LISS2108

Statistical Inference for Social Networks Analysis

Session 4. Fitting ERGMs

Santiago Quintero

santiago.quintero_suarez@kcl.ac.uk

March 27th 2025

KING'S College LONDON

LISS DTP

Session outline

1. Review:
   1.1. QAP
   1.2. Intro to ERGMs
2. Estimating network dependencies
3. ERGM coding walkthrough
4. Model selection and Goodness-Of-Fit
5. ERGM coding walkthrough 2
6. Class summary

1.1. Review: QAP

Modelling at the dyad level

Dyadic characteristics
- Similarity (homophily)
- Peer-effects
- Surrounding

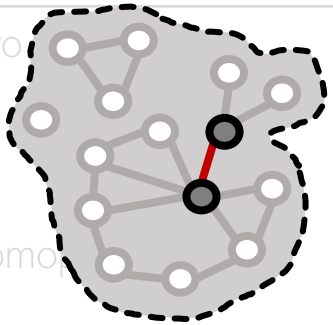
Dyad level

Presence or absence of a tie between two

Distance between two nodes

Number of common connections

Whether connected nodes are similar (homo



Dyadic outcomes
- Transfer

Examples:

- Who do people seek advice from?
- Why do firms do more business with some firms that with others?
- Why and which legislators work together to propose a bill?
- Are academics more likely to collaborate if they have a co-author in common?

Modelling at the dyad level: QAP

Quadratic Assignment Procedure (QAP)

For QAP, the rows and their corresponding columns in the (predictor) matrix are randomly permuted, while names and their attributes are held in the original order.

The main idea is that *we shuffle a node with which a particular label and set of attributes is associated while maintaining the structure of the network.*

	Ana	Tom	Jan	Lucy
Ana	0	1	0	0
Tom	1	0	1	3
Jan	0	1	0	4
Lucy	0	3	4	0

Original advice-seeking network

	~~Lucy~~ Ana	~~Jan~~ Tom	~~Ana~~ Jan	~~Tom~~ Lucy
~~Lucy~~ Ana	0	4	0	3
~~Jan~~ Tom	4	0	0	1
~~Ana~~ Jan	0	0	0	1
~~Tom~~ Lucy	3	1	1	0

Permuted advice-seeking network

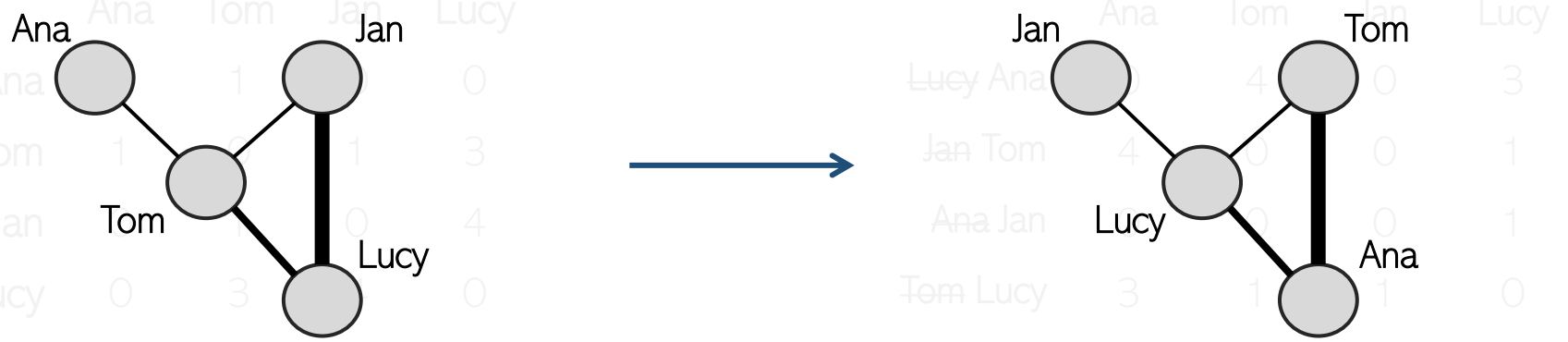
Modelling at the dyad level: QAP

Quadratic Assignment Procedure (QAP)

For QAP, the rows and their corresponding columns in the (predictor) matrix are randomly permuted, while names and their attributes are held in the original order.

The main idea is that *we shuffle a node with which a particular label and set of attributes is associated while maintaining the structure of the network.*

This way, **we implicitly account for all structural dependencies automatically**, even if we are unaware of them



Original advice-seeking network

Permuted advice-seeking network

Modelling at the dyad level: QAP

Quadratic Assignment Procedure (QAP)

QAP can be used for bivariate correlational analysis.

But there are also extensions to be used in a regression framework

- **Multiple Regression Quadratic Assignment Procedure (MRQAP)**
  - When the predicted network is weighted (i.e., the ties have values) it is essentially the same as an *OLS* but the *p*-value is calculated using QAP permutations.
  - When the predicted network is binary, it is essentially a *linear probability model*

- **Logistic Regression Quadratic Assignment Procedure (LRQAP)**
  - Logit model using QAP permutations to calculate statistical significance

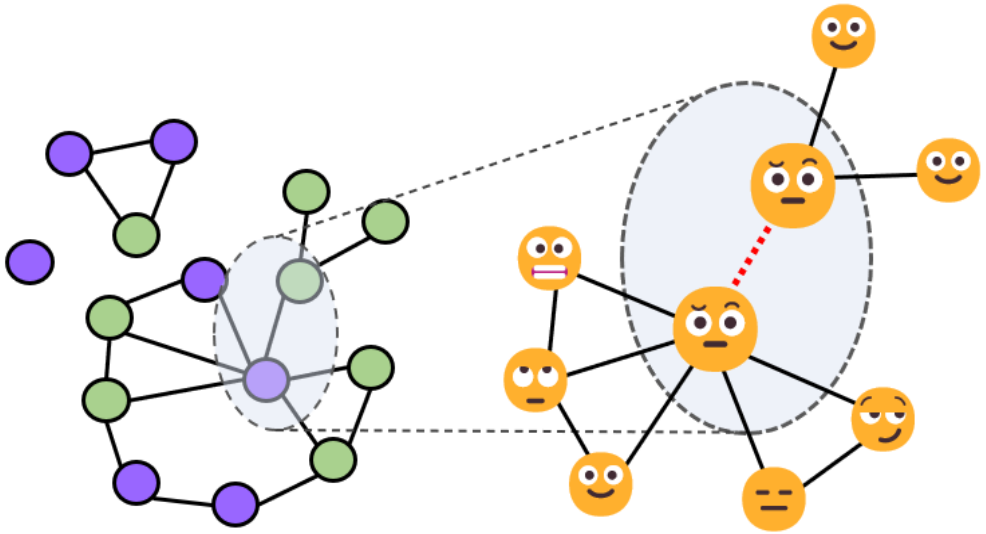
There are several methods for the specific QAP permutation, the most common being the *double semi-partialing method*, where we perform the permutations using the residual matrix from the original regression and we track the *t*-statistic rather than the regression coefficients (Dekker et al., 2007).

(Borgatti et al, 2020)

1.2. Review: Intro to ERGMs

A (somewhat) intuitive introduction to ERGMS

Consider an observed network as an aggregation of *local decision making-processes*—i.e., the sum of *individual **tie-based decisions** among pairs of nodes, taking into account a **local environment.***



Observed network

- This tie-based process is driven by an **aggregation** of **"local" social forces** that, we assume, led to the **current structure of the observed network.**

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

Y = Random variable for the state of the network
y = Actual network
$c(\theta)$ = Normalising constant to ensure probabilities $\{0,1\}$ (partition function)
θ_k = Coefficients for network statistics (e.g., number of edges, triangles, degree distribution, etc.)
$z_k(y)$ = Network statistics k

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Power, 2021)

A (somewhat) intuitive introduction to ERGMS

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

However, we don't simply care about predicting the whole network, but this general formulation implies that:

$$\text{logit}\left(P\left(Y_{ij} = 1 \middle| n \text{ actors}, Y_{ij}^c\right)\right) = \sum_{k=1}^{K} \theta_k \delta_{z_k(y)}$$

The "change statistic", δ, refers to the change in the network statistic when Y_{ij} goes from 0 to 1

The probability of a tie between i and j ...

... conditional on the rest of the network (Y_{ij}^c means all dyads other than Y_{ij})

We sum the product of all network statistics and their coefficients, $\theta\delta$

And if we rearrange to have the actual conditional probability of a tie between i and j in the left-hand side:

$$P\left(Y_{ij} = 1 \middle| n \; actors, Y_{ij}^c\right) = \text{logistic}(\theta_1 \delta_{z_1(y)} + \theta_2 \delta_{z_2(y)} + \theta_2 \delta_{z_3(y)} + \cdots)$$

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Power, 2021)

A (somewhat) intuitive introduction to ERGMS

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

The problem:

- There are too many possible networks to be simulated.
- The normalising constant requires summing all possible networks of the size of the observed one
- For a network with N nodes, the number of possible undirected graphs is $2^{\frac{N(N-1)}{2}}$ (each edge can be present or absent).
- This grows exponentially, making it **computationally intractable**!
- $c(\theta)$ has no closed-form solution and cannot be computed using Maximum Likelihood Estimation (as other GLM)

So, what do we do now?

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

A (somewhat) intuitive introduction to ERGMS

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

The problem:

- There are too many possible networks to be simulated.
- The normalising constant requires summing all possible networks of the size of the observed one
- For a network with N nodes, the number of possible undirected graphs is $2^{\frac{N(N-1)}{2}}$ (each edge can be present or absent).
- This grows exponentially, making it **computationally intractable**!
- $c(\theta)$ has no closed-form solution and cannot be computed using Maximum Likelihood Estimation (as other GLM)

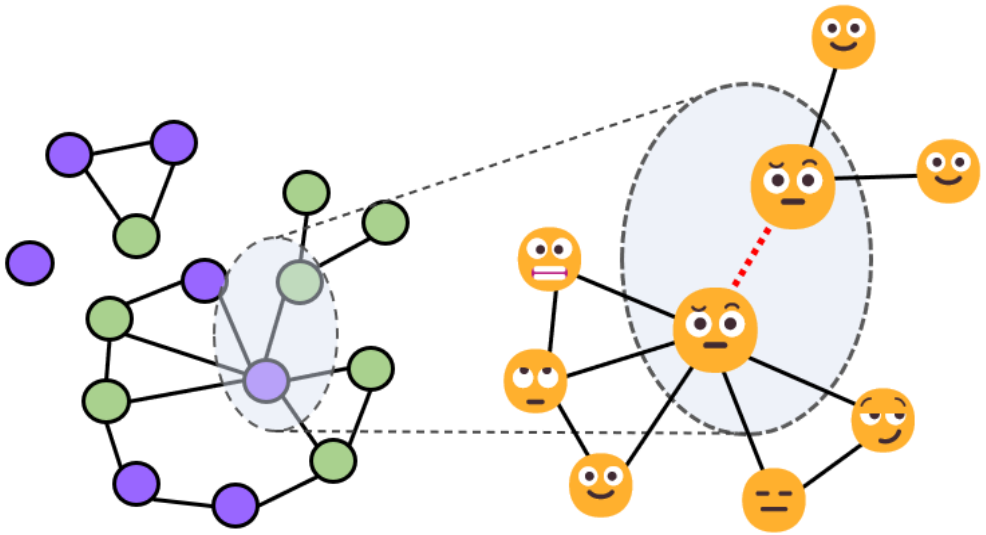
So, what do we do now?

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

2. Estimating network dependencies

Estimating network dependencies

The main objecting of ERGMs is estimating the parameters that express the forces that led to observing a network—their strength, direction and significance (*while controlling for other forces*).

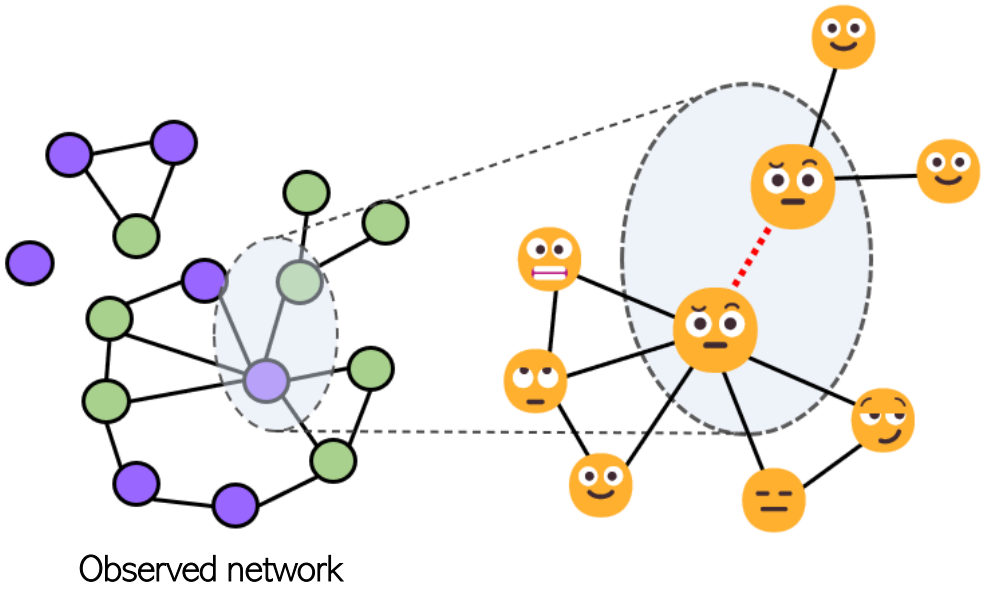


Observed network

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

The main objecting of ERGMs is estimating the parameters that express the forces that led to observing a network—their strength, direction and significance (*while controlling for other forces*).



Observed network

Is there an overall tendency to form or refrain from forming ties? (e.g., density)

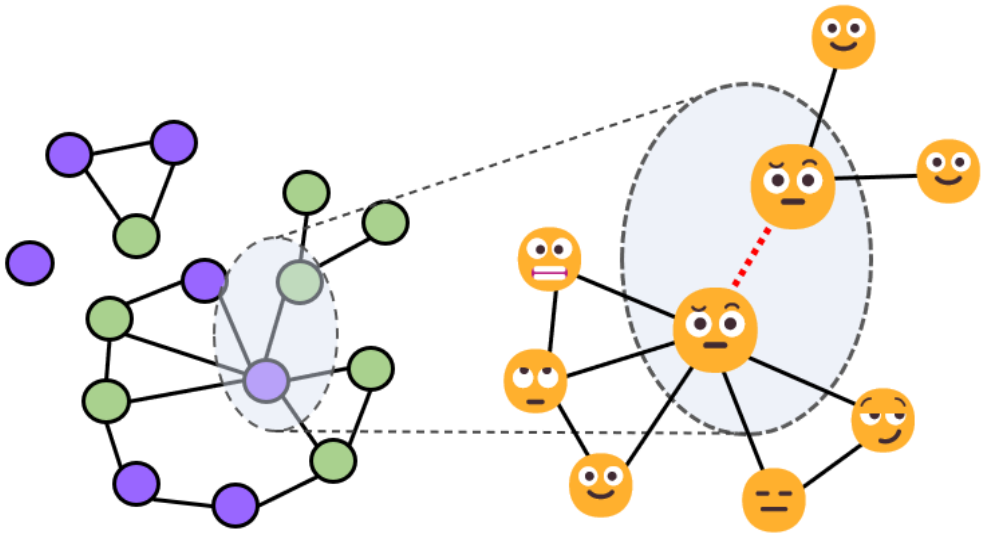
Is there a tendency for popular nodes to get more ties? (e.g., centralisation)

Is there a tendency for friends to be friends of friends? (e.g., triadic closure)

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

The main objecting of ERGMs is estimating the parameters that express the forces that led to observing a network—their strength, direction and significance (*while controlling for other forces*).



Observed network

Is there an overall tendency to form or refrain from forming ties? (e.g., density)

Forming connections demands time and energy -- 0 + Connections can be useful / People like being connected

Is there a tendency for popular nodes to get more ties? (e.g., centralisation)

People can only deal with a small number of connections -- 0 + People prefer to be connected to popular nodes

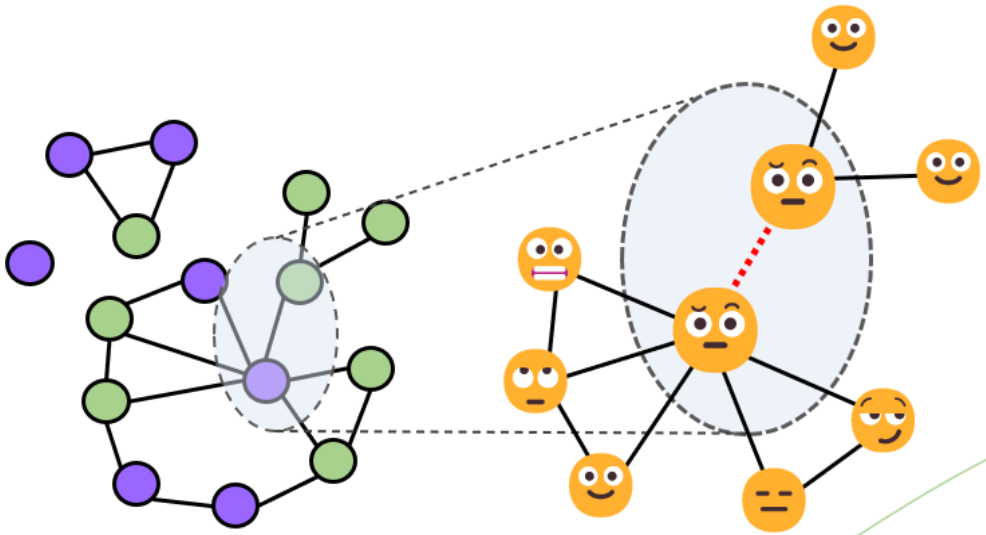
Is there a tendency for friends to be friends of friends? (e.g., triadic closure)

Redundancy might not be useful -- 0 + Group support might provide safety

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

The main objecting of ERGMs is estimating the parameters that express the forces that led to observing a network—their strength, direction and significance (*while controlling for other forces*).



Observed network

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

Is there an overall tendency to form or refrain from forming ties? (e.g., density)

Forming connections demands time and energy -- 0 + Connections can be useful / People like being connected

Is there a tendency for popular nodes to get more ties? (e.g., centralisation)

People can only deal with a small number of connections -- 0 + People prefer to be connected to popular nodes

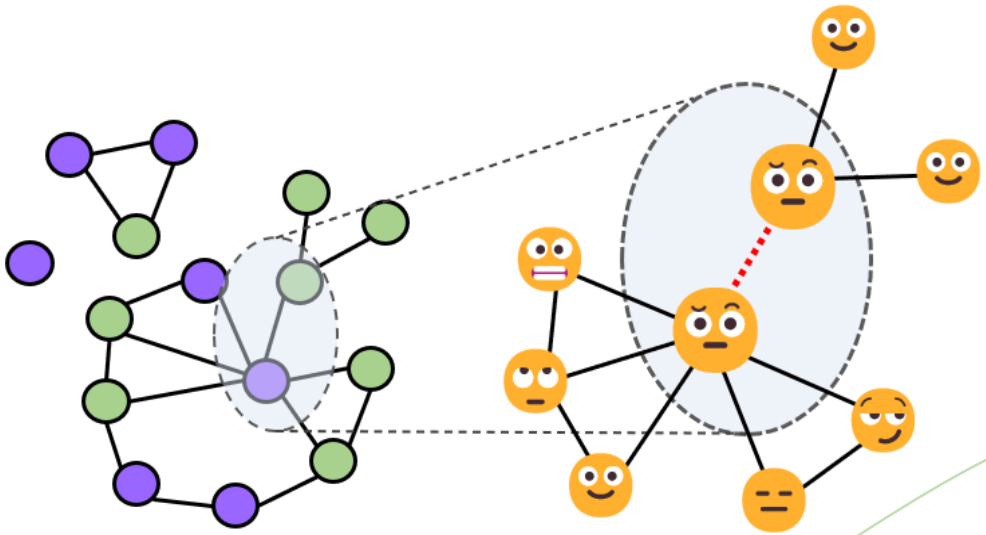
Is there a tendency for friends to be friends of friends? (e.g., triadic closure)

Redundancy might not be useful -- 0 + Group support might provide safety

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

The main objecting of ERGMs is estimating the parameters that express the forces that led to observing a network—their strength, direction and significance (*while controlling for other forces*).



Observed network

$$P(Y = y) = \frac{\exp\{\sum_{k=1}^{K} \theta_k z_k(y)\}}{c(\theta)}$$

The ERGM estimation tries to find the $\boldsymbol{\theta}$s that maximise the likelihood to create a network that resembles the characteristics of the observed network

Is there an overall tendency to form or refrain from forming ties? (e.g., density)

Forming connections demands time and energy -- 0 + Connections can be useful / People like being connected

Is there a tendency for popular nodes to get more ties? (e.g., centralisation)

People can only deal with a -- 0 + People prefer to be connected to popular nodes

ends to be friends of friends?

Redundancy might not be useful -- 0 + Group support might provide safety

(Agneessens, 2023; Robins et al. 2007; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

For an identified model, these θ are a set of sufficient statistics that represent the data generating process.

When specifying the model in R, these network statistics take the form of something like regression terms:

network ~ edges + netstat1 + netstat2 +⋯+ netstat k

- **edges** recovers the density of the observed network –it works as an intercept term that captures the overall tendency of tie formation
- The rest of network statistics try to model the different network formation processes we want to consider, and there are many:

(Robins et al. 2007; Harris, 2014; Power, 2021; Borgatti et al, 2020)

Estimating network dependencies

Node-level terms:

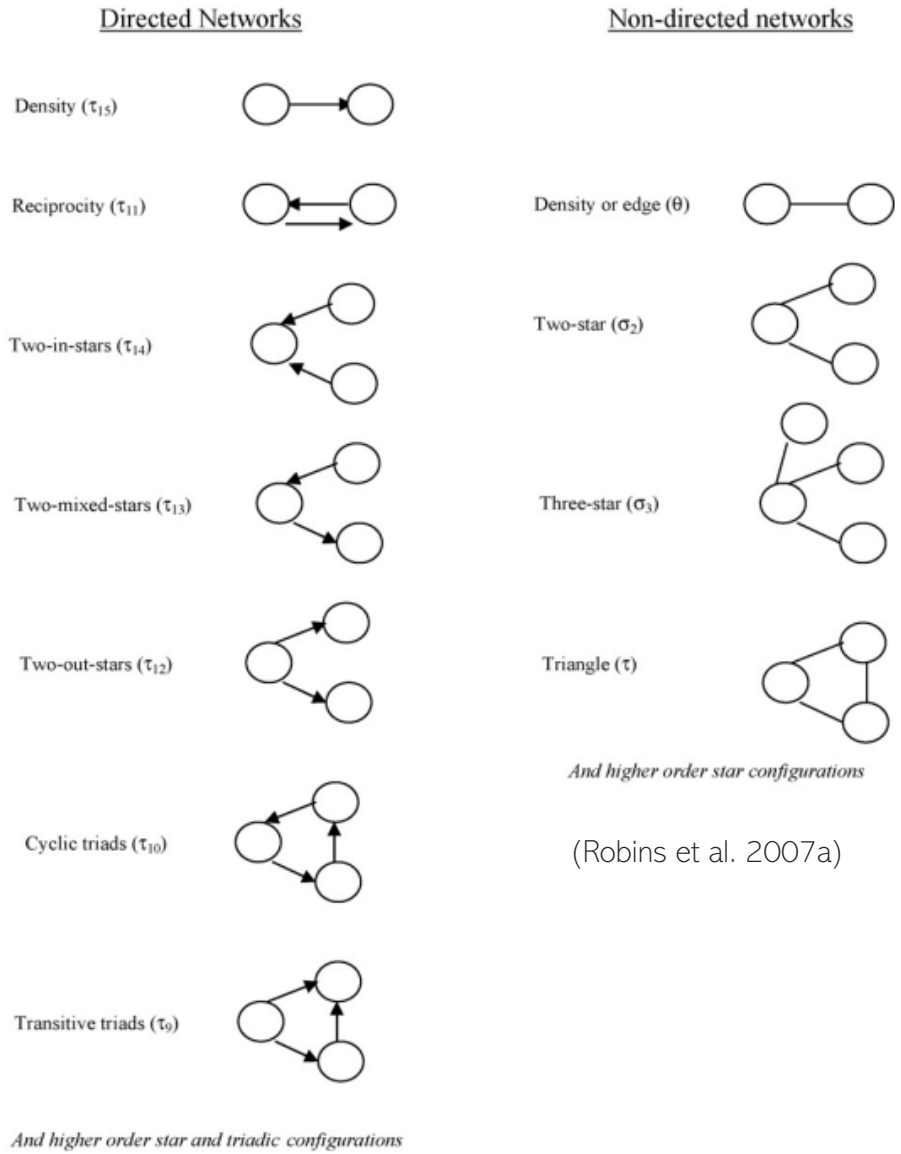
- `nodefactor(), nodeifactor(), nodeofactor()`
- `nodecov(), nodeicov(), nodeocov()`

Dyad-level terms

- Homophily (categorical variables): `nodematch()`
- Relative difference: `diff(), absdiff()`
- Dyad attributes: `edgecov()`

Structural terms:

- Reciprocity: `mutual()`
- Triangles: `triangles()`
- K-stars: `kstar()`
- Degree: `degree(), idegree(), odegree()`



(Robins et al. 2007a)

(Agneessens, 2023; Snijders et al., 2006; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

Node-level terms:

- `nodefactor(), nodeifactor(), nodeofactor()`
- `nodecov(), nodeicov(), nodeocov()`

Dyad-level terms

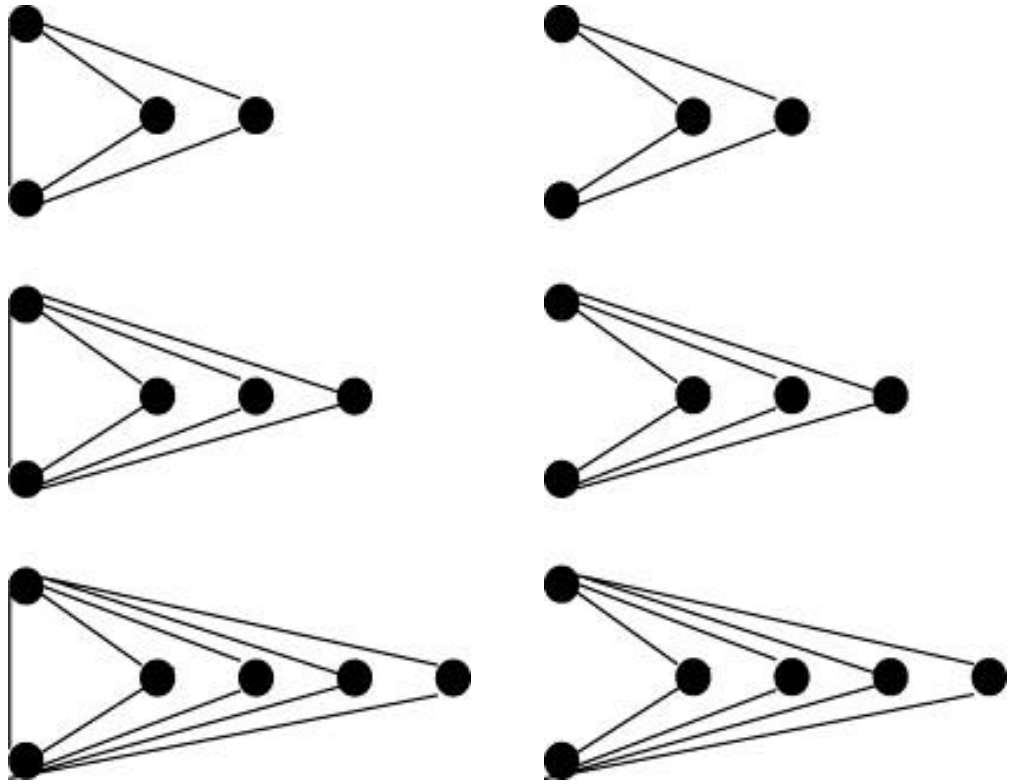
- Homophily (categorical variables): `nodematch()`
- Relative difference: `diff(), absdiff()`
- Dyad attributes: `edgecov()`

Structural terms:

- Reciprocity: `mutual()`
- Triangles: `triangles()`
- K-stars: `kstar()`
- Degree: `degree(), idegree(), odegree()`

Edge-wise shared partner: `gwesp()`

Dyad-wise shared partner: `gwdsp()`



(Robins et al. 2007b)

(Agneessens, 2023; Snijders et al., 2006; Harris, 2014; Borgatti et al, 2020)

Estimating network dependencies

Node-level terms:

- `nodefactor()`, `nodeifactor()`, `nodeofactor()`
- `nodecov()`, `nodeicov()`, `nodeocov()`

Dyad-level terms

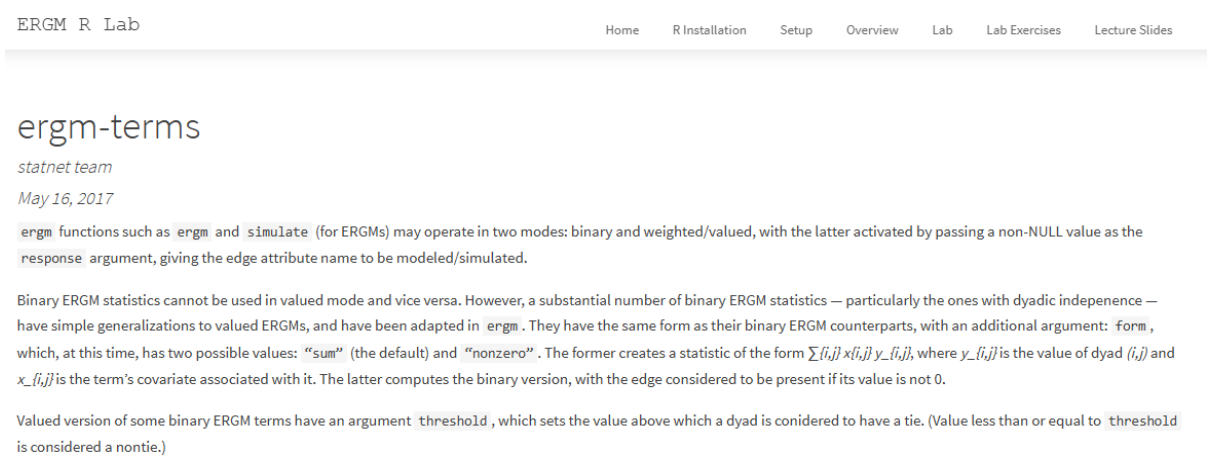
- Homophily (categorical variables): `nodematch()`
- Relative difference: `diff()`, `absdiff()`
- Dyad attributes: `edgecov()`

Structural terms:

- Reciprocity: `mutual()`
- Triangles: `triangles()`
- K-stars: `kstar()`
- Degree: `degree()`, `idegree()`, `odegree()`

There are many, many possibilities! BUT some of them are hard to fit and might lead to model degeneracy

Check here for an exhaustive list!



(Agneessens, 2023; Snijders et al., 2006; Harris, 2014; Borgatti et al, 2020)

3. ERGM coding walkthrough

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

ERGM_workshop.R* | QAP_workshop.R

Source on Save | Run | Source

```
1  ######### LISS2108 Statistical Inference for Social Networks Analysis ##########
2  # Author: Santiago Quintero (KCL) - 2025
3
4  # In this workshop, we'll work with the "General Relativity and Quantum Cosmolo
5  # collaboration network" collected by Leskovec et al (2007) (http://doi.acm.org
6  # This network contains all academic collaborations in the domain of GR&QC betw
7  # 1993-2003. It is an undirected network that connects nodes (authors) if they
8  # wrote a paper together. We'll practice fitting ERGMs with this network.
9
10 # Clear workspace and set seed for reproducibility
11 rm(list = ls())
12 set.seed(123)
13 options(scipen = 100) # turn off scientific notation
14
15
16 ## Libraries we'll work with
17 packages <- c("igraph", "network", "dplyr", "ergm", "intergraph")
18
19 # If the packages are not installed, run:
20 # install.packages(packages)
21
22 # Call the libraries
23 lapply(packages, require, character.only = TRUE)
24
25
```

123:1 | Fitting ERGMs | R Script

Environment | History | Connections | Tutorial

Import Dataset | 514 MiB | List

R | Global Environment

ergm.structure	List of 35
net	List of 5
net.igraph	List of 122
Values	
degree_centrality	num [1:122] 6 7 5 2 1 3 1 2 2 4 ...
packages	chr [1:5] "igraph" "network" "dplyr" "ergm" "intergraph"

Console | Terminal | Background Jobs

R 4.4.1 · ~/SANTIAGO/KCL/Inferential SNA/

```
This model was fit using MCMC.  To
examine model diagnostics and check for
degeneracy, use the mcmc.diagnostics()
function.
> summary(ergm.structure)
Call:
ergm(formula = net ~ edges + gwesp(0.5, fixed = TRUE) + gwdegree(0.5,
    fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

              Estimate Std. Error MCMC % z value Pr(>|z|)
edges          -4.6177     0.2184      0 -21.143 < 0.0001 ***
gwesp.fixed.0.5   0.7440     0.1301      0   5.720 < 0.0001 ***
gwdeg.fixed.0.5   1.3046     0.4683      0   2.786  0.00534 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 10232  on 7381  degrees of freedom
 Residual Deviance:  1488  on 7378  degrees of freedom

AIC: 1494  BIC: 1515  (Smaller is better. MC Std. Err. = 0.7177)
>
```

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish

GR & QC Co-Authorship Network

4. Model selection and Goodness-Of-Fit

Choosing models

How do we make sure that our model is actually capturing the network generating process?

Different combinations of network statistics could have created a decently similar network to that observed. How to be sure that the simulated networks did a god job? We have to evaluate the "Goodness of fit" of the model

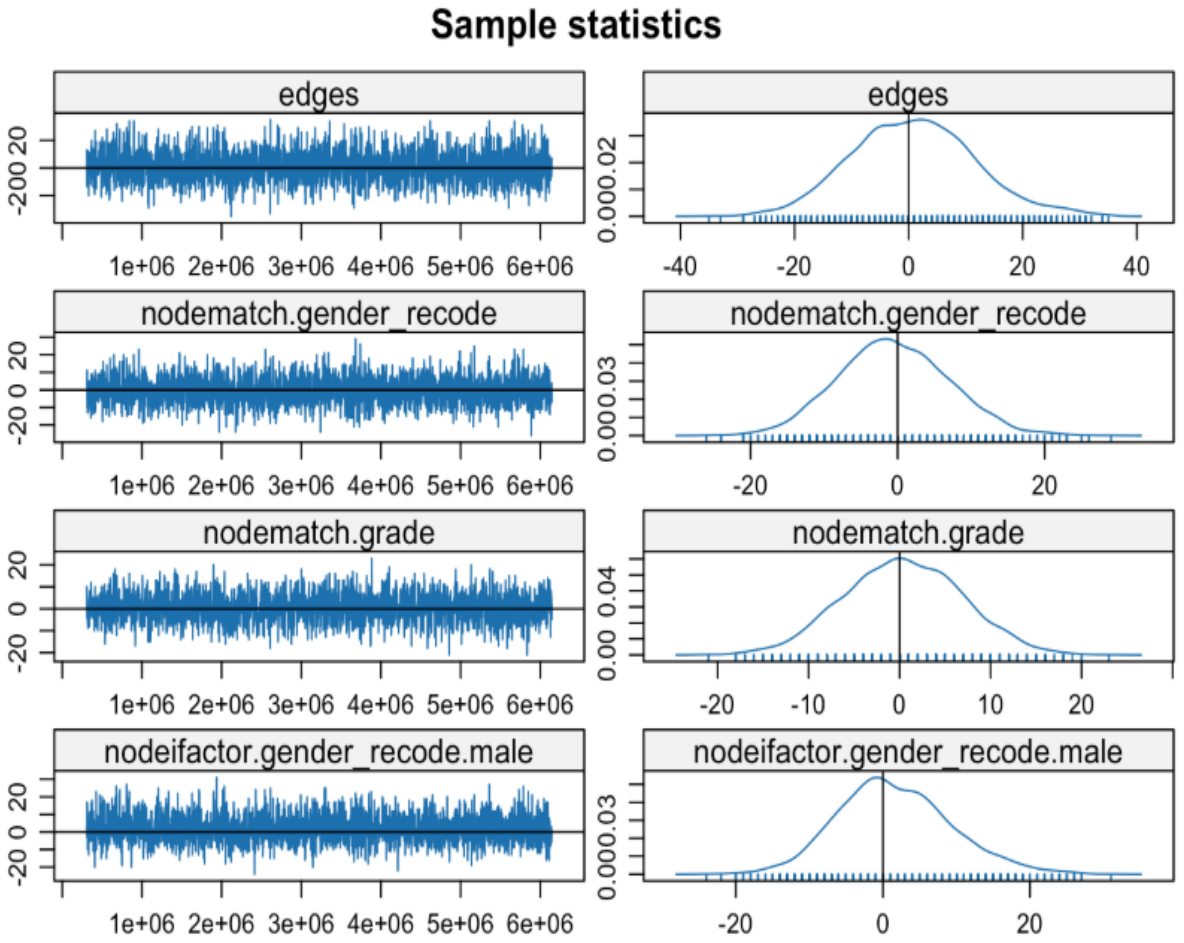
- When **choosing between models**, we can use the log-likelihood and the **Akaike Information Criteria** (AIC) and **Bayesian Information Criteria** (BIC) –the smaller the better (we want them to be close to 0)
- To make sure that the estimation process was adequate, we look at the MCMC process, **`mcmc.diagnostics()`**
- We evaluate the model looking at the Goodness-of-Fit plots, **`gof()`**

(Handcock, 2003; Robins et al. 2007; Borgatti et al, 2020)

Choosing models: MCMC diagnostics

You need to check that your MCMC diagnostics (how far the stats calculated from the sampled networks are from the observed network) look good. What to check for:

- No clear patterns in the " caterpillar": i.e., the MCMC sampling is not getting lost into a strange parameter space
- The distribution of sample stats should be loosely centered around 0, with some (but not extreme) variance.
- If these diagnostics don't look good, your model can be degenerate.
  - You can try to specify some control terms with **`control.ergm()`** inside the **`ergm`** formula, e.g., a longer burn-in or more iterations

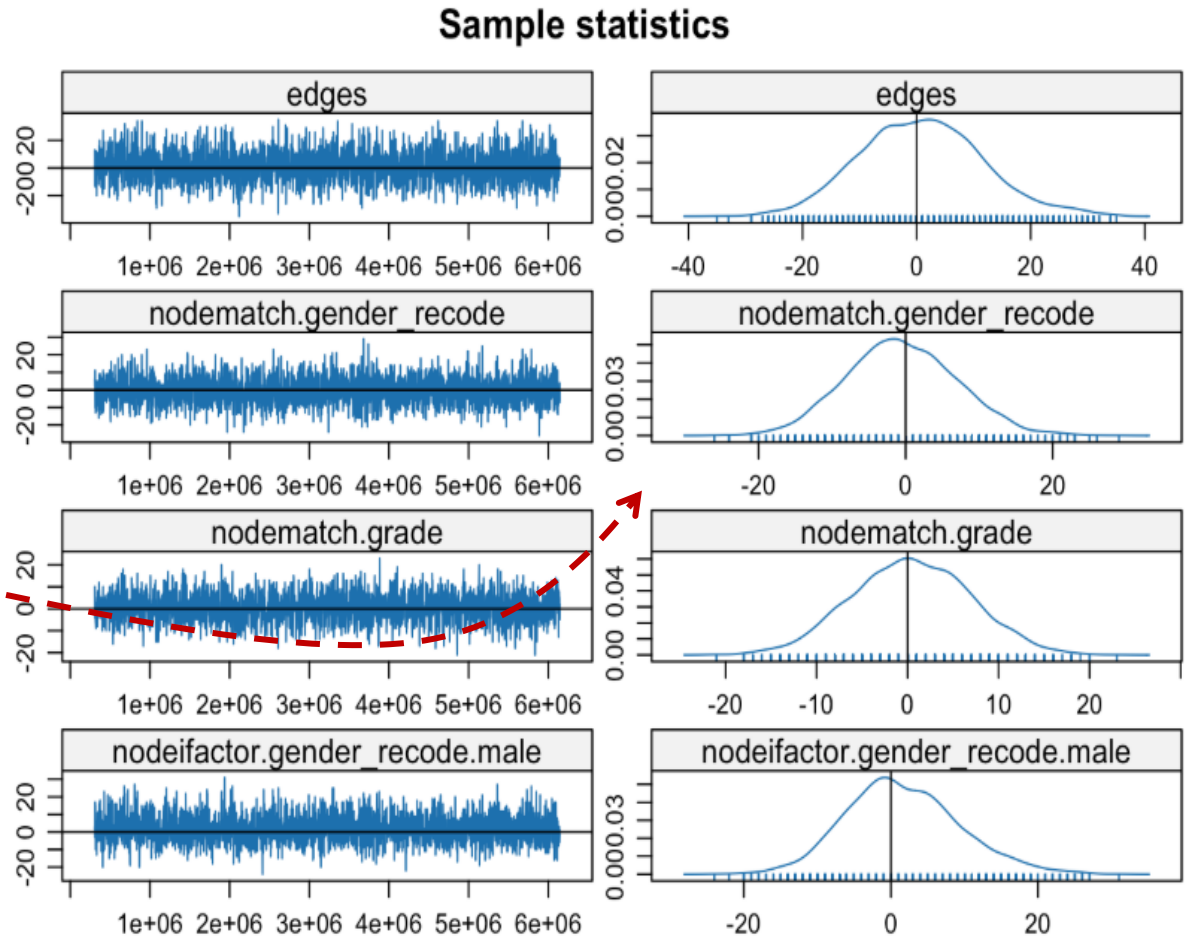


(Rawlings et al., 2023)

Choosing models: MCMC diagnostics

You need to check that your MCMC diagnostics (how far the stats calculated from the sampled networks are from the observed network) look good. What to check for:

- No clear patterns in the " caterpillar": i.e., the MCMC sampling is not getting lost into a strange parameter space
- The distribution of sample stats should be loosely centered around 0, with some (but not extreme) variance.
- If these diagnostics don't look good, your model can be degenerate.
  - You can try to specify some control terms with **`control.ergm()`** inside the **`ergm`** formula, e.g., a longer burn-in or more iterations



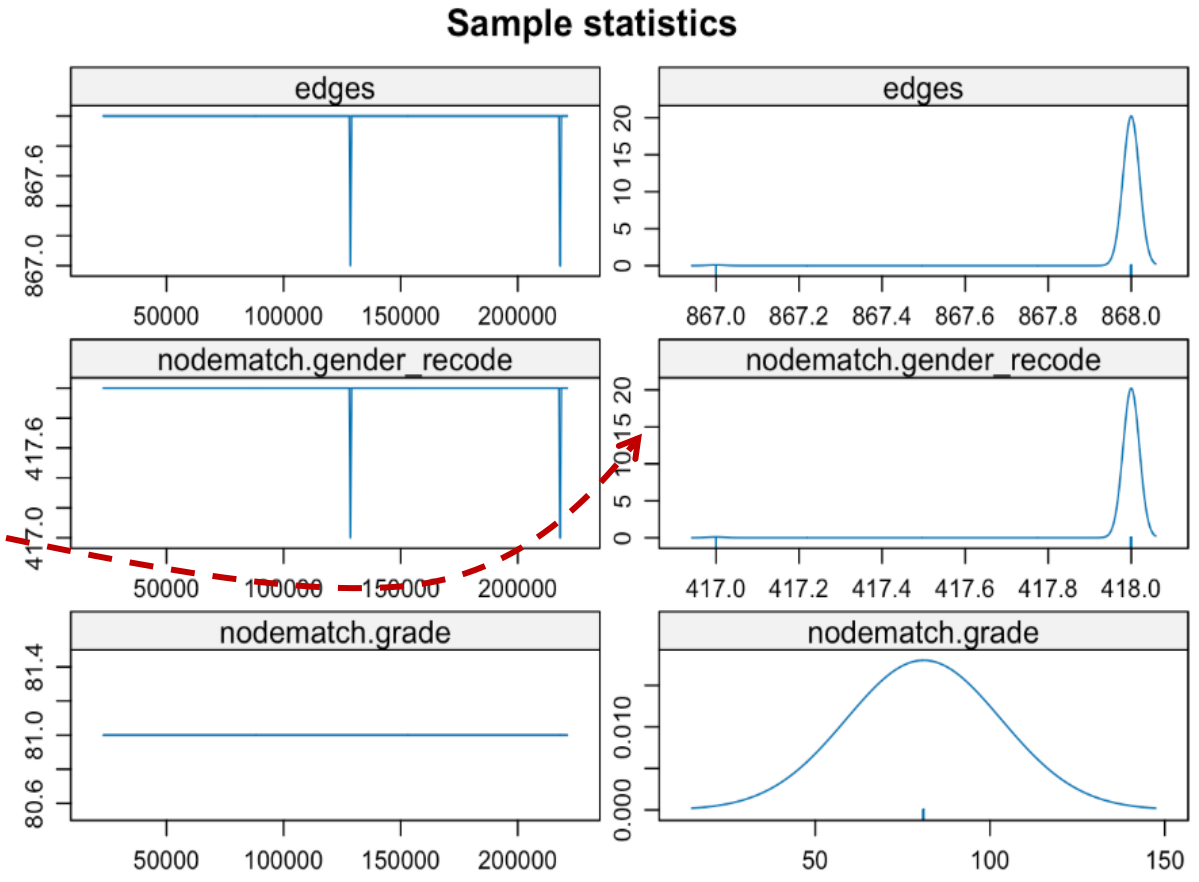
(Rawlings et al., 2023)

Choosing models: MCMC diagnostics

You need to check that your MCMC diagnostics (how far the stats calculated from the sampled networks are from the observed network) look good. What to check for:

- No clear patterns in the " caterpillar": i.e., the MCMC sampling is not getting lost into a strange parameter space
- The distribution of sample stats should be loosely centered around 0, with some (but not extreme) variance.
- If these diagnostics don't look good, your model can be degenerate.
  - You can try to specify some control terms with **`control.ergm()`** inside the **`ergm`** formula, e.g., a longer burn-in or more iterations

MCMC diagnostics **should not** look like this!

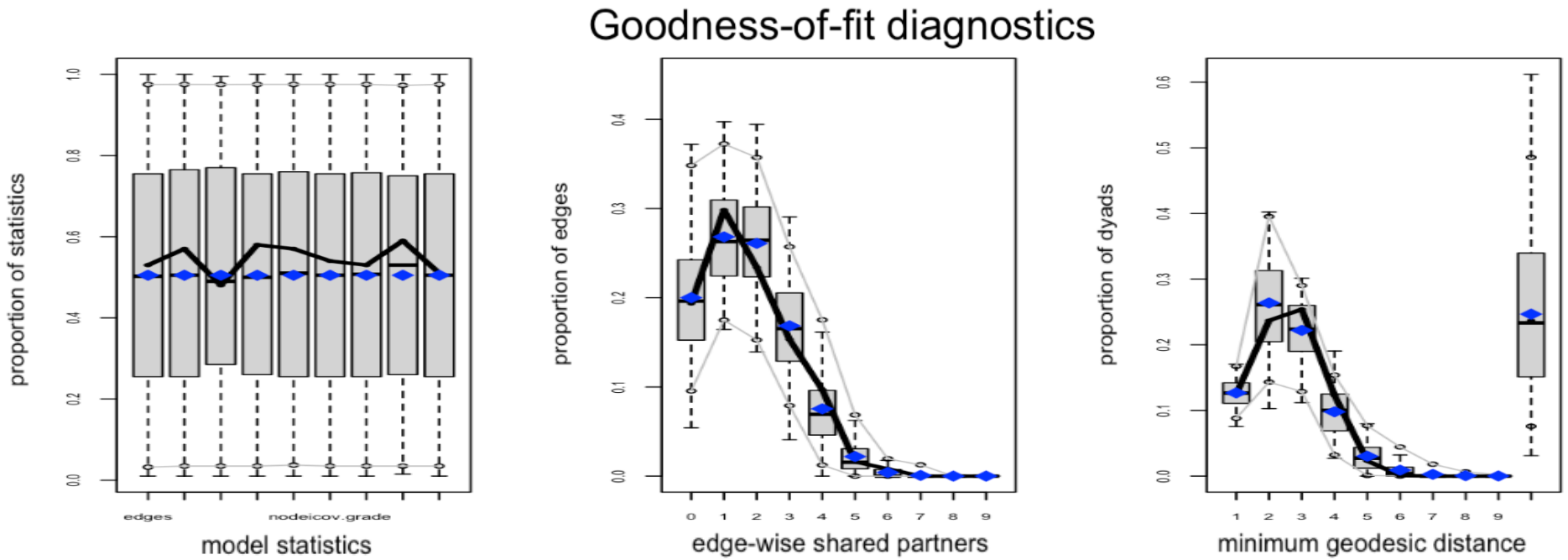


(Rawlings et al., 2023)

Choosing models: GOF plots

The Goodness-of-Fit plots tell you how much the simulated networks follow the structural characteristics (e.g., degree, edgewise shared partners, minimum geodesic distance) of the observed network (solid line).

- You want a close approximation! if the simulations are too off, you need to change the specification of the model (add or remove terms)
- Appropriate GOF plots look more or less like these:

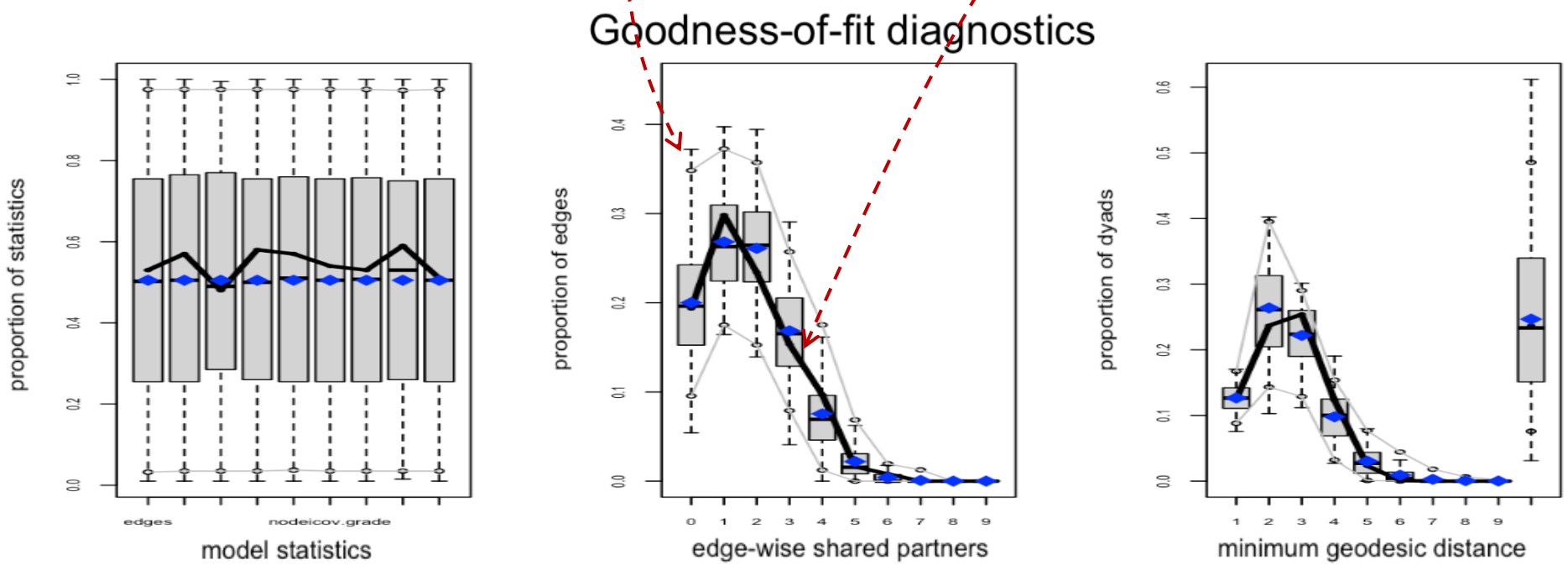


(Rawlings et al., 2023)

Choosing models: GOF plots

The Goodness-of-Fit plots tell you how much the simulated networks follow the structural characteristics (e.g., degree, edgewise shared partners, minimum geodesic distance) of the observed network (solid line).

- You want a close approximation! if the simulations are too off, you need to change the specification of the model (add or remove terms)
- Appropriate GOF plots look more or less like these:



(Rawlings et al., 2023)

5. ERGM coding walkthrough 2

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

ERGM_workshop.R* | QAP_workshop.R

Source on Save | Run | Source

```
1  ######### LISS2108 Statistical Inference for Social Networks Analysis ##########
2  # Author: Santiago Quintero (KCL) - 2025
3
4  # In this workshop, we'll work with the "General Relativity and Quantum Cosmolo
5  # collaboration network" collected by Leskovec et al (2007) (http://doi.acm.org
6  # This network contains all academic collaborations in the domain of GR&QC betw
7  # 1993-2003. It is an undirected network that connects nodes (authors) if they
8  # wrote a paper together. We'll practice fitting ERGMs with this network.
9
10 # Clear workspace and set seed for reproducibility
11 rm(list = ls())
12 set.seed(123)
13 options(scipen = 100) # turn off scientific notation
14
15
16 ## Libraries we'll work with
17 packages <- c("igraph", "network", "dplyr", "ergm", "intergraph")
18
19 # If the packages are not installed, run:
20 # install.packages(packages)
21
22 # Call the libraries
23 lapply(packages, require, character.only = TRUE)
24
25
```

123:1 | Fitting ERGMs | R Script

Environment | History | Connections | Tutorial

Import Dataset | 514 MiB | List

R | Global Environment

ergm.structure	List of 35
net	List of 5
net.igraph	List of 122
Values	
degree_centrality	num [1:122] 6 7 5 2 1 3 1 2 2 4 ...
packages	chr [1:5] "igraph" "network" "dplyr" "ergm" "intergraph"

Console | Terminal | Background Jobs

R 4.4.1 · ~/SANTIAGO/KCL/Inferential SNA/

```
This model was fit using MCMC.  To
examine model diagnostics and check for
degeneracy, use the mcmc.diagnostics()
function.
> summary(ergm.structure)
Call:
ergm(formula = net ~ edges + gwesp(0.5, fixed = TRUE) + gwdegree(0.5,
    fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

              Estimate Std. Error MCMC % z value Pr(>|z|)
edges          -4.6177     0.2184      0 -21.143 < 0.0001 ***
gwesp.fixed.0.5   0.7440     0.1301      0   5.720 < 0.0001 ***
gwdeg.fixed.0.5   1.3046     0.4683      0   2.786  0.00534 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 10232  on 7381  degrees of freedom
 Residual Deviance:  1488  on 7378  degrees of freedom

AIC: 1494  BIC: 1515  (Smaller is better. MC Std. Err. = 0.7177)
>
```

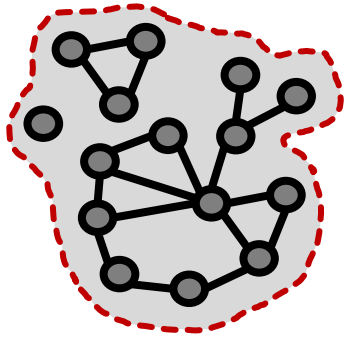
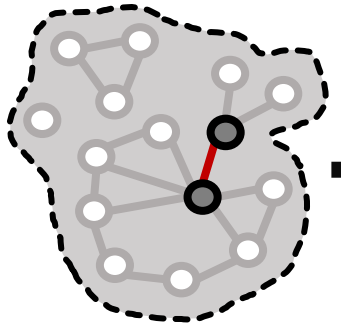
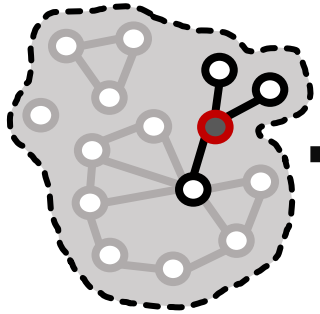
Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish

GR & QC Co-Authorship Network

6. Class summary

Statistical Inference for Networks

		Suggested modelling approach
Group level		• Regression with each group as a case • Use permutation-based statistical test if not a random sample
Density or centralisation		
Number of cliques		
Level of "core-peripheriness"		
Level of average homophily in a group		
Dyad level		• QAP • ERGMs
Presence or absence of a tie between two nodes		
Distance between two nodes		
Number of common connections		
Whether connected nodes are similar (homophily)		
Node level		• (Network/Saptial) Autoregerssive models with each node as a case
Degree, betweenness, closeness centrality		
Level at which a node is similar to its alters (direct connections)		
How much a node fills a structural hole (broker)		
How do node characteristics determine its position in the network		

(Agneessens, 2020)

Bibliography

- Agneessens, F. (2020). Dyadic, nodal, and group-level approaches to study the antecedents and consequences of networks: Which social network models to use and when? In: Light, R., & Moody, J. (Eds.). (2020). *The Oxford Handbook of Social Networks*. Oxford University Press
- Agneessens, F. (2023). *Statistical Analysis for Cross-Sectional and Longitudinal Social Network Analysis*. Essex Summer School in Social Science Data Analysis. University of Essex
- Borgatti, S., Everett, M.G., Johnson, J.C. & Agneessens, F. (2022). *Analyzing Social Networks Using R*. SAGE Publications
- Cranmer, S. J., Leifeld, P., McClurg, S. D., & Rolfe, M. (2017). Navigating the Range of Statistical Tools for Inferential Network Analysis. *American Journal of Political Science, 61(1)*, 237–251
- Dekker, D., Krackhardt, D. & Snijders, T.A.B. (2007). Sensitivity of MRQAP Tests to Collinearity and Autocorrelation Conditions. *Psychometrika, 72*, 563-81
- Handcock, M. (2003). *Assessing Degeneracy in Statistical Models of Social Networks*. Working Paper n0.39 Center for Statistics and the Social Sciences, University of Washington, Seattle.
- Harris, J. K. (2014). *An introduction to exponential random graph modeling*. Thousand Oaks: SAGE
- Power, E. (2021). *MY461. Social Network Analysis*. Department of Methodology, London School of Economics and Political Science
- Rawlings, C. M., Smith, J.A., Moody, J. & McFarland, D.A. (2023). *Network Analysis: Integrating Social Network Theory, Method, and Application with R*. New York: Cambridge University Press.
- Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007a). An introduction to exponential random graph (p*) models for social networks. *Social Networks, 29*(2), 173–191
- Robins, G., Snijders, T., Wang, P., Handcock, M. & Pattison, P. (2007b). Recent developments in exponential random graph (p*) models for social networks. *29*(2), 192–215