

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 consumption 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)

```
# 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
```

```
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)

```
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")
```

```
# 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 = 8
)

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) 12.941 1.360
## 2      rate constant friendship rate (period 2) 10.186 0.990
## 3 friendship          outdegree (density) -3.237 0.785      0 ***
## 4 friendship          reciprocity  1.785 0.118      0 ***
## 5 friendship          transitive triplets  0.207 0.118  0.081  .
## 6 friendship          3-cycles -0.207 0.090  0.021  *
## 7 friendship          transitive ties  0.709 0.093      0 ***
## 8 friendship          balance  0.158 0.053  0.003  **
## 9 friendship      indegree - popularity (sqrt) -0.026 0.117  0.824
## 10 friendship      outdegree - activity (sqrt)  0.092 0.261  0.726
## 11 friendship          logDistance -0.202 0.046      0 ***
## 12 friendship          gender alter -0.139 0.082  0.088  .
## 13 friendship          gender ego  0.040 0.103  0.695
## 14 friendship          same gender  0.657 0.083      0 ***
## 15 friendship      same alcoholConsumption  0.155 0.067  0.02  *
##      t.conv
## 1
## 2
## 3  0.188
## 4  0.026
## 5  0.026
## 6  0.03
## 7  0.035
## 8 -0.039
## 9 -0.009
## 10 -0.117
## 11 0.018
## 12 -0.062
## 13 -0.055
## 14 0.024
## 15 0.034
```

Not all t.conv are below 0.1 which indicates that the model is not converged

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

```
##      [,1]
## [1,] 4.872467
```

The t.conv.max value (2.052) is also indicates that the model is not converged as it is not below 0.2.

fit it again

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 8,
  prevAns = model.ev
)
```

t-conv_k

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

```
##      dependent                effect  theta  s.e. p.value sig.
## 1      rate constant friendship rate (period 1) 12.786 1.362
## 2      rate constant friendship rate (period 2) 10.151 0.972
## 3 friendship      outdegree (density) -5.458 0.621      0 ***
## 4 friendship      reciprocity  1.784 0.111      0 ***
## 5 friendship      transitive triplets -0.059 0.099  0.553
## 6 friendship      3-cycles -0.158 0.090  0.079 .
## 7 friendship      transitive ties  0.699 0.097      0 ***
## 8 friendship      balance  0.288 0.043      0 ***
## 9 friendship      indegree - popularity (sqrt) 0.224 0.088  0.011 *
## 10 friendship      outdegree - activity (sqrt) 0.815 0.209      0 ***
## 11 friendship      logDistance -0.207 0.048      0 ***
## 12 friendship      gender alter -0.150 0.080  0.06 .
## 13 friendship      gender ego  0.061 0.101  0.541
## 14 friendship      same gender  0.695 0.083      0 ***
## 15 friendship      same alcoholConsumption 0.169 0.069  0.014 *
##      t.conv
## 1
## 2
## 3  0.054
## 4 -0.001
## 5 -0.015
## 6 -0.029
## 7  0.011
## 8 -0.055
## 9  0.001
## 10 -0.022
## 11  0.022
## 12 -0.018
## 13 -0.003
## 14  0.024
## 15  0.006
```

All t.conv are below 0.1 which indicates that the model is converged for this measure .

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

```
##      [,1]
## [1,] 0.9568674
```

The t.conv.max value (0.9568674) is still not below 0.2 because the model is not yet converged.

fit it again

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 8,
  prevAns = model.ev
)
```

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

```
##      dependent                effect  theta  s.e. p.value sig.
## 1      rate constant friendship rate (period 1) 12.694 1.366
## 2      rate constant friendship rate (period 2) 10.086 1.025
## 3 friendship                outdegree (density) -5.902 0.547      0 ***
## 4 friendship                reciprocity   1.788 0.116      0 ***
## 5 friendship                transitive triplets -0.113 0.089    0.206
## 6 friendship                3-cycles -0.139 0.090    0.123
## 7 friendship                transitive ties   0.702 0.098      0 ***
## 8 friendship                balance    0.312 0.038      0 ***
## 9 friendship      indegree - popularity (sqrt) 0.265 0.078    0.001 ***
## 10 friendship      outdegree - activity (sqrt) 0.959 0.181      0 ***
## 11 friendship                logDistance -0.209 0.048      0 ***
## 12 friendship                gender alter -0.150 0.084    0.073  .
## 13 friendship                gender ego   0.065 0.101    0.52
## 14 friendship                same gender   0.706 0.087      0 ***
## 15 friendship      same alcoholConsumption 0.175 0.067    0.009  **
##      t.conv
## 1
## 2
## 3  0.012
## 4  0.019
## 5 -0.001
## 6  0.003
## 7  0.013
## 8  0.003
## 9 -0.014
## 10 0.002
## 11 -0.016
## 12 0.004
## 13 -0.001
## 14 0.026
## 15 0.016
```

All t.conv are below 0.1 which indicates that the model is converged for this measure .

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

```
##      [,1]
## [1,] 0.1394905
```

The t.conv.max value (0.333) is still not below 0.2 because the model is not yet converged.

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 8,
  prevAns = model.ev
)
```

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

```

##      dependent                      effect  theta  s.e. p.value sig.
## 1      rate constant friendship rate (period 1) 12.696 1.341
## 2      rate constant friendship rate (period 2) 10.066 0.991
## 3 friendship          outdegree (density) -5.917 0.538      0 ***
## 4 friendship          reciprocity  1.796 0.115      0 ***
## 5 friendship          transitive triplets -0.112 0.089    0.211
## 6 friendship          3-cycles -0.142 0.092    0.125
## 7 friendship          transitive ties  0.696 0.098      0 ***
## 8 friendship          balance  0.313 0.038      0 ***
## 9 friendship      indegree - popularity (sqrt) 0.271 0.083    0.001 **
## 10 friendship      outdegree - activity (sqrt) 0.961 0.182      0 ***
## 11 friendship          logDistance -0.208 0.048      0 ***
## 12 friendship          gender alter -0.148 0.083    0.075 .
## 13 friendship          gender ego  0.063 0.104    0.543
## 14 friendship          same gender  0.705 0.087      0 ***
## 15 friendship      same alcoholConsumption 0.174 0.069    0.012 *
##      t.conv
## 1
## 2
## 3 -0.009
## 4  0.01
## 5  0.017
## 6  0.01
## 7 -0.018
## 8 -0.007
## 9 -0.006
## 10 -0.007
## 11 -0.023
## 12 -0.002
## 13  0.011
## 14 -0.002
## 15  0.023

```

All t.conv are below 0.1 which indicates that the model is converged.

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

```

##      [,1]
## [1,] 0.1473373

```

The t.conv.max value (0.1195636) is below 0.2 indicating the model is converged.

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
)

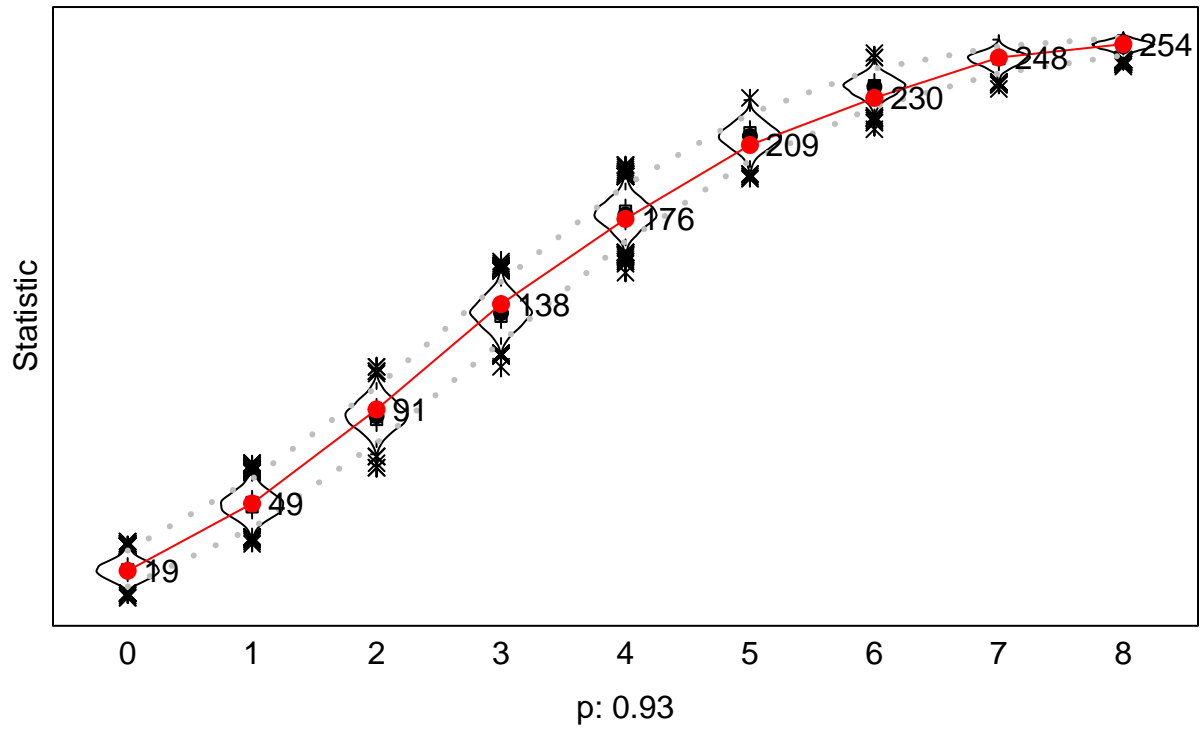
# Geodesic distance
GeodesicDistribution <- function(i, data, sims, period, groupName,
                                varName, levls = c(1:5, Inf), cumulative = TRUE) {
  x <- networkExtraction(i, data, sims, period, groupName, varName)
  require(sna)
  a <- sna::geodist(symmetrize(x))$gdist
  if (cumulative)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

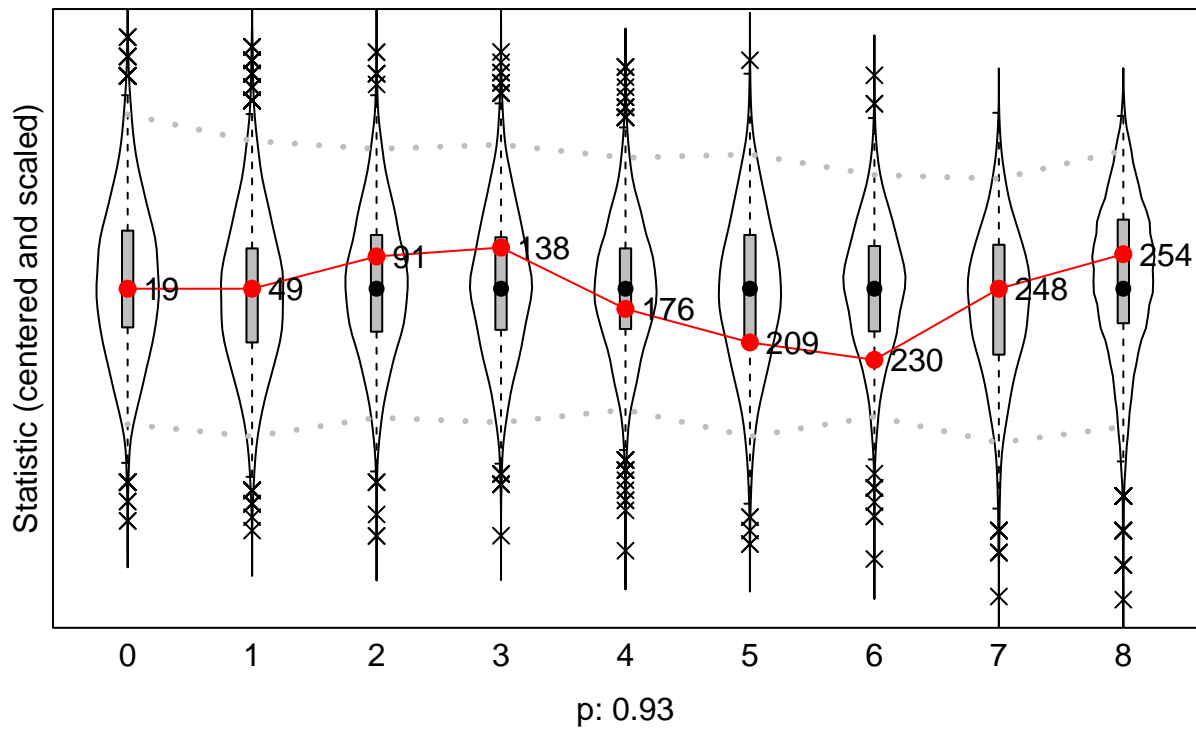
```


Goodness of Fit of IndegreeDistribution



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

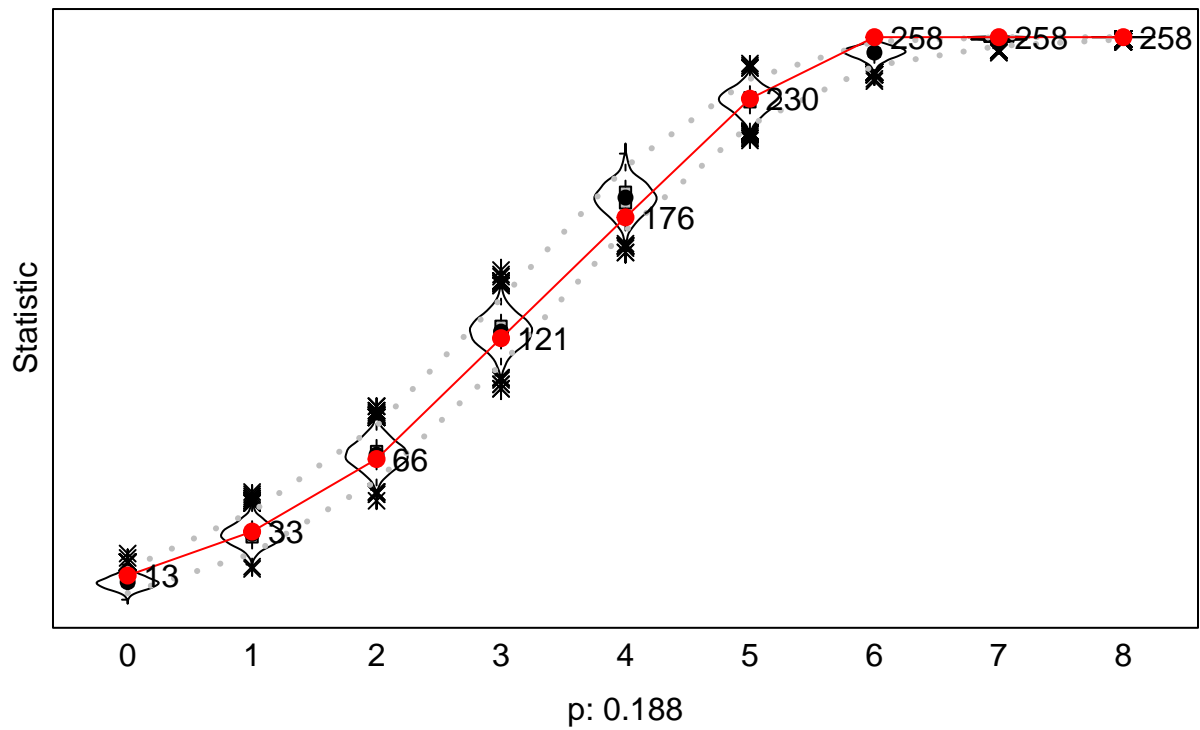
Goodness of Fit of IndegreeDistribution



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)
```

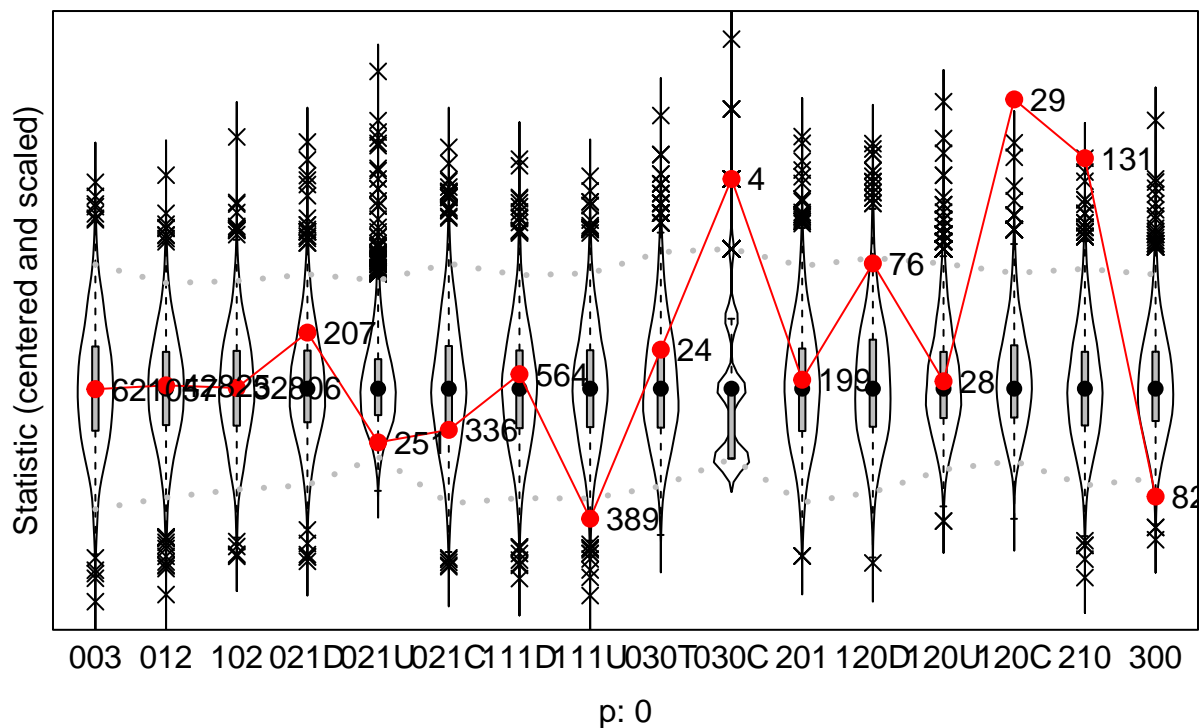
Goodness of Fit of OutdegreeDistribution



All out-degrees from 0 to 8 are in the 95% confidence interval except for 7 which is outside. Furthermore, the p-value is greater than 0.05 which indicates a good fit.

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

Goodness of Fit of TriadCensus

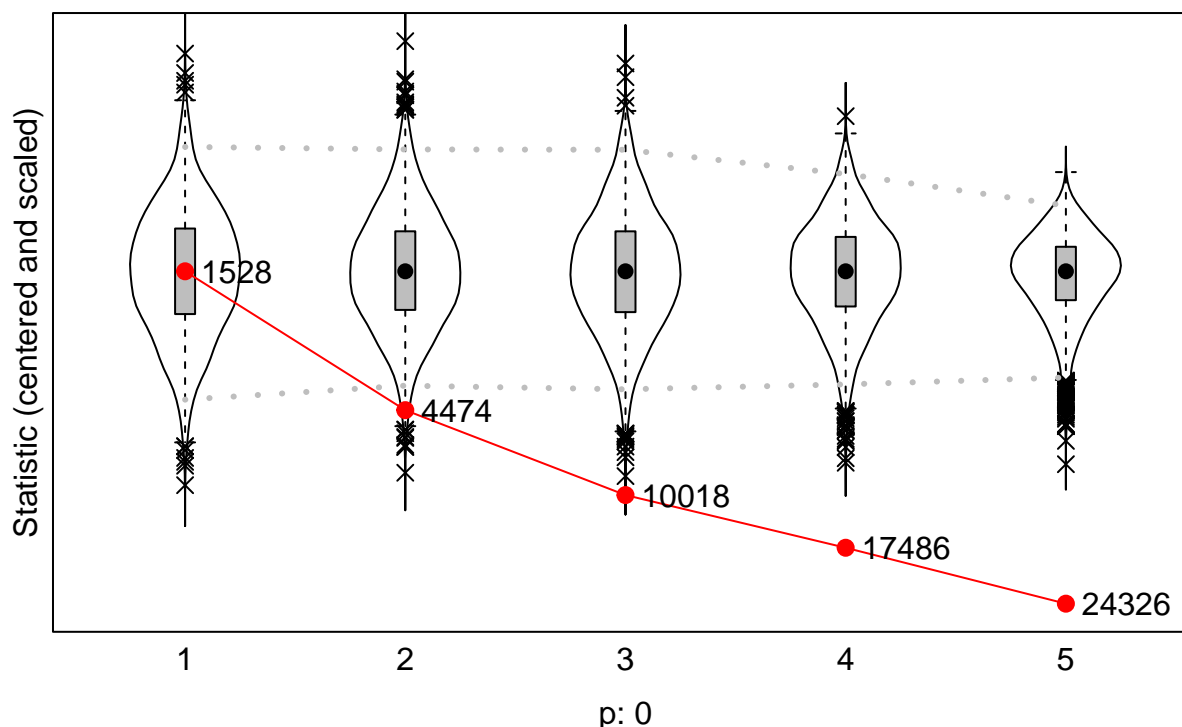


GOF for triad census does not fit very well. The observed triad census for the following states 111U, 030C, 120C, 210 and 300 lies outside the 95%-confidence interval. Moreover, the p-value is equal to zero which indicates a poor fit.

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

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

Goodness of Fit of GeodesicDistribution



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 Triadicensus 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 always the same as specified in the code. Our different trials were: 1. endogenous: transTrip, cycle3 2. endogenous: density, recip, transTrip, cycle3 3. endogenous: 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)		12.696	1.341		
## 2	rate constant friendship rate (period 2)		10.066	0.991		
## 3	friendship	outdegree (density)	-5.917	0.538	0	***
## 4	friendship	reciprocity	1.796	0.115	0	***
## 5	friendship	transitive triplets	-0.112	0.089	0.211	
## 6	friendship	3-cycles	-0.142	0.092	0.125	
## 7	friendship	transitive ties	0.696	0.098	0	***
## 8	friendship	balance	0.313	0.038	0	***
## 9	friendship	indegree - popularity (sqrt)	0.271	0.083	0.001	**
## 10	friendship	outdegree - activity (sqrt)	0.961	0.182	0	***
## 11	friendship	logDistance	-0.208	0.048	0	***

```

## 12 friendship          gender alter -0.148 0.083  0.075  .
## 13 friendship          gender ego   0.063 0.104  0.543
## 14 friendship          same gender  0.705 0.087    0 ***
## 15 friendship      same alcoholConsumption 0.174 0.069  0.012  *
##      t.conv
## 1
## 2
## 3 -0.009
## 4  0.01
## 5  0.017
## 6  0.01
## 7 -0.018
## 8 -0.007
## 9 -0.006
## 10 -0.007
## 11 -0.023
## 12 -0.002
## 13  0.011
## 14 -0.002
## 15  0.023

```

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)

```
# 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 .

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata2,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 8
)
```

model.ev

Estimates, standard errors and convergence t-ratios

```
##
##
##           Estimate   Standard   Convergence
##           Error      t-ratio
## Network Dynamics
##   1. rate constant friendship rate (period 1) 12.7183 ( 1.3740 ) 0.0180
##   2. rate constant friendship rate (period 2) 10.0708 ( 0.9762 ) -0.0205
##   3. eval outdegree (density) -4.4997 ( 0.7773 ) 0.0905
##   4. eval reciprocity 1.7392 ( 0.1270 ) 0.0136
##   5. eval transitive triplets 0.0591 ( 0.1168 ) -0.0321
##   6. eval 3-cycles -0.1733 ( 0.0991 ) -0.0287
##   7. eval transitive ties 0.6935 ( 0.1002 ) -0.0098
##   8. eval balance 0.2283 ( 0.0541 ) -0.1264
##   9. eval indegree - popularity (sqrt) 0.1115 ( 0.1053 ) -0.0236
##  10. eval outdegree - activity (sqrt) 0.4795 ( 0.2607 ) -0.0990
##  11. eval logDistance -0.2105 ( 0.0529 ) 0.0249
##  12. eval gender alter -0.1482 ( 0.0918 ) -0.0431
##  13. eval gender ego 0.0517 ( 0.1114 ) -0.0406
##  14. eval same gender 0.6704 ( 0.0831 ) 0.0008
##  15. eval same drinkbeh 0.4811 ( 0.2465 ) -0.0225
##
## Behavior Dynamics
##  16. rate rate drinkbeh (period 1) 1.6679 ( 0.2534 ) 0.0374
##  17. rate rate drinkbeh (period 2) 2.3554 ( 0.4278 ) 0.0170
##  18. eval drinkbeh linear shape 0.3388 ( 0.3971 ) -0.0384
##  19. eval drinkbeh quadratic shape 0.0251 ( 0.0615 ) 0.0437
##  20. eval drinkbeh average similarity 6.3819 ( 1.7770 ) -0.0280
##  21. eval drinkbeh indegree 0.1130 ( 0.1299 ) -0.0354
##  22. eval drinkbeh outdegree -0.1000 ( 0.1899 ) -0.0417
##
## Overall maximum convergence ratio: 3.0886
##
##
## Total of 2336 iteration steps.
```

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

```
model.ev <- siena07(
  myAlgorithm,
  data=mydata2,
  effects=myeff,
  returnDeps = TRUE,
  useCluster = TRUE,
  nbrNodes = 8,
  prevAns = model.ev
)
```

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

```
model.ev
```

##

##

Goodness of Fit

```
gof1.id <- sienaGOF(model.ev,
```

)

```

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

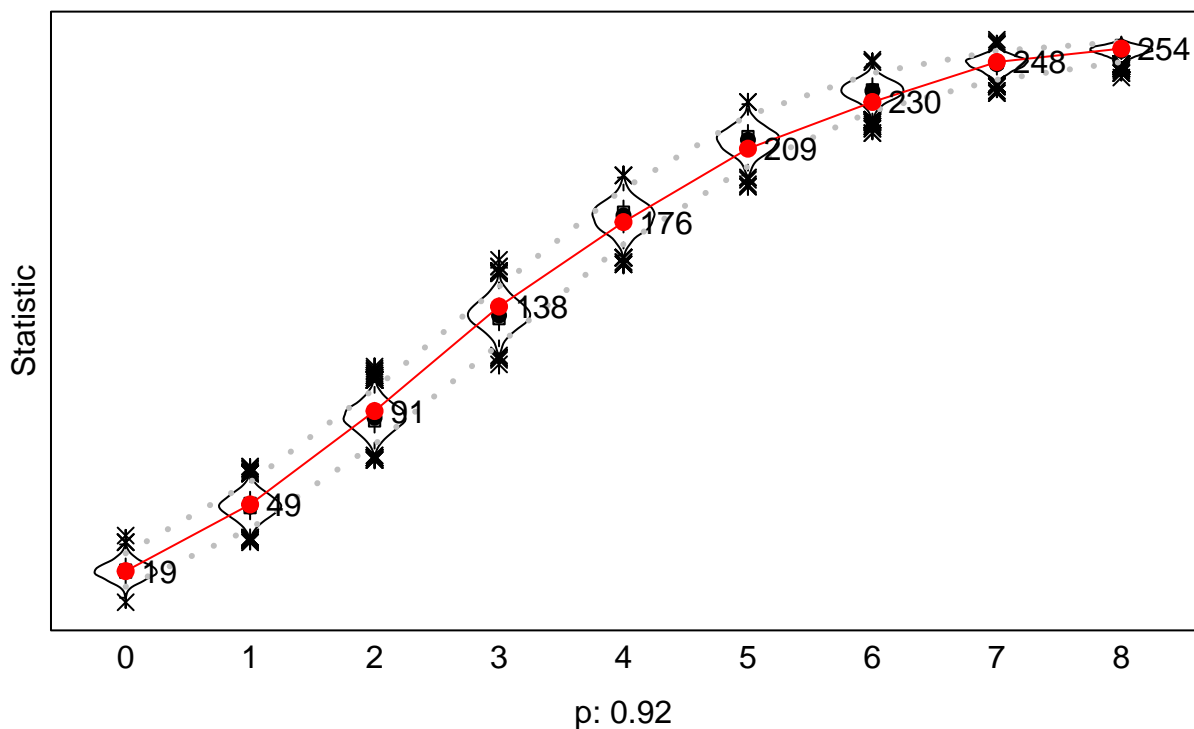
# Triad census
gof1.tc <- sienaGOF(model.ev,
  verbose = FALSE, varName = "friendship",
  TriadCensus
)

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

```

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

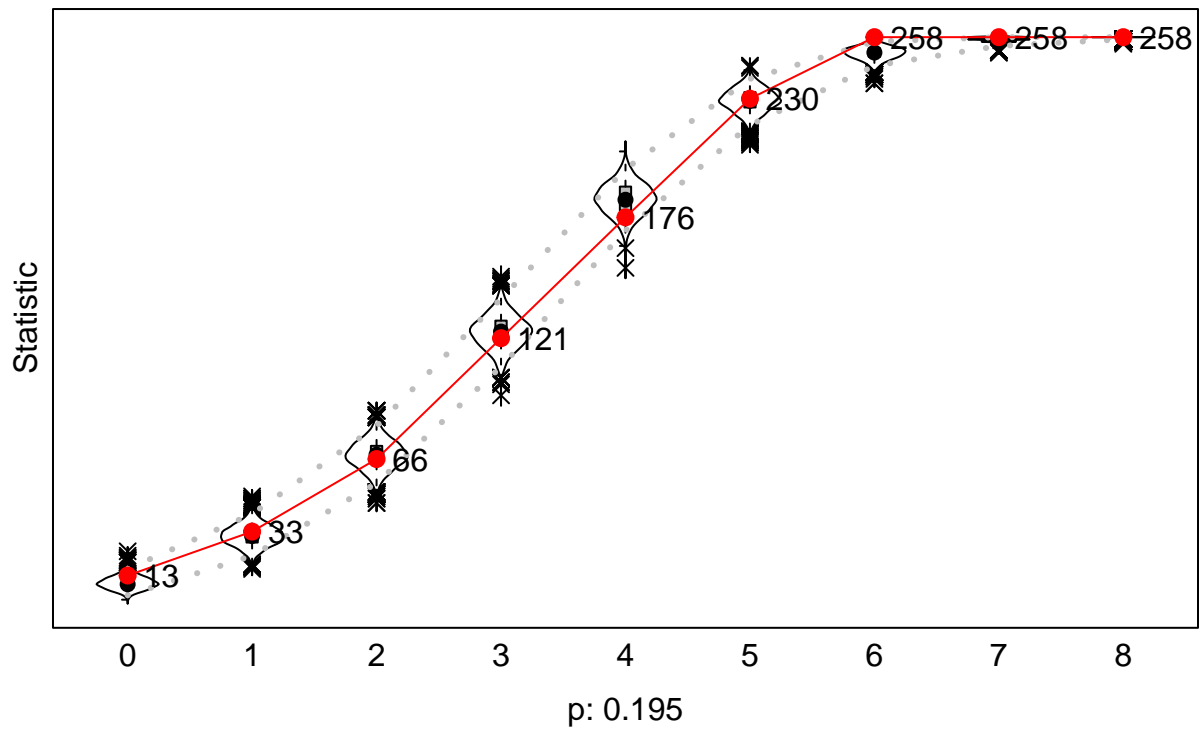
Goodness of Fit of IndegreeDistribution



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 )
```

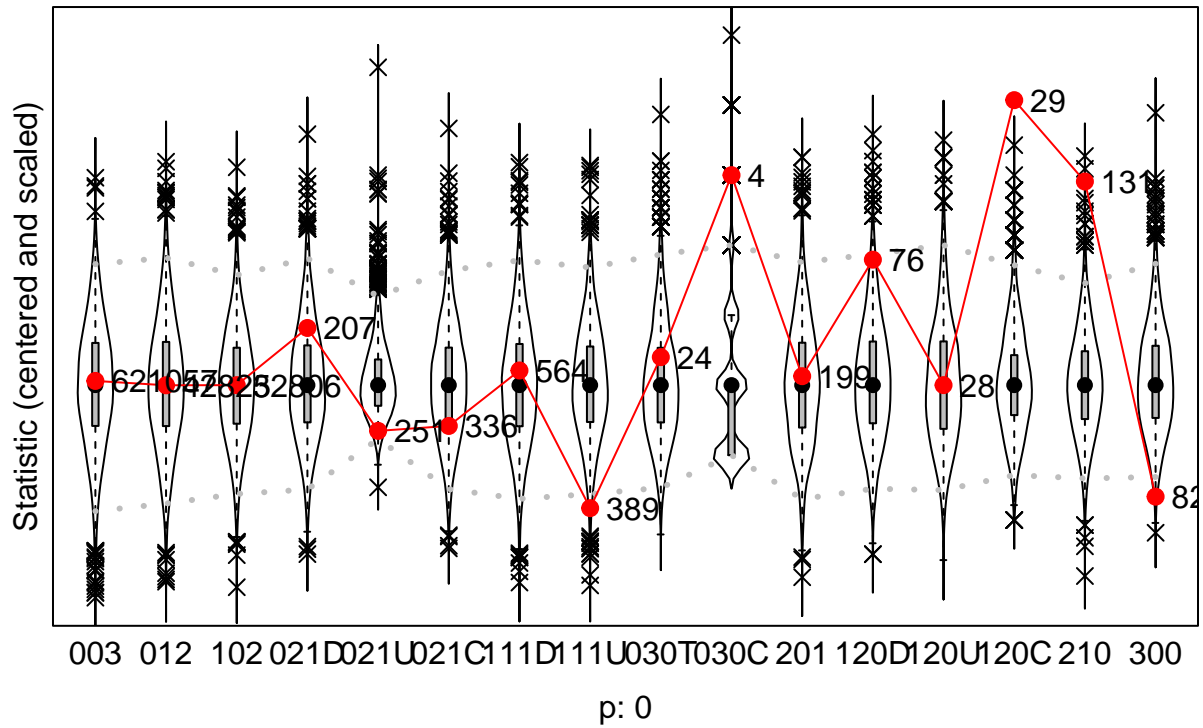
Goodness of Fit of OutdegreeDistribution



All out-degrees from 0 to 8 are in the 95% confidence interval except for 6 which is outside. Furthermore, the p-value is greater than 0.05 which indicates a good fit.

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

Goodness of Fit of TriadCensus

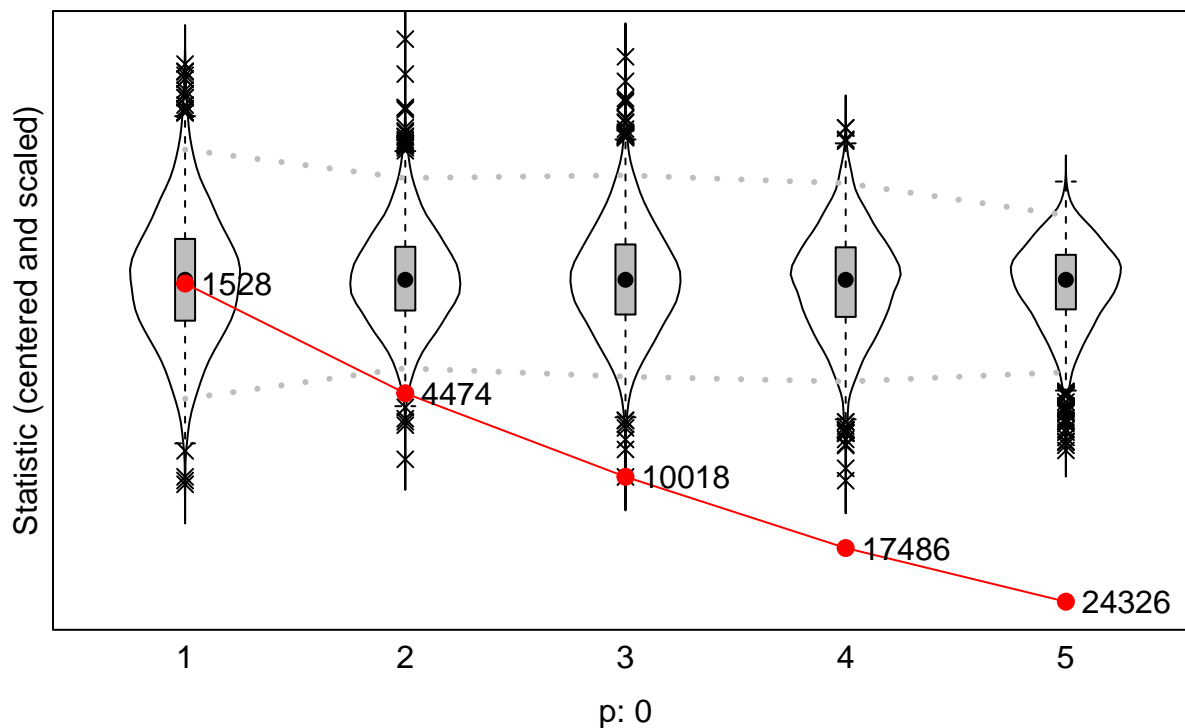


GOF for triad census does not fit very well. The observed triad census for the following states 030C, 120C, 210 and 300 lies outside the 95%-confidence interval. Moreover, the p-value is equal to zero which indicates a poor fit.

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

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

Goodness of Fit of GeodesicDistribution



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 Triadicensus 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)	12.407	1.188			
## 2	constant friendship rate (period 2)	9.924	0.906			
## 3	outdegree (density)	-6.013	0.596	0	***	0.059
## 4	reciprocity	1.745	0.119	0	***	0.014
## 5	transitive triplets	-0.120	0.093	0.198		0.026
## 6	3-cycles	-0.125	0.090	0.163		0.024
## 7	transitive ties	0.690	0.103	0	***	0.022
## 8	balance	0.315	0.040	0	***	-0.027
## 9	indegree - popularity (sqrt)	0.265	0.087	0.002	**	0.056
## 10	outdegree - activity (sqrt)	0.972	0.197	0	***	0.049
## 11	logDistance	-0.215	0.048	0	***	-0.006
## 12	gender alter	-0.149	0.083	0.073	.	0.013
## 13	gender ego	0.064	0.103	0.535		0.024
## 14	same gender	0.699	0.089	0	***	0.039
## 15	same drinkbeh	0.513	0.193	0.008	**	0.003

## 16	rate drinkbeh (period 1)	1.665	0.282			
## 17	rate drinkbeh (period 2)	2.363	0.372			
## 18	drinkbeh linear shape	0.280	0.337	0.407		-0.017
## 19	drinkbeh quadratic shape	0.025	0.065	0.705		-0.008
## 20	drinkbeh average similarity	6.524	1.778	0	***	0.018
## 21	drinkbeh indegree	0.107	0.101	0.288		-0.004
## 22	drinkbeh outdegree	-0.077	0.140	0.582		-0.007

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.