Assignment 3

Huang Yingshan, Kahlbacher Fabian, Wattenberg Yannick, Aggeler Samuel

11/25/2021

Task 2

```
# Loading the packages.
library(RSiena)
library(sna)
```

```
## Loading required package: statnet.common
```

```
##
## Attaching package: 'statnet.common'
```

```
## The following objects are masked from 'package:base':
##
##     attr, order
```

```
## Loading required package: network
```

```
##
## 'network' 1.17.1 (2021-06-12), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
## sna: Tools for Social Network Analysis
## Version 2.6 created on 2020-10-5.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.
```

```
#library(jaccard)
# Importing functions that are contained in .r files and will be useful for
# estimating the model and formatting the results
source("printSiena.R")
source("siena07ToConverge.R")

# loading data
# Importing the adjacency matrices of the networks
f1 <- as.matrix(read.csv("Glasgow/f1.csv", header = FALSE))
f2 <- as.matrix(read.csv("Glasgow/f2.csv", header = FALSE))
f3 <- as.matrix(read.csv("Glasgow/f3.csv", header = FALSE))
# Reading the demographic, alcohol comsuption and log distance characteristics of pupils
attributes <- read.csv("Glasgow/demographic.csv", header = TRUE)
logdistance <- as.matrix(read.csv("Glasgow/logdistance.csv", header = FALSE))
alcohol <- as.matrix(read.csv("Glasgow/alcohol.csv", header = TRUE))
```

1)

1.1)

```r
# Compute Jacard index
n <- nrow(f1)
j_12 <- sum(f1*f2)/(n^2-sum(f1+f2==0))
j_23 <- sum(f2*f3)/(n^2-sum(f2+f3==0))
j_12
```

```
## [1] 0.3036496
```

```r
j_23
```

```
## [1] 0.3506686
```

Jaccard index between wave 1 and wave 2 is 0.3036496, between wave 2 and wave 3 is 0.3506686. The Jaccard index is in each case higher than 0.3. Thus, it indicates that the data, for each wave, is sufficiently informative for applying SAOMs.

1.2)

```r
friendship <- sienaDependent(array(c(f1, f2, f3), dim = c(n, n, 3)))
gender <- coCovar(attributes$gender)
age <- coCovar(attributes$age)
alcoholConsumption <- varCovar(alcohol)
logDistance <- coDyadCovar(logdistance)
mydata <- sienaDataCreate(
  friendship,
  gender,
  age,
  alcoholConsumption,
  logDistance
)

# Data description
print01Report(mydata, modelname = "glasgow_init")
```

```r
# Model specification

# Include basic endogenous variables
myeff <- getEffects(mydata)

myeff <- includeEffects(
  myeff,
  density,
  recip,
  transTrip,
  transTies,
  cycle3,
  balance,
  outActSqrt
)
```

```
##   effectName                  include fix   test  initialValue parm
## 1 outdegree (density)         TRUE    FALSE FALSE   -1.61299   0
```

```
## 2 reciprocity                 TRUE    FALSE FALSE    0.00000   0
## 3 transitive triplets         TRUE    FALSE FALSE    0.00000   0
## 4 3-cycles                    TRUE    FALSE FALSE    0.00000   0
## 5 transitive ties             TRUE    FALSE FALSE    0.00000   0
## 6 balance                     TRUE    FALSE FALSE    0.00000   0
## 7 outdegree - activity (sqrt) TRUE    FALSE FALSE    0.00000   1
```

```
# Include basic exogenous variables
myeff <- includeEffects(myeff, egoX, altX, sameX, interaction1="gender")
```

```
##   effectName   include fix   test  initialValue parm
## 1 gender alter TRUE    FALSE FALSE          0   0
## 2 gender ego   TRUE    FALSE FALSE          0   0
## 3 same gender  TRUE    FALSE FALSE          0   0
```

```
# H1: Students tend to be friends with popular pupils
myeff <- includeEffects(myeff, inPopSqrt) # outPopSqrt
```

```
##   effectName                   include fix   test  initialValue parm
## 1 indegree - popularity (sqrt) TRUE    FALSE FALSE          0   0
```

```
# H2: Students tend to be friends with pupils with similar alcohol consumption to their own
myeff <- includeEffects(myeff, sameX, interaction1 ="alcoholConsumption")
```

```
##   effectName              include fix   test  initialValue parm
## 1 same alcoholConsumption TRUE    FALSE FALSE          0   0
```

```
# H3: Students tend to be friends with students that live in the same neighborhood (living nearby)
myeff <- includeEffects(myeff, X, interaction1 = "logDistance")
```

```
##   effectName  include fix   test  initialValue parm
## 1 logDistance TRUE    FALSE FALSE          0   0
```

```
# Estimate the model
myAlgorithm <- sienaAlgorithmCreate(
  projname="friends_res",
  nsub = 4, n3 = 3000, seed = 1908
)
```

```
## If you use this algorithm object, siena07 will create/use an output file friends_res.txt .
```

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4
)
```

```
t_conv.max <- function(model) {
  sqrt(
  t(apply(model$sf, 2, mean)) %*%
  solve(cov(model$sf)) %*%
  apply(model$sf, 2, mean)
 )
}
```

t-conv_k

```
printSiena(model.ev)
```

```
##     dependent                                effect  theta  s.e. p.value sig.
## 1        rate constant friendship rate (period 1) 12.880 1.338
## 2        rate constant friendship rate (period 2) 10.151 0.970
## 3  friendship                 outdegree (density) -4.881 0.633       0  ***
## 4  friendship                         reciprocity  1.767 0.112       0  ***
## 5  friendship                 transitive triplets  0.002 0.097   0.983
## 6  friendship                            3-cycles -0.161 0.088   0.067    .
## 7  friendship                     transitive ties  0.699 0.094       0  ***
## 8  friendship                             balance  0.256 0.043       0  ***
## 9  friendship        indegree - popularity (sqrt)  0.168 0.094   0.074    .
## 10 friendship         outdegree - activity (sqrt)  0.629 0.209   0.003   **
## 11 friendship                         logDistance -0.206 0.047       0  ***
## 12 friendship                        gender alter -0.145 0.082   0.077    .
## 13 friendship                          gender ego  0.050 0.105   0.631
## 14 friendship                         same gender  0.679 0.087       0  ***
## 15 friendship             same alcoholConsumption  0.164 0.068   0.015    *
##    t.conv
## 1
## 2
## 3   0.136
## 4   0.003
## 5   0.024
## 6   0.029
## 7   0.003
## 8  -0.052
## 9   0.045
## 10 -0.016
## 11 -0.018
## 12 -0.018
## 13  -0.02
## 14  0.039
## 15  0.001
```

all good everything <= 0.1

```
t_conv.max(model.ev)
```

```
##          [,1]
## [1,] 2.052842
```

not good 0.3297 (should be <= 0.2)

fit it again

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4,
  prevAns = model.ev
)
```

t-conv_k

```r
printSiena(model.ev)
```

```
##     dependent                                effect  theta  s.e. p.value sig.
## 1        rate constant friendship rate (period 1) 12.747 1.344
## 2        rate constant friendship rate (period 2) 10.083 0.996
## 3  friendship                  outdegree (density) -5.712 0.580       0  ***
## 4  friendship                          reciprocity  1.796 0.114       0  ***
## 5  friendship                  transitive triplets -0.083 0.092   0.367
## 6  friendship                             3-cycles -0.156 0.093   0.095    .
## 7  friendship                      transitive ties  0.689 0.094       0  ***
## 8  friendship                              balance  0.302 0.039       0  ***
## 9  friendship         indegree - popularity (sqrt)  0.249 0.087   0.004   **
## 10 friendship          outdegree - activity (sqrt)  0.895 0.191       0  ***
## 11 friendship                          logDistance -0.208 0.047       0  ***
## 12 friendship                         gender alter -0.151 0.084   0.073    .
## 13 friendship                           gender ego  0.063 0.105   0.547
## 14 friendship                          same gender  0.702 0.089       0  ***
## 15 friendship             same alcoholConsumption  0.171 0.066    0.01   **
##    t.conv
## 1
## 2
## 3   0.053
## 4   0.039
## 5   0.062
## 6   0.035
## 7   0.013
## 8   0.062
## 9    0.03
## 10   0.01
## 11 -0.009
## 12 -0.028
## 13 -0.024
## 14  0.023
## 15   0.03
```

all good everything <= 0.1

```r
t_conv.max(model.ev)
```

```
##            [,1]
## [1,] 0.5655077
```

still not good, 0.5655077

fit it again

```r
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4,
  prevAns = model.ev
)
```

```r
printSiena(model.ev)
```

```
##     dependent                                  effect  theta  s.e. p.value sig.
## 1        rate constant friendship rate (period 1) 12.730 1.341
## 2        rate constant friendship rate (period 2) 10.062 0.957
## 3  friendship                   outdegree (density) -5.813 0.621       0  ***
## 4  friendship                           reciprocity  1.784 0.114       0  ***
## 5  friendship                   transitive triplets -0.103 0.099   0.299
## 6  friendship                              3-cycles -0.139 0.089    0.12
## 7  friendship                       transitive ties  0.700 0.101       0  ***
## 8  friendship                               balance  0.308 0.042       0  ***
## 9  friendship          indegree - popularity (sqrt)  0.258 0.089   0.004   **
## 10 friendship          outdegree - activity (sqrt)  0.930 0.205       0  ***
## 11 friendship                           logDistance -0.209 0.047       0  ***
## 12 friendship                          gender alter -0.150 0.084   0.073    .
## 13 friendship                            gender ego  0.061 0.105   0.561
## 14 friendship                           same gender  0.704 0.084       0  ***
## 15 friendship             same alcoholConsumption  0.173 0.065   0.008   **
##    t.conv
## 1
## 2
## 3   0.004
## 4  -0.013
## 5  -0.002
## 6   0.001
## 7  -0.012
## 8  -0.048
## 9  -0.011
## 10 -0.012
## 11  -0.02
## 12 -0.049
## 13 -0.064
## 14 -0.003
## 15 -0.045
```

```r
t_conv.max(model.ev)
```

```
##           [,1]
## [1,] 0.3331558
```

```r
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4,
  prevAns = model.ev
)
```

```r
printSiena(model.ev)
```

```
##     dependent                                  effect  theta  s.e. p.value sig.
## 1        rate constant friendship rate (period 1) 12.689 1.315
## 2        rate constant friendship rate (period 2) 10.027 0.971
```

```
## 3  friendship                 outdegree (density) -5.976 0.565       0  ***
## 4  friendship                         reciprocity  1.792 0.119       0  ***
## 5  friendship                 transitive triplets -0.120 0.089   0.179
## 6  friendship                            3-cycles -0.140 0.087   0.109
## 7  friendship                     transitive ties  0.702 0.098       0  ***
## 8  friendship                             balance  0.317 0.039       0  ***
## 9  friendship         indegree - popularity (sqrt)  0.272 0.086   0.001   **
## 10 friendship         outdegree - activity (sqrt)  0.983 0.185       0  ***
## 11 friendship                         logDistance -0.211 0.047       0  ***
## 12 friendship                        gender alter -0.152 0.083   0.065    .
## 13 friendship                          gender ego  0.068 0.109    0.53
## 14 friendship                         same gender  0.708 0.086       0  ***
## 15 friendship            same alcoholConsumption  0.177 0.069   0.011    *
##    t.conv
## 1
## 2
## 3   0.032
## 4   0.023
## 5   0.049
## 6    0.03
## 7   0.032
## 8   0.054
## 9    0.03
## 10   0.03
## 11  -0.04
## 12 -0.009
## 13 -0.006
## 14  0.042
## 15   0.04
```

```
t_conv.max(model.ev)
```

```
##             [,1]
## [1,] 0.1195636
```

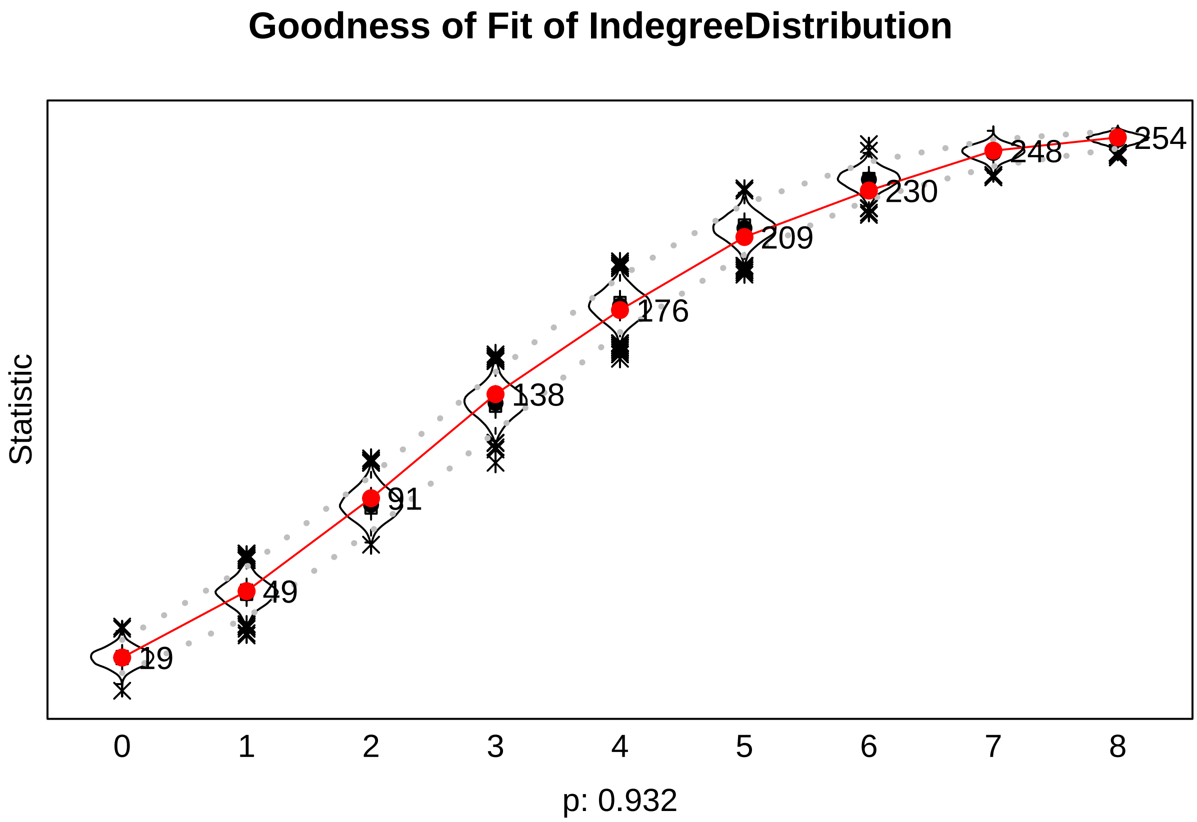
now good: 0.1195636

1.3)

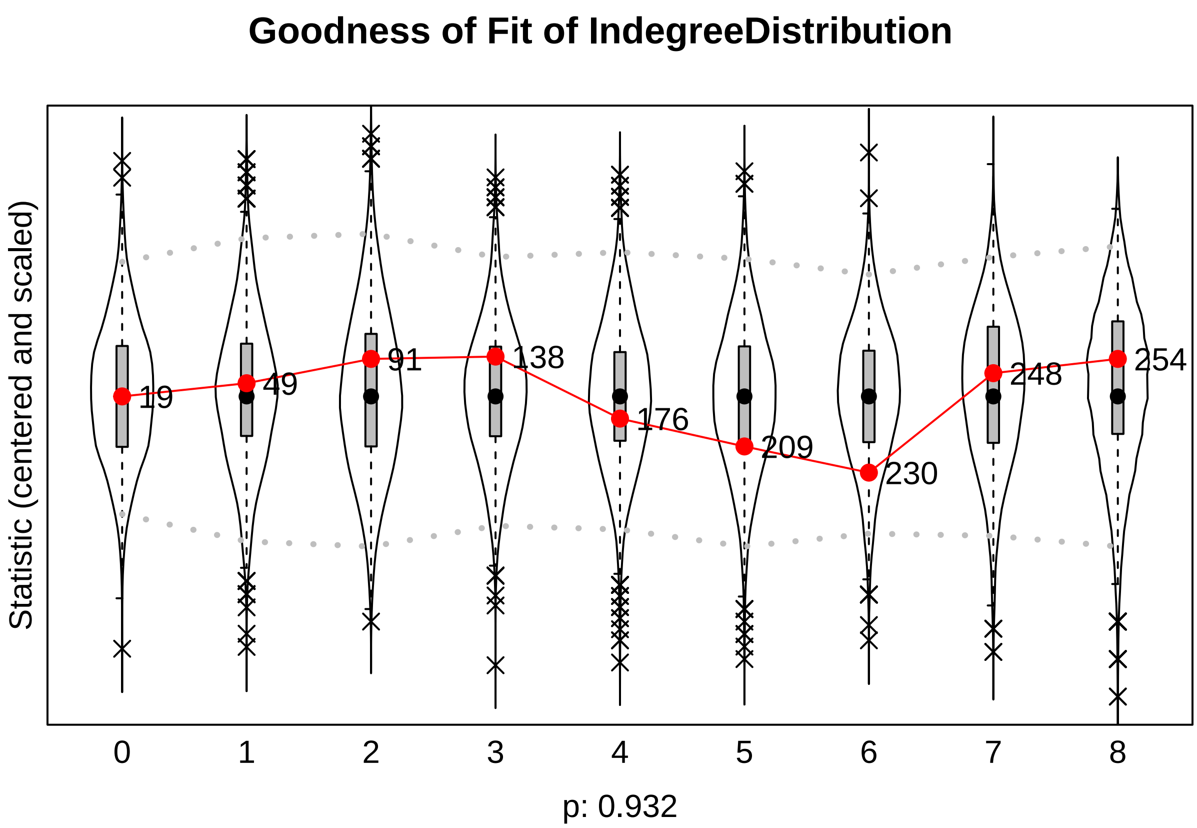
```
# Goodness of Fit
# Indegree distribution
gof1.id <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  IndegreeDistribution
)

# Outdegree distribution
gof1.od <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  OutdegreeDistribution
)

# Triad census
gof1.tc <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
```

```
  TriadCensus
)

# Geodesic distance
GeodesicDistribution <- function(i, data, sims, period, groupName,
                                 varName, levls = c(1:5, Inf), cumulative = TRUE) {
  x <- networkExtraction(i, data, sims, period, groupName, varName)
  require(sna)
  a <- sna::geodist(symmetrize(x))$gdist
  if (cumulative) {
    gdi <- sapply(levls, function(i) {
      sum(a <= i)
    })
  }
  else {
    gdi <- sapply(levls, function(i) {
      sum(a == i)
    })
  }
  names(gdi) <- as.character(levls)
  gdi
}

gof1.gd <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  GeodesicDistribution
)
```

```
plot(gof1.id)
```

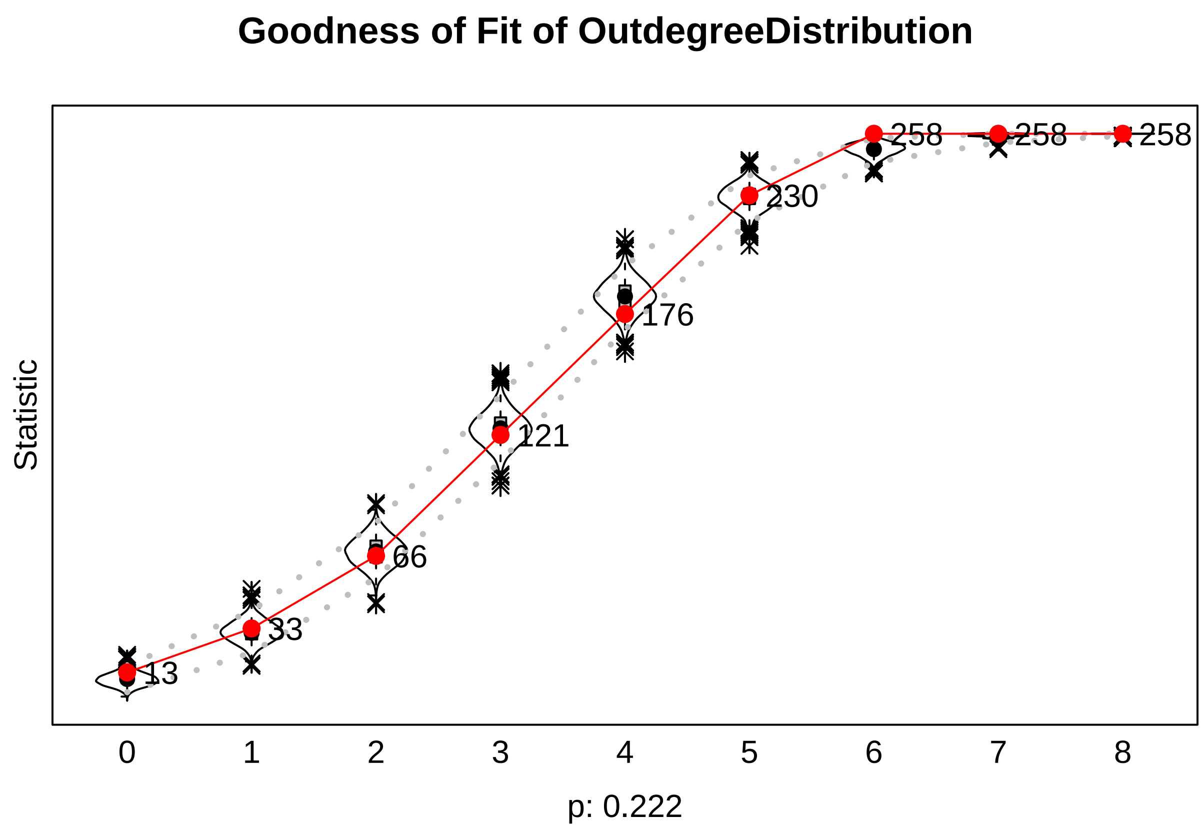



```
plot(gof1.id, center = TRUE, scale = TRUE)
```



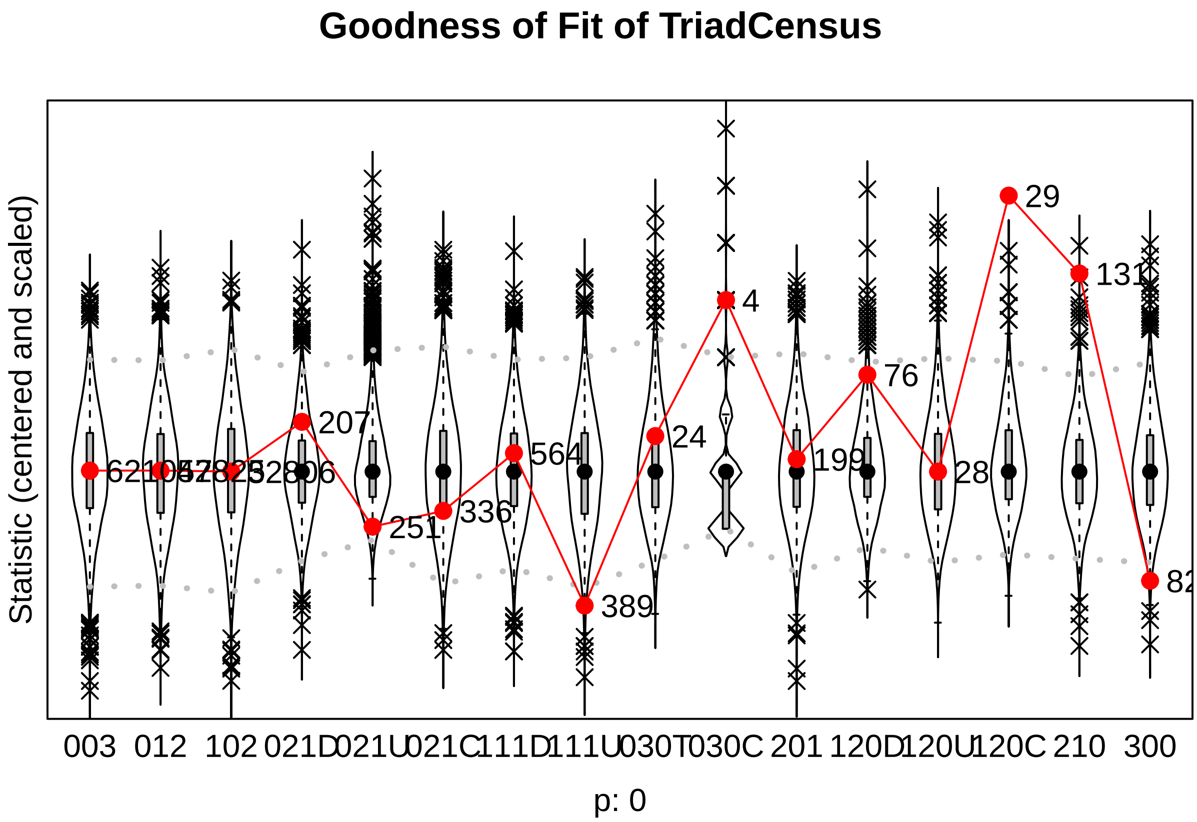
All in-degrees from 0 to 8 are in the 95% confidence interval. Furthermore, the p-value is greater than 0.05 which indicates a good fit (as mentioned in the lecture).

```
plot(gof1.od)
```



All out-degrees from 0 to 8 are in the 95% confidence interval. Furthermore, the p-value is greater than 0.05 which indicates a good fit.

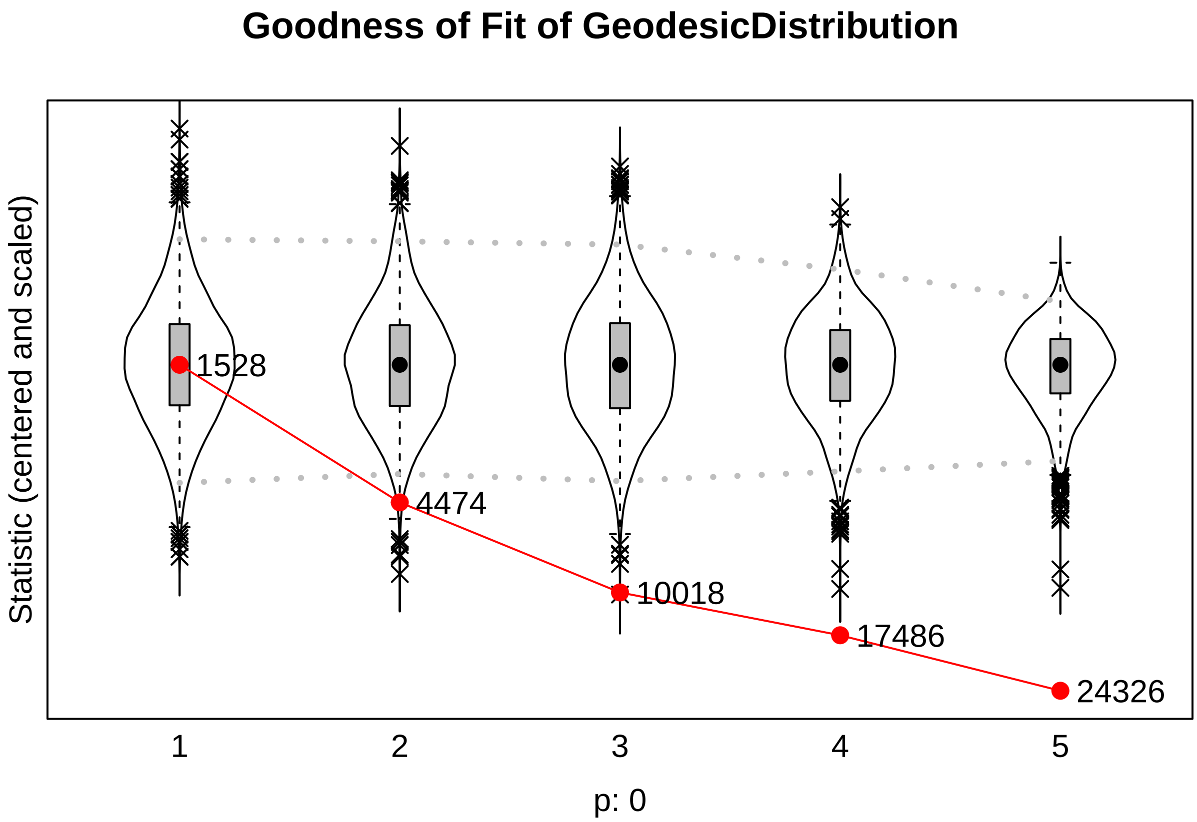
```
plot(gof1.tc, center = TRUE, scale = TRUE)
```



GOF for triad census does not fit very well. The observed triad census for the following states 111U, 030C, 120C, 210 and 300 lies outside the 95%-confidence interval. Moreover, the p-value is equal to zero which indicates a poor fit.

```
plot(gof1.gd, center = TRUE, scale = TRUE)
```

```
## Note: some statistics are not plotted because their variance is 0.
## This holds for the statistic: Inf.
```



The GOF for the Geodesic Distance does not fit well. Only the Geodesic Distance of one lies in the 95%-confidence interval the distances 2 to 5 are outside. Moreover, the p-value of zero indicates a poor fit.

To conclude, the in and out-degree fit well but the model does not accurately represents Triadcensus and Geodesic Distance. Thus, the significance of the assessments based on this model should be interpreted carefully.

To fit out-degree well we tried to include different endogenous variables. The exogeneous variables were alway the same as specified in the code. Our different trials were: 1. endogenous: transTrip, cycle3 2. endogenous: density, recip, transTrip, cycle3 3. engogenous: density, recip, transTrip, transTies, cycle3, balance, outActSqrt The 3. combination of endogeneous variables worked well to fit out-degree.

1.4

```
printSiena(model.ev)
```

```
##     dependent                              effect  theta  s.e. p.value sig.
## 1        rate constant friendship rate (period 1) 12.689 1.315
## 2        rate constant friendship rate (period 2) 10.027 0.971
## 3  friendship                 outdegree (density) -5.976 0.565       0  ***
## 4  friendship                         reciprocity  1.792 0.119       0  ***
## 5  friendship                 transitive triplets -0.120 0.089   0.179
## 6  friendship                            3-cycles -0.140 0.087   0.109
## 7  friendship                     transitive ties  0.702 0.098       0  ***
## 8  friendship                             balance  0.317 0.039       0  ***
## 9  friendship        indegree - popularity (sqrt)  0.272 0.086   0.001   **
## 10 friendship         outdegree - activity (sqrt)  0.983 0.185       0  ***
## 11 friendship                         logDistance -0.211 0.047       0  ***
```

```
## 12 friendship                        gender alter -0.152 0.083   0.065    .
## 13 friendship                          gender ego  0.068 0.109    0.53
## 14 friendship                         same gender  0.708 0.086       0  ***
## 15 friendship             same alcoholConsumption  0.177 0.069   0.011    *
##    t.conv
## 1
## 2
## 3   0.032
## 4   0.023
## 5   0.049
## 6    0.03
## 7   0.032
## 8   0.054
## 9    0.03
## 10   0.03
## 11  -0.04
## 12 -0.009
## 13 -0.006
## 14  0.042
## 15   0.04
```

The estimates of the coefficients and the corresponding standard errors are obtained from a model controlling for basic endogenous and exogenous variables (all variables can be seen in the summary output above).

H1: Students tend to be friends with popular pupils

The coefficient for "indegree - popularity (sqrt)" is significantly different from 0 on the 5% significance level. The coefficient is positive which means that actors are more likely to name popular (high in-degree) students as their friends. Hence, H1 is supported.

H2: Students tend to be friends with pupils with similar alcohol consumption to their own

The coefficient for "same alcoholConsumption" is significantly different from 0 on the 5% significance level. The coefficient is positive which means that an actor is more likely to name students as their friends if they have the same level of alcohol consumption. Same level of alcohol consumption is a proxy for similar level of alcohol consumption hence, H2 is supported.

H3: Students tend to be friends with students that live in the same neighborhood (living nearby)

The coefficient for "logDistance" is significantly different from 0 on the 5% significance level. The coefficient is negative which means that actors are less likely to make ties to students which are further away. Hence, H3 is supported.

2)

```
friendship <- sienaDependent(array(c(f1, f2, f3), dim = c(n, n, 3)))
drinkbeh <- sienaDependent(alcohol, type = "behavior")

gender <- coCovar(attributes$gender)
age <- coCovar(attributes$age)
logDistance <- coDyadCovar(logdistance)

mydata2 <- sienaDataCreate(
  friendship,
  drinkbeh,
  gender,
  age,
```

```
  logDistance
)

mydata2
```

```
## Dependent variables:  friendship, drinkbeh
## Number of observations: 3
##
## Nodeset                  Actors
## Number of nodes             129
##
## Dependent variable friendship
## Type               oneMode
## Observations       3
## Nodeset            Actors
## Densities          0.027 0.027 0.028
##
## Dependent variable drinkbeh
## Type               behavior
## Observations       3
## Nodeset            Actors
## Range              0 - 5
##
## Constant covariates:  gender, age
## Constant dyadic covariates:  logDistance
```

```
# Data description
print01Report(mydata2, modelname = "glasgow_coEvol")
```

The Jaccard Index for the first wave (0.304) and second wave (0.351) are at least 0.3 thus, the data is informative.

```
moran1 <- nacf(f1, alcohol[, 1], lag.max = 1, type = "moran",
neighborhood.type = "out", mode = "digraph")
moran2 <- nacf(f2, alcohol[, 2], lag.max = 1, type = "moran",
neighborhood.type = "out", mode = "digraph")
moran3 <- nacf(f3, alcohol[, 3], lag.max = 1, type = "moran",
neighborhood.type = "out", mode = "digraph")
autocorr <- rbind(moran1, moran2, moran3)
autocorr[, 2]
```

```
##    moran1    moran2    moran3
## 0.2450651 0.3394460 0.3316509
```

The Moran Index is about 0.3 in all waves which is low but indicates some behavioral structure in the network.

2.1)

```
# Model specification
# Include basic endogenous
myeff <- getEffects(mydata2)

myeff <- includeEffects(
  myeff,
  density,
```

```
  recip,
  transTrip,
  transTies,
  cycle3,
  balance,
  outActSqrt
)
```

```
##   effectName                     include fix   test  initialValue parm
## 1 outdegree (density)            TRUE    FALSE FALSE   -1.61299   0
## 2 reciprocity                    TRUE    FALSE FALSE    0.00000   0
## 3 transitive triplets            TRUE    FALSE FALSE    0.00000   0
## 4 3-cycles                       TRUE    FALSE FALSE    0.00000   0
## 5 transitive ties                TRUE    FALSE FALSE    0.00000   0
## 6 balance                        TRUE    FALSE FALSE    0.00000   0
## 7 outdegree - activity (sqrt) TRUE    FALSE FALSE    0.00000   1
```

```
# Include basic exogenous variables
myeff <- includeEffects(myeff, egoX, altX, sameX, interaction1="gender")
```

```
##   effectName   include fix   test  initialValue parm
## 1 gender alter TRUE    FALSE FALSE          0    0
## 2 gender ego   TRUE    FALSE FALSE          0    0
## 3 same gender  TRUE    FALSE FALSE          0    0
```

```
# H1: Students tend to be friends with popular pupils
myeff <- includeEffects(myeff, inPopSqrt) # outPopSqrt
```

```
##   effectName                   include fix   test  initialValue parm
## 1 indegree - popularity (sqrt) TRUE    FALSE FALSE          0    0
```

```
# H2: Students tend to be friends with pupils with similar alcohol consumption to their own
myeff <- includeEffects(myeff, sameX, interaction1 ="drinkbeh")
```

```
##   effectName    include fix   test  initialValue parm
## 1 same drinkbeh TRUE    FALSE FALSE          0    0
```

```
# H3: Students tend to be friends with students that live in the same neighborhood (living nearby)
myeff <- includeEffects(myeff, X, interaction1 = "logDistance")
```

```
##   effectName  include fix   test  initialValue parm
## 1 logDistance TRUE    FALSE FALSE          0    0
```

```
# H4: Popular students tend to increase or maintain their level of alcohol consumption
myeff <- includeEffects(myeff, outdeg, indeg,
                        name = "drinkbeh", interaction1 = "friendship")
```

```
##   effectName         include fix   test  initialValue parm
## 1 drinkbeh indegree  TRUE    FALSE FALSE          0    0
## 2 drinkbeh outdegree TRUE    FALSE FALSE          0    0
```

```
# H5:  Students tend to adjust their alcohol consumption to that of their friends
myeff <- includeEffects(myeff, avSim,
                        name = "drinkbeh", interaction1="friendship")
```

```
##   effectName                  include fix   test  initialValue parm
## 1 drinkbeh average similarity TRUE    FALSE FALSE          0    0
```

2.2)

```r
# Estimate the model
myAlgorithm <- sienaAlgorithmCreate(
  projname="friends_res",
  nsub = 4, n3 = 1000, seed = 1908
)
```

```
## If you use this algorithm object, siena07 will create/use an output file friends_res.txt .
```

```r
model.ev <- siena07(
  myAlgorithm,
  data=mydata2,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4
)
```

```r
model.ev
```

```
## Estimates, standard errors and convergence t-ratios
##
##                                              Estimate   Standard   Convergence
##                                                           Error      t-ratio
## Network Dynamics
##    1. rate constant friendship rate (period 1) 12.6022  ( 1.6604   )    0.0310
##    2. rate constant friendship rate (period 2) 10.0242  ( 0.8562   )   -0.0137
##    3. eval outdegree (density)                 -5.0558  ( 0.8603   )    0.0628
##    4. eval reciprocity                          1.7273  ( 0.1216   )   -0.0388
##    5. eval transitive triplets                 -0.0091  ( 0.1236   )   -0.0146
##    6. eval 3-cycles                            -0.1543  ( 0.0893   )   -0.0174
##    7. eval transitive ties                      0.6908  ( 0.1069   )   -0.0212
##    8. eval balance                              0.2613  ( 0.0553   )   -0.0557
##    9. eval indegree - popularity (sqrt)         0.1723  ( 0.1248   )   -0.0113
##   10. eval outdegree - activity (sqrt)          0.6607  ( 0.2642   )   -0.0745
##   11. eval logDistance                         -0.2136  ( 0.0506   )   -0.0549
##   12. eval gender alter                        -0.1422  ( 0.0898   )   -0.0081
##   13. eval gender ego                           0.0496  ( 0.1046   )   -0.0473
##   14. eval same gender                          0.6792  ( 0.0831   )    0.0051
##   15. eval same drinkbeh                        0.5014  ( 0.2238   )    0.0057
##
## Behavior Dynamics
##   16. rate rate drinkbeh (period 1)             1.6604  ( 0.2980   )    0.0342
##   17. rate rate drinkbeh (period 2)             2.3457  ( 0.5521   )    0.0203
##   18. eval drinkbeh linear shape                0.3246  ( 0.3756   )   -0.0291
##   19. eval drinkbeh quadratic shape             0.0253  ( 0.0688   )    0.0595
##   20. eval drinkbeh average similarity          6.5017  ( 1.9046   )   -0.0165
##   21. eval drinkbeh indegree                    0.1185  ( 0.0966   )   -0.0039
##   22. eval drinkbeh outdegree                  -0.1006  ( 0.1498   )   -0.0479
##
## Overall maximum convergence ratio:    2.0040
##
##
## Total of 2722 iteration steps.
```

All convergence t-ratios are below 0.1 but the overall is larger than 0.2 (2.0040).

```
myAlgorithm <- sienaAlgorithmCreate(
  projname="friends_res",
  nsub = 4, n3 = 3000, seed = 1908
)
```

```
## If you use this algorithm object, siena07 will create/use an output file friends_res.txt .
```

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata2,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4,
  prevAns = model.ev
)
```

```
model.ev
```

```
## Estimates, standard errors and convergence t-ratios
##
##                                                 Estimate   Standard   Convergence
##                                                              Error      t-ratio
## Network Dynamics
##    1. rate constant friendship rate (period 1) 12.4140  ( 1.1504   )   -0.0283
##    2. rate constant friendship rate (period 2)  9.9685  ( 0.9940   )    0.0205
##    3. eval outdegree (density)                 -5.9195  ( 0.6101   )   -0.0296
##    4. eval reciprocity                          1.7417  ( 0.1198   )   -0.0672
##    5. eval transitive triplets                 -0.1077  ( 0.0957   )   -0.0255
##    6. eval 3-cycles                            -0.1295  ( 0.0921   )   -0.0278
##    7. eval transitive ties                      0.6879  ( 0.1003   )   -0.0587
##    8. eval balance                              0.3092  ( 0.0416   )   -0.0048
##    9. eval indegree - popularity (sqrt)         0.2575  ( 0.0879   )   -0.0399
##   10. eval outdegree - activity (sqrt)          0.9411  ( 0.1970   )   -0.0524
##   11. eval logDistance                         -0.2126  ( 0.0505   )    0.0649
##   12. eval gender alter                        -0.1466  ( 0.0784   )    0.0034
##   13. eval gender ego                           0.0602  ( 0.0993   )    0.0206
##   14. eval same gender                          0.6974  ( 0.0851   )   -0.0381
##   15. eval same drinkbeh                        0.5087  ( 0.2034   )   -0.0516
##
## Behavior Dynamics
##   16. rate rate drinkbeh (period 1)             1.6815  ( 0.2738   )    0.0930
##   17. rate rate drinkbeh (period 2)             2.3482  ( 0.4276   )   -0.0069
##   18. eval drinkbeh linear shape                0.2908  ( 0.3274   )   -0.0004
##   19. eval drinkbeh quadratic shape             0.0244  ( 0.0677   )   -0.0110
##   20. eval drinkbeh average similarity          6.5181  ( 1.8879   )    0.0268
##   21. eval drinkbeh indegree                    0.1071  ( 0.1017   )   -0.0035
##   22. eval drinkbeh outdegree                  -0.0797  ( 0.1357   )   -0.0102
##
## Overall maximum convergence ratio:    0.3588
##
##
## Total of 4154 iteration steps.
```

All convergence t-ratios are below 0.1 but the overall is larger than 0.2 (0.3588).

```r
model.ev <- siena07(
  myAlgorithm,
  data=mydata2,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 4,
  prevAns = model.ev
)
```

```r
model.ev
```

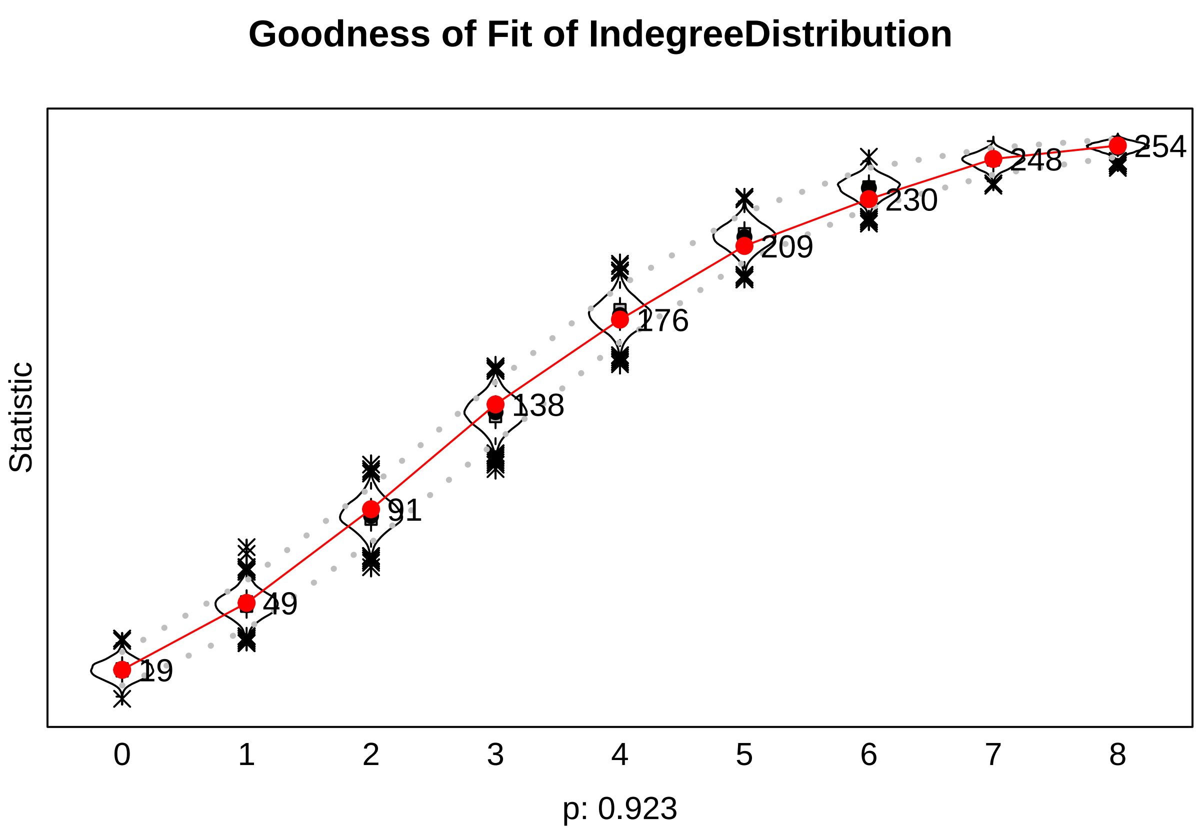
```
## Estimates, standard errors and convergence t-ratios
##
##                                                 Estimate   Standard   Convergence
##                                                              Error      t-ratio
## Network Dynamics
##    1. rate constant friendship rate (period 1) 12.3849  ( 1.4493   )   -0.0118
##    2. rate constant friendship rate (period 2)  9.9048  ( 1.0066   )   -0.0323
##    3. eval outdegree (density)                 -6.0408  ( 0.5976   )   -0.0091
##    4. eval reciprocity                          1.7435  ( 0.1204   )   -0.0126
##    5. eval transitive triplets                 -0.1221  ( 0.0970   )   -0.0028
##    6. eval 3-cycles                            -0.1256  ( 0.0920   )   -0.0007
##    7. eval transitive ties                      0.6972  ( 0.1019   )    0.0126
##    8. eval balance                              0.3158  ( 0.0415   )   -0.0037
##    9. eval indegree - popularity (sqrt)         0.2652  ( 0.0852   )   -0.0210
##   10. eval outdegree - activity (sqrt)          0.9788  ( 0.1987   )   -0.0103
##   11. eval logDistance                         -0.2161  ( 0.0499   )    0.0132
##   12. eval gender alter                        -0.1480  ( 0.0872   )   -0.0144
##   13. eval gender ego                           0.0620  ( 0.1031   )   -0.0116
##   14. eval same gender                          0.6997  ( 0.0869   )   -0.0187
##   15. eval same drinkbeh                        0.5303  ( 0.2254   )    0.0116
##
## Behavior Dynamics
##   16. rate rate drinkbeh (period 1)             1.6700  ( 0.2833   )    0.0268
##   17. rate rate drinkbeh (period 2)             2.3585  ( 0.3788   )    0.0208
##   18. eval drinkbeh linear shape                0.2683  ( 0.3351   )    0.0038
##   19. eval drinkbeh quadratic shape             0.0233  ( 0.0630   )   -0.0016
##   20. eval drinkbeh average similarity          6.4883  ( 1.8271   )    0.0041
##   21. eval drinkbeh indegree                    0.1066  ( 0.1049   )    0.0226
##   22. eval drinkbeh outdegree                  -0.0733  ( 0.1401   )    0.0162
##
## Overall maximum convergence ratio:    0.1359
##
##
## Total of 4098 iteration steps.
```

All convergence t-ratios are below 0.1 and the overall is below 0.2 (0.1359). The model converged well.

```r
# Goodness of Fit
# Indegree distribution
gof1.id <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  IndegreeDistribution
)
```

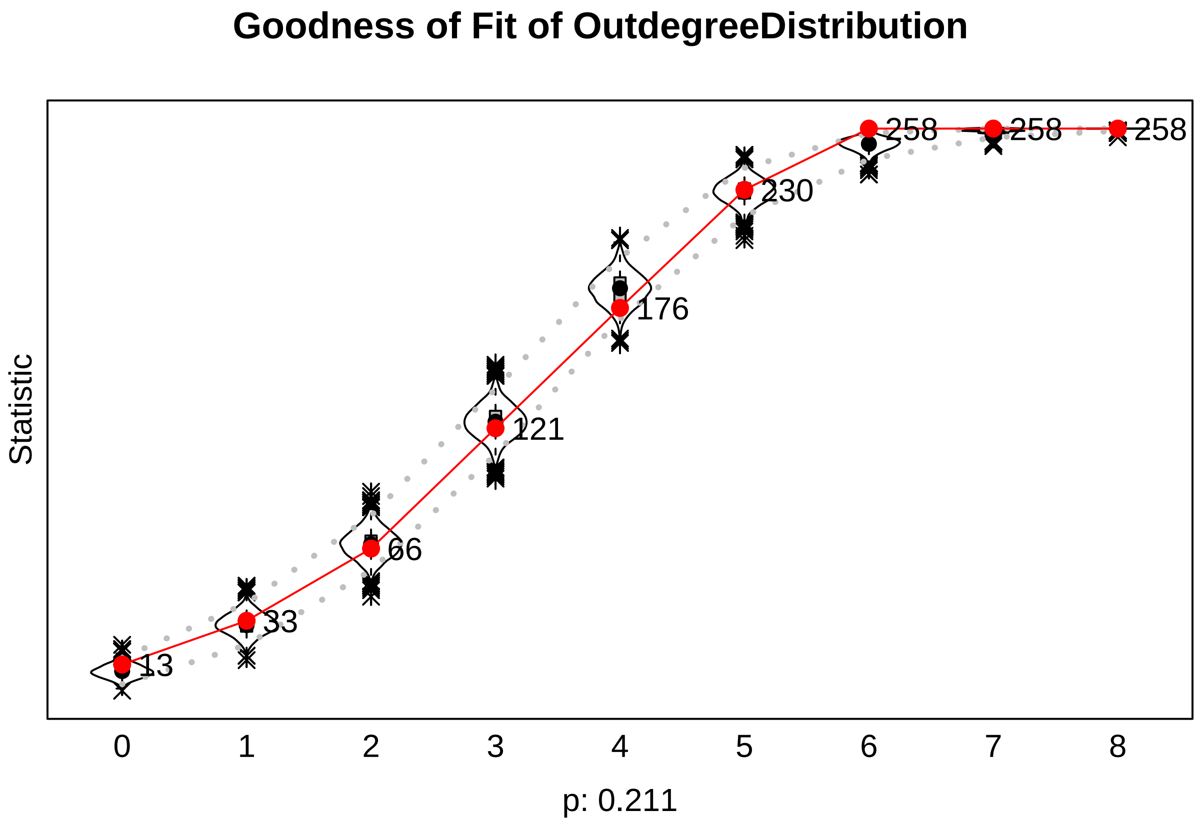
```r
# Outdegree distribution
gof1.od <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  OutdegreeDistribution
)

# Triad census
gof1.tc <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  TriadCensus
)

# Geodesic distance
gof1.gd <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  GeodesicDistribution
)
```

```r
plot(gof1.id)
```



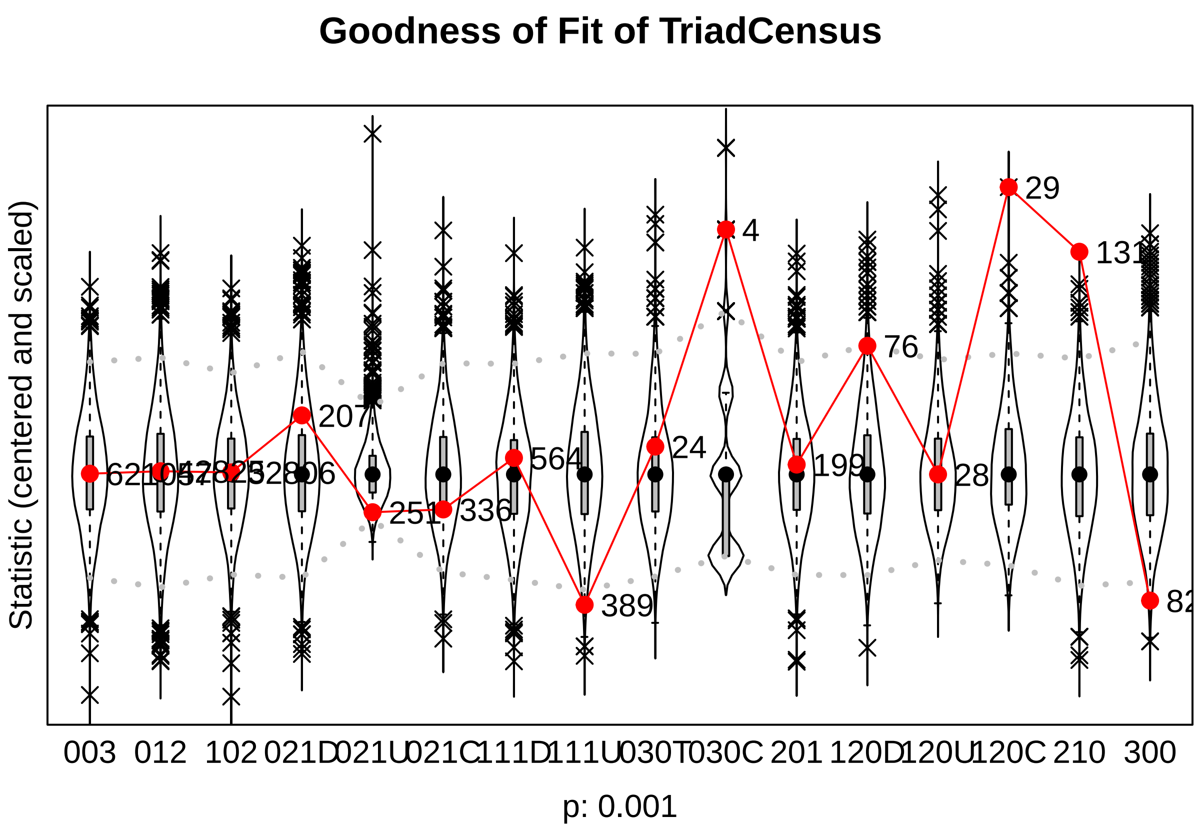
All in-degrees from 0 to 8 are in the 95% confidence interval. Furthermore, the p-value is greater than 0.05 which indicates a good fit (as mentioned in the lecture).

```r
plot(gof1.od )
```



All out-degrees from 0 to 8 are in the 95% confidence interval. Furthermore, the p-value is greater than 0.05 which indicates a good fit.

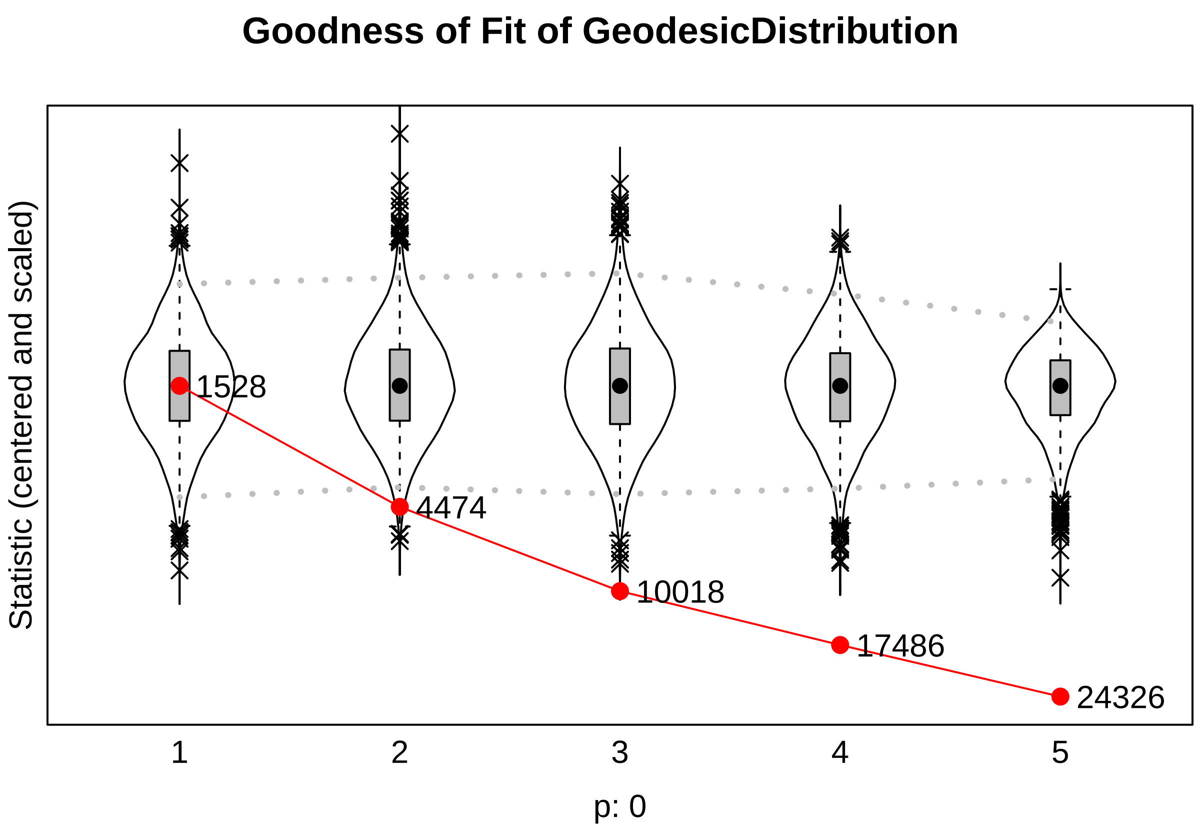
```
plot(gof1.tc, center=TRUE, scale=TRUE)
```



GOF for triad census does not fit very well. The observed triad census for the following states 030C, 120C, 210 and 300 lies outside the 95%-confidence interval. Moreover, the p-value is equal to zero which indicates a poor fit.

```
plot(gof1.gd, center=T, scale=T)
```

```
## Note: some statistics are not plotted because their variance is 0.
## This holds for the statistic: Inf.
```



The GOF for the Geodesic Distance does not fit well. Only the Geodesic Distance of one lies in the 95%-confidence interval the distances 2 to 5 are outside. Moreover, the p-value of zero indicates a poor fit.

To conclude, the in and out-degree fit well but the model does not accurately represents Triadcensus and Geodesic Distance. Thus, the significance of the assessments based on this model should be interpreted carefully.

2.3)

```
printSienaCoev(model.ev)
```

```
##                               effect  theta  s.e. p.value sig. t.conv
## 1  constant friendship rate (period 1) 12.385 1.449
## 2  constant friendship rate (period 2)  9.905 1.007
## 3                 outdegree (density) -6.041 0.598       0  *** -0.009
## 4                         reciprocity  1.743 0.120       0  *** -0.013
## 5                 transitive triplets -0.122 0.097   0.208      -0.003
## 6                            3-cycles -0.126 0.092   0.172      -0.001
## 7                     transitive ties  0.697 0.102       0  ***  0.013
## 8                             balance  0.316 0.042       0  *** -0.004
## 9          indegree - popularity (sqrt)  0.265 0.085   0.002   ** -0.021
## 10         outdegree - activity (sqrt)  0.979 0.199       0  ***  -0.01
## 11                        logDistance -0.216 0.050       0  ***  0.013
## 12                       gender alter -0.148 0.087    0.09    . -0.014
## 13                         gender ego  0.062 0.103   0.547      -0.012
## 14                        same gender  0.700 0.087       0  *** -0.019
## 15                     same drinkbeh  0.530 0.225   0.019    *  0.012
```

```
## 16            rate drinkbeh (period 1)  1.670 0.283
## 17            rate drinkbeh (period 2)  2.358 0.379
## 18               drinkbeh linear shape  0.268 0.335   0.423       0.004
## 19            drinkbeh quadratic shape  0.023 0.063   0.712      -0.002
## 20         drinkbeh average similarity  6.488 1.827       0  ***  0.004
## 21                   drinkbeh indegree  0.107 0.105   0.309       0.023
## 22                  drinkbeh outdegree -0.073 0.140   0.601       0.016
```

The estimates of the coefficients and the corresponding standard errors are obtained from a model controlling for basic endogenous and exogenous variables (all variables can be seen in the summary output above).

H4: Popular students tend to increase or maintain their level of alcohol consumption

The coefficient for "drinkbeh indegree" is not significantly different from 0 on the 5% significance level. According to our model we do not have evidence that popularity affects the evolution of drinking behavior. Hence, H4 is not supported by our model.

H5: Students tend to adjust their alcohol consumption to that of their friends.

The coefficient for "drinkbeh average similarity" is significantly different from 0 on the 5% significance level. The coefficient is positive which means that an actor adjust his drinking behavior to the ones of his friends. Hence, H5 is supported.

Differences for hypotheses 1-3

H1: The coefficient "indegree - popularity (sqrt)" is not significant any more unlike in the previous model. This means H1 is no longer supported in this model which indicates that the effects are now captured by behavior evolution.

H2: The coefficient "same alcoholConsumption" is now called "same drinkbeh" and is still significant and positive as in the previous model. Thus, H2 is still supported which indicates that the effects are not better captured by the newly included behavior evolution variables.

H3: The coefficient "logDistance" is still significant and negative as in the previous model. Thus, H3 is still supported.

2.4)

According to our model we have evidence that selection and influence processes are occurring. The underlying statistics that support H2 and H3 indicate that there are significant selection processes. Which means we have evidence that actors "select" their friends based on same alcohol consumption (H2) and the distance to them (H3).

Moreover, the underlying statistic that support H5 indicate that there are significant influence processes too. Meaning there is evidence that students tend to adjust their alcohol consumption to that of their friends.