Assignment 3 - PLSC 508

Joseph Risi

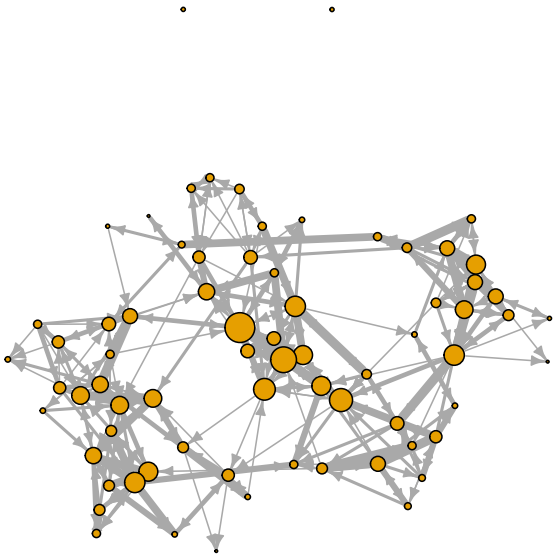
Data comes from the Add Health in-school questionnaire for the years 1994-1995. Students were asked about whom there friends were, and how many activities they completed together. For the purposes of this assignment, I only consider whether or not a student listed someone as their friend (1/0) and not the strength of friendship as indicated by how many activities the students participated in together.

Data and code documenation can be found here: https://github.com/zalmquist/networkdata/blob/master/man/

```
library(sna)
library(ergm)
library(here)
library(dplyr)
library(igraph)
library(latentnet)
library(intergraph)
```

```
# load in the data
load(here("create-networks", "src", "addhealth.rda"))

# Select one of the Add Health networks and represent it as a network object
# and an igraph object.
add_health1_network <- addhealth[[1]]
add_health1_igraph <- asIgraph(add_health1_network)

# Visualize network
plot(add_health1_igraph,
     vertex.size = degree(add_health1_igraph) / 2,
     edge.width = E(add_health1_igraph)$edgevalue,
     vertex.label = NA,
     edge.arrow.size = 0.5,
     main = "Friendship Network Using Add Health Data")
```

Friendship Network Using Add Health Data



QAP Estimation

Creating the Independent and Dependent Variables

```
# Check for grade sender, receiver, and distance effects
grade_send <- matrix(V(add_health1_igraph)$grade,
                     length(V(add_health1_igraph)),
                     length(V(add_health1_igraph)),
                     byrow = F)

grade_receive <- matrix(V(add_health1_igraph)$grade,
                        length(V(add_health1_igraph)),
                        length(V(add_health1_igraph)),
                        byrow = T)
```

```r
grade_dist <- as.matrix(dist(V(add_health1_igraph)$grade))

# Check for gender effects
sex_matrix1 <- matrix(V(add_health1_igraph)$sex,
                      length(V(add_health1_igraph)),
                      length(V(add_health1_igraph)),
                      byrow = T)

sex_matrix2 <- matrix(V(add_health1_igraph)$sex,
                      length(V(add_health1_igraph)),
                      length(V(add_health1_igraph)),
                      byrow = F)

# Are the pair of students the same sex or a different sex?
sex_same <- matrix(as.numeric(sex_matrix1 == sex_matrix2),
                   ncol = ncol(sex_matrix1))
diag(sex_same) <- 0

sex_diff <- matrix(as.numeric(sex_matrix1 != sex_matrix2),
                    ncol = ncol(sex_matrix1))

sex_check <- sum(sex_same + sex_diff) == ncol(sex_diff) * ncol(sex_diff) - ncol(sex_diff)

# Are the pair of students the same race or a different race?
race_matrix1 <- matrix(V(add_health1_igraph)$race,
                       length(V(add_health1_igraph)),
                       length(V(add_health1_igraph)),
                       byrow = T)

race_matrix2 <- matrix(V(add_health1_igraph)$race,
                       length(V(add_health1_igraph)),
                       length(V(add_health1_igraph)),
                       byrow = F)

race_same <- matrix(as.numeric(race_matrix1 == race_matrix2),
                    ncol = ncol(race_matrix1))
diag(race_same) <- 0

race_diff <- matrix(as.numeric(race_matrix1 != race_matrix2),
                    ncol = ncol(race_matrix1))
race_check <- sum(race_same + race_diff) == ncol(race_diff) * ncol(race_diff) - ncol(race_
```

```r
# Combine all the independent variables together
covariates <- list("grade_send" = grade_send, "grade_receive" = grade_receive,
                   "grade_dist" = grade_dist, "sex_same" = sex_same,
                   "race_same" = race_same)

# Dependent variable matrix
friendship_matrix <- add_health1_igraph %>% as_adjacency_matrix(sparse = F)
```

Estimate the QAP model

```r
set.seed(5)
qap <- netlogit(friendship_matrix, covariates, nullhyp = "qap", reps = 1000)
qap$names <- c("(Intercept)", names(covariates))
summary(qap)
```

```
Network Logit Model

Coefficients:
              Estimate   Exp(b)     Pr(<=b) Pr(>=b) Pr(>=|b|)
(Intercept)   -2.4223937 0.08870902 0.003   0.997   0.005
grade_send    -0.0487271 0.95244101 0.354   0.646   0.718
grade_receive  0.1477136 1.15918085 0.857   0.143   0.310
grade_dist    -1.9553064 0.14152111 0.000   1.000   0.000
sex_same       0.4521337 1.57166206 0.995   0.005   0.005
race_same      0.3280842 1.38830583 0.922   0.078   0.164

Goodness of Fit Statistics:

Null deviance: 6889.883 on 4970 degrees of freedom
Residual deviance: 1662.288 on 4964 degrees of freedom
Chi-Squared test of fit improvement:
     5227.595 on 6 degrees of freedom, p-value 0
AIC: 1674.288   BIC: 1713.355
Pseudo-R^2 Measures:
    (Dn-Dr)/(Dn-Dr+dfn): 0.5126302
    (Dn-Dr)/Dn: 0.758735
Contingency Table (predicted (rows) x actual (cols)):
```

```
       0      1
0   4665    305
1      0      0

    Total Fraction Correct: 0.9386318
    Fraction Predicted 1s Correct: NaN
    Fraction Predicted 0s Correct: 0.9386318
    False Negative Rate: 1
    False Positive Rate: 0

Test Diagnostics:

    Null Hypothesis: qap
    Replications: 1000
    Distribution Summary:

       (Intercept) grade_send grade_receive grade_dist  sex_same race_same
Min      -5.585732  -3.451648     -3.993952  -4.368445 -3.044712 -4.623027
1stQ     -1.382982  -0.892274     -1.077666  -0.814614 -0.812466 -1.041117
Median   -0.161794  -0.007678     -0.028934   0.005167 -0.024215 -0.038483
Mean     -0.076831   0.025158     -0.020509   0.007012  0.021723 -0.008732
3rdQ      1.240098   0.937195      0.924668   0.986982  0.808841  1.027121
Max       6.555040   4.066693      4.593300   3.736232  4.422629  4.772838
```

Results from the QAP model indicate model fit could be an issue considering it did very poorly on its predictive tasks. In terms of variable importance, it seems as if students being the same sex increased the probability of a tie forming and being further apart in grade decreased the probability of a tie forming. There were no significant effects for race, and there does not appear to any sender/receiver effects for grade.

Estimate ERGM model

Base model with variables

```r
# Estimate an ERGM model with no covariates
ergm_base <- ergm(add_health1_network ~ edges)
```

```
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(ergm_base)
```

```
Call:
ergm(formula = add_health1_network ~ edges)

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges  -2.7275     0.0591      0  -46.15   <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 6890  on 4970  degrees of freedom
 Residual Deviance: 2293  on 4969  degrees of freedom

AIC: 2295  BIC: 2302  (Smaller is better. MC Std. Err. = 0)
```

```
base_pr <- 1 / (1 + exp(-coef(ergm_base)[1]))
```

The base ERGM model indicates there is a 6.1368209% chance of a tie or friendship being mentioned on the survey. This indicates that ties are very rare. This may help explain why the QAP model performed so poorly because it could not find any combination of node-level and/or dyad-level variables in which the chance of a tie forming was more likely than not.

Estimate a ERGM model using only exogeneous variables

```
# Estimate an ERGM model using the covariates from above that we used in our
# QAP model
ergm_exog <- ergm(add_health1_network ~ edges + nodematch("race") + nodematch("sex") +
```

```
                        nodeicov("grade") + nodeocov("grade") + absdiff("grade"))
```

```
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(ergm_exog)
```

```
Call:
ergm(formula = add_health1_network ~ edges + nodematch("race") +
    nodematch("sex") + nodeicov("grade") + nodeocov("grade") +
    absdiff("grade"))

Maximum Likelihood Results:

               Estimate Std. Error MCMC % z value Pr(>|z|)
edges          -2.42239    0.44425      0  -5.453  < 1e-04 ***
nodematch.race  0.32808    0.15454      0   2.123 0.033755 *
nodematch.sex   0.45213    0.13028      0   3.471 0.000519 ***
nodeicov.grade  0.14771    0.09792      0   1.508 0.131429
nodeocov.grade -0.04873    0.09786      0  -0.498 0.618540
absdiff.grade  -1.95531    0.11308      0 -17.291  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 6890  on 4970  degrees of freedom
 Residual Deviance: 1662  on 4964  degrees of freedom

AIC: 1674  BIC: 1713  (Smaller is better. MC Std. Err. = 0)
```

```r
# Interpretation
base_pr <- 1 / (1 + exp(-coefficients(ergm_exog)[["edges"]]))
same_race <- 1 / (1 + exp(-coefficients(ergm_exog)[["edges"]] - coefficients(ergm_exog)[["
same_sex <- 1 / (1 + exp(-coefficients(ergm_exog)[["edges"]] - coefficients(ergm_exog)[["n
diff_grade <- 1 / (1 + exp(-coefficients(ergm_exog)[["edges"]] - coefficients(ergm_exog)[[
```

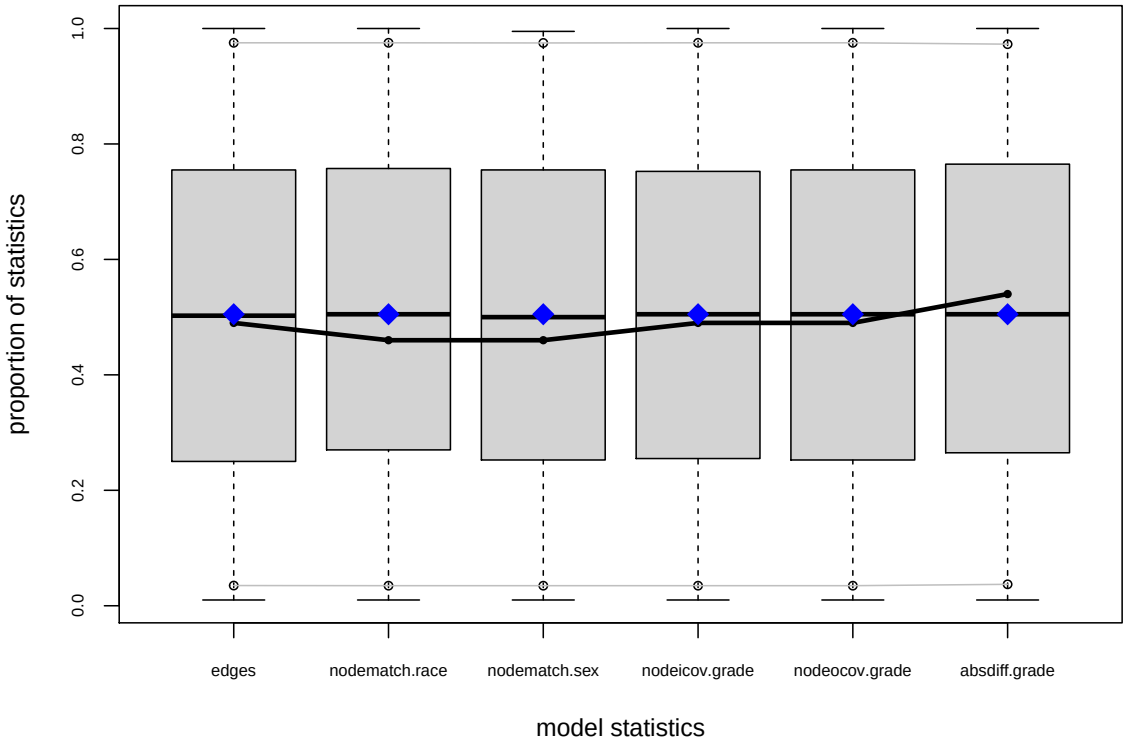
Results are similar but not exactly the same as the results we obtained from the QAP model. Two students being the same sex increases the probability of a tie forming (just like in the QAP model). Similar, as students become further apart grade-wise, the less of a chance there is for a tie to form (just like in the QAP model). The only big difference is that race is found to be statistically significant meaning students of the same race are more likely to report a friendship (unlike in QAP where it did not achieve statistical significance).

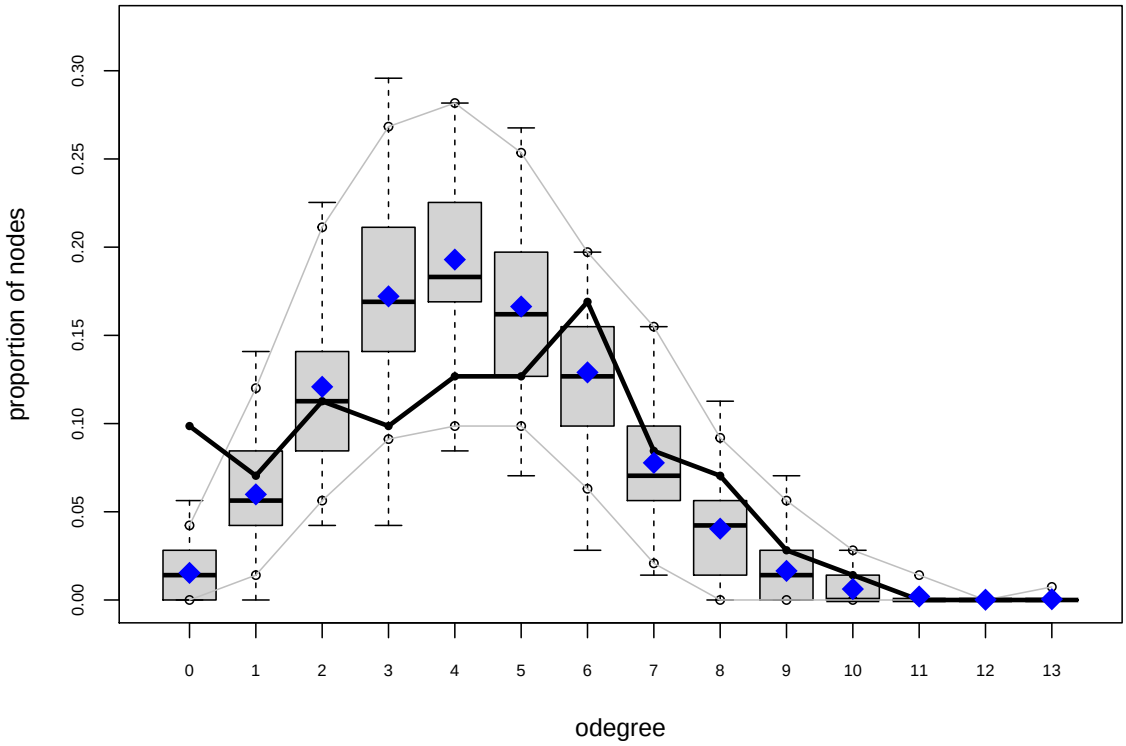
Similarly to QAP, there are no receiver or sender effects found for grade.

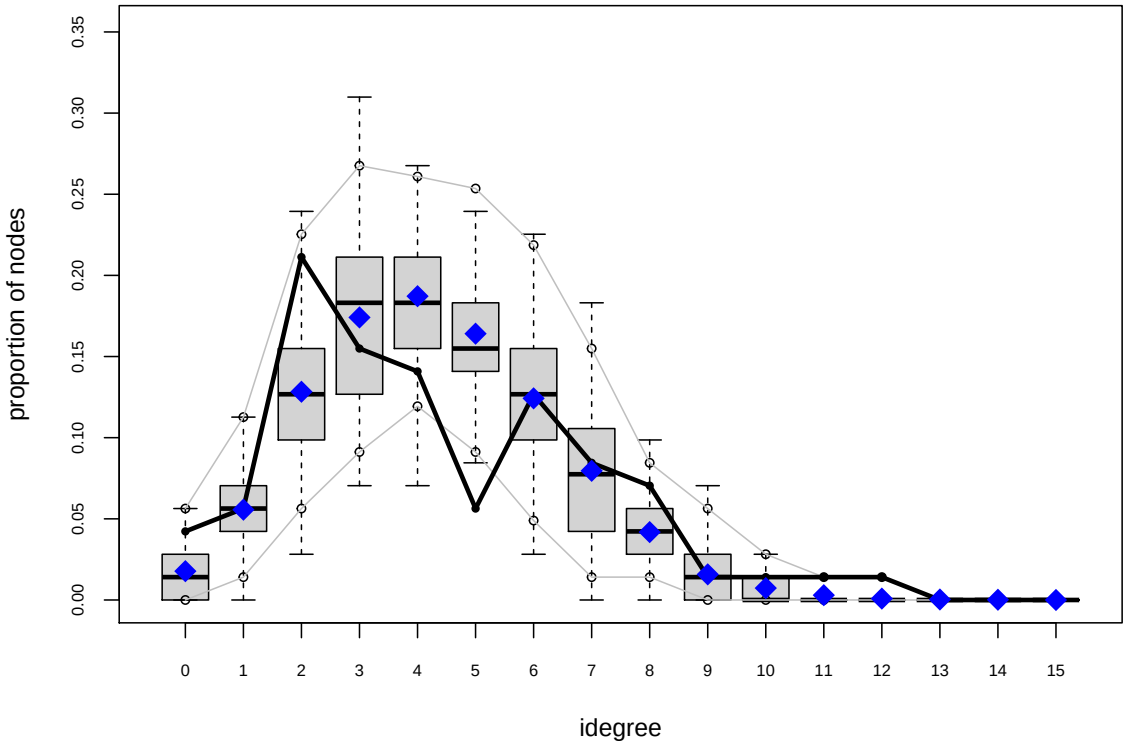
Below you can see how changes in the above mentioned variables corresponds to changes in the probability of a tie forming.

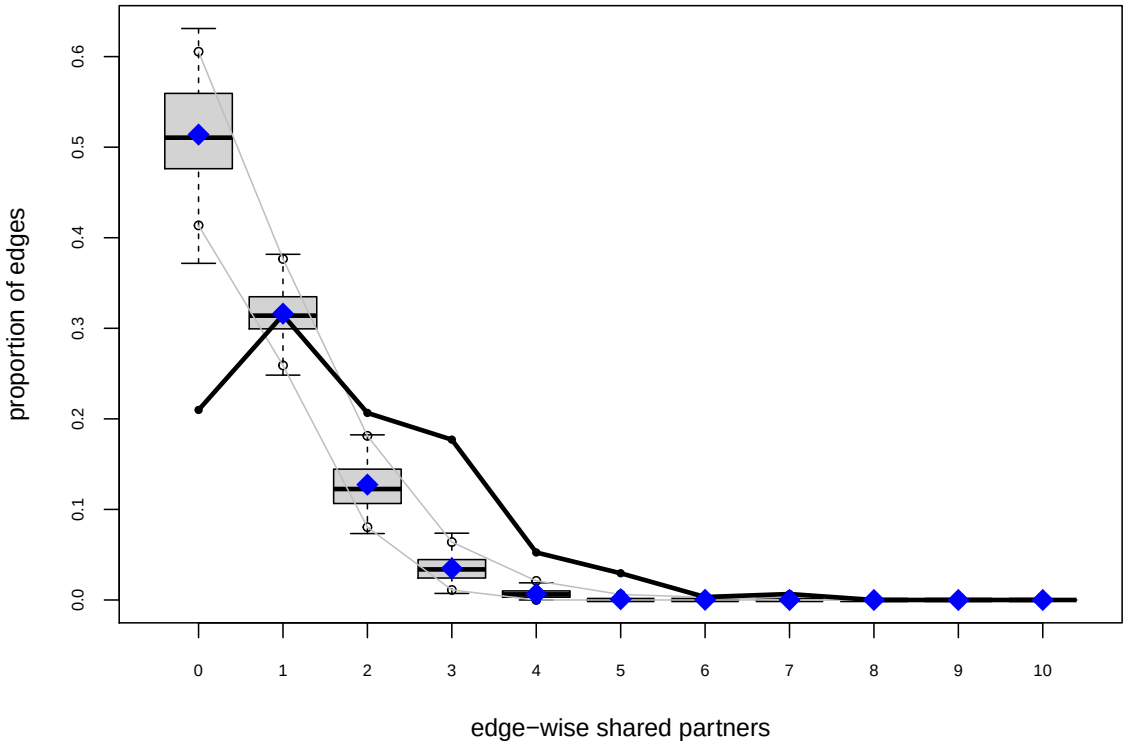
- Base probability: 8.148093%
- Being the same race: 10.9651141%
- Being the same sex: 12.2360966%
- Being one grade apart: 1.2398546%

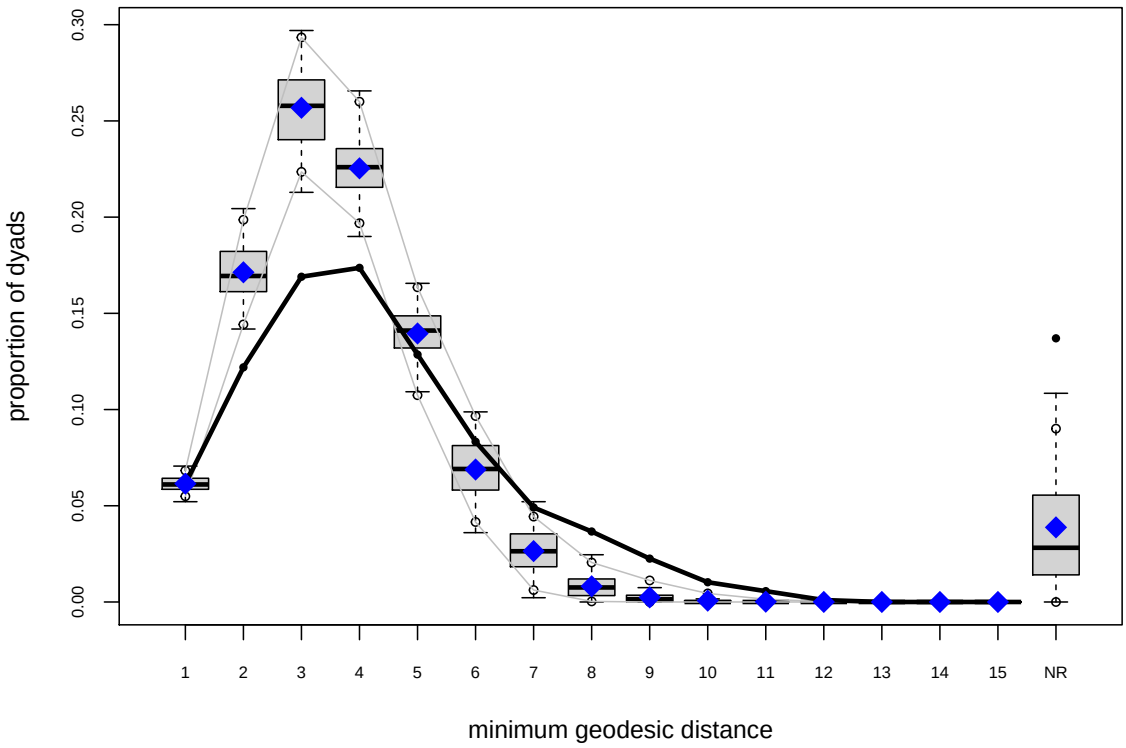
```r
# Goodness of Fit
gof_ergm_exog <- ergm::gof(ergm_exog)
plot(gof_ergm_exog)
```


proportion of statistics
0.0
0.2
0.4
0.6
0.8
1.0
edges
nodematch.race
nodematch.sex
nodeicov.grade
nodeocov.grade
absdiff.grade
model statistics


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
6
7
8
9
10
11
12
13
odegree


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
idegree


proportion of edges
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
5
6
7
8
9
10
edge−wise shared partners



We can see from the goodness of fit diagnostic plots that our model is not a particularly good fit. For many of the values, the true observed value from the data falls far outside the simulated values from our model. We can turn to using endogenous variables now to try and improve model fit.

Estimate a ERGM model using exogeneous variables and endogeneous variables

```
# Estimate an ERGM model with endogeneous and exogeneous variables
set.seed(5)
ergm_exog_endog <-
    ergm(add_health1_network ~ edges + gwidegree(1, fixed = TRUE) +
             gwodegree(1, fixed = TRUE) + mutual +
             nodematch("race") + nodematch("sex") + nodeicov("grade") +
             nodeocov("grade") + absdiff("grade"),
             control = control.ergm(MCMC.samplesize = 5000,
```

```
                                    MCMC.burnin = 10000,
                                    MCMLE.maxit = 10,
                                    parallel = 5),
              verbose = T)
```

```
Warning: 'glpk' selected as the solver, but package 'Rglpk' is not available;
falling back to 'lpSolveAPI'. This should be fine unless the sample size and/or
the number of parameters is very big.
```

```
summary(ergm_exog_endog)
```

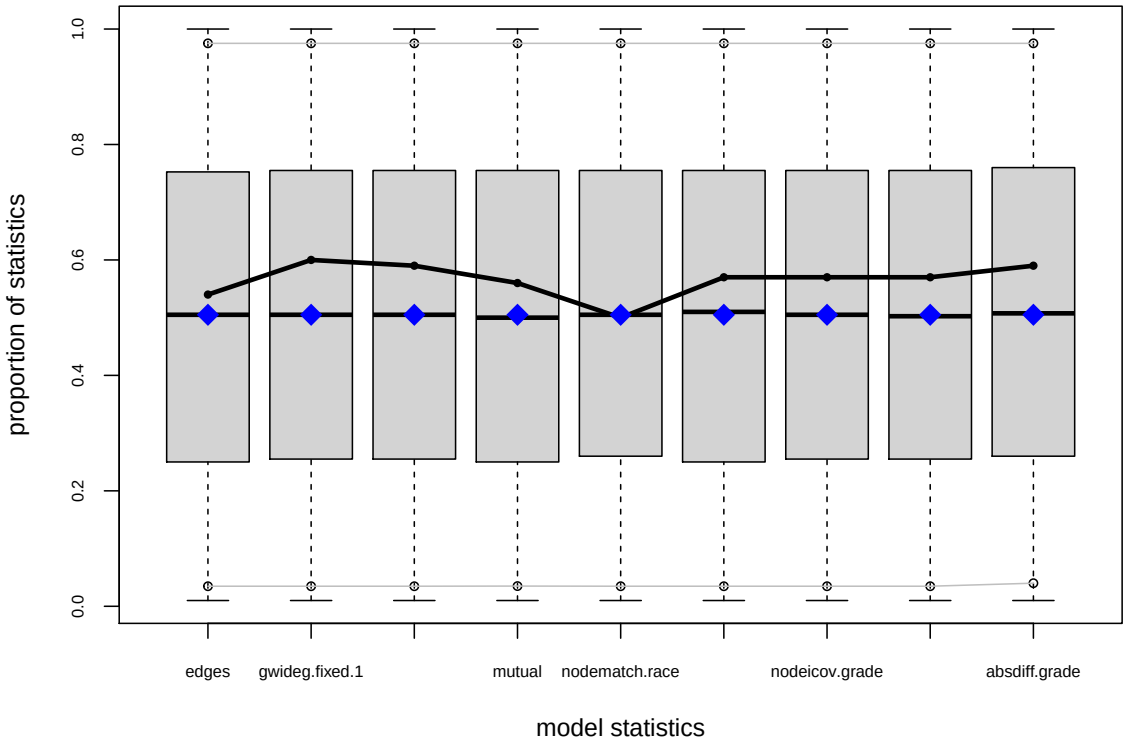
```
Call:
ergm(formula = add_health1_network ~ edges + gwidegree(1, fixed = TRUE) +
    gwodegree(1, fixed = TRUE) + mutual + nodematch("race") +
    nodematch("sex") + nodeicov("grade") + nodeocov("grade") +
    absdiff("grade"), control = control.ergm(MCMC.samplesize = 5000,
    MCMC.burnin = 10000, MCMLE.maxit = 10, parallel = 5), verbose = T)

Monte Carlo Maximum Likelihood Results:

               Estimate Std. Error MCMC % z value Pr(>|z|)
edges          -2.33557    0.26957      0  -8.664  < 1e-04 ***
gwideg.fixed.1 -0.53795    0.36837      0  -1.460 0.144193
gwodeg.fixed.1 -1.26395    0.32930      0  -3.838 0.000124 ***
mutual          2.51836    0.22968      0  10.965  < 1e-04 ***
nodematch.race  0.10591    0.12466      0   0.850 0.395570
nodematch.sex   0.31891    0.12313      0   2.590 0.009598 **
nodeicov.grade  0.14378    0.10356      0   1.388 0.165013
nodeocov.grade -0.08173    0.10237      0  -0.798 0.424635
absdiff.grade  -1.48843    0.10615      0 -14.022  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 6890  on 4970  degrees of freedom
 Residual Deviance: 1508  on 4961  degrees of freedom

AIC: 1526  BIC: 1585  (Smaller is better. MC Std. Err. = 1.391)
```

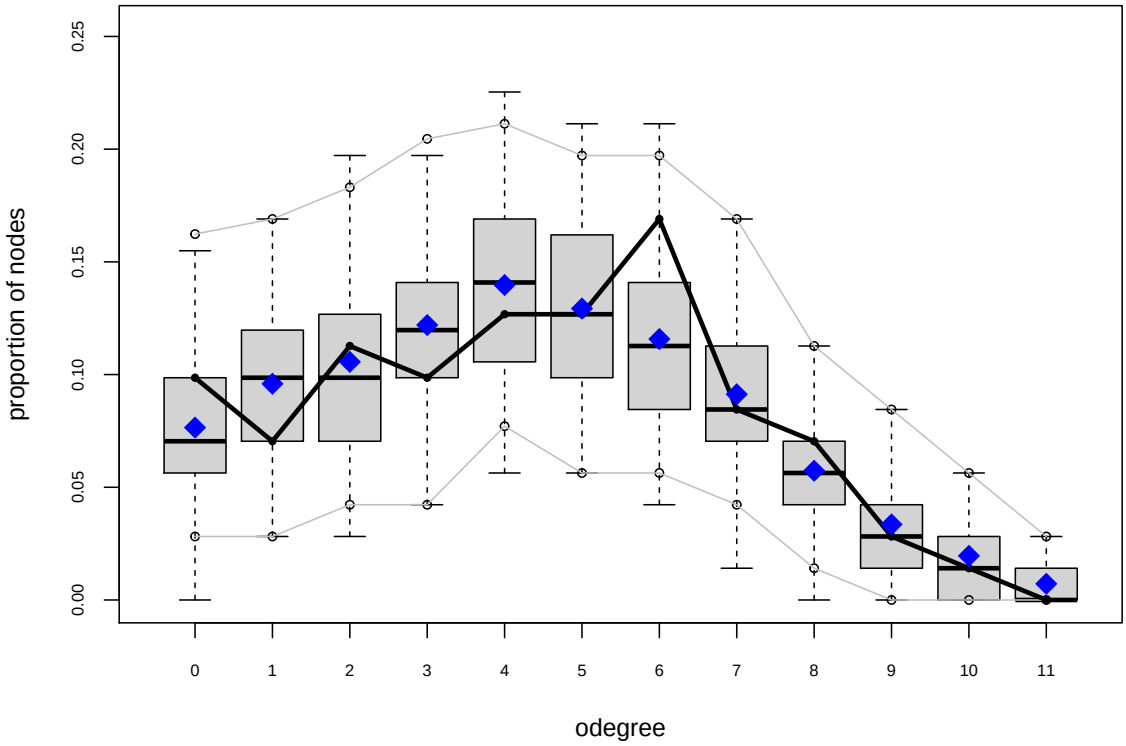
When we include some endogeneous terms, we find that the magnitude of the effect for grade and sex decreases, but they still remain significant. Race is no longer significant. We then

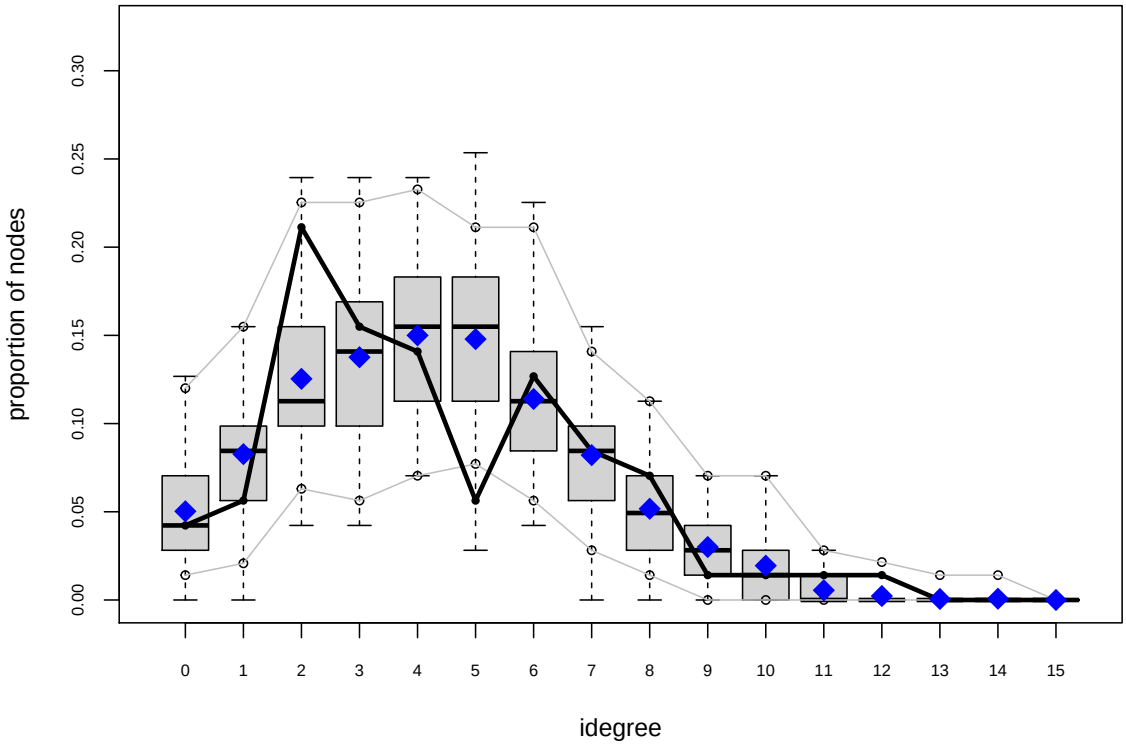
find there is a large effect for mutuality meaning that when A lists B as a friend, the chance for B to list A as a friend increases. We then also find a significant effect for the geometrically weighted out-degree indicating there are some very social students who are more likely to list others as friends than to be listed as friends themselves.

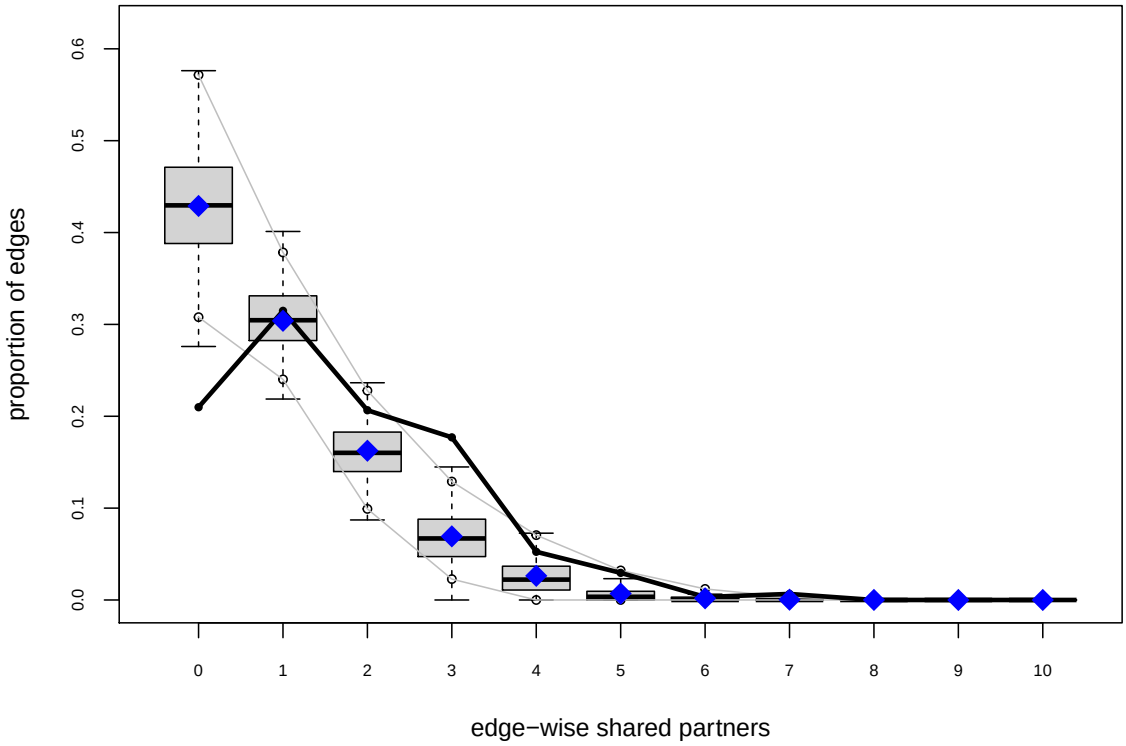
Encouragingly, the AIC and BIC scores are lower when include these endogeneous variables indicating better relative model fit.

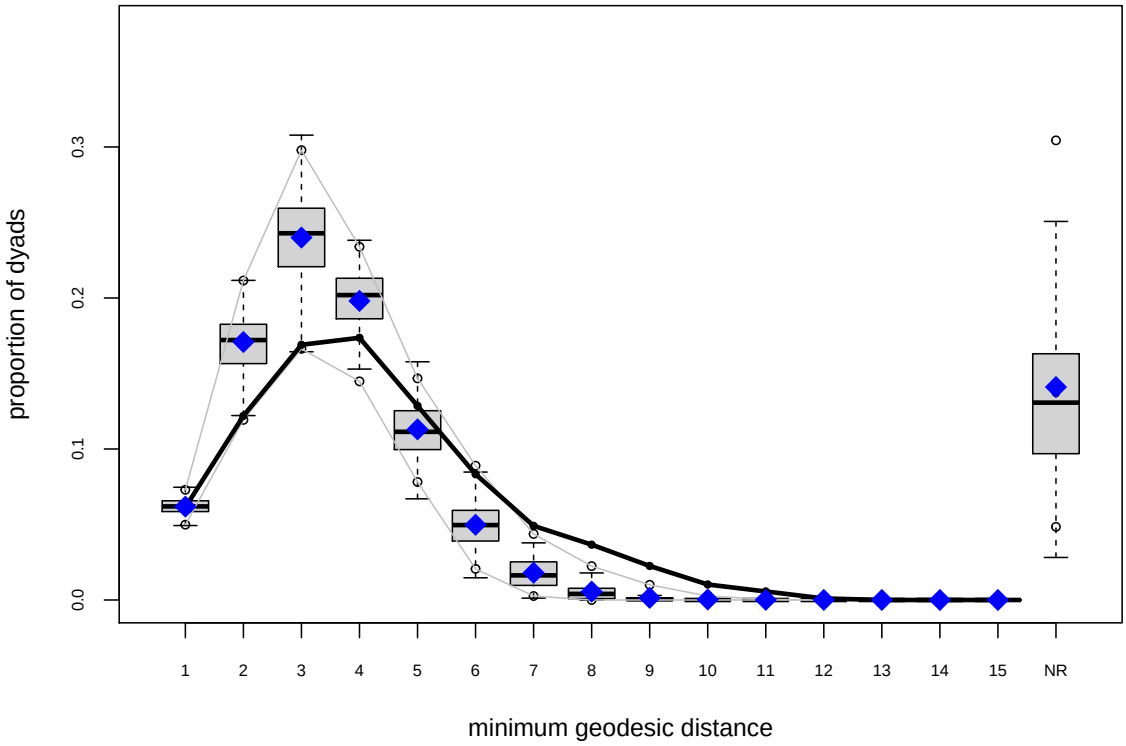
```
gof_ergm_exog_endog <- ergm::gof(ergm_exog_endog)
plot(gof_ergm_exog_endog)
```




proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0
1
2
3
4
5
6
7
8
9
10
11
odegree


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
idegree


proportion of edges
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
5
6
7
8
9
10
edge−wise shared partners



Visual inspections of the graphs indicate the model fit is still not great, but it is better when we include the endogeneous variables.

Checking ERGM model for convergence

```
mcmc.diagnostics(ergm_exog_endog)
```

```
Sample statistics summary:

Iterations = 109312:334592
Thinning interval = 128
Number of chains = 5
Sample size per chain = 1761
```

```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                    Mean      SD Naive SE Time-series SE
edges           -3.80500  31.932  0.34030         4.8209
gwideg.fixed.1  -1.13950   8.186  0.08724         0.9476
gwodeg.fixed.1  -1.43947   8.696  0.09267         1.0269
mutual          -1.25679  12.751  0.13589         2.0673
nodematch.race   0.01885  25.736  0.27427         3.3960
nodematch.sex   -2.64134  20.295  0.21628         2.5710
nodeicov.grade -42.91936 291.003  3.10122        44.7913
nodeocov.grade -42.45974 290.446  3.09528        44.5561
absdiff.grade   -0.65928  14.377  0.15321         0.7800

2. Quantiles for each variable:

                  2.5%      25%      50%     75%  97.5%
edges           -67.00  -25.000  -4.0000  18.000  60.00
gwideg.fixed.1  -19.13   -6.336  -0.3923   4.527  13.29
gwodeg.fixed.1  -19.75   -7.389  -0.5934   4.761  14.20
mutual          -26.00  -10.000  -1.0000   7.000  24.00
nodematch.race  -50.00  -18.000   0.0000  18.000  50.00
nodematch.sex   -42.00  -17.000  -3.0000  11.000  38.00
nodeicov.grade -600.00 -241.000 -50.0000 161.000 531.00
nodeocov.grade -595.90 -242.000 -49.0000 161.000 529.00
absdiff.grade   -28.00  -10.000  -1.0000   9.000  28.00


Are sample statistics significantly different from observed?
               edges gwideg.fixed.1 gwodeg.fixed.1     mutual nodematch.race
diff.      -3.8049972     -1.1395025     -1.4394723 -1.2567859    0.018852924
test stat. -1.0359782     -1.3103621     -1.5186859 -0.7598723    0.006331141
P-val.      0.3002123      0.1900734      0.1288416  0.4473309    0.994948514
           nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
diff.         -2.6413401    -42.9193640     -42.459739    -0.6592845
test stat.    -1.1040033     -1.2418566      -1.217213    -0.8770336
P-val.         0.2695917      0.2142895       0.223523     0.3804684
           Overall (Chi^2)
diff.                   NA
test stat.    5.023729e+01
P-val.        6.594887e-07

Sample statistics cross-correlations:
```

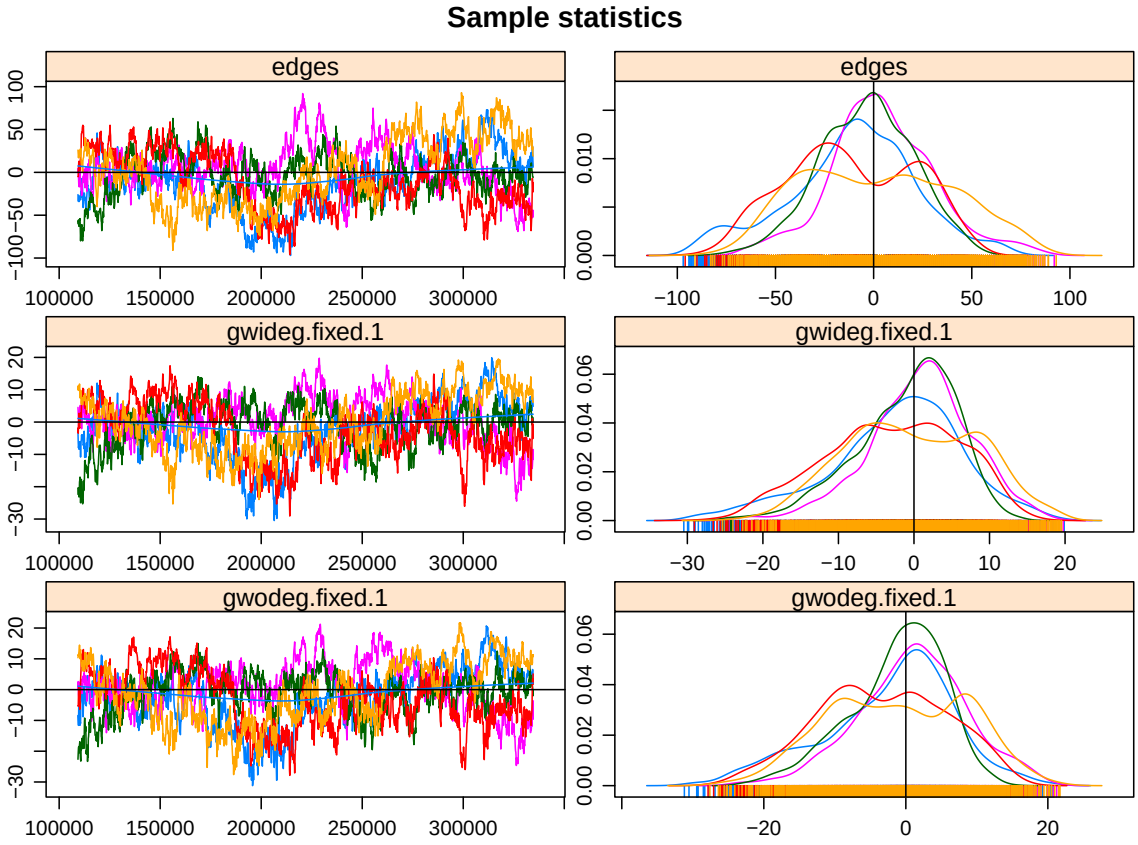
```
                    edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
edges           1.0000000      0.9119271      0.9024448 0.9278675      0.9467272
gwideg.fixed.1 0.9119271      1.0000000      0.9238588 0.8279895      0.8407235
gwodeg.fixed.1 0.9024448      0.9238588      1.0000000 0.8278448      0.8254427
mutual          0.9278675      0.8279895      0.8278448 1.0000000      0.8756010
nodematch.race 0.9467272      0.8407235      0.8254427 0.8756010      1.0000000
nodematch.sex  0.9141305      0.8380678      0.8211942 0.8588414      0.8711684
nodeicov.grade 0.9883676      0.8940845      0.8888144 0.9212100      0.9408748
nodeocov.grade 0.9879953      0.8932037      0.8879687 0.9223793      0.9406830
absdiff.grade  0.6978076      0.6352594      0.6191053 0.5500915      0.6655069
               nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
edges              0.9141305      0.9883676      0.9879953     0.6978076
gwideg.fixed.1     0.8380678      0.8940845      0.8932037     0.6352594
gwodeg.fixed.1     0.8211942      0.8888144      0.8879687     0.6191053
mutual             0.8588414      0.9212100      0.9223793     0.5500915
nodematch.race     0.8711684      0.9408748      0.9406830     0.6655069
nodematch.sex      1.0000000      0.9104773      0.9101801     0.6353229
nodeicov.grade     0.9104773      1.0000000      0.9994067     0.6899036
nodeocov.grade     0.9101801      0.9994067      1.0000000     0.6847970
absdiff.grade      0.6353229      0.6899036      0.6847970     1.0000000

Sample statistics auto-correlation:
Chain 1
            edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
Lag 0   1.0000000      1.0000000      1.0000000 1.0000000      1.0000000
Lag 128 0.9816258      0.9725065      0.9753508 0.9907904      0.9756821
Lag 256 0.9643445      0.9519607      0.9545320 0.9820874      0.9539778
Lag 384 0.9492683      0.9331560      0.9361050 0.9740696      0.9367172
Lag 512 0.9355673      0.9181560      0.9211854 0.9667464      0.9197397
Lag 640 0.9224212      0.9054204      0.9068404 0.9605228      0.9048303
        nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
Lag 0       1.0000000      1.0000000      1.0000000     1.0000000
Lag 128     0.9760955      0.9816178      0.9818728     0.9050568
Lag 256     0.9544585      0.9645568      0.9648621     0.8226040
Lag 384     0.9355196      0.9500315      0.9504544     0.7566731
Lag 512     0.9177720      0.9369561      0.9372385     0.6969872
Lag 640     0.9023645      0.9244591      0.9245948     0.6442677
Chain 2
            edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
Lag 0   1.0000000      1.0000000      1.0000000 1.0000000      1.0000000
Lag 128 0.9698507      0.9592307      0.9681479 0.9801564      0.9691166
Lag 256 0.9429905      0.9226188      0.9413985 0.9612402      0.9421306
Lag 384 0.9185164      0.8919116      0.9151655 0.9417820      0.9180304
```

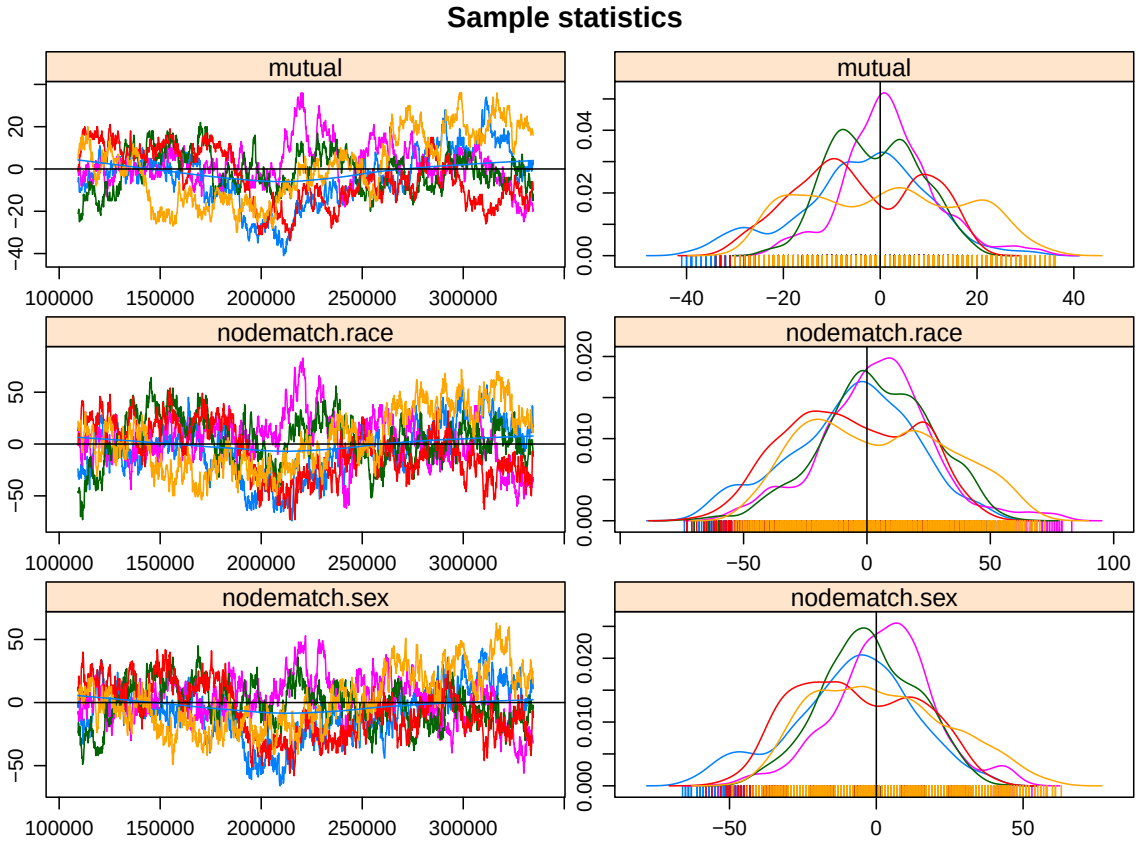
```
Lag 512 0.8963963      0.8675620      0.8902795 0.9237558      0.8952666
Lag 640 0.8755639      0.8452570      0.8707210 0.9071124      0.8717841
        nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
Lag 0       1.0000000      1.0000000      1.0000000     1.0000000
Lag 128     0.9630675      0.9716448      0.9723152     0.8970959
Lag 256     0.9284736      0.9463364      0.9478913     0.8100094
Lag 384     0.8992339      0.9231534      0.9254420     0.7303793
Lag 512     0.8725844      0.9027576      0.9051998     0.6672747
Lag 640     0.8487510      0.8834107      0.8858742     0.6200315
Chain 3
            edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
Lag 0   1.0000000      1.0000000      1.0000000 1.0000000      1.0000000
Lag 128 0.9620783      0.9524787      0.9492928 0.9768585      0.9671449
Lag 256 0.9326309      0.9163647      0.9073302 0.9547548      0.9398208
Lag 384 0.9055252      0.8853221      0.8691115 0.9330066      0.9159545
Lag 512 0.8809712      0.8580064      0.8364030 0.9139105      0.8946500
Lag 640 0.8558148      0.8305439      0.8076815 0.8925325      0.8717163
        nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
Lag 0       1.0000000      1.0000000      1.0000000     1.0000000
Lag 128     0.9585537      0.9609707      0.9614028     0.8914845
Lag 256     0.9250769      0.9305277      0.9310607     0.8115565
Lag 384     0.8993835      0.9026642      0.9037492     0.7469980
Lag 512     0.8776809      0.8774968      0.8795542     0.6905156
Lag 640     0.8548149      0.8520927      0.8545173     0.6326987
Chain 4
            edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
Lag 0   1.0000000      1.0000000      1.0000000 1.0000000      1.0000000
Lag 128 0.9808446      0.9707457      0.9720871 0.9908084      0.9768103
Lag 256 0.9640867      0.9469797      0.9478347 0.9823154      0.9556811
Lag 384 0.9488621      0.9258466      0.9250181 0.9741712      0.9371971
Lag 512 0.9329080      0.9051201      0.9038959 0.9662964      0.9193536
Lag 640 0.9187027      0.8871599      0.8874675 0.9592586      0.9023926
        nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
Lag 0       1.0000000      1.0000000      1.0000000     1.0000000
Lag 128     0.9760103      0.9813142      0.9816865     0.8969718
Lag 256     0.9552477      0.9652534      0.9658473     0.8081335
Lag 384     0.9366377      0.9507240      0.9516012     0.7348346
Lag 512     0.9184415      0.9357440      0.9368807     0.6653157
Lag 640     0.9033574      0.9228192      0.9240787     0.6081933
Chain 5
            edges gwideg.fixed.1 gwodeg.fixed.1    mutual nodematch.race
Lag 0   1.0000000      1.0000000      1.0000000 1.0000000      1.0000000
Lag 128 0.9854553      0.9733955      0.9778098 0.9933547      0.9807885
```

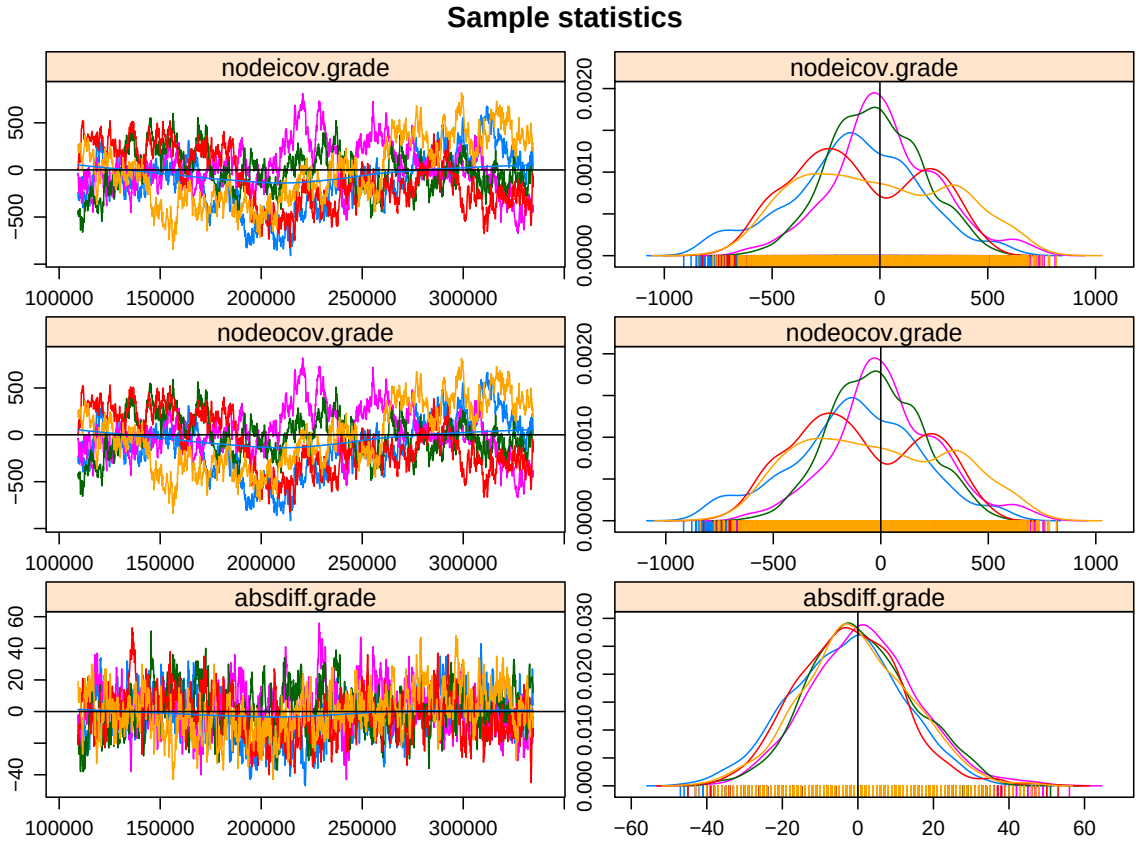
```
Lag 256 0.9729018      0.9530455      0.9595672 0.9867009      0.9656587
Lag 384 0.9619917      0.9358599      0.9437866 0.9803763      0.9526078
Lag 512 0.9518778      0.9195782      0.9303384 0.9745134      0.9400977
Lag 640 0.9419512      0.9050212      0.9175184 0.9692711      0.9287331
        nodematch.sex nodeicov.grade nodeocov.grade absdiff.grade
Lag 0       1.0000000      1.0000000      1.0000000     1.0000000
Lag 128     0.9802136      0.9854009      0.9857482     0.8950341
Lag 256     0.9627886      0.9727014      0.9732632     0.8098697
Lag 384     0.9461977      0.9615199      0.9621778     0.7380764
Lag 512     0.9313316      0.9511299      0.9518016     0.6691608
Lag 640     0.9165707      0.9411574      0.9419649     0.6083680

Sample statistics burn-in diagnostic (Geweke):
Chain 1

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
       -0.9191        -0.9279        -1.3740        -0.6206        -1.6175
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
        0.1979        -1.0459        -1.0563        -0.4835

Individual P-values (lower = worse):
         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
     0.3580418      0.3534732      0.1694550      0.5348339      0.1057606
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
     0.8431496      0.2956183      0.2908237      0.6287549
Joint P-value (lower = worse):  0.003009469 .
Chain 2

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
     -0.509161      -0.453387      -0.388303      -1.053377       0.006585
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
     -0.134708      -1.432616      -1.422088       0.659862

Individual P-values (lower = worse):
         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
     0.6106396      0.6502701      0.6977917      0.2921682      0.9947459
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
```

```
     0.8928426      0.1519675      0.1550006      0.5093421
Joint P-value (lower = worse):  0.0005326626 .
Chain 3

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
        -4.121         -4.314         -5.241         -3.994         -2.354
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
        -1.699         -2.829         -2.888         -3.065

Individual P-values (lower = worse):
         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
  3.772123e-05   1.601982e-05   1.598290e-07   6.488630e-05   1.855970e-02
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
  8.931137e-02   4.670481e-03   3.881111e-03   2.179551e-03
Joint P-value (lower = worse):  0.006644012 .
Chain 4

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
        7.7453         4.8735         4.9666         7.9581         7.9398
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
        8.3294         9.7047         9.7583         0.9509

Individual P-values (lower = worse):
         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
  9.532126e-15   1.096566e-06   6.812769e-07   1.747673e-15   2.025444e-15
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
  8.126987e-17   2.878751e-22   1.700243e-22   3.416526e-01
Joint P-value (lower = worse):  3.447474e-10 .
Chain 5

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

         edges gwideg.fixed.1 gwodeg.fixed.1         mutual nodematch.race
       -1.7121        -0.6682        -0.2186        -1.1489        -2.0453
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
       -2.2868        -1.3500        -1.3429        -1.0518
```

```
Individual P-values (lower = worse):
         edges gwideg.fixed.1 gwodeg.fixed.1          mutual nodematch.race
    0.08688042     0.50399110     0.82697861     0.25061568     0.04082325
 nodematch.sex nodeicov.grade nodeocov.grade  absdiff.grade
    0.02220781     0.17701152     0.17929863     0.29289715
Joint P-value (lower = worse):  0.06430935 .
```




Sample statistics
mutual
mutual
nodematch.race
nodematch.race
nodematch.sex
nodematch.sex
100000
150000
200000
250000
300000
−40
−20
0
20
40
−50
0
50
100
0.00
0.02
0.04
0.000
0.010
0.020




```
MCMC diagnostics shown here are from the last round of simulation, prior to computation of fi
```

The trace plots do not exhibit any trend, and the density plots approximate normal distributions indicating degeneracy is not really a problem. My apologies because this output takes up a lot of pages.

Estimate LSM models

```
set.seed(42)

lsm <-
    ergmm(add_health1_network ~ nodematch("race") + nodematch("sex") +
              nodeicov("grade") + nodeocov("grade") + absdiff("grade") +
              euclidean(d = 2),
```

```
            control = control.ergmm(burnin = 25000),
            verbose = TRUE)
```

```
Generating initial values for MCMC:
Computing geodesic distances... Finished.
Computing MDS locations... Finished.
Computing other initial values... Finished.
Finding the conditional posterior mode... Finished.
Burning in... Backing off: too few acceptances. If you see this message several times in a r
Finished.
Starting sampling run... Finished.
Post-processing the MCMC output:
Fitting the MKL locations... Finished.
MKL MBC is not available or non-latent-cluster model.
Performing Procrustes transformation... Finished.
```

```
lsm_bi <-
    ergmm(add_health1_network ~ nodematch("race") + nodematch("sex") +
              nodeicov("grade") + nodeocov("grade") + absdiff("grade") +
              bilinear(d = 2),
            control = control.ergmm(burnin = 25000),
            verbose = TRUE)
```

```
Generating initial values for MCMC:
Computing geodesic distances... Finished.
Computing MDS locations... Finished.
Computing other initial values... Finished.
Finding the conditional posterior mode... Finished.
Burning in... Backing off: too few acceptances. If you see this message several times in a r
Finished.
Starting sampling run... Finished.
Post-processing the MCMC output:
Fitting the MKL locations... Finished.
MKL MBC is not available or non-latent-cluster model.
Performing Procrustes transformation... Finished.
```

```
summary(lsm)
```

```
NOTE: It is not certain whether it is appropriate to use latentnet's BIC to select latent spa


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

Formula:   add_health1_network ~ nodematch("race") + nodematch("sex") +
    nodeicov("grade") + nodeocov("grade") + absdiff("grade") +
    euclidean(d = 2)
Attribute: edges
Model:     Bernoulli
MCMC sample of size 4000, draws are 10 iterations apart, after burnin of 25000 iterations.
Covariate coefficients posterior means:
                Estimate      2.5%   97.5% 2*min(Pr(>0),Pr(<0))
(Intercept)     0.566256 -1.494813  2.5429               0.5460
nodematch.race  0.283577 -0.205607  0.7489               0.2630
nodematch.sex   0.443919 -0.025298  0.8405               0.0635 .
nodeicov.grade  0.179637 -0.026546  0.4382               0.1070
nodeocov.grade -0.028586 -0.243666  0.1639               0.8100
absdiff.grade  -2.182305 -2.564862 -1.8568               <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Overall BIC:        1711.732
Likelihood BIC:     1161.277
Latent space/clustering BIC:     550.455

Covariate coefficients MKL:
                  Estimate
(Intercept)    -0.58579137
nodematch.race  0.25805446
nodematch.sex   0.40411069
nodeicov.grade -0.02334153
nodeocov.grade  0.16622548
absdiff.grade  -1.94925393
```

```
summary(lsm_bi)
```

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

Formula:   add_health1_network ~ nodematch("race") + nodematch("sex") +
    nodeicov("grade") + nodeocov("grade") + absdiff("grade") +
    bilinear(d = 2)
Attribute: edges
Model:     Bernoulli
MCMC sample of size 4000, draws are 10 iterations apart, after burnin of 25000 iterations.
Covariate coefficients posterior means:
                Estimate      2.5%   97.5% 2*min(Pr(>0),Pr(<0))
(Intercept)    -2.802115 -4.327932 -1.1573               <2e-16 ***
nodematch.race -0.140693 -0.655378  0.3768               0.5850
nodematch.sex   0.724935  0.364024  1.0749               <2e-16 ***
nodeicov.grade  0.154760 -0.043969  0.3714               0.1325
nodeocov.grade -0.066018 -0.286787  0.1290               0.5685
absdiff.grade  -2.457110 -2.795759 -2.1501               <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Overall BIC:        1555.701
Likelihood BIC:     1120.499
Latent space/clustering BIC:     435.2022

Covariate coefficients MKL:
                  Estimate
(Intercept)    -2.66108951
nodematch.race -0.09194354
nodematch.sex   0.64627941
nodeicov.grade -0.05172181
nodeocov.grade  0.14674653
absdiff.grade  -2.13883580
```

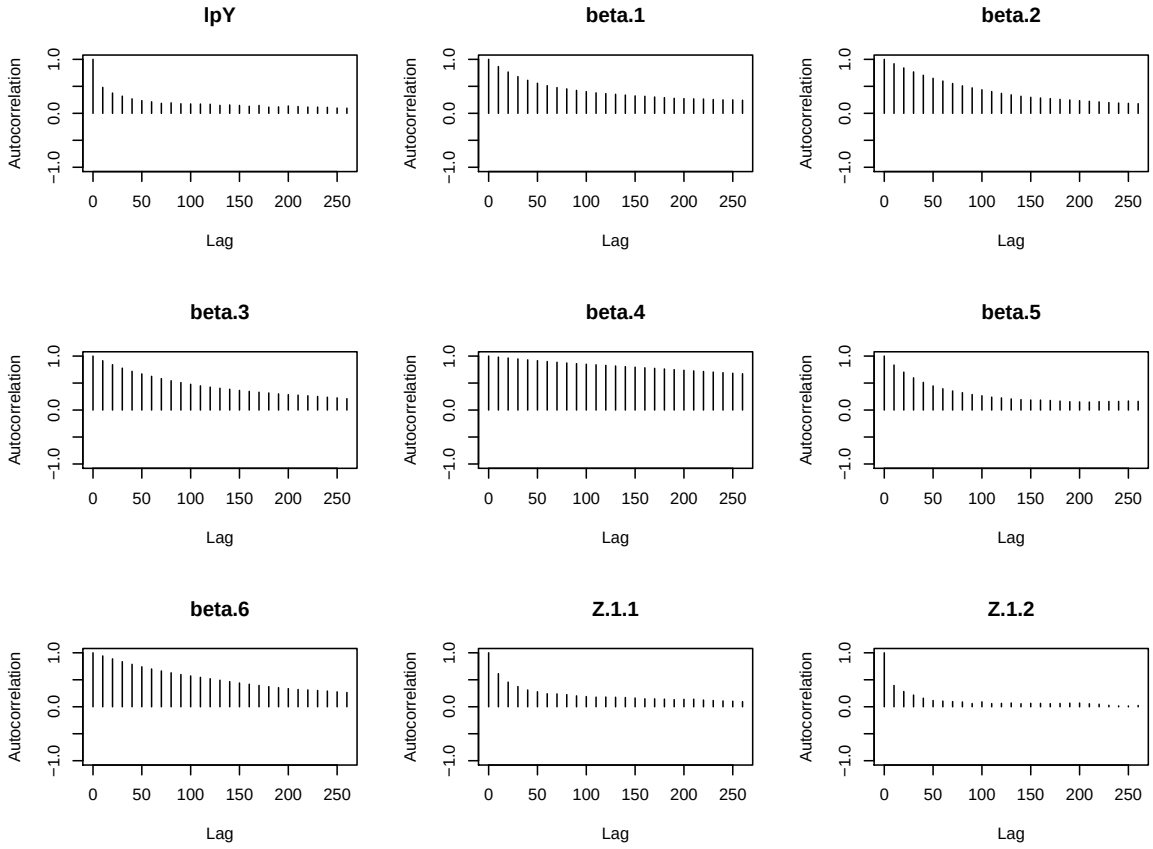
Using a Euclidean distance function actually produces a worse-fitting model as indicated by BIC. Similarly to the ERGM models (and QAP model before that), the LSM models find grade difference significant and same sex significant (although for the Euclidean LSM model, it is only marginally significant).

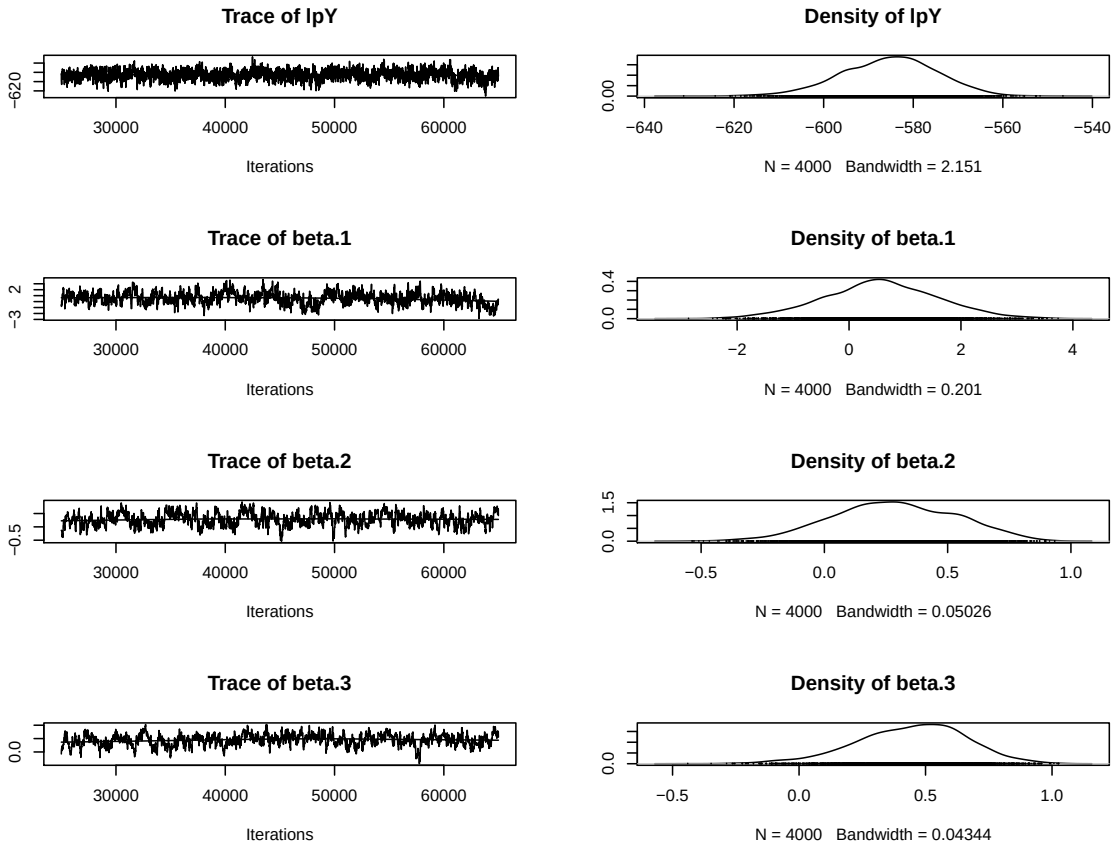
Checking LSM models for convergence

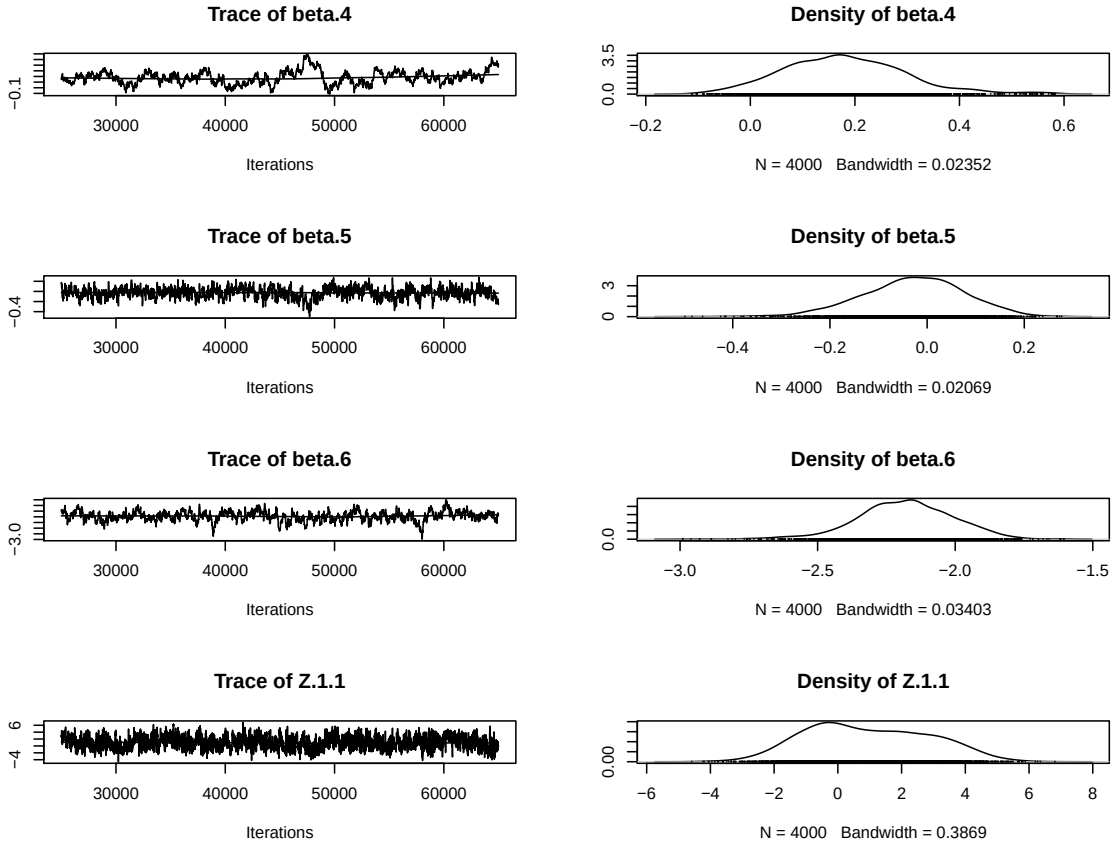
```
mcmc.diagnostics(lsm)
```

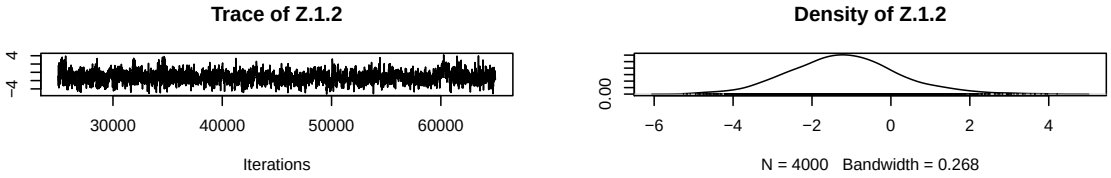
```
Chain 1
Lag 0
               lpY      beta.1       beta.2       beta.3      beta.4
lpY     1.000000000  0.22322878 -0.008229500 -0.023310197 -0.12146916
beta.1  0.223228781  1.00000000 -0.028720401 -0.220703356 -0.52875320
beta.2 -0.008229500 -0.02872040  1.000000000  0.158069641 -0.16018079
beta.3 -0.023310197 -0.22070336  0.158069641  1.000000000  0.05705832
beta.4 -0.121469164 -0.52875320 -0.160180790  0.057058315  1.00000000
beta.5 -0.009614386 -0.40480879  0.001758967  0.009940360 -0.51076718
beta.6  0.053200605  0.07556126 -0.056004633  0.011751210 -0.24052697
Z.1.1   0.088843948  0.14151345 -0.099203865 -0.173761347 -0.13803894
Z.1.2  -0.028624486  0.03096330  0.032071733  0.006195096 -0.03267460
            beta.5      beta.6        Z.1.1        Z.1.2
lpY    -0.009614386  0.05320061  0.088843948 -0.028624486
beta.1 -0.404808785  0.07556126  0.141513448  0.030963298
beta.2  0.001758967 -0.05600463 -0.099203865  0.032071733
beta.3  0.009940360  0.01175121 -0.173761347  0.006195096
beta.4 -0.510767176 -0.24052697 -0.138038944 -0.032674598
beta.5  1.000000000  0.06407984  0.047315283 -0.027486630
beta.6  0.064079836  1.00000000  0.072294603  0.147847678
Z.1.1   0.047315283  0.07229460  1.000000000 -0.003822537
Z.1.2  -0.027486630  0.14784768 -0.003822537  1.000000000

Lag 10
               lpY      beta.1        beta.2       beta.3      beta.4
lpY     0.480148632  0.22526690 -5.979782e-03 -0.024573302 -0.11768531
beta.1  0.211786753  0.86649070 -2.689395e-02 -0.198875352 -0.52660776
beta.2 -0.002727471 -0.02533591  9.179932e-01  0.145341570 -0.14499917
beta.3 -0.005885052 -0.20301523  1.584587e-01  0.913376662  0.06451536
beta.4 -0.119417134 -0.52987291 -1.466669e-01  0.066862292  0.98112531
beta.5 -0.010769790 -0.26856844  8.270029e-05 -0.008401146 -0.49243678
beta.6  0.054623890  0.08210530 -4.858887e-02 -0.000646172 -0.23093044
Z.1.1   0.068511987  0.13725658 -9.693328e-02 -0.171810645 -0.13995749
Z.1.2  -0.026043958  0.03102050  3.285459e-02  0.004096236 -0.02742299
              beta.5      beta.6       Z.1.1        Z.1.2
lpY    -0.0232455301  0.06158260  0.07132310 -0.027100534
beta.1 -0.2750143423  0.08682970  0.14839850  0.014080137
```

```
beta.2  0.0009037417 -0.04961690 -0.10419524  0.038577873
beta.3 -0.0039588153  0.01016470 -0.16336743  0.003671255
beta.4 -0.4899796081 -0.23348850 -0.13505575 -0.036401030
beta.5  0.8337391446  0.04965569  0.03552005 -0.004189314
beta.6  0.0517233439  0.94170103  0.06963708  0.147620146
Z.1.1   0.0552486137  0.06551258  0.61267258  0.011276096
Z.1.2  -0.0304732192  0.14213253  0.01019989  0.394739303
```




Trace of lpY
−620
30000
40000
50000
60000
Iterations
Density of lpY
0.00
−640
−620
−600
−580
−560
−540
N = 4000 Bandwidth = 2.151
Trace of beta.1
−3
2
30000
40000
50000
60000
Iterations
Density of beta.1
0.0
0.4
−2
0
2
4
N = 4000 Bandwidth = 0.201
Trace of beta.2
−0.5
30000
40000
50000
60000
Iterations
Density of beta.2
0.0
1.5
−0.5
0.0
0.5
1.0
N = 4000 Bandwidth = 0.05026
Trace of beta.3
0.0
30000
40000
50000
60000
Iterations
Density of beta.3
0.0
−0.5
0.0
0.5
1.0
N = 4000 Bandwidth = 0.04344


Trace of beta.4
−0.1
30000
40000
50000
60000
Iterations
Density of beta.4
0.0
3.5
−0.2
0.0
0.2
0.4
0.6
N = 4000 Bandwidth = 0.02352
Trace of beta.5
−0.4
30000
40000
50000
60000
Iterations
Density of beta.5
0
3
−0.4
−0.2
0.0
0.2
N = 4000 Bandwidth = 0.02069
Trace of beta.6
−3.0
30000
40000
50000
60000
Iterations
Density of beta.6
0.0
−3.0
−2.5
−2.0
−1.5
N = 4000 Bandwidth = 0.03403
Trace of Z.1.1
−4
6
30000
40000
50000
60000
Iterations
Density of Z.1.1
0.00
−6
−4
−2
0
2
4
6
8
N = 4000 Bandwidth = 0.3869

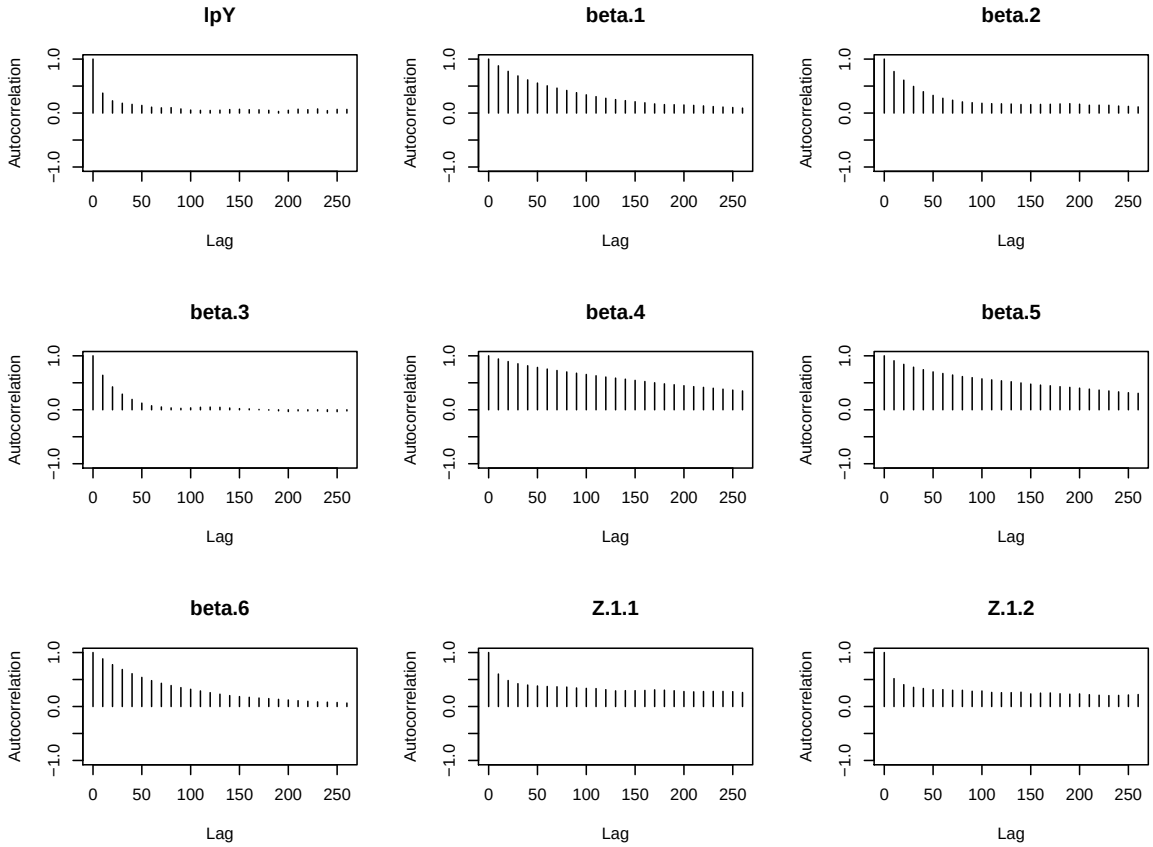


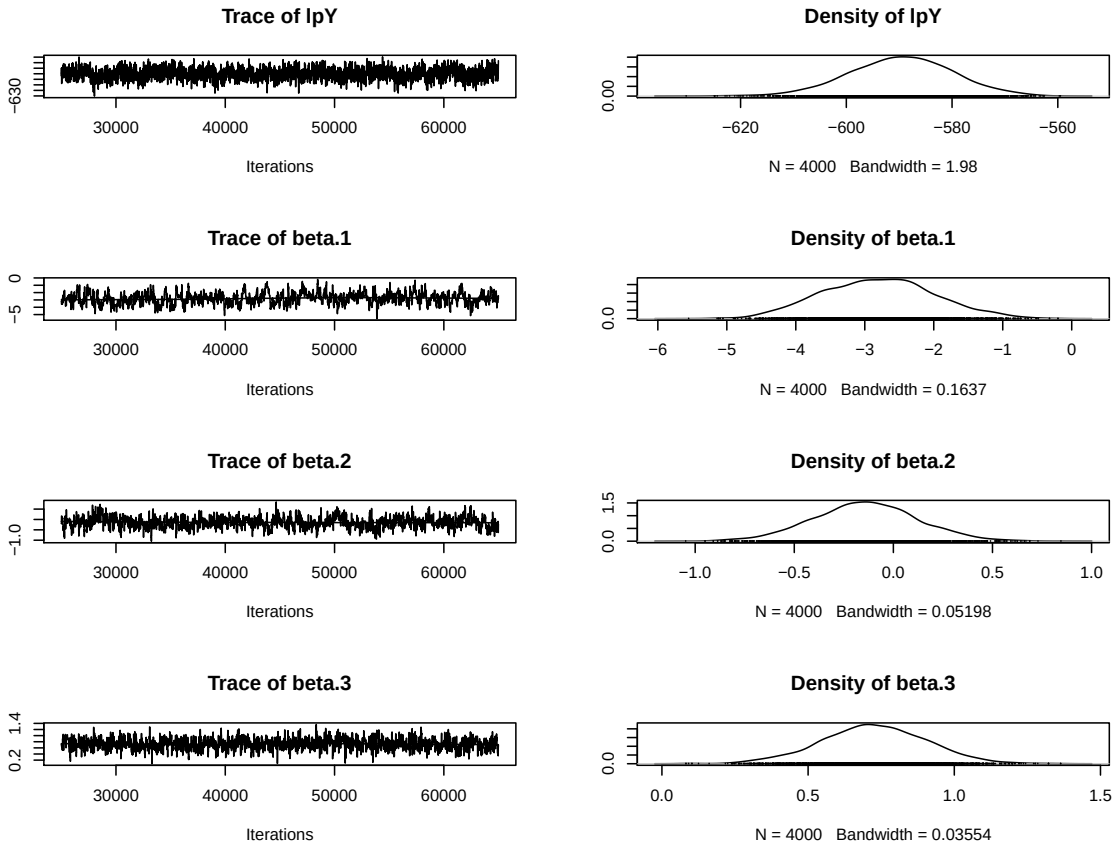

```
[[1]]

Quantile (q) = 0.025
Accuracy (r) = +/- 0.0125
Probability (s) = 0.95

        Burn-in  Total  Lower bound  Dependence
        (M)      (N)    (Nmin)       factor (I)
 lpY    80       16800  600           28.0
 beta.1 180      31370  600           52.3
 beta.2 230      40020  600           66.7
 beta.3 320      53140  600           88.6
 beta.4 520      87160  600          145.0
 beta.5 180      30920  600           51.5
 beta.6 600      105540 600          176.0
 Z.1.1  40       8480   600           14.1
 Z.1.2  30       6890   600           11.5
```

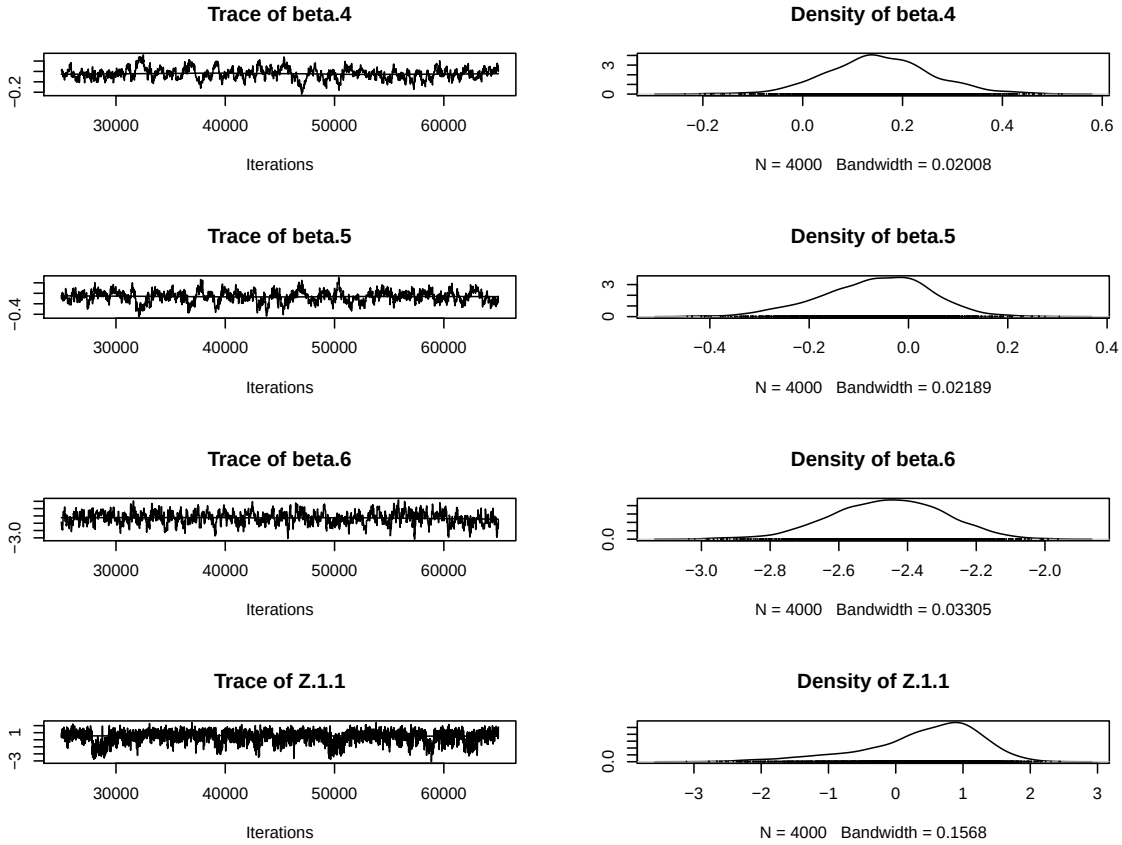
```
mcmc.diagnostics(lsm_bi)
```

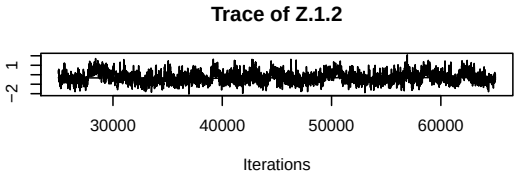
```
Chain 1
Lag 0
               lpY      beta.1      beta.2      beta.3      beta.4
lpY     1.000000000 -0.02341884 -0.20138905  0.01193067  0.01697528
beta.1 -0.023418837  1.00000000 -0.24488048 -0.02465613 -0.36336250
beta.2 -0.201389053 -0.24488048  1.00000000  0.05326397  0.02206501
beta.3  0.011930668 -0.02465613  0.05326397  1.00000000 -0.02757523
beta.4  0.016975281 -0.36336250  0.02206501 -0.02757523  1.00000000
beta.5  0.008948784 -0.40771679  0.00934092 -0.06468162 -0.67476324
beta.6 -0.113476343 -0.10432874 -0.03285337 -0.09616742 -0.10077160
Z.1.1   0.178089919  0.13017375 -0.23244025 -0.01694841  0.04255201
Z.1.2  -0.190201076 -0.18810366  0.23155010  0.06731602  0.01945780
             beta.5       beta.6        Z.1.1       Z.1.2
lpY     0.008948784 -0.113476343  0.178089919 -0.19020108
beta.1 -0.407716790 -0.104328742  0.130173754 -0.18810366
beta.2  0.009340920 -0.032853372 -0.232440253  0.23155010
beta.3 -0.064681616 -0.096167416 -0.016948414  0.06731602
beta.4 -0.674763245 -0.100771603  0.042552015  0.01945780
beta.5  1.000000000  0.152744523 -0.123722176  0.11225768
beta.6  0.152744523  1.000000000 -0.007513983 -0.03177947
Z.1.1  -0.123722176 -0.007513983  1.000000000 -0.57988502
Z.1.2   0.112257680 -0.031779472 -0.579885018  1.00000000

Lag 10
               lpY       beta.1      beta.2      beta.3      beta.4
lpY     0.369384552 -0.028401171 -0.18825447  0.01757010  0.01272128
beta.1 -0.016189043  0.875065900 -0.23726843  0.00595310 -0.31302839
beta.2 -0.199430324 -0.235012805  0.76942835  0.06339224 -0.01272795
beta.3  0.026659181  0.001497511  0.06492508  0.63907953 -0.04890382
beta.4  0.011880030 -0.311182752 -0.01141777 -0.04302636  0.94176459
beta.5  0.007483299 -0.361343233  0.07607145 -0.04072829 -0.64420385
beta.6 -0.098162945 -0.090868590 -0.02477201 -0.09504016 -0.09790603
Z.1.1   0.166075963  0.124247838 -0.24042321 -0.02582807  0.03927191
Z.1.2  -0.175943111 -0.184214850  0.23489234  0.06013045  0.01771264
            beta.5       beta.6         Z.1.1       Z.1.2
lpY     0.01729042 -0.110325097  0.1464100808 -0.13762271
beta.1 -0.36034382 -0.102105514  0.1305076477 -0.18474598
beta.2  0.07424965 -0.022610202 -0.2253409899  0.23258072
beta.3 -0.03200864 -0.086357506 -0.0263281268  0.08378709
```

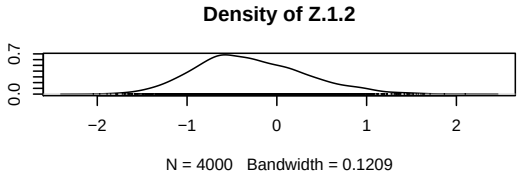
```
beta.4 -0.64639802 -0.095647544  0.0399905424  0.01941515
beta.5  0.90790163  0.147940557 -0.1210062911  0.10735831
beta.6  0.14457295  0.883304270 -0.0007903386 -0.04101788
Z.1.1  -0.11286250  0.005557362  0.6015644459 -0.44692269
Z.1.2   0.11060797 -0.030188332 -0.4627952766  0.51669073
```




Trace of lpY
−630
30000
40000
50000
60000
Iterations
Density of lpY
0.00
−620
−600
−580
−560
N = 4000 Bandwidth = 1.98
Trace of beta.1
−5
0
30000
40000
50000
60000
Iterations
Density of beta.1
0.0
−6
−5
−4
−3
−2
−1
0
N = 4000 Bandwidth = 0.1637
Trace of beta.2
−1.0
30000
40000
50000
60000
Iterations
Density of beta.2
0.0
1.5
−1.0
−0.5
0.0
0.5
1.0
N = 4000 Bandwidth = 0.05198
Trace of beta.3
0.2
1.4
30000
40000
50000
60000
Iterations
Density of beta.3
0.0
0.0
0.5
1.0
1.5
N = 4000 Bandwidth = 0.03554


Trace of beta.4
−0.2
30000
40000
50000
60000
Iterations
Density of beta.4
0
3
−0.2
0.0
0.2
0.4
0.6
N = 4000 Bandwidth = 0.02008
Trace of beta.5
−0.4
30000
40000
50000
60000
Iterations
Density of beta.5
0
3
−0.4
−0.2
0.0
0.2
0.4
N = 4000 Bandwidth = 0.02189
Trace of beta.6
−3.0
30000
40000
50000
60000
Iterations
Density of beta.6
0.0
−3.0
−2.8
−2.6
−2.4
−2.2
−2.0
N = 4000 Bandwidth = 0.03305
Trace of Z.1.1
−3
1
30000
40000
50000
60000
Iterations
Density of Z.1.1
0.0
−3
−2
−1
0
1
2
3
N = 4000 Bandwidth = 0.1568



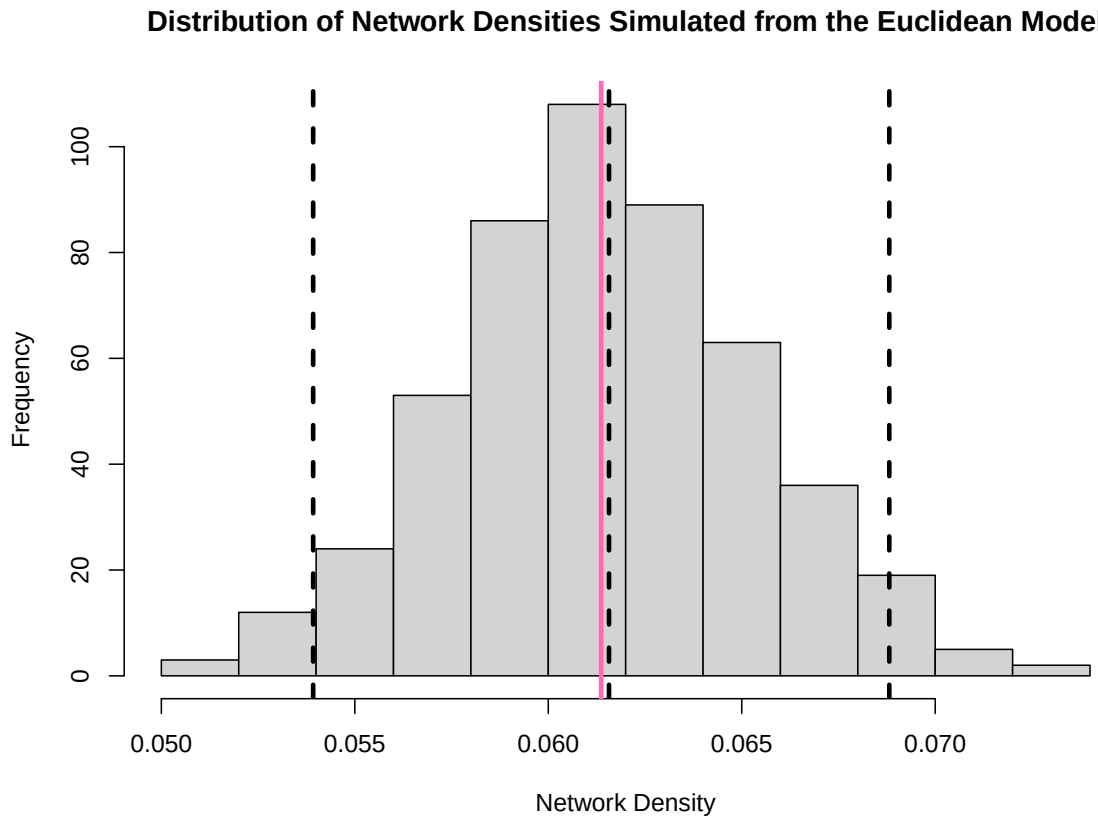



```
[[1]]

Quantile (q) = 0.025
Accuracy (r) = +/- 0.0125
Probability (s) = 0.95

        Burn-in  Total Lower bound  Dependence
        (M)      (N)   (Nmin)       factor (I)
 lpY    30       7030  600          11.7
 beta.1 200      35250 600          58.8
 beta.2 110      18470 600          30.8
 beta.3 90       15750 600          26.2
 beta.4 340      53820 600          89.7
 beta.5 240      41400 600          69.0
 beta.6 250      43860 600          73.1
 Z.1.1  200      26760 600          44.6
 Z.1.2  50       9240  600          15.4
```

The convergence diagnostics look good for both LSM models. There is no indication of a trend in the trace plots, and the plots all look approximately normal.

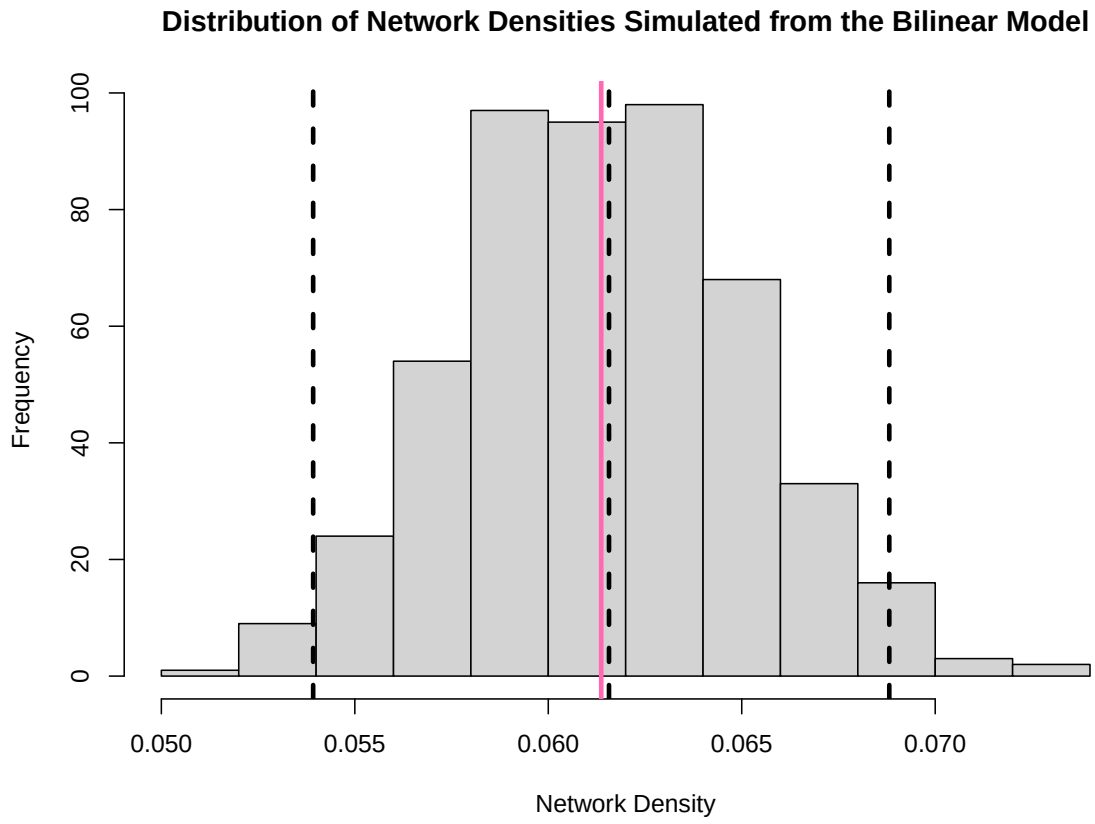
Checking LSM models for goodness of fit

```
lsm_density <- sapply(simulate(lsm, 500)[[2]], network.density)

hist(lsm_density, main = "Distribution of Network Densities Simulated from the Euclidean M
     xlab = "Network Density")
abline(v = network.density(add_health1_network), col = "hot pink", lty = 1, lwd = 3)
abline(v = quantile(lsm_density, probs = c(.025, .5, .975)), lty = 2, lwd = 3)
```



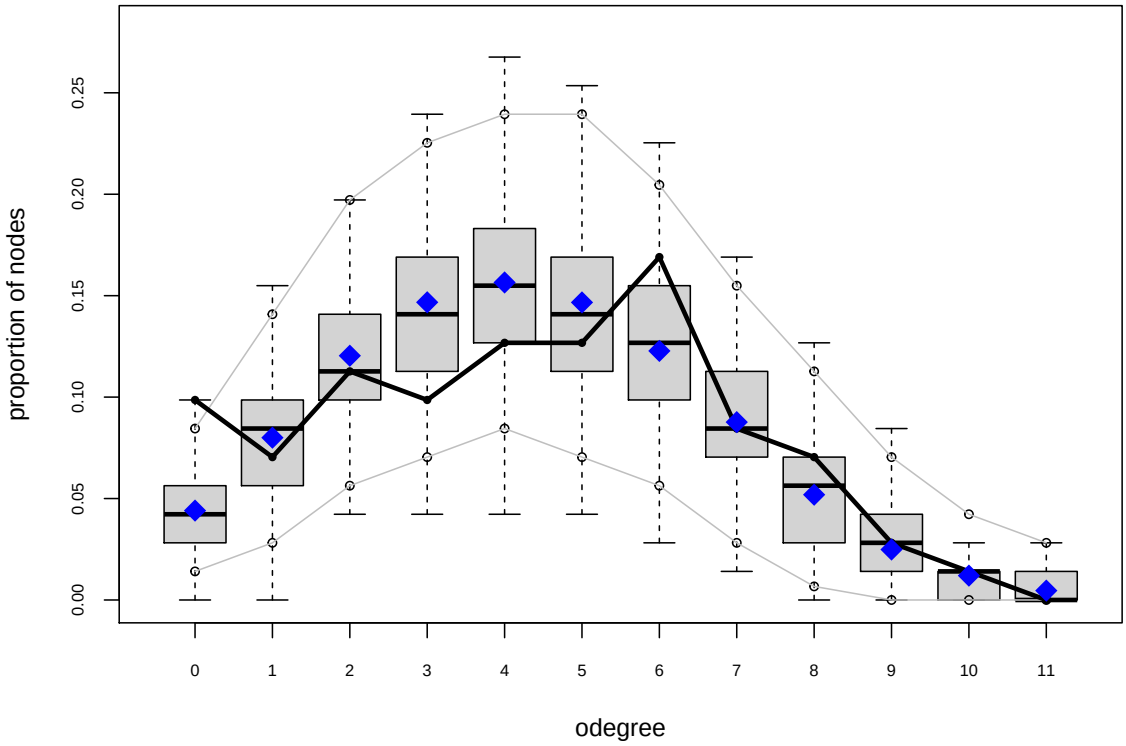

```
lsm_density_bi <- sapply(simulate(lsm_bi, 500)[[2]], network.density)
```

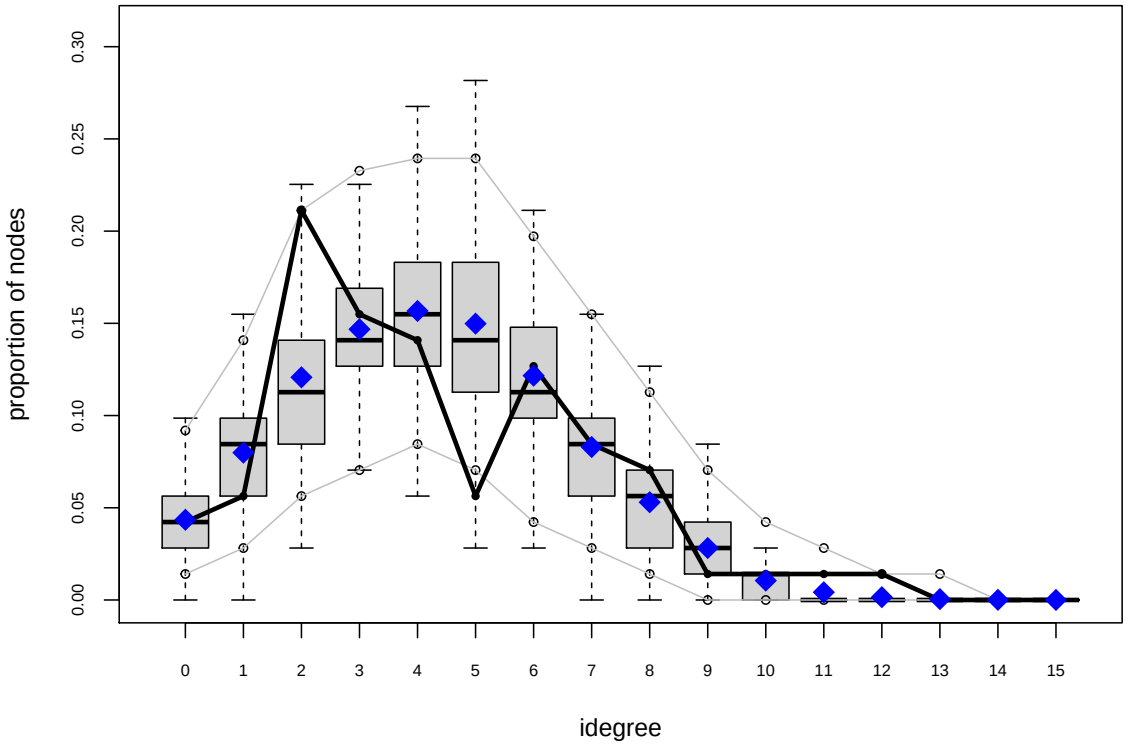
```
hist(lsm_density_bi, main = "Distribution of Network Densities Simulated from the Bilinear
     xlab = "Network Density")
abline(v = network.density(add_health1_network), col = "hot pink", lty = 1, lwd = 3)
abline(v = quantile(lsm_density, probs = c(.025, .5, .975)), lty = 2, lwd = 3)
```

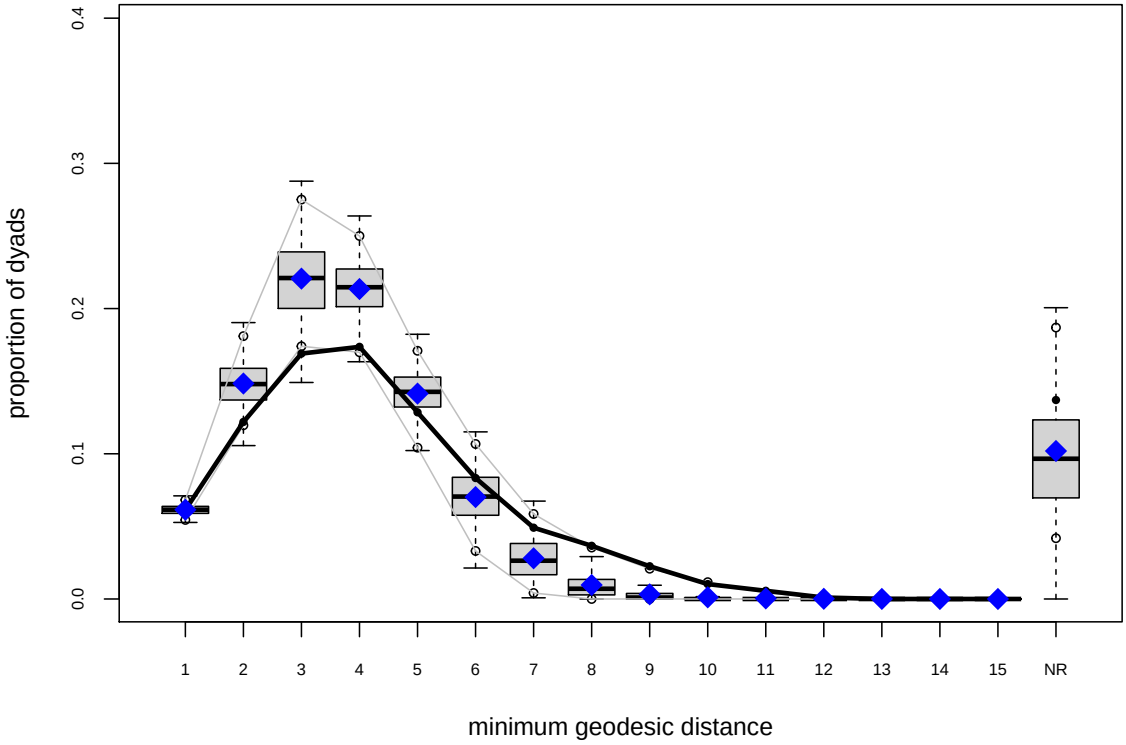


Both models produce network densities in line with the simulations which is good.

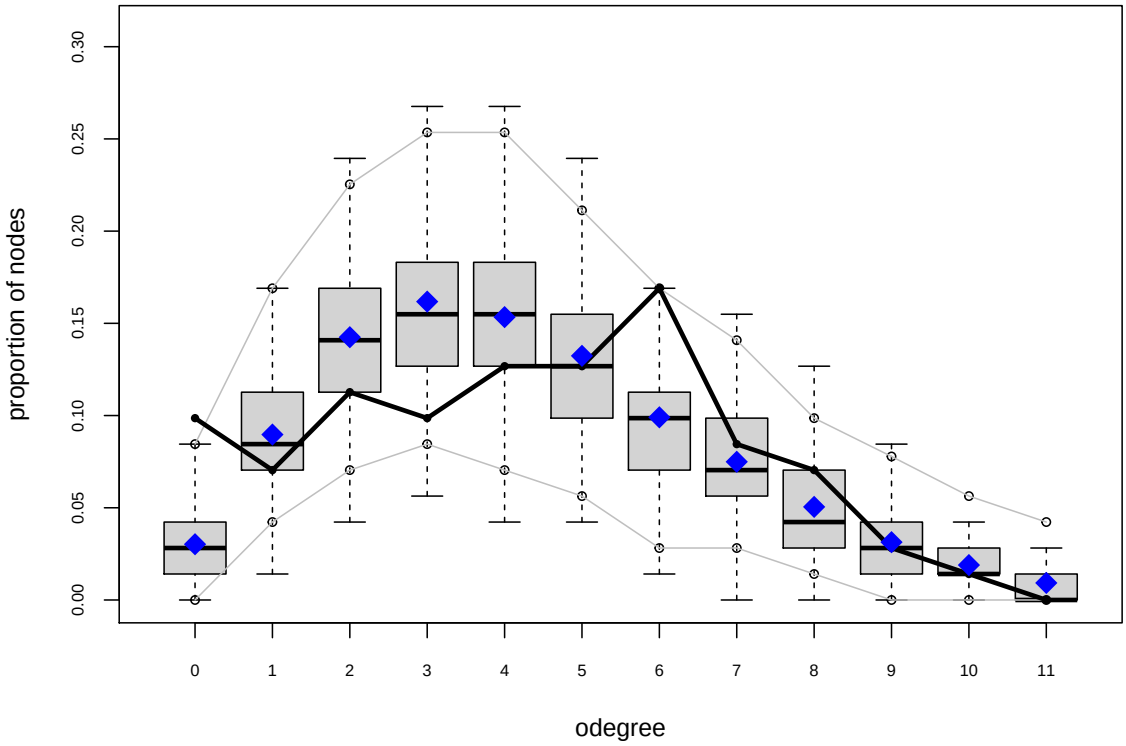
```
plot(gof(lsm, GOF = ~ idegree + odegree + dist, nsim = 500))
```

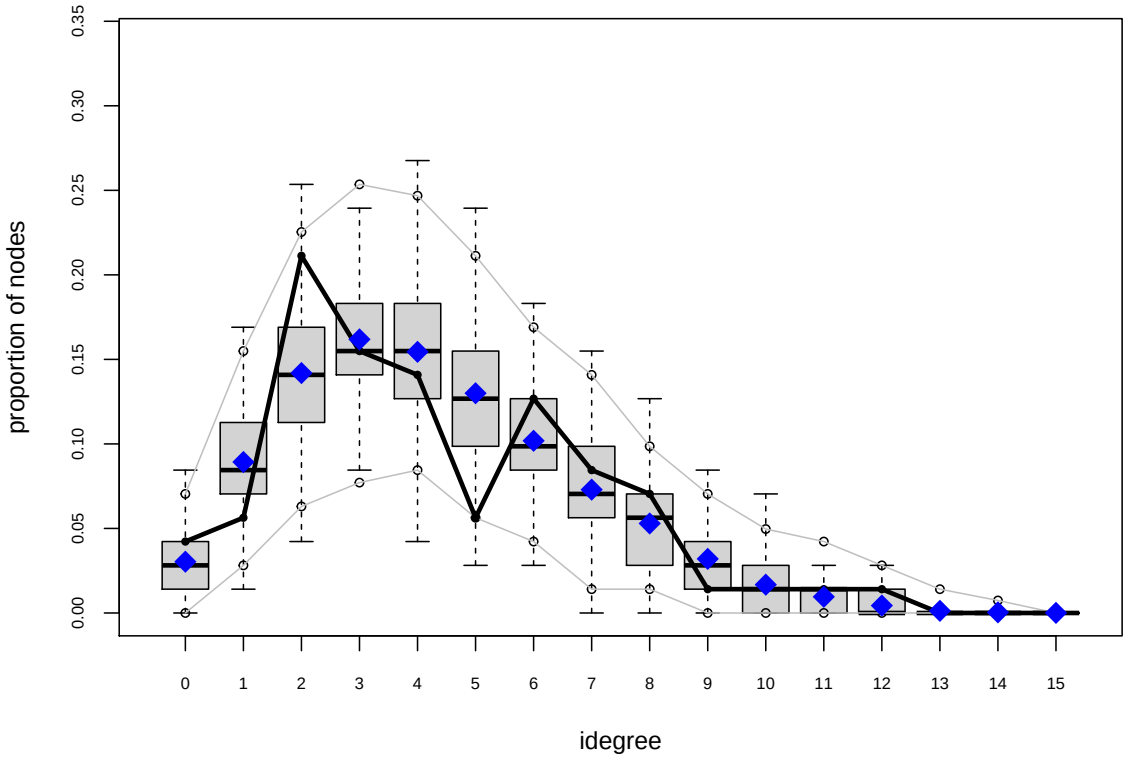

proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0
1
2
3
4
5
6
7
8
9
10
11
odegree

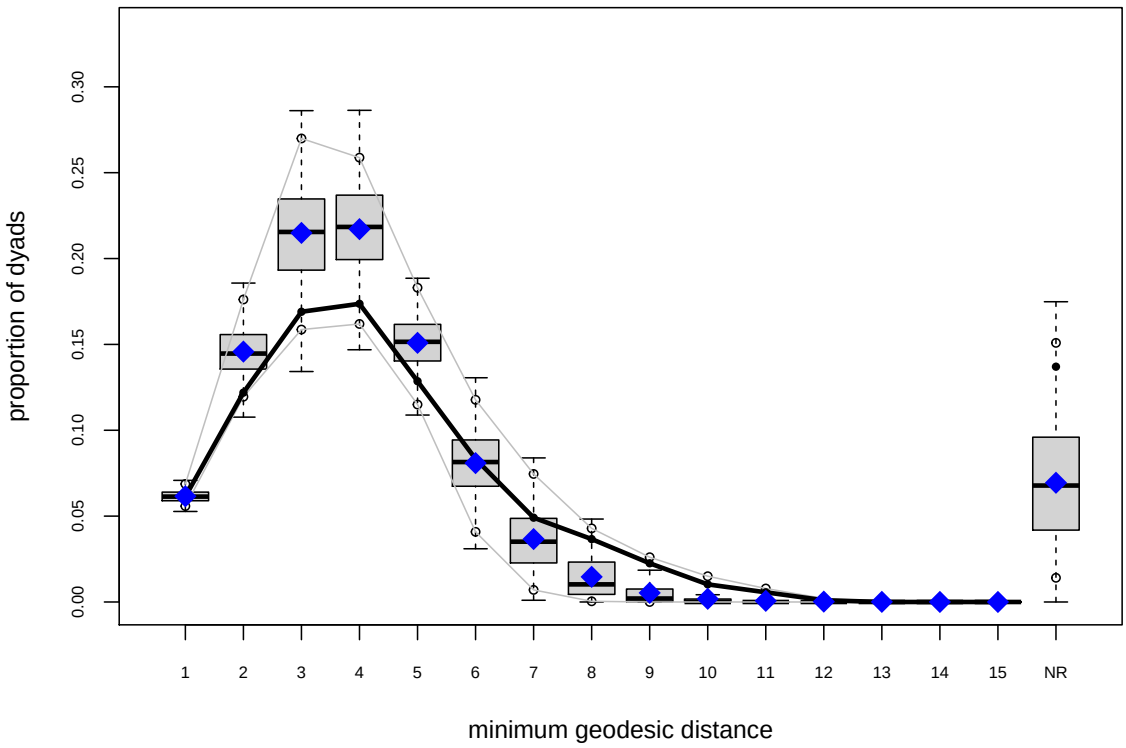

proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
idegree




```
plot(gof(lsm_bi, GOF = ~ idegree + odegree + dist, nsim = 500))
```


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
6
7
8
9
10
11
odegree


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
idegree



Other visual inspections of model fit indicate, though, these models still are not great at fitting the data.

Conclusion

The models all generally agreed upon which exogeneous variables were most important (being the same sex and being in the same grade). The ERGM and LSM models also indicates there are some endogeneous variables at play, too. In particular, using the ERGM model I was able to find that mutuality and geometrically weighted out degree (sociality) were important.