# HW4

Zihan Lin

2024-11-07

```
library(networkdata)
library(network)
library(ergm)
library(sna)
data(ffef)
data(hansell)
data(butland_ppi)
```

## Question 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.

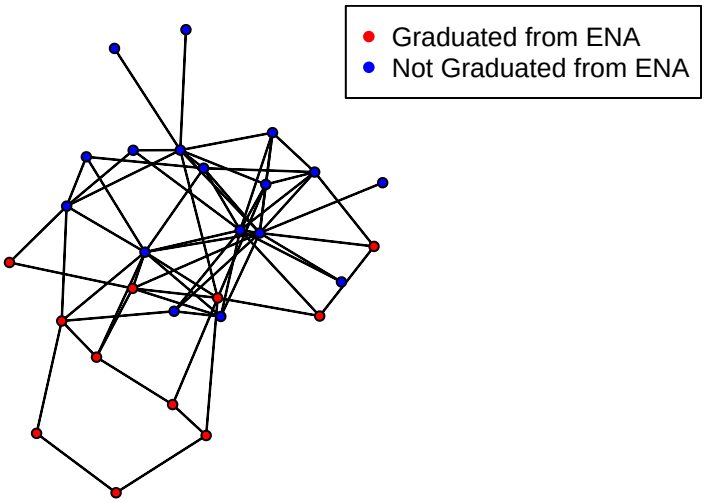
There are many vertex covariates, including:

- 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.

**a)**

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

```
plot(ffef,
     vertex.col = ifelse(network::get.vertex.attribute(ffef, "ena") == 2, "red", "blue"))
legend("topright", pch = 16, cex = 0.8,
       legend = c("Graduated from ENA", "Not Graduated from ENA"),
       col = c("red", "blue"))
```



**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)). Hint: Use ergm.tapered() as it better deals with strong dependence between terms.

```
ffef_fit <- ergm(ffef ~ edges +
                        nodematch("ena") +
                        nodematch("prestige") +
                        nodematch("party") +
                        gwesp(0.5, fixed = TRUE) +
                        gwdsp(0.5, fixed = TRUE))

summary(ffef_fit)
```

```
## Call:
## ergm(formula = ffef ~ edges + nodematch("ena") + nodematch("prestige") +
##     nodematch("party") + gwesp(0.5, fixed = TRUE) + gwdsp(0.5,
##     fixed = TRUE))
##
## Monte Carlo Maximum Likelihood Results:
##
##                    Estimate Std. Error MCMC % z value Pr(>|z|)
## edges              -4.68705    0.52578      0  -8.914  < 1e-04 ***
## nodematch.ena       1.59471    0.37864      0   4.212  < 1e-04 ***
## nodematch.prestige  0.68555    0.30720      0   2.232  0.02564 *
## nodematch.party     1.23641    0.43837      0   2.820  0.00480 **
## gwesp.fixed.0.5     0.43483    0.19693      0   2.208  0.02724 *
## gwdsp.fixed.0.5     0.19034    0.06742      0   2.823  0.00475 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 524.0  on 378  degrees of freedom
##  Residual Deviance: 296.2  on 372  degrees of freedom
##
## AIC: 308.2  BIC: 331.8  (Smaller is better. MC Std. Err. = 0.3057)
```

**c)**

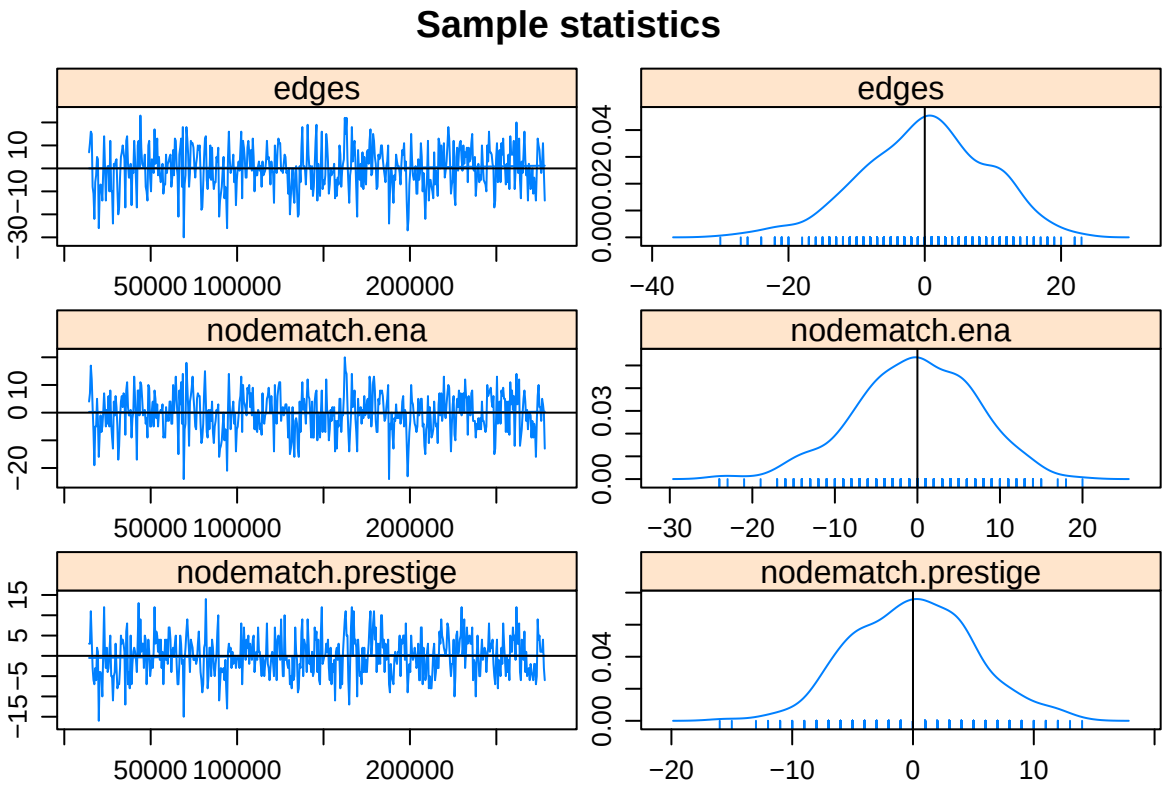
Give an interpretation for each of the coefficients in the model in terms of what it means and also what its magnitude indicates about the nature of social relations in the network.

- Edges (-4.6870521): This value shows the basic likelihood of two people in the network being friends. Since it's a large negative number, it suggests that friendships are generally uncommon unless there are specific reasons to form connections.
- ENA (1.5947063): This positive value means that people who both attended ENA, a prestigious school, are much more likely to be friends. ENA attendance is a strong factor in forming friendships, meaning people are drawn to others with the same educational background.
- Prestige (0.6855549): This positive value shows that people with similar social prestige (or status) are more likely to be friends. However, the effect is smaller than that for ENA attendance, so while status matters, it's not as important as educational background.
- Party (1.2364051): This positive value indicates that people in the same political party tend to form friendships. It's a fairly strong factor in forming connections, showing that political alignment also influences friendships, though it's less powerful than shared education.
- GWESP (0.4348274): This value shows a tendency for "friend-of-a-friend" connections, meaning if two people have a mutual friend, they're more likely to become friends themselves. This tendency to form "triangles" or small, connected groups adds some structure to the network.
- GWDSP (0.1903399): This value represents connections where two people have mutual friends but don't necessarily become friends themselves. It shows that people tend to have indirect social connections, even if they're not all part of tightly connected groups.

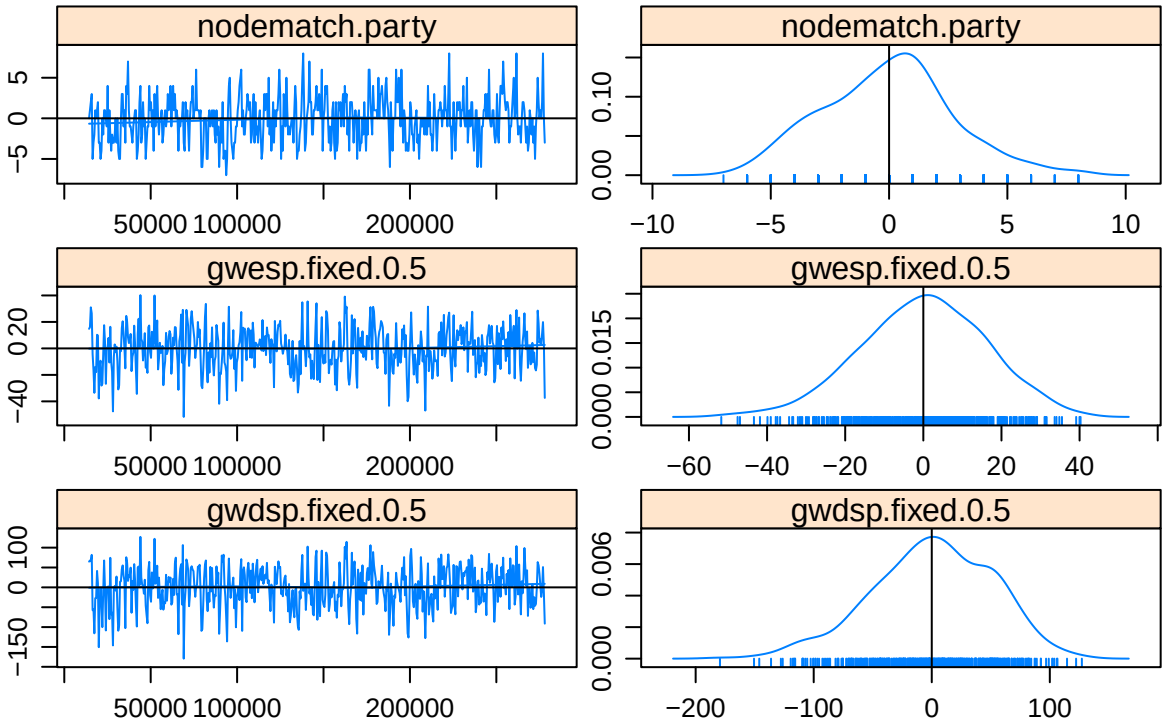
**d)**

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(ffef_fit)
```



## Sample statistics




```
## Sample statistics summary:
##
## Iterations = 14336:278016
## Thinning interval = 512
## Number of chains = 1
## Sample size per chain = 516
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                         Mean     SD Naive SE Time-series SE
## edges                0.03101  9.220   0.4059         0.5587
## nodematch.ena       -0.26744  7.241   0.3187         0.4805
## nodematch.prestige   0.00000  4.991   0.2197         0.2634
## nodematch.party     -0.04845  2.751   0.1211         0.1699
## gwesp.fixed.0.5     -0.03089 16.037   0.7060         0.9779
## gwdsp.fixed.0.5      0.24205 51.312   2.2589         2.9106
##
## 2. Quantiles for each variable:
##
##                        2.5%    25%    50%   75% 97.5%
## edges                -20.12  -6.00 0.0000  6.00 16.12
## nodematch.ena        -15.00  -5.00 0.0000  5.00 13.00
## nodematch.prestige    -9.00  -4.00 0.0000  3.00 10.12
## nodematch.party       -5.00  -2.00 0.0000  2.00  6.00
## gwesp.fixed.0.5      -32.27 -10.26 0.2827 11.11 29.35
## gwdsp.fixed.0.5     -109.53 -32.56 2.1275 39.77 87.76
##
##
```

```
## Are sample statistics significantly different from observed?
##                 edges nodematch.ena nodematch.prestige nodematch.party
## diff.      0.03100775    -0.2674419                 0     -0.04844961
## test stat. 0.05550058    -0.5566246                 0     -0.28515972
## P-val.     0.95573966     0.5777839                 1      0.77552178
##            gwesp.fixed.0.5 gwdsp.fixed.0.5    (Omni)
## diff.          -0.03088805      0.24205347        NA
## test stat.     -0.03158747      0.08316401 5.3719465
## P-val.          0.97480104      0.93372113 0.5078659
##
## Sample statistics cross-correlations:
##                       edges nodematch.ena nodematch.prestige nodematch.party
## edges              1.0000000     0.9081023          0.7505135       0.5369692
## nodematch.ena      0.9081023     1.0000000          0.6473320       0.4713472
## nodematch.prestige 0.7505135     0.6473320          1.0000000       0.4054200
## nodematch.party    0.5369692     0.4713472          0.4054200       1.0000000
## gwesp.fixed.0.5    0.9326568     0.8809071          0.6920892       0.5038914
## gwdsp.fixed.0.5    0.9397351     0.8092261          0.6976084       0.4708972
##                    gwesp.fixed.0.5 gwdsp.fixed.0.5
## edges                    0.9326568       0.9397351
## nodematch.ena            0.8809071       0.8092261
## nodematch.prestige       0.6920892       0.6976084
## nodematch.party          0.5038914       0.4708972
## gwesp.fixed.0.5          1.0000000       0.9161384
## gwdsp.fixed.0.5          0.9161384       1.0000000
##
## Sample statistics auto-correlation:
## Chain 1
##                edges nodematch.ena nodematch.prestige nodematch.party
## Lag 0     1.00000000    1.00000000        1.000000000     1.000000000
## Lag 512   0.27983795    0.29376648        0.250896337     0.325295145
## Lag 1024  0.03518595    0.08709496        0.043725643     0.109775309
## Lag 1536 -0.07892470   -0.09200070        0.047077163    -0.007082446
## Lag 2048 -0.08247567   -0.02994743        0.010678098    -0.032544215
## Lag 2560 -0.05043874   -0.02153249        0.001480904     0.026012037
##          gwesp.fixed.0.5 gwdsp.fixed.0.5
## Lag 0         1.00000000      1.00000000
## Lag 512       0.27830335      0.25157601
## Lag 1024      0.01825343     -0.01333814
## Lag 1536     -0.10352187     -0.07662432
## Lag 2048     -0.09060297     -0.06021990
## Lag 2560     -0.04406080     -0.04578733
##
## 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.party
##         -1.4831422         -0.1992715         -1.7899051         -1.7416459
##    gwesp.fixed.0.5    gwdsp.fixed.0.5
##         -0.7752653         -0.7411071
##
```

```
## Individual P-values (lower = worse):
##              edges      nodematch.ena nodematch.prestige    nodematch.party
##         0.13803664         0.84205034         0.07346916         0.08157042
##    gwesp.fixed.0.5    gwdsp.fixed.0.5
##         0.43818291         0.45862850
## Joint P-value (lower = worse):  0.01723092
##
## Note: MCMC diagnostics shown here are from the last round of
##   simulation, prior to computation of final parameter estimates.
##   Because the final estimates are refinements of those used for this
##   simulation run, these diagnostics may understate model performance.
##   To directly assess the performance of the final model on in-model
##   statistics, please use the GOF command: gof(ergmFitObject,
##   GOF=~model).
```

The MCMC diagnostics show that the model has mostly converged well. The trace plots are stable, meaning the model's estimates are not wandering and are centered around a consistent average. The low time-series standard errors mean the estimates are reliable. Most of the p-values are above 0.05, showing that the model fits the observed data well, although the gwesp term has a slightly low p-value (0.033), suggesting the model may not fully capture clustering effects. The decreasing autocorrelations show that the samples are mixing well, and the Geweke test's high p-value (0.885) confirms the model is stable over time. Overall, the model is a good fit for the data, with only a small improvement needed for clustering.

**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?

```
fit_extended <- ergm(ffef ~ edges +
                            nodematch("ena") +
                            nodematch("prestige") +
                            nodematch("party") +
                            nodematch("masons") +
                            gwesp(0.5, fixed = TRUE) +
                            gwdsp(0.5, fixed = TRUE))
summary(fit_extended)
```

```
## Call:
## ergm(formula = ffef ~ edges + nodematch("ena") + nodematch("prestige") +
##     nodematch("party") + nodematch("masons") + gwesp(0.5, fixed = TRUE) +
##     gwdsp(0.5, fixed = TRUE))
##
## Monte Carlo Maximum Likelihood Results:
##
##                     Estimate Std. Error MCMC % z value Pr(>|z|)
## edges               -4.82188    0.54878      0  -8.786  < 1e-04 ***
## nodematch.ena        1.60263    0.39206      0   4.088  < 1e-04 ***
## nodematch.prestige   0.70559    0.30669      0   2.301  0.02141 *
## nodematch.party      1.25715    0.43779      0   2.872  0.00408 **
## nodematch.masons     0.23381    0.27991      0   0.835  0.40353
## gwesp.fixed.0.5      0.43103    0.19590      0   2.200  0.02778 *
## gwdsp.fixed.0.5      0.18950    0.06404      0   2.959  0.00308 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 524.0  on 378  degrees of freedom
##  Residual Deviance: 296.7  on 371  degrees of freedom
##
## AIC: 310.7  BIC: 338.3  (Smaller is better. MC Std. Err. = 0.3219)
```

The social structure of the French financial elite network shows that friendships are selective and built around common backgrounds and affiliations. Having attended ENA, a prestigious school, is the biggest factor in forming friendships, followed by being in the same political party and having similar social status. There is a strong tendency for friends of friends to also become friends, which creates tightly connected social circles. Indirect connections through shared friends are also common, even if they don't form fully connected groups. Membership in the Masons does not seem to have much impact on friendships, showing that not all shared memberships matter in this network. Overall, this model reveals a network where friendships are shaped by shared education, social status, political ties, and clustering, with social and political boundaries influencing who becomes friends.

## Question 2

Modeling Balance in Friendship Relations: Here we consider again the network introduced in Homework 3 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 statistics ttriad and ctriad count the number of transitive triads and the number of cyclic triads, respectively.

```
network::get.vertex.attribute(hansell, "sex")
```

```
##  [1] "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"
##  [9] "male"   "male"   "male"   "male"   "male"   "female" "female" "female"
## [17] "female" "female" "female" "female" "female" "female" "female" "female"
## [25] "female" "female" "female"
```

### a)

Is the friendship network balanced in Heider's definition of balance? Give a reason why or why not.

```
summary(hansell ~ ttriad)
```

```
## ttriple
##     400
```

```
summary(hansell ~ ctriad)
```

```
## ctriple
##      64
```

Since the number of transitive triads is 400 and the number of cyclic triads is 64, $400 > 64$, the network is more likely to be balanced according the Heider;s definition, as transitive triads are more common than cyclic triads.

**b)**

We can measure the statistical degree of balance in a network by including the ttriad and ctriad statistics in the model, as these count of the number of transitive triads and the count of the number of cyclic triads, respectively. If the coefficient of the transitive triad statistic is positive, then the model places higher probability on networks with transitive triads - that is, on balanced networks. We may also see the same or less cyclic triads compared to a neutral random network.

Fit the model with transitive and cyclic triads as well as foundational statistics for the overall density, the mutuality of ties and the homophily on sex using the ergm.tapered command:

```
fit <- ergm(hansell ~ edges + mutual + nodematch("sex", diff = TRUE) +
                    ttriad + ctriad, estimate = "MPLE")
summary(fit)
```

```
## Call:
## ergm(formula = hansell ~ edges + mutual + nodematch("sex", diff = TRUE) +
##     ttriad + ctriad, estimate = "MPLE")
##
## Maximum Pseudolikelihood Results:
##
##                       Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                -3.17910    0.24078      0 -13.204  < 1e-04 ***
## mutual                0.28255    0.27575      0   1.025  0.30554
## nodematch.sex.female  0.77929    0.27961      0   2.787  0.00532 **
## nodematch.sex.male    0.96416    0.26174      0   3.684  0.00023 ***
## ttriple               0.33445    0.03453      0   9.687  < 1e-04 ***
## ctriple              -0.37629    0.09046      0  -4.160  < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Warning:  The standard errors are based on naive pseudolikelihood and are suspect. Set control.ergm$
##
##      Null Pseudo-deviance: 973.2  on 702  degrees of freedom
##  Residual Pseudo-deviance: 570.2  on 696  degrees of freedom
##
## AIC: 582.2  BIC: 609.6  (Smaller is better. MC Std. Err. = 0)
```

**Give a brief interpretation of the coefficients of the first three terms.**

**Does there appear to be a general preference for mutual friendship ties? Does there appear to be a general preference for same-sex friendship ties?**

- Edges (-3.1790993): This value shows the basic likelihood of two students in the network being friends. Since it's a large negative number, it suggests that friendships are generally uncommon unless there are specific reasons for students to connect.
- Mutual (0.2825465): This positive value indicates a tendency for friendships to be mutual, meaning if one student considers another a friend, the feeling is likely to be returned. However, the effect is not statistically significant, suggesting that there isn't a strong or consistent preference for friendships to be mutual in this network.
- Same-Sex Friendship (Female: 0.7792941, Male: 0.9641633): These positive values show that students are much more likely to form friendships with others of the same sex. Both coefficients are statistically significant, indicating a strong preference for same-sex friendships. Boys are slightly more likely to prefer same-sex friendships than girls in this classroom network.

**c)**

**Give a brief interpretation of the coefficients of the ttriad and ctriad terms.**

- Transitive Triads (ttriad: 0.3344506): This positive value shows that students in the network tend to form "friend-of-a-friend" relationships. If student A is friends with student B, and student B is friends with student C, student A is also likely to be friends with student C. This preference for forming closed triangles means that the network has a strong tendency toward clustering, where groups of friends are closely connected.
- Cyclic Triads (ctriad: -0.376291): This negative value shows that students in the network tend to avoid forming cyclic relationships. A cyclic triad is when student A is friends with student B, student B is friends with student C, and student C is friends with student A, but without the triangle closing in a transitive way. The negative coefficient suggests that these kinds of unbalanced, looping relationships are rare in the network.

**Describe the pattern of transitive and cyclic ties.**

- The positive value for transitive triads and negative value for cyclic triads indicate that students prefer balanced, tightly connected friendships. Friendships tend to close into triangles, creating cohesive groups rather than open loops. This pattern supports a stable social structure where students are more likely to have friends in common, fostering a harmonious network in the classroom.

**d)**

**Intuitive, what will the coefficient of the transitive triad statistics be if the network is balanced?**

- If the network is balanced, we would expect the coefficient of the transitive triad statistic to be positive.

**Based on this model, does there appear to be a general preference for balanced friendship ties?**

- The model shows a positive and statistically significant coefficient for transitive triads (ttriad: 0.3344506), which suggests a preference for balanced relationships. Additionally, the negative coefficient for cyclic triads (ctriad: -0.376291) indicates that students avoid cyclic, unbalanced relationships. Together, these coefficients imply a general preference for balanced friendship ties in this classroom network, as students tend to form stable, triangular friendships rather than unbalanced loops.

## Question 3

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

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

```
but_net <- network(el2sm(butland_ppi, directed = TRUE), directed = TRUE)
but_net
```

```
##  Network attributes:
##   vertices = 270
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 716
##     missing edges= 0
##     non-missing edges= 716
##
```

```
##  Vertex attribute names:
##     vertex.names
##
## No edge attributes
```

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

```
#fit1 <- ergm(
  #but_net ~ edges + istar(2) + ostar(2) + gwodegree(0.5, fixed = TRUE) + gwidegree(0.5, fixed = TRUE)
#)

#fit2 <- ergm(
  #but_net ~ edges + ctriple + ttriple + dgwest(0.5, fixed = TRUE) + dgwdsp(0.5, fixed = TRUE)
#)

#fit3 <- ergm(
  #but_net ~ edges + istar(2) + ostar(2) + ctriple + ttriple + gwodegree(0.5, fixed = TRUE) + gwidegree
#)

#summary(fit1)
#summary(fit2)
#summary(fit3)
```