Networks Short Course Homework

Patrick Rickert

Data Processing

```
setwd("~/Downloads")
load("nigeria.rda")

##Build an adjacency matrix from the data
length(unique(nigeria$receiver))

## [1] 37

mat<-matrix(0, nrow = 37, ncol = 37)
rownames(mat)<-sort(unique(nigeria$sender))
colnames(mat)<-sort(unique(nigeria$receiver))
slice = nigeria[nigeria$conflict==1,]
for (i in 1:nrow(slice)){
  a1 = as.character(slice[i, 'sender'])
  a2 = as.character(slice[i, 'receiver'])
  mat[a1,a2] = slice[i, 'conflict']

}
diag(mat)<-NA
```

Measurements and Community Detection

```
##Explore different centrality measures
library(igraph)

## Warning:  package 'igraph' was built under R version 3.4.1
##
## Attaching package:  'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
```

```
g = graph_from_adjacency_matrix(mat,
                                mode='directed')
igraph::degree(g)
```

```
## Area\nBoys\nMilitia      Bassa\nMilitia      Berom\nMilitia
##                    1                   1                   5
##   Christian\nMilitia      Ebira\nMilitia     Fulani\nMilitia
##                    8                   1                  22
##       Hausa\nMilitia       Igbo\nMilitia       Ijaw\nMilitia
##                   12                   6                   6
##      Ilajes\nMilitia   Itsekiri\nMilitia      Jukun\nMilitia
##                    0                   5                   5
##     Kanberi\nMilitia      Kuteb\nMilitia      Kutep\nMilitia
##                    1                   1                   1
##               MASSOB Military\n(Nigeria)     Muslim\nMilitia
##                    4                  19                   5
##       Ogoni\nMilitia                 OPC   Police\n(Nigeria)
##                    1                   4                  28
##      Shiite\nMilitia      Sunni\nMilitia        Tiv\nMilitia
##                    4                   2                   6
##         Udu\nMilitia     Urhobo\nMilitia      Uvwie\nMilitia
##                    1                   3                   1
##      Yoruba\nMilitia    Bakassi\nMilitia      Deken\nMilitia
##                    6                   2                   1
##       Deyor\nMilitia     Gbagyi\nMilitia      Tarok\nMilitia
##                    1                   3                   3
## Kalo\nKato\nMilitia  Vigilante\nMilitia                MEND
##                    2                   6                   4
##          Boko\nHaram
##                    7
```

```
eigen_centrality(g, directed = TRUE)
```

```
## $vector
## Area\nBoys\nMilitia      Bassa\nMilitia      Berom\nMilitia
##         1.399478e-01        4.114870e-18        3.997374e-01
##   Christian\nMilitia      Ebira\nMilitia     Fulani\nMilitia
##         4.792448e-01        0.000000e+00        8.345416e-01
```

```
##      Hausa\nMilitia       Igbo\nMilitia       Ijaw\nMilitia
##        5.346153e-01        4.495339e-01        3.625924e-01
##     Ilajes\nMilitia   Itsekiri\nMilitia      Jukun\nMilitia
##        0.000000e+00        1.733933e-01        2.898356e-01
##    Kanberi\nMilitia      Kuteb\nMilitia      Kutep\nMilitia
##        0.000000e+00        0.000000e+00        5.499463e-02
##              MASSOB Military\n(Nigeria)     Muslim\nMilitia
##        3.296920e-01        7.375600e-01        2.806782e-01
##      Ogoni\nMilitia                 OPC   Police\n(Nigeria)
##        0.000000e+00        1.897443e-01        1.000000e+00
##     Shiite\nMilitia      Sunni\nMilitia        Tiv\nMilitia
##        1.451745e-01        2.754602e-02        1.583495e-01
##        Udu\nMilitia     Urhobo\nMilitia      Uvwie\nMilitia
##        0.000000e+00        1.017002e-01        1.549474e-17
##     Yoruba\nMilitia    Bakassi\nMilitia      Deken\nMilitia
##        3.450861e-01        0.000000e+00        0.000000e+00
##      Deyor\nMilitia     Gbagyi\nMilitia      Tarok\nMilitia
##        0.000000e+00        2.597897e-01        2.982972e-01
## Kalo\nKato\nMilitia  Vigilante\nMilitia                MEND
##        0.000000e+00        4.259877e-01        3.296920e-01
##         Boko\nHaram
##        4.105207e-01
##
## $value
## [1] 5.270252
##
## $options
## $options$bmat
## [1] "I"
##
## $options$n
## [1] 37
##
## $options$which
## [1] "LR"
##
## $options$nev
## [1] 1
##
## $options$tol
## [1] 0
##
## $options$ncv
## [1] 0
```

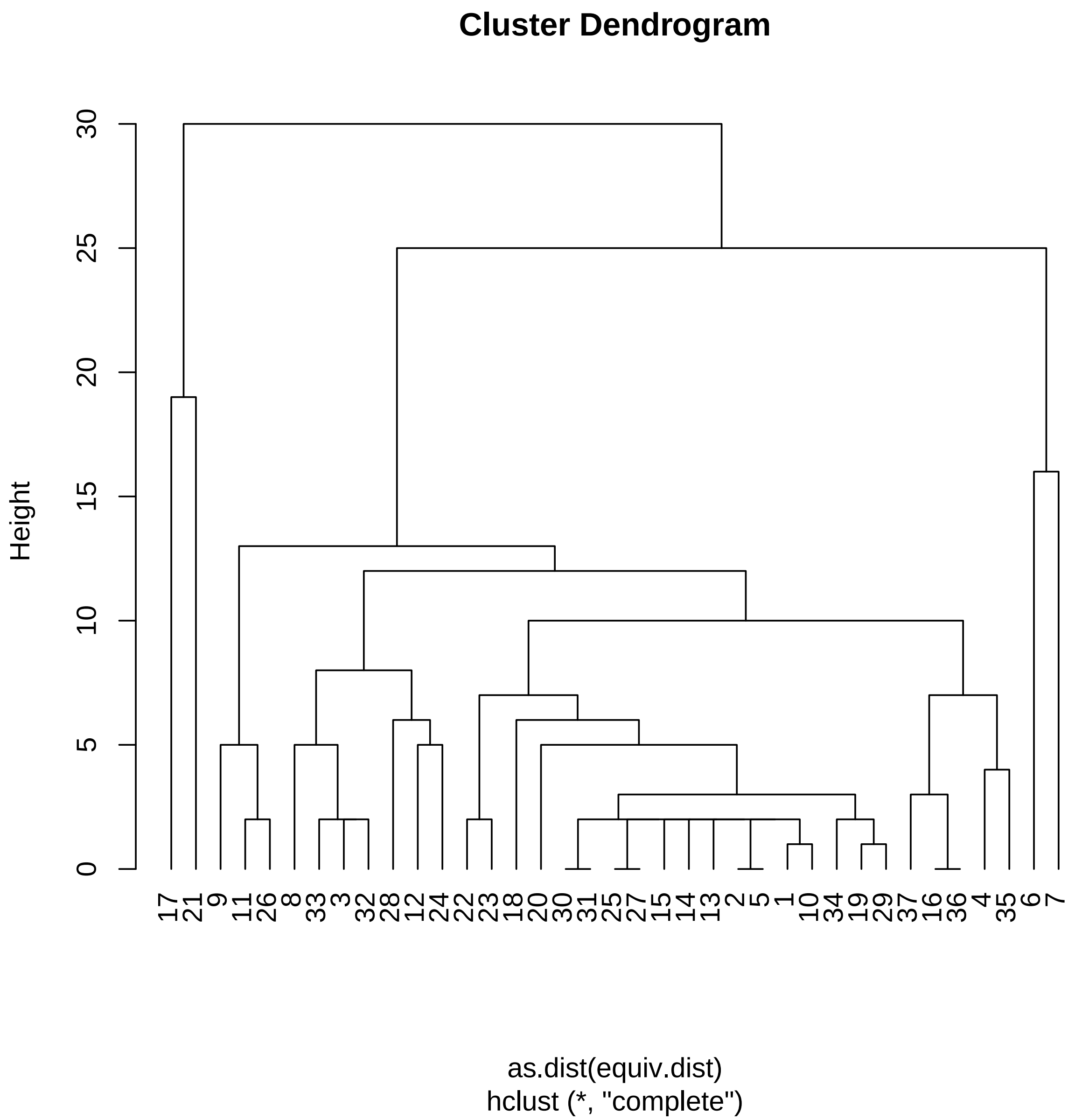
```
##
## $options$ldv
## [1] 0
##
## $options$ishift
## [1] 1
##
## $options$maxiter
## [1] 1000
##
## $options$nb
## [1] 1
##
## $options$mode
## [1] 1
##
## $options$start
## [1] 1
##
## $options$sigma
## [1] 0
##
## $options$sigmai
## [1] 0
##
## $options$info
## [1] 0
##
## $options$iter
## [1] 2
##
## $options$nconv
## [1] 1
##
## $options$numop
## [1] 27
##
## $options$numopb
## [1] 0
##
## $options$numreo
## [1] 22
```

```
igraph::betweenness(g)
```

```
## Area\nBoys\nMilitia      Bassa\nMilitia      Berom\nMilitia
##             0.000000             0.000000            13.400000
##  Christian\nMilitia      Ebira\nMilitia     Fulani\nMilitia
##             9.416667             0.000000           173.416667
##      Hausa\nMilitia       Igbo\nMilitia       Ijaw\nMilitia
##            80.966667            27.283333            73.250000
##     Ilajes\nMilitia   Itsekiri\nMilitia      Jukun\nMilitia
##             0.000000            29.750000            27.000000
##    Kanberi\nMilitia      Kuteb\nMilitia      Kutep\nMilitia
##             0.000000             0.000000             0.000000
##               MASSOB Military\n(Nigeria)     Muslim\nMilitia
##             0.000000           146.816667             0.500000
##      Ogoni\nMilitia                  OPC   Police\n(Nigeria)
##             0.000000             0.000000           298.266667
##     Shiite\nMilitia      Sunni\nMilitia        Tiv\nMilitia
##            49.000000             0.000000            24.000000
##        Udu\nMilitia     Urhobo\nMilitia      Uvwie\nMilitia
##             0.000000             0.000000             0.000000
##     Yoruba\nMilitia    Bakassi\nMilitia      Deken\nMilitia
##            31.183333             0.000000             0.000000
##      Deyor\nMilitia     Gbagyi\nMilitia      Tarok\nMilitia
##             0.000000             3.950000             0.000000
## Kalo\nKato\nMilitia  Vigilante\nMilitia                 MEND
##             0.000000             2.466667             0.000000
##          Boko\nHaram
##             2.333333
```

It is clear that using various different metrics of centrality, the most important actor in the network is the Nigerian Police. They have the most edges, meaning they were involved in the most conflicts, the highest eigenvector centrality, indicating that they were connected to the most connected nodes, and the highest betweenness score, meaning that the most shortest paths went through the police.

```
library(PRROC)

library(network,quietly=T)
require(sna)
## Create clusters based on structural equivalence
NigeriaNet <- network(mat, mode = "directed")
clusts <- equiv.clust(NigeriaNet)
## Plot dendrogram
plot(clusts,hang=-1)
```




```r
## Use K clusters to determine group membership
NigeriaBlock <- blockmodel(NigeriaNet, clusts, k=2)
bmems2 <- NigeriaBlock$block.membership[order(NigeriaBlock$order.vector)]
NigeriaBlock <- blockmodel(NigeriaNet, clusts, k=3)
bmems3 <- NigeriaBlock$block.membership[order(NigeriaBlock$order.vector)]
NigeriaBlock <- blockmodel(NigeriaNet, clusts, k=4)
bmems4 <- NigeriaBlock$block.membership[order(NigeriaBlock$order.vector)]
NigeriaBlock <- blockmodel(NigeriaNet, clusts, k=5)
bmems5 <- NigeriaBlock$block.membership[order(NigeriaBlock$order.vector)]
NigeriaBlock <- blockmodel(NigeriaNet, clusts, k=6)
bmems6 <- NigeriaBlock$block.membership[order(NigeriaBlock$order.vector)]
```

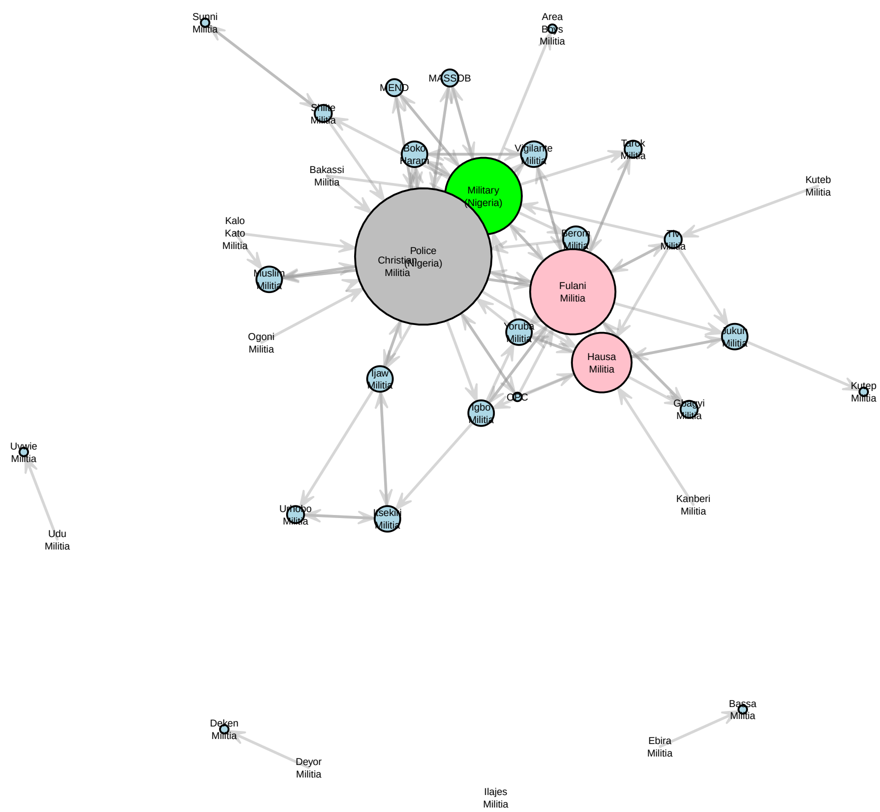
```r
##Make dataframe for testing
library(reshape2)
data<-melt(mat)
data$groupmembershipk2<-rep(bmems2, 37)
data$groupmembershipk3<-rep(bmems3, 37)
data$groupmembershipk4<-rep(bmems4, 37)
data$groupmembershipk5<-rep(bmems5, 37)
data$groupmembershipk6<-rep(bmems6, 37)
vec<-seq(1,1369, by = 38)
data<-data[-vec,]

library(cvTools)
folds <-cvFolds(nrow(data), K = 10)
data$predictk2 <- rep(0, nrow(data))
data$predictk3 <- rep(0, nrow(data))
data$predictk4 <- rep(0, nrow(data))
data$predictk5 <- rep(0, nrow(data))
data$predictk6 <- rep(0, nrow(data))
##Perform cross validation
set.seed(15)
for(i in 1:10){
  train <- data[folds$subsets[folds$which !=i],]
  test <- data[folds$subsets[folds$which == i],]

  mod2<- glm(value~groupmembershipk2, data = train, family = binomial(link="logit"))
  predict2 <-predict(mod2, newdata = test, type = "response")
  data[folds$subsets[folds$which == i],]$predictk2<- predict2

  mod3<- glm(value~groupmembershipk3, data = train, family = binomial(link="logit"))
  predict3 <-predict(mod3, newdata = test, type = "response")
  data[folds$subsets[folds$which == i],]$predictk3<- predict3

  mod4<- glm(value~groupmembershipk4, data = train, family = binomial(link="logit"))
  predict4 <-predict(mod4, newdata = test, type = "response")
  data[folds$subsets[folds$which == i],]$predictk4<- predict4

  mod5<- glm(value~groupmembershipk5, data = train, family = binomial(link="logit"))
  predict5 <-predict(mod5, newdata = test, type = "response")
  data[folds$subsets[folds$which == i],]$predictk5<- predict5

  mod6<- glm(value~as.factor(groupmembershipk6), data = train, family = binomial(link="l
  predict6 <-predict(mod6, newdata = test, type = "response")
```

```
  data[folds$subsets[folds$which == i],]$predictk6<- predict6
}

fg <- data$predictk2[data$value==1]
bg <- data$predictk2[data$value == 0]
roc2 <- roc.curve(scores.class0 = bg, scores.class1 = fg, curve = T)
pr2 <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)


fg <- data$predictk3[data$value==1]
bg <- data$predictk3[data$value == 0]
roc3 <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
pr3 <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)


fg <- data$predictk4[data$value==1]
bg <- data$predictk4[data$value == 0]
roc4 <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
pr4 <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)

fg <- data$predictk5[data$value==1]
bg <- data$predictk5[data$value == 0]
roc5 <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
pr5 <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)

fg <- data$predictk6[data$value==1]
bg <- data$predictk6[data$value == 0]
roc6 <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
pr6 <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)

### Check AUC
c(roc2$auc, roc3$auc, roc4$auc, roc5$auc, roc6$auc)

## [1] 0.4793378 0.6110705 0.6167635 0.6158397 0.6202179

c(pr2$auc.integral, pr3$auc.integral, pr4$auc.integral, pr5$auc.integral, pr6$auc.integr

## [1] 0.1109063 0.1540755 0.1632348 0.1594146 0.1525581
```

The AUC indicates that 4 or 5 groups should be used. I opt to use four, for simplicity's sake.

```
colVec <- c("lightblue", "pink", "green", "grey")

# Assign colors to individual nodes based on block membership
```

```
bcols <- colVec[bmems4]


#check colors and membership
head(cbind(bcols,bmems4))

##      bcols       bmems4
## [1,] "lightblue" "1"
## [2,] "lightblue" "1"
## [3,] "lightblue" "1"
## [4,] "lightblue" "1"
## [5,] "lightblue" "1"
## [6,] "pink"      "2"

set.seed(5)
# Now plot
vertexSize = degree(NigeriaNet, cmode = 'indegree')/2
plot(NigeriaNet,displaylabels=T,
     vertex.cex=vertexSize,label.cex=.35,edge.col=rgb(150,150,150,100,maxColorValue=255)
     label.pos=5,vertex.col=bcols)
```



ERGMs

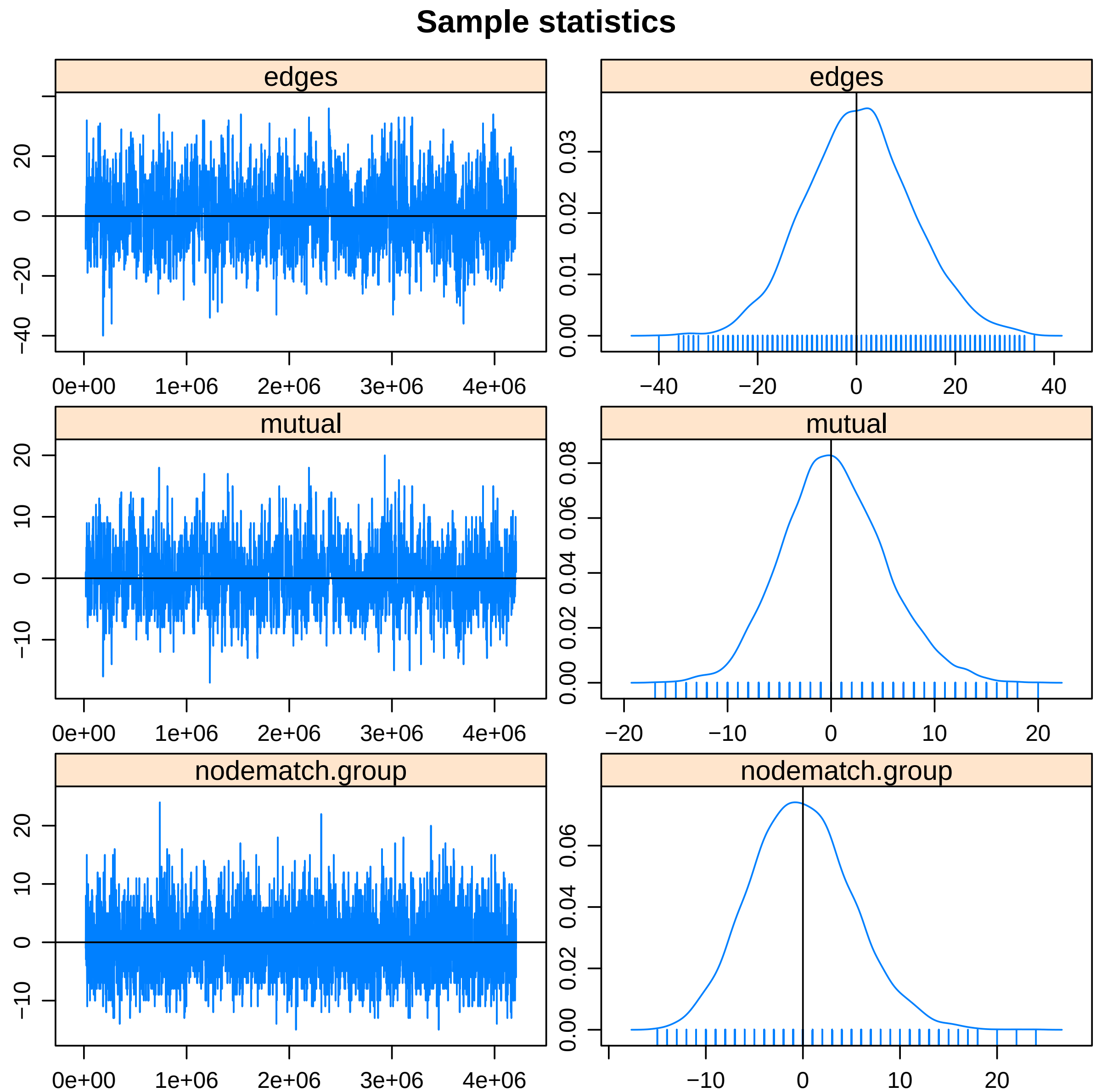
```
##Run ERGM
set.vertex.attribute(NigeriaNet, "group", bmems4)
library(btergm)
model<-ergm(NigeriaNet~edges + mutual + nodematch("group"))
summary(model)

##
## ==========================
```

```
## Summary of model fit
## ========================
##
## Formula:   NigeriaNet ~ edges + mutual + nodematch("group")
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##                 Estimate Std. Error MCMC % p-value
## edges            -2.1344     0.2132      0  <1e-04 ***
## mutual            2.9939     0.4141      0  <1e-04 ***
## nodematch.group  -1.9815     0.2463      0  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 1846.5  on 1332  degrees of freedom
##  Residual Deviance:  474.3  on 1329  degrees of freedom
##
## AIC: 480.3    BIC: 495.8    (Smaller is better.)
```

```r
##Check convergence
mcmc.diagnostics(model)
```

```
## 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            0.5945 10.873  0.16989        0.34516
## mutual           0.3103  4.936  0.07712        0.17493
## nodematch.group -0.1211  5.227  0.08167        0.09046
##
## 2. Quantiles for each variable:
##
##                 2.5% 25% 50% 75% 97.5%
## edges            -21  -7   0   8    23
## mutual            -9  -3   0   3    11
## nodematch.group  -10  -4   0   3    11
##
```

```
##
## Sample statistics cross-correlations:
##                     edges    mutual nodematch.group
## edges           1.0000000 0.8430040       0.5201927
## mutual          0.8430040 1.0000000       0.2478091
## nodematch.group 0.5201927 0.2478091       1.0000000
##
## Sample statistics auto-correlation:
## Chain 1
##              edges    mutual nodematch.group
## Lag 0    1.0000000 1.0000000      1.00000000
## Lag 1024 0.5045427 0.6260771      0.10176064
## Lag 2048 0.3485195 0.4428445      0.02507331
## Lag 3072 0.2366235 0.2934440      0.02709460
## Lag 4096 0.1598600 0.1984774      0.03134939
## Lag 5120 0.1085781 0.1366590      0.02333484
##
## Sample statistics burn-in diagnostic (Geweke):
## Chain 1
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##           edges          mutual nodematch.group
##          0.6227          0.6228         -0.3404
##
## Individual P-values (lower = worse):
##           edges          mutual nodematch.group
##       0.5334924       0.5334101       0.7335906
## Joint P-value (lower = worse):  0.7887937 .
```




```
##
## MCMC diagnostics shown here are from the last round of simulation, prior to computati
```

My hypotheses tested in the model are that there is a high degree of reciprocity in the network, which is to say that nodes that send ties to another node are more likely to also receive a tie from that node. Additionally, I expect the node match coefficient on the group variable to be negative, because I expect there to be more conflict between groups than within groups. For example, I expect the militia groups to engage in conflict with the military more than each other. The ERGM results bear this out, and the model diagnostics show that the model converged.