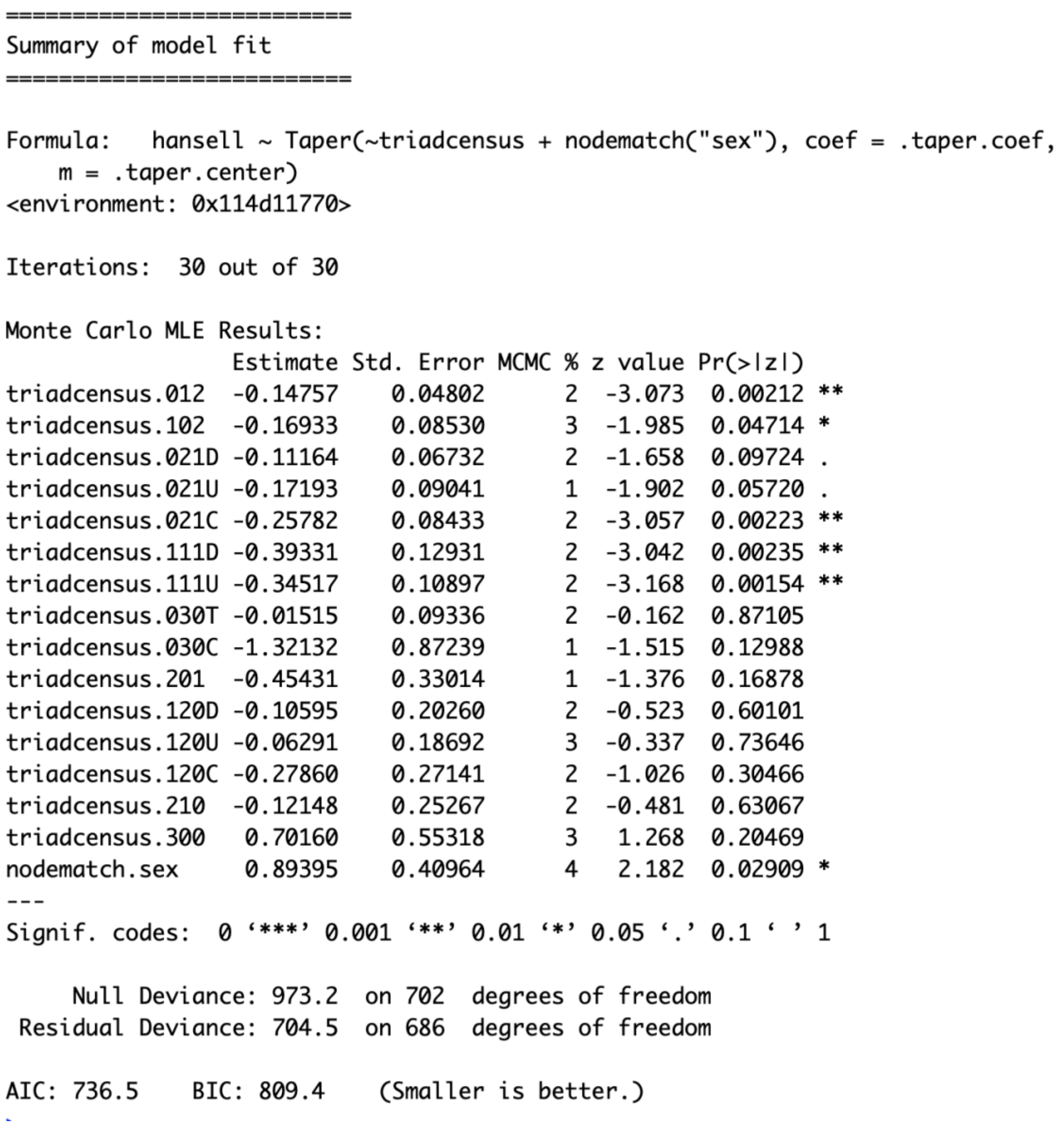
Stats218 Homework 4
Name: Ruchen Zhen
UID: 205036407

Problem 1: Modeling a Triad Census in Friendship Relations: Hansell Network

(a)

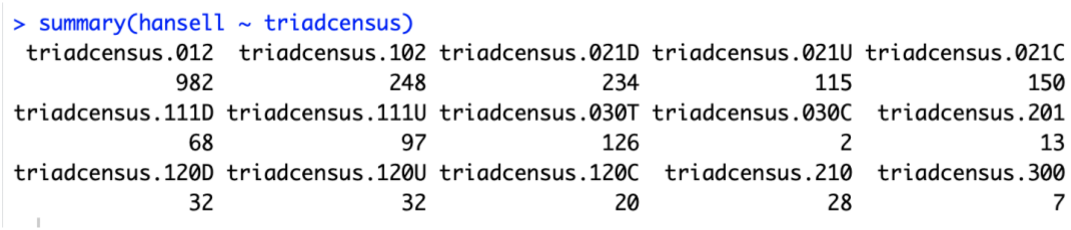


```
==========================
Summary of model fit
==========================

Formula:   hansell ~ Taper(~triadcensus + nodematch("sex"), coef = .taper.coef,
    m = .taper.center)
<environment: 0x114d11770>

Iterations:  30 out of 30

Monte Carlo MLE Results:
                 Estimate Std. Error MCMC % z value Pr(>|z|)
triadcensus.012  -0.14757    0.04802      2  -3.073  0.00212 **
triadcensus.102  -0.16933    0.08530      3  -1.985  0.04714 *
triadcensus.021D -0.11164    0.06732      2  -1.658  0.09724 .
triadcensus.021U -0.17193    0.09041      1  -1.902  0.05720 .
triadcensus.021C -0.25782    0.08433      2  -3.057  0.00223 **
triadcensus.111D -0.39331    0.12931      2  -3.042  0.00235 **
triadcensus.111U -0.34517    0.10897      2  -3.168  0.00154 **
triadcensus.030T -0.01515    0.09336      2  -0.162  0.87105
triadcensus.030C -1.32132    0.87239      1  -1.515  0.12988
triadcensus.201  -0.45431    0.33014      1  -1.376  0.16878
triadcensus.120D -0.10595    0.20260      2  -0.523  0.60101
triadcensus.120U -0.06291    0.18692      3  -0.337  0.73646
triadcensus.120C -0.27860    0.27141      2  -1.026  0.30466
triadcensus.210  -0.12148    0.25267      2  -0.481  0.63067
triadcensus.300   0.70160    0.55318      3   1.268  0.20469
nodematch.sex     0.89395    0.40964      4   2.182  0.02909 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 973.2  on 702  degrees of freedom
 Residual Deviance: 704.5  on 686  degrees of freedom

AIC: 736.5    BIC: 809.4    (Smaller is better.)
```

(b)

As shown in the fitting result, the coefficient of homophily of sex is positive, with low p-value and large z-value. Triad 300 also comes up with a coefficient of 0.7, but the p-value is large, which indicates that it is not significant enough. Triadsencus 030C comes up with a coefficient of -1.32, which indicates that 030C occurs relatively rare in the network (and 030C does occurs only 2 times in the network). Based on the model, there is homophily by sex. Besides, since there exist many transitive triads, 030T, 120D, 120U and 300, there does appear to be a preference for transitive friendship ties. However, this is not that significant in the model fitted.

Summary of triadcensus:

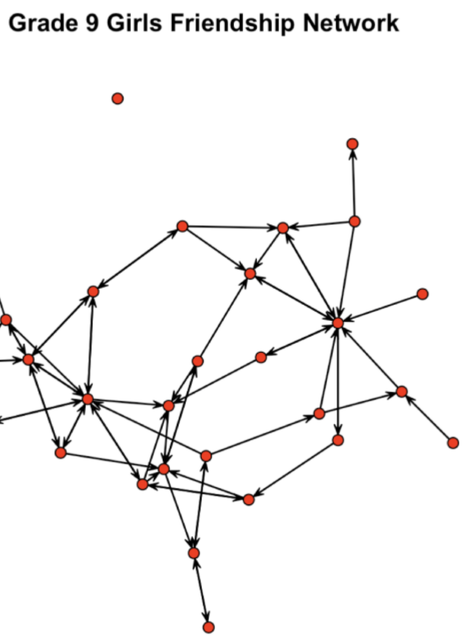


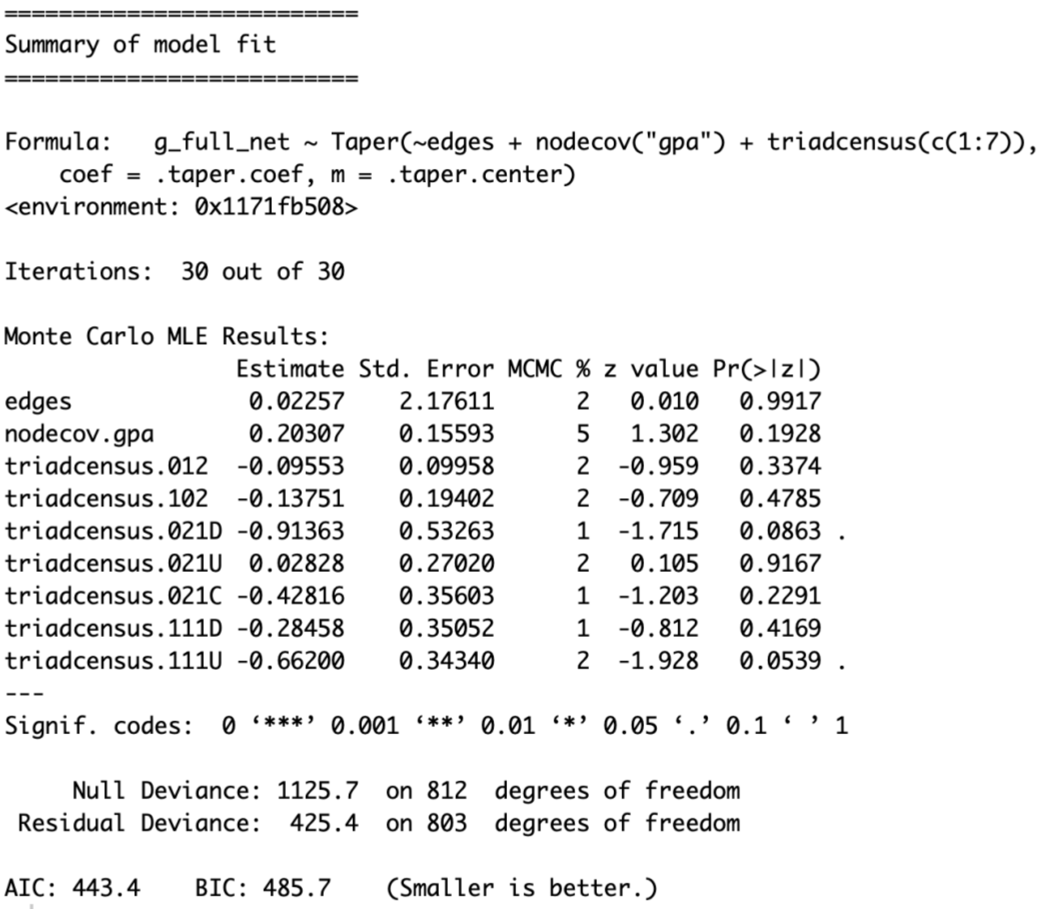
```
> summary(hansell ~ triadcensus)
 triadcensus.012  triadcensus.102 triadcensus.021D triadcensus.021U triadcensus.021C
             982              248              234              115              150
triadcensus.111D triadcensus.111U triadcensus.030T triadcensus.030C  triadcensus.201
              68               97              126                2               13
triadcensus.120D triadcensus.120U triadcensus.120C  triadcensus.210  triadcensus.300
              32               32               20               28                7
```

012 is dominant, 102, 021D, 021U, 021C, 030T also contributed a lot.

Problem 2: Modeling a Triad Census in Friendship Relations: gfriends Network

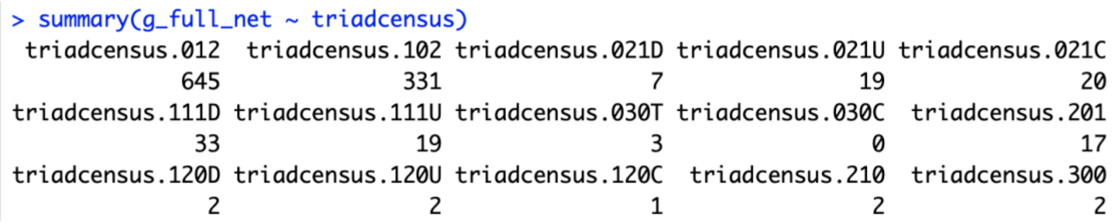
(a)




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

Formula:   g_full_net ~ Taper(~edges + nodecov("gpa") + triadcensus(c(1:7)),
    coef = .taper.coef, m = .taper.center)
<environment: 0x1171fb508>

Iterations:  30 out of 30

Monte Carlo MLE Results:
                 Estimate Std. Error MCMC % z value Pr(>|z|)
edges             0.02257    2.17611      2   0.010   0.9917
nodecov.gpa       0.20307    0.15593      5   1.302   0.1928
triadcensus.012  -0.09553    0.09958      2  -0.959   0.3374
triadcensus.102  -0.13751    0.19402      2  -0.709   0.4785
triadcensus.021D -0.91363    0.53263      1  -1.715   0.0863 .
triadcensus.021U  0.02828    0.27020      2   0.105   0.9167
triadcensus.021C -0.42816    0.35603      1  -1.203   0.2291
triadcensus.111D -0.28458    0.35052      1  -0.812   0.4169
triadcensus.111U -0.66200    0.34340      2  -1.928   0.0539 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 1125.7  on 812  degrees of freedom
 Residual Deviance:  425.4  on 803  degrees of freedom

AIC: 443.4    BIC: 485.7    (Smaller is better.)
```

(b)

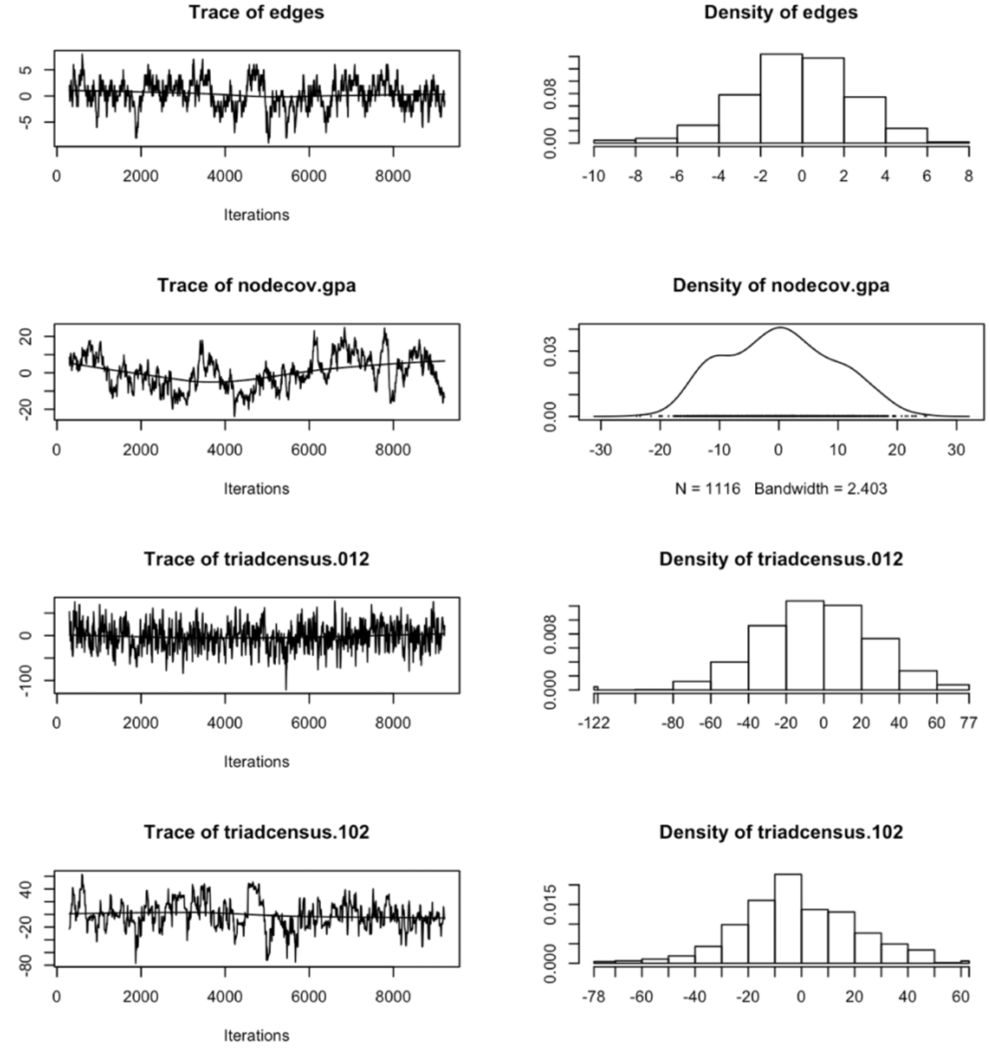
Most estimates of parameters are around 0. The only one which is kind of significant, is 021D, with the estimate of -0.91 and p-value of 0.08. This indicates that 021D occurs relatively rarely in the network. And it fits the fact that there is only 7 021D appears.

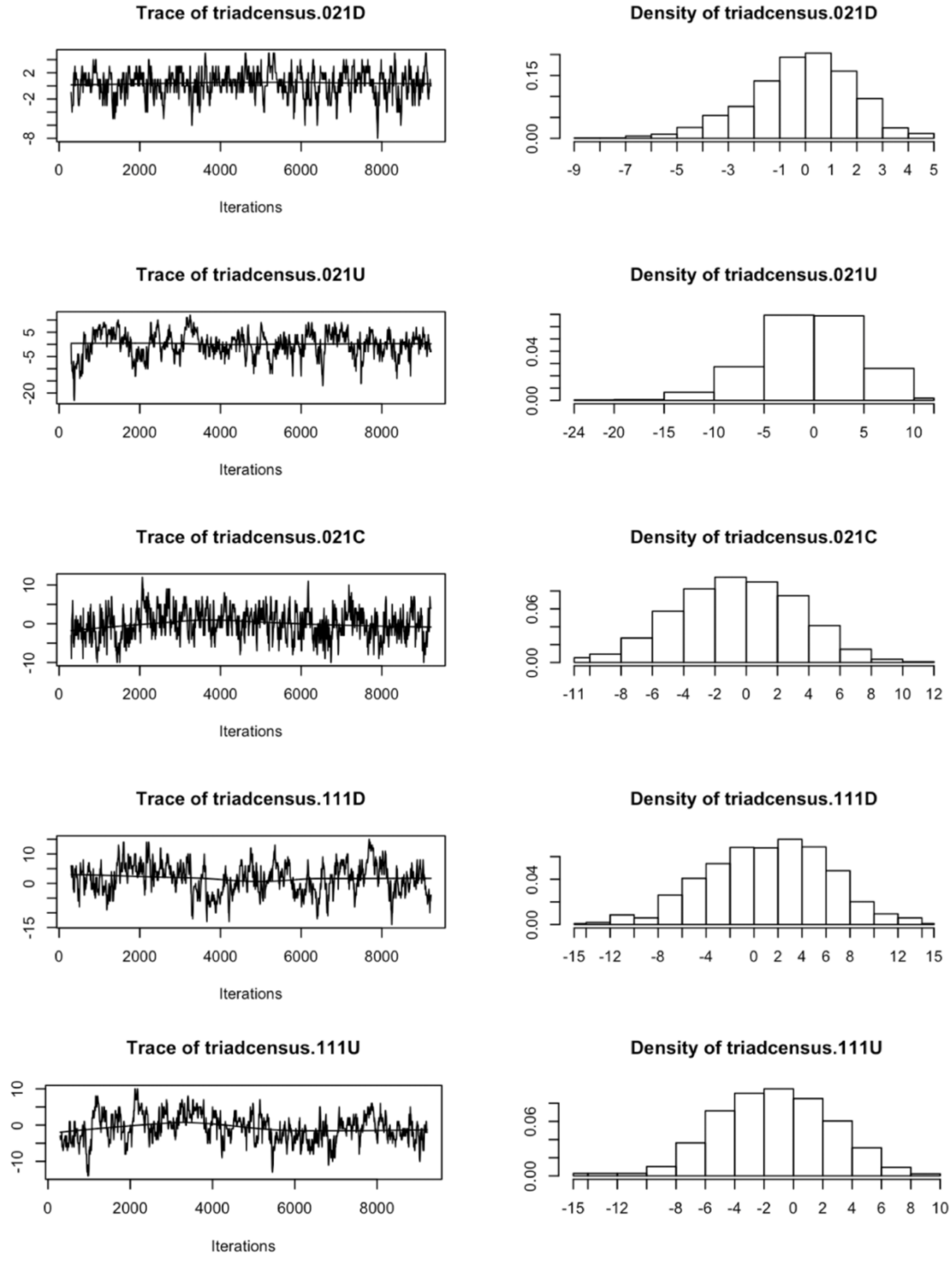


```
> summary(g_full_net ~ triadcensus)
 triadcensus.012  triadcensus.102 triadcensus.021D triadcensus.021U triadcensus.021C
             645              331                7               19               20
triadcensus.111D triadcensus.111U triadcensus.030T triadcensus.030C  triadcensus.201
              33               19                3                0               17
triadcensus.120D triadcensus.120U triadcensus.120C  triadcensus.210  triadcensus.300
               2                2                1                2                2
```

As for the triadcensus, type 012 and 102 dominates. There isn't much transitive triads, 120D, 120U, 300, 030T have a total number of 9.

(c)

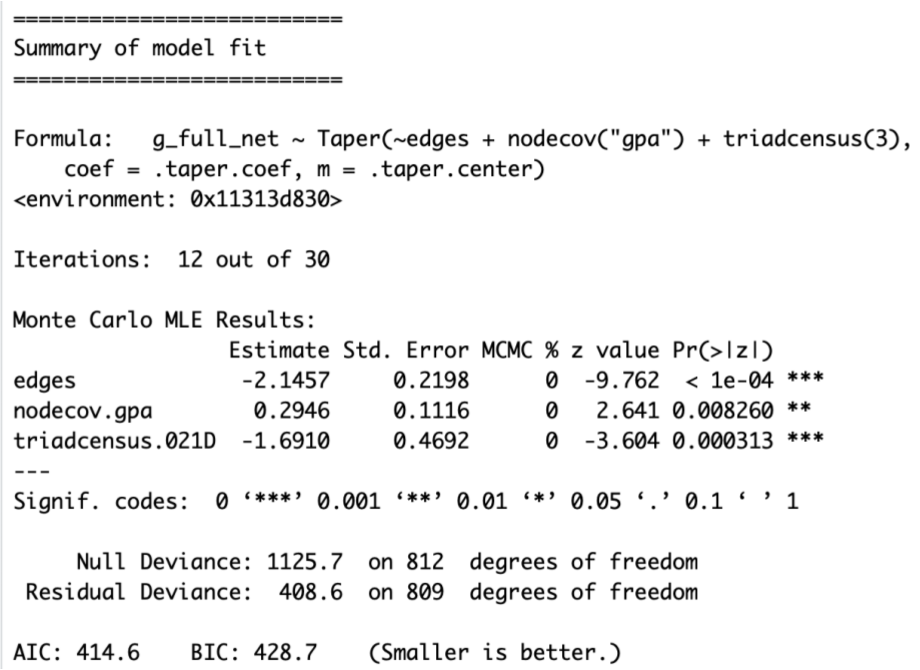




The MCMC diagnostics shows that the model greatly converges. From the trace plots on the left, with the iterations increases, the results still isolated near the centers. For the density plots on the right, all densities of parameter differences center around zero, and basically form Gaussian distributions.

(d)

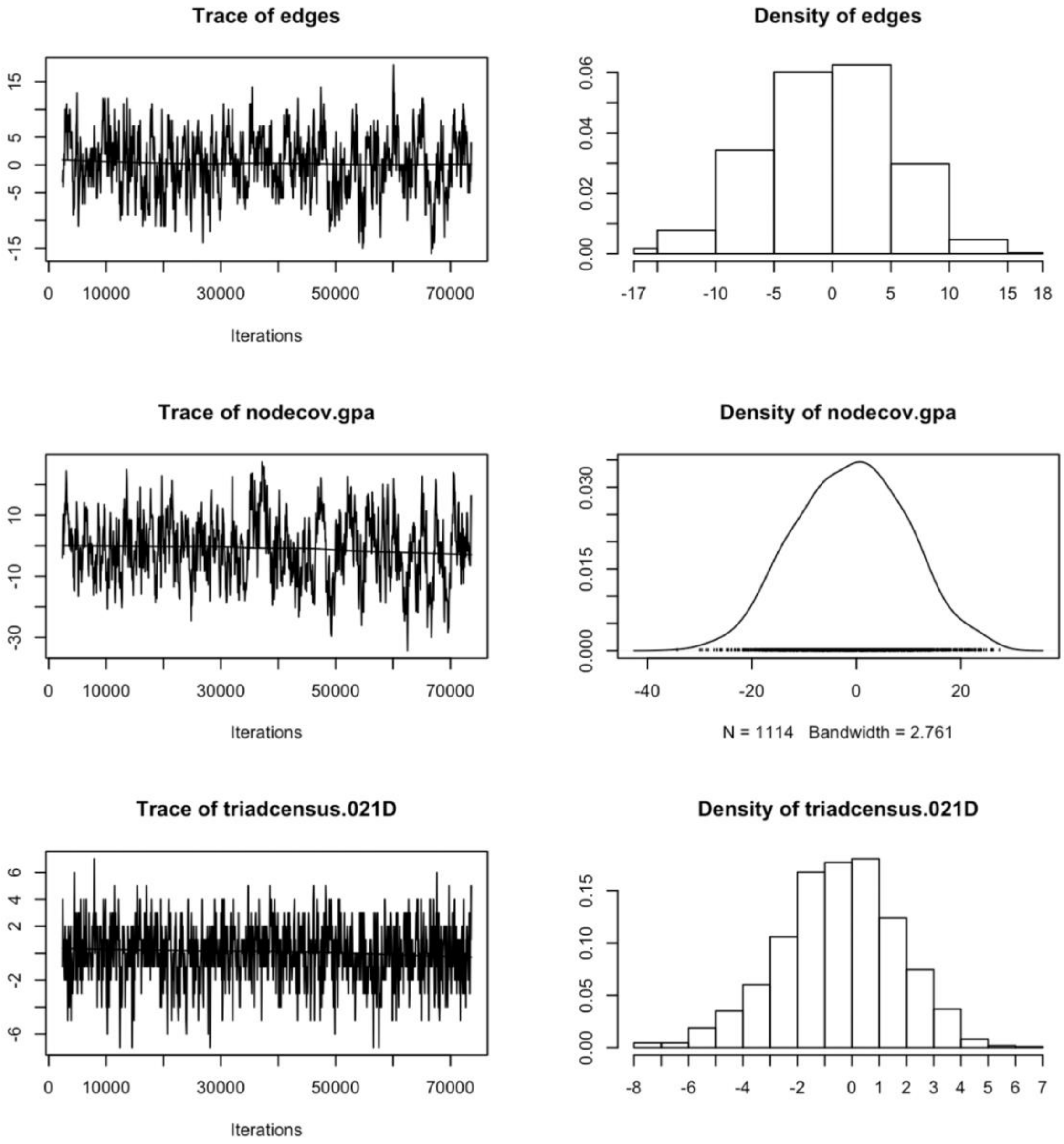
From the last model, I find that the 021D term is the only significant term. Thus, in the abbreviated model, I only keep the 021D triadcensus term, plus edges and nodecov(“gpa”).



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

Formula:   g_full_net ~ Taper(~edges + nodecov("gpa") + triadcensus(3),
    coef = .taper.coef, m = .taper.center)
<environment: 0x11313d830>

Iterations:  12 out of 30

Monte Carlo MLE Results:
                 Estimate Std. Error MCMC % z value Pr(>|z|)
edges             -2.1457     0.2198      0  -9.762  < 1e-04 ***
nodecov.gpa        0.2946     0.1116      0   2.641 0.008260 **
triadcensus.021D  -1.6910     0.4692      0  -3.604 0.000313 ***
---
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 809  degrees of freedom

AIC: 414.6    BIC: 428.7    (Smaller is better.)
```

The summary shows again that triad 021D occur rarely in the network. The MCMC diagnostics shows that the model converges well.

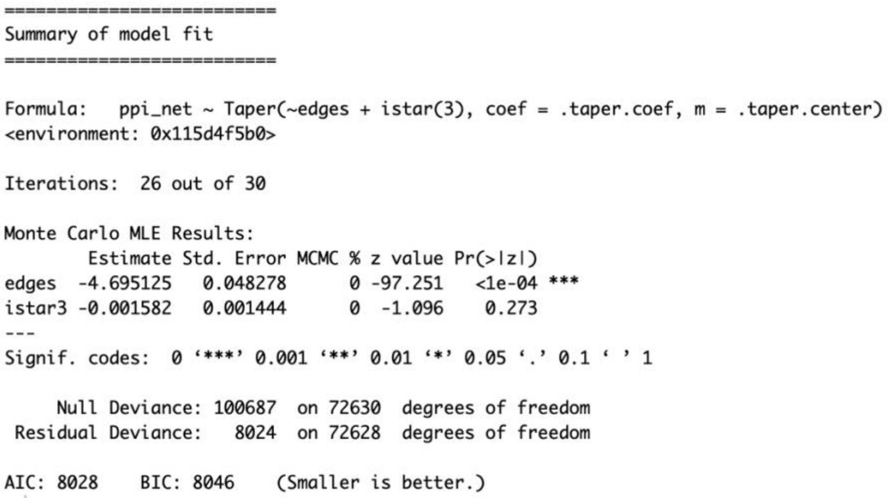
This model is not worse than the previous one, and with less parameters, which is easy to fit as well as less likely to overfit. In fact, by looking at the goodness-of-fit plots, both of the two models do not fit perfect.


Trace of edges
15
5
0
-5
-15
0
10000
30000
50000
70000
Iterations
Density of edges
0.06
0.04
0.02
0.00
-17
-10
-5
0
5
10
15
18
Trace of nodecov.gpa
10
-10
-30
0
10000
30000
50000
70000
Iterations
Density of nodecov.gpa
0.030
0.015
0.000
-40
-20
0
20
N = 1114 Bandwidth = 2.761
Trace of triadcensus.021D
6
4
2
-2
-6
0
10000
30000
50000
70000
Iterations
Density of triadcensus.021D
0.15
0.10
0.05
0.00
-8
-6
-4
-2
0
1
2
3
4
5
6
7

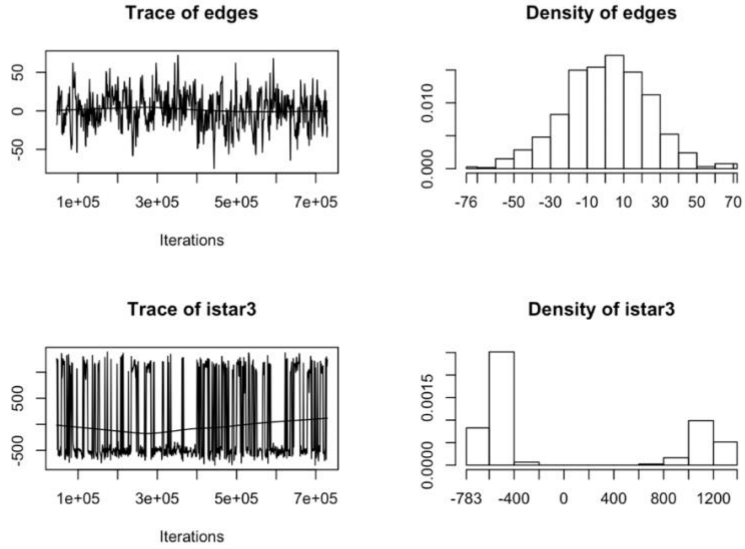
Problem 3: Model for Protein-protein interaction data

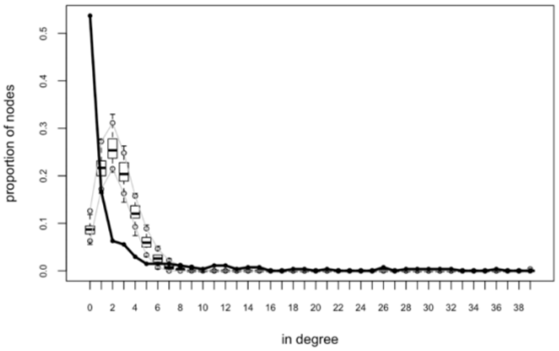
1.Model with edges+istar(3)

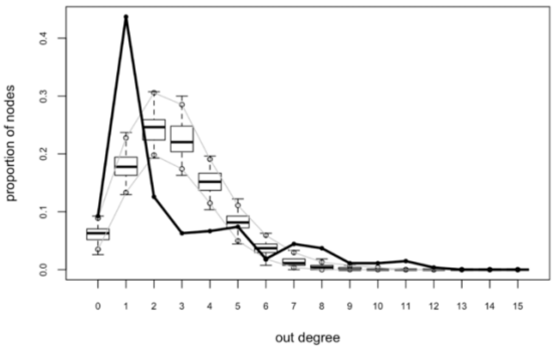
From the MCMC plot, the density of istar(3) definitely not converge. The goodness-of-fit plot also shows that the fit is terrible. For example, both the in degree and out degree do not fit well. The estimate is 0.001044.



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

Formula:   ppi_net ~ Taper(~edges + istar(3), coef = .taper.coef, m = .taper.center)
<environment: 0x115d4f5b0>

Iterations:  26 out of 30

Monte Carlo MLE Results:
        Estimate Std. Error MCMC % z value Pr(>|z|)
edges  -4.695125   0.048278      0 -97.251   <1e-04 ***
istar3 -0.001582   0.001444      0  -1.096    0.273
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 100687  on 72630  degrees of freedom
 Residual Deviance:   8024  on 72628  degrees of freedom

AIC: 8028    BIC: 8046    (Smaller is better.)
```







Since that there are too many plots, I will skip the plots of each individual terms' fitting and only perform the conclusions.

2. Model with edges + ostar(3)
Converge with bad goodness-of-fit. The estimate is 0.009651.

3. edges + gwodegree(0.5,fixed = TRUE)
Converge with bad goodness-of-fit. The estimate is -1.46103

4. edges + gwidegree(0.5,fixed = TRUE)
Converge with bad goodness-of-fit. The estimate is -4.91226.

5. edges + dgwesp(0.5,fixed = TRUE)
Converge with bad goodness-of-fit. But better than the previous models. The estimate is 1.63378.

6. edges + dgwdsp(0.5,fixed = TRUE)
Converge with bad goodness-of-fit. The estimate is -0.1680.

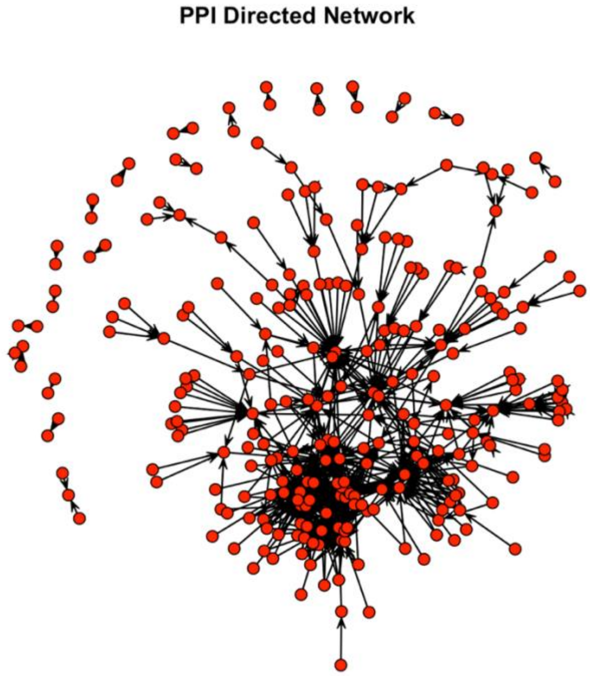
7. edges + ctriple
Converge with bad goodness-of-fit. The estimate for cyclic triple is -31.38, which indicates that this term might not be important.

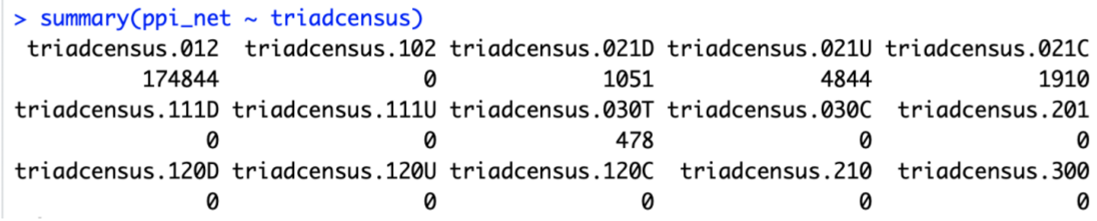
8. edges + ttriple
Converge with bad goodness-of-fit. The estimate for transitive triple is 0.41173.

Besides, the estimate of edges is -4.673342.

Since the individual parameter do not perform well, I will try some combinations of the relative significant parameters.

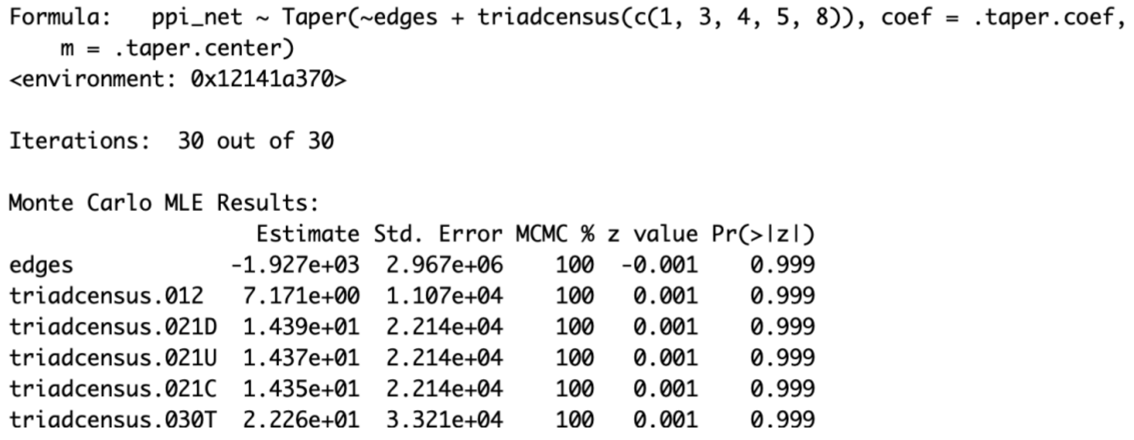
First check some basic performance of the network.



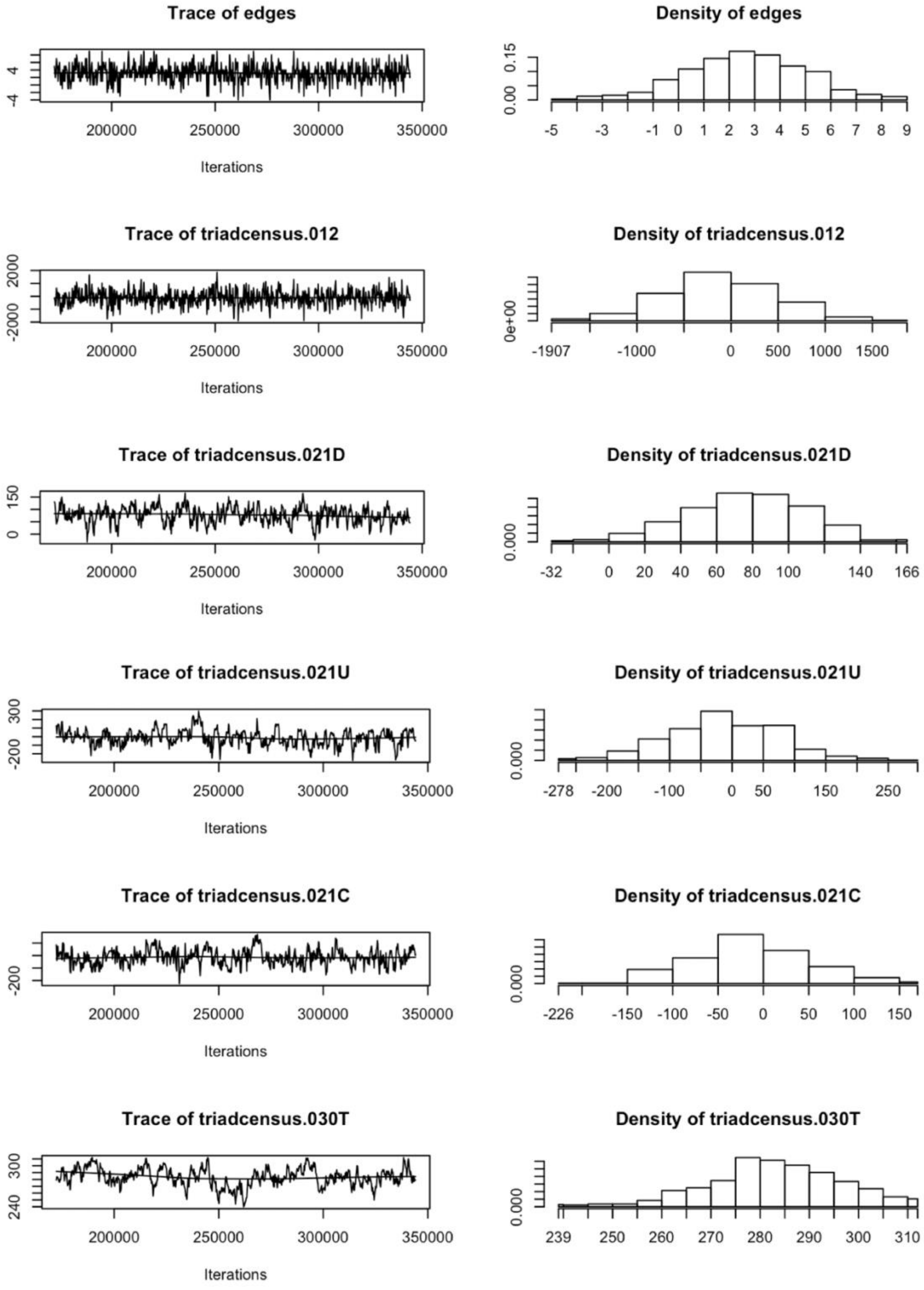


```
> summary(ppi_net ~ triadcensus)
 triadcensus.012  triadcensus.102 triadcensus.021D triadcensus.021U triadcensus.021C
          174844                0             1051             4844             1910
triadcensus.111D triadcensus.111U triadcensus.030T triadcensus.030C  triadcensus.201
               0                0              478                0                0
triadcensus.120D triadcensus.120U triadcensus.120C  triadcensus.210  triadcensus.300
               0                0                0                0                0
```

Fit a model which considers the 5 kinds of existed traids:



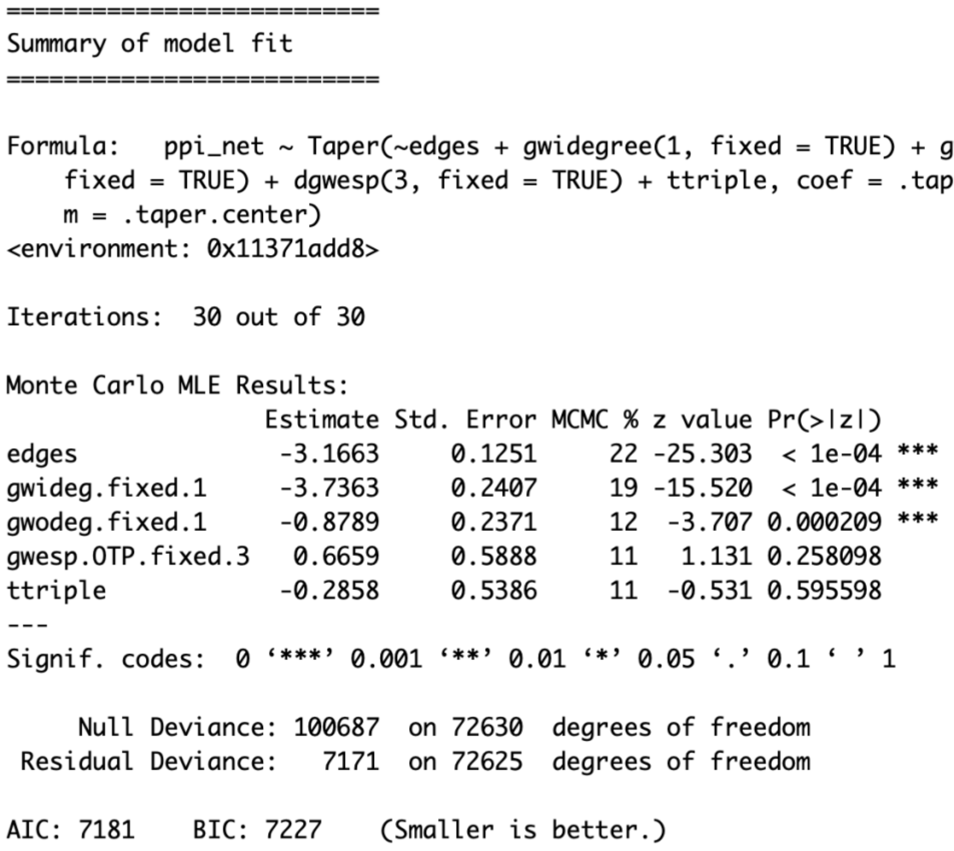
```
Formula:   ppi_net ~ Taper(~edges + triadcensus(c(1, 3, 4, 5, 8)), coef = .taper.coef,
    m = .taper.center)
<environment: 0x12141a370>

Iterations:  30 out of 30

Monte Carlo MLE Results:
                   Estimate Std. Error MCMC % z value Pr(>|z|)
edges            -1.927e+03  2.967e+06    100  -0.001    0.999
triadcensus.012   7.171e+00  1.107e+04    100   0.001    0.999
triadcensus.021D  1.439e+01  2.214e+04    100   0.001    0.999
triadcensus.021U  1.437e+01  2.214e+04    100   0.001    0.999
triadcensus.021C  1.435e+01  2.214e+04    100   0.001    0.999
triadcensus.030T  2.226e+01  3.321e+04    100   0.001    0.999
```



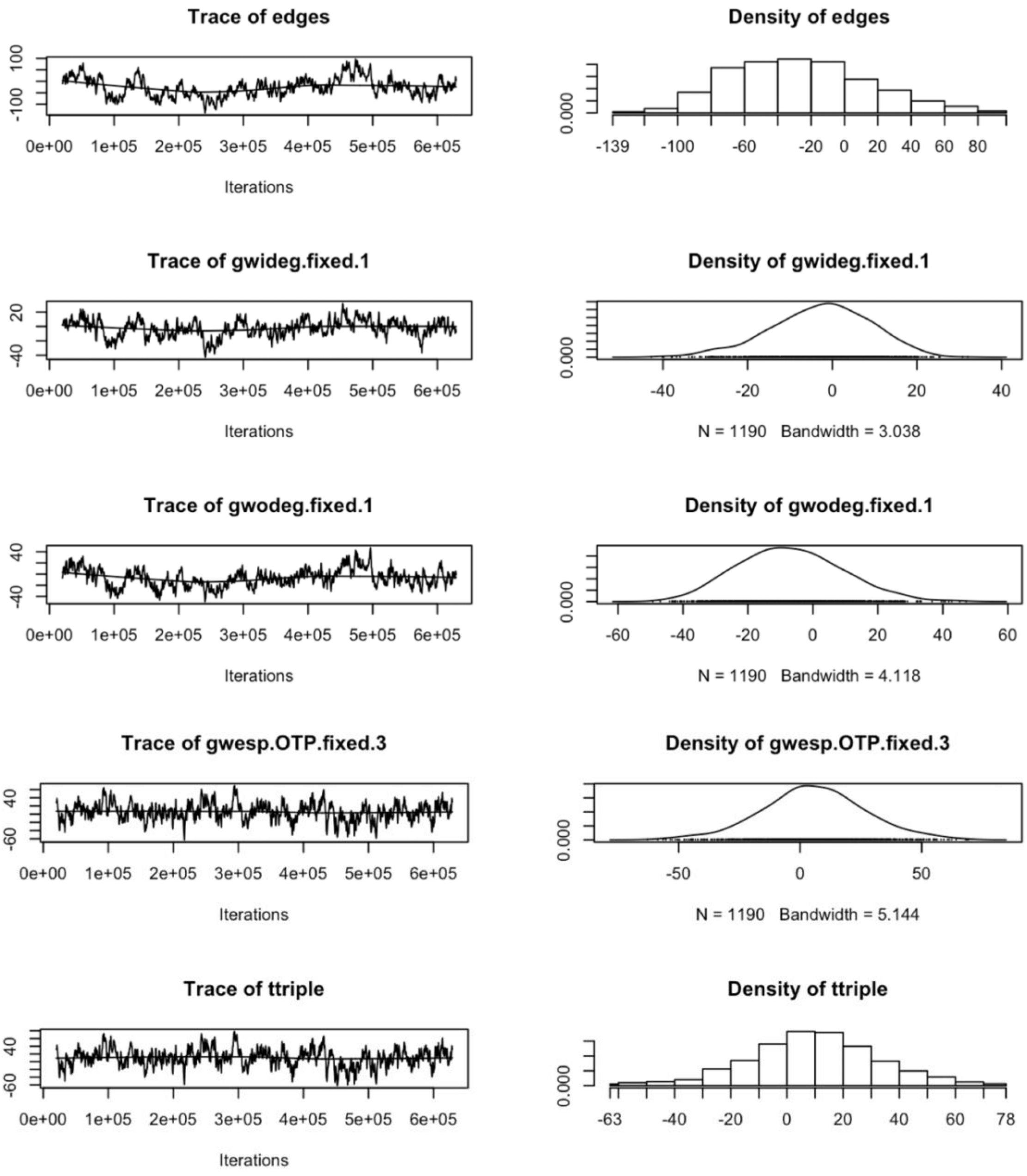
The estimates are large, however, the p-value is extremely large. The MCMC diagnostics show that the density of 030T and 021D are not centered around zero.

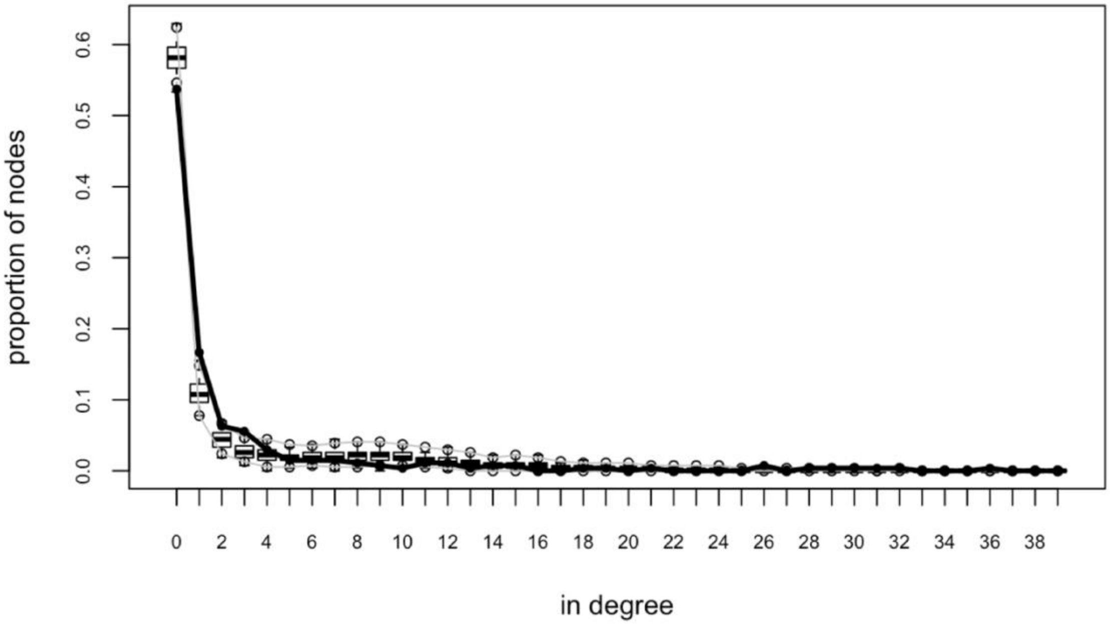
Try some combinations of significant parameters. We cannot choose too many parameters given that it may leads to overfit.

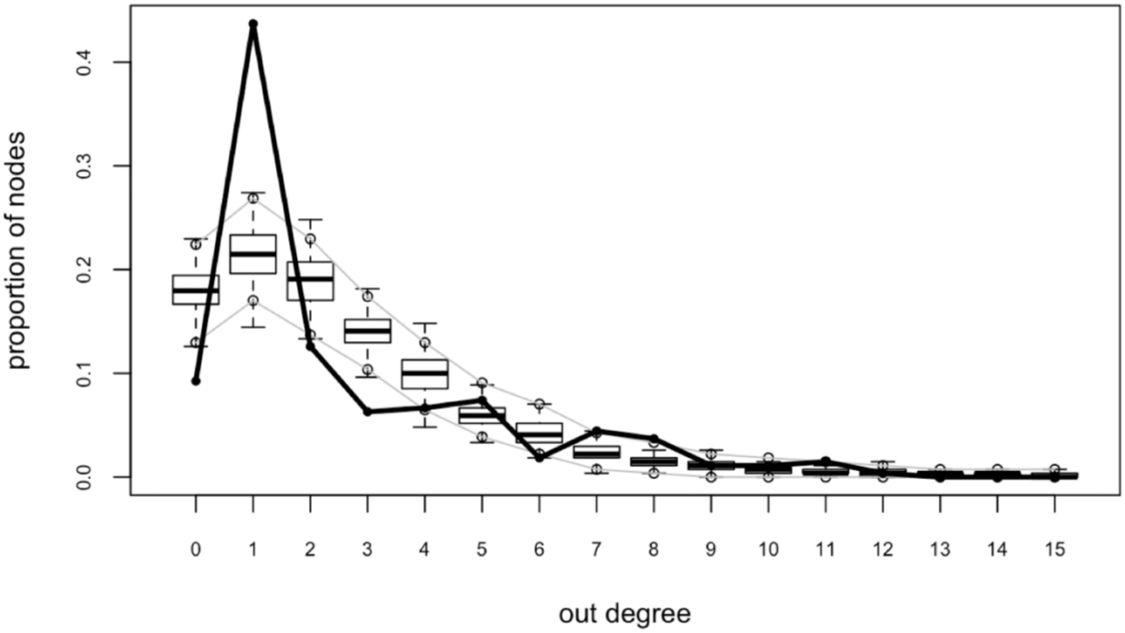
After multiple attempts, the best fit model I find is: edges + gwidegree(1,fixed = TRUE) + gwodegree(1,fixed = TRUE) + dgwesp(3,fixed = TRUE) + ttriple.

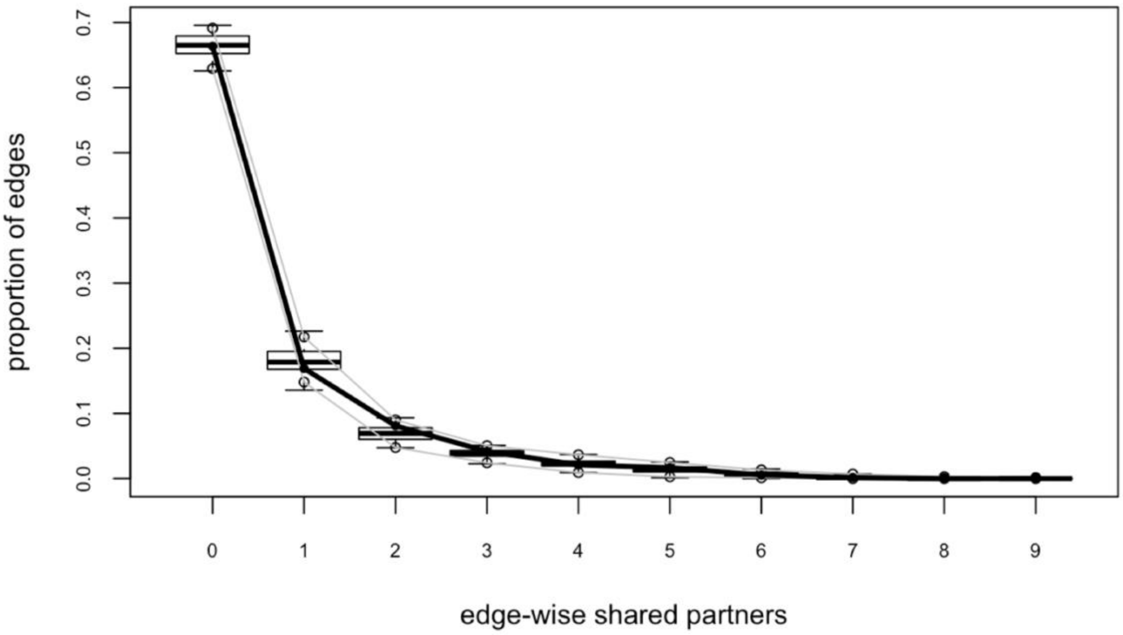


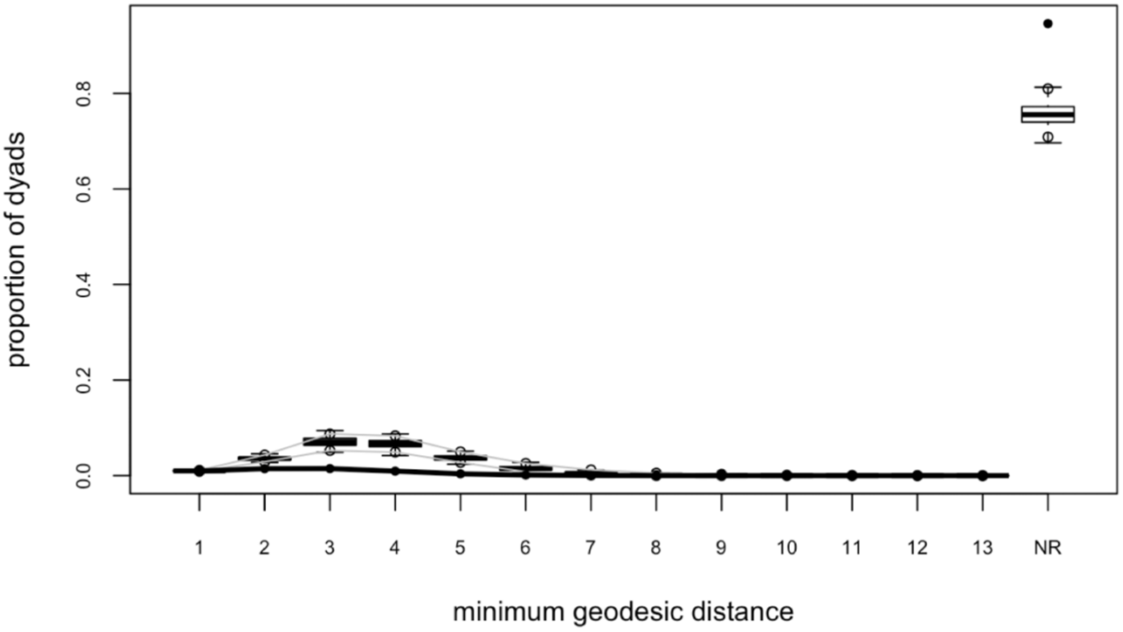
```
==========================
Summary of model fit
==========================

Formula:   ppi_net ~ Taper(~edges + gwidegree(1, fixed = TRUE) + gwodegree(1, 
    fixed = TRUE) + dgwesp(3, fixed = TRUE) + ttriple, coef = .taper.coef, 
    m = .taper.center)
<environment: 0x11371add8>

Iterations:  30 out of 30 

Monte Carlo MLE Results:
                  Estimate Std. Error MCMC % z value Pr(>|z|)    
edges              -3.1663     0.1251     22 -25.303  < 1e-04 ***
gwideg.fixed.1     -3.7363     0.2407     19 -15.520  < 1e-04 ***
gwodeg.fixed.1     -0.8789     0.2371     12  -3.707 0.000209 ***
gwesp.OTP.fixed.3   0.6659     0.5888     11   1.131 0.258098    
ttriple            -0.2858     0.5386     11  -0.531 0.595598    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 100687  on 72630  degrees of freedom
 Residual Deviance:   7171  on 72625  degrees of freedom
 
AIC: 7181    BIC: 7227    (Smaller is better.) 
```

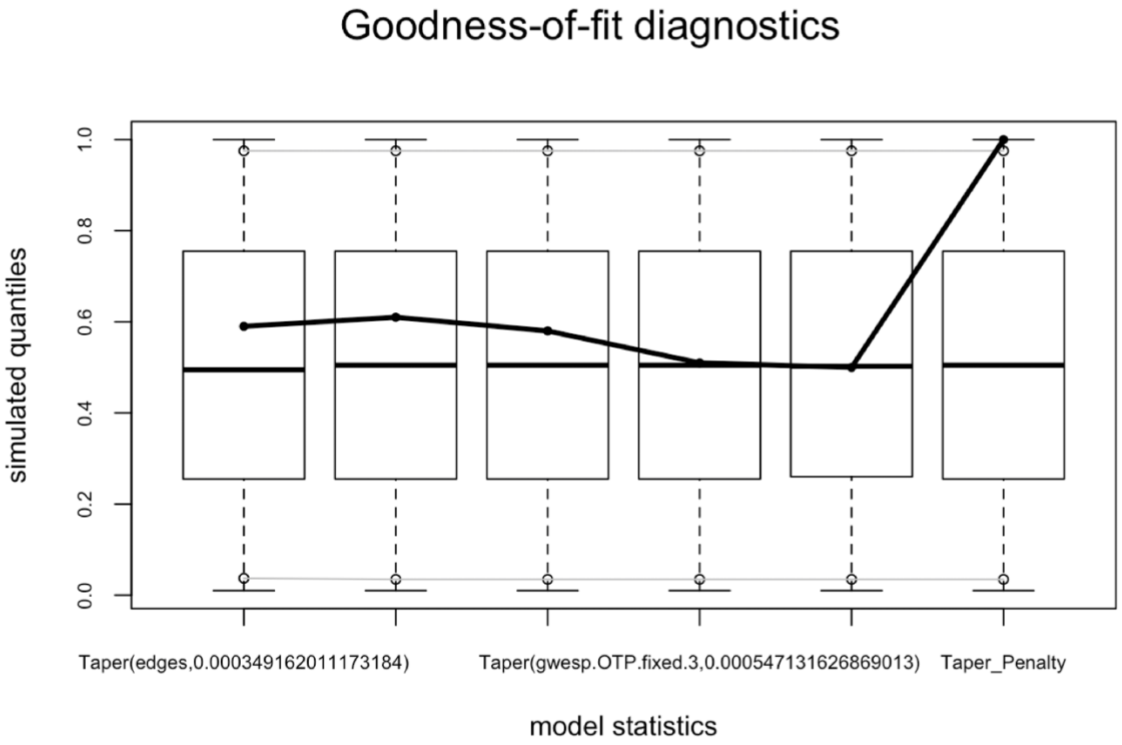

Trace of edges
100
-100
0e+00 1e+05 2e+05 3e+05 4e+05 5e+05 6e+05
Iterations
Density of edges
0.000
-139 -100 -60 -20 0 20 40 60 80
Trace of gwideg.fixed.1
20
-40
0e+00 1e+05 2e+05 3e+05 4e+05 5e+05 6e+05
Iterations
Density of gwideg.fixed.1
0.000
-40 -20 0 20 40
N = 1190 Bandwidth = 3.038
Trace of gwodeg.fixed.1
40
-40
0e+00 1e+05 2e+05 3e+05 4e+05 5e+05 6e+05
Iterations
Density of gwodeg.fixed.1
0.000
-60 -40 -20 0 20 40 60
N = 1190 Bandwidth = 4.118
Trace of gwesp.OTP.fixed.3
40
-60
0e+00 1e+05 2e+05 3e+05 4e+05 5e+05 6e+05
Iterations
Density of gwesp.OTP.fixed.3
0.000
-50 0 50
N = 1190 Bandwidth = 5.144
Trace of ttriple
40
-60
0e+00 1e+05 2e+05 3e+05 4e+05 5e+05 6e+05
Iterations
Density of ttriple
0.000
-63 -40 -20 0 20 40 60 78


proportion of nodes
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
in degree


proportion of nodes
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
out degree


proportion of edges
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
1
2
3
4
5
6
7
8
9
edge-wise shared partners


proportion of dyads
0.0
0.2
0.4
0.6
0.8
1
2
3
4
5
6
7
8
9
10
11
12
13
NR
minimum geodesic distance



The path to this model is that:

1. Gwesp term is most significant. Definitely we need to keep it. By increasing the decay coefficient, the model performs better.
2. At first, I select one from each of the 4 pairs, however, the istar and ostar terms do not contribute to the model. Thus, I remove them.
3. Ttriple and gwodeg is the more significant term in their pairs. I try to add ctriple, but in that case, the fitting process remains in the second round.
4. The in-degree is fitted badly at first. Then I add the gwideg term. Although its estimate is negative, the fitting of in-degree improves dramatically.
5. Tuning the decay parameters.

This model fits overall good, instead of the minimum geodesic distance. My approaches do not work on it.