Stats218_HW3

Peter Racioppo

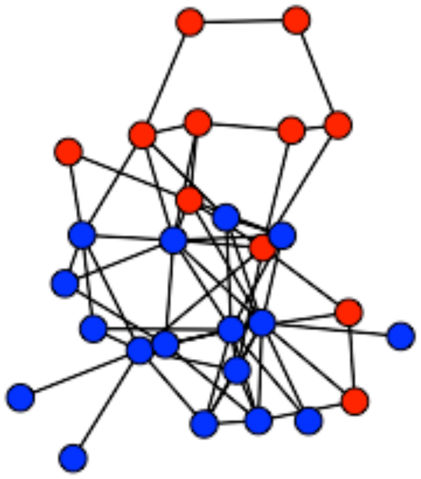
11/28/2020

1) Modeling the French Financial Elite: Here we consider a network collected by Charles Kadushin and described in the Kadushin (1990). He collected data from 127 members of the French financial elite. He used various criteria to determine the top 28 and recorded their who-to-whom responses to questions about who was influential, who were members of the elite and who were friends. He also recorded a large amount of information on their individual backgrounds and characteristics. We will focus on the (undirected) friendship network. There are many vertex covariates, including: • prestige: (coded as 0 if respondent has neither a particule nor a social register listing; 1 if a respondent has either a particule or a social register listing; and 2 if respondent has both a particule and social register listing) • party: An indicator of the party membership. There are 11 parties. • masons: A member of the masons? 1=no; 2=yes. • ena: Graduated from ENA? 1=no; 2=yes. • boards: Number of top boards they are a member of. Many more are described in the paper. The data are in the networkdata package

```
library(networkdata)
library(ergm)

##
## ergm: version 3.11.0, created on 2020-10-14
## Copyright (c) 2020, Mark S. Handcock, University of California -- Los Ange
les
##                     David R. Hunter, Penn State University
##                     Carter T. Butts, University of California -- Irvine
##                     Steven M. Goodreau, University of Washington
##                     Pavel N. Krivitsky, UNSW Sydney
##                     Martina Morris, University of Washington
##                     with contributions from
##                     Li Wang
##                     Kirk Li, University of Washington
##                     Skye Bender-deMoll, University of Washington
##                     Chad Klumb
##                     Michał Bojanowski, Kozminski University
##                     Ben Bolker
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").

## NOTE: Versions before 3.6.1 had a bug in the implementation of the bd()
## constraint which distorted the sampled distribution somewhat. In
## addition, Sampson's Monks datasets had mislabeled vertices. See the
## NEWS and the documentation for more details.
## NOTE: Some common term arguments pertaining to vertex attribute and
## level selection have changed in 3.10.0. See terms help for more
## details. Use 'options(ergm.term=list(version="3.9.4"))' to use old
## behavior.

data(ffef)
help(ffef)
```

a) Plot the network with the vertex color being the ENA attendance. What do you see?

```
# Record ena values in array ena
v = ffef$val
ena = 1:length(v)*0
for(i in 1:length(v)){
        x=v[[i]]
        if(x$ena == 1){
                ena[i] = "blue"
        }
        else if(x$ena == 2){
                ena[i] = "red"
        }
        else {
                ena[i] = "black"}
}

plot.network(ffef, vertex.cex = 3,vertex.col = ena)
```



In the above plot, blue represents not graduating from ENA and red represents graduating. Clustering based on ENA value can be easily observed in the plot.

b) Fit a model to the network that includes terms for the homophily on ENA attendance, prestige and party affiliation. Include terms for geometrically weighted edgewise shared partners with the scale parameter fixed at 0.5 (i.e., gwesp(0.5,fixed=T)). Include a similar term for geometrically weighted dyadwise shared partners with the scale parameter fixed at 0.5 (i.e., gwdsp(0.5,fixed=T)).

```
fit <- ergm(ffef ~ edges + nodematch("ena") + nodematch("prestige") + nodemat
ch("party") + gwesp(0.5,fixed=T) + gwdsp(0.5,fixed=T))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:
```

```
## Optimizing with step length 1.

## The log-likelihood improved by 0.3751.

## Step length converged once. Increasing MCMC sample size.

## Iteration 2 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.0536.

## Step length converged twice. Stopping.

## Finished MCMLE.

## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.

summary(fit)

## Call:
## ergm(formula = ffef ~ edges + nodematch("ena") + nodematch("prestige") +
##     nodematch("party") + gwesp(0.5, fixed = T) + gwdsp(0.5, fixed = T))
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##                      Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                -4.71272    0.44705      0 -10.542  < 1e-04 ***
## nodematch.ena         1.59646    0.37614      0   4.244  < 1e-04 ***
## nodematch.prestige    0.68878    0.30159      0   2.284  0.02238 *
## nodematch.party       1.21577    0.42881      0   2.835  0.00458 **
## gwesp.fixed.0.5       0.43465    0.20141      0   2.158  0.03092 *
## gwdsp.fixed.0.5       0.19331    0.04341      0   4.453  < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 524.0  on 378  degrees of freedom
##  Residual Deviance: 296.6  on 372  degrees of freedom
##
## AIC: 308.6    BIC: 332.2    (Smaller is better.)
```

c) Give an interpretation for each of the coefficients in the model in terms of what it means and also what its magnitude indicates about the nature of social relations in the network.

The “edges” term (which indicates the total number of edges in the graph) has a large magnitude, indicating that it is a strong predictor of graph characteristics. It is negative, which indicates that graphs with fewer edges are more likely. The coefficients for “ena,” “prestige,” and “party,” all are positive, indicating that there is homophily (that is, for all three characteristics. The effect is strongest for “ena,” then “party” and lastly “prestige.”

Given two nodes A and B, connected by an edge, the edge-wise shared partners are the nodes which are connected by a single edge to either A or B. The dyad-wise shared partners are the same thing, except A and B do not have to be connected by an edge. The geometric ewsp and dwsp measures weight larger neighborhoods geometrically less. These terms are measures of clustering or transitivity.

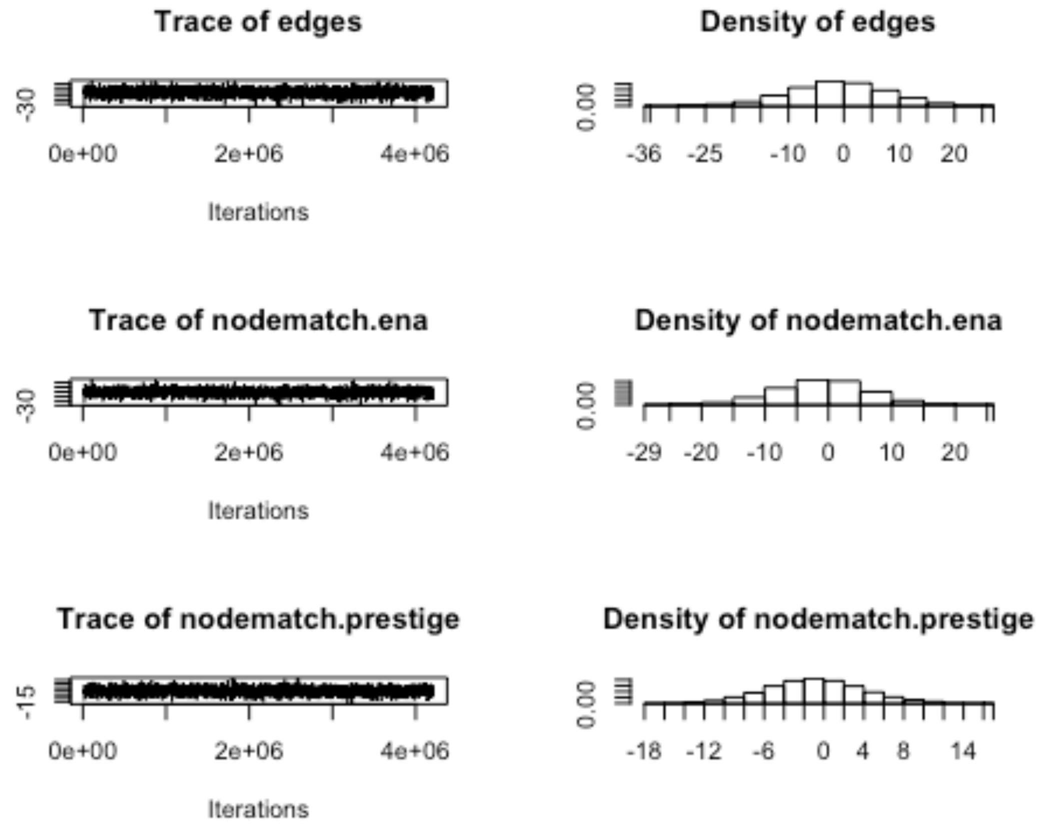
The MLE fit gives positive coefficients for both measures, but a higher value for the edge-wise fixed partners than the dyad-wise fixed partners.

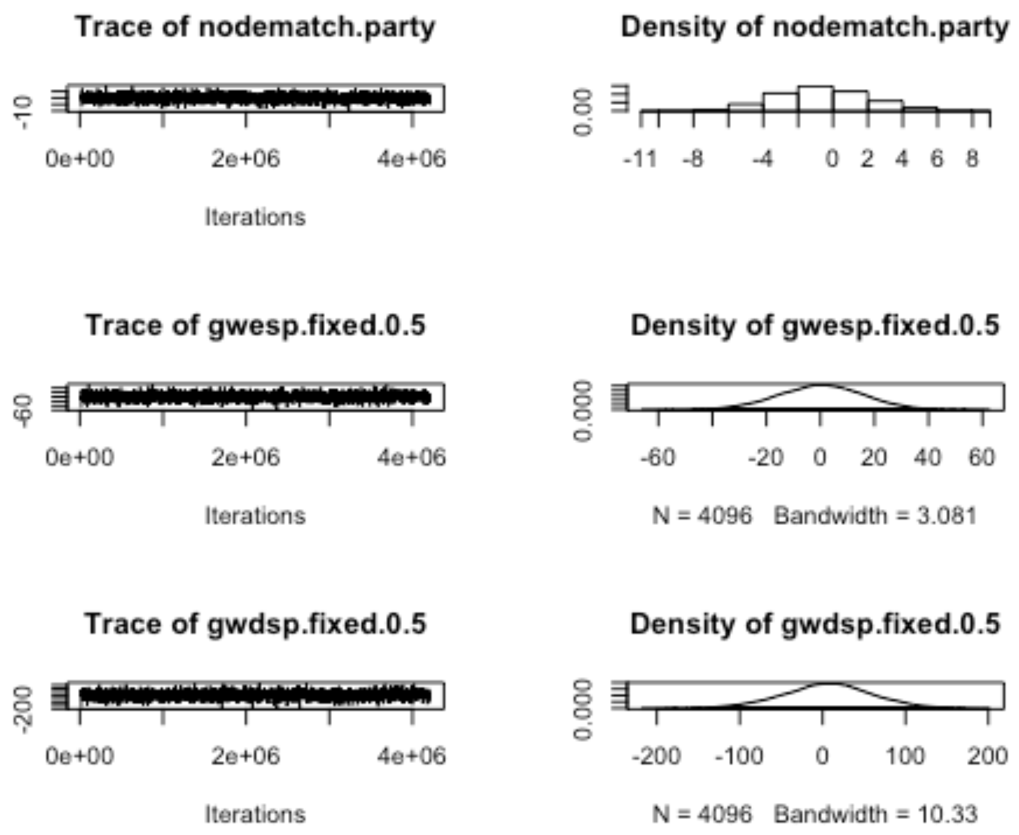
The edge term, the ena homophily, and the gwdsp terms are the most statistically significant, but all terms are least 0.05 statistically significant.

d) Look at the MCMC diagnostics for the model (via, e.g., mcmc.diagnostics(fit)). What does it say about the convergence of your model?

```
mcmc.diagnostics(fit)

## Sample statistics summary:
##
## Iterations = 16384:4209664
## Thinning interval = 1024
## Number of chains = 1
## Sample size per chain = 4096
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean     SD Naive SE Time-series SE
## edges               -0.6699  8.850  0.13829        0.15168
## nodematch.ena       -0.6533  7.029  0.10983        0.12106
## nodematch.prestige -0.3701  4.792  0.07487        0.08678
## nodematch.party     -0.1394  2.646  0.04134        0.04683
## gwesp.fixed.0.5      0.1654 15.341  0.23970        0.26131
## gwdsp.fixed.0.5      5.6257 52.895  0.82649        1.10659
##
## 2. Quantiles for each variable:
##
##                        2.5%    25%     50%   75%  97.5%
## edges                -19.00  -6.00 -1.0000  5.00  16.00
## nodematch.ena        -15.00  -5.00 -1.0000  4.00  13.00
## nodematch.prestige   -10.00  -4.00  0.0000  3.00   9.00
## nodematch.party       -5.00  -2.00  0.0000  2.00   5.00
## gwesp.fixed.0.5      -29.96 -10.23  0.3671 10.56  29.94
## gwdsp.fixed.0.5    -100.42 -28.73  6.8369 40.21 107.09
##
##
## Sample statistics cross-correlations:
##                          edges nodematch.ena nodematch.prestige nodematch.pa
rty
## edges              1.0000000     0.9047113          0.7130987       0.4587
150
## nodematch.ena      0.9047113     1.0000000          0.6293393       0.4035
657
## nodematch.prestige 0.7130987     0.6293393          1.0000000       0.2914
729
## nodematch.party    0.4587150     0.4035657          0.2914729       1.0000
000
## gwesp.fixed.0.5    0.9243554     0.8501274          0.6408701       0.4052
647
## gwdsp.fixed.0.5    0.8334131     0.6711268          0.5536546       0.3420
616
##                    gwesp.fixed.0.5 gwdsp.fixed.0.5
## edges                    0.9243554       0.8334131
## nodematch.ena            0.8501274       0.6711268
## nodematch.prestige       0.6408701       0.5536546
## nodematch.party          0.4052647       0.3420616
## gwesp.fixed.0.5          1.0000000       0.8671474
## gwdsp.fixed.0.5          0.8671474       1.0000000
##
## Sample statistics auto-correlation:
## Chain 1
##                  edges nodematch.ena nodematch.prestige nodematch.party
## Lag 0     1.000000000   1.000000000        1.000000000     1.000000000
## Lag 1024  0.092044654   0.096944400        0.111435237     0.123853511
## Lag 2048  0.016602961   0.030493913        0.047392442     0.034958632
## Lag 3072  0.013515281   0.007908609        0.022540978     0.004086096
## Lag 4096  0.003298574  -0.003921301        0.001373326     0.030707028
## Lag 5120 -0.018552876  -0.004872526       -0.019048668     0.022166111
##          gwesp.fixed.0.5 gwdsp.fixed.0.5
## Lag 0        1.000000000     1.000000000
## Lag 1024     0.086000046     0.193440215
## Lag 2048     0.004563534     0.089969218
## Lag 3072     0.009326685     0.066271117
## Lag 4096    -0.010586366     0.038602007
## Lag 5120    -0.024891819    -0.006914318
##
## Sample statistics burn-in diagnostic (Geweke):
## Chain 1
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##               edges      nodematch.ena nodematch.prestige    nodematch.part
```

```
y
##             1.3651            1.3223            1.6627            -0.239
9
##    gwesp.fixed.0.5    gwdsp.fixed.0.5
##             0.9010            0.4512
##
## Individual P-values (lower = worse):
##              edges      nodematch.ena nodematch.prestige    nodematch.part
y
##         0.17222585         0.18607699         0.09637015         0.8103703
2
##    gwesp.fixed.0.5    gwdsp.fixed.0.5
##         0.36761330         0.65184892
## Joint P-value (lower = worse):  0.634297 .
```


Trace of edges
-30
0e+00
2e+06
4e+06
Iterations
Density of edges
0.00
-36
-25
-10
0
10
20
Trace of nodematch.ena
-30
0e+00
2e+06
4e+06
Iterations
Density of nodematch.ena
0.00
-29
-20
-10
0
10
20
Trace of nodematch.prestige
-15
0e+00
2e+06
4e+06
Iterations
Density of nodematch.prestige
0.00
-18
-12
-6
0
4
8
14



```
##
## MCMC diagnostics shown here are from the last round of simulation, prior t
o computation of final parameter estimates. Because the final estimates are r
efinements of those used for this simulation run, these diagnostics may under
state model performance. To directly assess the performance of the final mode
l on in-model statistics, please use the GOF command: gof(ergmFitObject, GOF=
~model).
```

The MCMC error percentages are 0 for all six covariates. The histograms of all six covariates appear to be approaching relatively smooth, Gaussian-like curves, which suggests that the model is converging and has well-structured statistics.

e) Extend the model to include other covariates in the network and other terms that you think are interesting in explaining the social structure. Feel free to consult the reference paper for ideas. Overall, what are the important features of the social structure of this network?

I began by including most of the other covariates. Only ena, party, prestige, birthdate, and normal-sch have statistically significant effects and only ena was highly significant.

```
fit <- ergm(ffef ~ edges + nodematch("birthdate") + nodematch("birthplace") +
nodematch("sciencepoly") + nodematch("polytech") + nodematch("university") +
nodematch("normal-sch") + nodematch("ena") + nodematch("inspec-gen") + nodema
tch("cabinet") + nodematch("finance-min") + nodematch("party") + nodematch("r
eligion") + nodematch("masons") + nodematch("zipcode") + nodematch("socialreg
") + nodematch("elitevote") + nodematch("eliteprom") + nodematch("prestige")
+ nodematch("clubs") + nodematch("topboards"))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Stopping at the initial estimate.

## Evaluating log-likelihood at the estimate.

summary(fit)

## Call:
## ergm(formula = ffef ~ edges + nodematch("birthdate") + nodematch("birthpla
ce") +
##     nodematch("sciencepoly") + nodematch("polytech") + nodematch("universi
ty") +
##     nodematch("normal-sch") + nodematch("ena") + nodematch("inspec-gen") +
##     nodematch("cabinet") + nodematch("finance-min") + nodematch("party") +
##     nodematch("religion") + nodematch("masons") + nodematch("zipcode") +
##     nodematch("socialreg") + nodematch("elitevote") + nodematch("eliteprom
") +
##     nodematch("prestige") + nodematch("clubs") + nodematch("topboards"))
##
## Iterations:  6 out of 20
##
## Monte Carlo MLE Results:
##                         Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                   -5.12905    1.19071      0  -4.308   <1e-04 ***
## nodematch.birthdate      1.61413    0.73836      0   2.186   0.0288 *
## nodematch.birthplace     0.39880    0.30660      0   1.301   0.1934
## nodematch.sciencepoly   -0.09246    0.35463      0  -0.261   0.7943
## nodematch.polytech      -0.19614    0.34805      0  -0.564   0.5731
## nodematch.university    -0.50516    0.31896      0  -1.584   0.1132
## nodematch.normal-sch     2.07305    1.10797      0   1.871   0.0613 .
## nodematch.ena            1.65452    0.40579      0   4.077   <1e-04 ***
## nodematch.inspec-gen     0.30113    0.33437      0   0.901   0.3678
## nodematch.cabinet        0.18730    0.32845      0   0.570   0.5685
## nodematch.finance-min    0.22340    0.36320      0   0.615   0.5385
## nodematch.party          1.02812    0.47249      0   2.176   0.0296 *
## nodematch.religion       0.12250    0.31988      0   0.383   0.7018
## nodematch.masons         0.07844    0.32522      0   0.241   0.8094
## nodematch.zipcode     -0.69016    0.54509      0  -1.266   0.2055
## nodematch.socialreg   -0.20159    0.38500      0  -0.524   0.6006
## nodematch.elitevote   -0.68622    1.26496      0  -0.542   0.5875
## nodematch.eliteprom   -0.06802    0.34002      0  -0.200   0.8415
## nodematch.prestige     0.79349    0.39338      0   2.017   0.0437 *
## nodematch.clubs        0.26470    0.34109      0   0.776   0.4377
## nodematch.topboards    0.04544    0.47785      0   0.095   0.9242
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 524.0  on 378  degrees of freedom
##  Residual Deviance: 290.2  on 357  degrees of freedom
##
## AIC: 332.2    BIC: 414.8    (Smaller is better.)
```

I kept only birthdate, ena, party, and prestige. I tried fitting a model for differential homophily on the covariates, but this did't appear to have a big effect. I also tried nodefactor, nodecov, and absdiff, which were also of no use. I also tried nodemix("ena"), which gave no significant improvement.

```
fit <- ergm(ffef ~ edges + nodematch("birthdate") + nodematch("ena") + nodema
tch("party") + nodematch("prestige") + gwesp(0.5,fixed=T) + gwdsp(0.5,fixed=T
))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.3947.

## Step length converged once. Increasing MCMC sample size.

## Iteration 2 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.02978.

## Step length converged twice. Stopping.

## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```
summary(fit)
```

```
## Call:
## ergm(formula = ffef ~ edges + nodematch("birthdate") + nodematch("ena") +
##     nodematch("party") + nodematch("prestige") + gwesp(0.5, fixed = T) +
##     gwdsp(0.5, fixed = T))
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##                      Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                -4.75665    0.50655      0  -9.390  < 1e-04 ***
## nodematch.birthdate   1.53255    0.69075      0   2.219  0.02651 *
## nodematch.ena         1.59812    0.37907      0   4.216  < 1e-04 ***
## nodematch.party       1.23754    0.43109      0   2.871  0.00410 **
## nodematch.prestige    0.67562    0.30489      0   2.216  0.02670 *
## gwesp.fixed.0.5       0.43858    0.19652      0   2.232  0.02564 *
## gwdsp.fixed.0.5       0.18781    0.05849      0   3.211  0.00132 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 524.0  on 378  degrees of freedom
##  Residual Deviance: 291.9  on 371  degrees of freedom
##
## AIC: 305.9    BIC: 333.5    (Smaller is better.)
```

It appears that the important features of this social network are degree, ena, birthdate, party, prestige, and transitivity (as measured by gwesp and gwdsp).

2) Balance in signed networks: Consider the alliance structure among three tribal groups of the Eastern Central Highlands of New Guinea. considered by Read (1954). Among these groups (the Gahuku-Gama) the enemy of an enemy can be either a friend or an enemy. The data is in the networkdata package as a combination of the rona and hina networks. The rova binary network has 1 for alliance ("rova") relations and 0 otherwize. The hina binary network has 1 for antagonistic ("hina") relations. The default (non)edge value of 0 means neither of these relations hold.

a) First we construct a valued network to measure the degree of alliance on a count scale. This will have 2 for alliance ("rova") relations, 0 for antagonistic ("hina") relations, and 1 for neither of these. Store these in the edge attribute count. This representation reduces the signed network onto an ordinal alliance scale. Here is some code to do it:

```
# library(ergd)
library(ergm.count)
```

```
##
## ergm.count: version 3.4.0, created on 2019-05-15
## Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
##                      with contributions from
##                      Mark S. Handcock, University of California -- Los Ange
les
##                      David R. Hunter, Penn State University
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm.count").
```

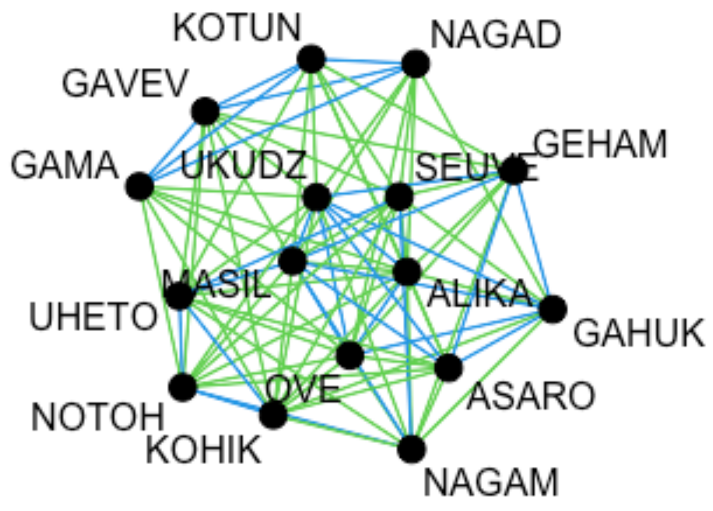
```
## NOTE: The form of the term 'CMP' has been changed in version 3.2 of
## 'ergm.count'. See the news or help('CMP') for more information.
```

```
data(rova)
data(hina)
gama <- rova[,]+(1-hina[,])
diag(gama) <- 0
gama <- as.network(gama, directed = FALSE, matrix.type = "a",
ignore.eval = FALSE, names.eval = "count")
```

In this analysis we will treat the value as a binomial count with two trials and fit ERGM via the ergm.count package.

b) Plot the network, labeling the nodes and using the edge.col parameter. Briefly describe the structure you see.

```
# Record ena values in array s
v = gama$val
names = 1:length(v)*0
for(i in 1:length(v)){
        x=v[[i]]
        names[i] = x[2]
}

# Record the edge values in array s
s = rova[,]+(1-hina[,])
diag(s) <- 0
s2 = s+2

plot.network(gama, vertex.cex = 3,vertex.col = "black",edge.col = s2,label=na
mes)
```



```
gama_dir <- as.network(s, directed = TRUE, matrix.type = "a", ignore.eval = F
ALSE, names.eval = "count")
```

There appears to be clustering based on edge value. Tribes with the most number of alliances seem to be more central, while tribes with many antagonisms are more outlying.

c) Fit a model with the nonzero and sum terms. Interpret what the terms in the model mean and what their estimates mean for the relations network. Hint: See help("ergm-terms").

```
# fit <- ergm(gama_dir ~ edges + absdiff(s,"sum"))
# fit <- ergm(gama_dir ~ edges + edgecov(s, "nonzero") + edgecov(s, "sum"))
# fit <- ergm.count(gama ~ edges + CMP)

# fit <- ergm.count(gama ~ edges + dyadcov(s, "nonzero") + dyadcov(s, "sum"))
fit <- ergm(gama ~ edges + edgecov(s, "nonzero") + edgecov(s, "sum"))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.
## Warning in ergm.mple(nw, fd, m, MPLEtype = MPLEtype, init = init, control
=
## control, : glm.fit: algorithm did not convergeglm.fit: fitted probabilitie
s
## numerically 0 or 1 occurred

## Finished MPLE.

## Warning: Model statistics 'edgecov.sum' are linear combinations of some se
t of
## preceding statistics at the current stage of the estimation. This may indi
cate
## that the model is nonidentifiable.

## Stopping at the initial estimate.

## Evaluating log-likelihood at the estimate.
```

```
summary(fit)
```

```
## Call:
## ergm(formula = gama ~ edges + edgecov(s, "nonzero") + edgecov(s,
##     "sum"))
##
## Iterations:  25 out of 20
##
## Monte Carlo MLE Results:
##                   Estimate Std. Error MCMC % z value Pr(>|z|)
## edges               -27.13   87795.81      0   0.000        1
## edgecov.nonzero      53.30   95310.14      0   0.001        1
## edgecov.sum             NA       0.00      0      NA       NA
##
##      Null Deviance: 1.664e+02  on 120  degrees of freedom
##  Residual Deviance: 6.301e-10  on 117  degrees of freedom
##
## AIC: 6    BIC: 14.36    (Smaller is better.)
```

The nonzero edge covariate is higher-valued than the edges statistic.

d) Add to this model the term transitiveties(threshold=1.5). Give an interpretation of this term and what the estimates of each of the terms mean for the relations network.

```
# fit <- ergm.count(gama_dir ~ edges + transitiveties(threshold=1.5))
# summary(fit)
# This returns the following error:
# <<Error in ergm.count(gama_dir ~ edges + transitiveties(threshold = 1.5)) :
# could not find function "ergm.count">>

# fit <- ergm(gama_dir ~ edges + transitiveties(threshold=1.5))
# summary(fit)
# This returns the following error:
# <<Error: In term ‘transitiveties’ in package ‘ergm’: Model term does not re
cognize ‘threshold’ argument.>>

# Instead, I attempted to use the related transitive, ttriple, and ctriple st
atistics:
# fit <- ergm(gama_dir ~ edges + transitive)
fit <- ergm(gama_dir ~ edges + ttriple)

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.004122.

## Step length converged once. Increasing MCMC sample size.

## Iteration 2 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.0006225.

## Step length converged twice. Stopping.

## Finished MCMLE.

## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.

# fit <- ergm(gama_dir ~ edges + ttriple + ctriple)
summary(fit)

## Call:
## ergm(formula = gama_dir ~ edges + ttriple)
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##         Estimate Std. Error MCMC % z value Pr(>|z|)
## edges   0.008556   0.898243      0   0.010    0.992
## ttriple 0.047281   0.037719      0   1.254    0.210
```

```
##
##      Null Deviance: 332.7  on 240  degrees of freedom
##  Residual Deviance: 264.2  on 238  degrees of freedom
##
## AIC: 268.2    BIC: 275.2    (Smaller is better.)
```

The “transitiveties” term adds a statistic on the number of paths of length 2 in the network. On the other hand, the “ttriples” term adds a statistic on the number of transitive triples in the network, and “ctriples” counts the cyclic tripels. The “transitive” term adds a statistic on the number of triads in the network that are transitive. None of the fits using these terms was statistically significant.

e) See if you can extend the model by adding further terms. Briefly report if you can find a model that fits better than the models in parts a) and b).

```
fit <- ergm(gama_dir ~ edges + ttriple + gwesp)

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Warning: Model statistics 'gwesp' are linear combinations of some set of
## preceding statistics at the current stage of the estimation. This may indi
cate
## that the model is nonidentifiable.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:

## Warning: Model statistics 'gwesp.decay' are linear combinations of some se
t of
## preceding statistics at the current stage of the estimation. This may indi
cate
## that the model is nonidentifiable.

## Optimizing with step length 0.952104629204805.

## The log-likelihood improved by 0.8825.

## Iteration 2 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.7527.

## Step length converged once. Increasing MCMC sample size.
## Iteration 3 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by 0.3123.

## Step length converged twice. Stopping.

## Finished MCMLE.

## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.

summary(fit)

## Call:
## ergm(formula = gama_dir ~ edges + ttriple + gwesp)
##
## Iterations:  3 out of 20
##
## Monte Carlo MLE Results:
##              Estimate Std. Error MCMC % z value Pr(>|z|)
## edges       -11.62786   20.34376      0  -0.572    0.568
## ttriple       0.07329    0.06858      0   1.069    0.285
## gwesp         4.77017   12.52290      0   0.381    0.703
## gwesp.decay   0.74453    0.67677      0   1.100    0.271
##
##      Null Deviance: 332.7  on 240  degrees of freedom
##  Residual Deviance: 392.5  on 236  degrees of freedom
##
## AIC: 400.5    BIC: 414.4    (Smaller is better.)
```

Adding a gwesp term greatly improved convergence. The model now converges, and the ttriple term is now statistically significant.

f) Finally, we have represented the signed network on an ordinal alliance scale. Comment on what is lost by this, if anything, compared to the signed network.

No information is lost by converting the signed network to an ordinal network, because each of the possible relations in the signed network can be represented in the ordinal network. However, since these relationships are represented on a count scale, “no alliance” is represented as taking an intermediate value (1) between “alliance” (2) and “antagonism” (0), which is different than in the signed graph.

3) Modeling a Triad Census in Friendship Relations: Here we consider again the network introduced in Homework 2 of strong friendship ties among 13 boys and 14 girls in a

sixth- grade classroom, as collected by Hansell (1984). Each student was asked if they liked each other student “a lot”, “some”, or “not much”. Here we consider a strong friendship tie to exist if a student likes another student “a lot.” Also recorded is the sex of each student. The data is in the networkdata package

```
detach("package:ergm.count", unload=TRUE)
library(networkdata)
library(ergm.tapered)

## Loading required package: ergd

## Registered S3 methods overwritten by 'ergd':
##   method                                    from
##   -.ergm_levels_spec_function               ergm
##   anova.ergm                                ergm
##   anova.ergmlist                            ergm
##   as.edgelist.pending_update_network        ergm
##   as.edgelist.rlebdm                        ergm
##   as.matrix.pending_update_network          ergm
##   as.matrix.rlebdm                          ergm
##   as.network.numeric                        ergm
##   as.network.pending_update_network         ergm
##   c.ergm_model                              ergm
##   coef.ergm                                 ergm
##   coefficients.ergm                         ergm
##   dim.rlebdm                                ergm
##   logLik.ergm                               ergm
##   network.dyadcount.pending_update_network  ergm
##   network.edgecount.pending_update_network  ergm
##   network.naedgecount.pending_update_network ergm
##   network.size.pending_update_network       ergm
##   nobs.ergm                                 ergm
##   plot.gof                                  ergm
##   print.ergm                                ergm
##   print.gof                                 ergm
##   print.network.list                        ergm
##   print.rlebdm                              ergm
##   print.summary.ergm                        ergm
##   simulate.ergm                             ergm
##   simulate.ergm_model                       ergm
##   simulate.formula                          ergm
##   summary.ergm                              ergm
##   summary.ergm_model                        ergm
##   summary.formula                           ergm
##   summary.network.list                      ergm
##   update.network                            ergm
##   vcov.ergm                                 ergm

##
## ergd: version 4.1-5197, created on 2019-11-21
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Ange
```

```
les
##                        David R. Hunter, Penn State University
##                        Carter T. Butts, University of California -- Irvine
##                        Steven M. Goodreau, University of Washington
##                        Pavel N. Krivitsky, University of Wollongong
##                        Martina Morris, University of Washington
##                        with contributions from
##                        Li Wang
##                        Kirk Li, University of Washington
##                        Skye Bender-deMoll, University of Washington
##                        Chad Klumb
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergd").

## NOTE: Versions before 3.6.1 had a bug in the implementation of the bd()
## constraint which distorted the sampled distribution somewhat. In
## addition, Sampson's Monks datasets had mislabeled vertices. See the
## NEWS and the documentation for more details.

## NOTE: Some common term arguments pertaining to vertex attribute and
## level selection have changed in 3.10.0. See terms help for more
## details. Use 'options(ergm.term=list(version="3.9.4"))' to use old
## behavior.

##
## Attaching package: 'ergd'

## The following objects are masked from 'package:ergm':
##
##     approx.hotelling.diff.test, as.ergm_model, as.rlebdm,
##     check.ErgmTerm, COLLAPSE_SMALLEST, control.ergm,
##     control.ergm.bridge, control.ergm.godfather, control.gof.ergm,
##     control.gof.formula, control.logLik.ergm, control.san,
##     control.simulate, control.simulate.ergm, control.simulate.formula,
##     control.simulate.formula.ergm, degreedist, enformulate.curved,
##     ergm_attr_levels, ergm_Clist, ergm_get_vattr, ergm_Init_abort,
##     ergm_Init_inform, ergm_Init_warn, ergm_MCMC_sample,
##     ergm_MCMC_slave, ergm_model, ergm_plot.mcmc.list, ergm_proposal,
##     ergm_proposal_table, ergm.allstats, ergm.bridge.0.llk,
##     ergm.bridge.dindstart.llk, ergm.bridge.llr, ergm.Cprepare,
##     ergm.design, ergm.estfun, ergm.eta, ergm.etagrad, ergm.etagradmult,
##     ergm.exact, ergm.geodistdist, ergm.geodistn, ergm.getCluster,
##     ergm.getnetwork, ergm.godfather, ergm.pl, ergm.restartCluster,
##     ergm.stopCluster, ergmMPLE, fix.curved, get.node.attr,
##     geweke.diag.mv, gof, is.curved, is.durational, is.dyad.independent,
##     is.ergm, is.inCH, is.pending_update_network, LARGEST, logLikNull,
##     mcmc.diagnostics, network.list, nparam, nthreads,
##     nvattr.copy.network, offset.info.formula, param_names,
##     pending_update_network, rlebdm, san, search.ergmTerms,
##     simulate_formula, SMALLEST, spectrum0.mvar, summary_formula,
##     to_ergm_Cdouble, wtd.median

## The following object is masked from 'package:sna':
##
##     symmetrize

## Detaching prior version of 'ergm'.

data(hansell)
help(hansell)
```

a) Fit a triad census model using ergm.tapered. Include a term for homophily by sex.

```
fit <- ergm.tapered(hansell ~ edges + nodematch("sex") + triadcensus(1:7))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.6695.

## Estimating equations are not within tolerance region.

## Iteration 2 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.6519.

## Estimating equations are not within tolerance region.

## Iteration 3 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 1.849.

## Estimating equations are not within tolerance region.

## Estimating equations did not move closer to tolerance region more than 1 t
ime(s) in 4 steps; increasing sample size.

## Iteration 4 of at most 15:
```

```
## Optimizing with step length 1.

## The log-likelihood improved by 0.1313.

## Estimating equations are not within tolerance region.

## Iteration 5 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1222.

## Estimating equations are not within tolerance region.

## Iteration 6 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1784.

## Estimating equations are not within tolerance region.

## Iteration 7 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.22.

## Estimating equations are not within tolerance region.

## Estimating equations did not move closer to tolerance region more than 1 t
ime(s) in 4 steps; increasing sample size.

## Iteration 8 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1582.

## Estimating equations are not within tolerance region.

## Iteration 9 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.6128.

## Estimating equations are not within tolerance region.

## Iteration 10 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.06926.
## Convergence test p-value: 0.992649152552287. Not converged with 99% confid
ence; increasing sample size.
## Iteration 11 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.0285.
## Convergence test p-value: 0.834166466242242. Not converged with 99% confid
ence; increasing sample size.
## Iteration 12 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.01669.
## Convergence test p-value: 0.0651496081470339. Not converged with 99% confi
dence; increasing sample size.
## Iteration 13 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.003664.
## Convergence test p-value: 0.156233150148109. Not converged with 99% confid
ence; increasing sample size.
## Iteration 14 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.004097.
## Convergence test p-value: 0.000194523152950733. Converged with 99% confide
nce.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```
summary(fit)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   hansell ~ Taper(~edges + nodematch("sex") + triadcensus(1:7),
##     coef = .taper.coef, m = .taper.center)
## <environment: 0x7fb9737ca320>
##
## Iterations:  14 out of 15
##
## Monte Carlo MLE Results:
##                  Estimate Std. Error MCMC % z value Pr(>|z|)
## edges            -0.84972    0.23541      0  -3.610 0.000307 ***
## nodematch.sex     1.12398    0.24945      0   4.506  < 1e-04 ***
## triadcensus.012  -0.11160    0.03629      0  -3.075 0.002105 **
## triadcensus.102  -0.12146    0.06194      0  -1.961 0.049869 *
## triadcensus.021D -0.02937    0.05210      0  -0.564 0.572986
## triadcensus.021U -0.05069    0.07967      0  -0.636 0.524579
## triadcensus.021C -0.31242    0.07398      0  -4.223  < 1e-04 ***
## triadcensus.111D -0.38910    0.12189      0  -3.192 0.001411 **
## triadcensus.111U -0.26474    0.09147      0  -2.894 0.003799 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 973.2  on 702  degrees of freedom
##  Residual Deviance: 665.2  on 693  degrees of freedom
##
## AIC: 683.2    BIC: 724.2    (Smaller is better.)
```

b) Give a brief interpretation of the coefficients of the terms. Based on this model, does there appear to be a general preference for transitive friendship ties? Is there homophily by sex? Give a brief summary of the overall triad census pattern.

The homophily coefficient is positive, has the largest magnitude of any coefficient, and is highly statistically significant, so we can conclude that there is homophily by sex. The edges statistic is negative and also highly statistically significant. The 021C triad coefficient is highly statistically significant and negative. This suggests that there is a significant preference against 021C triads, which can be thought of as hierarchies of length 2, so the generative model for the network likely has an hierarchical structure. In fact, 5 of the 7 triad statistics are statisistically significant, and they are all negative. In summary, the model suggests there is a preference against transitive friendship terms.

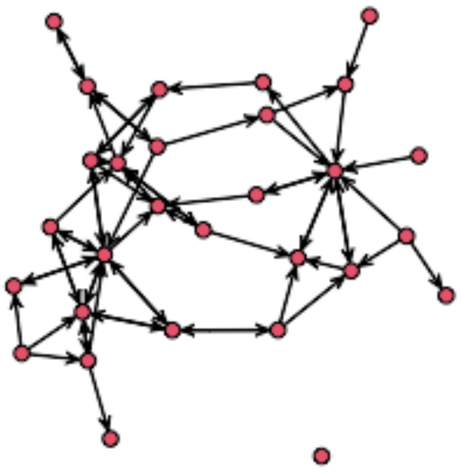
4) Modeling a Triad Census in Friendship Relations: Here we consider the network of friendship ties of girls in grade 9. The data is in the networkdata package.

```r
# detach("package:ergm", unload=TRUE)
library(networkdata)
data(gfriends)
help(gfriends)
```

a) First, extract out the grade nine students using gfriends$X[,1]==9 and create a directed network. Also add a vertex variable equal to the student gpa.

```r
grade9 = (gfriends$X[,1]==9)*1
idx = which(grade9==1)
gpa = gfriends$X[,2]
gpa9 = gpa[idx]
smokes = gfriends$X[,3]
smokes9 = smokes[idx]
adj = gfriends$Y
adj9 = adj[idx,idx]

gf <- network(x = adj9,directed = TRUE,loops = FALSE,matrix.type = "adjacency
")
```

```r
set.vertex.attribute(gf, # the name of the network object
                     "gpa", # the name we want to reference the variable by
                     gpa9 # the value we are giving that variable
                     )
set.vertex.attribute(gf, # the name of the network object
                     "smoke", # the name we want to reference the variable by
                     smokes9 # the value we are giving that variable
                     )

gplot(gf)
```



Fit a triad census model using a subset of the first seven triad types (012, 102, 021D, 021U, 021C, 111D, 111U). Recall that 003 is the reference category. Also include GPA and smoking effects, using something like:

```r
# library(ergd)
library(ergm.tapered)
fit <- ergm.tapered(gf ~ edges + nodecov("gpa") + nodecov("smoke") + triadcen
sus(c(1:7)))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.
## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.3956.

## Estimating equations are not within tolerance region.

## Iteration 2 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.497.

## Estimating equations are not within tolerance region.

## Iteration 3 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.6182.

## Estimating equations are not within tolerance region.

## Iteration 4 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1216.

## Estimating equations are not within tolerance region.

## Iteration 5 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.07933.

## Estimating equations are not within tolerance region.

## Iteration 6 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.4282.

## Estimating equations are not within tolerance region.

## Estimating equations did not move closer to tolerance region more than 1 t
ime(s) in 4 steps; increasing sample size.
```

```
## Iteration 7 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.2512.

## Estimating equations are not within tolerance region.

## Iteration 8 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1818.

## Estimating equations are not within tolerance region.

## Iteration 9 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.304.

## Estimating equations are not within tolerance region.

## Iteration 10 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1282.

## Estimating equations are not within tolerance region.

## Estimating equations did not move closer to tolerance region more than 1 t
ime(s) in 4 steps; increasing sample size.

## Iteration 11 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.09275.

## Estimating equations are not within tolerance region.

## Iteration 12 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.07959.

## Estimating equations are not within tolerance region.

## Iteration 13 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.02245.
## Convergence test p-value: 0.82576999891422. Not converged with 99% confide
nce; increasing sample size.
## Iteration 14 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.01592.
## Convergence test p-value: 0.806997972731138. Not converged with 99% confid
ence; increasing sample size.
## Iteration 15 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.00642.
## Convergence test p-value: 0.705658396460849. Not converged with 99% confid
ence; increasing sample size.
## MCMLE estimation did not converge after 15 iterations. The estimated coeff
icients may not be accurate. Estimation may be resumed by passing the coeffic
ients as initial values; see 'init' under ?control.ergm for details.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```
summary(fit)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   gf ~ Taper(~edges + nodecov("gpa") + nodecov("smoke") + triadce
nsus(c(1:7)),
##     coef = .taper.coef, m = .taper.center)
## <environment: 0x7fb976e3fa78>
##
## Iterations:  15 out of 15
##
## Monte Carlo MLE Results:
##                   Estimate Std. Error MCMC % z value Pr(>|z|)
## edges             -1.381301   1.290477      0  -1.070   0.2844
## nodecov.gpa        0.178780   0.124529      0   1.436   0.1511
## nodecov.smoke     -0.244150   0.294060      0  -0.830   0.4064
## triadcensus.012   -0.018656   0.069490      0  -0.268   0.7883
## triadcensus.102    0.007047   0.131379      0   0.054   0.9572
## triadcensus.021D -0.895634   0.524522      0  -1.708   0.0877 .
## triadcensus.021U  0.116129   0.236120      0   0.492   0.6228
## triadcensus.021C -0.360562   0.280943      0  -1.283   0.1994
## triadcensus.111D -0.156544   0.230441      0  -0.679   0.4969
## triadcensus.111U -0.497264   0.282892      0  -1.758   0.0788 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##      Null Deviance: 1125.7  on 812  degrees of freedom
##  Residual Deviance:  421.1  on 802  degrees of freedom
##
## AIC: 441.1    BIC: 488.1    (Smaller is better.)
```

b) Give a brief interpretation of the coefficients of the terms. Give a brief summary of the overall triad census pattern.

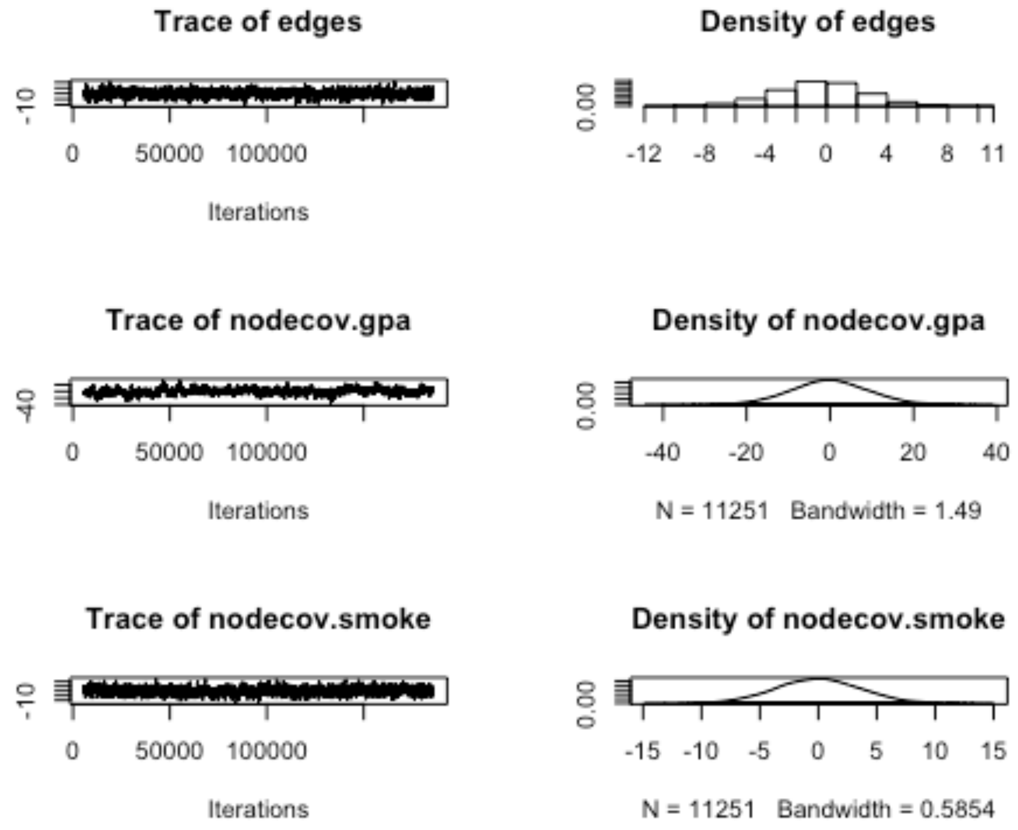
There is a small (though not statistically significant) dependence on the node covariates, with a positive relationship for gpa and negative for smoking. Most of the triad coefficients are negative and not statistically significant.

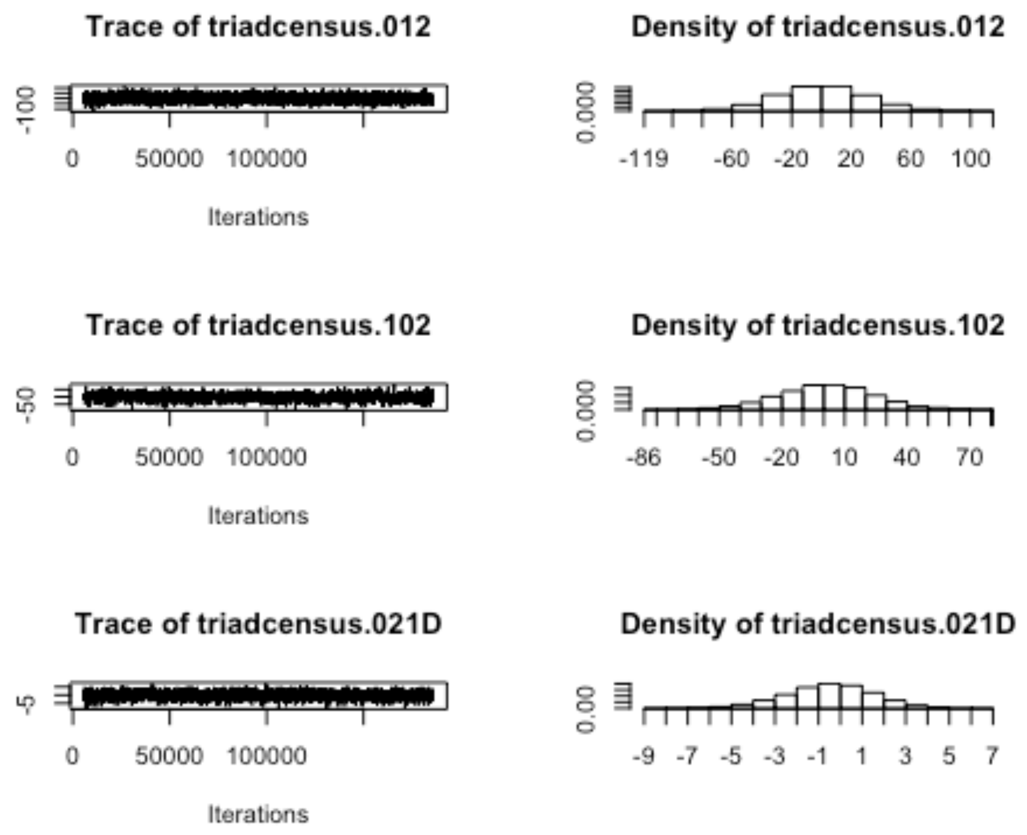
The two triad types which are most significant and have the largest magnitude are 021D and 111U. The 021D triad is a directed network with two edges pointing from a single node into the other two nodes in the triad. A preponderance of triads of this type would indicate that the data is hierarchical. The 111U triad has one multi-directional edge and a directed edge leaving one of the nodes connected to the multi-directional edge. In other words, the 111U triad is like the 021D triad, but with an additional directed edge pointing back along one of the edges. Thus, a network with many 111U triads can be thought of as a hierarchical directed network, but with some edges pointing back up the hierarchy. That is, twice as many edges point down the hierarchy as point up.

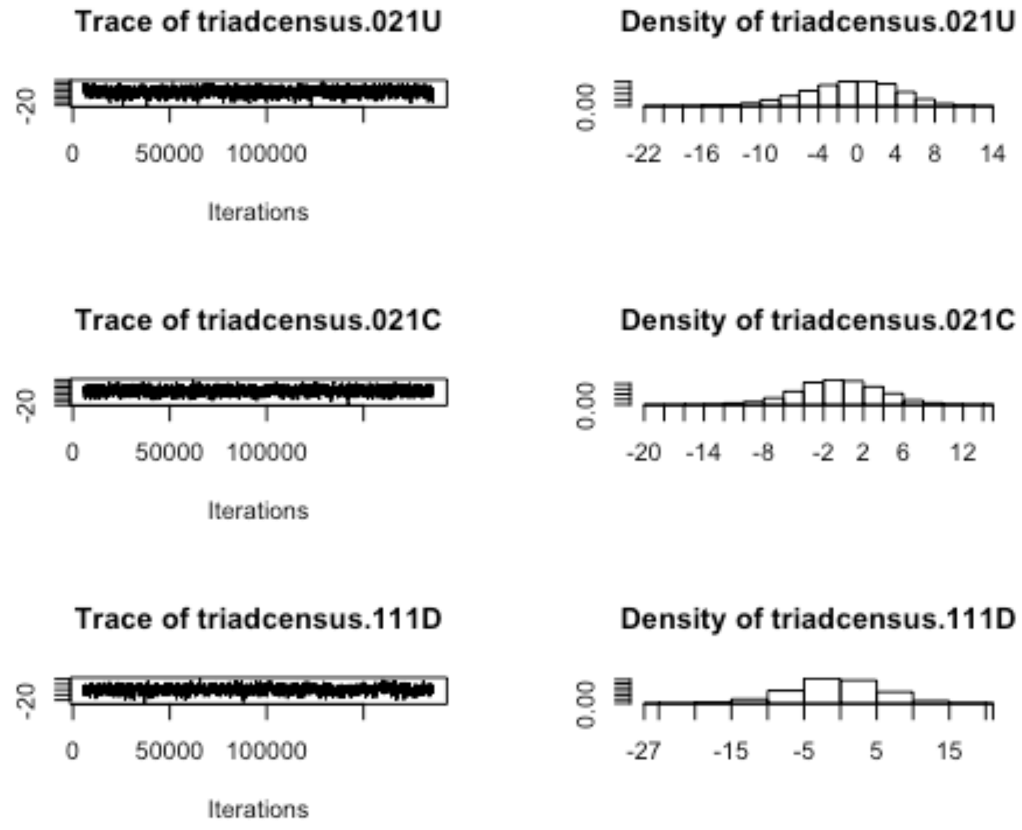
In summary, since the coefficients of both the 021D triad and the 111U are negative, high in magnitude, and statistically significant, we can conclude that the generative model for this network has a preference for non-hierarchical relationships.

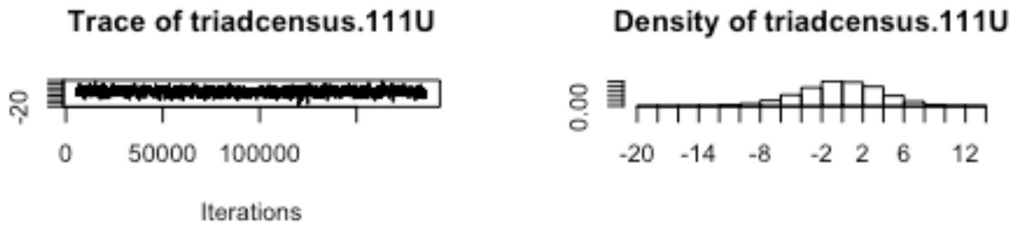
c) Look at the MCMC diagnostics for the model (via, e.g., mcmc.diagnostics(fit)). What does it say about the convergence of your model?

```
mcmc.diagnostics(fit)

## Sample statistics summary:
##
## Iterations = 5824:185824
## Thinning interval = 16
## Number of chains = 1
## Sample size per chain = 11251
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean     SD Naive SE Time-series SE
## edges             0.040085  2.843  0.02680        0.10266
## nodecov.gpa      -0.018211  9.184  0.08658        0.68088
## nodecov.smoke    -0.014150  3.568  0.03363        0.12159
## triadcensus.012   0.513288 29.207  0.27535        0.45458
## triadcensus.102   1.431073 22.589  0.21296        0.71951
## triadcensus.021D -0.036886  2.006  0.01891        0.04091
## triadcensus.021U -0.250022  4.685  0.04417        0.12342
## triadcensus.021C -0.001778  4.021  0.03791        0.08812
## triadcensus.111D -0.074038  5.917  0.05578        0.19800
## triadcensus.111U  0.285219  3.836  0.03617        0.12274
## 
## 2. Quantiles for each variable:
## 
##                    2.5%    25%   50%   75%  97.5%
## edges             -6.00  -2.00  0.00  2.00  5.000
## nodecov.gpa      -17.79  -6.15 -0.11  6.02 18.115
## nodecov.smoke     -6.92  -2.46 -0.03  2.39  6.927
## triadcensus.012  -57.00 -19.00  1.00 20.00 58.000
## triadcensus.102  -43.75 -14.00  2.00 17.00 45.000
## triadcensus.021D  -4.00  -1.00  0.00  1.00  4.000
## triadcensus.021U -10.00  -3.00  0.00  3.00  8.000
## triadcensus.021C  -8.00  -3.00  0.00  3.00  8.000
## triadcensus.111D -12.00  -4.00  0.00  4.00 11.000
## triadcensus.111U  -7.00  -2.00  0.00  3.00  7.000
## 
## 
## Are sample statistics significantly different from observed?
##                 edges nodecov.gpa nodecov.smoke triadcensus.012 triadcensu
s.102
## diff.      0.04008533 -0.01821083   -0.01414985       0.5132877      1.431
07279
## test stat. 0.39047913 -0.02674606   -0.11636962       1.1291408      1.988
94266
## P-val.     0.69618228  0.97866228    0.90735961       0.2588384      0.046
70753
##            triadcensus.021D triadcensus.021U triadcensus.021C triadcensus.
111D
## diff.           -0.03688561      -0.25002222      -0.00177762      -0.0740
3786
## test stat.      -0.90163124      -2.02581559      -0.02017241      -0.3739
3784
## P-val.           0.36725279       0.04278369       0.98390583       0.7084
5056
##            triadcensus.111U Overall (Chi^2)
## diff.            0.28521909              NA
## test stat.       2.32377541    24.851351983
## P-val.           0.02013753     0.006380713
## 
## Sample statistics cross-correlations:
##                         edges nodecov.gpa nodecov.smoke triadcensus.012
## edges              1.00000000  0.40497938   -0.04385906      0.40117492
## nodecov.gpa        0.40497938  1.00000000   -0.15449762      0.07107859
## nodecov.smoke     -0.04385906 -0.15449762    1.00000000      0.04085940
## triadcensus.012    0.40117492  0.07107859    0.04085940      1.00000000
## triadcensus.102    0.71767763  0.25447608    0.04373846     -0.18923054
## triadcensus.021D  0.16399021  0.03174566    0.03946099      0.27974010
## triadcensus.021U  0.13795004  0.07660578   -0.03940063      0.17899299
## triadcensus.021C  0.17789390  0.03298597    0.02334379      0.30774172
## triadcensus.111D  0.41377429  0.20440561   -0.10080895      0.07873964
## triadcensus.111U  0.28577313  0.07795858    0.06451232      0.07892790
##                  triadcensus.102 triadcensus.021D triadcensus.021U
## edges                 0.71767763       0.16399021      0.137950043
## nodecov.gpa           0.25447608       0.03174566      0.076605784
## nodecov.smoke         0.04373846       0.03946099     -0.039400627
## triadcensus.012      -0.18923054       0.27974010      0.178992988
## triadcensus.102       1.00000000      -0.01333222     -0.115181043
## triadcensus.021D     -0.01333222       1.00000000      0.037906767
## triadcensus.021U     -0.11518104       0.03790677      1.000000000
## triadcensus.021C     -0.08899275       0.07532660      0.003519666
## triadcensus.111D      0.14190545       0.03682564      0.113756496
## triadcensus.111U      0.19769741      -0.03011590      0.040273876
##                  triadcensus.021C triadcensus.111D triadcensus.111U
## edges                 0.177893898       0.41377429       0.28577313
## nodecov.gpa           0.032985966       0.20440561       0.07795858
## nodecov.smoke         0.023343786      -0.10080895       0.06451232
## triadcensus.012       0.307741724       0.07873964       0.07892790
## triadcensus.102      -0.088992750       0.14190545       0.19769741
## triadcensus.021D      0.075326601       0.03682564      -0.03011590
## triadcensus.021U      0.003519666       0.11375650       0.04027388
## triadcensus.021C      1.000000000      -0.05102796       0.13656840
## triadcensus.111D     -0.051027956       1.00000000      -0.02391001
## triadcensus.111U      0.136568396      -0.02391001       1.00000000
##
## Sample statistics auto-correlation:
## Chain 1
##            edges nodecov.gpa nodecov.smoke triadcensus.012 triadcensus.102
## Lag 0  1.0000000   1.0000000     1.0000000      1.00000000       1.0000000
## Lag 16 0.7527986   0.9137466     0.7428588      0.30728856       0.7920342
## Lag 32 0.6331847   0.8491952     0.5761720      0.13789969       0.6523283
## Lag 48 0.5473971   0.7962424     0.4667701      0.08040812       0.5420028
## Lag 64 0.4904975   0.7538931     0.3945169      0.07289654       0.4639331
## Lag 80 0.4369748   0.7175780     0.3464740      0.04115885       0.3996580
##        triadcensus.021D triadcensus.021U triadcensus.021C triadcensus.111D
## Lag 0         1.0000000        1.0000000        1.0000000        1.0000000
## Lag 16        0.5978485        0.7163682        0.5803922        0.8235679
## Lag 32        0.3975490        0.5450187        0.3813740        0.6976970
## Lag 48        0.2725387        0.4176029        0.2610164        0.5967599
## Lag 64        0.1892317        0.3320668        0.1976986        0.5145900
## Lag 80        0.1284858        0.2608478        0.1472994        0.4466797
##        triadcensus.111U
## Lag 0         1.0000000
## Lag 16        0.7672324
## Lag 32        0.6037371
## Lag 48        0.4818231
## Lag 64        0.3941184
## Lag 80        0.3317499
##
## Sample statistics burn-in diagnostic (Geweke):
## Chain 1
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##            edges      nodecov.gpa    nodecov.smoke  triadcensus.012
##          -0.1024          -1.1524           0.8482           0.3978
##  triadcensus.102 triadcensus.021D triadcensus.021U triadcensus.021C
##           0.5445          -0.7284          -0.4587           0.5111
## triadcensus.111D triadcensus.111U
##          -1.1299           0.9530
##
## Individual P-values (lower = worse):
##            edges      nodecov.gpa    nodecov.smoke  triadcensus.012
##        0.9184064        0.2491428        0.3963057        0.6907833
##  triadcensus.102 triadcensus.021D triadcensus.021U triadcensus.021C
##        0.5861136        0.4663551        0.6464709        0.6092473
## triadcensus.111D triadcensus.111U
##        0.2585083        0.3405963
## Joint P-value (lower = worse):  0.6062509 .
```


Trace of edges
-10
0
50000
100000
Iterations
Density of edges
0.00
-12
-8
-4
0
4
8
11
Trace of nodecov.gpa
-40
0
50000
100000
Iterations
Density of nodecov.gpa
0.00
-40
-20
0
20
40
N = 11251 Bandwidth = 1.49
Trace of nodecov.smoke
-10
0
50000
100000
Iterations
Density of nodecov.smoke
0.00
-15
-10
-5
0
5
10
15
N = 11251 Bandwidth = 0.5854


Trace of triadcensus.012
-100
0
50000
100000
Iterations
Density of triadcensus.012
0.000
-119
-60
-20
20
60
100
Trace of triadcensus.102
-50
0
50000
100000
Iterations
Density of triadcensus.102
0.000
-86
-50
-20
10
40
70
Trace of triadcensus.021D
-5
0
50000
100000
Iterations
Density of triadcensus.021D
0.00
-9
-7
-5
-3
-1
1
3
5
7


Trace of triadcensus.021U
-20
0
50000
100000
Iterations
Density of triadcensus.021U
0.00
-22
-16
-10
-4
0
4
8
14
Trace of triadcensus.021C
-20
0
50000
100000
Iterations
Density of triadcensus.021C
0.00
-20
-14
-8
-2
2
6
12
Trace of triadcensus.111D
-20
0
50000
100000
Iterations
Density of triadcensus.111D
0.00
-27
-15
-5
5
15



```
##
## MCMC diagnostics shown here are from the last round of simulation, prior t
o computation of final parameter estimates. Because the final estimates are r
efinements of those used for this simulation run, these diagnostics may under
state model performance. To directly assess the performance of the final mode
l on in-model statistics, please use the GOF command: gof(ergmFitObject, GOF=
~model).
```

All of the statistics in this model follow distributions that appear to be roughly Gaussian. They are all approximately zero-mean and symmetric, although the distributions for triads 102, 021D, and 021U appear to be slightly skewed left with slightly long tails. Sparse, locally dependent random graph models have statistics that are asymptotically Gaussian. In view of the closeness of the model statistics to Gaussian distributions, we can conclude that the model is converging.

d) Try to fit a model with fewer terms than that in a). What is the preferred model?

The original model had an AIC of 441.2 and a BIC of 488.2, with statistically significant 012D and 111U triad terms.

I began by removing the two least statistically significant triad terms: triads 012 and 102. This fit gave an AIC of 422.4 and a BIC of 460, with statistically significant 021D and 021C triad terms and edges.

I then removed triad 021U. This gave an AIC of 422.2 and a BIC of 455.1, with statisically significant 021D, 021C, 111U triad terms, edges, and gpa covariate.

I then removed triad 111D, which gave an AIC of 418 and a BIC of 446.2. Removing the smoking covariate gave a model with an AIC of 414.5 and BIC of 438. Finally, removing the 111U triad gave an AIC of 415.8 and BIC of 434.6.

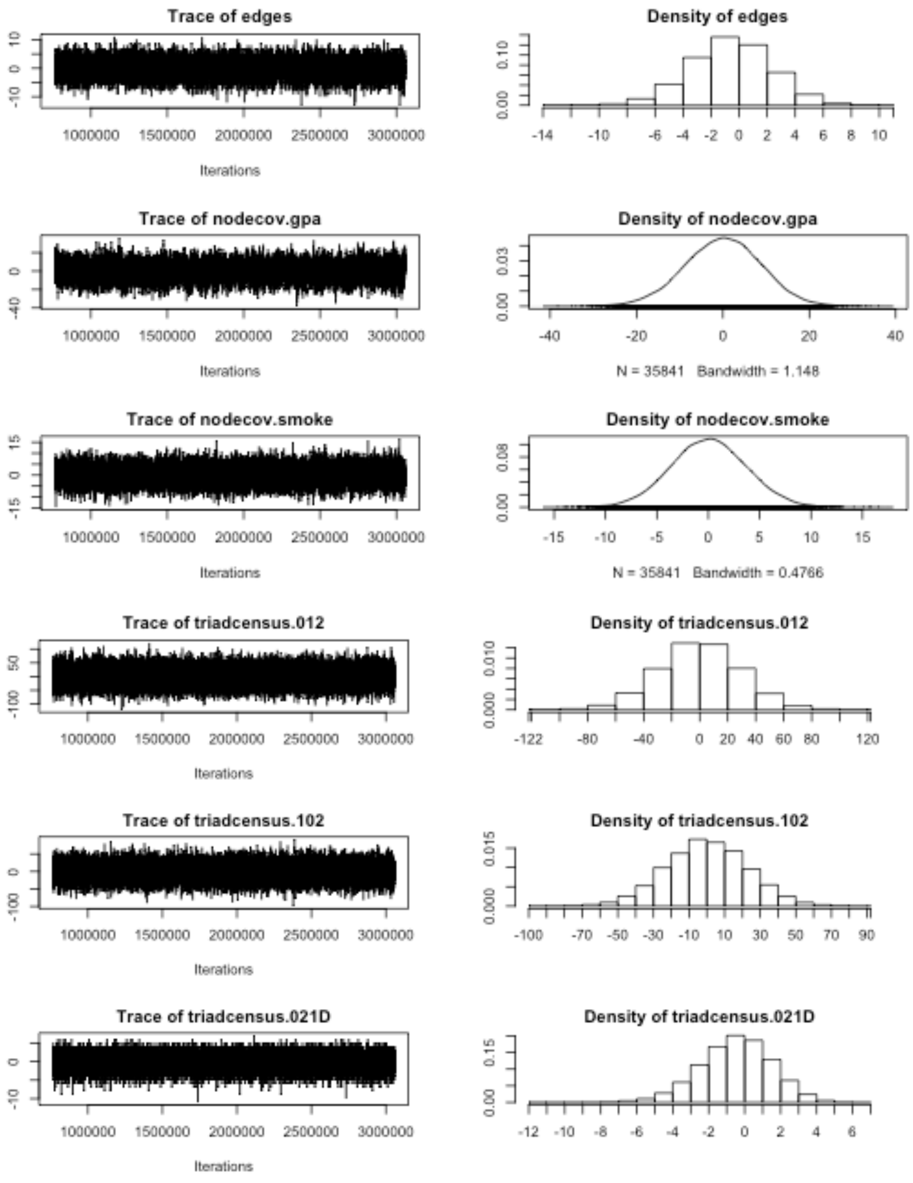
The final model includes edges, the GPA covariate, and triads 021D and 021C. All terms are statistically significant. There is a positive dependence on edges and GPA and negative dependence on the triad terms. The MCMC diagnostics look good (Gaussian).

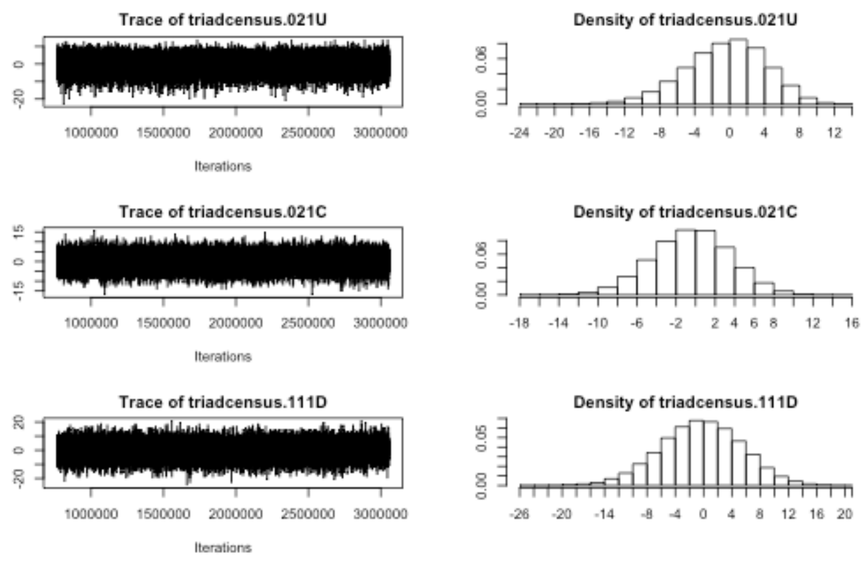
```
library(ergm.tapered)
fit <- ergm.tapered(gf ~ edges + nodecov("gpa") + triadcensus(c(3,5)))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.0453.

## Convergence test p-value: 0.601353978176763. Not converged with 99% confid
ence; increasing sample size.
## Iteration 2 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.1126.
## Estimating equations are not within tolerance region.
## Iteration 3 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.0094.
## Convergence test p-value: 0.298783121534873. Not converged with 99% confid
ence; increasing sample size.
## Iteration 4 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.06031.
## Convergence test p-value: 0.0895353131794767. Not converged with 99% confi
dence; increasing sample size.
## Iteration 5 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.0208.
## Convergence test p-value: 0.0164950322388892. Not converged with 99% confi
dence; increasing sample size.
## Iteration 6 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.02508.
## Convergence test p-value: 0.00357534723886852. Converged with 99% confiden
ce.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```
summary(fit)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   gf ~ Taper(~edges + nodecov("gpa") + triadcensus(c(3, 5)), coef
= .taper.coef,
##     m = .taper.center)
## <environment: 0x7fb974673af8>
##
## Iterations:  6 out of 15
##
## Monte Carlo MLE Results:
##                  Estimate Std. Error MCMC % z value Pr(>|z|)
## edges             -1.9639     0.2322      0  -8.458   <1e-04 ***
## nodecov.gpa        0.2797     0.1099      0   2.544   0.0110 *
## triadcensus.021D  -1.0520     0.4625      0  -2.275   0.0229 *
## triadcensus.021C  -0.5076     0.2323      0  -2.185   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 1125.7  on 812  degrees of freedom
##  Residual Deviance:  408.6  on 808  degrees of freedom
##
## AIC: 416.6    BIC: 435.4    (Smaller is better.)
```

```
mcmc.diagnostics(fit)
```


Trace of edges
Density of edges
Iterations
Trace of nodecov.gpa
Density of nodecov.gpa
N = 35841 Bandwidth = 1.148
Trace of nodecov.smoke
Density of nodecov.smoke
N = 35841 Bandwidth = 0.4766
Trace of triadcensus.012
Density of triadcensus.012
Trace of triadcensus.102
Density of triadcensus.102
Trace of triadcensus.021D
Density of triadcensus.021D

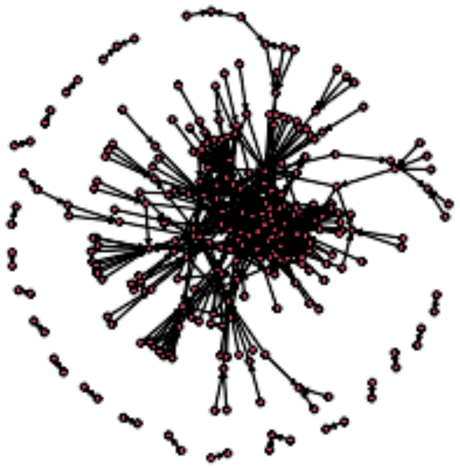


5) Model for Protein-protein interaction data: Butland et al (2005) “Interaction network containing conserved and essential protein complexes in Escherichia coli” reported a net- work of protein-protein interactions (bindings) that we obtained from http://pil.phys. uniroma1.it/~gcalda/cosinsite/extra/data/proteins/. This data is available in the networkdata package

```
library(networkdata)
data(butland_ppi)
help(butland_ppi)
```

Convert the edgelist to a directed network (The el2sm function may be helpful).

```
adj = el2sm(butland_ppi)
bl <- network(x = adj,directed = TRUE,loops = FALSE,matrix.type = "adjacency"
)
gplot(bl)
```



Fit various tapered ERGM models to the network using ergm.tapered. Consider terms documented under ergm-terms. Good candidates include istar, ostar, gwodegree, gwidegree, dgwest, dgwdsp, ctriple, ttriple.

I began by fitting a model on istars and ostars from 1 to 3. The models did not converge.

I next fit models on gwodegree and gwidegree, the geometrically-weighted degree distributions. These also did not converge.

I next fit a dgwdsp term (dgwdsp is the geometrically-weighted edgewise shared partner distribution). This also did not converge.

I next fit a model on ttriple, which did converge. The edges and ttriple statistics are both highly statistically significant. The edges coefficient is negative and larger in magnitude, while the ttriple statistic is positive, which indicates a tendency to form transitive triads. On the other hand, ctriple did not converge.

I next fit a model on ttriple and istar(1), which did not converge.

```r
library(ergm.tapered)
# fit <- ergm.tapered(bl ~ edges + istar(1))
# fit <- ergm.tapered(bl ~ edges + gwodegree(1))
# fit <- ergm.tapered(bl ~ edges + dgwdsp)
fit <- ergm.tapered(bl ~ edges + ttriple)

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 15:

## Optimizing with step length 1.

## The log-likelihood did not improve.

## Estimating equations are not within tolerance region.

## Iteration 2 of at most 15:

## Optimizing with step length 1.

## The log-likelihood did not improve.

## Estimating equations are not within tolerance region.

## Iteration 3 of at most 15:

## Optimizing with step length 1.

## The log-likelihood did not improve.

## Estimating equations are not within tolerance region.

## Iteration 4 of at most 15:

## Optimizing with step length 1.

## The log-likelihood did not improve.

## Estimating equations are not within tolerance region.

## Iteration 5 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 14.78.

## Estimating equations are not within tolerance region.

## Iteration 6 of at most 15:

## Optimizing with step length 1.
## The log-likelihood improved by 15.05.

## Estimating equations are not within tolerance region.

## Iteration 7 of at most 15:

## Optimizing with step length 1.

## The log-likelihood did not improve.

## Estimating equations are not within tolerance region.

## Iteration 8 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 14.93.

## Estimating equations are not within tolerance region.

## Iteration 9 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.1079.

## Estimating equations are not within tolerance region.

## Iteration 10 of at most 15:

## Optimizing with step length 1.

## The log-likelihood improved by 0.004827.

## Convergence test p-value: 0.5375418681348. Not converged with 99% confiden
ce; increasing sample size.
## Iteration 11 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.263.
## Estimating equations are not within tolerance region.
## Iteration 12 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.3366.
## Estimating equations are not within tolerance region.
## Estimating equations did not move closer to tolerance region more than 1 t
ime(s) in 4 steps; increasing sample size.
## Iteration 13 of at most 15:
## Optimizing with step length 1.
## The log-likelihood improved by 0.01617.
## Convergence test p-value: 0.00392123202364821. Converged with 99% confiden
ce.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 .
```

```
## This model was fit using MCMC.  To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```r
summary(fit)
```

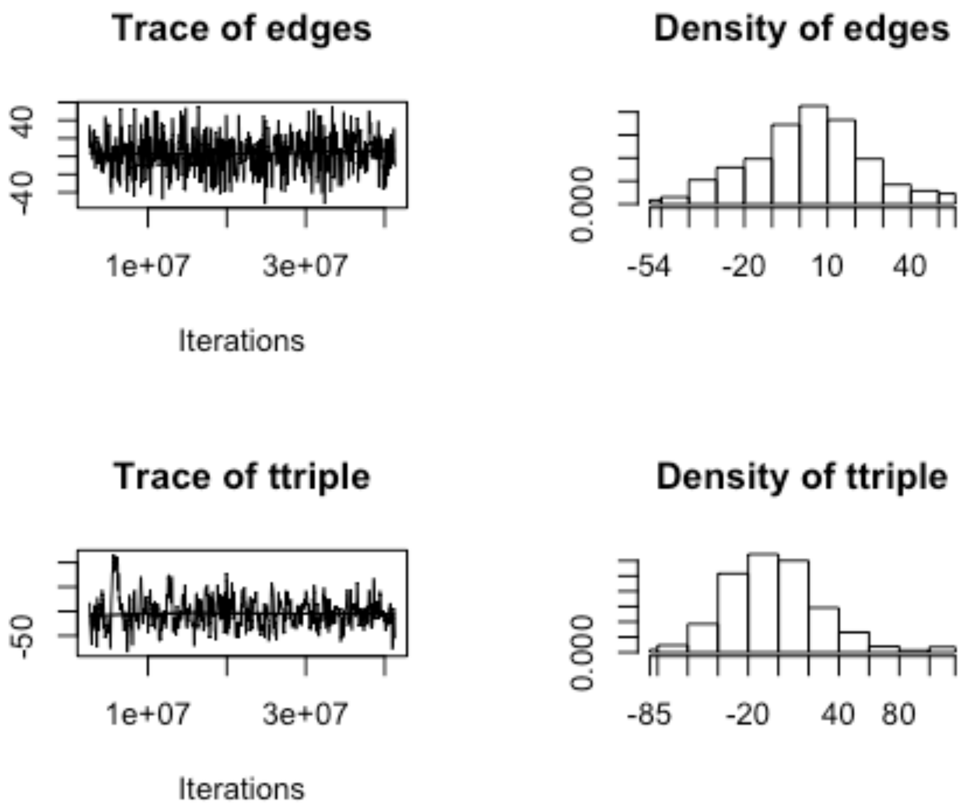
```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   bl ~ Taper(~edges + ttriple, coef = .taper.coef, m = .taper.cen
ter)
## <environment: 0x7fb973141b68>
##
## Iterations:  13 out of 15
##
## Monte Carlo MLE Results:
##         Estimate Std. Error MCMC % z value Pr(>|z|)
## edges   -4.75096    0.04820      0  -98.56   <1e-04 ***
## ttriple  0.41220    0.03071      0   13.42   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 100687  on 72630  degrees of freedom
##  Residual Deviance:   7676  on 72628  degrees of freedom
##
## AIC: 7680    BIC: 7699    (Smaller is better.)
```

Check the MCMC diagnostics with mcmc.diagnostics. Overall, how does the goodness-of-fit look?

```r
mcmc.diagnostics(fit)
```

```
## Sample statistics summary:
##
## Iterations = 2621440:41287680
## Thinning interval = 65536
## Number of chains = 1
## Sample size per chain = 591
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean    SD Naive SE Time-series SE
## edges     3.323 20.64   0.8491         0.8491
## ttriple -2.406 31.70   1.3039         2.8581
##
## 2. Quantiles for each variable:
##
```

```
##          2.5% 25% 50% 75% 97.5%
## edges   -39.0  -9   4  17 45.25
## ttriple -56.5 -25  -4  16 71.50
##
##
## Are sample statistics significantly different from observed?
##                    edges    ttriple Overall (Chi^2)
## diff.      3.323181e+00 -2.4060914              NA
## test stat. 3.913785e+00 -0.8418612    1.463831e+01
## P-val.     9.086068e-05  0.3998657    8.268632e-04
##
## Sample statistics cross-correlations:
##              edges    ttriple
## edges   1.00000000 0.02438012
## ttriple 0.02438012 1.00000000
##
## Sample statistics auto-correlation:
## Chain 1
##                    edges   ttriple
## Lag 0       1.000000000 1.0000000
## Lag 65536  -0.016612767 0.5916281
## Lag 131072  0.021203816 0.4167055
## Lag 196608 -0.036812756 0.2847929
## Lag 262144 -0.009619795 0.1862836
## Lag 327680 -0.029672458 0.1230926
##
## Sample statistics burn-in diagnostic (Geweke):
## Chain 1
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   edges ttriple
## -0.7556  0.5413
##
## Individual P-values (lower = worse):
##     edges   ttriple
## 0.4499091 0.5882855
## Joint P-value (lower = worse):  0.4468981 .
```



```
##
## MCMC diagnostics shown here are from the last round of simulation, prior t
o computation of final parameter estimates. Because the final estimates are r
efinements of those used for this simulation run, these diagnostics may under
state model performance. To directly assess the performance of the final mode
l on in-model statistics, please use the GOF command: gof(ergmFitObject, GOF=
~model).
```

Th edges and ttriple statistics are not perfectly symmetric, but nevertheless they are roughly Guassian, which indicates that the model is approaching convergence.