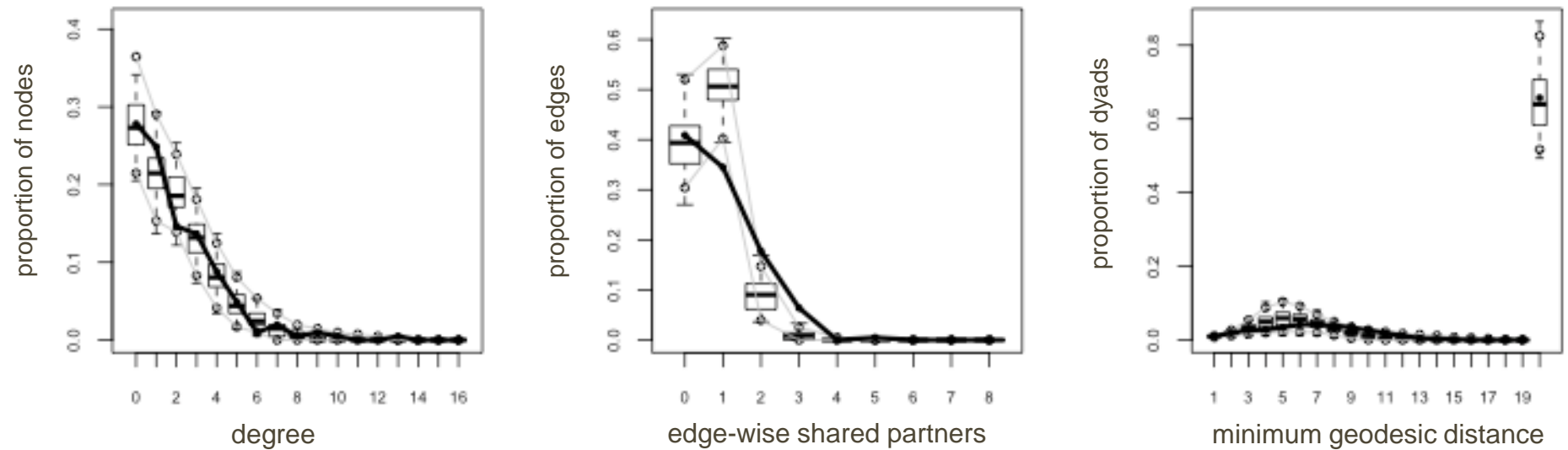
- Traditional GOF stats can be used
 - AIC, BIC in the model summary
- We also take another approach
 - We are interested in how well we fit aggregate properties of the network structure that we did **not** include as model terms
 - This helps to identify *what* the model gets wrong
 - We use 3 “higher order” statistics:
 - Degree distribution
 - Shared partner distribution (non-parametric) (local clustering)
 - Geodesic distance distribution (global clustering)



Goodness of Fit

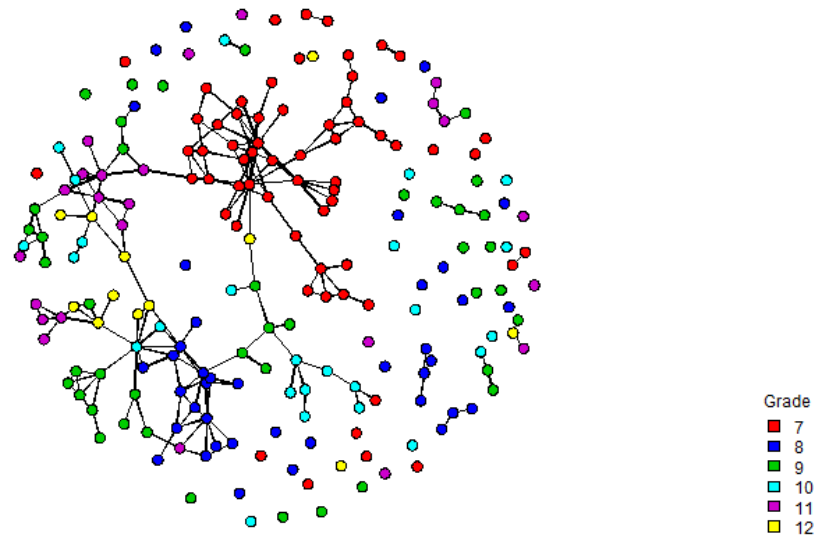
- Run the default set of GOF terms for this model

`faux.mesa.high ~ edges + gwesp(0.25, fixed = TRUE)`



And the eyeball test...

Even though our model isn't degenerate, we haven't fit the structure of the observed data - specifically, the levels of clustering.



So, back to our original question:

How much of the clustering is due to homophily, and how much to transitivity?

Test this by comparing four models

Model	Network Statistics $g(y)$
Edges	# of edges
Edges + Attributes (homophily)	# of edges # of edges for each race, sex ,grade # of edges that are within-race, within-grade, within-sex
Edges + GWESP (transitivity)	# of edges weighted shared partners
Edges + Attributes + GWESP (both)	# of edges # of edges for each race, sex ,grade # of edges that are within-race, within-grade, within-sex weighted shared partners

Fitting and saving models

■ statnetWeb allows you to save up to five models at a time

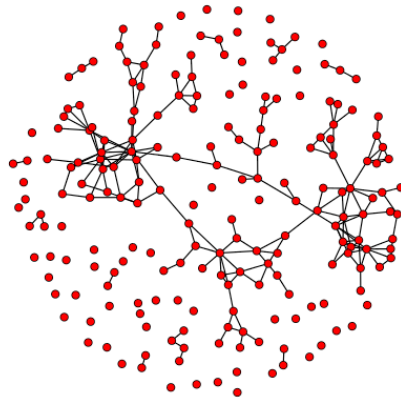
1. edges
 - Fit model, save model, reset formula
2. edges + nodefactor("Grade") + nodefactor("Race") + nodefactor("Sex") + nodematch("Grade", diff = TRUE) + nodematch("Race", diff = FALSE) + nodematch("Sex", diff = FALSE)
 - Fit model, save model, reset formula
3. edges + gwesp(0.25, fixed = TRUE)
 - Fit model, save model, reset formula
4. edges + nodefactor("Grade") + nodefactor("Race") + nodefactor("Sex") + nodematch("Grade", diff = TRUE) + nodematch("Race", diff = FALSE) + nodematch("Sex", diff = FALSE) + gwesp(0.25, fixed = TRUE)
 - Fit model, save model

Model Comparison

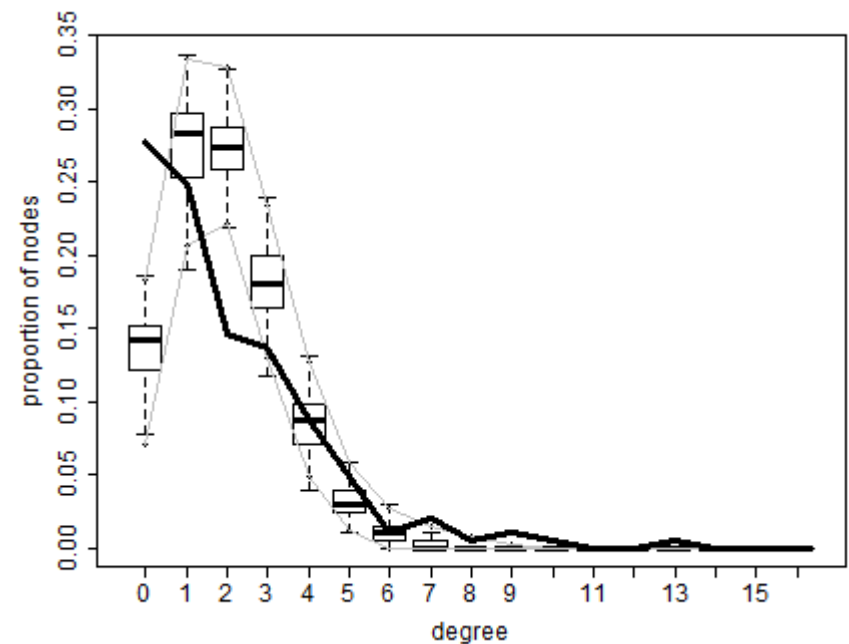
	Model1	Model2	Model3	Model4
edges	-4.63***	-8.491***	-5.58***	-9.056***
nodefactor.Grade.8	NA	1.562*	NA	1.413*
nodefactor.Grade.9	NA	2.533***	NA	2.184***
nodefactor.Grade.10	NA	2.942***	NA	2.520***
nodefactor.Grade.11	NA	2.660***	NA	2.260***
nodefactor.Grade.12	NA	3.470***	NA	2.901***
nodefactor.Race.Hisp	NA	-1.571***	NA	-1.015***
nodefactor.Race.NatAm	NA	-1.103***	NA	-0.758***
nodefactor.Race.Other	NA	-2.916**	NA	-2.013*
nodefactor.Race.White	NA	-0.809**	NA	-0.499*
nodefactor.Sex.M	NA	-0.335***	NA	-0.163*
nodematch.Grade.7	NA	7.441***	NA	5.974***
nodematch.Grade.8	NA	4.330***	NA	3.254***
nodematch.Grade.9	NA	2.060***	NA	1.645***
nodematch.Grade.10	NA	1.234*	NA	1.036.
nodematch.Grade.11	NA	2.525***	NA	1.912***
nodematch.Grade.12	NA	1.358.	NA	1.057.
nodematch.Race	NA	0.832***	NA	0.734***
nodematch.Sex	NA	0.638***	NA	0.543***
gwesp.fixed.0.25	NA	NA	1.86***	1.388***
AIC	2288	1809	1999	1648
BIC	2296	1960	2015	1807

Goodness of fit measure 1: degree distribution

Data:

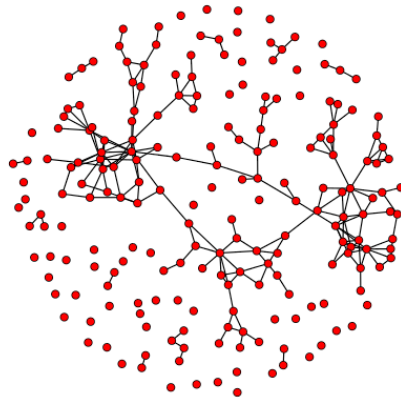


Model: Bernoulli
(i.e. edges only)

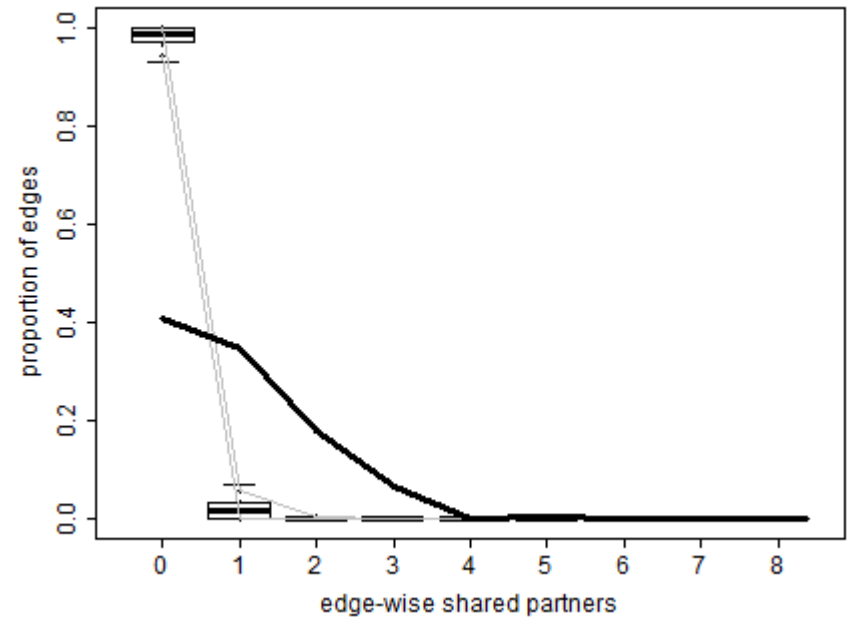


Goodness of fit measure 2: ESP distribution (local clustering)

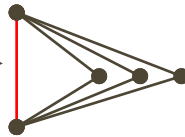
Data:



Model: Bernoulli
(i.e. edges only)

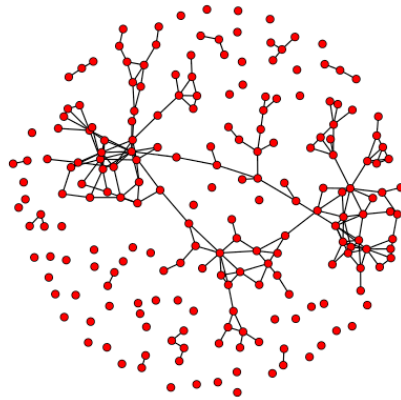


This edge has an ESP value of 3 →

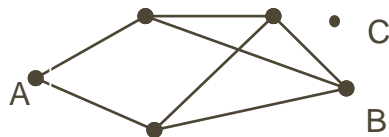


Goodness of fit measure 3: geodesic distribution (global clustering)

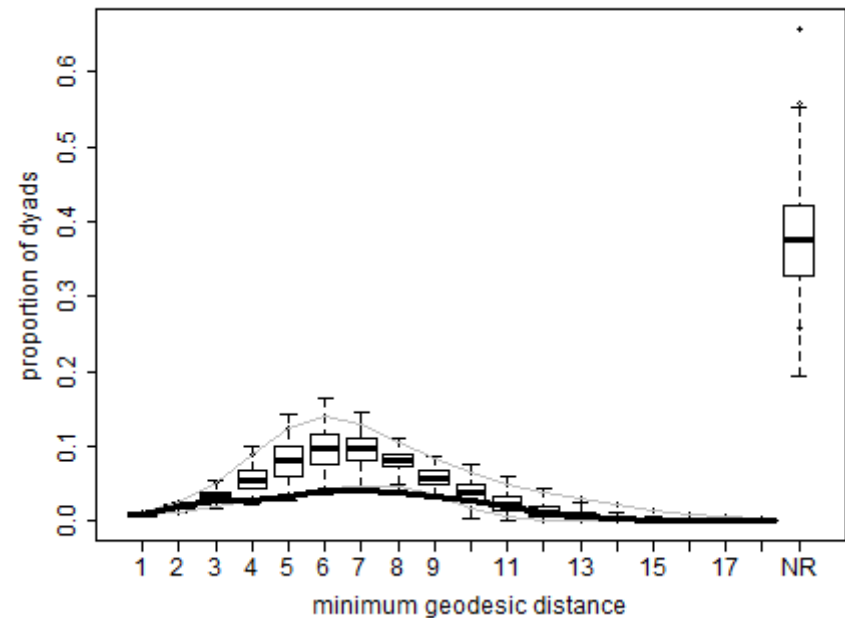
Data:



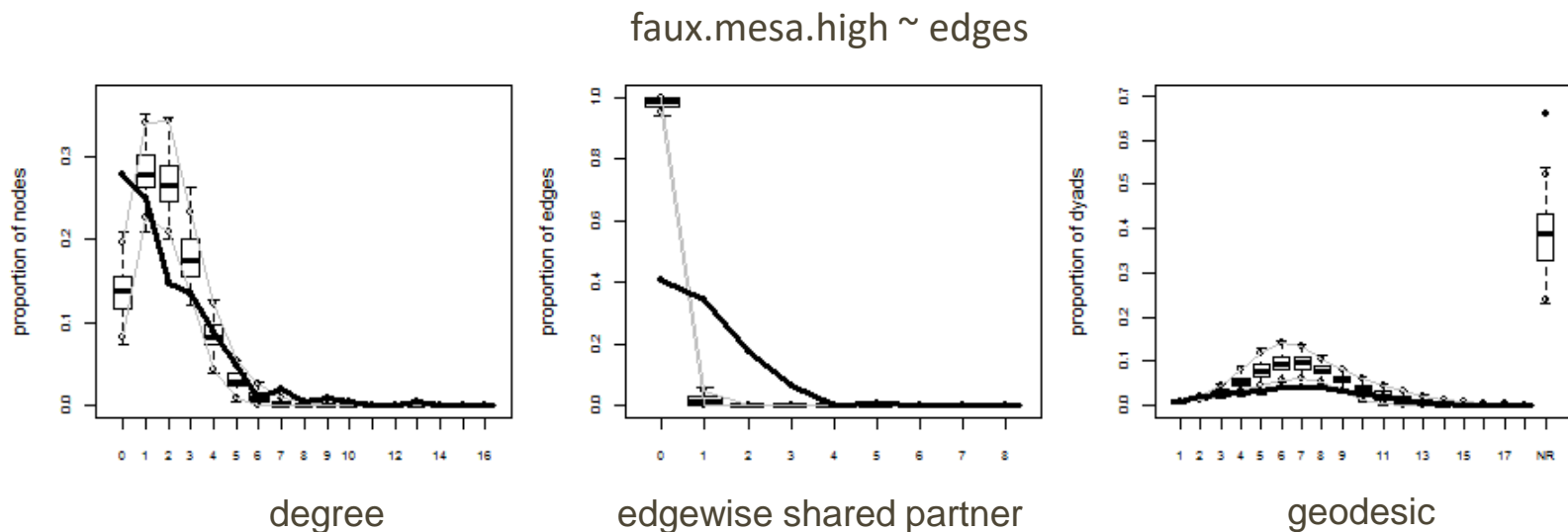
Model: Bernoulli
(i.e. edges only)



A/B have geodesic 2
A/C have geodesic ∞



Goodness of fit measures assembled



Summary: Not a good fit to any of the aggregate structural properties observed

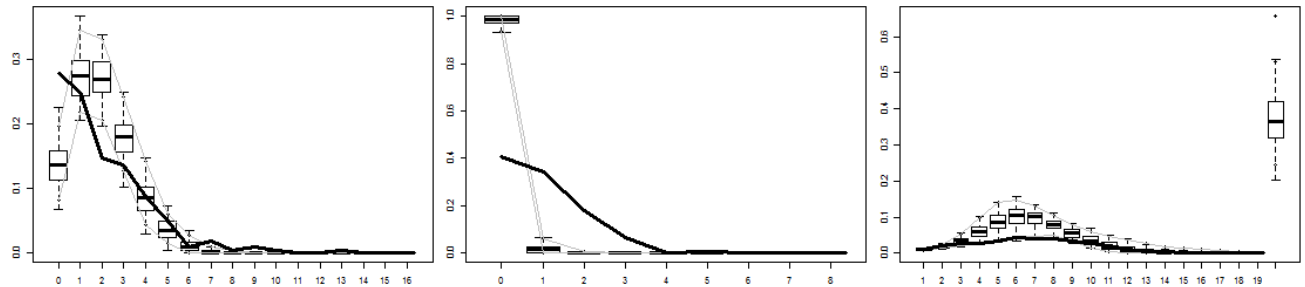
All 4 models:

degree

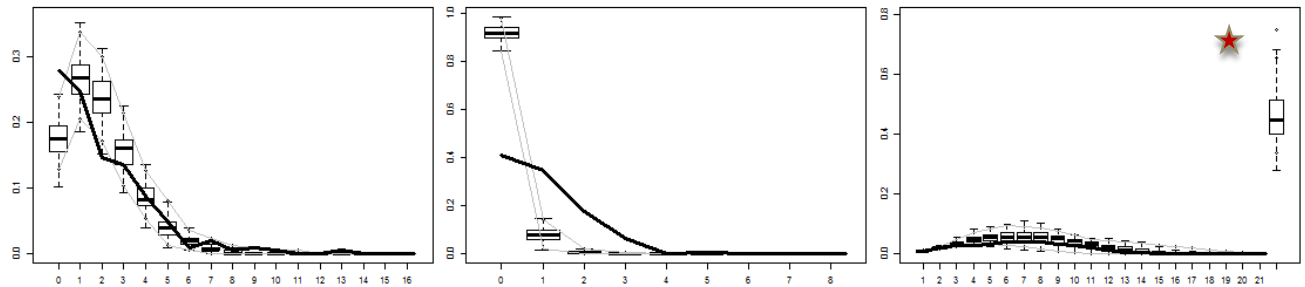
edgewise shared partner

geodesic

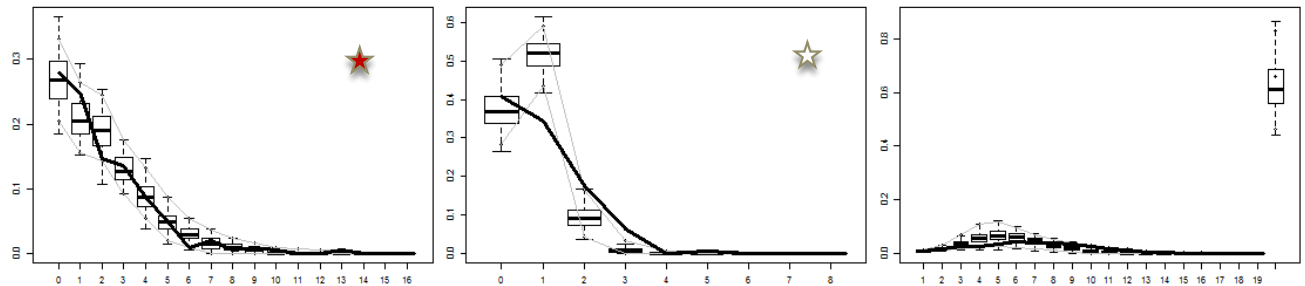
Model: Edges
AIC: 2288



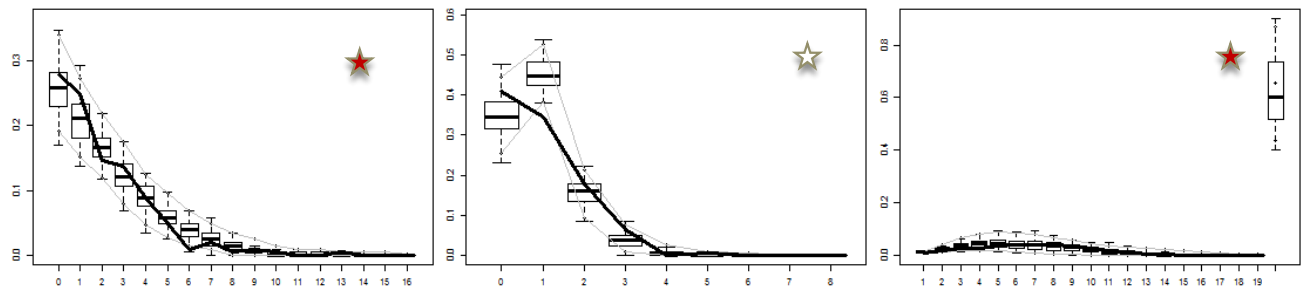
Model:
Edges + Attributes
AIC: 1809



Model:
Edges + GWESP
AIC: 1999



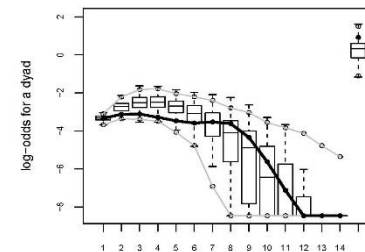
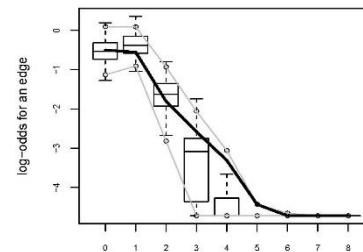
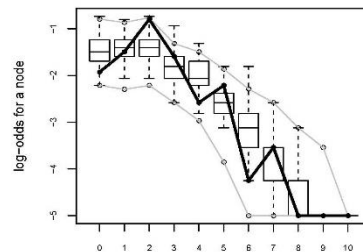
Model: Edges +
Attributes + GWESP
AIC: 1648



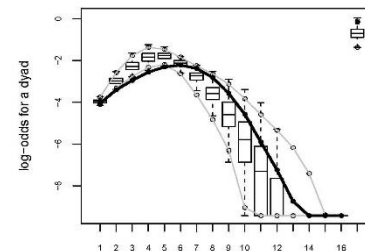
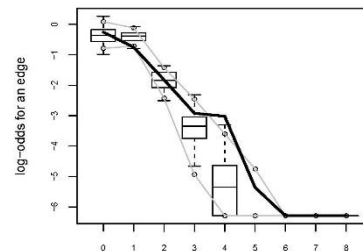
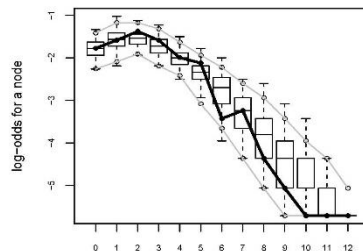
Attributes + GWESP Model as network size increases

n:

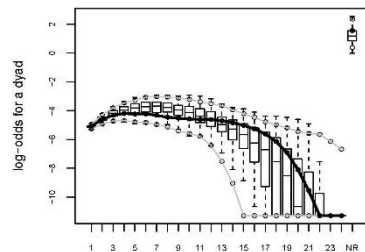
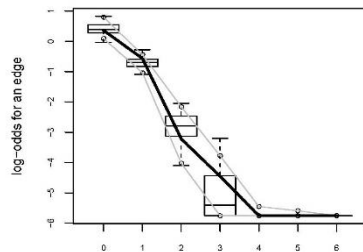
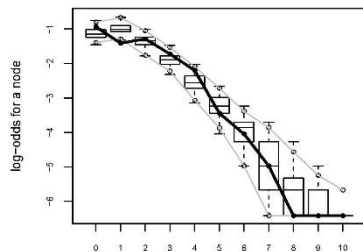
71



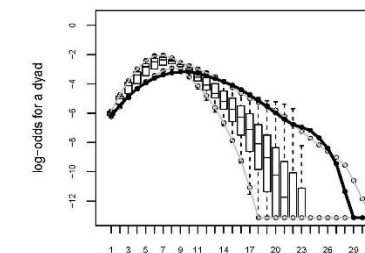
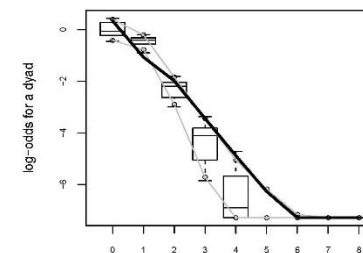
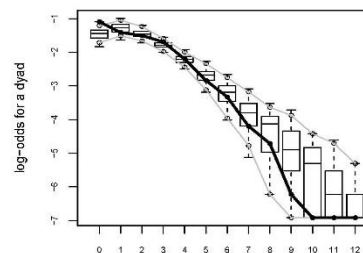
159



291



1011

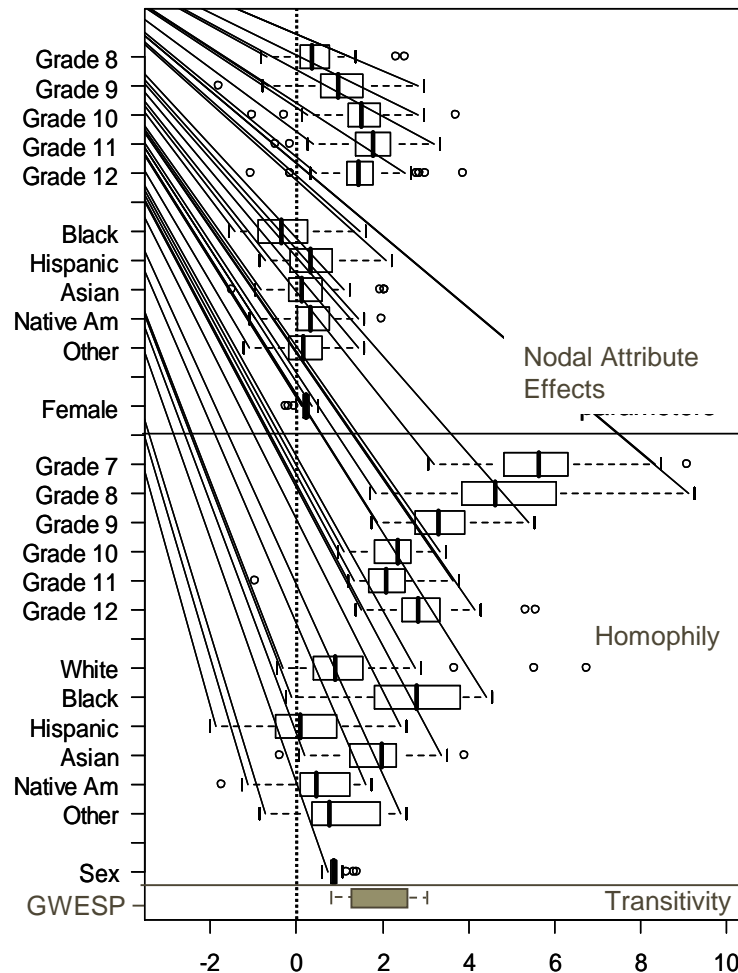


degree

edgewise shared partner

geodesic

Parameter values: Across all 59 schools



While there is variation across the schools, there are also clear and systematic patterns that are shared.

And the structural effects are generally stronger than the nodal attribute effects.

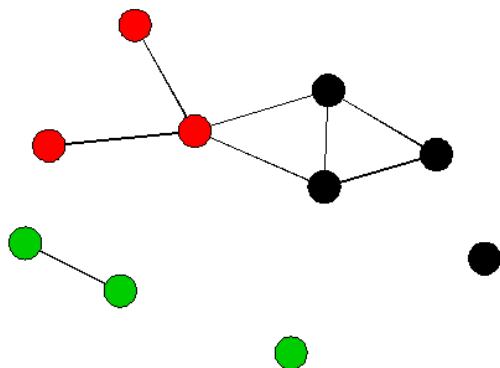
Findings

- Both transitivity and homophily play a role in clustering friendships
 - Homophily alone would generate the distribution of path lengths
 - A simple parametric form of transitivity captures local clustering
 - 25% of the transitivity effect is a by-product of homophily
 - Grade mixing is typically stronger than race mixing
 - but also less robust to the transitivity confound
- All 4 of these models begin to fail on the largest schools
 - There is more clustering than these models predict
 - Suggests additional sources of heterogeneity
 - Or perhaps an endogeneous “fissioning” of groups

Simulations

- Choose one of the models that you have saved and run 100 simulations with the default control settings
 - Choose the model on the Simulations page next to “ergm formula”
 - Do you see autocorrelation in the simulation statistics?
- Increase the MCMC interval to 10,000 and re-run the simulations to see how this changes

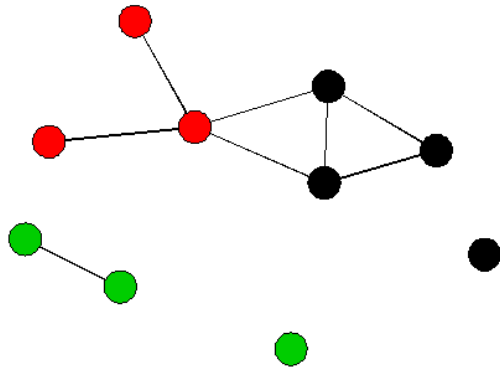
Common statistics in ergms



undirected network of 10 nodes,
including nodal attribute “color”, with
values *1=black, 2=red, 3=green*

Term	Formula	Unit	Value(s)
Edges	# of edges	edges	
degree (0)	# of nodes of degree 0	nodes	
degree (2:5)	# of nodes of degrees 2, 3, 4, 5 each	nodes	
concurrent	# of nodes of at least degree 2	nodes	

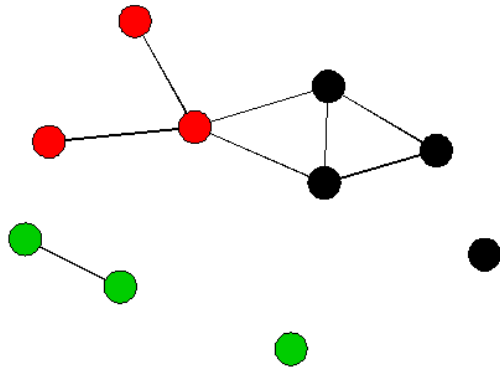
Common statistics in ergms



undirected network of 10 nodes,
including nodal attribute "col", with
values *1=black, 2=red, 3=green*

Term	Formula	Unit	Value(s)
<code>nodemix("col", base=1)</code>	# of edges between nodes of each color combo	edges	.
<code>nodefactor("col")</code>	Sum of degrees for nodes of each color	nodes/ edges*	
<code>nodefactor("col", base=2)</code>	Sum of degrees for nodes of each color uses group 2 (red) as the omitted baseline category	nodes/ edges*	
<code>nodematch("col")</code>	# of edges between nodes of same color	edges	
<code>nodematch("col", diff=T)</code>	# of edges between nodes of same color, for each color	edges	

Common statistics in ergms



undirected network of 10 nodes,
including nodal attribute “color”, with
values *1=black, 2=red, 3=green*

Term	Formula	Unit	Value(s)
<code>~triangle</code>	# of triangles (beware!)	triangles	
<code>~gwesp(0)</code>	# of edges in at least one triangle	edges	
<code>~gwesp(∞)</code>	# of edges in triangles total ($=3 * \# \text{ triangles}$)	triangles	

Selected References

Goodreau, S., et al. (2009). "Birds of a Feather, or Friend of a Friend? Using Statistical Network Analysis to Investigate Adolescent Social Networks." Demography **46(1): 103–125**.

Journal of Statistical Software (v42) 2008 – Eight papers on ERGMs and statnet

Pavel N. Krivitsky and Mark S. Handcock (2014). [A Separable Model for Dynamic Networks](#). *Journal of the Royal Statistical Society*, Series B, Volume 76, Issue 1, pages 29–46.