ERGMs: General statistical models

Can you control for more than just density?

What if you want to test multiple network features?

And you want a model grounded in generative theory?

… That’s when you need ERGMs

Limitations of simple null model tests

- If we are ***only*** interested in whether the triangle counts are different than expected given the density of the graph
  - One can use these simple null hypothesis tests
  - Like a t-test in traditional statistics

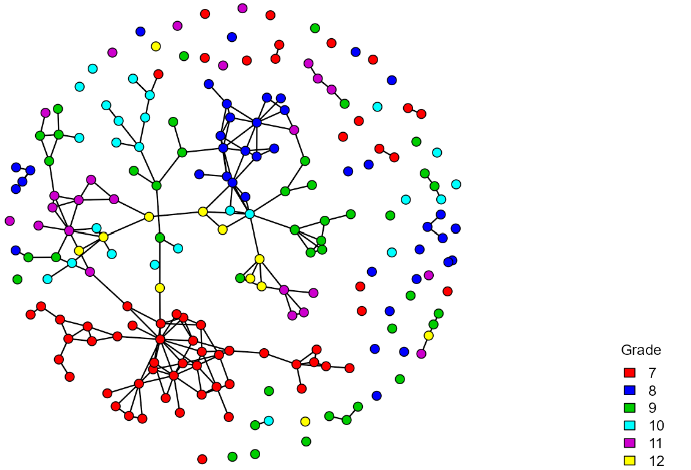
- But if we want to understand the underlying generative process, quantify the impact of each process on our network, and control for other network features …
  - This requires a *general statistical modeling framework*

Motivation: explaining clusters

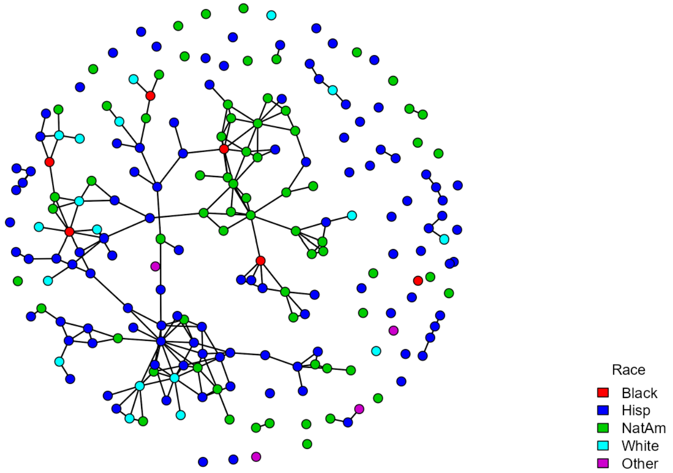
What do you see when color-coding the nodes by attributes?

- How much of the clustering is based on grade?
- How much of it is based on race?

Coloring by Grade



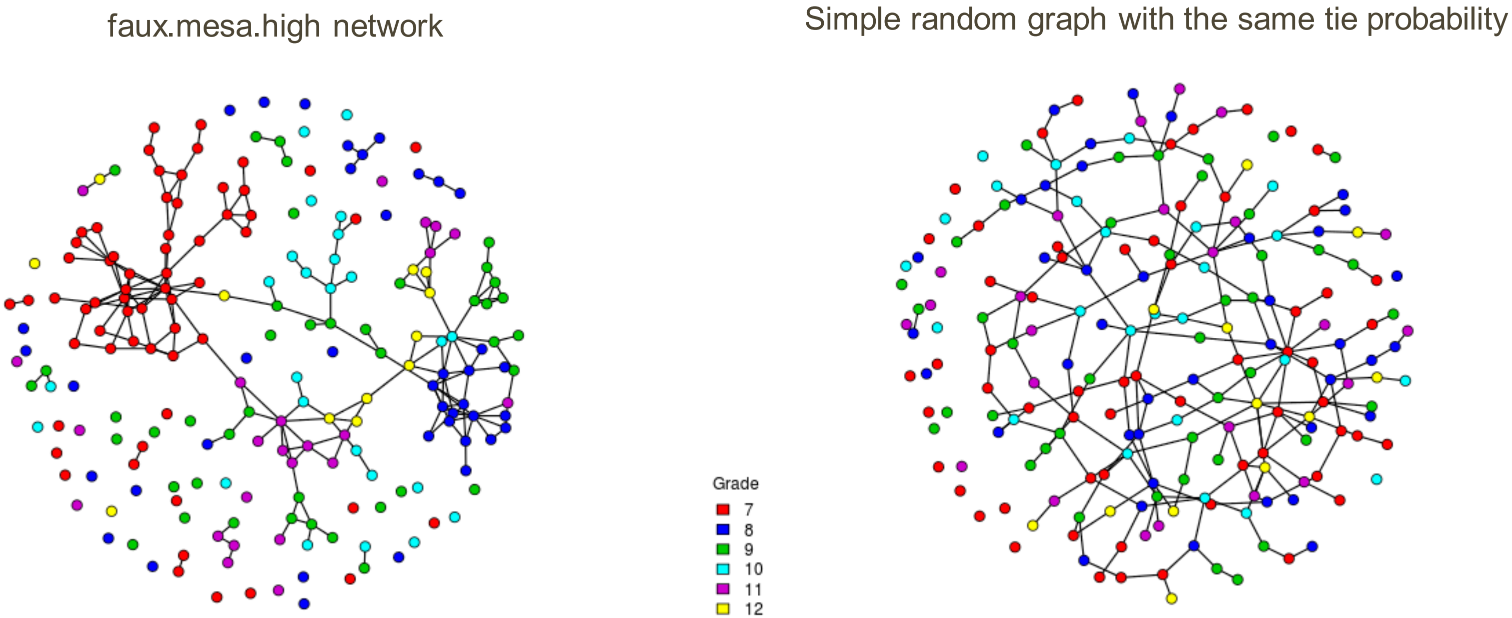
Coloring by Race



Motivation: explaining triangles

Why are there so many more triangles in faux.mesa.high?

- Is it a propensity for triad formation?
- Or just a by-product of grade homophily?



faux.mesa.high network

Simple random graph with the same tie probability

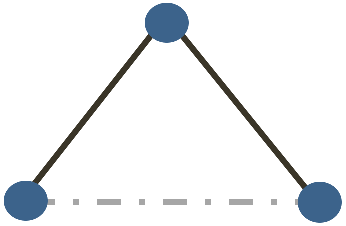
Friend of a friend, or birds of a feather?

Two theories about the process that generates triangles:

1. **Homophily**: People tend to chose friends who are like them, in terms of grade, race, etc. (*"birds of a feather"*), triad closure is a by-product

2. **Transitivity**: People who have friends in common tend to become friends (*"friend of a friend"*), triad closure is the key process

So, for three actors in the same grade



A cycle-closing tie may form due to transitivity

But it may be due instead to homophily

Transitivity and homophily are *partially* confounded

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

	Triangle forming:	
Within Grade:	*Yes*	*No*
Yes	Both	Homophily
No	Transitivity	*Neither*

The cells represent how the processes jointly influence that tie, so the distribution of ties in this table is informative.

This suggests we should be able to disentangle the two processes statistically

ERGMs: Basic idea

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

ERGMs are *a type of* generalized linear model

If you’re familiar with logistic regression, much of this will look familiar

- Both use a logit link: $logit(p) = log\left(\frac{p}{(1-p)}\right)$
- Both have a likelihood function for the data
- Both are exponential-family models

  They inherit all the theory and nice properties that go with those ☺
- And both provide a general framework for modeling

  Not just a way to model one or two specific effects

But ERGMs are also different

- They relax the assumption of independence
  - So they are technically “auto-logistic” regression
- The observations on the LHS are links, not nodes
  - So every prediction is referring to a pair of nodes
  - This is not an “individual-based” model
- And the terms on the RHS are network statistics

ERGM: for the probability of the network

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

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

Note: this is the likelihood function

Where: $\mathbf{Y}$ = the network (as a random variable)

$\boldsymbol{g}(\boldsymbol{y})$ = vector of network statistics

$\boldsymbol{\theta}$ = vector of model parameters

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

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

$$\exp\left(\boldsymbol{\theta}' \boldsymbol{g}(\boldsymbol{y})\right) = exp(\theta_1 g_1(y) + \theta_2 g_2(y) + \cdots + \theta_p g_p(y))$$

ERGM: for the conditional log-odds of a tie

The conditional log odds of a specific tie, y_{ij}

$$logit\left(P\left(Y_{ij}=1\middle|\text{rest of the graph}\right)\right)=log\left(\frac{P\left(Y_{ij}=1\middle|\text{rest of the graph}\right)}{P\left(Y_{ij}=0\middle|\text{rest of the graph}\right)}\right)$$

$$=\boldsymbol{\theta}'\boldsymbol{\partial}\left(\boldsymbol{g}(\boldsymbol{y})\right)$$

After some algebra…

Where: $\boldsymbol{\partial}\left(\boldsymbol{g}(\boldsymbol{y})\right)$ = the <u>change</u> in $\boldsymbol{g}(\boldsymbol{y})$ when y_{ij} is toggled between 0 and 1

$$logit(\boldsymbol{p}|\boldsymbol{y}^{c})=\theta_1\partial(g_1(y))+\theta_2\partial(g_2(y))+\cdots+\theta_p\partial\left(g_p(y)\right)$$

This is an <u>auto</u> logistic regression (auto because of the possible dependence)

ERGM model specification: $\boldsymbol{g}(\boldsymbol{y})$

The $\boldsymbol{g}(\boldsymbol{y})$ terms in the model are summary "network statistics"

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

Just examples, any other configurations can be counted

- A key distinction in the types of terms:
  - Dyad independent (1 & 2 are examples)
  - Dyad dependent (3 & 4 are examples)

Origins of this modeling class

- Paper by Ove Frank and David Strauss in 1986

  *Markov Graphs. JASA, 1986. 81: 832-842***.**

  Defined 1-step Markov dependence in graphs:

  links are only dependent if they share a node

  Showed the sufficient statistics for Markov dependence are k-stars and triangles

This caused a few problems later, as we'll see in module 3…

ERGM specification: $\boldsymbol{\theta}'\boldsymbol{g}(\boldsymbol{y})$

Model specification involves:

1. Choosing the set of network statistics $\boldsymbol{g}(\boldsymbol{y})$
   - From minimal : # of edges
   - To saturated: one term for every dyad in the network

   NB: statnetWeb allows you to choose from the list of terms and retrieve documentation for each one

2. Choosing "homogeneity constraints" on the parameter θ, for example, with edges:
   - all homogeneous
   - heterogeneous by group
   - heterogeneous by node (as fixed or random effects)

Lab

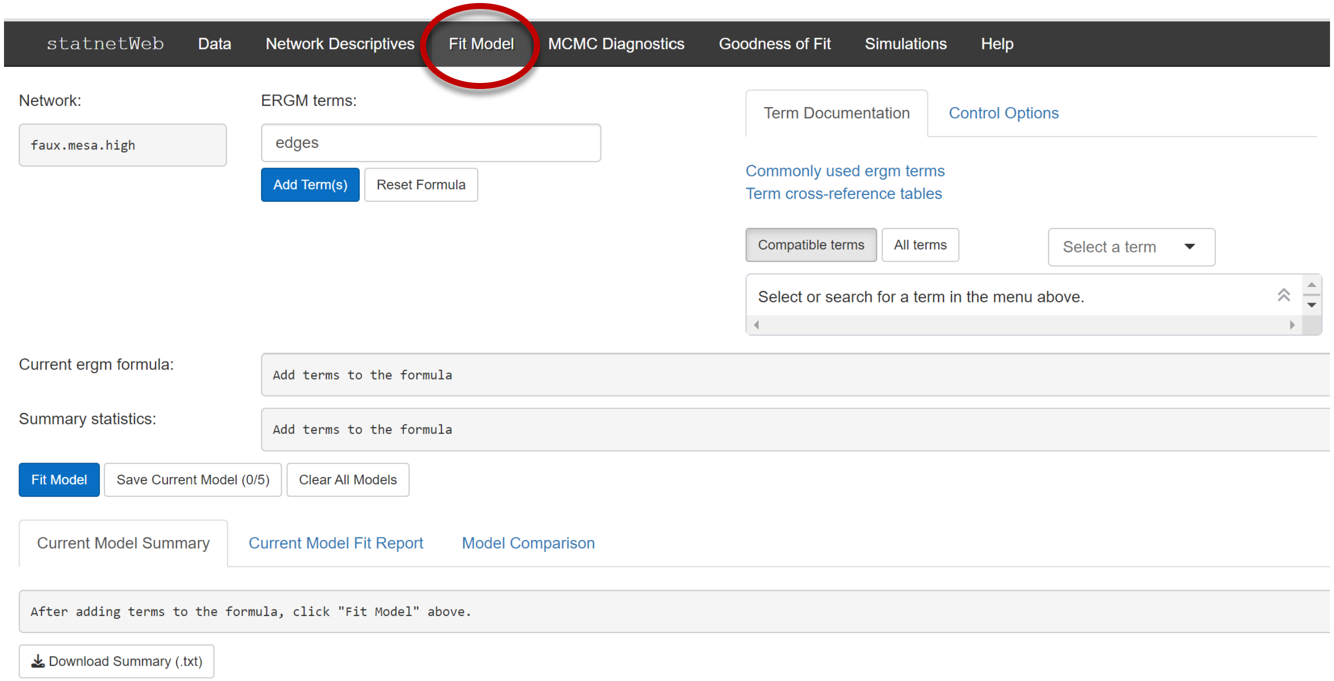
Let's explore ERGMs for faux.mesa.high in statnetWeb

We will compare five models

Model	Network Statistics g(y)
Edges	# edges
Edges + nodal attributes (activity levels)	# edges # edges for each grade and race group
Edges + nodal attributes + mixing by attributes (homophily)	# of edges # edges for each grade and race group # edges that are within-race & within-grade *Model 3: uniform homophily; Model 4: differential homophily*
Edges + nodal attributes + mixing by attributes + degree(0)	# of edges # edges for each grade and race group # edges that are within-race & within-grade (DH) # Isolates

In statnetWeb

- Load the faux.mesa.high data again
- Select the “Fit Model” tab



Network: faux.mesa.high

ERGM terms: edges

Select or search for a term in the menu above.

Current ergm formula:

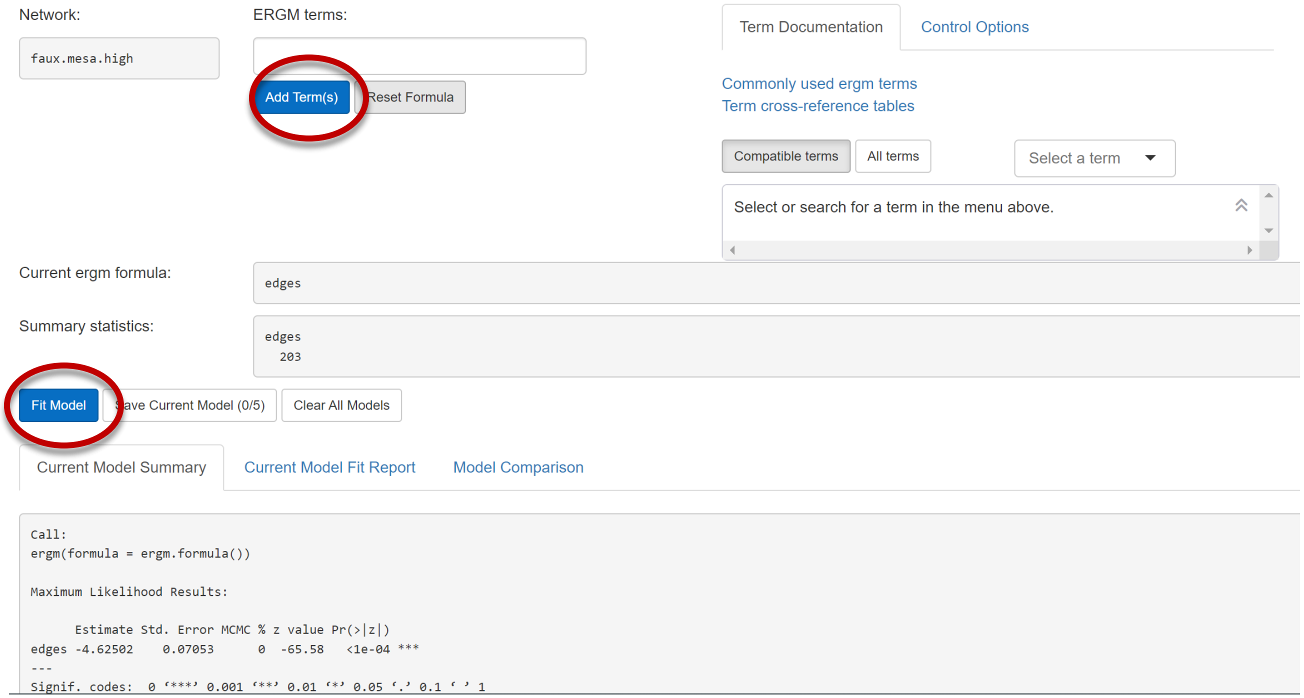
```
Add terms to the formula
```

Summary statistics:

```
Add terms to the formula
```

```
After adding terms to the formula, click "Fit Model" above.
```

Add the edges term and fit the model



Network: faux.mesa.high

ERGM terms:

Current ergm formula: edges

Summary statistics:

```
edges
  203
```

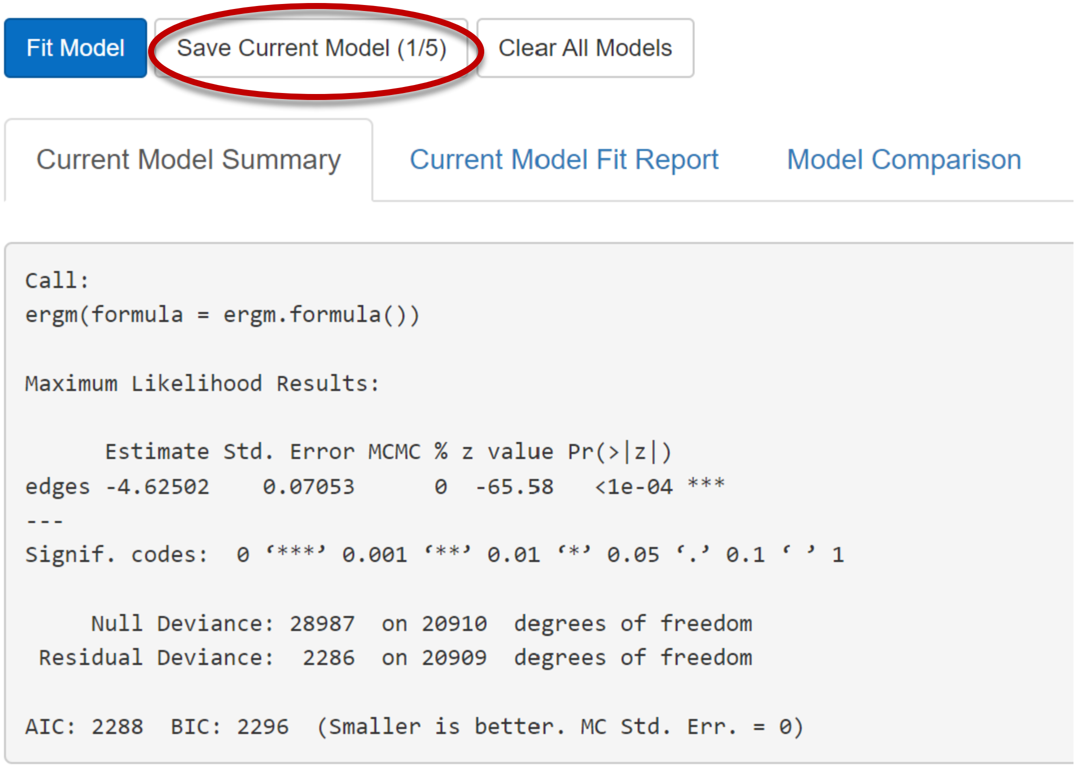
```
Call:
ergm(formula = ergm.formula())

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges -4.62502    0.07053      0  -65.58   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Note how the value of the network statistic is displayed

Note the model output, and save it



```
Call:
ergm(formula = ergm.formula())

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges -4.62502    0.07053      0  -65.58   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 28987  on 20910  degrees of freedom
 Residual Deviance:  2286  on 20909  degrees of freedom

AIC: 2288  BIC: 2296  (Smaller is better. MC Std. Err. = 0)
```

- Interpret the coefficient

  Log-odds of a tie = -4.6 so

  P(tie) = 0.009 (about 1%)

- Statistically significant?

  Yes, p < .0001

- Meaning?

  Density is much less than 50%

  (density = 50% if coef = 0)

This model is not intrinsically interesting
But as a natural null model it is good for comparison

Repeat the steps for each model

Add the terms below; fit the models; save the fits

- Model 2: `nodefactor("Grade")+nodefactor("Race")`
- Model 3: `nodematch("Grade")+nodematch("Race")`

- Model 4: *reset the model and add this:*

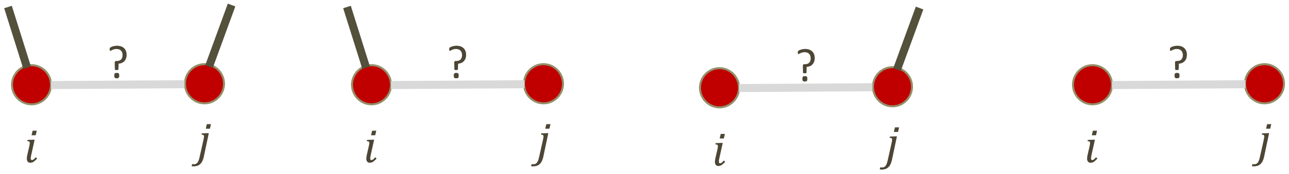
```
edges+nodefactor("Grade")+nodefactor("Race")+nodematch("Grade",
diff=T)+nodematch("Race", diff=T)
```

- Model 5: `degree(0)` that's a zero, not the letter "O"

The fit takes longer for Model 5. Why?

- Because `degree(0)` is a “**dyad-dependent**” term

- Now the probability of a tie between nodes i and j depends on whether either node currently has any ties
  - If i or j has no ties, then this will change the number of isolates
  - There are 4 possible cases (since >1 tie has the same effect as 1 tie):

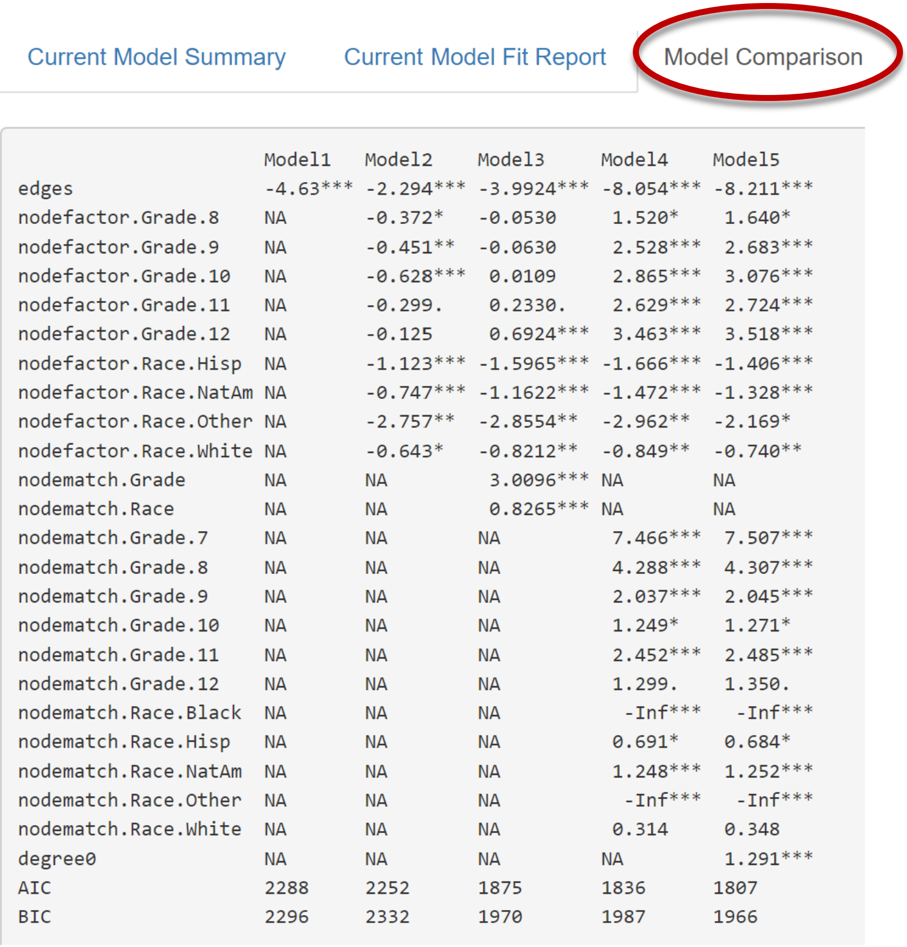


$\partial(g(y))$	0	-1	-1	-2	Change statistic value

Dyad dependent terms change estimation

- When all model terms are “dyad-independent”
  - ergm uses the same algorithm as logistic regression
  - very quick
- When you add a dyad dependent term
  - This changes the estimation algorithm to MCMC
    - Markov Chain Monte Carlo
  - This takes longer
- We’ll explain all this in module 3

Click the model comparison tab



```
                      Model1   Model2    Model3     Model4    Model5
edges                 -4.63*** -2.294*** -3.9924*** -8.054*** -8.211***
nodefactor.Grade.8    NA       -0.372*   -0.0530     1.520*    1.640*
nodefactor.Grade.9    NA       -0.451**  -0.0630     2.528***  2.683***
nodefactor.Grade.10   NA       -0.628***  0.0109     2.865***  3.076***
nodefactor.Grade.11   NA       -0.299.    0.2330.    2.629***  2.724***
nodefactor.Grade.12   NA       -0.125     0.6924***  3.463***  3.518***
nodefactor.Race.Hisp  NA       -1.123*** -1.5965*** -1.666*** -1.406***
nodefactor.Race.NatAm NA       -0.747*** -1.1622*** -1.472*** -1.328***
nodefactor.Race.Other NA       -2.757**  -2.8554**  -2.962**  -2.169*
nodefactor.Race.White NA       -0.643*   -0.8212**  -0.849**  -0.740**
nodematch.Grade       NA       NA         3.0096*** NA        NA
nodematch.Race        NA       NA         0.8265*** NA        NA
nodematch.Grade.7     NA       NA        NA          7.466***  7.507***
nodematch.Grade.8     NA       NA        NA          4.288***  4.307***
nodematch.Grade.9     NA       NA        NA          2.037***  2.045***
nodematch.Grade.10    NA       NA        NA          1.249*    1.271*
nodematch.Grade.11    NA       NA        NA          2.452***  2.485***
nodematch.Grade.12    NA       NA        NA          1.299.    1.350.
nodematch.Race.Black  NA       NA        NA           -Inf***   -Inf***
nodematch.Race.Hisp   NA       NA        NA          0.691*    0.684*
nodematch.Race.NatAm  NA       NA        NA          1.248***  1.252***
nodematch.Race.Other  NA       NA        NA           -Inf***   -Inf***
nodematch.Race.White  NA       NA        NA          0.314     0.348
degree0               NA       NA        NA         NA         1.291***
AIC                   2288     2252      1875       1836      1807
BIC                   2296     2332      1970       1987      1966
```

Looks like standard statistical output

And that's exactly the point

This is a principled, fully general approach to statistical estimation and inference for network analysis

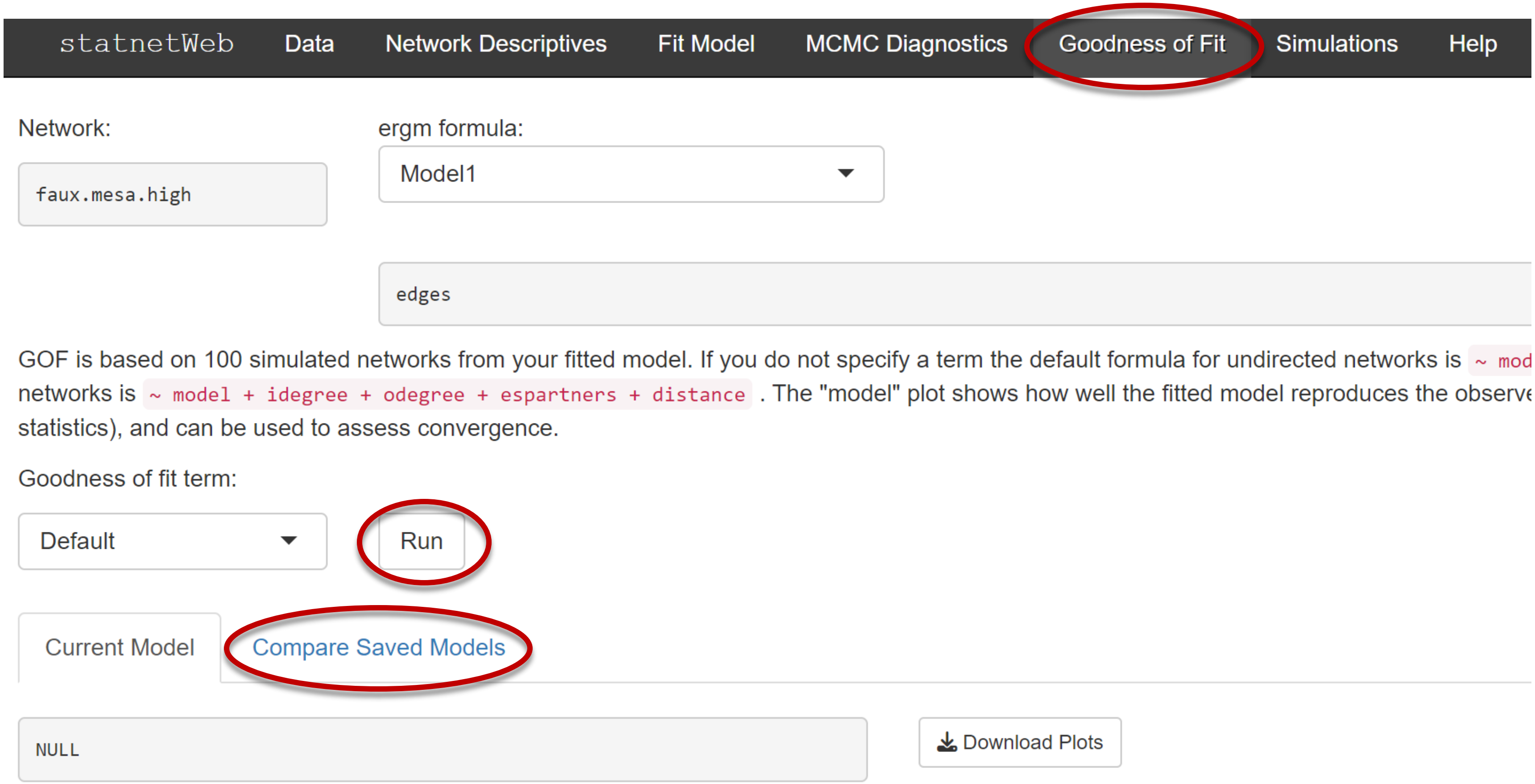
What does it tell you?

Quick preview of model assessment

- The model summaries on the previous slide are one form of assessment
  - Which individual terms are significant
  - AIC and BIC for model comparison

- But there's a network-specific assessment also
  - Does the model reproduce network statistics that are NOT included in the model?
  - If so, then it is a parsimonious summary of the generative processes that produce the overall network structure
  - Like "out of sample" prediction in other settings

Select in order: Goodness of Fit, Compare Saved Models, and Run



Network: faux.mesa.high

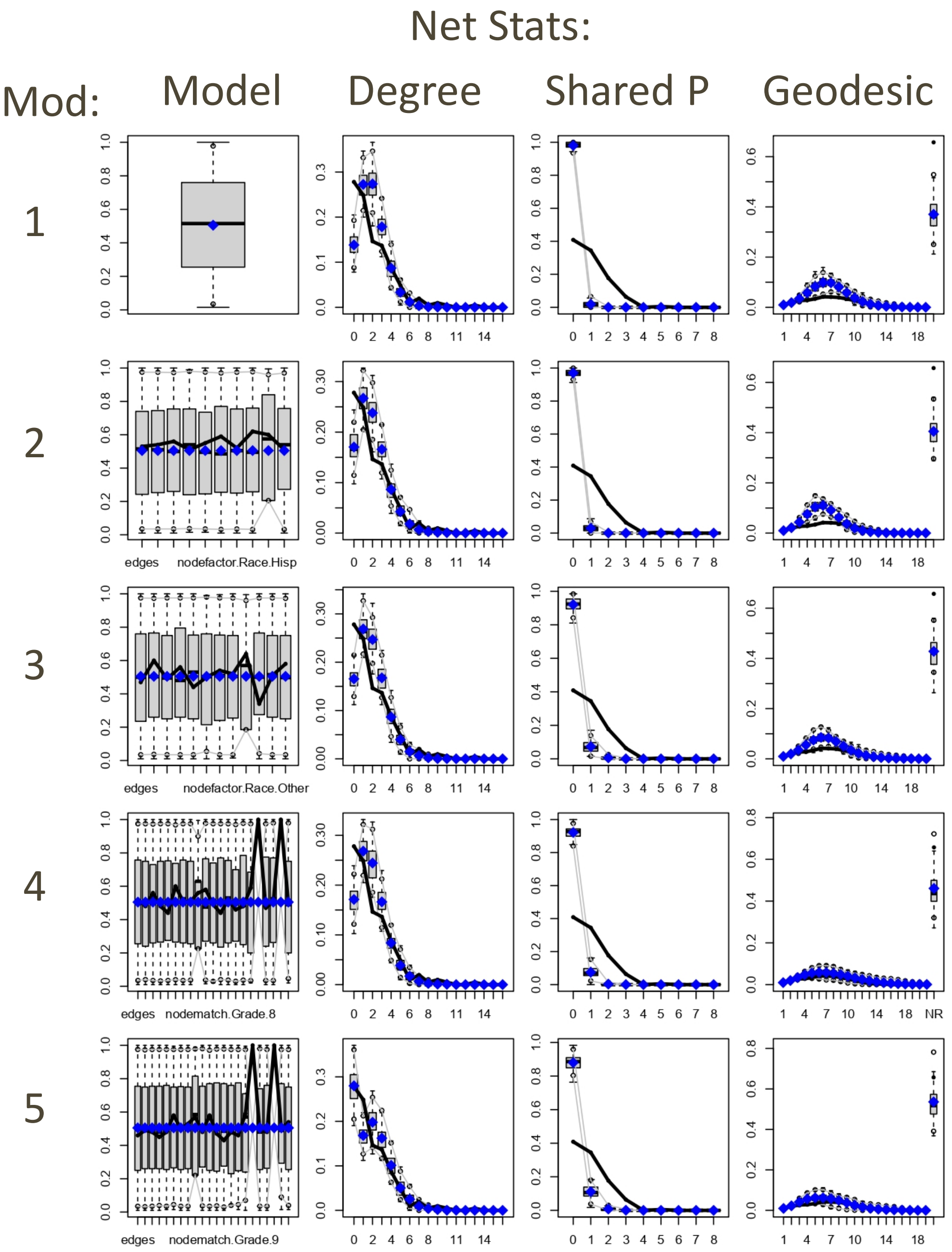
ergm formula: Model1

edges

GOF is based on 100 simulated networks from your fitted model. If you do not specify a term the default formula for undirected networks is `~ mod` networks is `~ model + idegree + odegree + espartners + distance` . The "model" plot shows how well the fitted model reproduces the observe statistics), and can be used to assess convergence.

Goodness of fit term: Default

```
NULL
```



There is a LOT of information here

And we haven't discussed how these plots are constructed yet

But the key take home messages are:

1. Attribute levels and mixing dial in the geodesics
2. Degree(0) captures the isolates
3. None of these models captures the shared partner distribution

So this was a very quick tour

- Of ERGMs in practice

- Just to give a sense of:
  - How easy it is to explore different model specifications
  - How easy it is to jointly estimate the impact of an arbitrary number of covariates
  - How familiar the statistical inference feels

- And we haven’t shown simulations from the model
  - But those are easy too

Next:

Temporal ERGMs – TERGMs

But first … a break