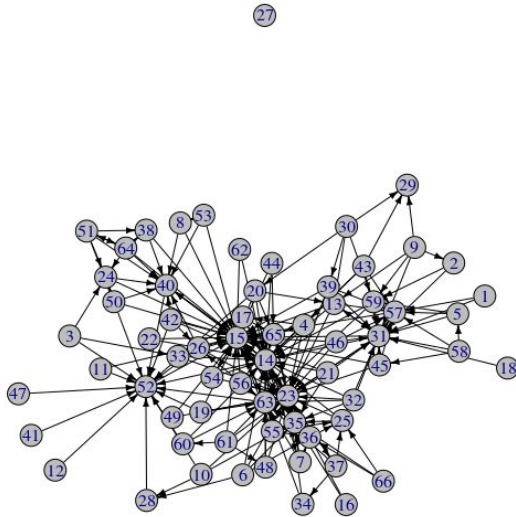


Lab 2

Exponential Random Graph Models

Part I Tips

- Plot the base (buy-in) network and include it in your report. Explain whether this plot supports or rejects Hypothesis 1 (Indegree popularity effects)?



- A. This question is asking what you think about H1 based on the plot. That said, there is no right and wrong to support or reject H1. Your reasoning is more important.
- B. Focus on whether there are few employees receiving more ties than others (that's what indegree popularity effects are).

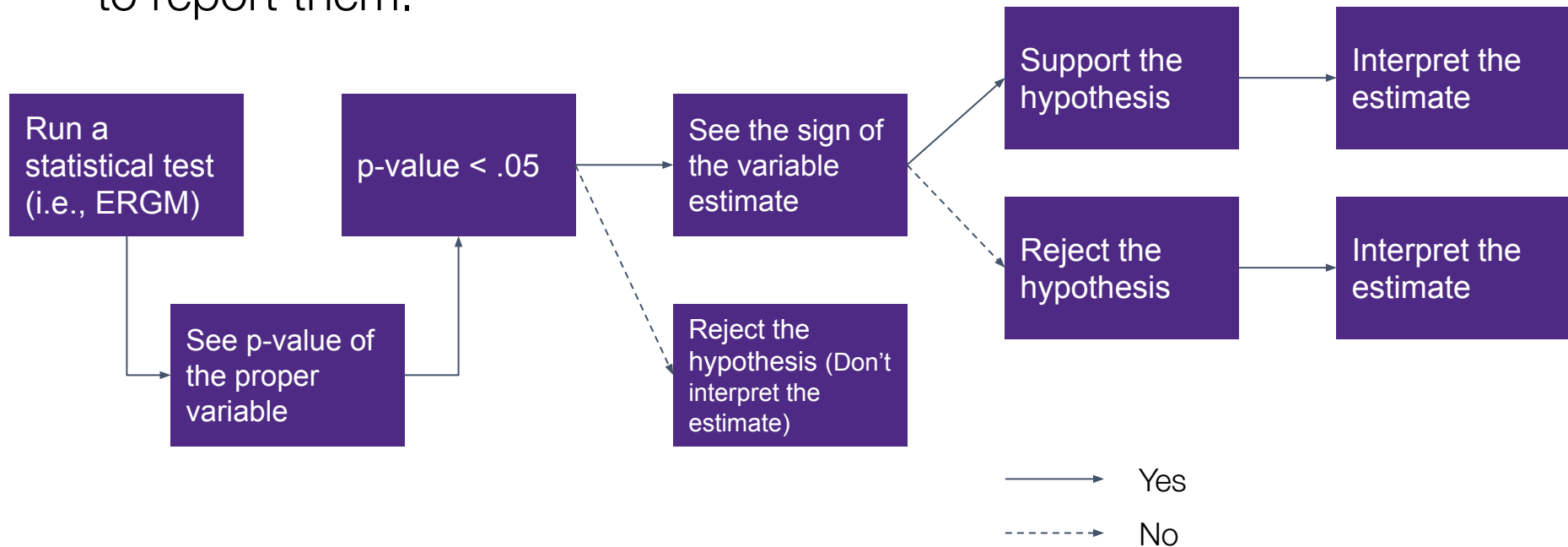
Hypothesis Testing

- A **hypothesis**: a conjecture statement about how two or more variables are related based on “educated guess”
- H_A : Individuals are *less* likely to report buy-in ties from others than not to report them.
- → The probability of having a buy-in tie in the network is *lower* than 50% (random chance).
- → There is a *negative* relationship between the number of buy-in ties and the network.

$$Network \sim -\theta * Edges$$

Hypothesis Testing

- H_A : Individuals are *less* likely to report buy-in ties from others than not to report them.



Results of Model 1

```
> summary(model1)
```

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

```
Formula:   buyIn ~ edges + mutual + edgecov(hundreds_messages)
```

```
Iterations: 5 out of 20
```

```
Monte Carlo MLE Results:
```

	Estimate	Std. Error	MCMC %	z value	Pr(> z)
edges	-2.73196	0.10011	0	-27.288	< 1e-04 ***
mutual	0.76882	0.31794	0	2.418	0.015601 *
edgecov.hundreds_messages	0.31061	0.08494	0	3.657	0.000255 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Null Deviance:    0.0  on 4290  degrees of freedom
Residual Deviance: -389.5  on 4287  degrees of freedom
```

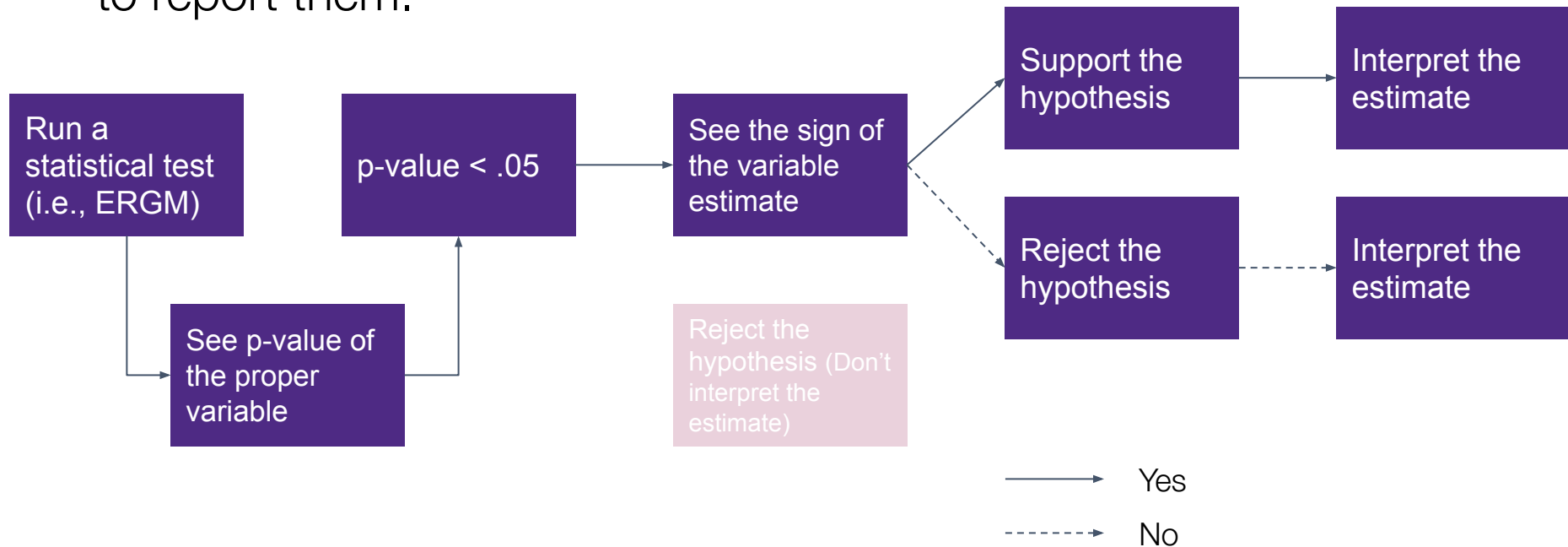
Note that the null model likelihood and deviance are defined to be 0. This means that all likelihood-based inference (LRT, Analysis of Deviance, AIC, BIC, etc.) is only valid between models with the same reference distribution and constraints.

```
AIC: -383.5    BIC: -364.4    (Smaller is better.)
```

$p < .05$ (meaning the estimate is not 0 ($0 \neq 0$))

Hypothesis Testing

- H_A : Individuals are **less** likely to report buy-in ties from others than not to report them.



Testing Hypothesis A

```
> summary(model1)
```

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

```
Formula:   buyIn ~ edges + mutual + edgecov(hundreds_messages)
```

```
Iterations: 5 out of 20
```

Negative → support the hypothesis

```
Monte Carlo MLE Results:
```

	Estimate	Std. Error	MCMC %	z value	Pr(> z)	
edges	-2.73196	0.10011	0	-27.288	< 1e-04	***
mutual	0.76882	0.31794	0	2.418	0.015601	*
edgecov.hundreds_messages	0.31061	0.08494	0	3.657	0.000255	***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

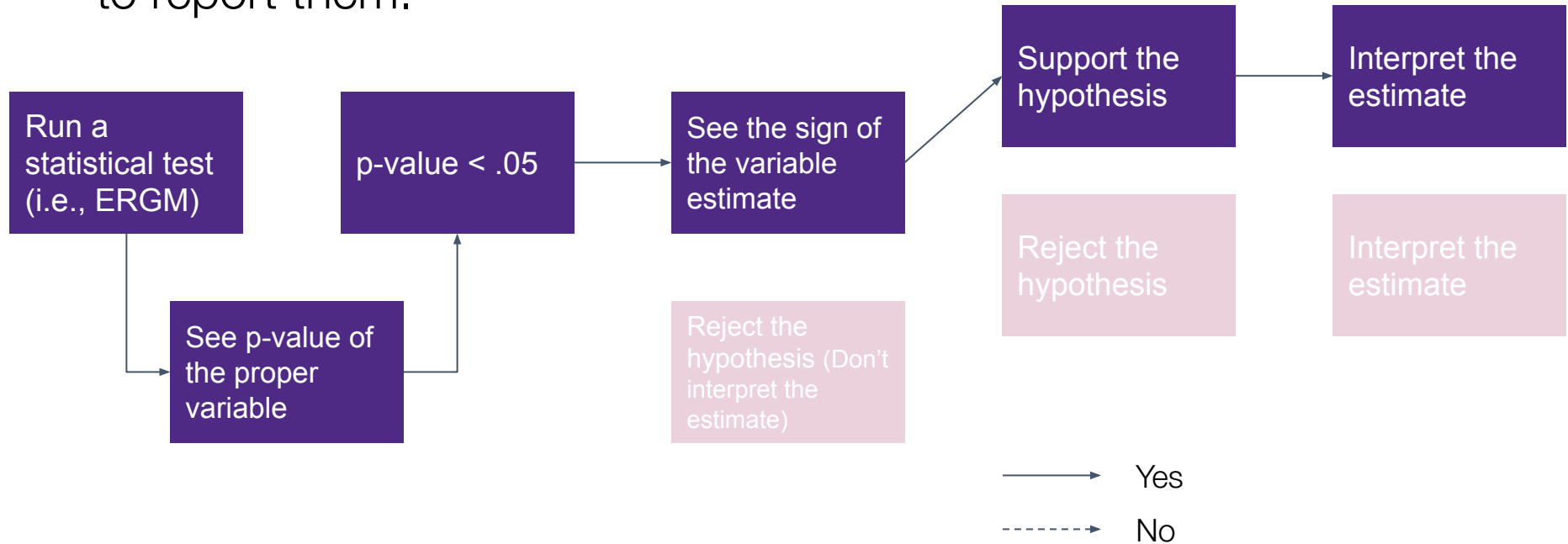
```
Null Deviance:    0.0  on 4290  degrees of freedom
Residual Deviance: -389.5  on 4287  degrees of freedom
```

Note that the null model likelihood and deviance are defined to be 0. This means that all likelihood-based inference (LRT, Analysis of Deviance, AIC, BIC, etc.) is only valid between models with the same reference distribution and constraints.

```
AIC: -383.5    BIC: -364.4    (Smaller is better.)
```

Testing Hypothesis A

- H_A : Individuals are **less** likely to report buy-in ties from others than not to report them.



Interpret the Estimate of Edges

Run $\exp(-2.73196)$

→ 0.06509159

- The estimate (θ): Conditional log-odds ratio
$$Network \sim -2.73196 * Edges$$
- Exponential: **odds ratio (if an odds ratio is 1, odds are even)**
 - The odds of individuals reporting buy-in ties are 0.07 times lower than the odds of individuals *not* reporting them

Interpret the Estimate of Edges

Run $\exp(-2.73196)/(1+\exp(-2.73196))$

→ 0.0611136

- The estimate (θ): Conditional log-odds ratio
$$Network \sim -2.73196 * Edges$$
- Inverse logit (p_{logis}): **probability (random chance is 50%)**
 - The probability of having a buy-in tie in the network is 6%.

One More Example in Model 2

- H1: There will be *indegree popularity effects* (tendency of a small number of nodes to receive many ties) in who people report is capable of getting buy in from them.

Geometrically weighted indegree measures a tendency against indegree preferential attachment

Monte Carlo MLE Results:

	Estimate	Std. Error	MCMC %	z value	Pr(> z)	
edges	-3.24101	0.40986	0	-7.908	< 1e-04	***
mutual	0.79745	0.67111	0	1.188	0.23474	
gwideg.fixed.1.06	-2.25771	0.35076	0	-6.437	< 1e-04	***
gwodeg.fixed.0.693147180559945	0.25273	0.64754	0	0.390	0.69632	
gwesp.OTP.fixed.0.693147180559945	0.92603	0.14131	0	6.553	< 1e-04	***
gwdsp.RTP.fixed.0.693147180559945	-1.37944	0.60425	0	-2.283	0.02244	*
nodematch.female	0.19908	0.15834	0	1.257	0.20865	
mix.leader.0.0	-0.66742	0.21215	0	-3.146	0.00166	**
mix.leader.1.0	-1.42147	0.61474	0	-2.312	0.02076	*
mix.leader.1.1	-0.50916	0.63961	0	-0.796	0.42601	
nodematch.department	2.02933	0.17996	0	11.276	< 1e-04	***
nodeicov.office	-0.30094	0.14439	0	-2.084	0.03714	*
nodeocov.office	0.20150	0.21665	0	0.930	0.35233	
diff.t-h.tenure	-0.13790	0.02374	0	-5.809	< 1e-04	***
edgcov.hundreds_messages	0.39560	0.09560	0	4.138	< 1e-04	***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Interpret the Estimate of H1

Run $\exp(-2.25771) / (1 + \exp(-2.25771))$

→ 0.0946865

Or run $\exp(2.25771) / (1 + \exp(2.25771))$

→ 0.9053135

- Inverse logit (`plogis`): **probability**
 - The probability of those who are *not* popular receiving a tie in the network is 9%.
 - The probability of those who are popular receiving a tie in the network is 91%.

Part III: Model Convergence

- **Model convergence:** examining whether the estimate process is converged or not
- **Goodness-of-fit test:** examining whether your model estimate represents your data

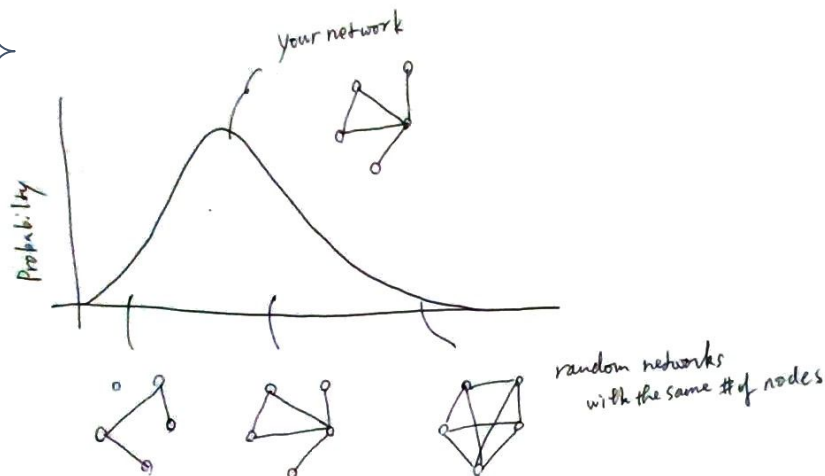
Model Convergence

- When you run `ergm(buyIn ~ edges + ...)` in R

```
The log-likelihood improved by 0.006663.  
Starting Monte Carlo maximum likelihood estimation (MCMLE):  
Iteration 1 of at most 20:  
Optimizing with step length 0.201157773508558.  
The log-likelihood improved by 3.067.  
Iteration 2 of at most 20:  
Optimizing with step length 0.234619875876688.  
The log-likelihood improved by 3.125.  
Iteration 3 of at most 20:  
Optimizing with step length 0.451674345548064.  
The log-likelihood improved by 3.155.  
Iteration 4 of at most 20:  
Optimizing with step length 1.  
The log-likelihood improved by 3.055.  
Step length converged once. Increasing MCMC sample size.  
Iteration 5 of at most 20:  
NOTE: Messages 'Error in mcexit(0L)...' may appear; please disregard them.  
Optimizing with step length 1.  
The log-likelihood improved by 1.101.  
Step length converged twice. Stopping.  
Finished MCMLE.  
Note: The constraint on the sample space is not dyad-independent. Null model likelihood  
is only implemented for dyad-independent constraints at this time. Number of  
observations is similarly poorly defined. This means that all likelihood-based  
inference (LRT, Analysis of Deviance, AIC, BIC, etc.) is only valid between models with  
the same reference distribution and constraints.  
Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14  
18 19 20 .  
This model was fit using MCMC. To examine model diagnostics and check for degeneracy,  
use the mcmc.diagnostics() function.
```

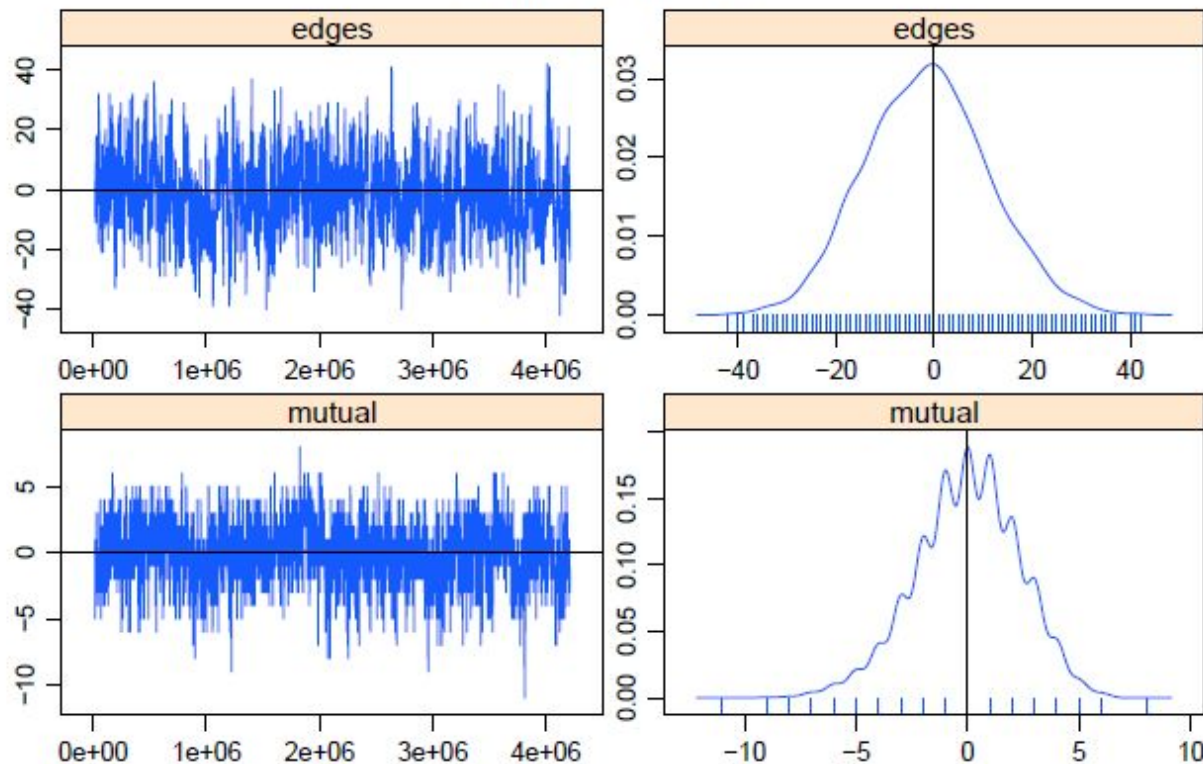
MCMC-MLE

- Markov Chain: because it simulates network Y_{t+1} randomly based on Y_t
- Monte Carlo: because of the computational implementation of the “randomly” generated part
- Maximum Likelihood Estimation



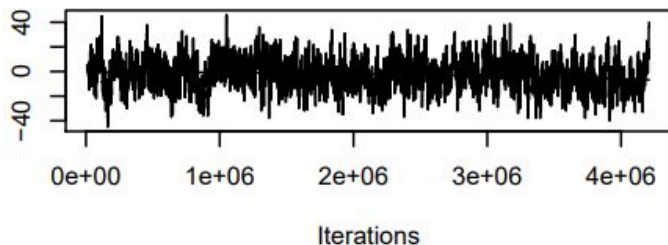
Producing a PDF file

Sample statistics

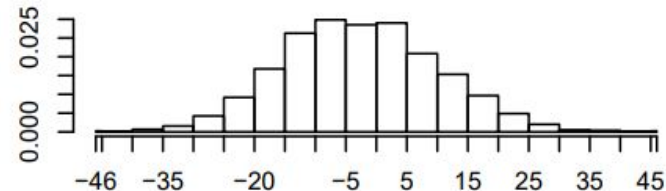


Some People Got This

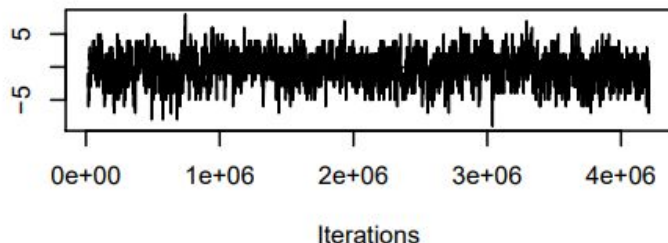
Trace of edges



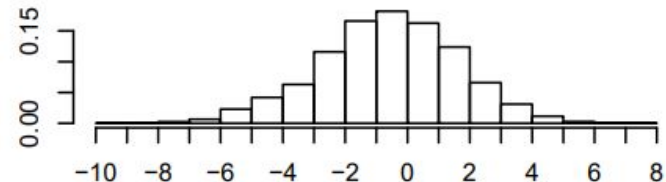
Density of edges



Trace of mutual

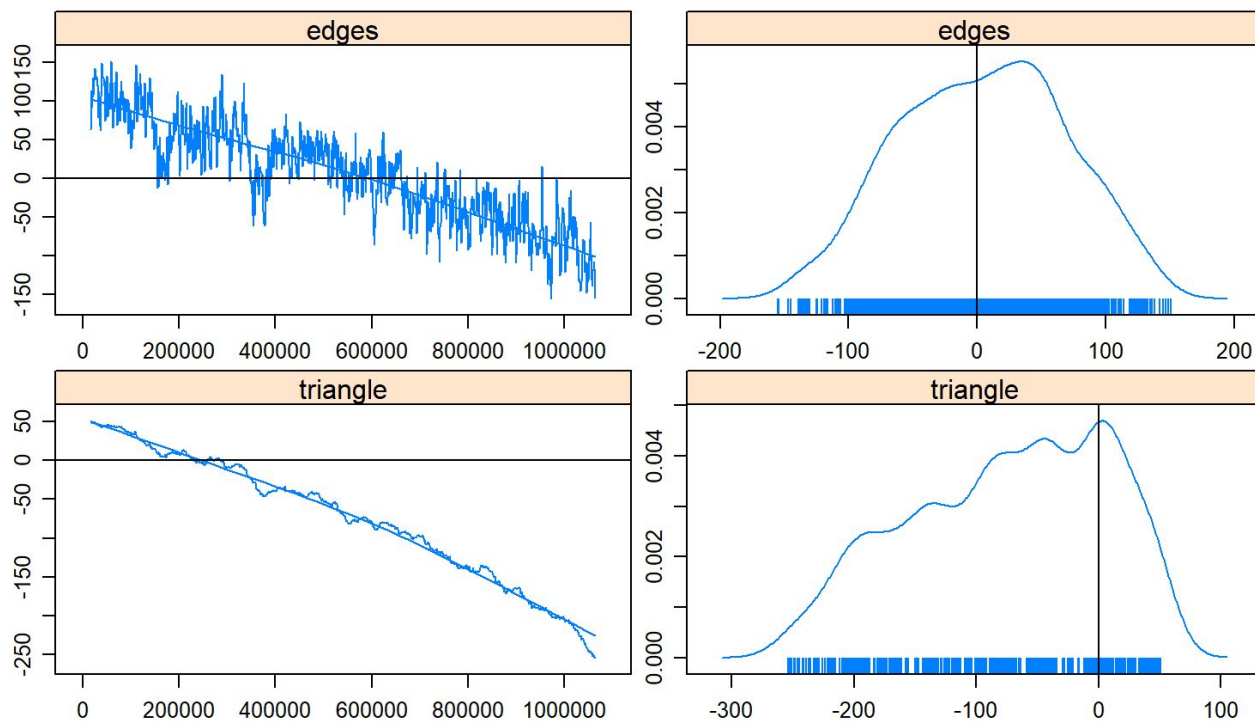


Density of mutual



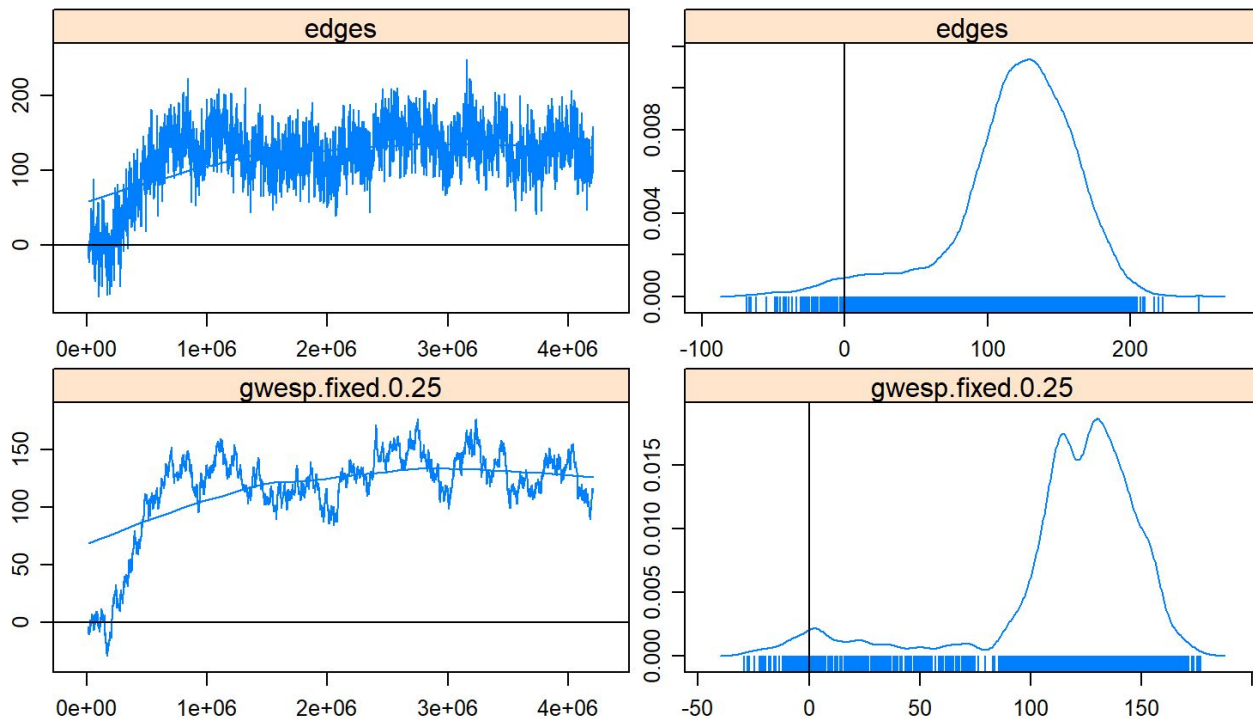
Bad Cases

Sample statistics



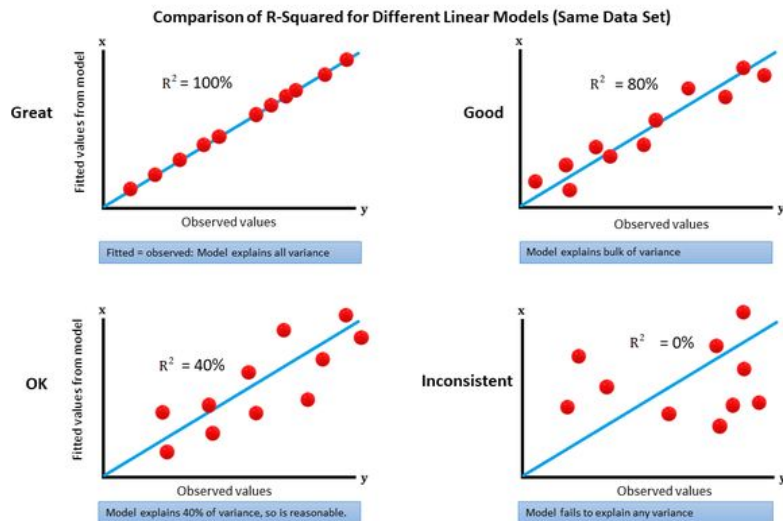
Bad Cases

Sample statistics



Goodness-of-fit Test


- **Does my model (i.e., the results of ERGM) fit well with my network data?**
- Think of a goodness-of-fit test as R-squared in regression



Goodness-of-fit Test

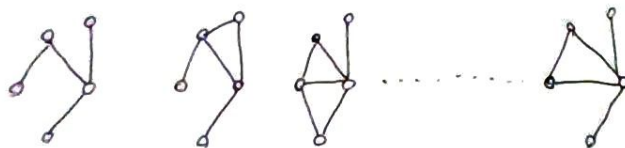
If you run `gof(model1 ~ ...)`

```
> gof <- gof(model ~ idegree + odegree + espartners + distance, verbose=T, burnin=1e+5, interval=1e+5,
control = control.gof.ergm(nsim = 200))
Starting GOF for the given ERGM formula.
Starting GOF for the given ERGM formula.
Calculating observed network statistics.
Starting simulations.
Sim 1 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 2 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 3 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 4 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 5 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 6 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 7 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 8 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 9 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 10 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 11 of 200: Starting MCMC iterations to generate 1 network
Finished simulation 1 of 1.
Sim 12 of 200: Starting MCMC iterations to generate 1 network
```

eg, the original = 

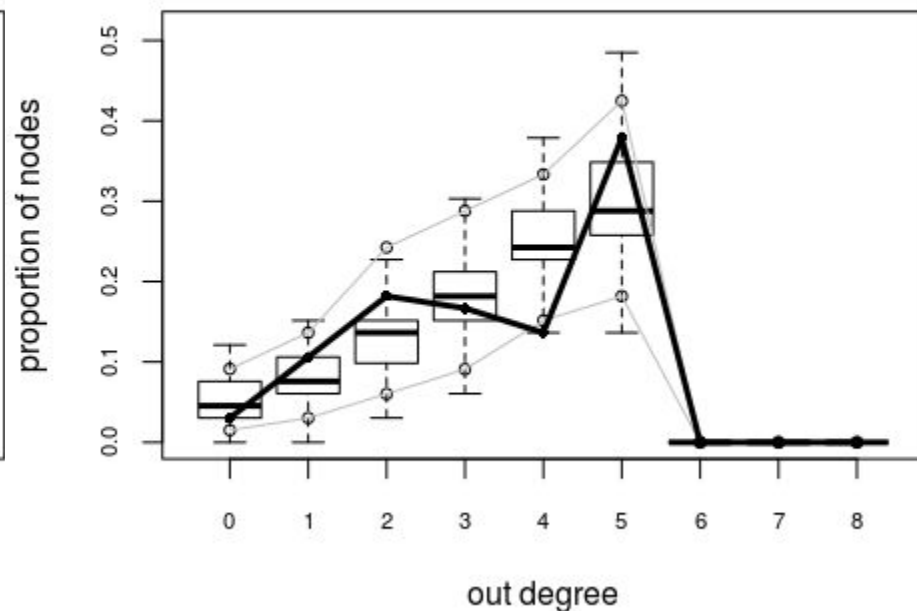
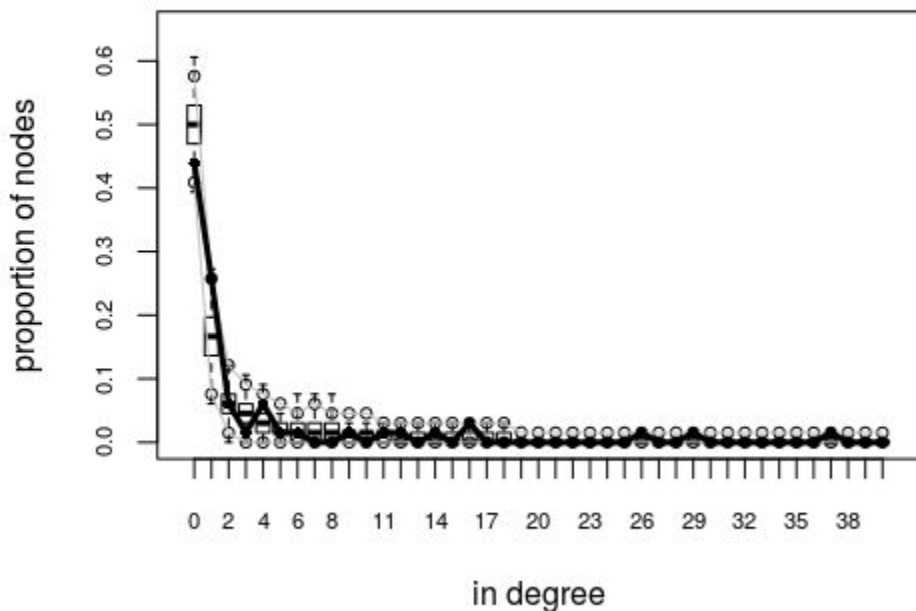
Network $\sim -3.24 \times \text{edges} + 0.79 \times \text{mutual} - 2.25 \times \text{gwdegree} + \dots + 0.39 \times \text{edgescov}$

↓ generate networks



Produce Plots in Your Plots Panel

If you run `plot(gof)`



Interpret the Output of GOF

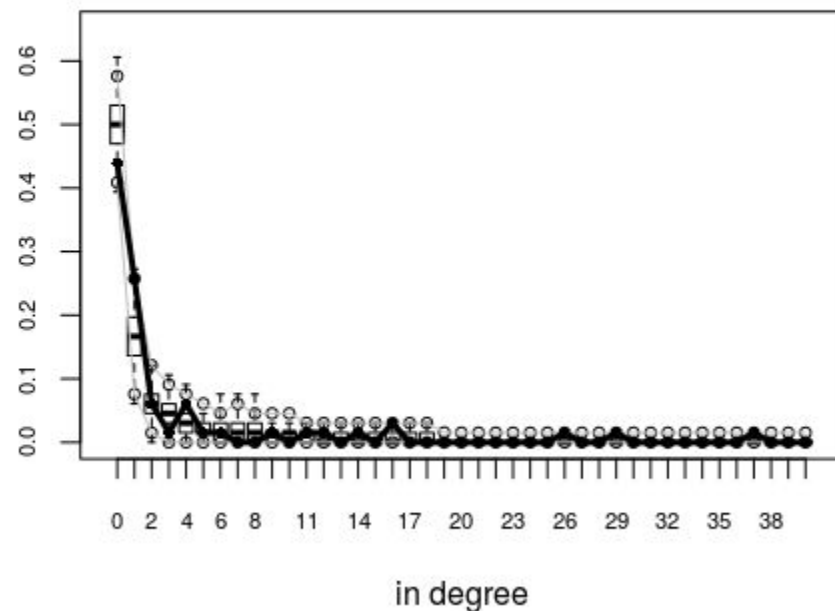
If you run `gof`

Goodness-of-fit for in-degree

	obs	min	mean	max	MC	p-value
0	29	23	32.905	43		0.27
1	17	4	10.685	20		0.08
2	4	0	4.100	10		1.00
3	1	0	2.765	8		0.45
4	4	0	1.875	6		0.21
5	1	0	1.640	6		1.00
6	1	0	1.255	5		1.00
7	0	0	1.335	5		0.59
8	0	0	1.050	5		0.64
9	1	0	0.930	5		1.00
10	0	0	0.745	4		0.92
11	1	0	0.735	3		1.00
12	1	0	0.570	4		0.90
13	0	0	0.435	3		1.00
14	1	0	0.395	3		0.69
15	0	0	0.360	3		1.00
16	2	0	0.375	3		0.09
17	0	0	0.380	2		1.00
18	0	0	0.295	3		1.00
19	0	0	0.260	2		1.00
20	0	0	0.185	2		1.00
21	0	0	0.200	2		1.00
22	0	0	0.165	2		1.00
23	0	0	0.180	2		1.00
24	0	0	0.140	1		1.00

=

proportion of nodes



Interpret the Output of GOF

If you run `gof`

Goodness-of-fit for in-degree

	obs	min	mean	max	MC p-value
0	29	23	32.905	43	0.27
1	17	4	10.685	20	0.08
2	4	0	4.100	10	1.00
3	1	0	2.765	8	0.45
4	4	0	1.875	6	0.21
5	1	0	1.640	6	1.00
6	1	0	1.255	5	1.00
7	0	0	1.335	5	0.59
8	0	0	1.050	5	0.64
9	1	0	0.930	5	1.00
10	0	0	0.745	4	0.92
11	1	0	0.735	3	1.00
12	1	0	0.570	4	0.90
13	0	0	0.435	3	1.00
14	1	0	0.395	3	0.69
15	0	0	0.360	3	1.00
16	2	0	0.375	3	0.09
17	0	0	0.380	2	1.00
18	0	0	0.295	3	1.00
19	0	0	0.260	2	1.00
20	0	0	0.185	2	1.00
21	0	0	0.200	2	1.00
22	0	0	0.165	2	1.00
23	0	0	0.180	2	1.00
24	0	0	0.140	1	1.00

Actual count from the original network

Counts from the simulated network based on the model (i.e., the results of ERGM)

p-value

Interpret the Output of GOF

If you run `gof`

Goodness-of-fit for in-degree

	obs	min	mean	max	MC p-value
0	29	23	32.905	43	0.27
1	17	4	10.685	20	0.08
2	4	0	4.100	10	1.00
3	1	0	2.765	8	0.45
4	4	0	1.875	6	0.21
5	1	0	1.640	6	1.00
6	1	0	1.255	5	1.00
7	0	0	1.335	5	0.59
8	0	0	1.050	5	0.64
9	1	0	0.930	5	1.00
10	0	0	0.745	4	0.92
11	1	0	0.735	3	1.00
12	1	0	0.570	4	0.90
13	0	0	0.435	3	1.00
14	1	0	0.395	3	0.69
15	0	0	0.360	3	1.00
16	2	0	0.375	3	0.09
17	0	0	0.380	2	1.00
18	0	0	0.295	3	1.00
19	0	0	0.260	2	1.00
20	0	0	0.185	2	1.00
21	0	0	0.200	2	1.00
22	0	0	0.165	2	1.00
23	0	0	0.180	2	1.00
24	0	0	0.140	1	1.00

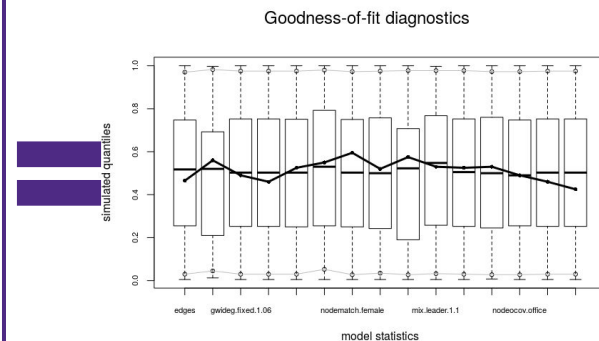
If $p > .05$, it indicates that the model fits well with the original network in terms of the in-degree distribution

Interpret the Output of GOF

If you run `gof`

Goodness-of-fit for model statistics

	obs	min	mean	max	MC p-value
edges	225.00000	192.00000	223.74500	255.00000	0.93
mutual	12.00000	7.00000	11.89000	18.00000	1.00
gwideg.fixed.1.06	66.19405	51.63342	65.82389	81.20335	0.98
gwodeg.fixed.0.693147180559945	109.56250	99.50000	108.90937	116.68750	0.92
gwesp.OTP.fixed.0.693147180559945	214.12500	161.37500	214.08812	266.75000	0.95
gwdsp.RTP.fixed.0.693147180559945	5.00000	0.00000	5.12625	14.00000	1.00
nodematch.female	162.00000	132.00000	163.50500	197.00000	0.96
mix.leader.0.0	94.00000	62.00000	93.81500	132.00000	1.00
mix.leader.1.0	8.00000	3.00000	8.00000	17.00000	1.00
mix.leader.1.1	12.00000	5.00000	11.85500	18.00000	1.00
nodematch.department	112.00000	87.00000	111.38500	142.00000	1.00
nodeicov.office	182.00000	155.00000	182.89000	222.00000	1.00
nodeocov.office	173.00000	152.00000	172.70000	193.00000	0.98
diff.t-h.tenure	-737.25753	-938.29315	-750.30788	-594.08219	0.92
edgescov.hundreds_messages	56.69000	25.12000	53.87505	86.44000	0.85

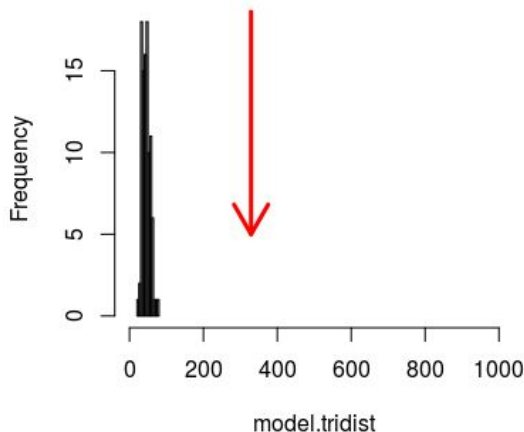


If $p > .05$, it indicates that the model fits well with the original network in terms of the variables

GOF for Triangles

Model 1

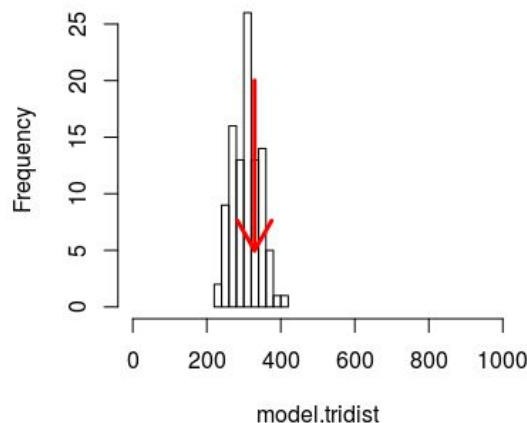
Histogram of model.tridist



t-statistic = 26.3656
→ $p < .001$

Model 2

Histogram of model.tridist



t-statistic = 0.6205
→ $p > .50$

If $p > .05$, it indicates that the model fits well with the original network in terms of the triangles

`2*pt(t, df=100, lower.tail = FALSE)`

Q & A

ERGM Terms

- This is not for Lab 2, but if you use ERGMs for your final project, please take a look at this page:
<https://cran.r-project.org/web/packages/ergm/vignettes/ergm-term-crossRef.html>

Basic / Frequently-used term category matrix

For convenience, this table lists a subset of the most commonly-used ergm terms and categories.

Term name	binary	valued	directed	undirected	bipartite	dyad-independent
absdiff	✓		✓	✓		✓
b1cov	✓			✓	✓	✓
b1cov		✓		✓	✓	✓
b1degree	✓			✓	✓	
b1factor	✓			✓	✓	✓
b1factor		✓		✓	✓	✓
b1nodematch	✓			✓	✓	✓
b2concurrent	✓			✓	✓	