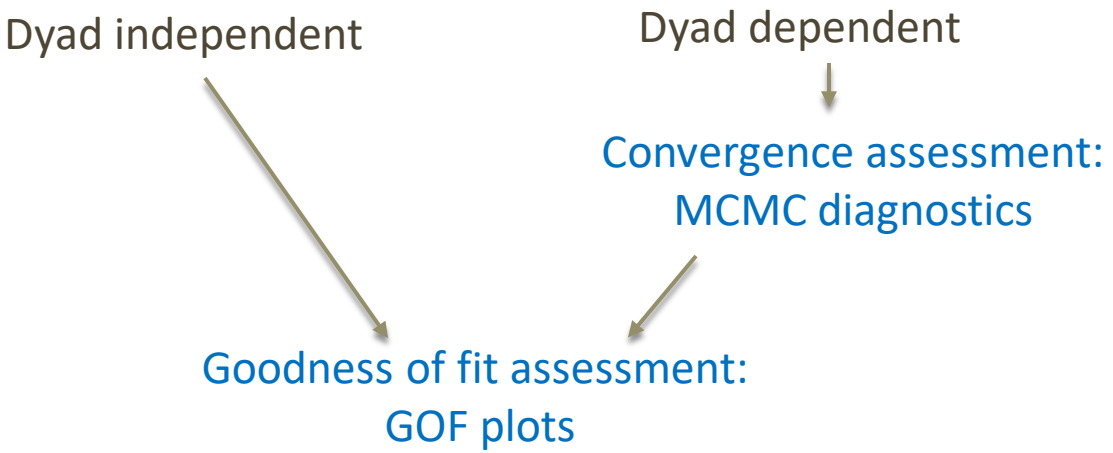
ERGM model assessment

Also varies by whether the model include dyadic dependent terms or not

Fitting and diagnosing a model

- The steps depend on the type of model you have
  - If you have a **dyad dependent** model, you first check convergence

- In both cases you end with goodness of fit:



We'll start with MCMC dx

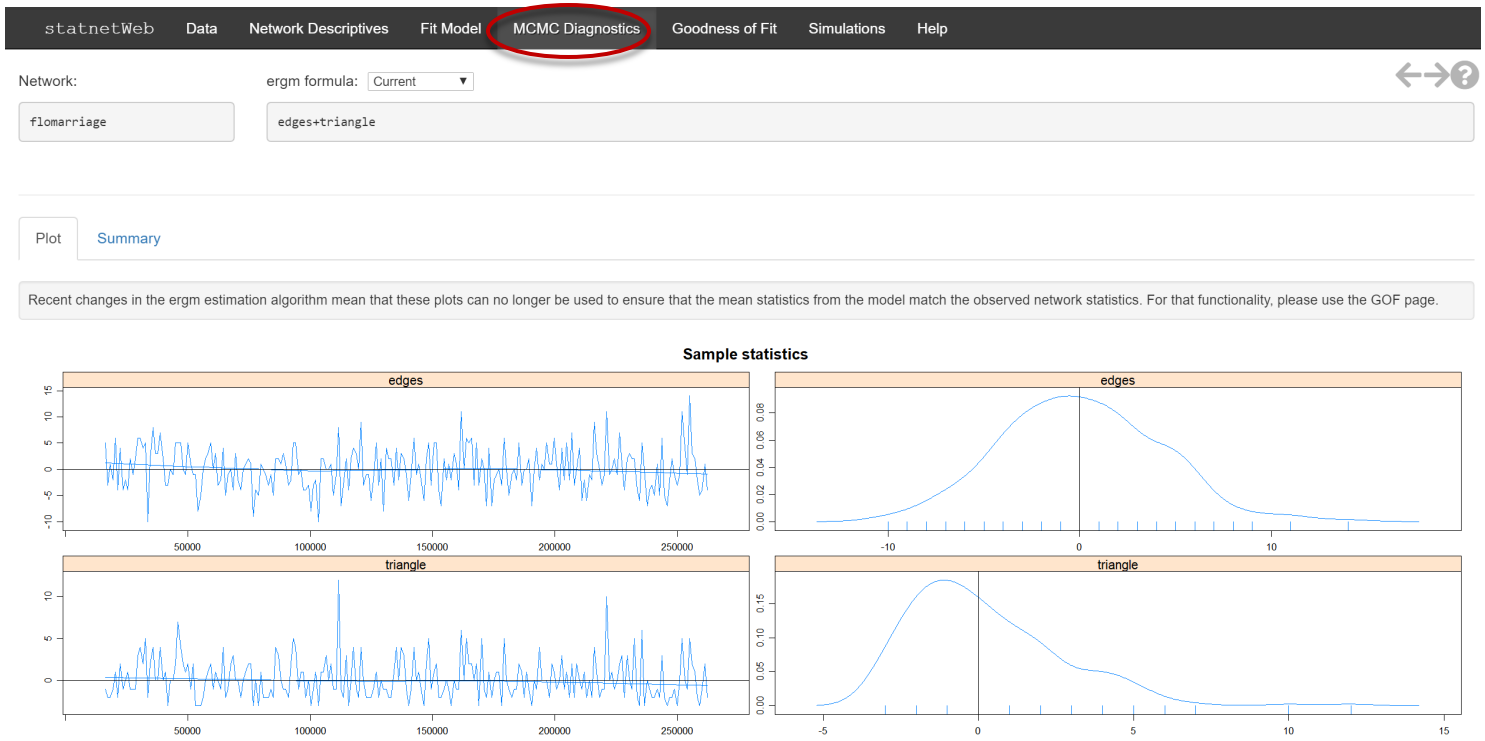
- Because you need to do this first
- But it's a whole workshop in it's own right
- So this is just to get started

What are MCMC Diagnostics?

- MCMC Dx show the details of the sampling process
  - Traditionally taken from the last iteration in the MCMC chain

- They tell us if the estimation algorithm:
  - Is "mixing well"
    - Is it getting stuck in part of the space for many timesteps?
    - Is there a lot of autocorrelation in the samples?
  - Converged to the target value
    - Is it still bouncing around a lot?
    - Does the sampling distribution of the stats look bell shaped or not?

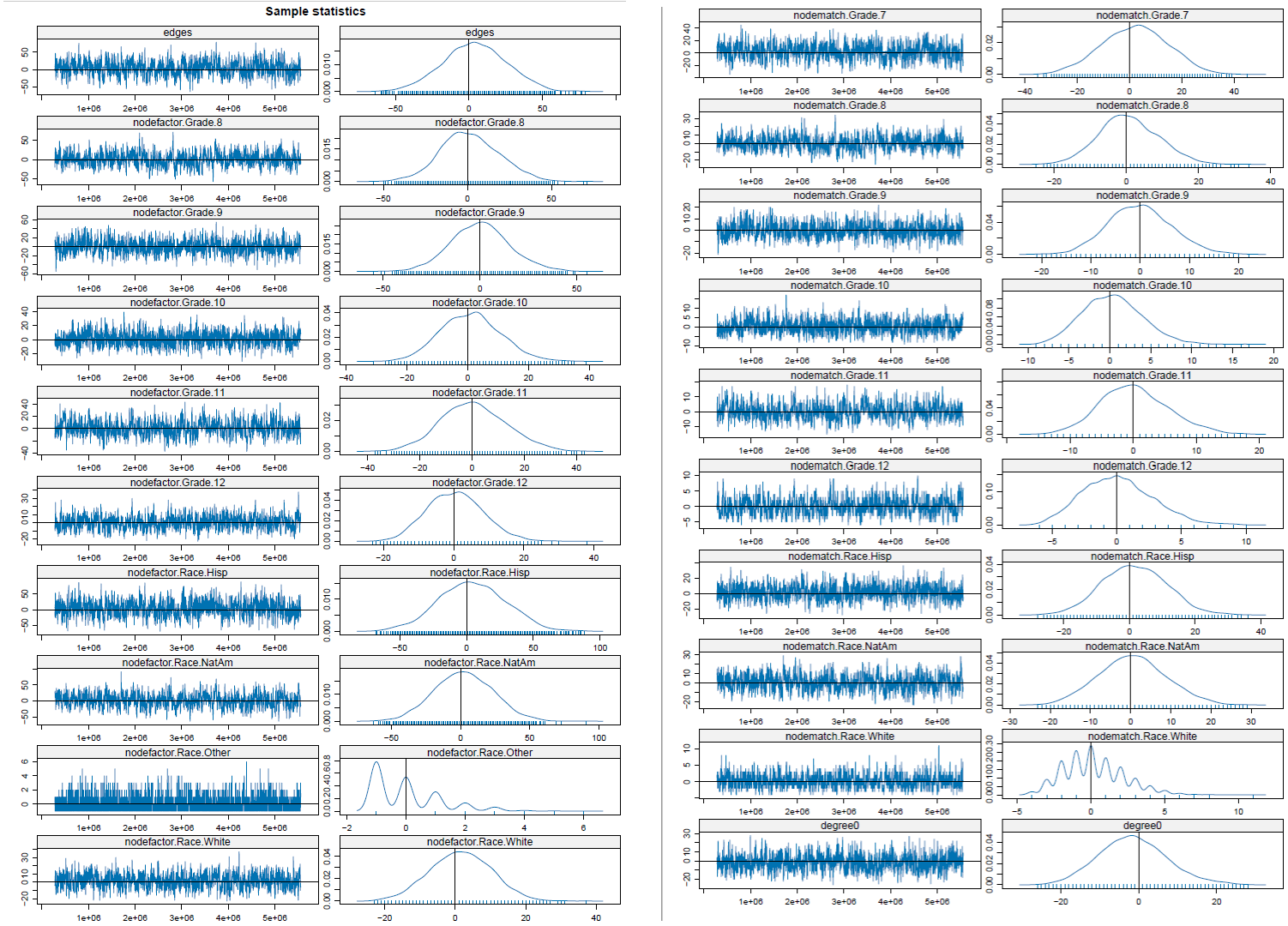
Example of good MCMC Diagnostics



The **traceplots** on the left here display a good random walk pattern around the target value (a fuzzy caterpillar)

The distribution of sampled statistics on the right is roughly bell-shaped and centered here on the target values (does not have to be exact)

Note: Dx are plotted for each term (!)



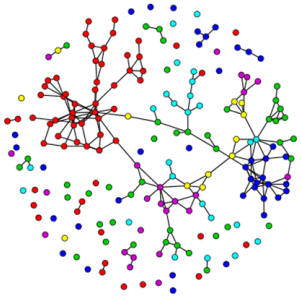
These are all the plots from model 5 in Module 2: ERGMs

What to do if this doesn't look good

- That depends on what you see
  - Major problems are things like bimodal distributions and traceplots or large deviations from the target values
  - Minor problems are things like moderate autocorrelation in the traceplots
- For major problems – think about changing model specification
- For minor problems – think about using one of the many (many) MCMC control parameters in ergm
- This is a deep subject
  - With lots of helpful information out on the web

Goodness of Fit (GOF)

- Traditional GOF stats can be used
  - AIC, BIC are included in the model summary
- We also take another approach
  - Does the model reproduce other network properties that were not included as model terms? Kind of like “out of sample” prediction.
  - We use the full distributions of 3 “higher order” statistics:
    - Degree
    - Shared partners (local clustering)
    - Geodesic distances (global clustering)



DATA

MODEL

ESTIMATED COEFFICIENTS

SIMULATED DATA
(draws from the prob. dist.)

UNMODELED GRAPH STATISTICS FROM DATA

UNMODELED GRAPH STATISTICS FROM SIMULATED DATA

GOODNESS OF FIT OF MODEL TO DATA

Note: Using MCMC here again

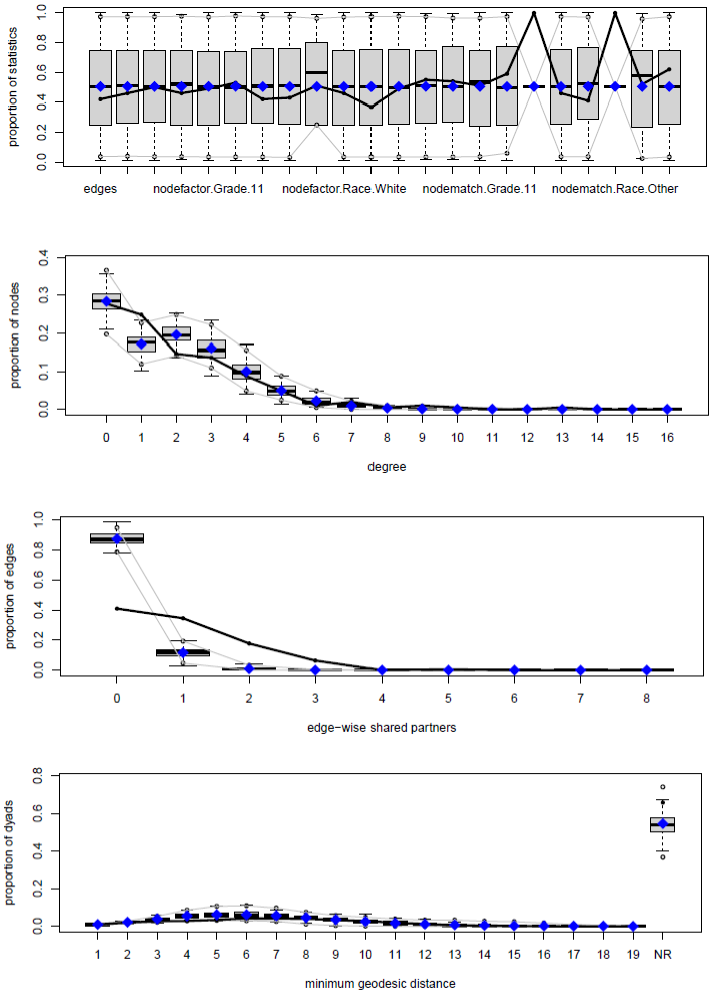
- In this case we're using it for network simulation

- We simulate 100 networks from the fitted model
  - This is a sample from the probability distribution defined by the fitted model
  - On average, the MODELED network statistics from these simulated networks should match the observed targets
    - If not, there's a problem
  - But here we can also see what this model produces for the UNMODELED statistics

Default GOF stats used in `ergm` package

- Degree distribution
  - Node based measure
  - Counts the number of ties on each node
  - Very basic property of a network, important to get right
- Edgewise Shared Partner (ESP) distribution
  - Tie based measure
  - Counts the shared partners for each edge in the network
  - A measure of triadic closure, important for local clustering
- Geodesic distribution
  - Dyad based measure
  - Counts the shortest path between all dyads in the network
  - A measure of global connectivity

GOF plots in `ergm` (the defaults)



- The top plot is the model statistics — *Calibration assessment*
- The bottom 3 plots are the unmodeled statistics — *Validation assessment*
  - Degree
  - Shared partners
  - Geodesics

Also taken from Model 5 in Module 2: ERGMs