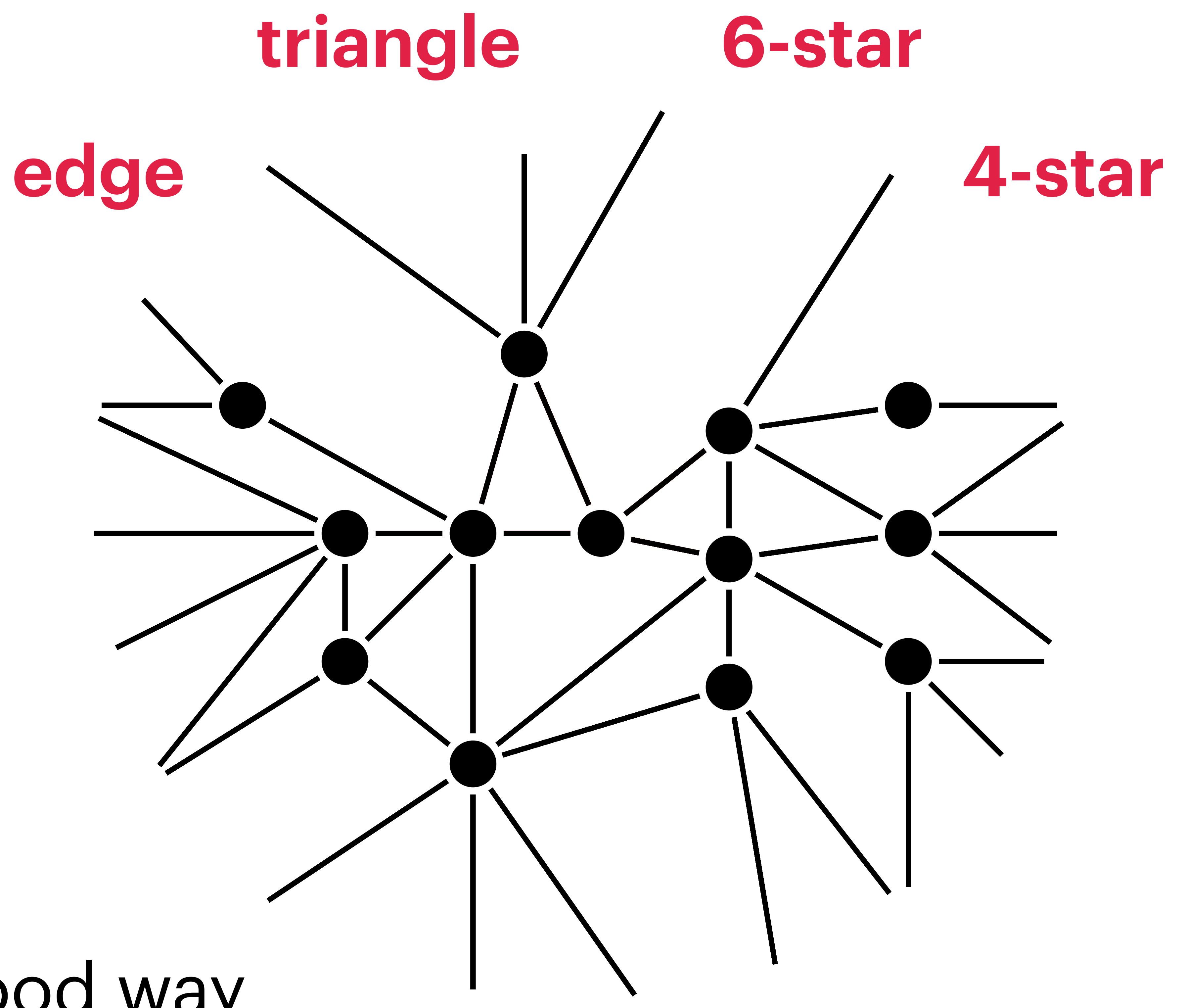
Exponential Random Graph Models

beyond the basic random graph models

network configurations

- are nested in each other
- represent competing explanatory mechanisms

how much of each mechanism/effect do we need to reproduce the structure?



we need a model that can

- control for more than one network feature at a time
- that can be parametrized to represent effects in a good way (not extreme when the model is simulated)

$\longrightarrow$ **Exponential Random Graph Models (ERGMs)**

can provide such models for many configurations relevant to social network theory

four generations of dependence assumptions

- **Bernoulli dependence**

  network variables are independent of each other

- **dyadic dependence**

  dependence within dyads for directed networks

- **Markov dependence**

  network variables are conditionally dependent if they share at least one node

- **social circuit dependence**

  network variables are conditionally dependent if they create 4-cycles

increasing level of nested subgraphs

[we will look at exponential random graph models (ERGMs) specified to include these dependence assumptions]

exponential random graph models (intuitively)

ERGMs are a class of random graph models

the probability of a graph is a function of two components

(1) set of network characteristics (statistics) that may influence the probability of a graph

- choice of statistics often motivated by social science theory

(2) set of parameters (associated with statistics)

- determine how network statistics increase/decrease graph probabilities
- estimated from observed network to test hypotheses

$$\boxed{P(G) = \frac{1}{\kappa} \exp\left(\sum_{i=1}^{p} \theta_i \cdot s_i \right)}$$

exponential random graph models (intuitively)

normalizing constant

$$\kappa = \sum_{G'} \exp\left(\sum_{i=1}^{p} \theta_i \cdot s_i(G') \right)$$

needed to generate full outcomes space for all possible graphs (G')
causes much difficulty in estimation of parameters

interpreting parameter estimates

- **positive** value indicate more configurations in the observed network than expected by chance
- **negative** value indicate fewer configurations in the observed network than expected by chance

$$\boxed{P(G) = \frac{1}{\kappa} \exp\left(\sum_{i=1}^{p} \theta_i \cdot s_i \right)}$$

exponential random graph models

ERGM modelling outline

(1) specify model parameters that govern graph evolution
  - reciprocity, transitivity, homophily, etc.

(2) simulation and estimation
  - maximum likelihood estimation
  - simulate other random networks based on this model
  - Markov Chain Monte Carlo (MCMC) algorithms:
    - generate sample of random networks following model rules
    - resulting networks should resemble the observed network

(3) compare goodness of fit of observed to modelled networks
  - is the model a good fit for the data?

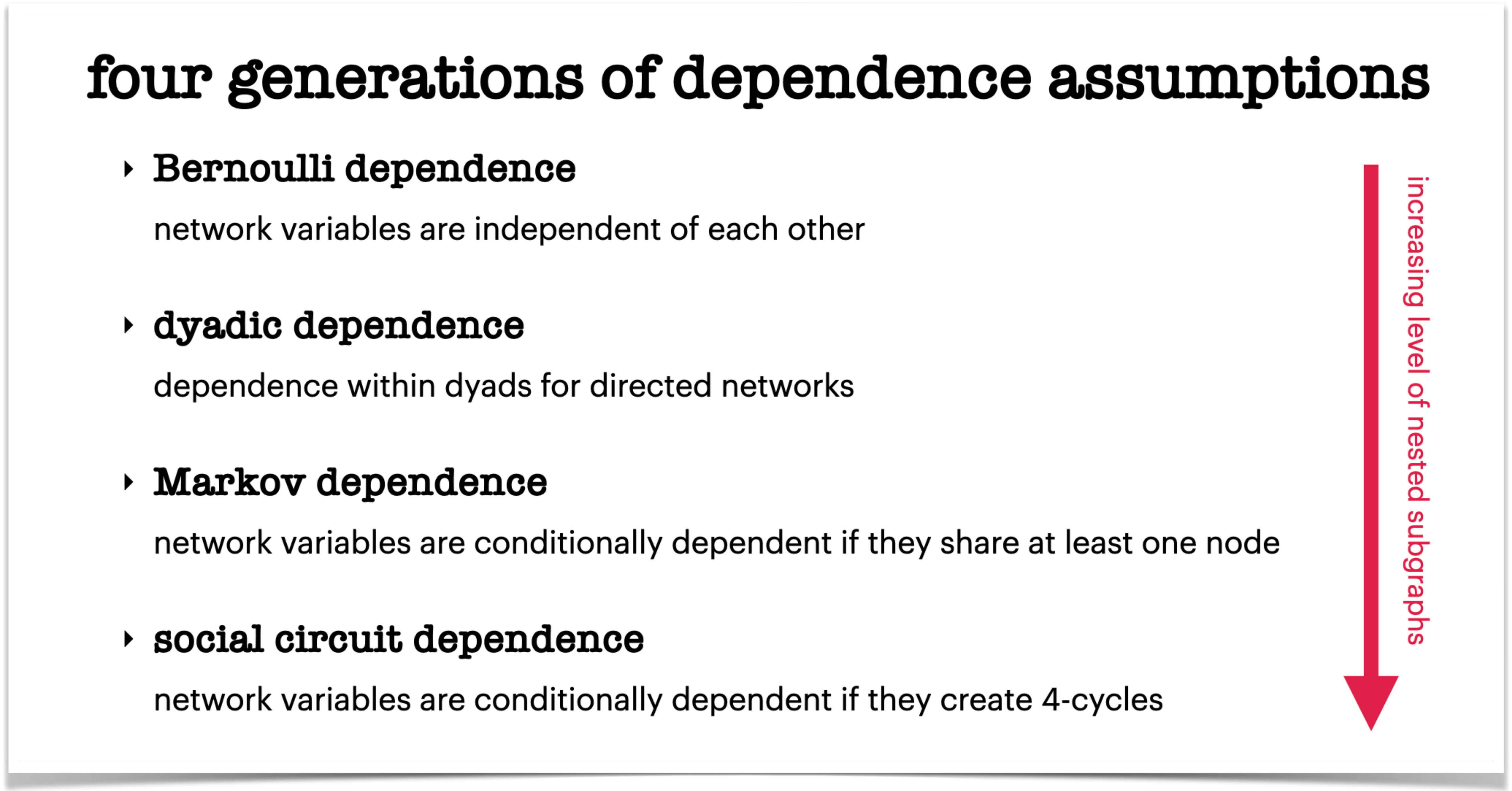
(1) specify model parameters that govern graph evolution

how do we specify the different statistics (s_i) algebraically in the model specification?

let y_{uv} denotes single possible edges between nodes u and v in the graph:

$$y_{uv} = \begin{cases} 1 \text{ if tie between } u \text{ and } v \\ 0 \text{ otherwise} \end{cases}$$

(as given in adjacency the matrix of graph G)



four generations of dependence assumptions

- **Bernoulli dependence**
  network variables are independent of each other
- **dyadic dependence**
  dependence within dyads for directed networks
- **Markov dependence**
  network variables are conditionally dependent if they share at least one node
- **social circuit dependence**
  network variables are conditionally dependent if they create 4-cycles

increasing level of nested subgraphs

specify statistics in ERGMs based on these dependence assumptions

we will assume undirected graphs
unless clearly stated otherwise

(1) specify model parameters that govern graph evolution

Bernoulli dependence

possible edges are independent of one another

configurations/statistics in this model relate to single possible edges

model specification (a first attempt)

$$P(G) = \frac{1}{\kappa} \exp\left(\sum_{u<v} \theta_{uv} y_{uv} \right)$$

but this means we have to estimate one parameter θ for each edge $\Longrightarrow$ too many!

homogeneity assumption

assumes that the edge effect is the same across the entire network

that is $\theta_{uv} = \theta$ for all (u, v)

(1) specify model parameters that govern graph evolution

Bernoulli dependence

possible edges are independent of one another

configurations/statistics in this model relate to single possible edges

model specification modified by homogeneity assumption:

$$P(G) = \frac{1}{\kappa} \exp\left(\sum_{u<v} \underbrace{\theta_{uv}}_{=\theta} y_{uv} \right) = \frac{1}{\kappa} \exp\left(\sum_{u<v} \theta y_{uv} \right) = \frac{1}{\kappa} \exp\left(\theta \underbrace{\sum_{u<v} y_{uv}}_{=L} \right) = \frac{1}{\kappa} \exp\left(\theta L\right)$$

where L is the only statistic (s_i) in the model and θ is an edge/density parameter to be estimated

all ERGMs start with the edge statistic!

(1) specify model parameters that govern graph evolution

Bernoulli dependence

statistic L counts the number of edges

positive (**negative**) parameter associated with L **increases** (**decreases**) the expected density

the Bernoulli graph $\mathscr{G}(n, p)$ belongs to the ERGM class

$\mathscr{G}(n, p)$ is identical with the ERGM defined by

$$P(G) = \frac{1}{\kappa} \exp\left(\theta \cdot L\right)$$

relation between θ and p:

- $\theta < 0 \iff$ expected density $p < 0.5$
- $\theta = 0 \iff$ expected density $p = 0.5$
- $\theta > 0 \iff$ expected density $p > 0.5$

does not hold in general (if the ERGM contains other statistics)

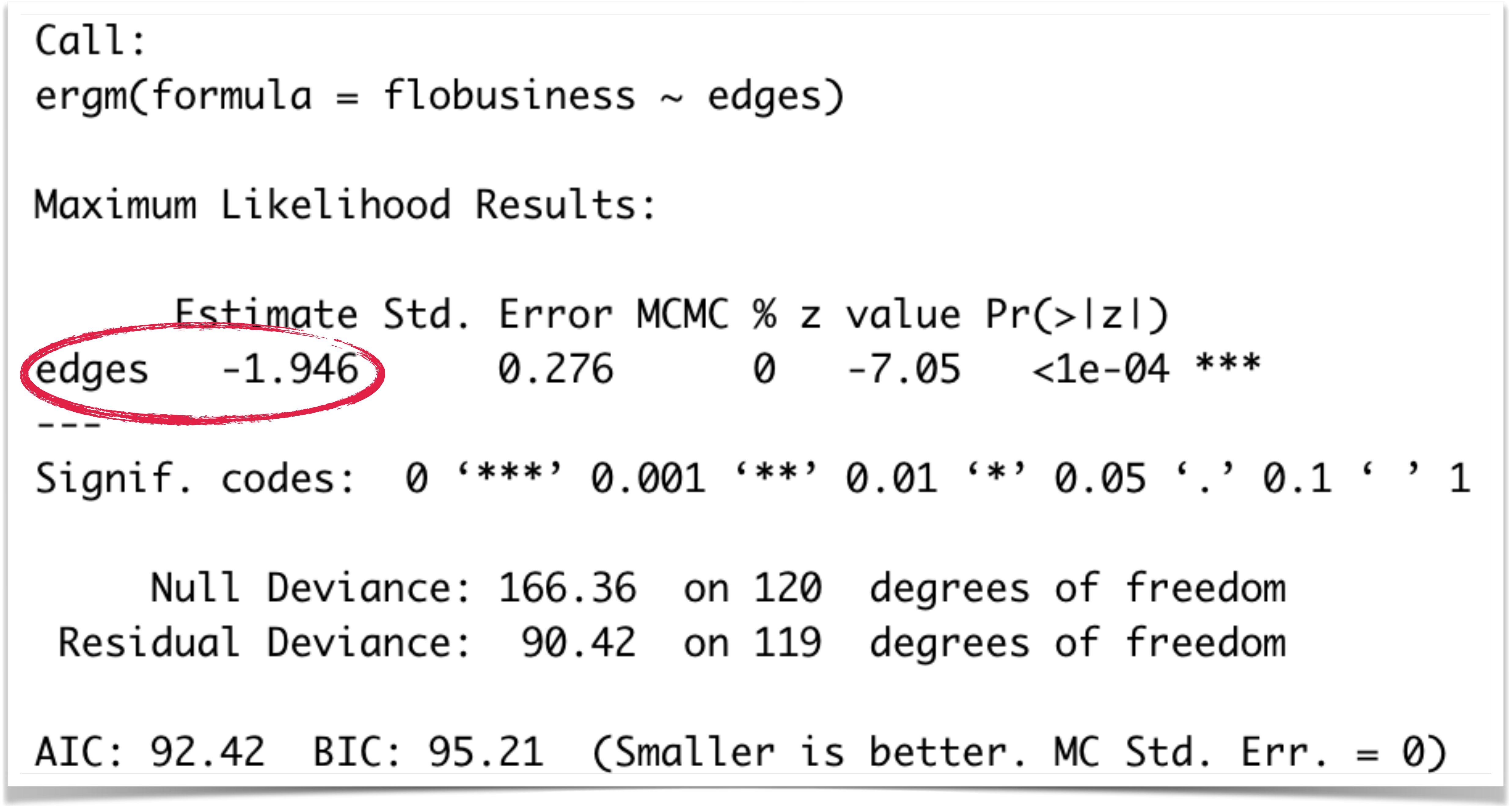
(1) specify model parameters that govern graph evolution

Bernoulli dependence

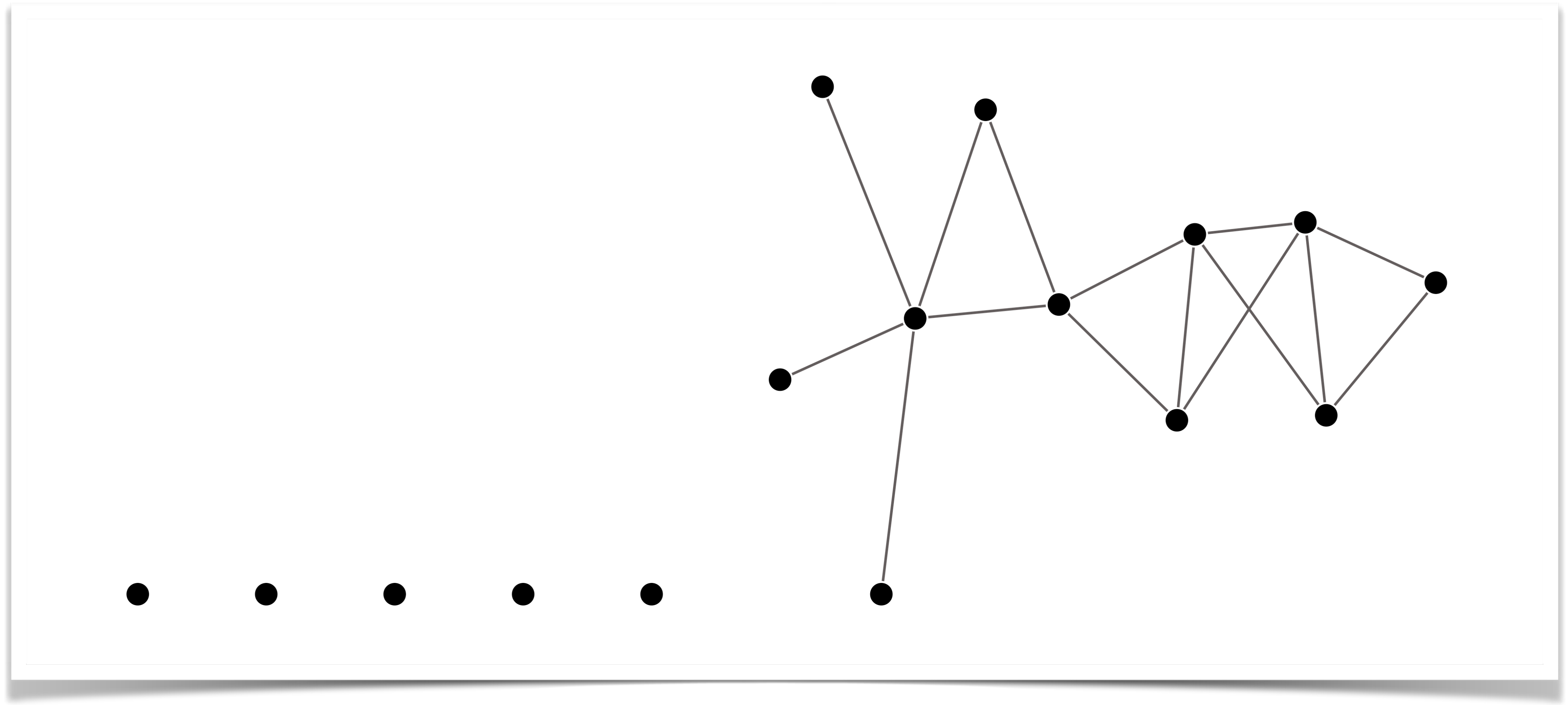
example. Florentine business network

density of network $\hat{p} =$ 15/120 = 0.125

if an ERGM with **only** edges is specified then $\hat{\theta} < 0$



```
Call:
ergm(formula = flobusiness ~ edges)

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges   -1.946      0.276      0   -7.05   <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 166.36  on 120  degrees of freedom
 Residual Deviance:  90.42  on 119  degrees of freedom

AIC: 92.42  BIC: 95.21  (Smaller is better. MC Std. Err. = 0)
```



we have fewer edges in the observed network than what is expected by chance

(1) specify model parameters that govern graph evolution

homophily and heterophily

another commonly used network statistics implying dyadic independence

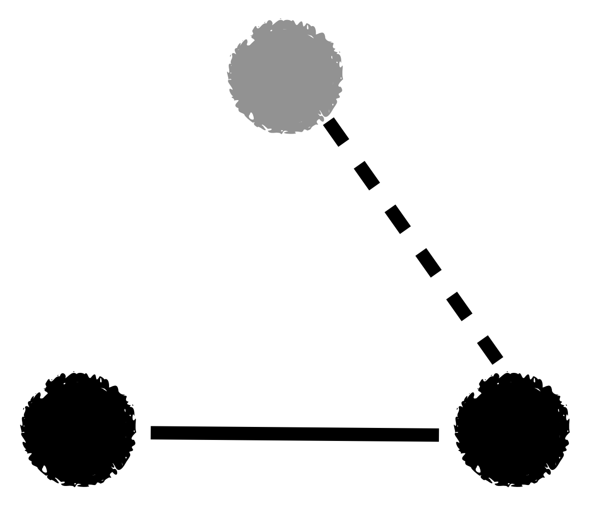
assume actors have attribute values $a : V \rightarrow \{1, \ldots, c\}$

for example such age, gender, income, behaviour, attitude, nationality, religion, etc.

let statistic

$$m_a(G) = |\{\{u, v\} \in E : a(u) = a(v)\}|$$

count number of ties connecting actors with same attribute value



positive (**negative**) parameter models the tendency **for** (**against**) creating edges to similar actors **homophily** (**heterophily**)

(1) specify model parameters that govern graph evolution

dyadic dependence

for **directed graphs**: dependence within dyads

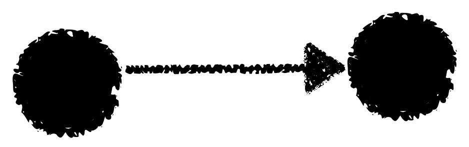
we can include a statistic that counts the number of ordered node-pairs with ties (u, v) and ties (v, u) $\implies$ **reciprocity**

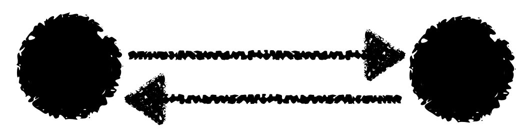
model specification

an ERGM with an edge statistic L and reciprocity statistic R is specified as

$$P(G) = \frac{1}{\kappa} \exp\left(\theta_1 \cdot L + \theta_2 \cdot R\right)$$

where $L = \sum_{u \neq v} y_{uv}$ and $R = \sum_{u \neq v} y_{uv} \cdot y_{vu}$





positive (negative) parameter models the tendency for (against) reciprocating ties

(1) specify model parameters that govern graph evolution

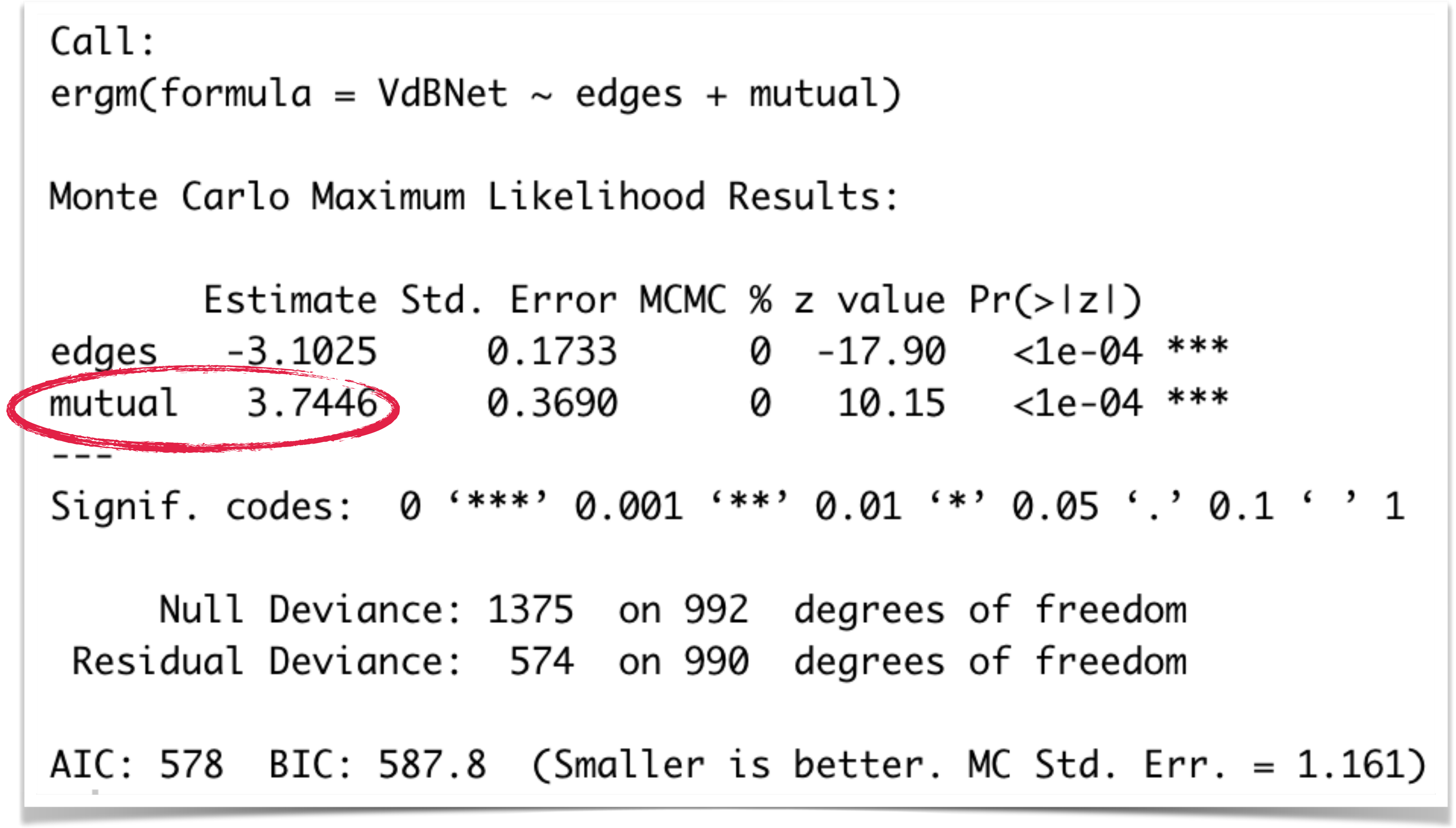
dyadic dependence

for **directed graphs**: dependence within dyads

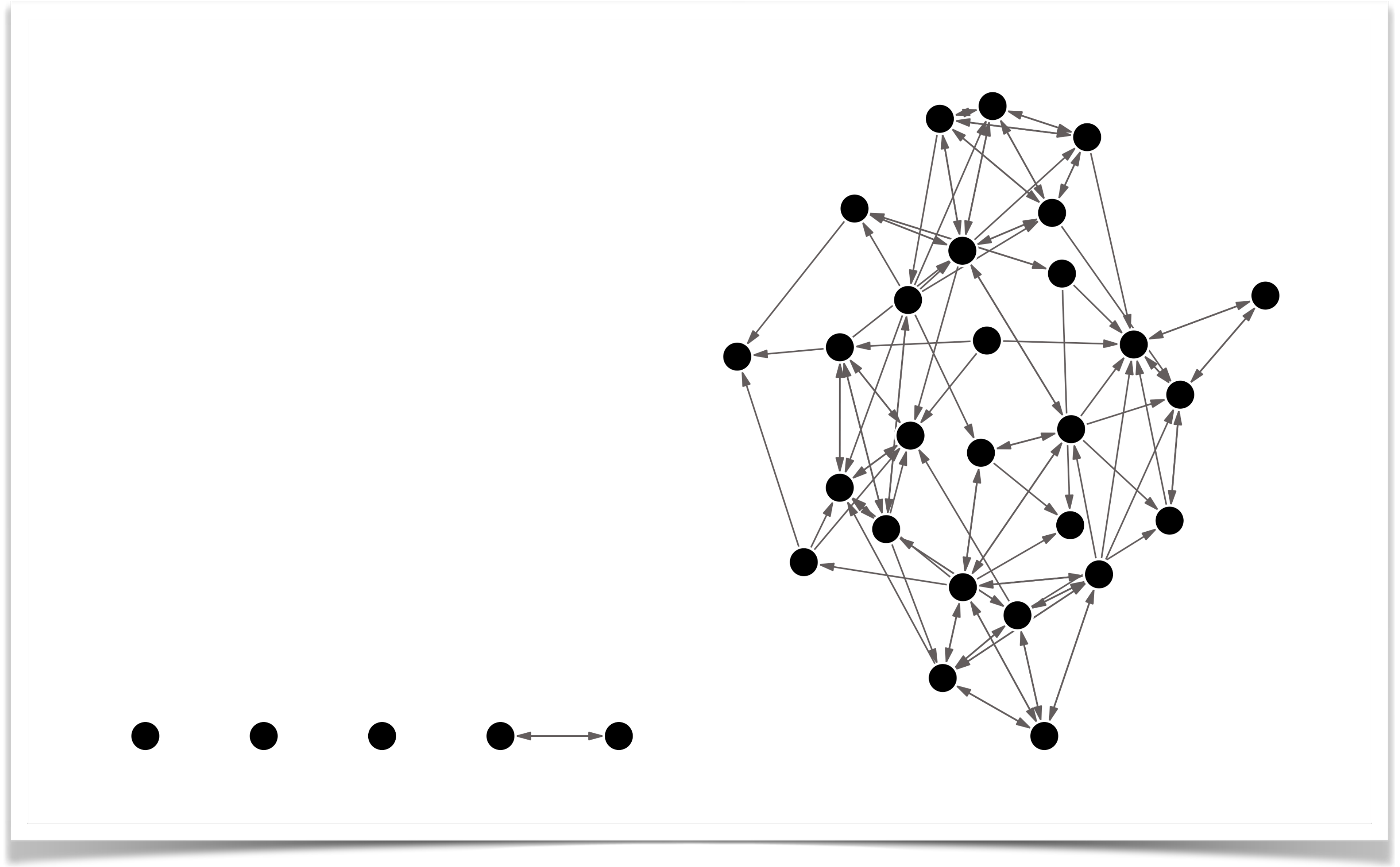
example. friendship among university freshmen (Van de Bunt, 1999)

$$\boxed{P(G) = \frac{1}{\kappa} \exp\left(\theta_1 \cdot L + \theta_2 \cdot R\right)}$$

mutual	asymmetric	null
36	38	422



```
Call:
ergm(formula = VdBNet ~ edges + mutual)

Monte Carlo Maximum Likelihood Results:

       Estimate Std. Error MCMC % z value Pr(>|z|)
edges   -3.1025     0.1733      0  -17.90   <1e-04 ***
mutual   3.7446     0.3690      0   10.15   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 1375  on 992  degrees of freedom
 Residual Deviance:  574  on 990  degrees of freedom

AIC: 578  BIC: 587.8  (Smaller is better. MC Std. Err. = 1.161)
```



more mutual ties in the observed network than what is expected by chance

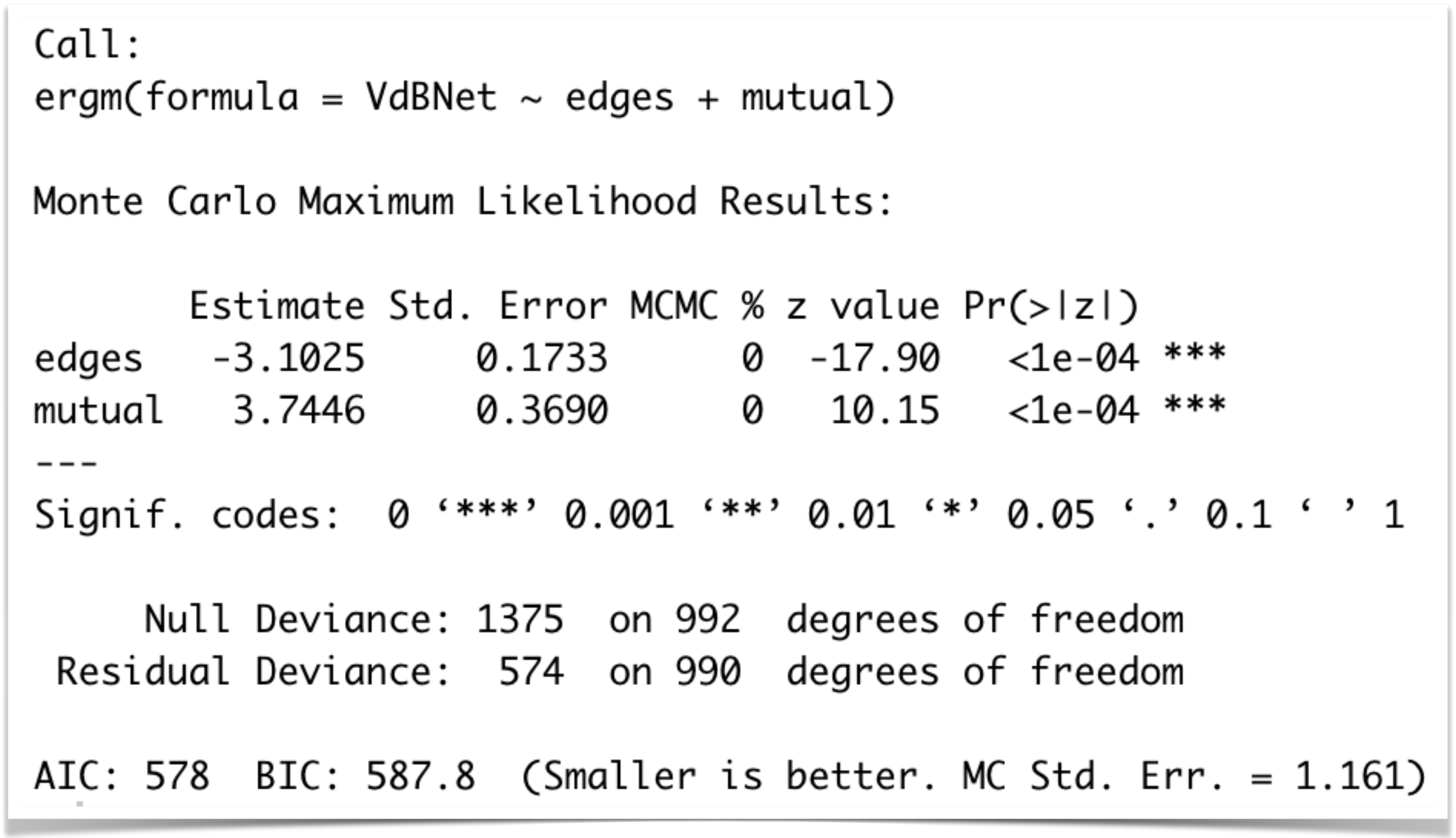
(1) specify model parameters that govern graph evolution

dyadic dependence

for **directed graphs**: dependence within dyads

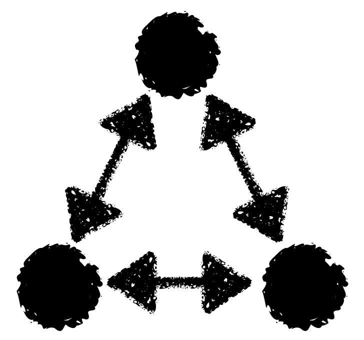
example. friendship among university freshmen (Van de Bunt, 1999)

$$\boxed{P(G) = \frac{1}{\kappa} \exp\left(\theta_1 \cdot L + \theta_2 \cdot R\right)}$$

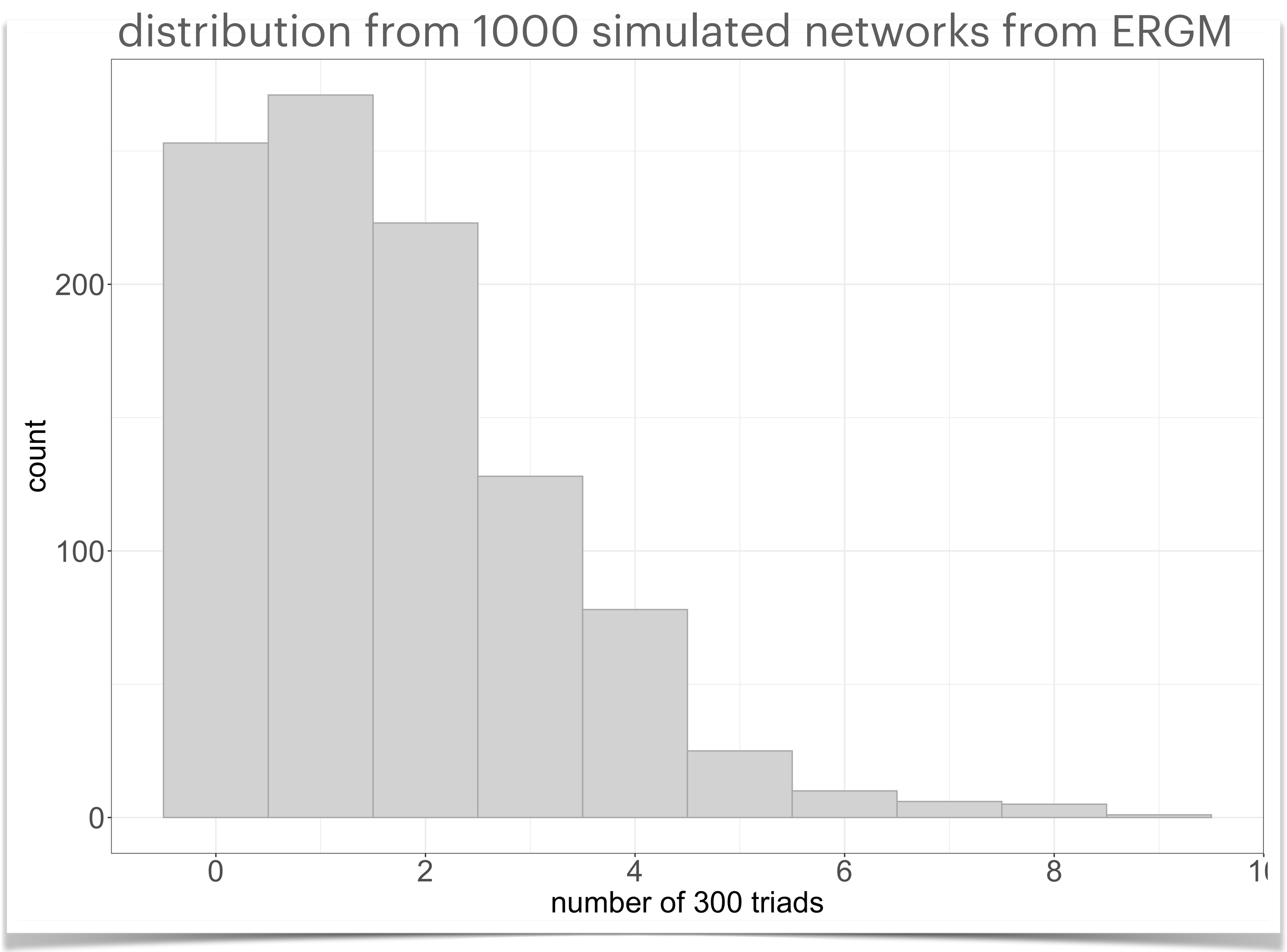


```
Call:
ergm(formula = VdBNet ~ edges + mutual)

Monte Carlo Maximum Likelihood Results:

       Estimate Std. Error MCMC % z value Pr(>|z|)
edges   -3.1025     0.1733      0  -17.90   <1e-04 ***
mutual   3.7446     0.3690      0   10.15   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 1375  on 992  degrees of freedom
 Residual Deviance:  574  on 990  degrees of freedom

AIC: 578  BIC: 587.8  (Smaller is better. MC Std. Err. = 1.161)
```

what about transitivity?



number of observed 300 triads $= 18$



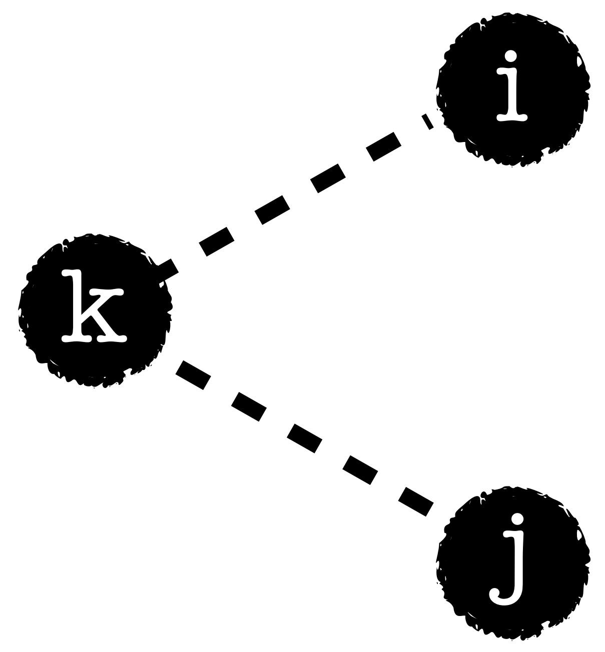
(1) specify model parameters that govern graph evolution

Markov dependence

Frank & Strauss (1986) drew on the work of Besag (1974) in spatial statistics

- **the Hammersley-Clifford theorem:**
  sets out constraints on model form implied by dependence assumptions
- they proposed a network dependence assumption
  Markov dependence

two ties are conditionally independent unless they share a node



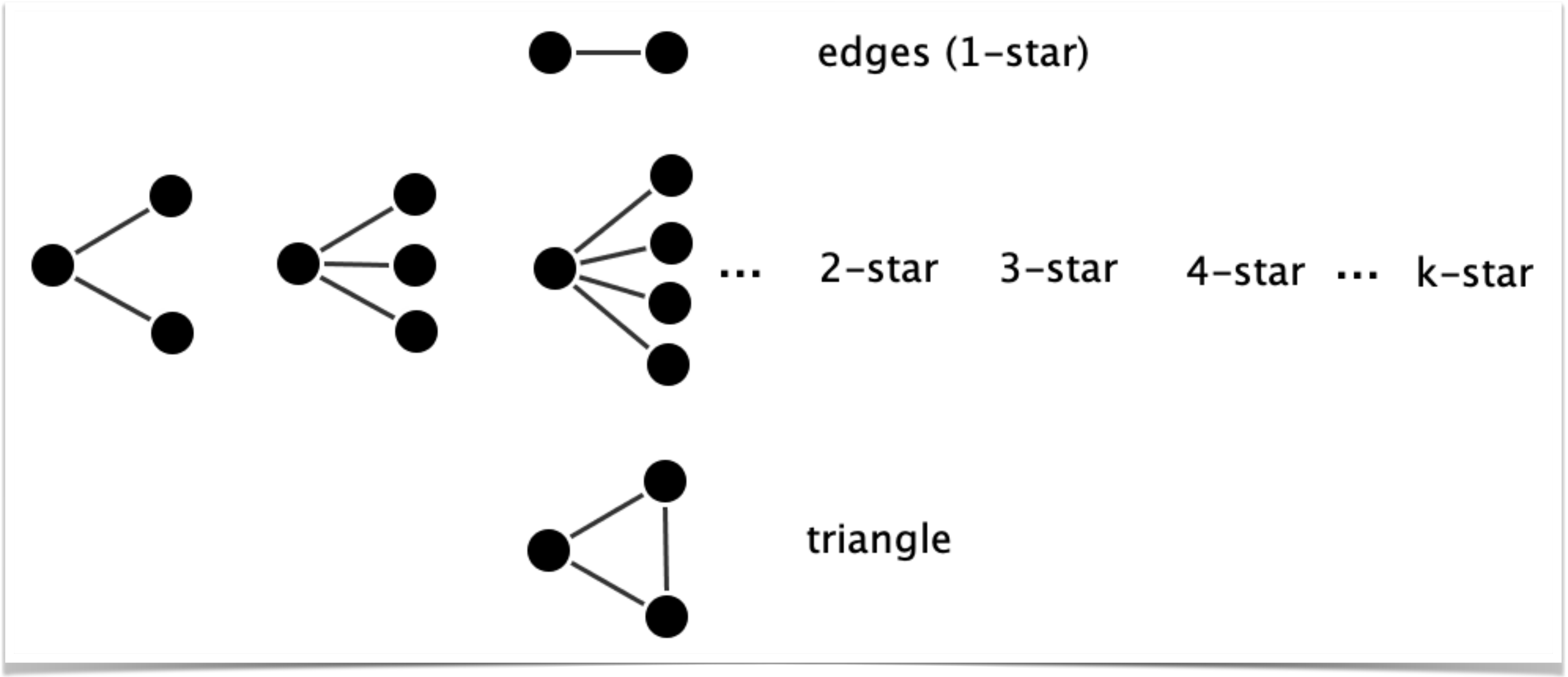
(1) specify model parameters that govern graph evolution

Markov dependence

Markov random graphs

suppose edges are conditionally dependent if and only if they share a node

configurations in such a model comprise edges, stars and triangles:



(1) specify model parameters that govern graph evolution

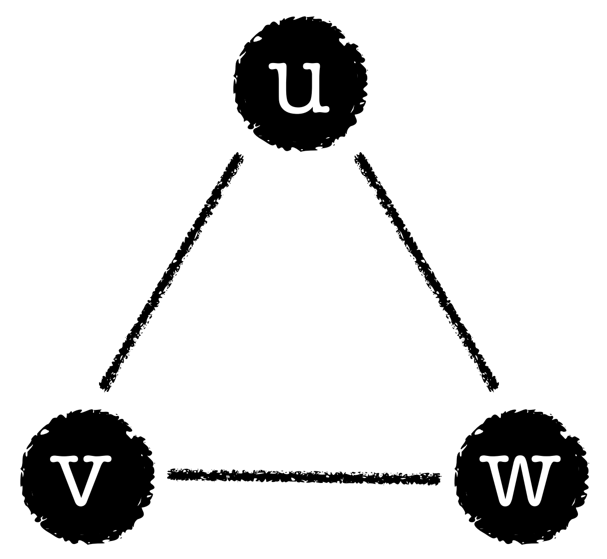
Markov dependence

Markov random graphs

statistic T counts the number of **triangles** in G

$$T = \sum_{u<v<w} y_{uw} \cdot y_{vw} \cdot y_{vu}$$

positive (**negative**) parameter models a **preference** (**reluctance**) to triadic closure (transitivity)



'friend of a friend is a friend'

for directed networks: distinguish between transitive and cyclic triads

(1) specify model parameters that govern graph evolution

Markov dependence

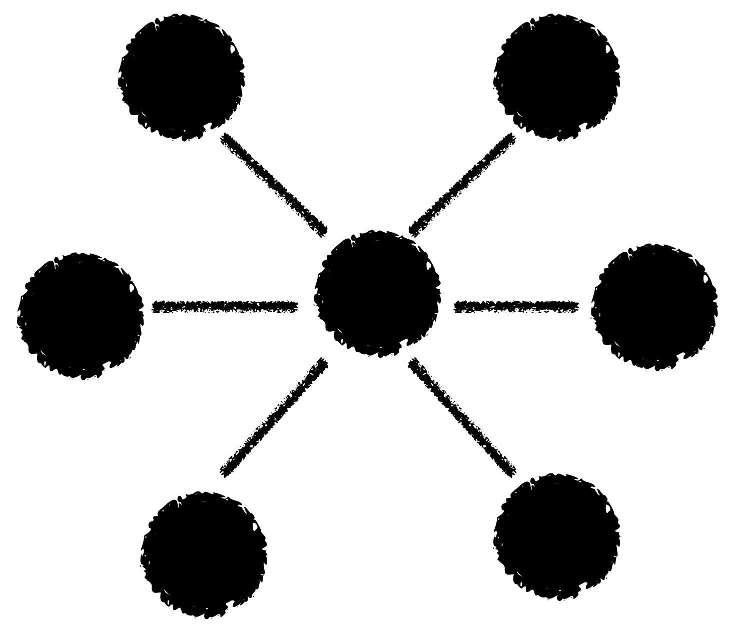
$k-$stars

for $k = 2,\ldots,n-1$ statistic S_k counts the number of $k-$stars

$$S_k = \sum_{u} \sum_{v_1<\cdots<v_\ell\neq u} y_{uv_1}\cdot\cdots\cdot y_{uv_k}$$

positive (negative) parameter models tendency for (against) connecting to high-degree nodes

(the propensities for/against individuals to have connections with multiple network partners)



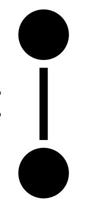
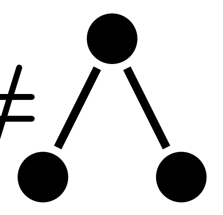
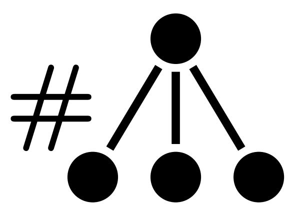
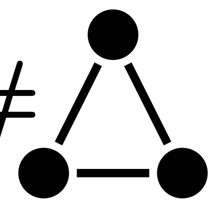
for directed networks:
distinguish between out-stars and in-stars and effects such as 'popularity' and ‘activity

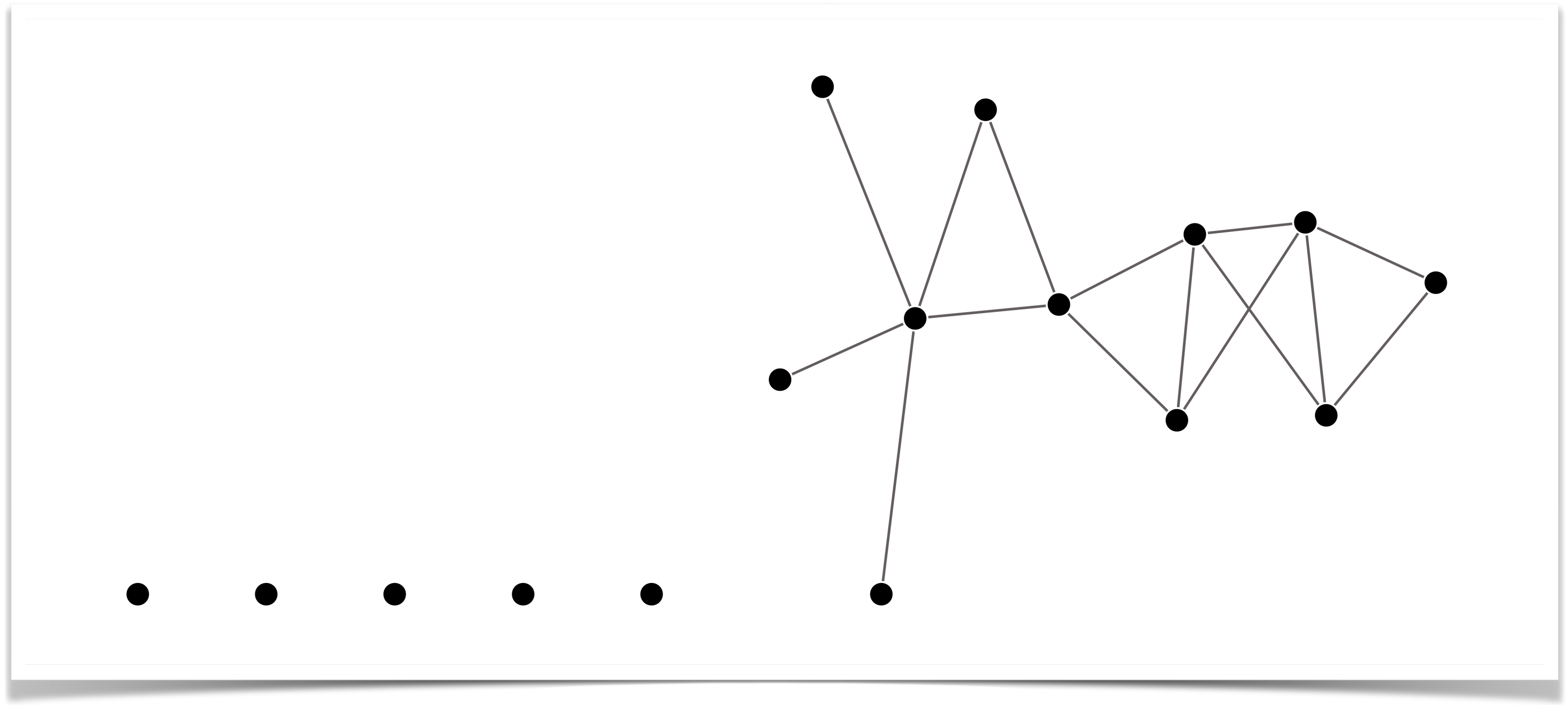
(1) specify model parameters that govern graph evolution

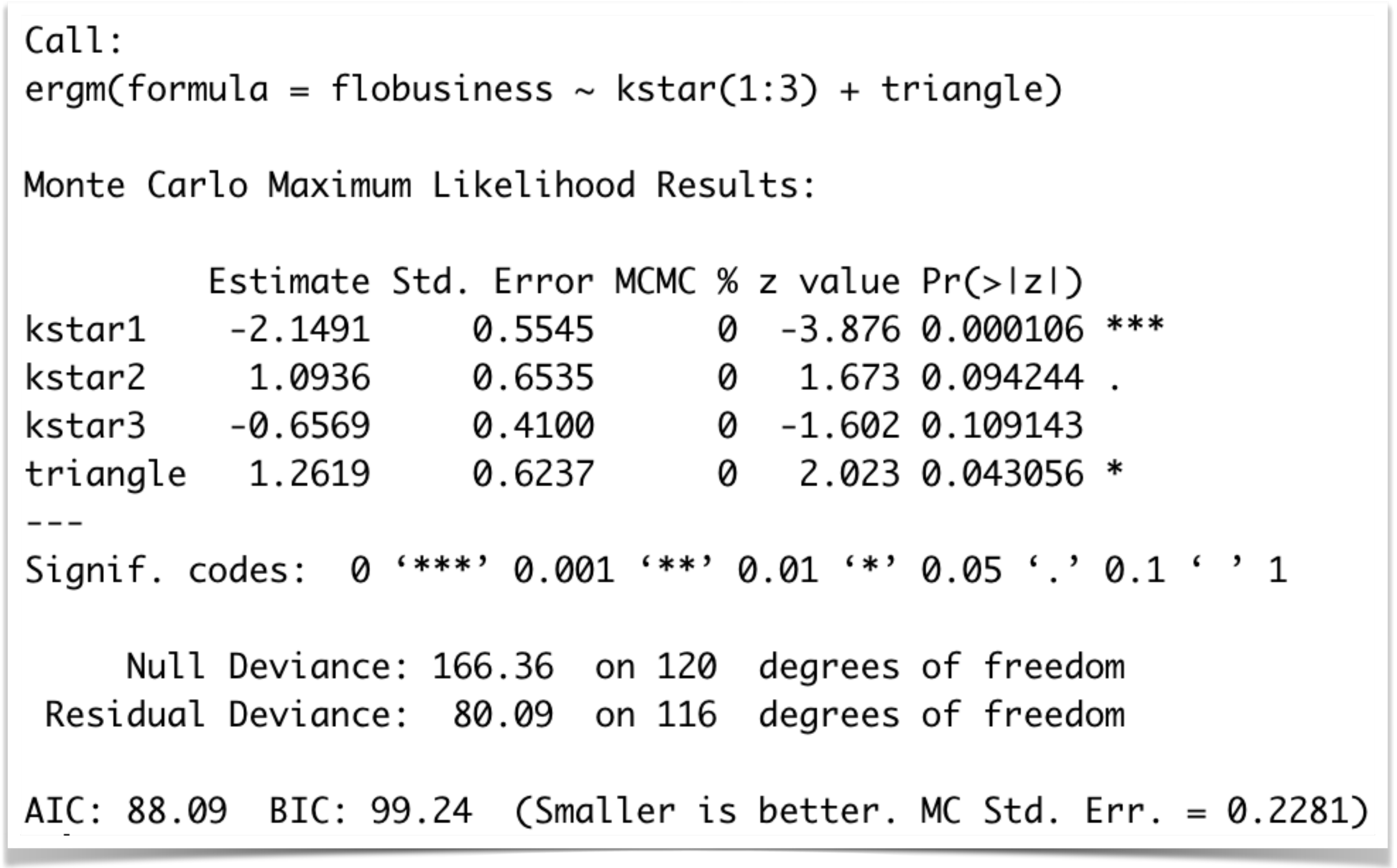
Markov dependence

example. Florentine business network

$$P(G) = \frac{1}{\kappa} \exp\left(\theta_1 \cdot L + \theta_2 \cdot S_2 + \theta_3 S_3 + \theta_4 T\right)$$

$\approx \theta_1 \times \#$  $+ \; \theta_2 \times \#$  $+ \; \theta_3 \times \#$  $+ \; \theta_4 \times \#$ 





```
Call:
ergm(formula = flobusiness ~ kstar(1:3) + triangle)

Monte Carlo Maximum Likelihood Results:

          Estimate Std. Error MCMC % z value Pr(>|z|)
kstar1     -2.1491     0.5545      0  -3.876 0.000106 ***
kstar2      1.0936     0.6535      0   1.673 0.094244 .
kstar3     -0.6569     0.4100      0  -1.602 0.109143
triangle    1.2619     0.6237      0   2.023 0.043056 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 166.36  on 120  degrees of freedom
 Residual Deviance:  80.09  on 116  degrees of freedom

AIC: 88.09  BIC: 99.24  (Smaller is better. MC Std. Err. = 0.2281)
```

positive (negative) parameter associated with a configuration means we observe more (less) of that configuration than expected by chance

significance of estimates:

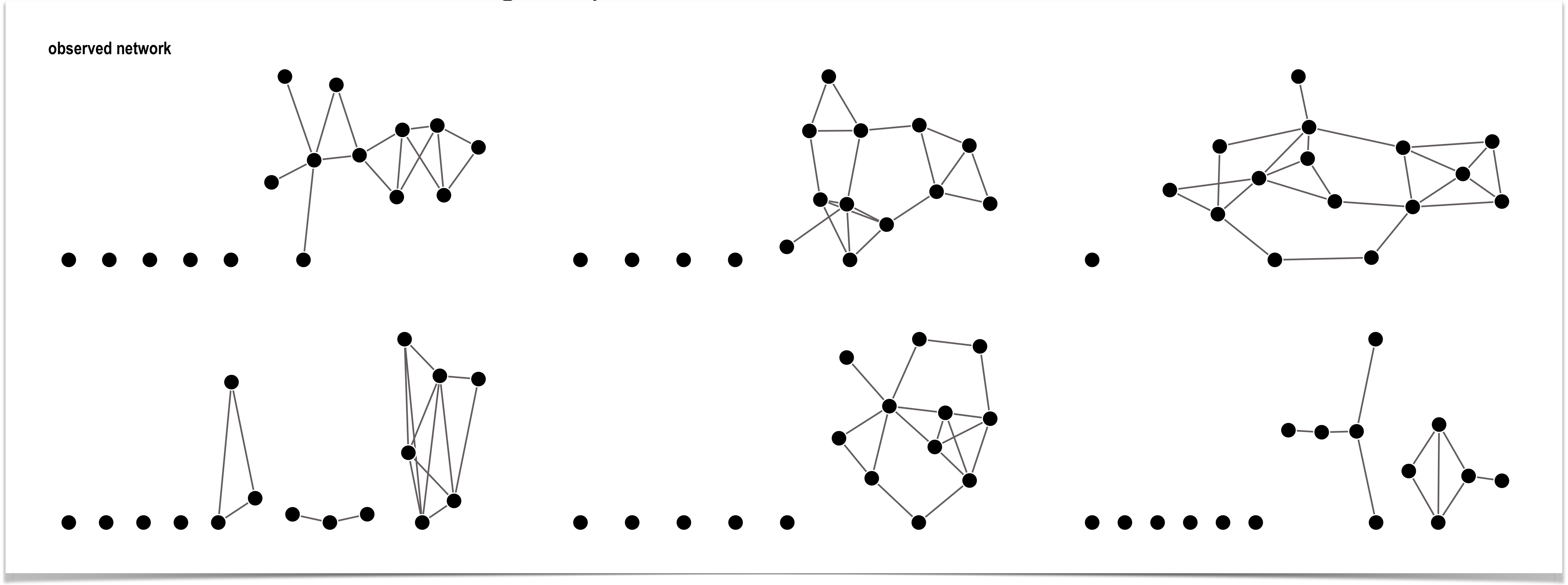
|Estimate| > 2 x `Std. Error`

(1) specify model parameters that govern graph evolution

Markov dependence

example. Florentine business network

simulated networks according to specified ERGM



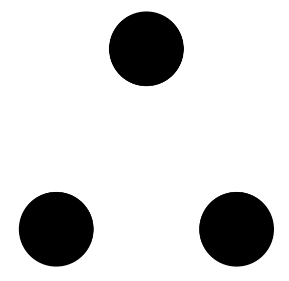
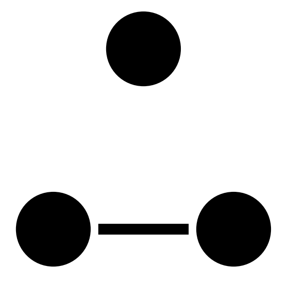
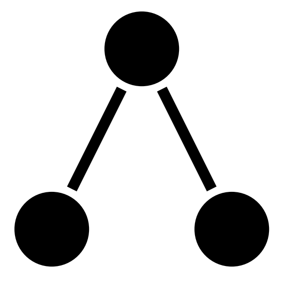
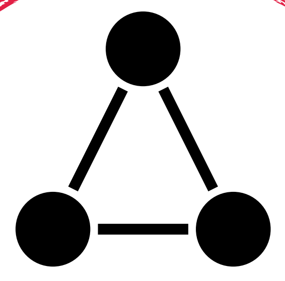
(1) specify model parameters that govern graph evolution

Markov dependence

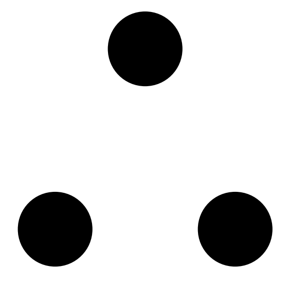
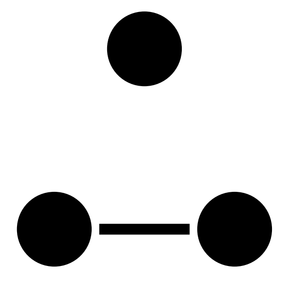
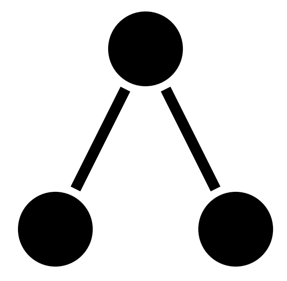
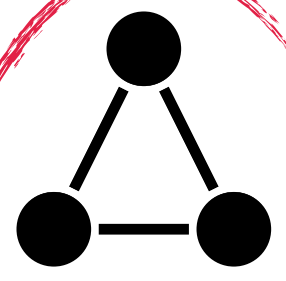
example. Florentine business network

triad census

observed network

			
381	153	21	5

simulated networks

			
335	191	27	7
281	229	43	7
363	179	9	9
362	164	28	6
432	118	8	2

we will return to the goodness of fit of this model later...

(1) specify model parameters that govern graph evolution

Markov dependence

the good news:

Markov random graph distributions provide the possibility to model plausible assumptions

the bad news:

- they don't always work
- some parameter values give not coherent models: two/more entirely different graphs
- difficult estimating when clustering is high
- estimating an edge/2-star/3-star/triangle model $\Longrightarrow$ very bad convergence ratios

☠ **model degeneracy** ☠

more on this later...

(1) specify model parameters that govern graph evolution

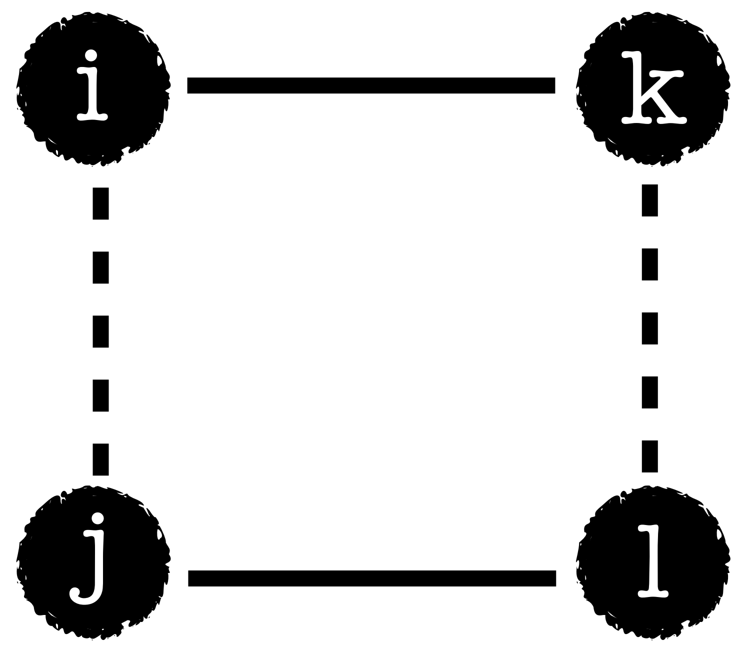
Markov dependence

‘New specifications of ERGMs’

- network ties self-organise within 4–cycles (Pattison & Robins, 2002)
- configurations that may avoid degeneracy

social circuit dependence

two possible network ties are conditionally dependent if they would form a 4–cycle



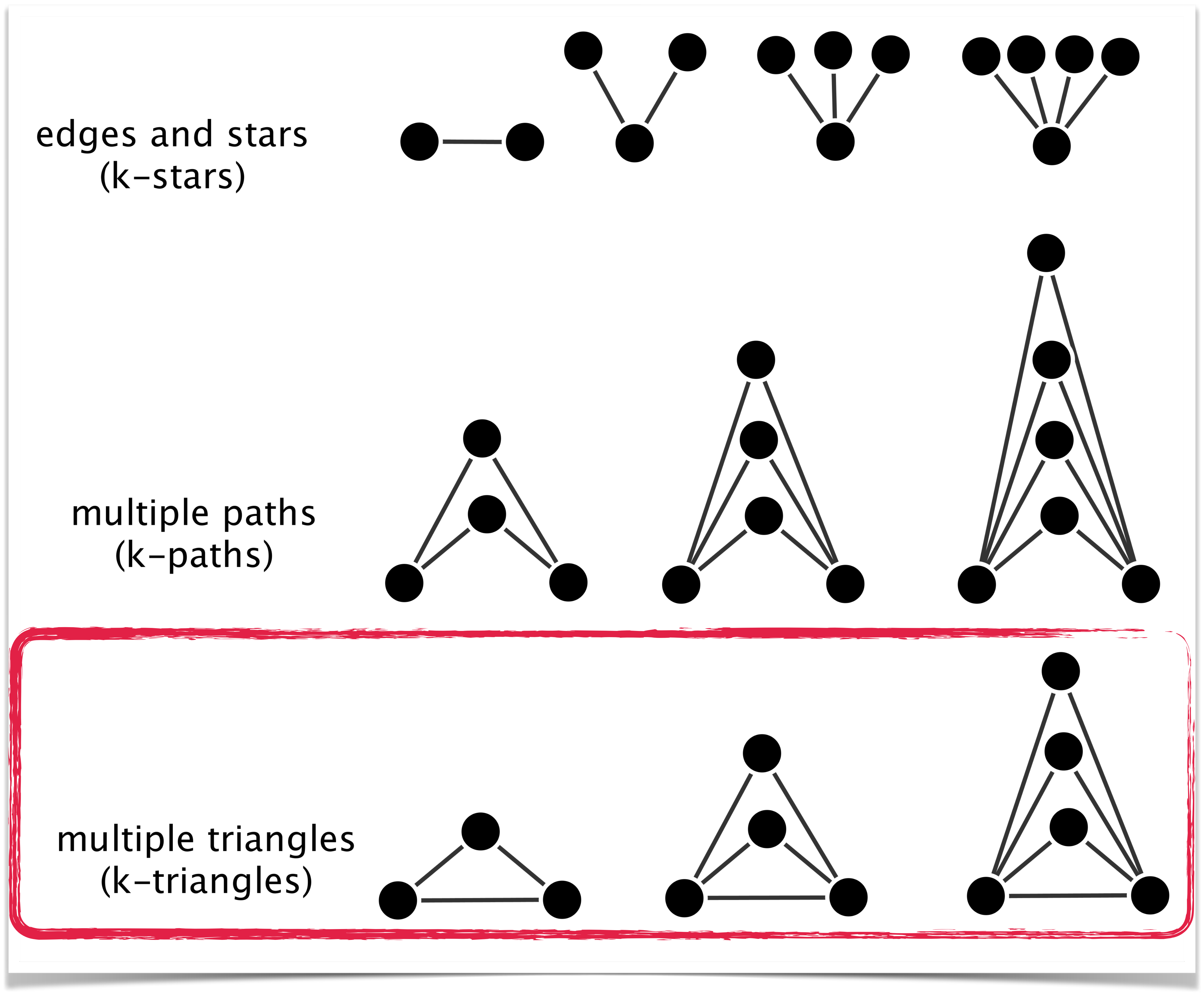
tie variables (i, j) and (k, l) are conditionally independent, given the rest of the graph, unless the existence of these two ties would imply a 4-cycle in the graph

(1) specify model parameters that govern graph evolution

Markov dependence

‘New specifications of ERGMs’

parameters correspond to configurations of the following types:



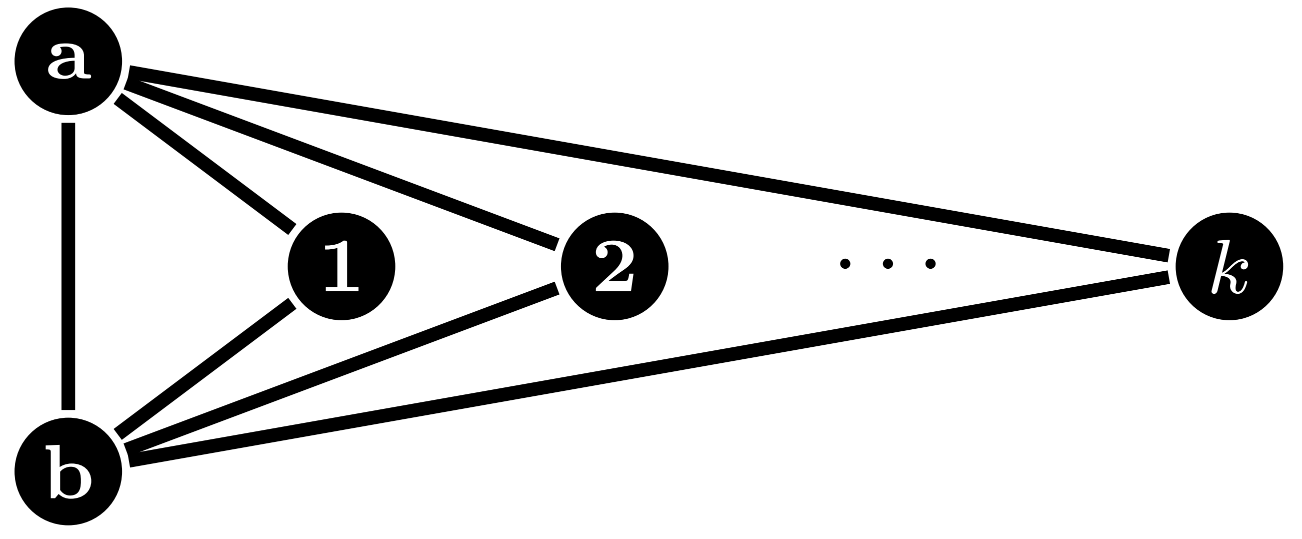
(1) specify model parameters that govern graph evolution

social circuit dependence

‘New specifications of ERGMs’

(geometrically weighted) edgewise shared partners

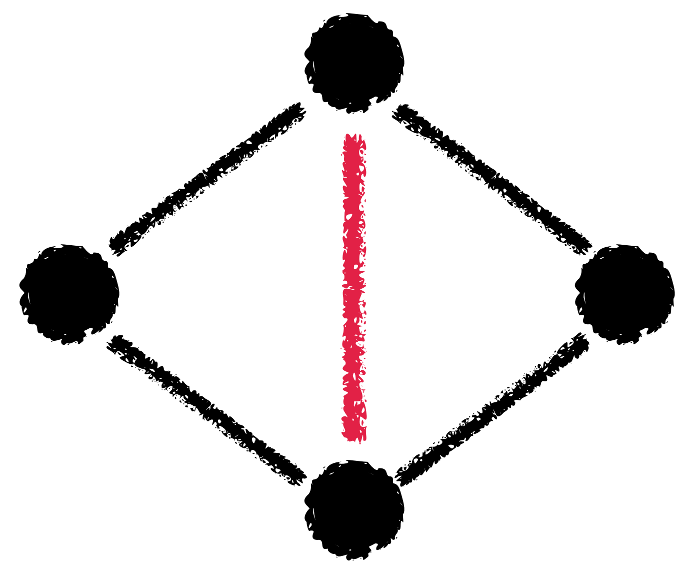
gwesp



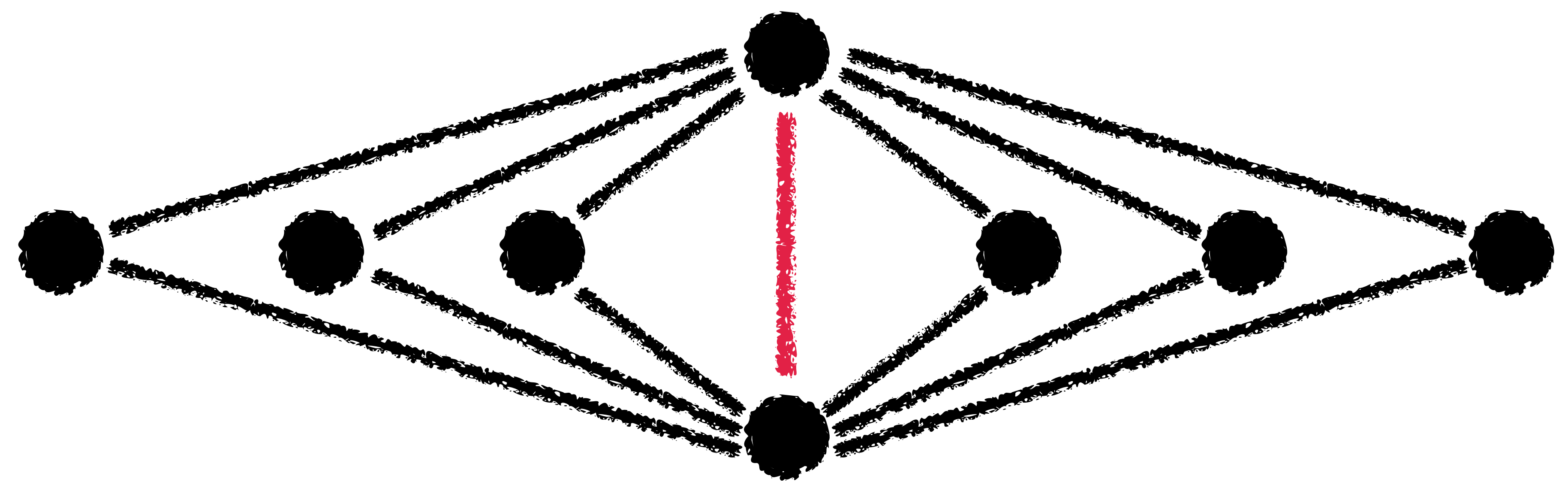
`gwesp` is

- a measure of higher order clustering
- a special effect that measures triadic closure in a network
- helps avoid degeneracy:
  - having one shared friend makes a tie 25% more likely,
  - having six shared friends makes a tie 150% more likely (!)

one edge completes two triangles



one edge completes six triangles

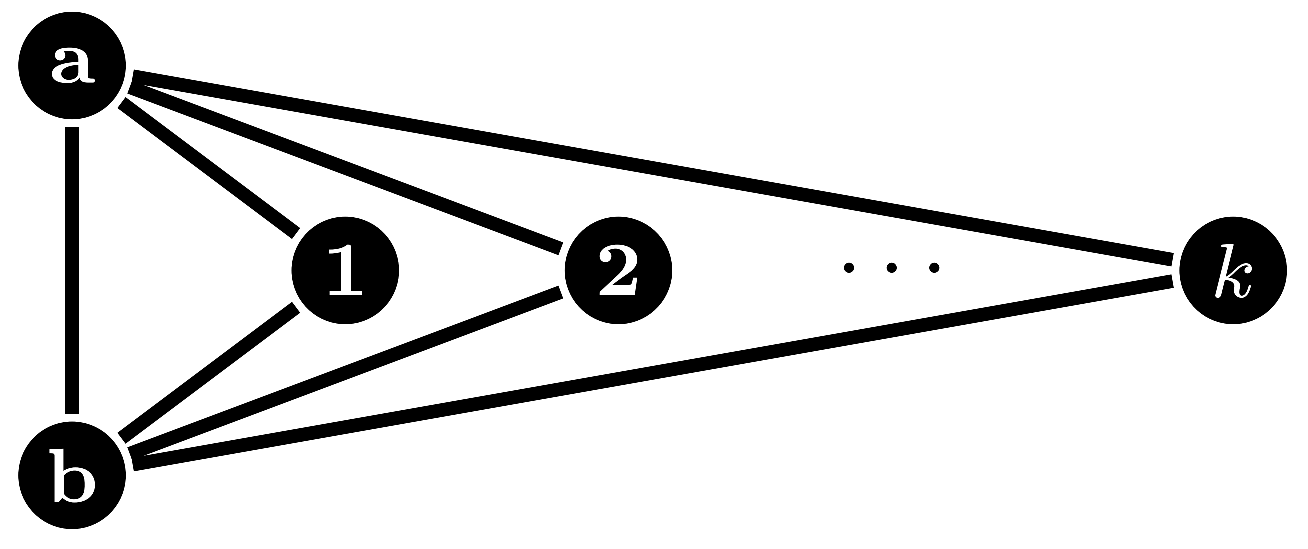


(1) specify model parameters that govern graph evolution

social circuit dependence

‘New specifications of ERGMs’

(geometrically weighted) edgewise shared partners



gwesp

`gwesp` is

- a measure of higher order clustering
- a special effect that measures triadic closure in a network
- helps avoid degeneracy:
  - having one shared friend makes a tie 25% more likely,
  - having six shared friends makes a tie 150% more likely (**!**)
  - thus we need to discount each additional tie added
  - done with a parameter **decay:** controls discounting 2nd, 3rd, etc. shared partners
  - **decay** takes on values between 0 and 1 (usual default is 0.693)
  - **decay** $\rightarrow 0$: the stronger is the discounting applied to subsequent shared partners

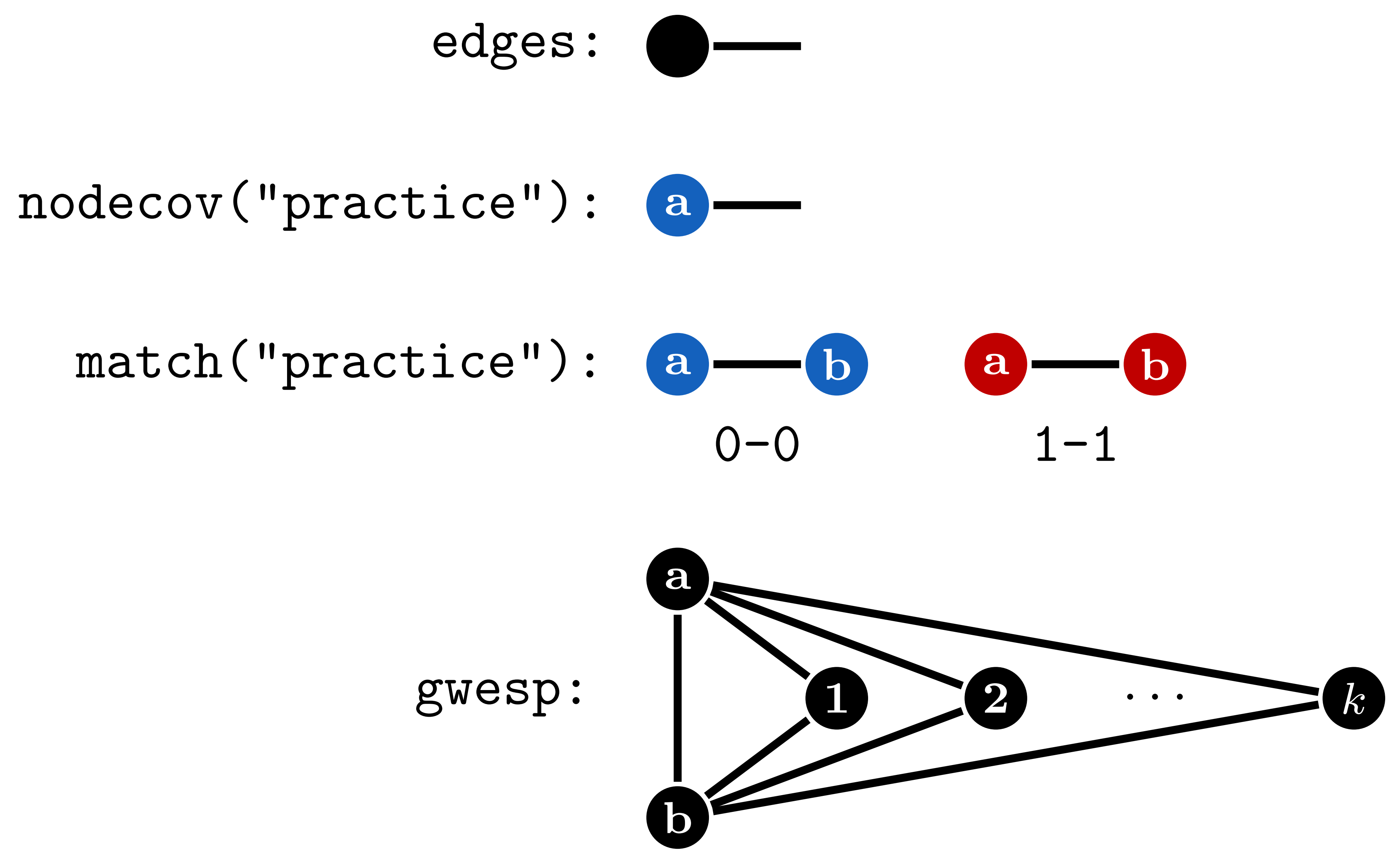
(1) specify model parameters that govern graph evolution

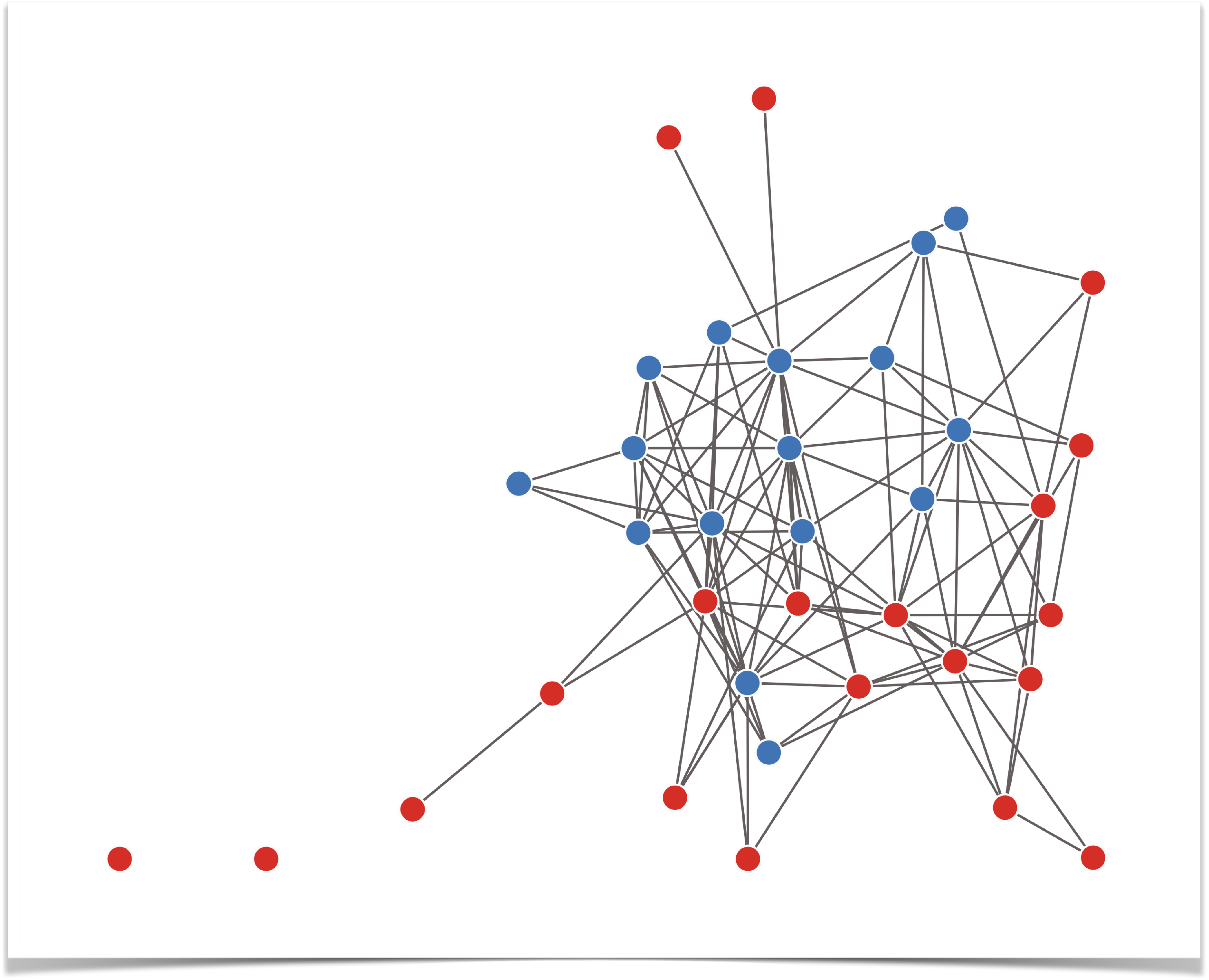
social circuit dependence

example. Lazega's lawyers (36 partners)

lawyers coloured by practice (undirected co-work network)

statistics included in an ERGM:



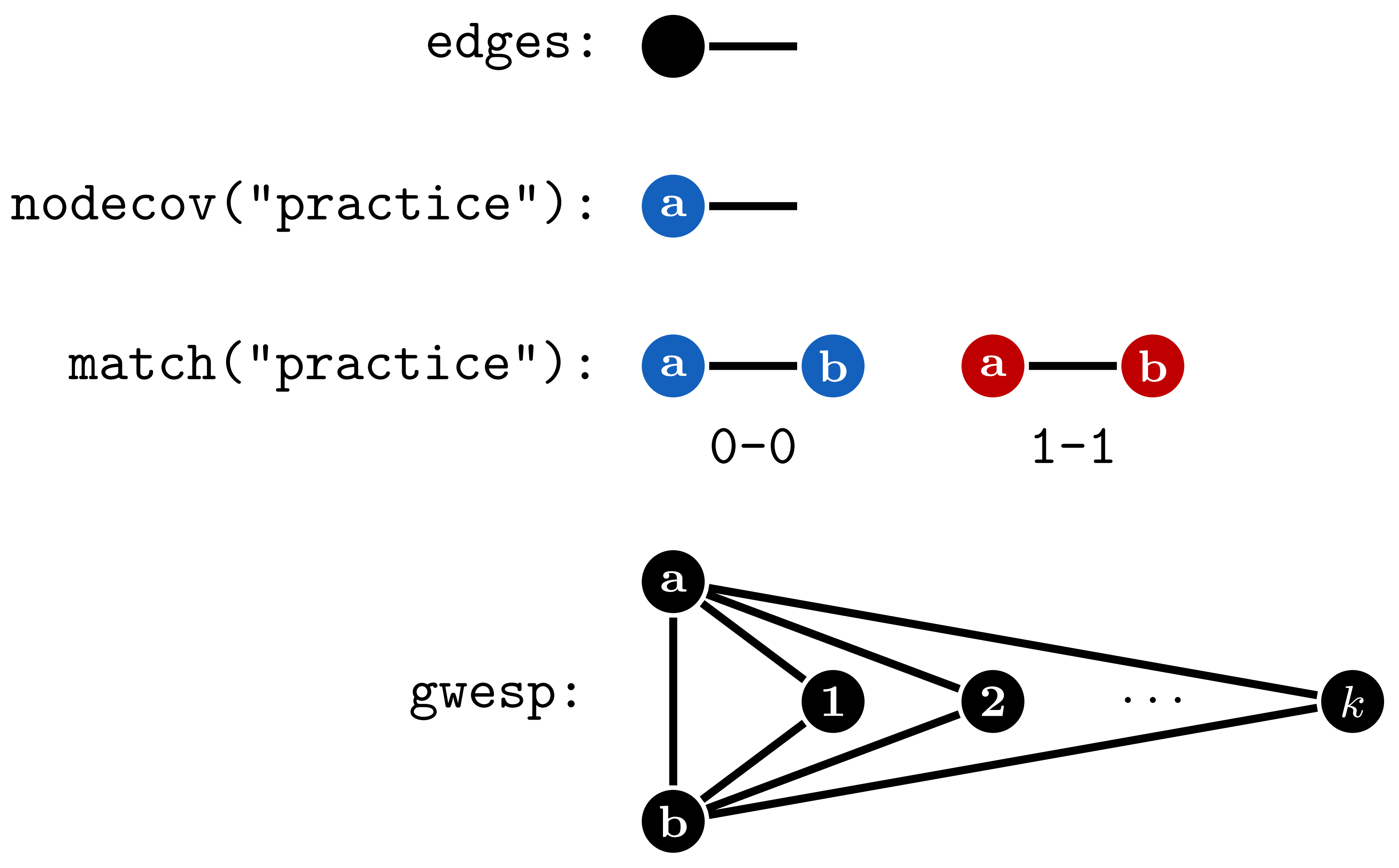


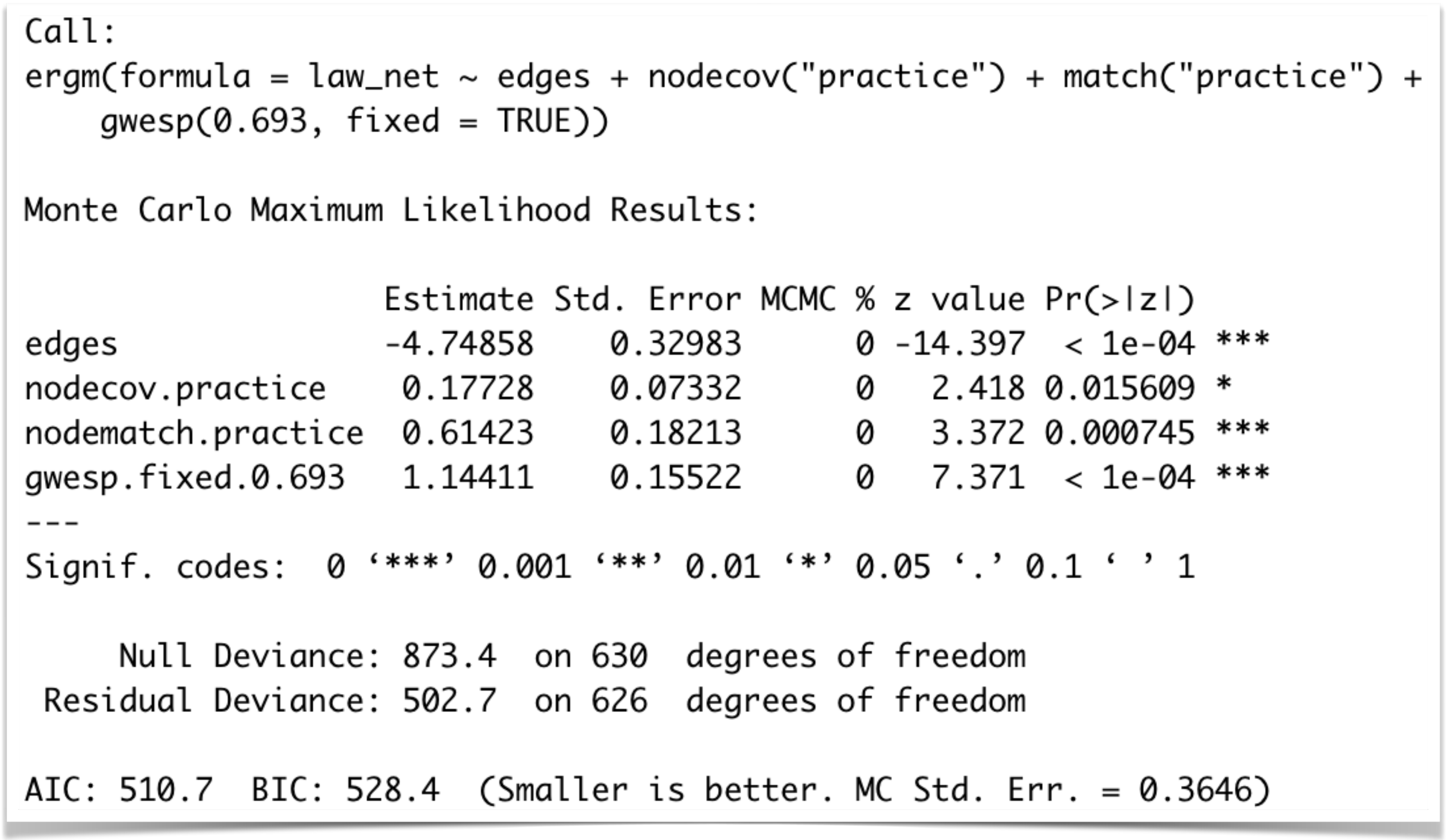
(1) specify model parameters that govern graph evolution

social circuit dependence

example. Lazega's lawyers (36 partners)

statistics included in an ERGM:





```
Call:
ergm(formula = law_net ~ edges + nodecov("practice") + match("practice") +
    gwesp(0.693, fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
edges              -4.74858    0.32983      0 -14.397  < 1e-04 ***
nodecov.practice    0.17728    0.07332      0   2.418 0.015609 *
nodematch.practice  0.61423    0.18213      0   3.372 0.000745 ***
gwesp.fixed.0.693   1.14411    0.15522      0   7.371  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 873.4  on 630  degrees of freedom
 Residual Deviance: 502.7  on 626  degrees of freedom

AIC: 510.7  BIC: 528.4  (Smaller is better. MC Std. Err. = 0.3646)
```

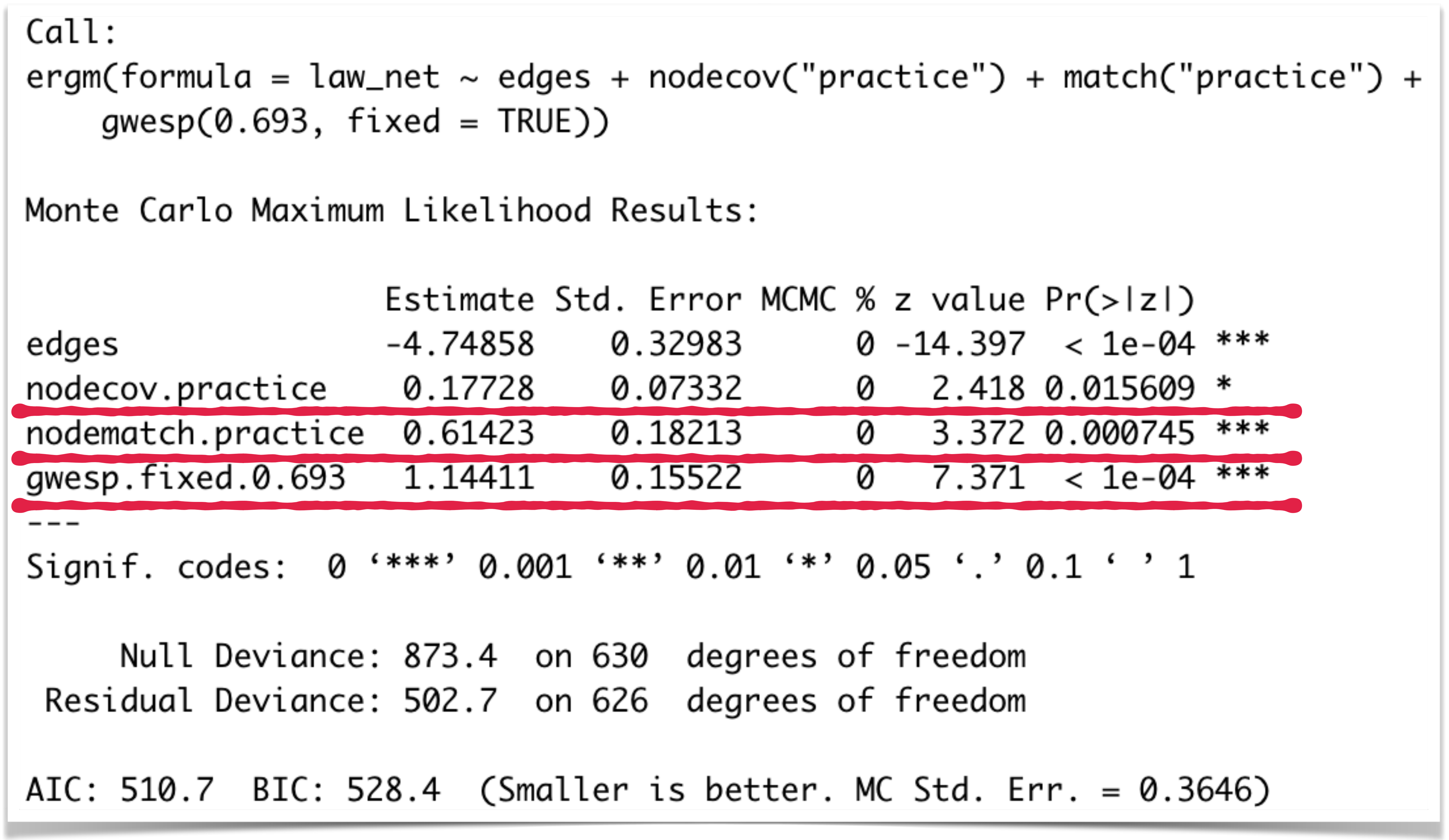
(1) specify model parameters that govern graph evolution

social circuit dependence

example. Lazega's lawyers (36 partners)

lawfirm partners that practice corporate law (= 1) collaborate more
with other partners also practicing corporate law

two actors that collaborate together tend to share many other collaborators



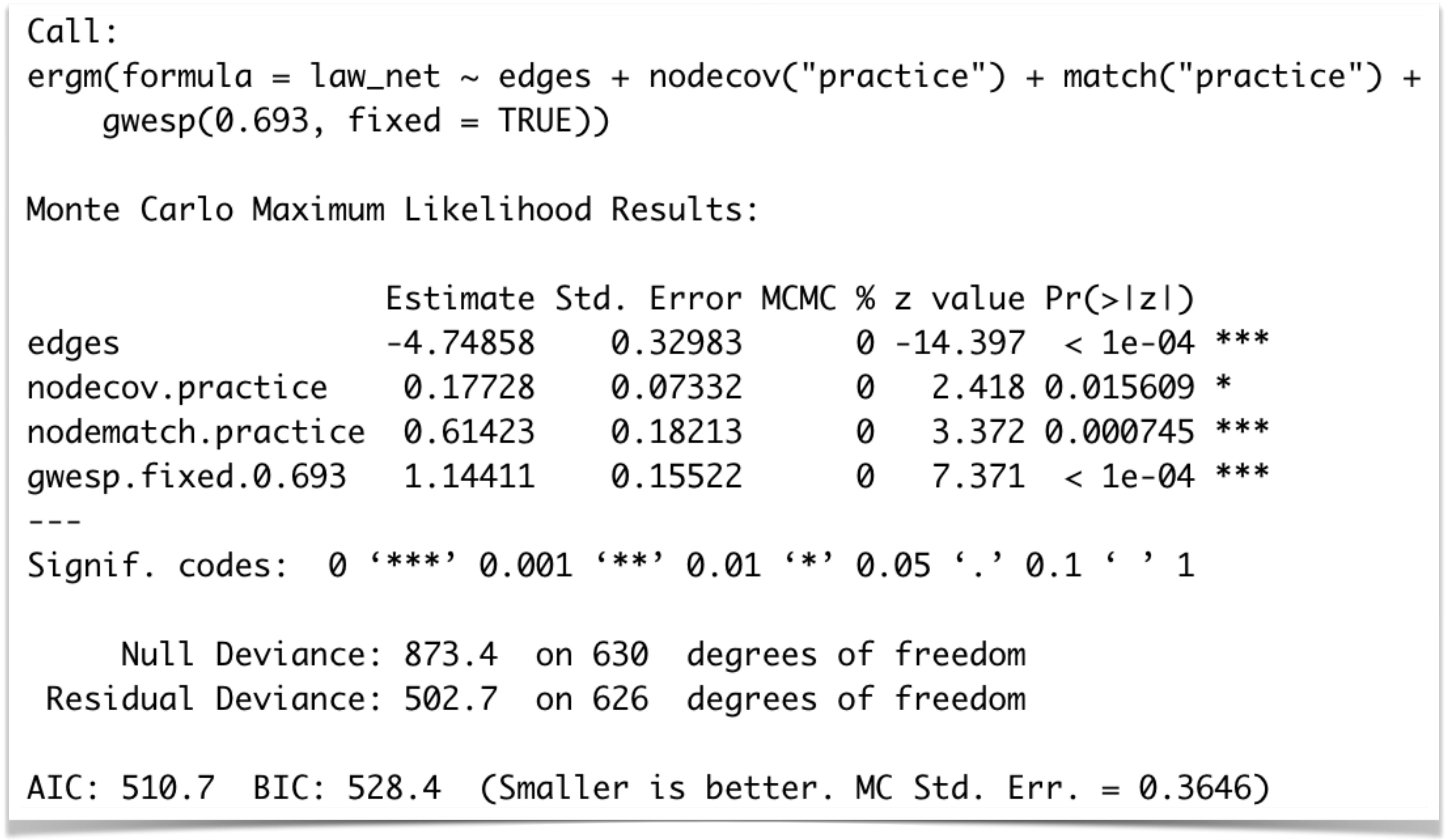
```
Call:
ergm(formula = law_net ~ edges + nodecov("practice") + match("practice") +
    gwesp(0.693, fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

                  Estimate Std. Error MCMC % z value Pr(>|z|)
edges             -4.74858    0.32983      0 -14.397  < 1e-04 ***
nodecov.practice   0.17728    0.07332      0   2.418 0.015609 *
nodematch.practice 0.61423    0.18213      0   3.372 0.000745 ***
gwesp.fixed.0.693  1.14411    0.15522      0   7.371  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 873.4  on 630  degrees of freedom
 Residual Deviance: 502.7  on 626  degrees of freedom

AIC: 510.7  BIC: 528.4  (Smaller is better. MC Std. Err. = 0.3646)
```

(1) specify model parameters that govern graph evolution
social circuit dependence

example. Lazega's lawyers (36 partners)

we find evidence for
- attribute-related activity
- homophily
- clustering/closure/balance

working simultaneously



```
Call:
ergm(formula = law_net ~ edges + nodecov("practice") + match("practice") +
    gwesp(0.693, fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
edges              -4.74858    0.32983      0 -14.397  < 1e-04 ***
nodecov.practice    0.17728    0.07332      0   2.418 0.015609 *
nodematch.practice  0.61423    0.18213      0   3.372 0.000745 ***
gwesp.fixed.0.693   1.14411    0.15522      0   7.371  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 873.4  on 630  degrees of freedom
 Residual Deviance: 502.7  on 626  degrees of freedom

AIC: 510.7  BIC: 528.4  (Smaller is better. MC Std. Err. = 0.3646)
```

exponential random graph models

ERGM modelling outline

(1) specify model parameters that govern graph evolution

- reciprocity, transitivity, homophily, etc.

(2) simulation and estimation

- maximum likelihood estimation
- simulate other random networks based on this model
- Markov Chain Monte Carlo (MCMC) algorithms:
  - generate sample of random networks following model rules
  - resulting networks should resemble the observed network

(3) compare goodness of fit of observed to modelled networks

- is the model a good fit for the data?

(2) simulation and estimation

maximum likelihood estimation

find the estimates of parameters that make our observed data/network most likely

$\implies$ maximize the probability of our observed network

but

normalising constant $$\kappa = \sum_{G'} \exp\left(\sum_{i=1}^{p} \theta_i \cdot s_i(G') \right)$$

needed to generate full outcomes space for all possible graphs (G')

causes much difficulty in estimation of parameters

makes direct calculation on the likelihood function very difficult/impossible

(works for up to approximately 10 nodes)

(2) simulation and estimation
Markov chain Monte Carlo

- direct estimation is analytically intractable
- we use numerical approximations based on simulations instead:

  Markov chain Monte Carlo

  - iterative method for simulating draws from a given distribution
  - we sample from the graph distribution by recording states from the chain (Gibbs samper, Metropolis-Hastings algorithm)
  - the R package `ergm`:
    MCMC and MCMC-MLE methods implemented for likelihood-based inference

    (1) pseudo maximum likelihood estimation (MPLE) is used to guess θ

    (2) networks are simulated using initial guess

    (3) the simulated sample is used to find θ using MLE

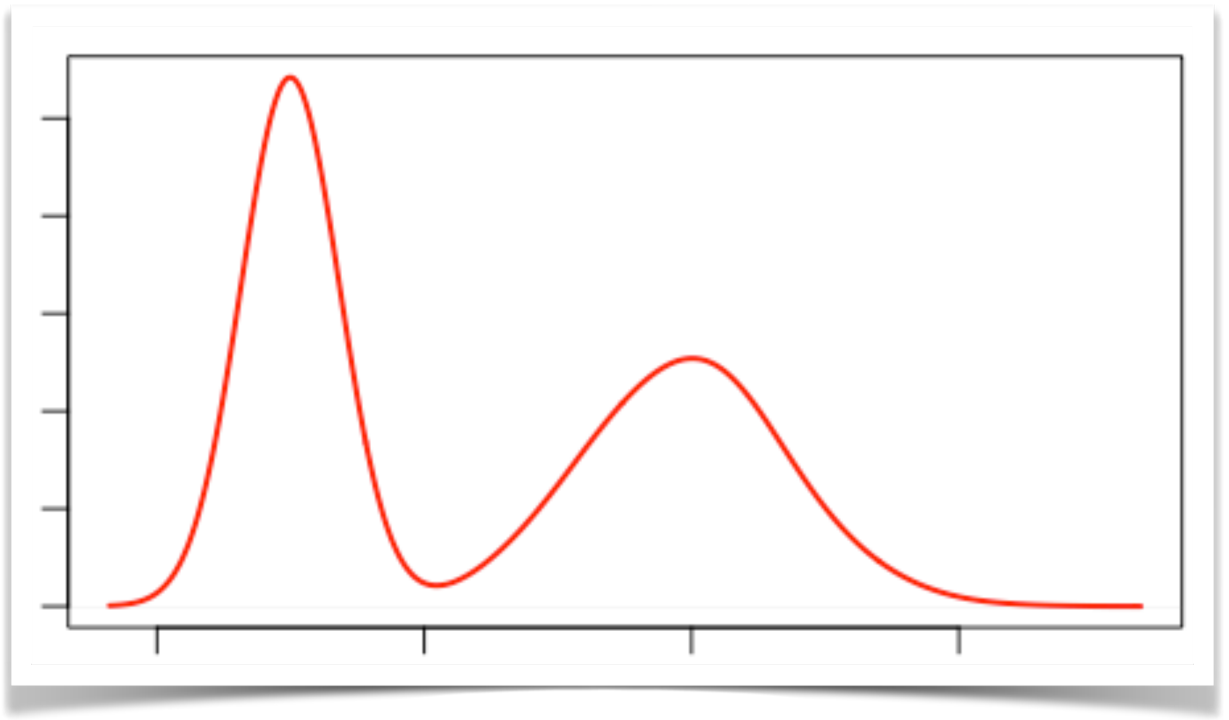
    (4) re-iterate from (2) since initial estimates are unstable and likely wrong

(2) simulation and estimation
model degeneracy

MCMC requires more "care and feeding" requiring user intervention

the MCMC simulation fails if we have

- insufficient burn-in: initial samples are still wrong
- insufficient post-burn samples: we do not get convergence to the stationary distribution



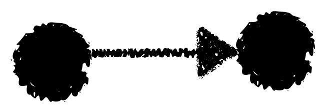
unstable ERGMs give rise to **multi-modal probability distributions:**

- probability mass centered on a small set of graphs:
  - only empty or complete graphs have a non-vanishing probability
  - this are **degenerate** graphs since real networks are not close to either
- rest of the graphs in the model space have negligible probability i.e. they are very unlikely

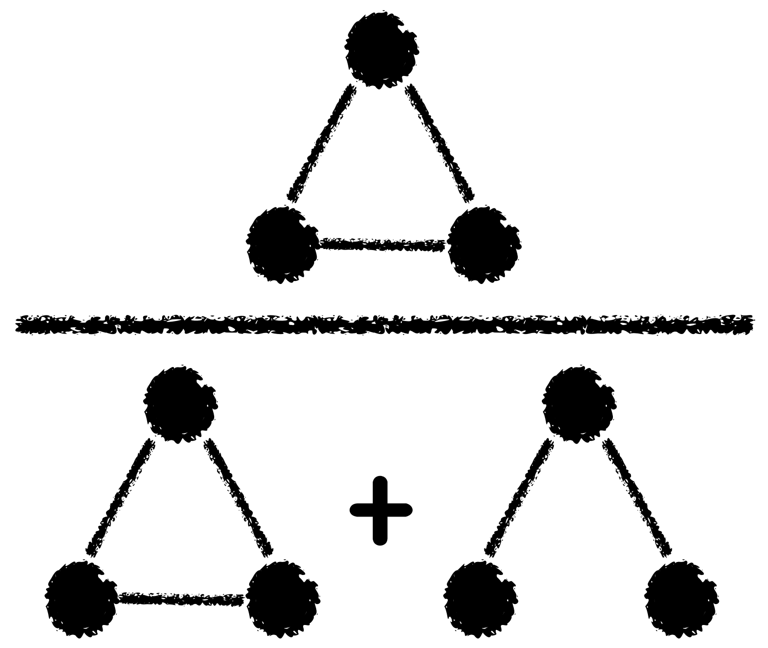
(2) simulation and estimation
model degeneracy

example.

consider an ERGM with two statistics:

edges (density) 

governs the overall number of ties

clustering 

captures propensity for or against transitivity

many combinations of possible values of the parameters produce

- networks that are full: every tie exists
- networks that are empty: no ties exist

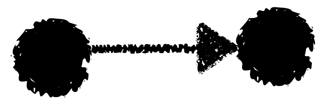
} **degenerate networks**

(2) simulation and estimation
model degeneracy

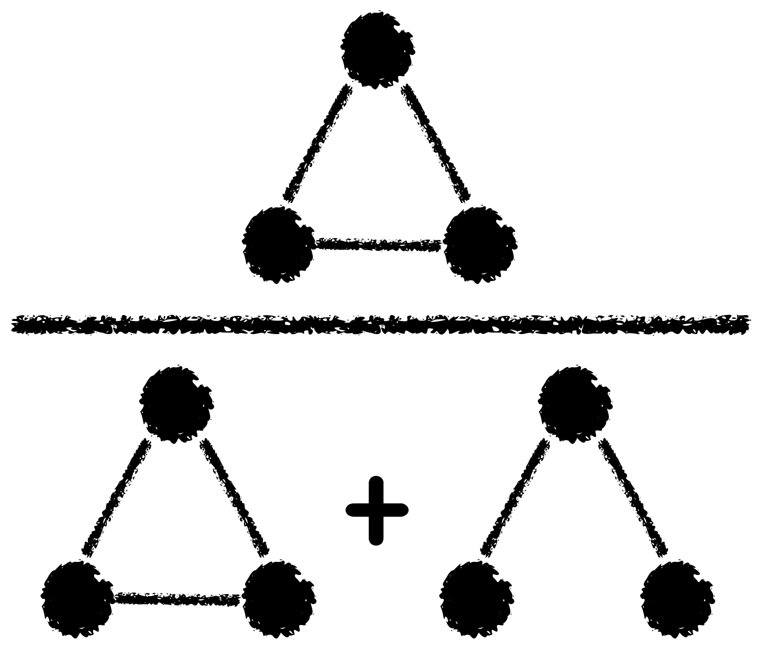
example.

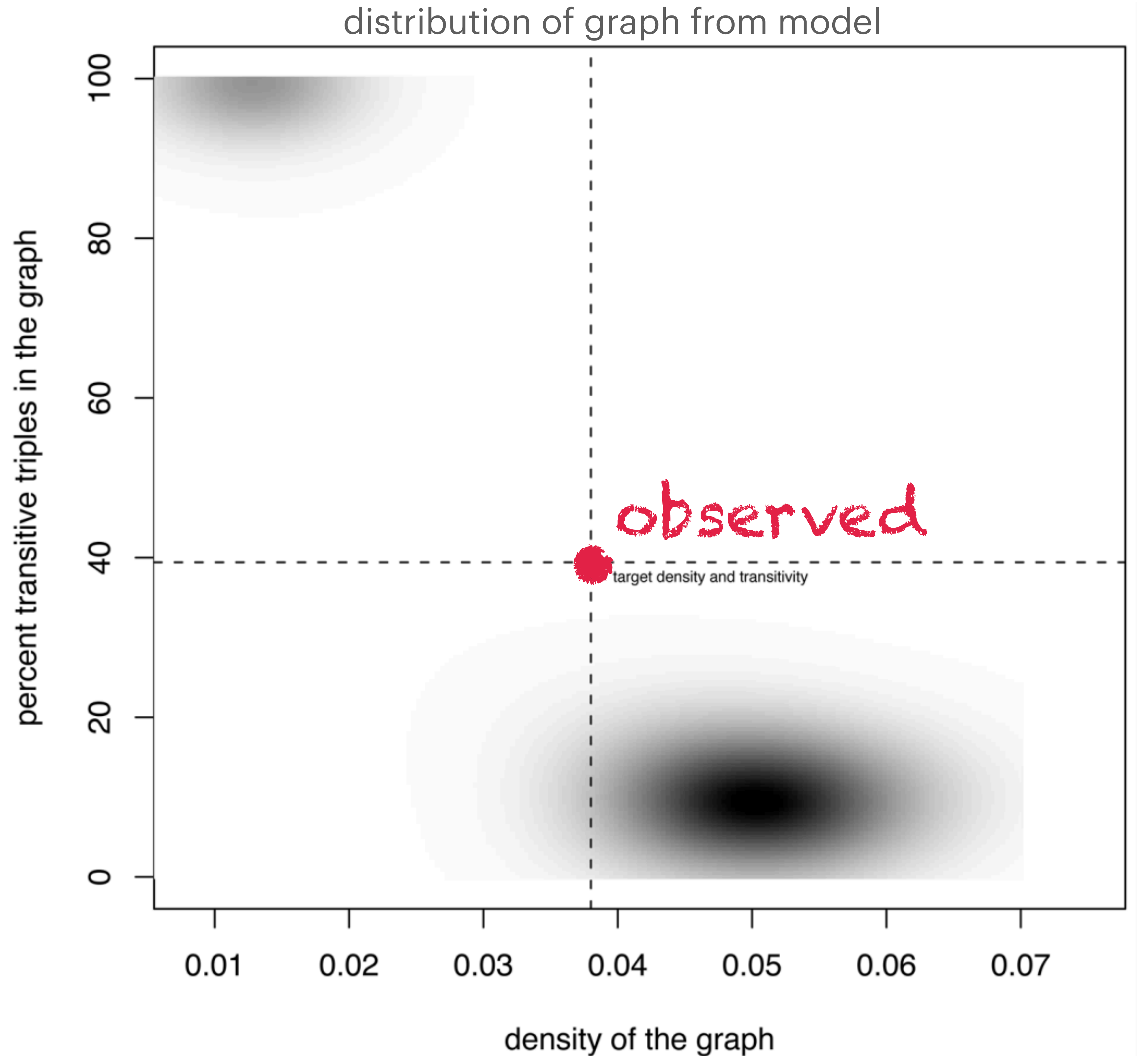
consider an ERGM with two statistics:

edges (density)



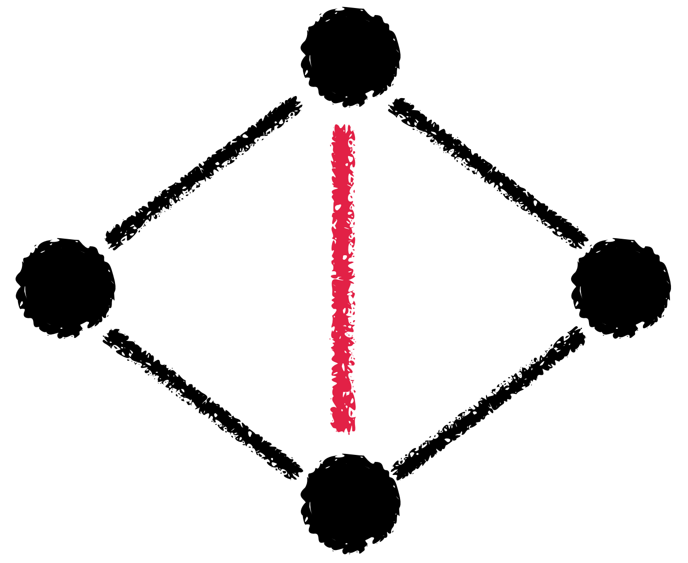
clustering




distribution of graph from model
percent transitive triples in the graph
density of the graph
0
20
40
60
80
100
0.01
0.02
0.03
0.04
0.05
0.06
0.07
observed
target density and transitivity

model almost never produces networks
at average density and average triad closure

one edge completes two triangles



..why it is better to use gwesp

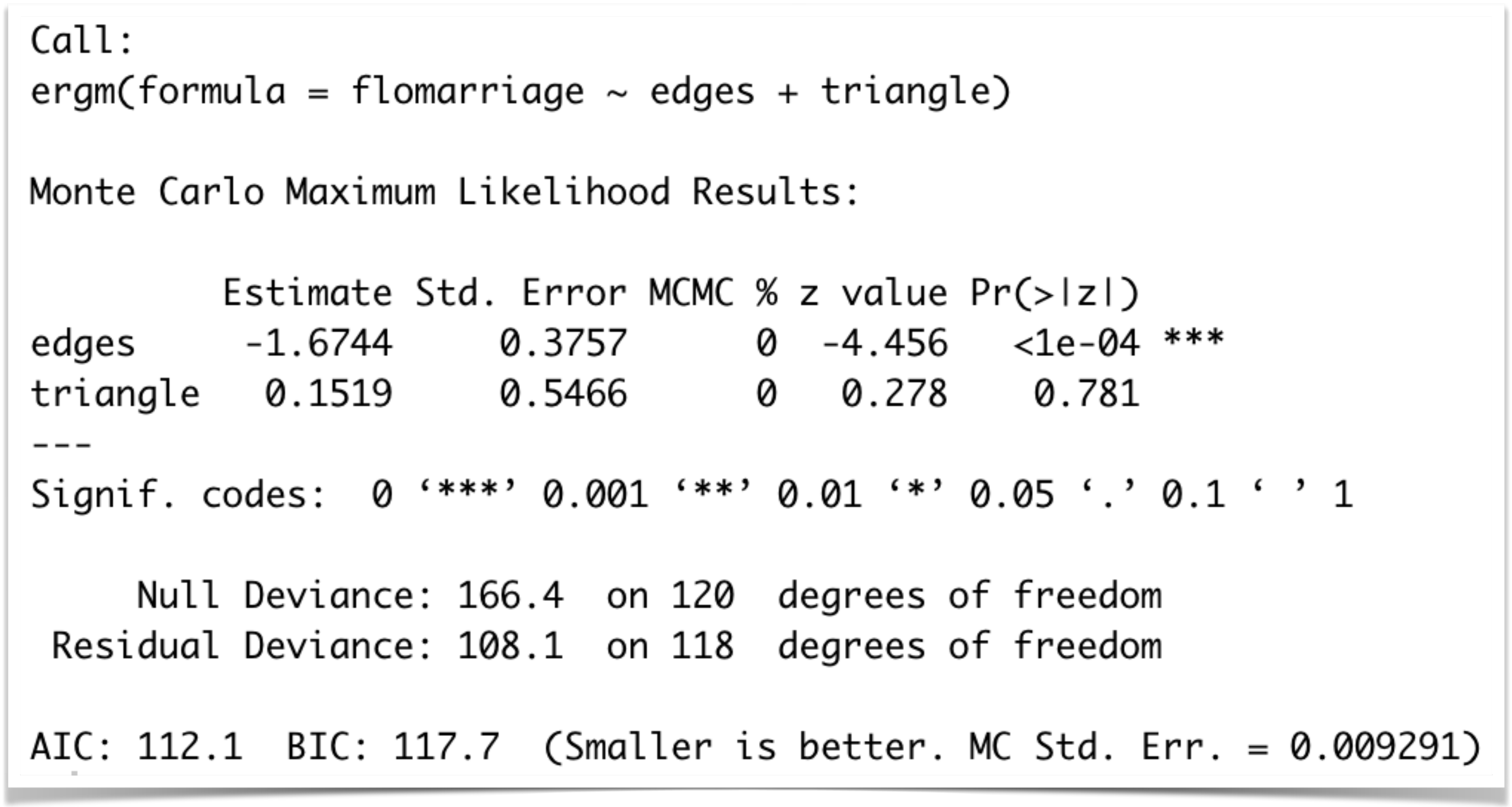
from Handcock et. al. (2008): *statnet: Software tools for the representation, visualization, analysis and simulation of network data*

(2) simulation and estimation
model degeneracy

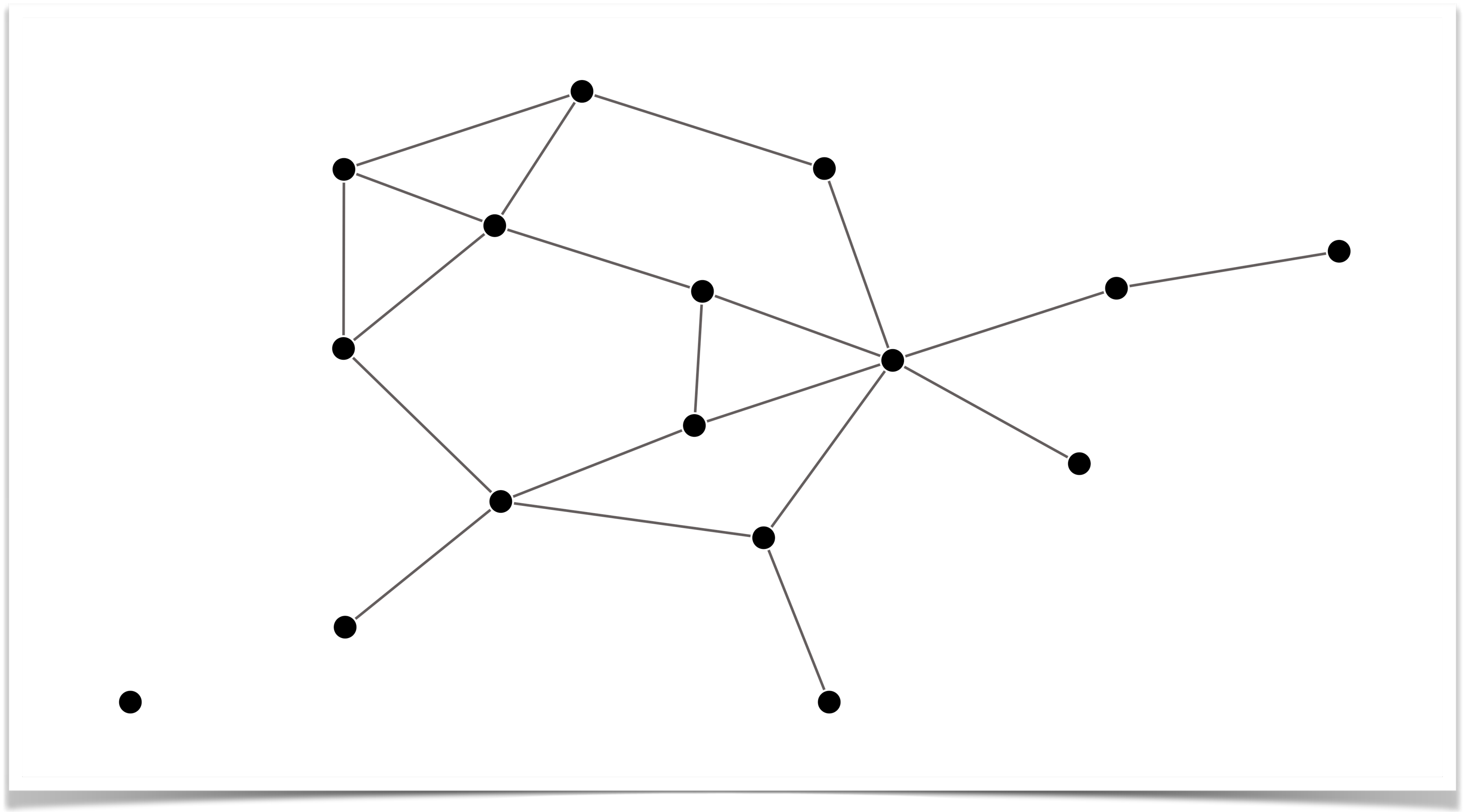
tools to assess MCMC simulation are available in the package `ergm`

example. Florentine marriage network

$$\boxed{P(G) = \frac{1}{\kappa} \exp\left(\theta_1 \cdot L + \theta_2 \cdot T\right)}$$



```
Call:
ergm(formula = flomarriage ~ edges + triangle)

Monte Carlo Maximum Likelihood Results:

         Estimate Std. Error MCMC % z value Pr(>|z|)
edges     -1.6744     0.3757      0  -4.456   <1e-04 ***
triangle   0.1519     0.5466      0   0.278    0.781
---
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 118  degrees of freedom

AIC: 112.1  BIC: 117.7  (Smaller is better. MC Std. Err. = 0.009291)
```



let’s assess the MCMC run

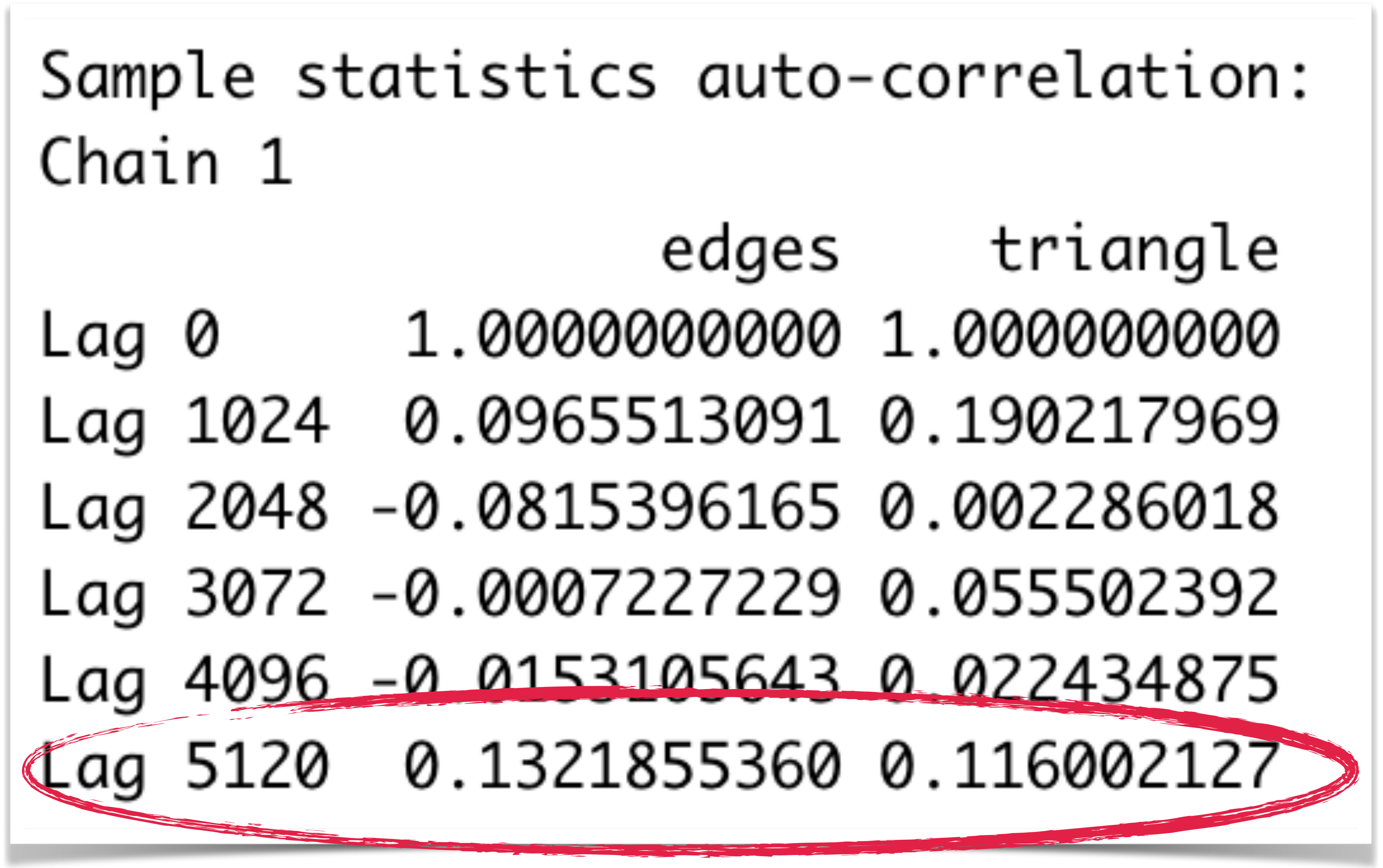
(2) simulation and estimation
model degeneracy

tools to assess MCMC simulation are available in the package `ergm`

example. Florentine marriage network

(1) sample statistic auto-correlation
the correlation between sample statistics at different points in the MCMC chain

- a good chain that is randomly mixing should have low auto-correlation values (close to zero) at all sampling points after Lag 0



```
Sample statistics auto-correlation:
Chain 1
                edges     triangle
Lag 0      1.0000000000 1.000000000
Lag 1024   0.0965513091 0.190217969
Lag 2048  -0.0815396165 0.002286018
Lag 3072  -0.0007227229 0.055502392
Lag 4096  -0.0153105643 0.022434875
Lag 5120   0.1321855360 0.116002127
```

a little too high $\longrightarrow$

(2) simulation and estimation

model degeneracy

tools to assess MCMC simulation are available in the package `ergm`

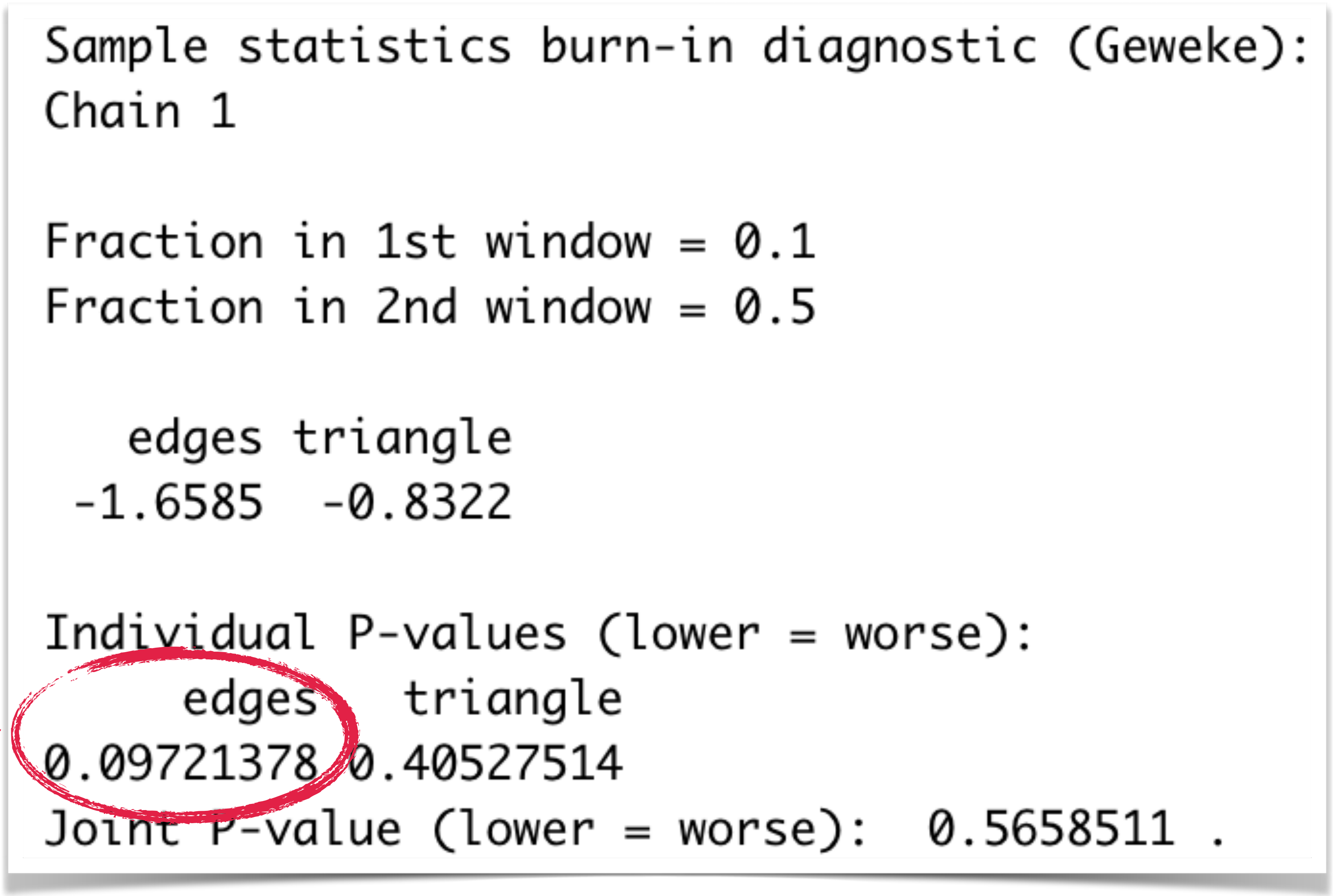
example. Florentine marriage network

(2) sample statistic burn-in diagnostic (Geweke)

a measure of convergence by comparing the means
of the sample statistic at different places in the Markov chain

- if the chains are stationary (randomly mixing), then means at different locations of the chains should be equal
- we are looking for *p*-values close to one (far from zero) for all the individual sample statistics

not looking good



```
Sample statistics burn-in diagnostic (Geweke):
Chain 1

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

   edges triangle
 -1.6585  -0.8322

Individual P-values (lower = worse):
     edges   triangle
0.09721378 0.40527514
Joint P-value (lower = worse):  0.5658511 .
```

(2) simulation and estimation

model degeneracy

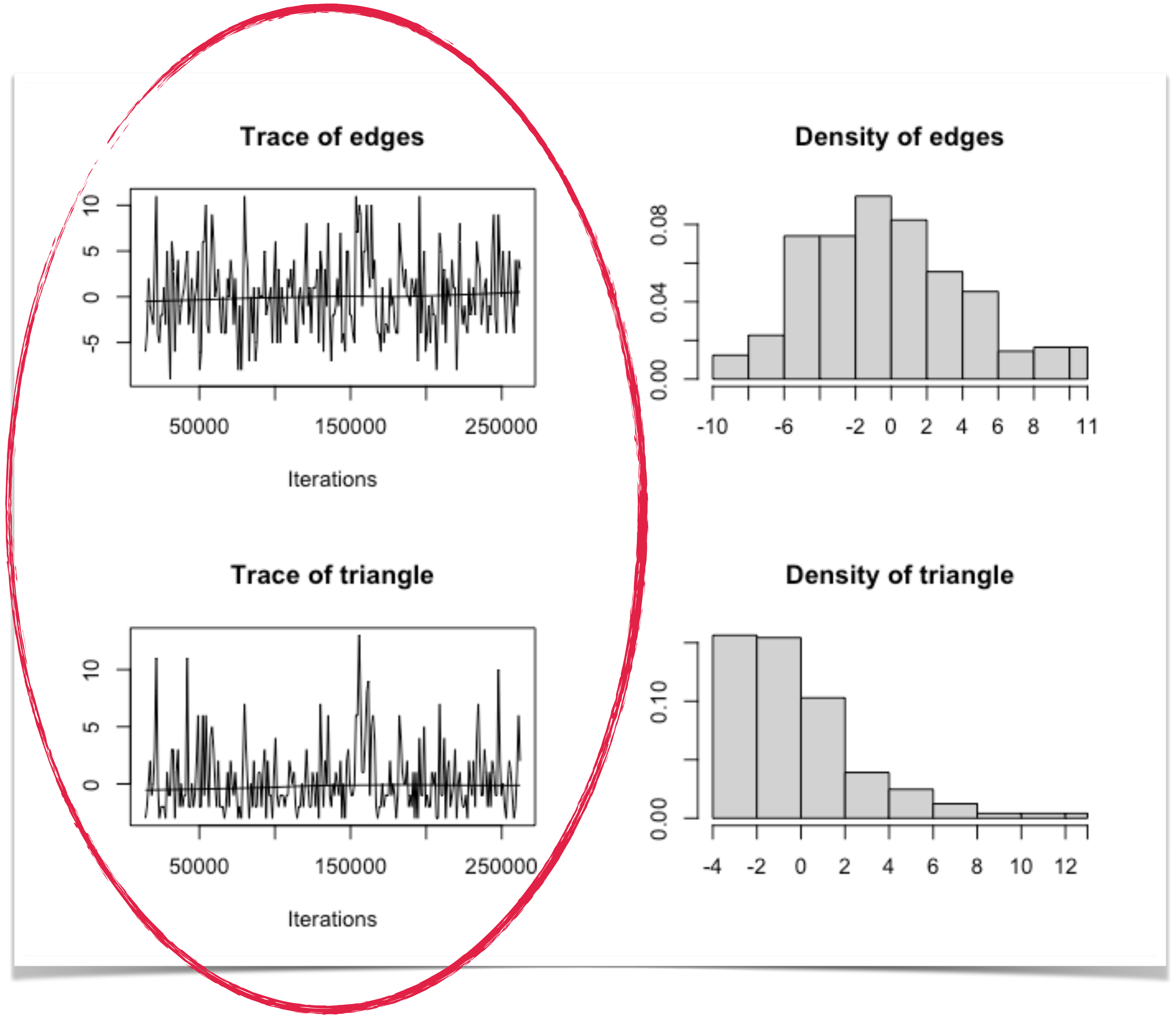
tools to assess MCMC simulation are available in the package `ergm`

example. Florentine marriage network

(3) MCMC trace plots

- plots of difference between sample statistics and your observed network for every step of the simulation
- should show evidence of "mixing" (random variation at each step) centered around zero

not looking good



(2) simulation and estimation

model degeneracy

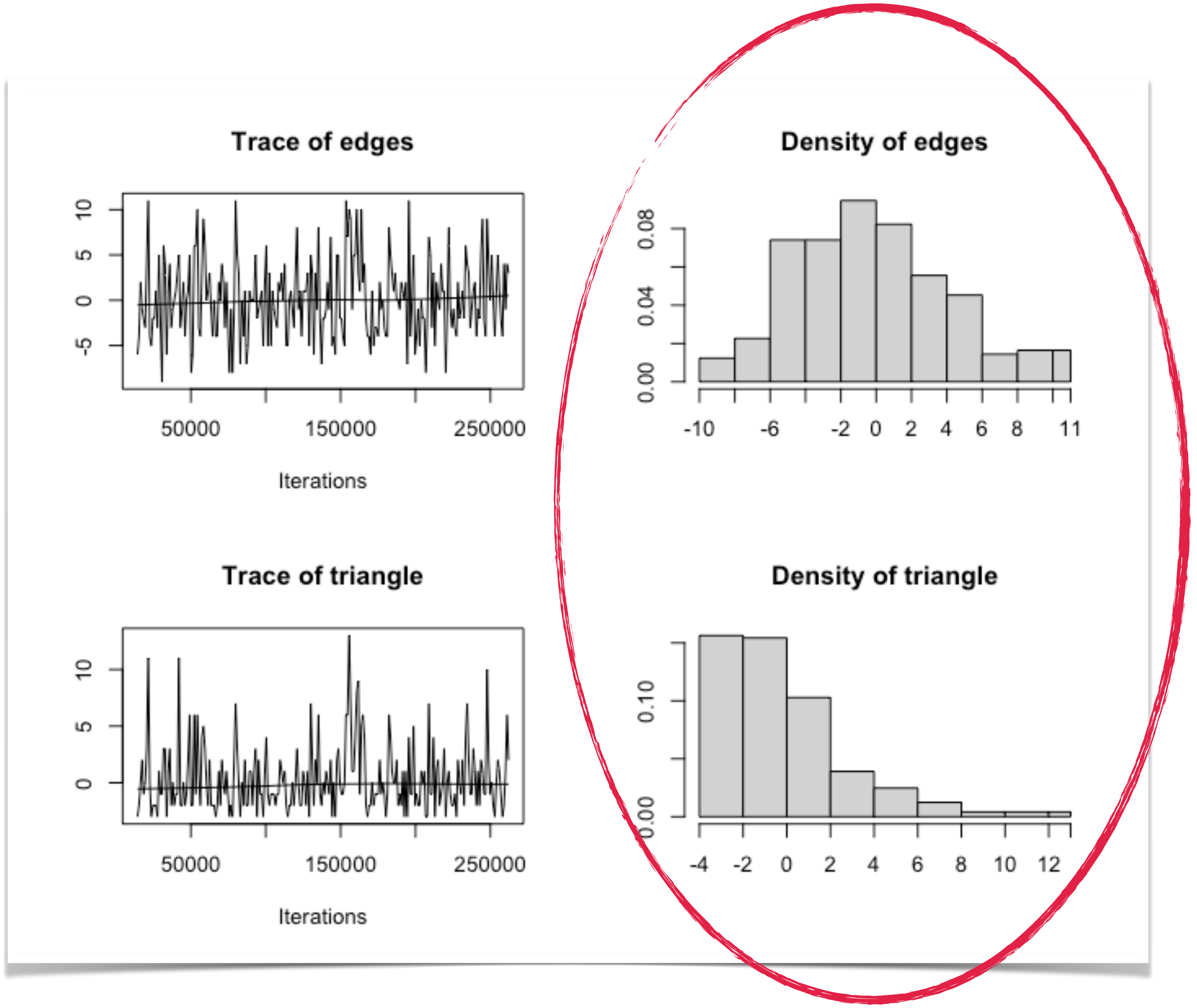
tools to assess MCMC simulation are available in the package `ergm`

example. Florentine marriage network

(3) MCMC density plots

- plots of difference between values of the sample statistics
- should have a bell-shaped distribution centered at zero (no difference to observed)

not looking good



(2) simulation and estimation

model degeneracy

tools to assess MCMC simulation are available in the package `ergm`

what to do when MCMC fails:

- adjusting the MCMC control parameters by increasing
  - the sample size
  - burn-in (helps with Geweke diagnostics)
  - interval (helps with sample auto-correlation problems)

```
control=control.ergm(MCMC.burnin=X, MCMC.interval=Y, MCMC.samplesize=Z))
```

degeneracy is not a property of ERGMs but an indicator of a poorly specified model

exponential random graph models

ERGM modelling outline

(1) specify model parameters that govern graph evolution

- reciprocity, transitivity, homophily, etc.

(2) simulation and estimation

- maximum likelihood estimation
- simulate other random networks based on this model
- Markov Chain Monte Carlo (MCMC) algorithms:
  - generate sample of random networks following model rules
  - resulting networks should resemble the observed network

(3) compare goodness of fit of observed to modelled networks

- is the model a good fit for the data?

(3) compare goodness of fit of observed to modelled networks

**simulate networks using fitted parameters of specified model
and calculate different structural measures on each simulated graph**

goodness of fit can be done in two ways directly in R using function `gof()`

(1) evaluate the fit to the specified terms in the model (default)

- to evaluate how well the estimates are reproducing the terms that are in the model
- MLE's reproduce observed sufficient statistics

(2) evaluate the fit of terms not specified in the model

three terms on three levels can be used to evaluate fit to emergent global network properties:

1. degree (node level)
2. edgewise share partners (edge level)
3. geodesic distances (dyad level)

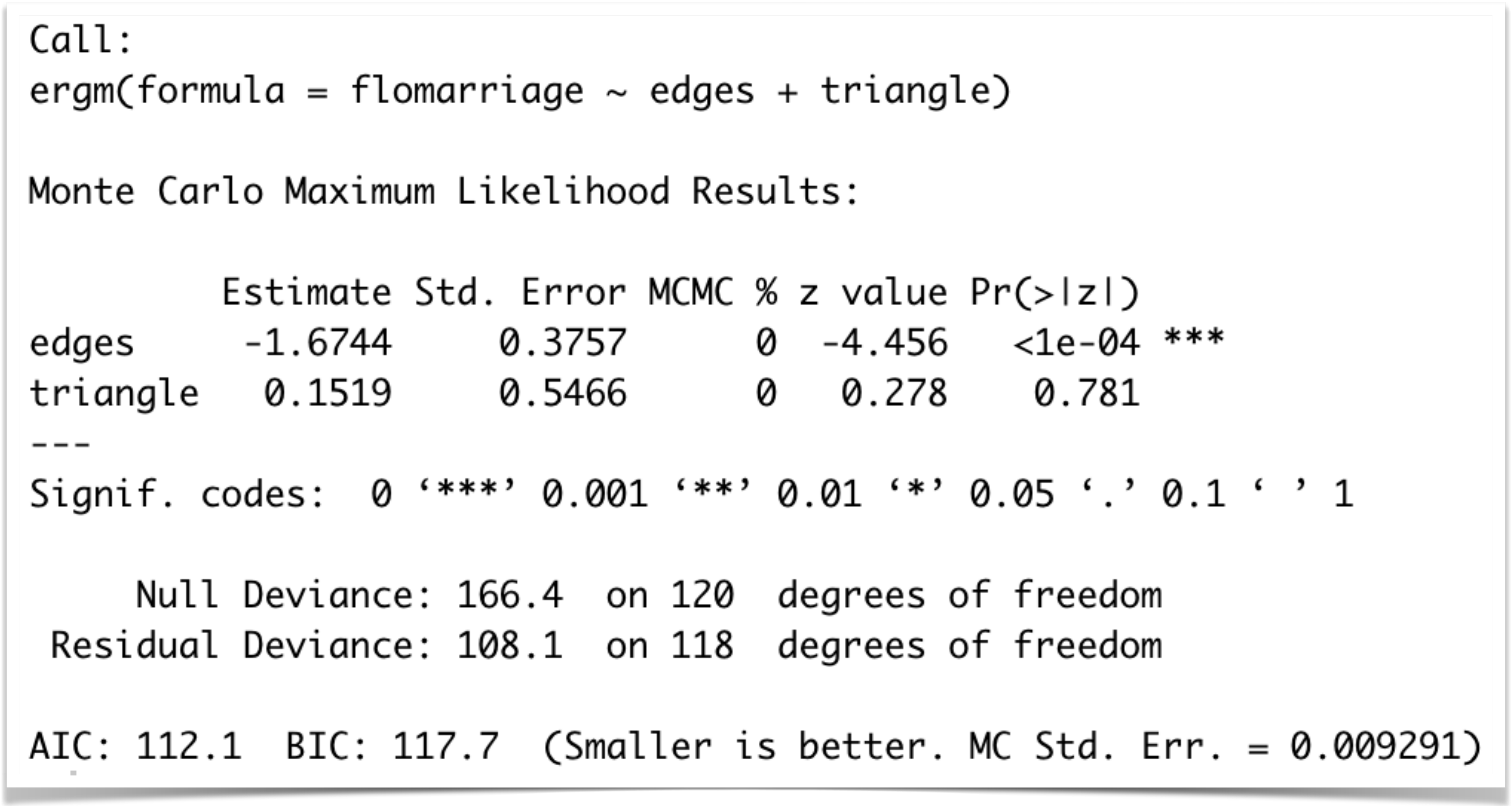
for undirected graphs $\Longrightarrow$ `~ degree + espartners + distance + model`

for directed graphs $\Longrightarrow$ `~ idegree + odegree + espartners + distance + model`

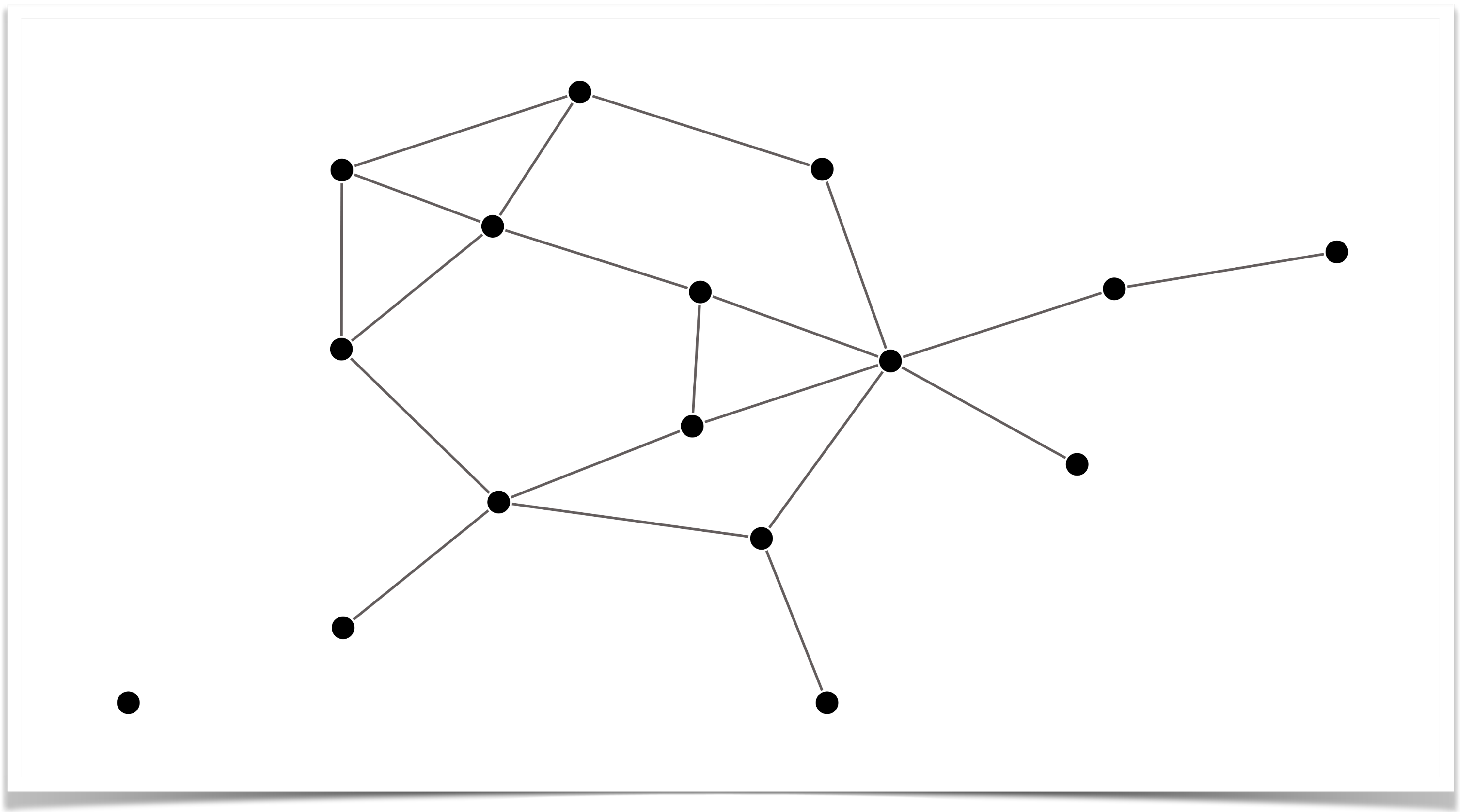
(3) compare goodness of fit of observed to modelled networks

example. Florentine marriage network

$$\boxed{P(G) = \frac{1}{\kappa}\exp\left(\theta_1 \cdot L + \theta_2 \cdot T\right)}$$



```
Call:
ergm(formula = flomarriage ~ edges + triangle)

Monte Carlo Maximum Likelihood Results:

         Estimate Std. Error MCMC % z value Pr(>|z|)
edges     -1.6744     0.3757      0  -4.456   <1e-04 ***
triangle   0.1519     0.5466      0   0.278    0.781
---
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 118  degrees of freedom

AIC: 112.1  BIC: 117.7  (Smaller is better. MC Std. Err. = 0.009291)
```

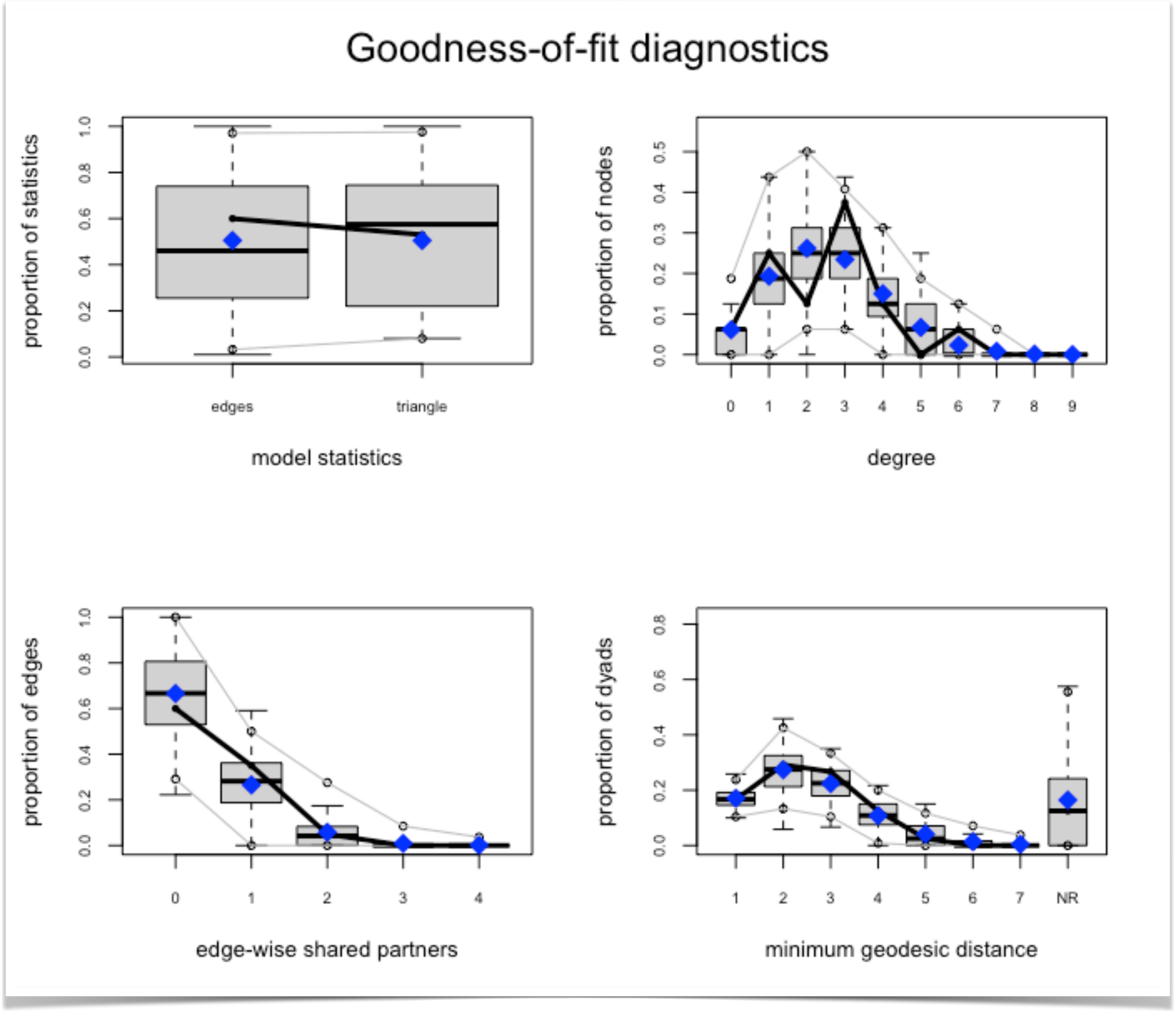


let’s assess the goodness of fit

(3) compare goodness of fit of observed to modelled networks

example. Florentine marriage network

- box plots represent the simulated counts and observed graph statistics are overlayed

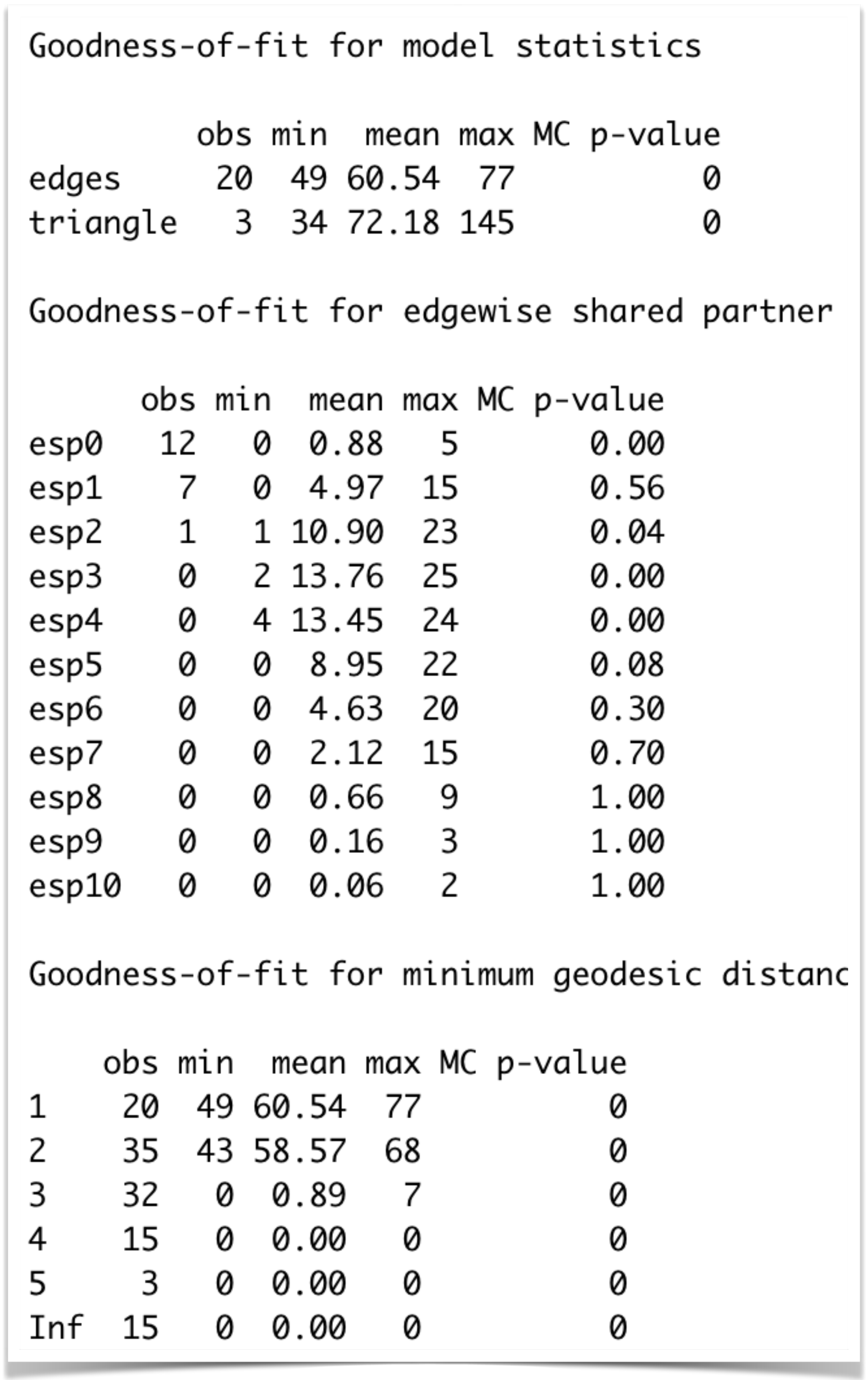


(3) compare goodness of fit of observed to modelled networks

example. Florentine marriage network

- box plots represent the simulated counts and observed graph statistics are overlayed
- you can also get output to interpret:
  ex: model + esp + dist
- low *p*-value suggests that there may be a problem with the fit for that graph statistic

major red flags here 🚩

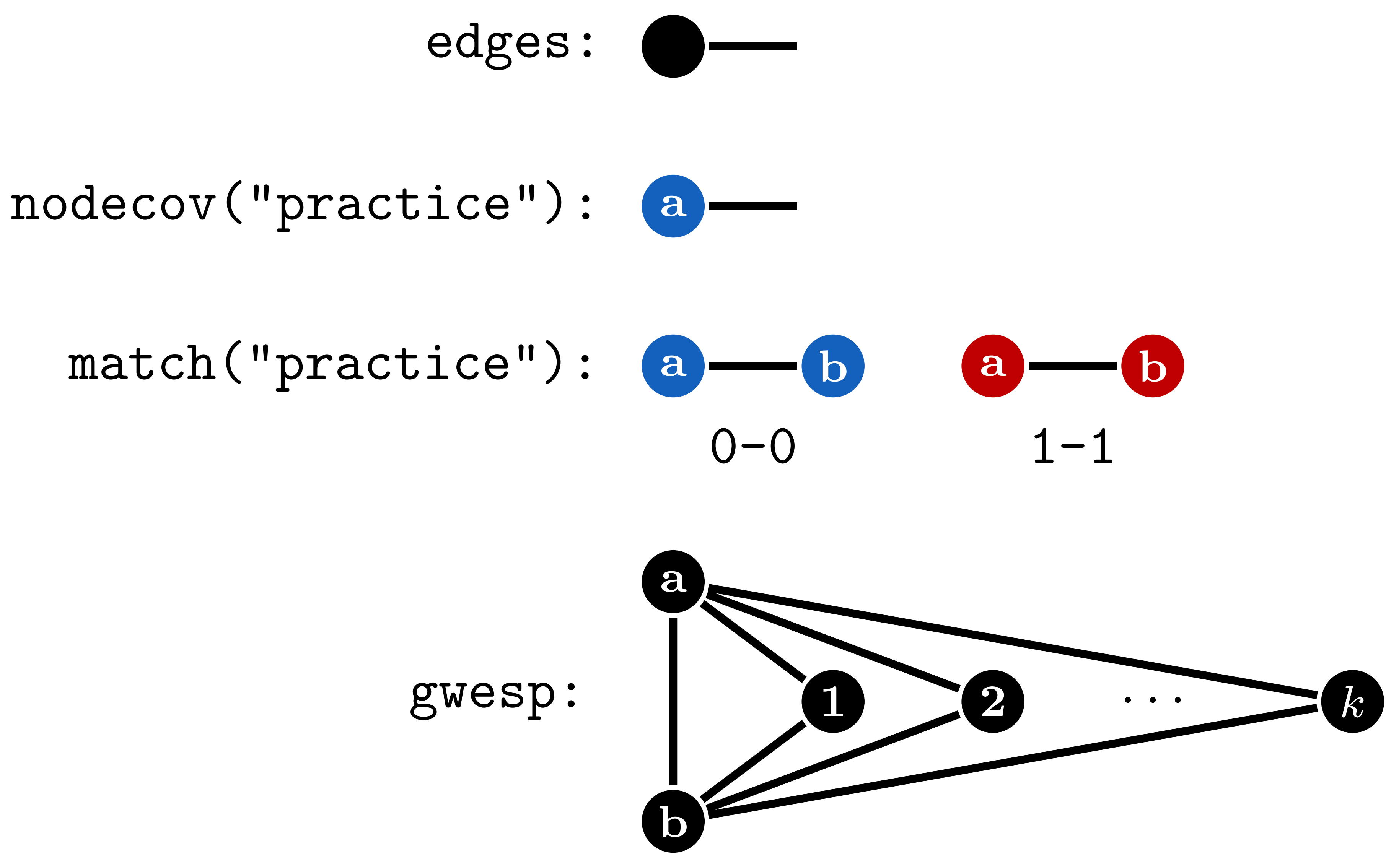


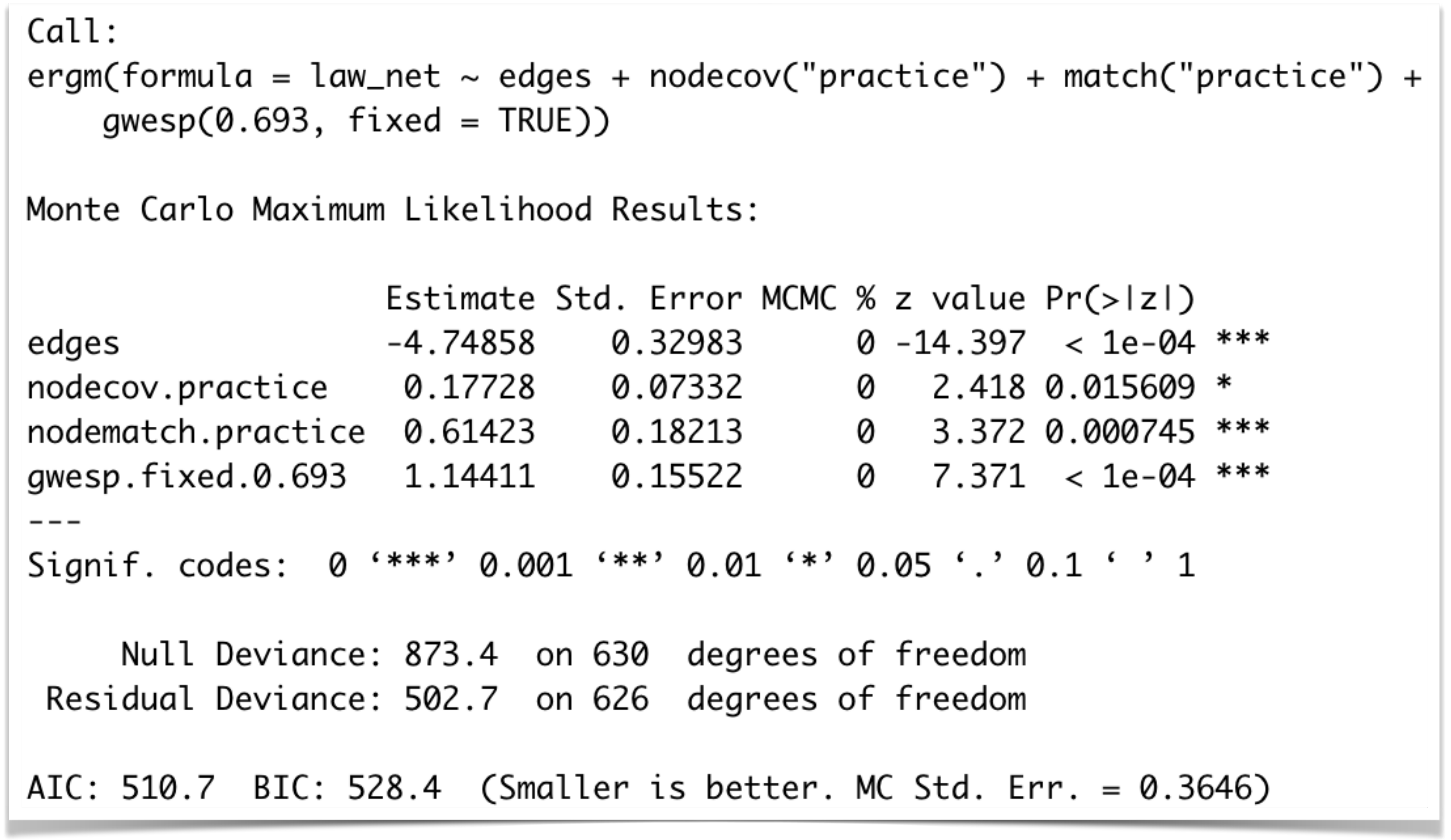
```
Goodness-of-fit for model statistics

          obs min  mean max MC p-value
edges      20  49 60.54  77          0
triangle    3  34 72.18 145          0

Goodness-of-fit for edgewise shared partner

      obs min  mean max MC p-value
esp0   12   0  0.88   5       0.00
esp1    7   0  4.97  15       0.56
esp2    1   1 10.90  23       0.04
esp3    0   2 13.76  25       0.00
esp4    0   4 13.45  24       0.00
esp5    0   0  8.95  22       0.08
esp6    0   0  4.63  20       0.30
esp7    0   0  2.12  15       0.70
esp8    0   0  0.66   9       1.00
esp9    0   0  0.16   3       1.00
esp10   0   0  0.06   2       1.00

Goodness-of-fit for minimum geodesic distanc

    obs min  mean max MC p-value
1    20  49 60.54  77          0
2    35  43 58.57  68          0
3    32   0  0.89   7          0
4    15   0  0.00   0          0
5     3   0  0.00   0          0
Inf  15   0  0.00   0          0
```

(3) compare goodness of fit of observed to modelled networks

example. Lazega's lawyers (36 partners)

statistics included in an ERGM:



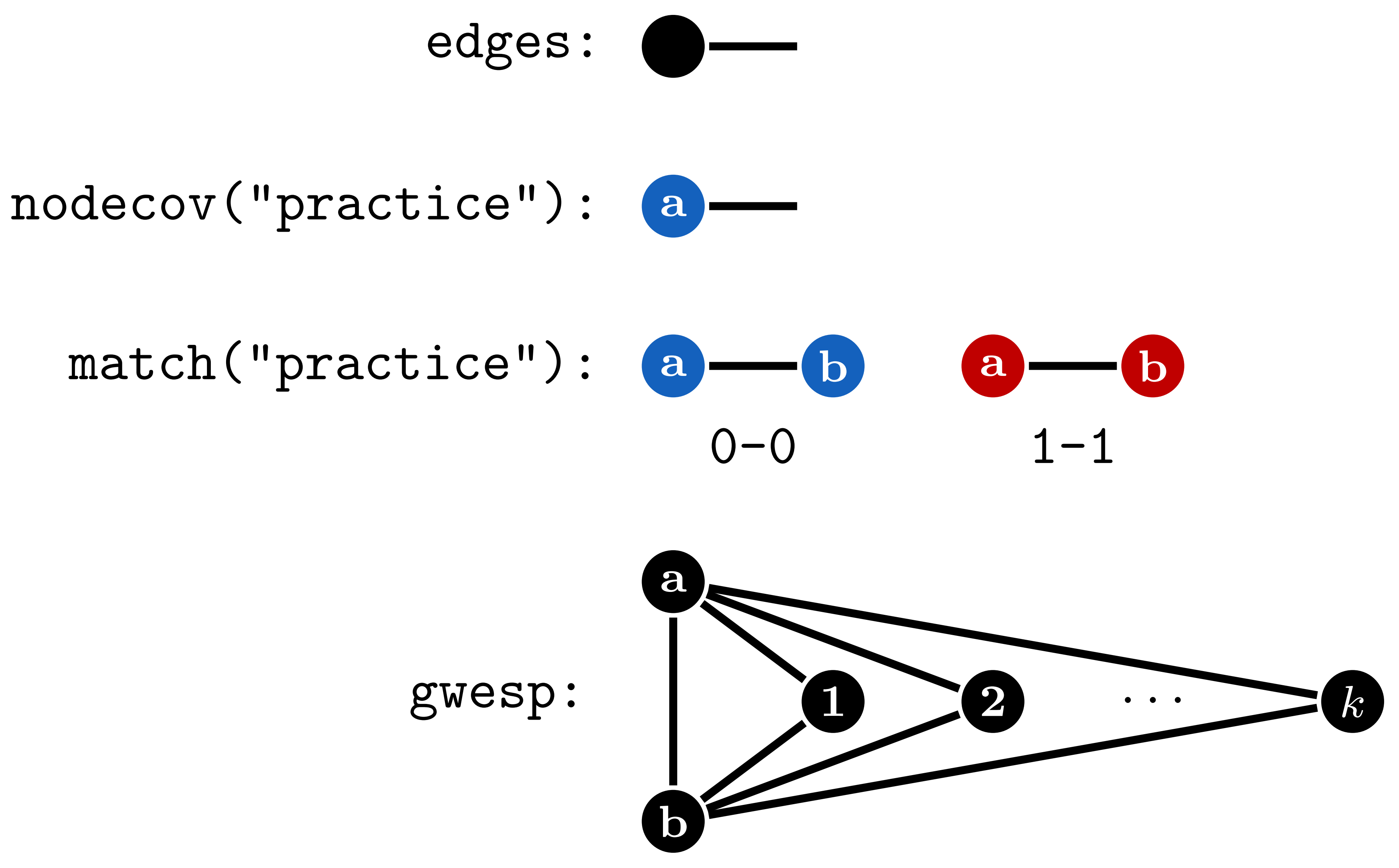


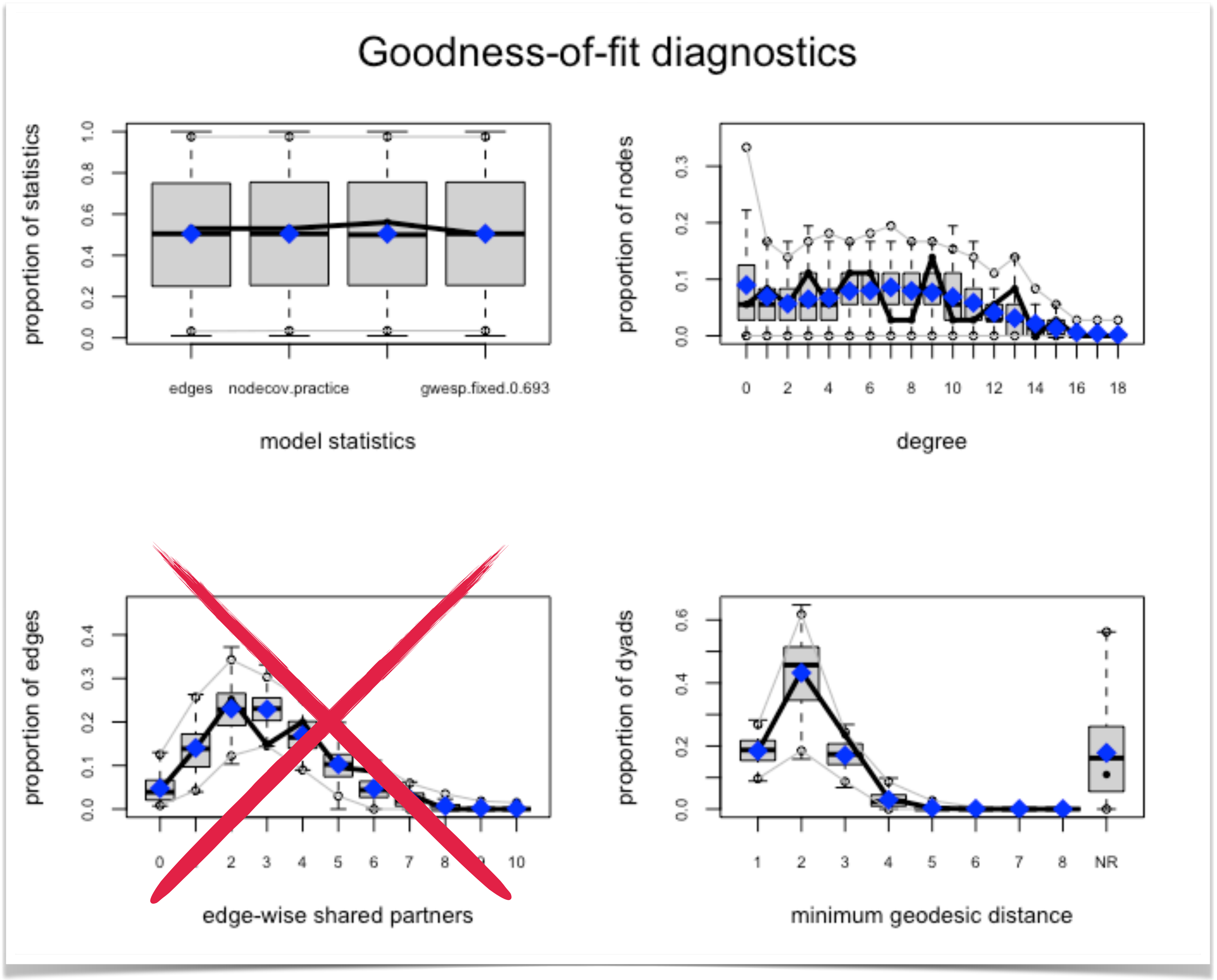
```
Call:
ergm(formula = law_net ~ edges + nodecov("practice") + match("practice") +
    gwesp(0.693, fixed = TRUE))

Monte Carlo Maximum Likelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
edges              -4.74858    0.32983      0 -14.397  < 1e-04 ***
nodecov.practice    0.17728    0.07332      0   2.418 0.015609 *
nodematch.practice  0.61423    0.18213      0   3.372 0.000745 ***
gwesp.fixed.0.693   1.14411    0.15522      0   7.371  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 873.4  on 630  degrees of freedom
 Residual Deviance: 502.7  on 626  degrees of freedom

AIC: 510.7  BIC: 528.4  (Smaller is better. MC Std. Err. = 0.3646)
```

(3) compare goodness of fit of observed to modelled networks

example. Lazega's lawyers (36 partners)

statistics included in an ERGM:



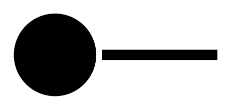


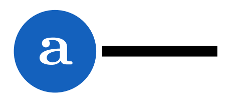
you should not use esp since it was explicitly modelled via the gwesp term

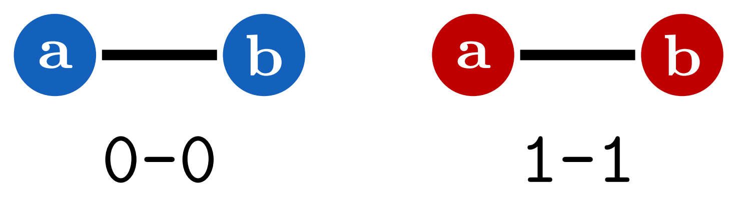
(3) compare goodness of fit of observed to modelled networks

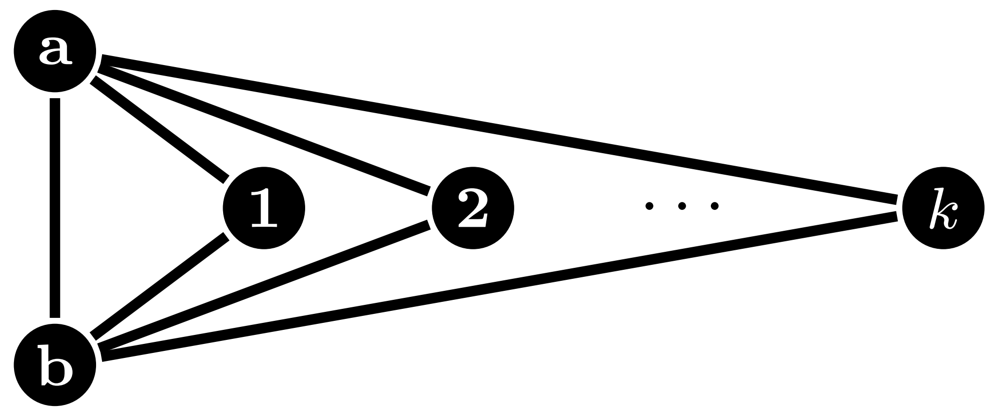
example. Lazega's lawyers (36 partners)

statistics included in an ERGM:

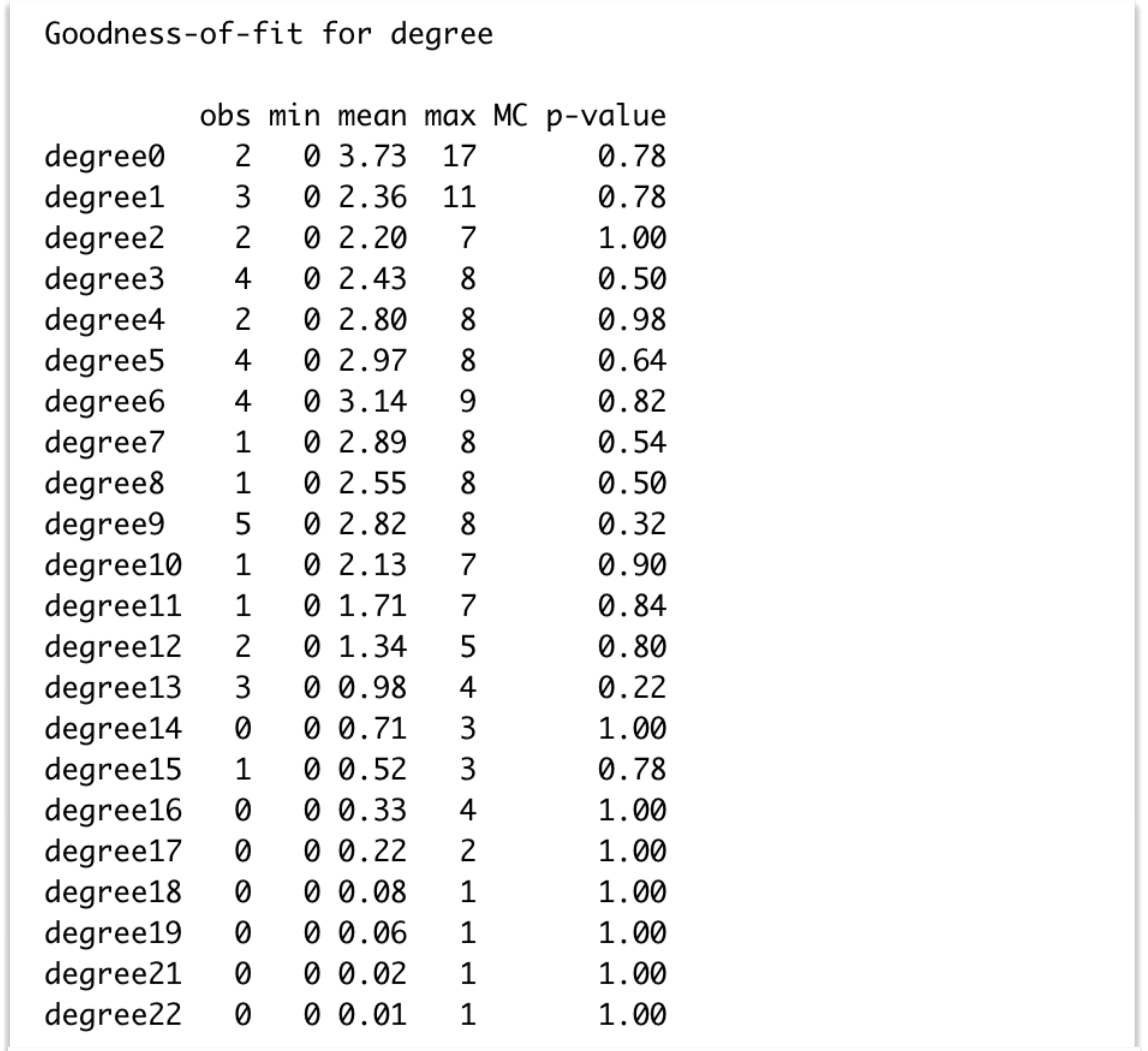
edges:


nodecov("practice"):


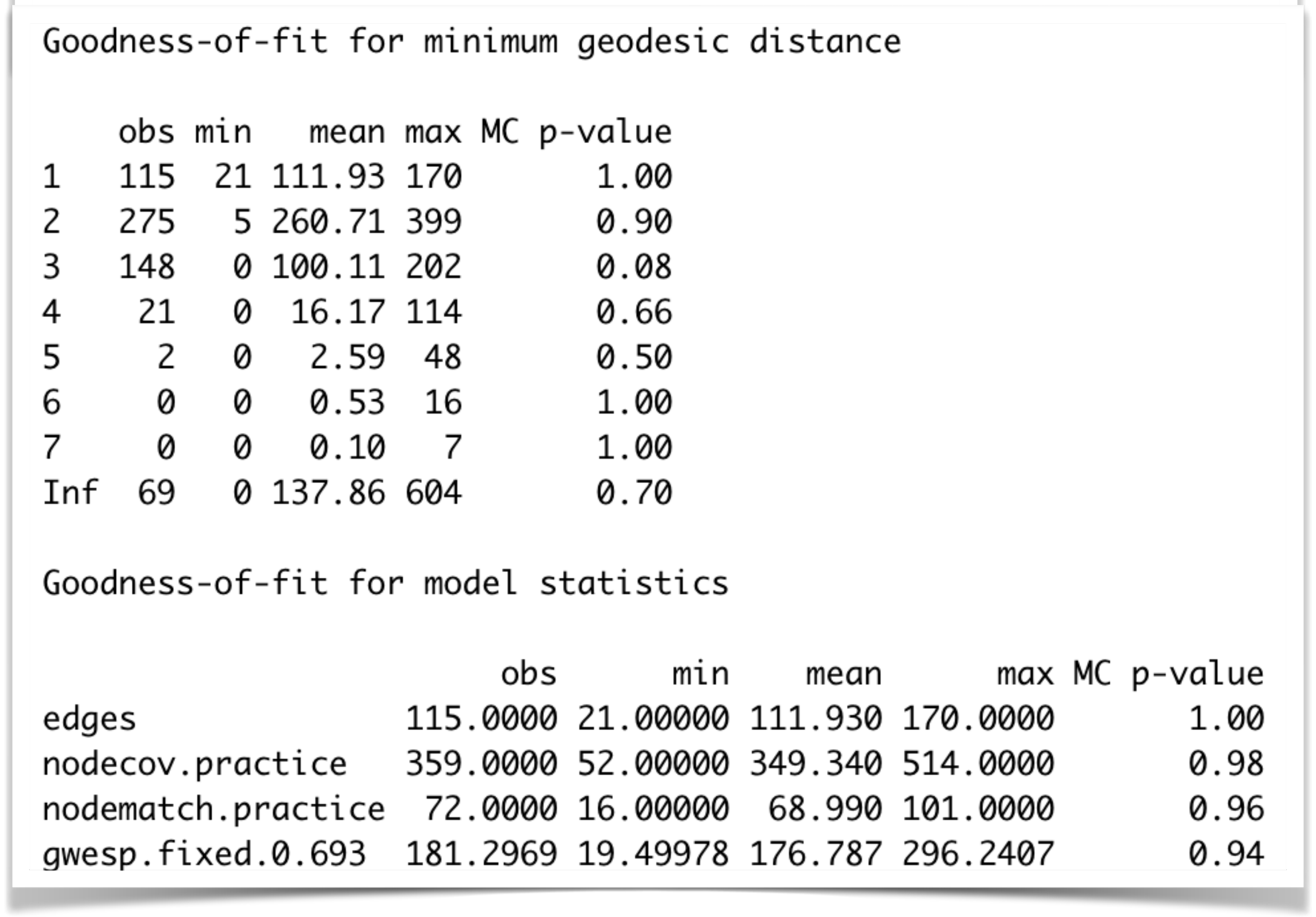
match("practice"):


gwesp:


this looks much better



```
Goodness-of-fit for degree

          obs min mean max MC p-value
degree0     2   0 3.73  17       0.78
degree1     3   0 2.36  11       0.78
degree2     2   0 2.20   7       1.00
degree3     4   0 2.43   8       0.50
degree4     2   0 2.80   8       0.98
degree5     4   0 2.97   8       0.64
degree6     4   0 3.14   9       0.82
degree7     1   0 2.89   8       0.54
degree8     1   0 2.55   8       0.50
degree9     5   0 2.82   8       0.32
degree10    1   0 2.13   7       0.90
degree11    1   0 1.71   7       0.84
degree12    2   0 1.34   5       0.80
degree13    3   0 0.98   4       0.22
degree14    0   0 0.71   3       1.00
degree15    1   0 0.52   3       0.78
degree16    0   0 0.33   4       1.00
degree17    0   0 0.22   2       1.00
degree18    0   0 0.08   1       1.00
degree19    0   0 0.06   1       1.00
degree21    0   0 0.02   1       1.00
degree22    0   0 0.01   1       1.00
```



```
Goodness-of-fit for minimum geodesic distance

    obs min   mean max MC p-value
1   115  21 111.93 170       1.00
2   275   5 260.71 399       0.90
3   148   0 100.11 202       0.08
4    21   0  16.17 114       0.66
5     2   0   2.59  48       0.50
6     0   0   0.53  16       1.00
7     0   0   0.10   7       1.00
Inf  69   0 137.86 604       0.70

Goodness-of-fit for model statistics

                        obs      min    mean      max MC p-value
edges              115.0000 21.00000 111.930 170.0000       1.00
nodecov.practice   359.0000 52.00000 349.340 514.0000       0.98
nodematch.practice  72.0000 16.00000  68.990 101.0000       0.96
gwesp.fixed.0.693  181.2969 19.49978 176.787 296.2407       0.94
```

some practical considerations

ERGMs are powerful **but**

- ▸ much attention and “hand-holding” is needed
  - MCMC diagnostics and model goodness-of-fit requires scrutiny
  - multiple dependency terms in model lead to convergence issues
- ▸ generally constrained to consider only networks with similar density
- ▸ some data sets are simply not amenable to fitting ERGMs

modelling strategy and inclusion of statistics
should be based on social theory and well defined hypotheses