Network Analysis for Test Security

Joe Grochowalski

October 11, 2018

CollegeBoard

Overview

The topics we will cover in this presentation

Presentation Overview	Network analysis Primer	Network analysis for test security	Demonstration in R
What is network analysis, and how is it useful for test security?	Basics of network analysis Concepts in network analysis Hypothesis testing in network analysis	Combining networks and security analytics Cautions specific to test security networks	Analyzing and plotting networks in R

Why network analysis?

Motivation and purposes for using network analysis

Network Analysis studies relationships

- Test misconduct almost always involves two or more actors

Network Analysis provides rich information

- Strengthens weak circumstantial evidence of misconduct

Network Analysis combines descriptive and inferential information about collaboration

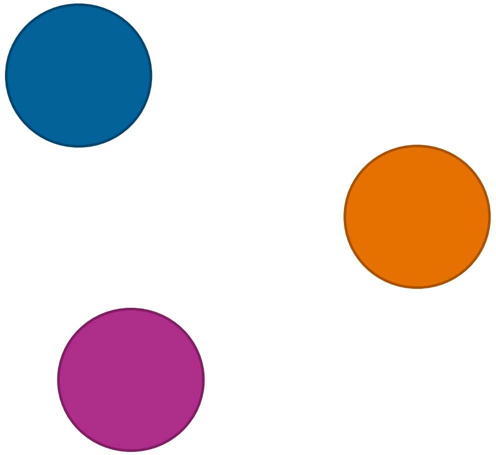
- Enhances investigations, theory building, and quality of inference

Basics of Networks

The building blocks for analyzing network relationships

Actors

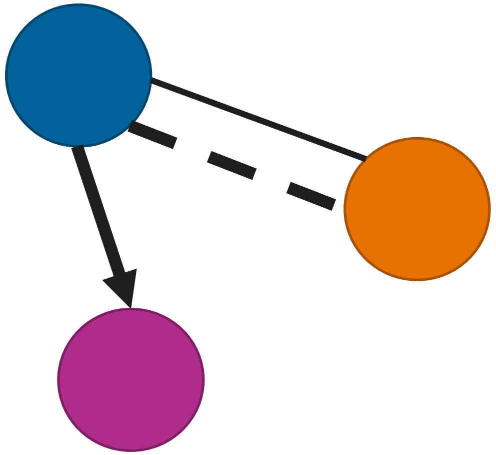
Actors are the objects that share relationships



- *Actors* are the basic unit of analysis
- It is the relationship between actors that interests us
- Examples:
  - Test taker
  - Item
  - Test location
- Typically depicted as a circle in a network graph

Ties

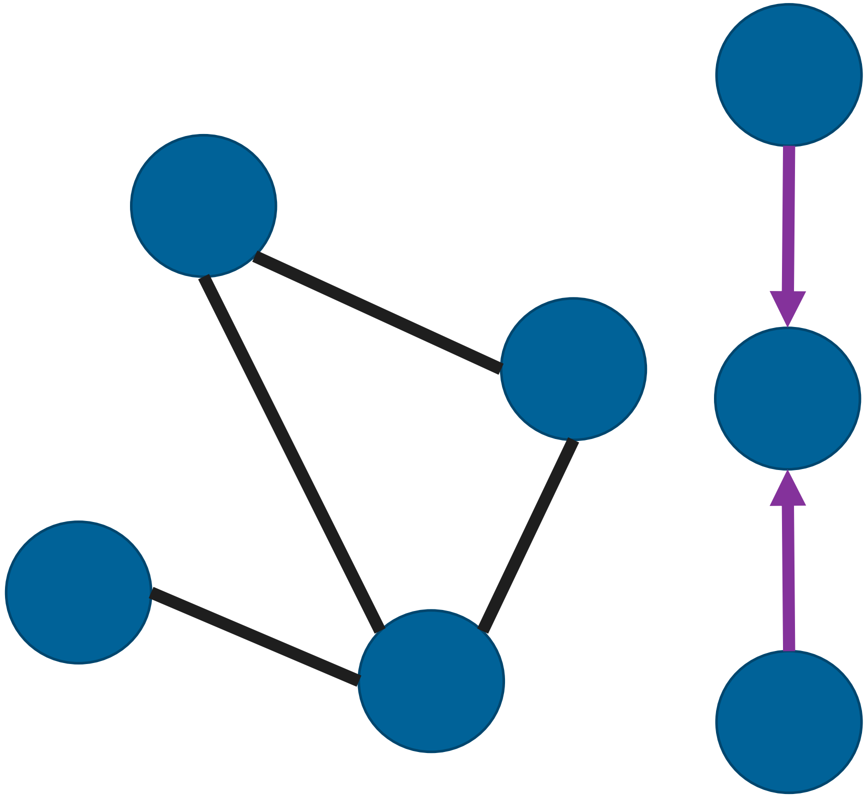
Ties represent relationships between actors



- *Ties* are the relationships between two actors
- Ties can be based on any variable type
  - Nominal
  - Ordinal
  - Continuous
- Ties can be directional
  - One actor in a relationship affects another in temporal order
- Examples of ties:
  - Sibling
  - Feedback delivery
- Ties are usually depicted as a line connecting two circles (actors)
  - Directional ties are usually arrows with the arrow head identifying the recipient of information or action

Networks

A network is a collection of actors and ties

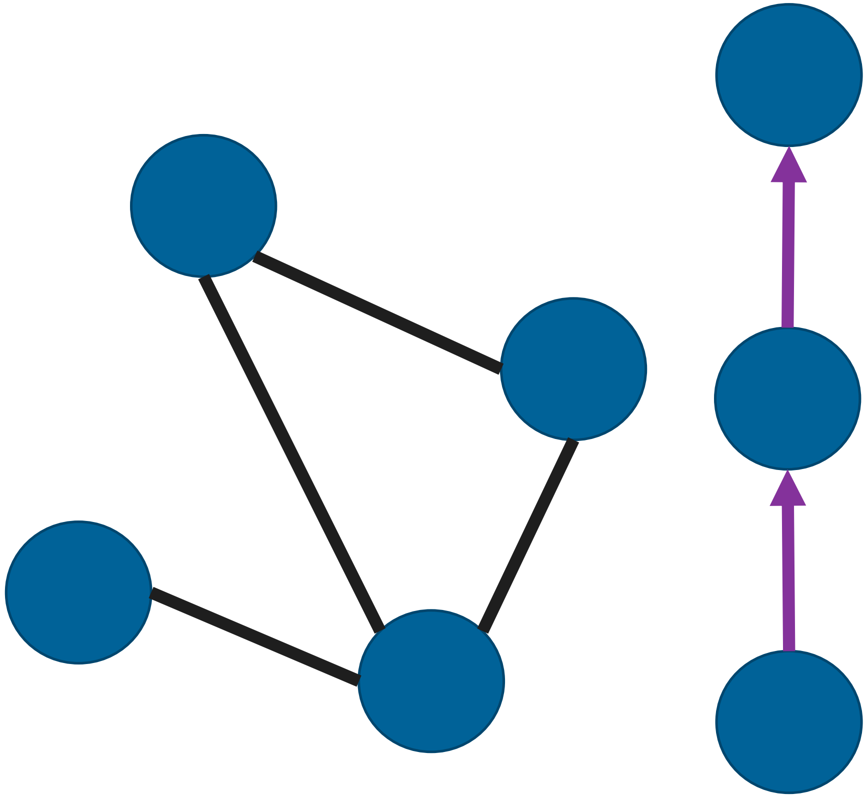


- A network is two or more actors sharing one or more ties
- Two actors are a dyad, three are a triad, etc.
- Characteristics of interest:
  - How dense is the network?
  - Who are the central actors?
  - How does the information flow?

Concepts in networks

Distance

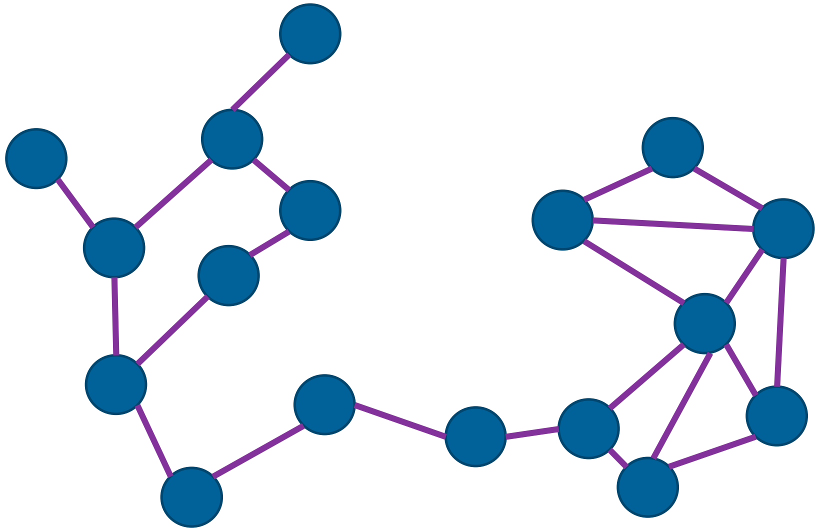
How many ties between actors?



- Distance is a measure of how many actors are connected by ties between two actors
  - Typically we are interested in “traveling” from one actor to another through ties, and we want to take the shortest route
- Longer distances indicate increasingly remote relationships
- Distance is often reported as a measure of the shortest distance (i.e., number of ties) between two persons
  - When the ties are weighted, then the distance can be a weighted distance

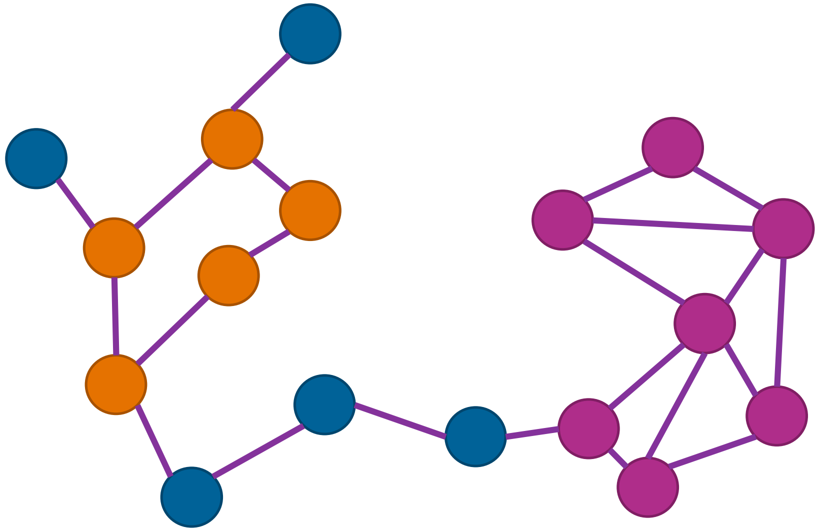
Density or Community

Identifying clusters based on shared ties



- Groups of actors with many interrelationships (i.e., mutual ties) are communities
- Communities indicate greater amounts of exchange, cooperation, sharing, etc.
- Communities can be “loosely” connected through distant ties

Density or Community



- Groups of actors with many interrelationships (i.e., mutual ties) are communities
- Communities indicate greater amounts of exchange, cooperation, sharing, etc.
- Communities can be “loosely” connected through distant ties

Hypothesis testing

Evaluating how meaningful are the observed network relationships

Philosophy of hypothesis testing

Expected network behavior versus observed

- For example, “null hypothesis testing”
- Model fit testing
  - Is the model that I have in mind consistent with the relationships in the data?

Networks can be tested for many things

- Number of actors involved
- Number of ties
- Density of the networks or groups
- Distances between actors
- Number of attributes

Example hypotheses

- “How dense do we expect this network to be by chance?”
- “What should the average distance between actors be in nature?”

Limitations of traditional tests

Typical hypothesis tests often cannot be applied to network data

Absence of independence

- Observations are related

No known parameter distributions

- Complex multivariate distributions

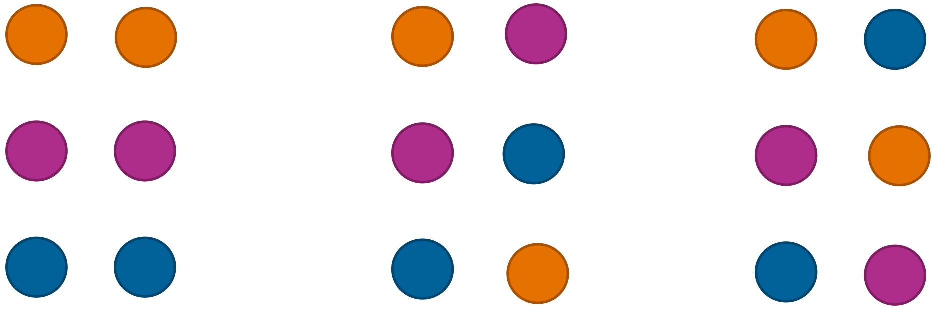
Data source?

- a random sample?
- A population?

Randomization tests

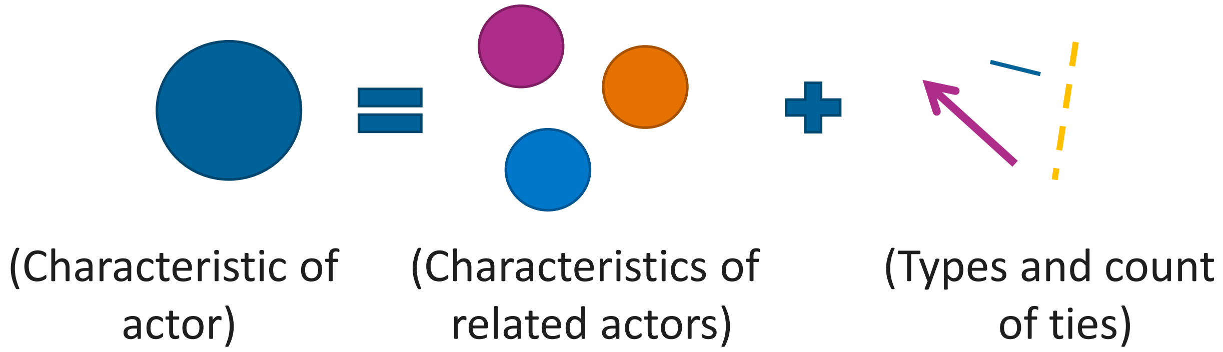
An alternative to traditional testing

- Many statistical tests look for associations in data, and compare the found associations to hypothetical data with no associations
- This is not generally possible for network analysis
- A popular alternative for inference is randomization tests
  - A randomization test uses the data at hand rather than a hypothetical data set
  - Randomization tests reassign observed relationships randomly
  - Compare the observed relationship to the randomly reassigned relationship



Actor-centered hypotheses

- Characteristics of actors are of interest in actor-centered (monadic) hypotheses
- What can we infer about actors based on their ties to others?



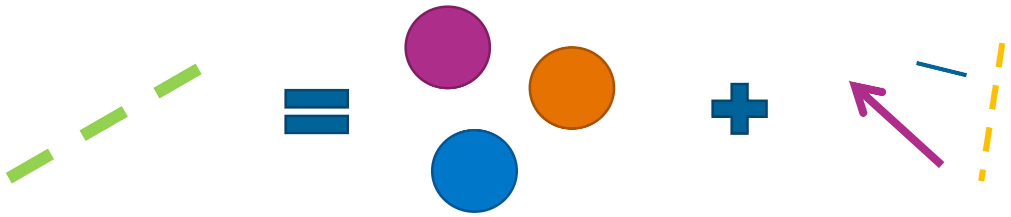
(Characteristic of actor)

(Characteristics of related actors)

(Types and count of ties)

Tie-based hypotheses

- Hypotheses about ties between actors
- Using information about the network and relationships, what do we predict for a tie between two actors?
- QAP (Quadratic assignment procedure) regression method:
  - Create two adjacency matrices, normalize them, correlate them, get an independent distribution
  - Model one tie using multiple other ties for a multiple regression



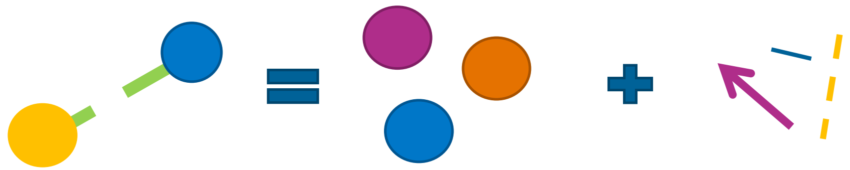
(Relationship)

(Characteristics of related actors)

(Types and count of ties)

Mixed hypotheses

Studying network relationships



(Relationship) (Characteristics of related actors) (Types and count of ties)

Mixed hypotheses combine information about actors and ties

- Mixed hypotheses attempt to explain network behavior

Analyzes actors, ties, and relationships between actors

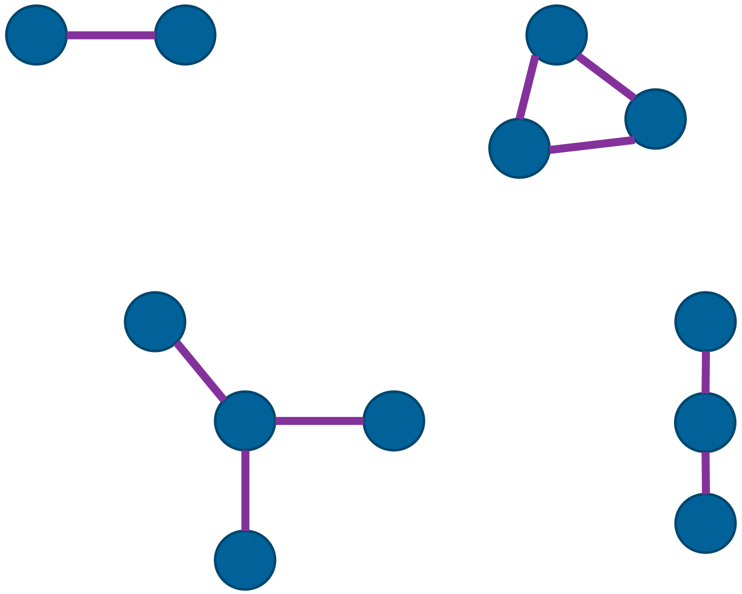
- Studies these in relationship to network characteristics
- Example: Is density of ties is related to sex (i.e., do males tend to have communities with males, etc.?)

Diffusion versus selection hypotheses

- Diffusion – Ties cause (or influence) actors
- Selection – Actors cause (or influence) ties

Exponential random graphs

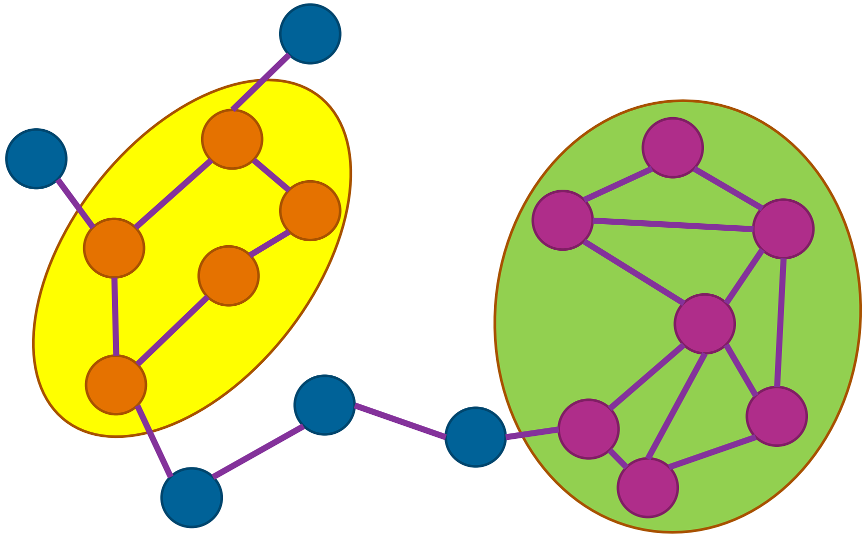
Are there more configurations than expected by chance?



- Set up a hypothetical model and then see if the characteristics of the network are the same as the hypothetical model
- General linear models with modifications for independence
  - Like logistic regression for predicting ties, based on network characteristics
- This method looks for structures in the network (e.g., triads), and asks if they are there because of an underlying process (or by chance)
- Models can become very complicated and difficult to fit/converge

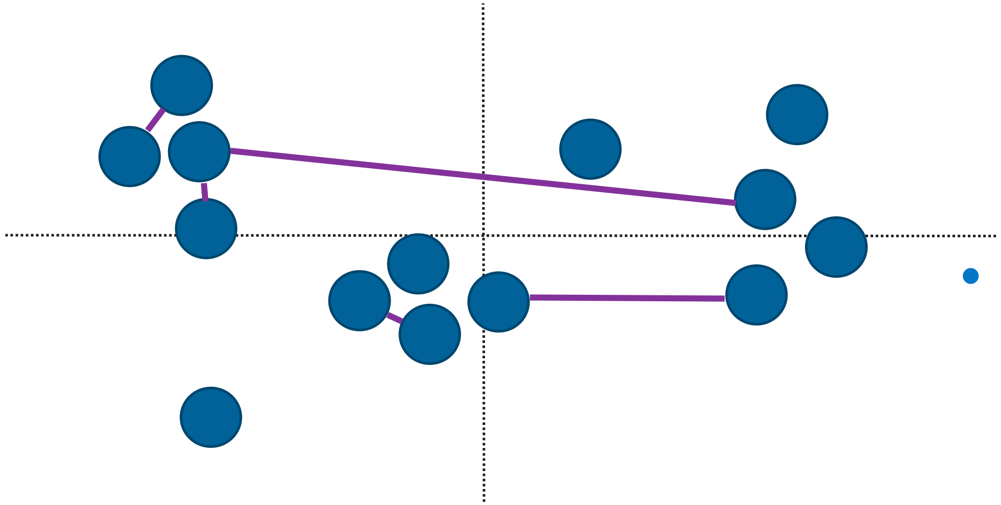
Clustering

Grouping similar test takers based on relationships



- Test takers can be sorted into groups based on similarity
- The objective is to group test takers who are most similar while forming groups that are maximally different
- Clustering could be based on…
  - Distance
  - Number of ties shared
  - Types of ties shared
- Clustering is typically conducted using hierarchical clustering methods as they are the most applicable to the data structure
  - E.g., distance matrix

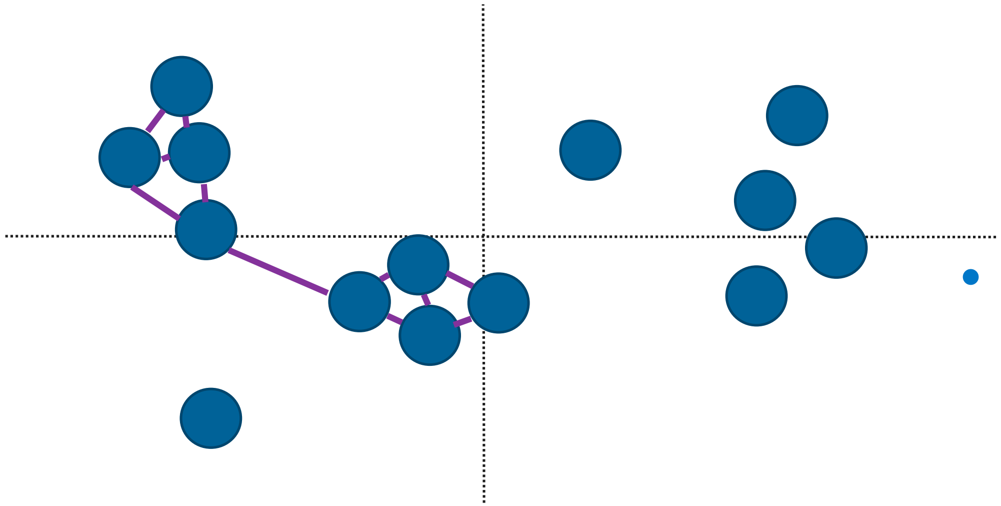
Dimension analysis



- Dimension analysis in graph theory allows us to create meaningful spatial representation of the actors
- One of the few graphs in which distance between actors is meaningful
- Scale actor distances by response similarity
  - Principal components analysis
  - Multidimensional scaling
  - Correspondence analysis
- Interpretations of distances can provide rich information when reading a graph

Dimension analysis

Dimensions give better visual representation



- Dimension analysis in graph theory allows us to create meaningful spatial representation of the actors
- One of the few graphs in which distance between actors is meaningful
- Scale actor distances by response similarity
  - Principal components analysis
  - Multidimensional scaling
  - Correspondence analysis
- Interpretations of distances can provide rich information when reading a graph

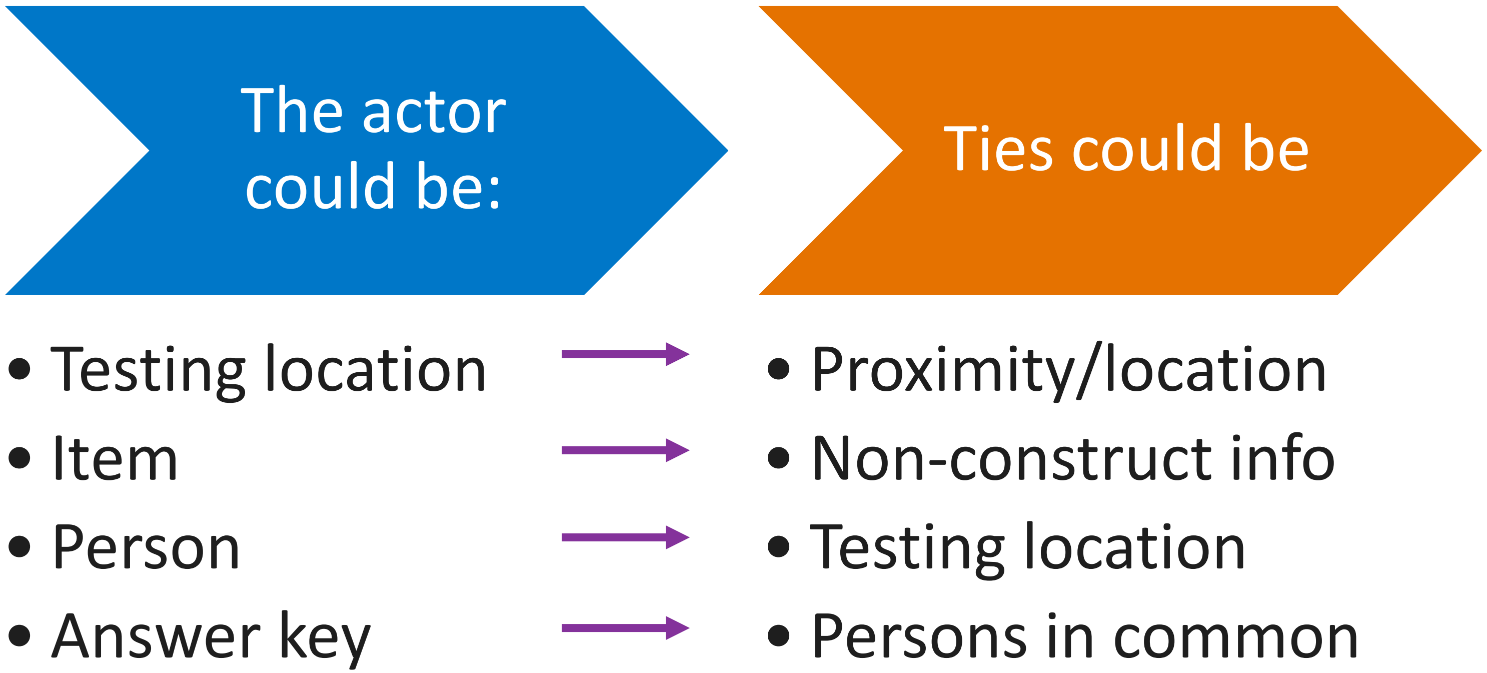
Test security network analysis

Combining inference from test security analytics with networks

Objects of measurement

What are the actors and what are the ties?

- The actor does not have to be a person/test taker
- The actor can be any object of interest that shares information or has a relationship with another actor



The actor could be:

- Testing location
- Item
- Person
- Answer key

Ties could be

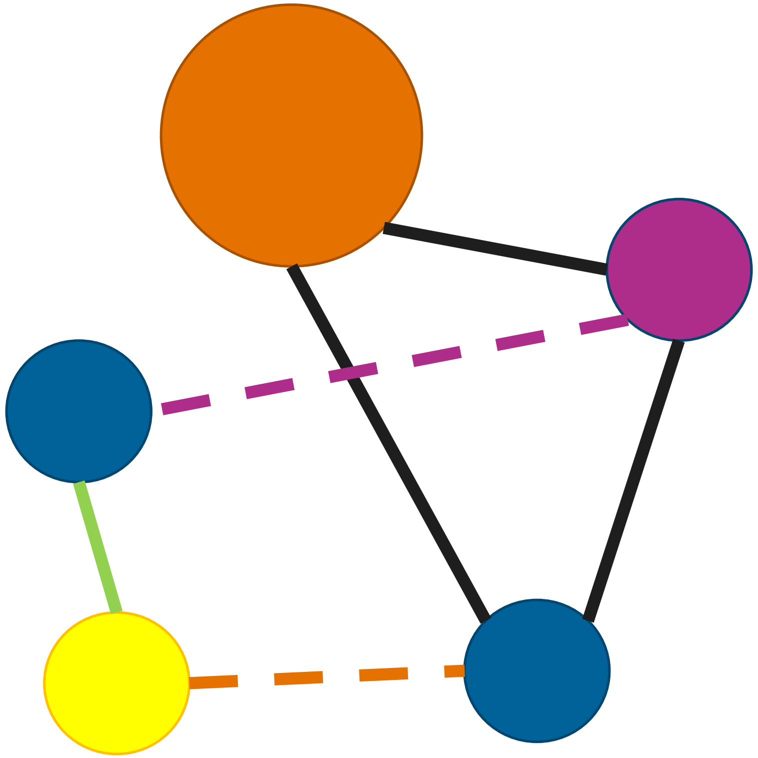
- Proximity/location
- Non-construct info
- Testing location
- Persons in common

CollegeBoard

Test fraud in theory

Theoretical relationships that could exist in a test security case

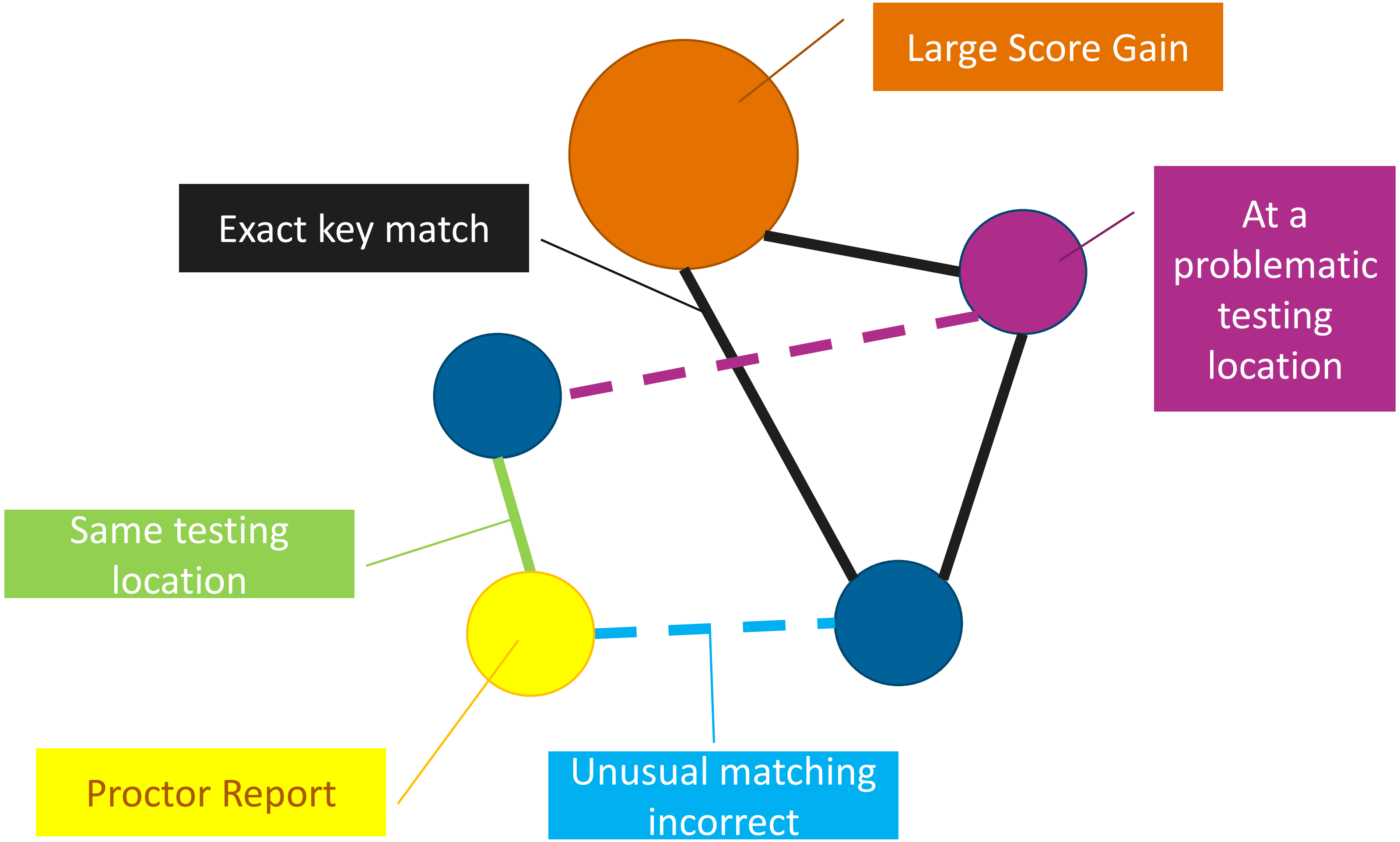
- This hypothetical relationship is what a test fraud scenario could look like
- Test fraud analytics often provide information about persons and relationships simultaneously



Test fraud in theory

Theoretical relationships that could exist in a test security case

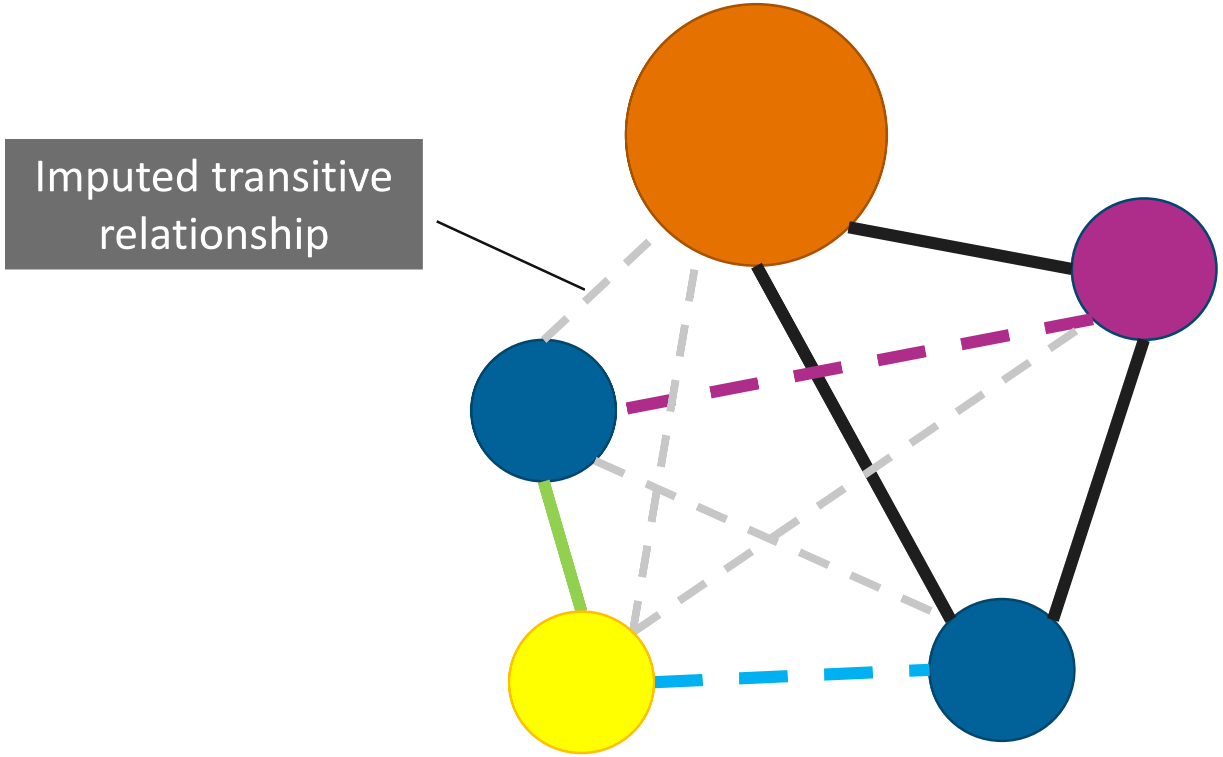
- Actor characteristics
  - Unusual gains
  - Fraud reported
  - Bad test location
  - History of fraud
  - Finished too quickly
  - Numerous erasures
  - Poor person fit
- Tie characteristics
  - Matching responses
    - Erasures
    - Incorrects (K-index)
  - Same testing location
    - Seating proximity
  - From the same school



Test fraud in theory

Theoretical relationships that could exist in a test security case

- If A and B are known to collaborate, and B and C are known to collaborate, then A and C are tied
- Imputing relationships is a (controversial?) way of strengthening statistical power

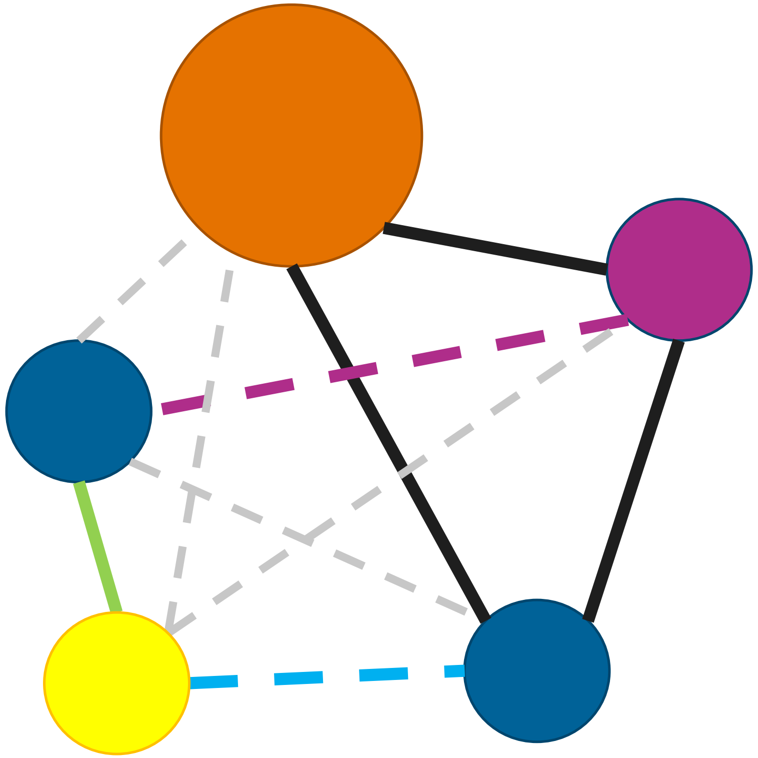


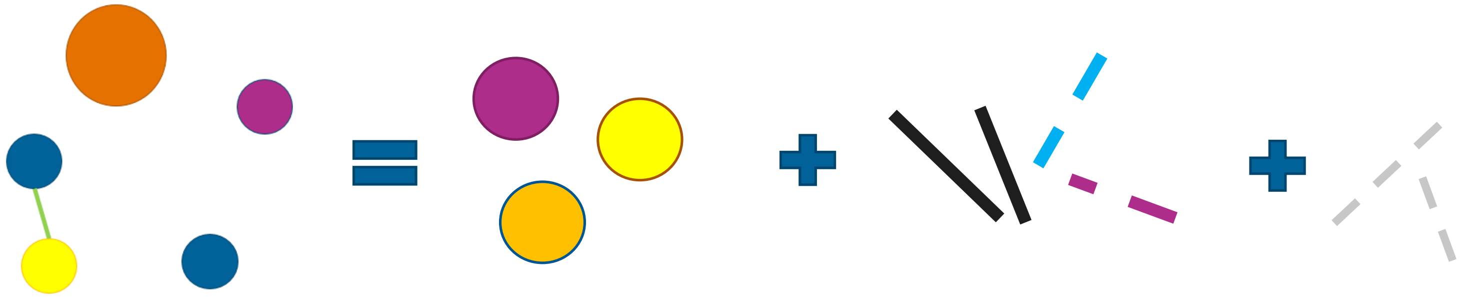
Imputed transitive relationship

Test fraud in theory

Theoretical relationships that could exist in a test security case

- Mixed method or exponential graph models can inform us about observing this formation by chance





(Characteristics of related actors)

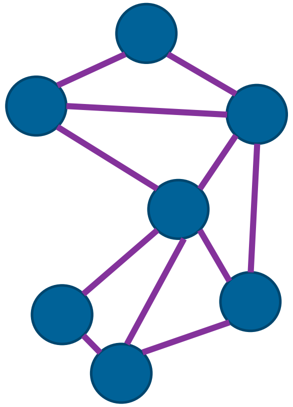
(Types and count of ties)

(Transitive relationships)

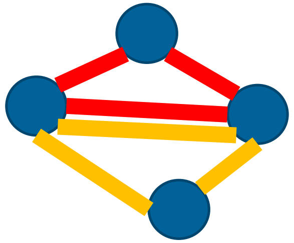
Cliques

Cliques are the most direct way of detecting coordinated fraud

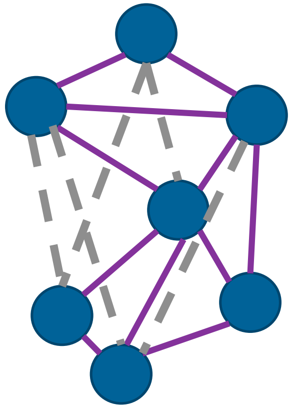
- Cliques are groups of actors that are all interconnected
- When using test security analytics to form ties, cliques become very important indicators of test misconduct
- Method:
  - Count the number of cliques and the density of each
  - Use a randomization method to determine how unusual observed clique size and density are



Collection of cliques

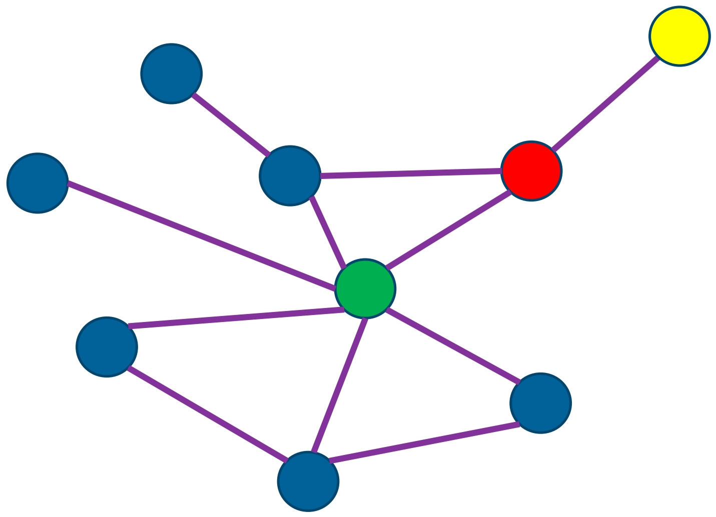


Two cliques can be combined transitively



Content acquisition

- How difficult is it for an actor in the network to acquire information about an item or content?
- How important is a specific actor in distributing content?



Example:

- Suppose actor yellow has foreknowledge of items 1-20
- Remaining actors have items 21-40
- Green is a central figure in the network
- Red is a gatekeeper

Useful information could include:

- What is the average distance one has to travel to get items 1-20?
- How many different paths lead to 1-20?

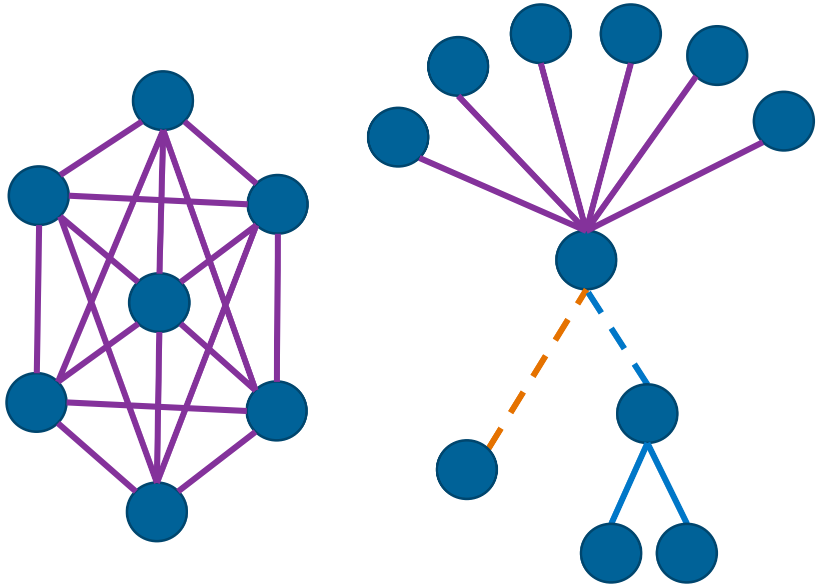
Security planning

Use known past and current information about relationships to plan

- What are the characteristics of actors and ties that result in the emergence of test fraud?
- Attributes that can contribute to planning
  - Test location size
  - Number of ties in a testing location
    - test takers from the same class or school
  - History of testing misconduct
- Example:
  - Enhanced security efforts might be deployed for testing locations that have 2+ known cliques, plus one test taker with a history of suspected testing misconduct

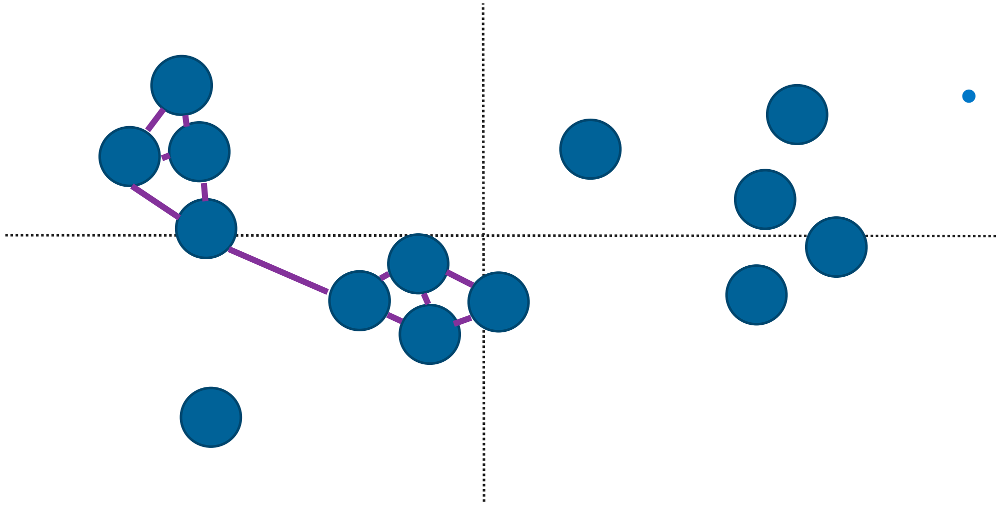
Key-centric analysis

Having keys at the center of the analysis gives different descriptive information

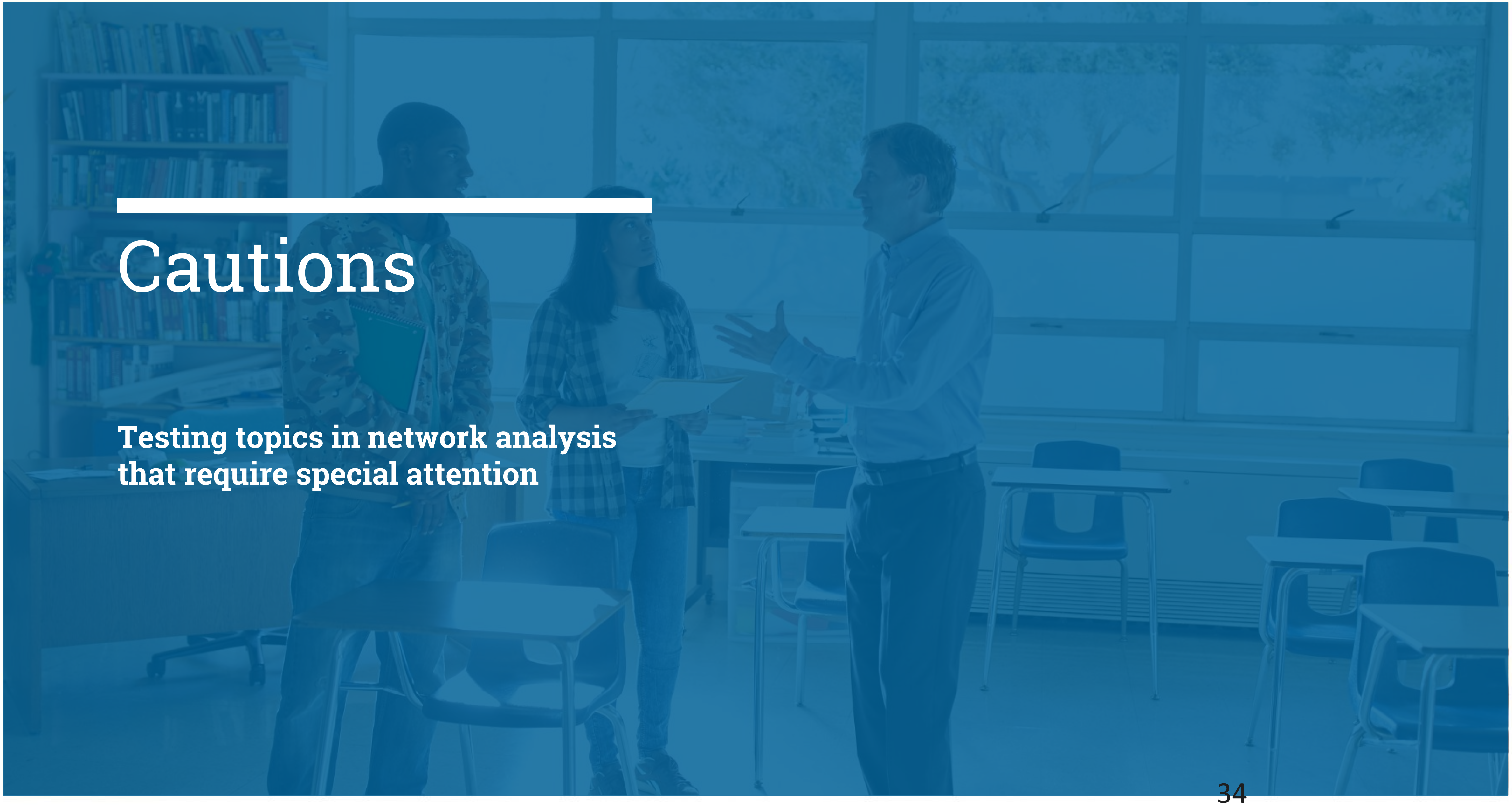


- Test takers do not have to be the actors in the network
- Test keys can be actors, too
  - Test keys are the given answers to the items on a test
- Use match statistics to relate keys that do not match exactly
- Fortify the key analysis with ties
  - Test location
  - Test security history
  - Test speed
  - Unusual score (gain)

Assessing fit and groups via dimensional analysis



- When correspondence analysis is applied to item-level data, the resulting plot optimally organizes actors according to similarity in their responses
- Actors on the right side of the plot have higher overall scores
- Actors on the left side of the plot have lower scores
- Actors high or low on the second (vertical) dimension have unusual responses, indicating poor fit
- Cliques that have extreme locations on the vertical axis are of special interest, and could be the focus of hypothesis testing

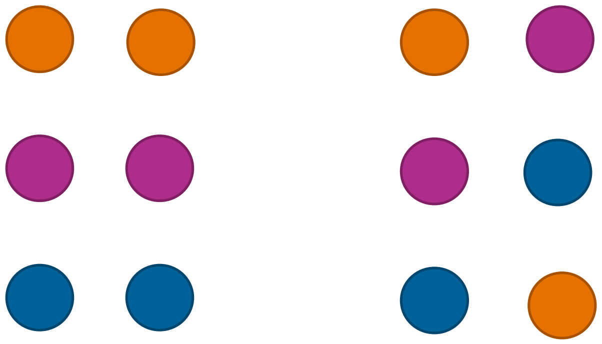


Cautions

Testing topics in network analysis that require special attention

Randomization pitfalls

Test analytics often complicate network analysis



Ties are not necessarily independent of person characteristics

- Whether two actors can be tied might depend on the actors
- For example, “over-the-shoulder” copying can only be a tie between two actors from the same testing location

Randomization methods should match actors who have the same characteristics as the observations

- When matching an actor with a new “independent” actor, the new actor should have similar characteristics to the original actor
- Otherwise some comparisons might not make sense

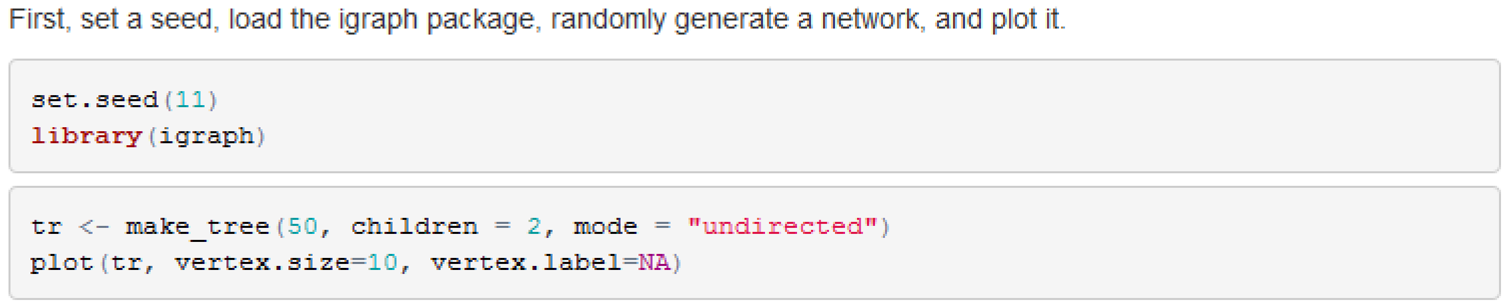
Test fraud statistics often depend on scores

- Incorrect item responses are often critical for detecting fraud
- More incorrect items = greater chance of being detected (power)
  - And greater false positive rate
- Randomization and permutation tests have to take these into account

R Code

Generate graphs

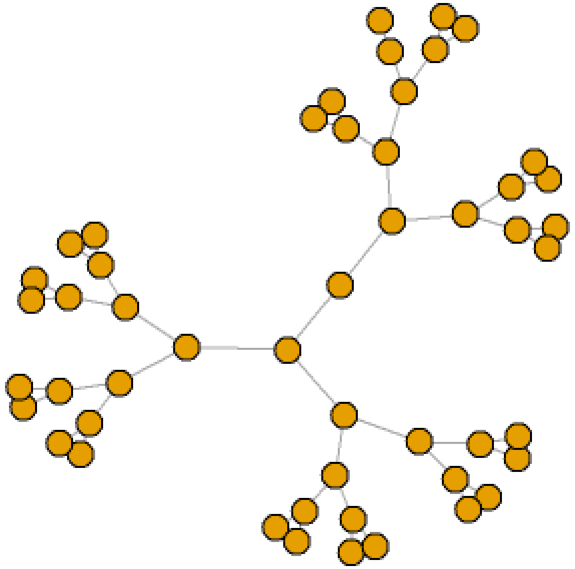
- Set a seed so we get the same (random) results
- Create the undirected network with 50 actors
- Plot the network



First, set a seed, load the igraph package, randomly generate a network, and plot it.

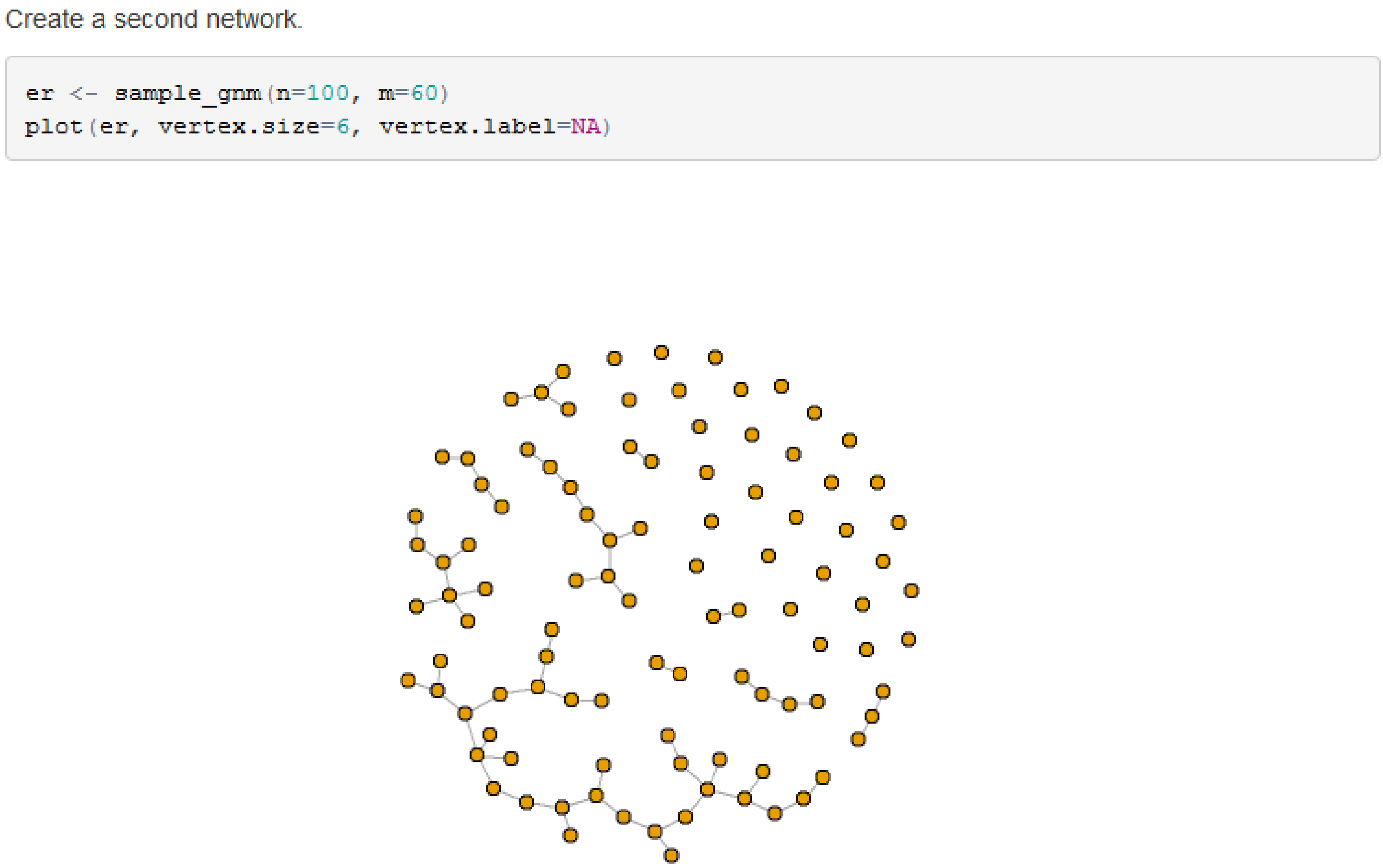
```
set.seed(11)
library(igraph)
```

```
tr <- make_tree(50, children = 2, mode = "undirected")
plot(tr, vertex.size=10, vertex.label=NA)
```



Generate network 2

- Generate a random graph with some varying degrees of association
- Plot it

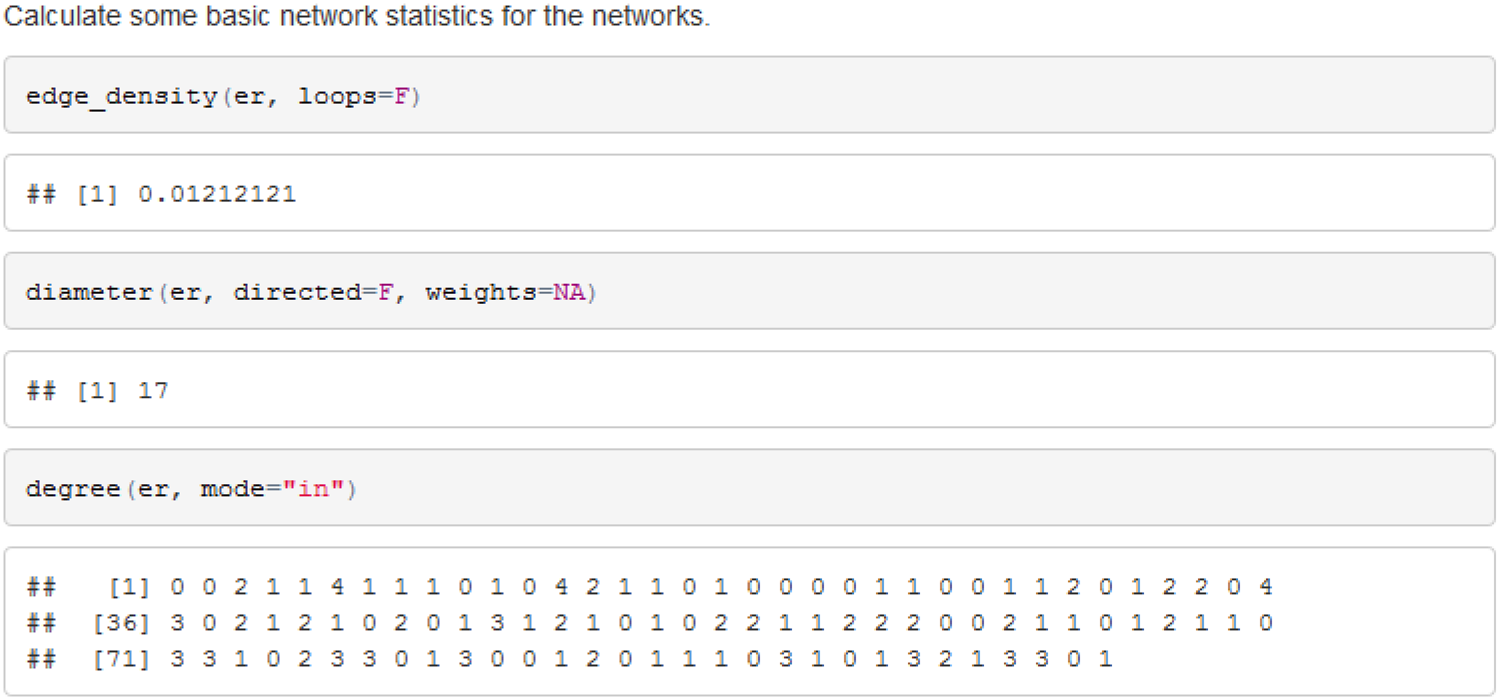


Create a second network.

```
er <- sample_gnm(n=100, m=60)
plot(er, vertex.size=6, vertex.label=NA)
```

Basic Descriptives

- Calculate the density
- Calculate the diameter (the widest distance between two actors)
- Calculate the degree for each actor



Calculate some basic network statistics for the networks.

```
edge_density(er, loops=F)
```

```
## [1] 0.01212121
```

```
diameter(er, directed=F, weights=NA)
```

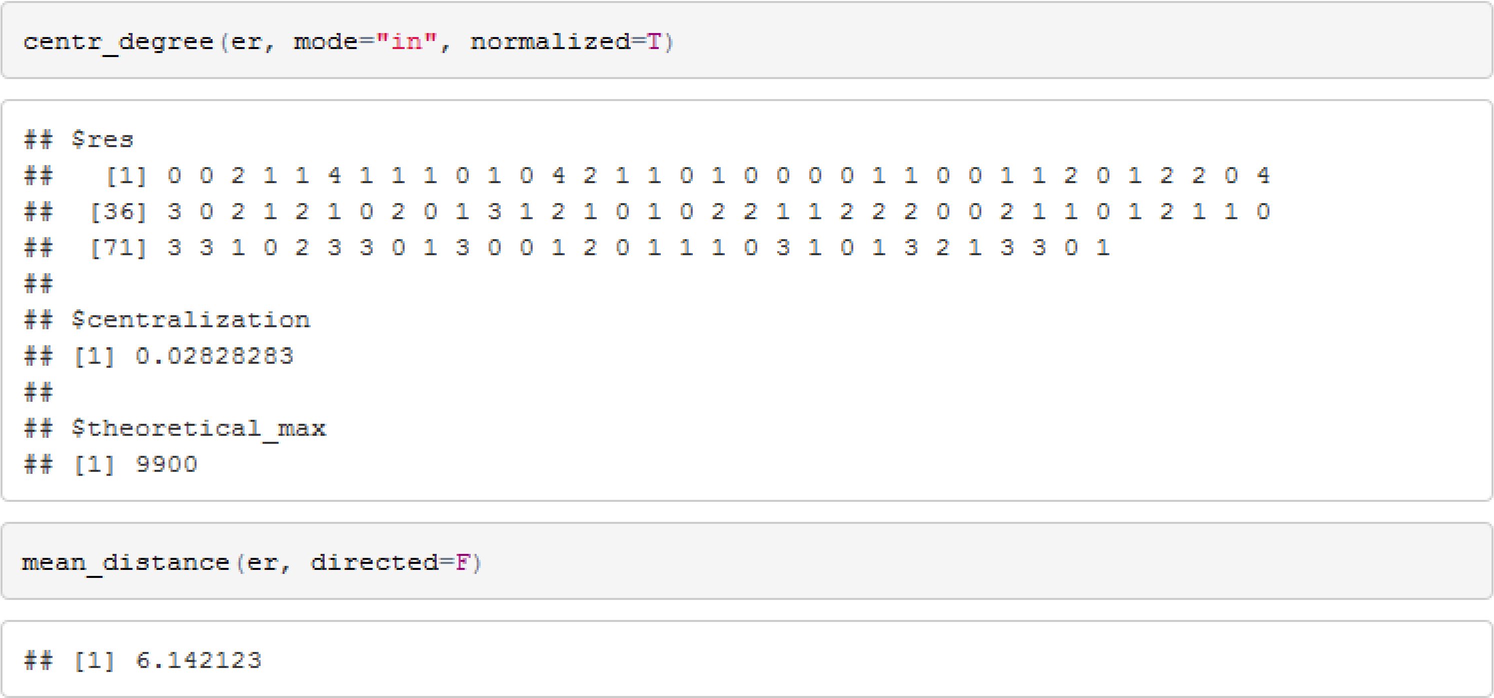
```
## [1] 17
```

```
degree(er, mode="in")
```

```
##   [1] 0 0 2 1 1 4 1 1 1 0 1 0 4 2 1 1 0 1 0 0 0 0 1 1 0 0 1 1 2 0 1 2 2 0 4
##  [36] 3 0 2 1 2 1 0 2 0 1 3 1 2 1 0 1 0 2 2 1 1 2 2 2 0 0 2 1 1 0 1 2 1 1 0
##  [71] 3 3 1 0 2 3 3 0 1 3 0 0 1 2 0 1 1 1 0 3 1 0 1 3 2 1 3 3 0 1
```

Centrality and distances

- Calculate the centrality
- Calculate the mean distance between actors (edges)



```
centr_degree(er, mode="in", normalized=T)
```

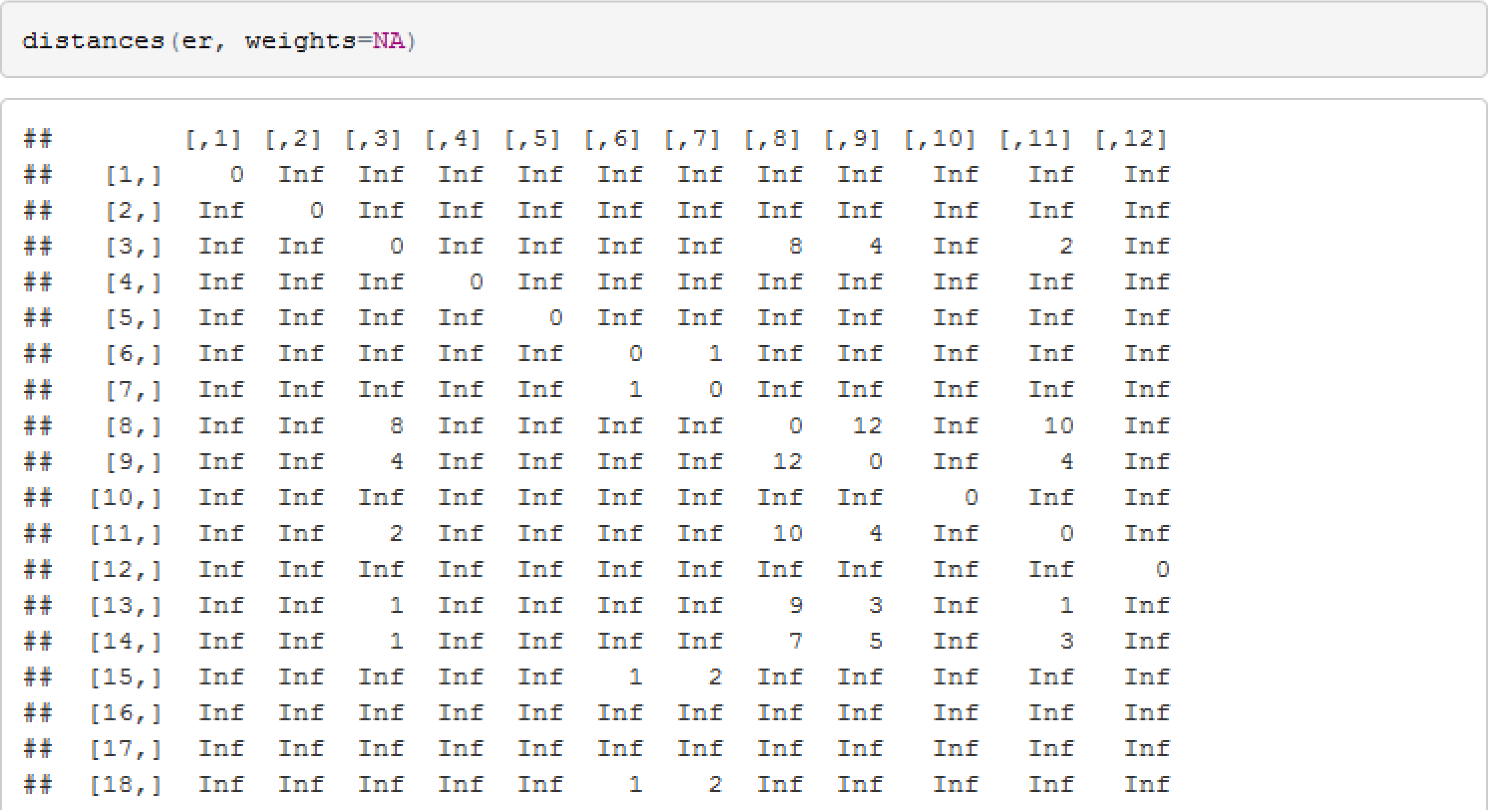
```
## $res
##   [1] 0 0 2 1 1 4 1 1 1 0 1 0 4 2 1 1 0 1 0 0 0 0 1 1 0 0 1 1 2 0 1 2 2 0 4
##  [36] 3 0 2 1 2 1 0 2 0 1 3 1 2 1 0 1 0 2 2 1 1 2 2 2 0 0 2 1 1 0 1 2 1 1 0
##  [71] 3 3 1 0 2 3 3 0 1 3 0 0 1 2 0 1 1 1 0 3 1 0 1 3 2 1 3 3 0 1
##
## $centralization
## [1] 0.02828283
##
## $theoretical_max
## [1] 9900
```

```
mean_distance(er, directed=F)
```

```
## [1] 6.142123
```

Distance matrix

- Look at a the matrix of distances between all actors

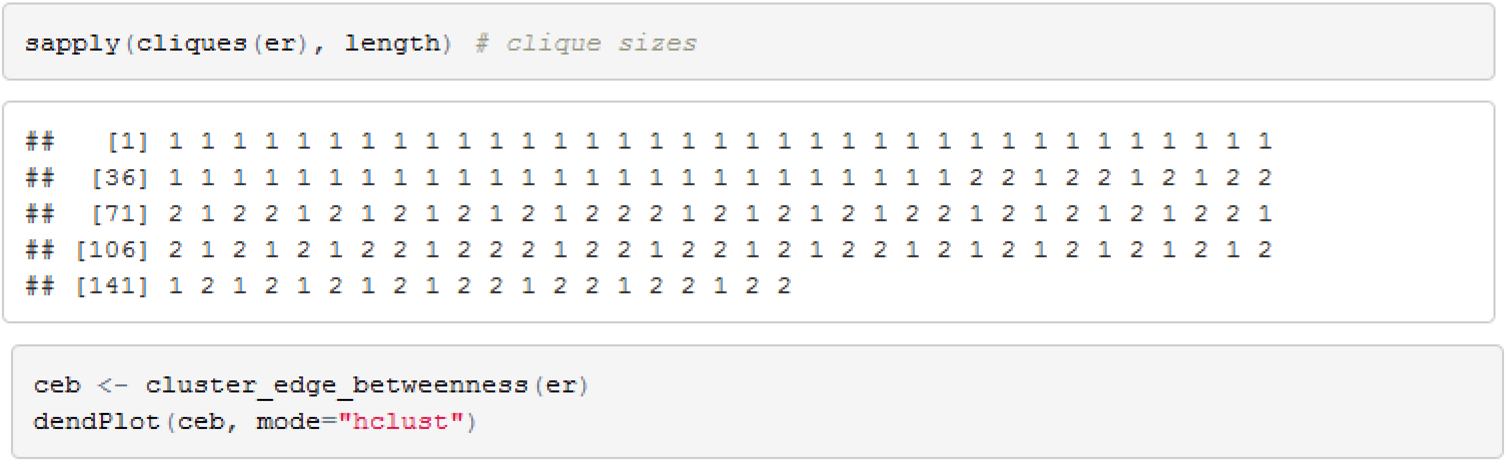


```
distances(er, weights=NA)
```

```
##        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]     0  Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf   Inf   Inf
##  [2,]   Inf    0  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf   Inf   Inf
##  [3,]   Inf  Inf    0  Inf  Inf  Inf  Inf    8    4   Inf     2   Inf
##  [4,]   Inf  Inf  Inf    0  Inf  Inf  Inf  Inf  Inf   Inf   Inf   Inf
##  [5,]   Inf  Inf  Inf  Inf    0  Inf  Inf  Inf  Inf   Inf   Inf   Inf
##  [6,]   Inf  Inf  Inf  Inf  Inf    0    1  Inf  Inf   Inf   Inf   Inf
##  [7,]   Inf  Inf  Inf  Inf  Inf    1    0  Inf  Inf   Inf   Inf   Inf
##  [8,]   Inf  Inf    8  Inf  Inf  Inf  Inf    0   12   Inf    10   Inf
##  [9,]   Inf  Inf    4  Inf  Inf  Inf  Inf   12    0   Inf     4   Inf
## [10,]   Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf     0   Inf   Inf
## [11,]   Inf  Inf    2  Inf  Inf  Inf  Inf   10    4   Inf     0   Inf
## [12,]   Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf   Inf     0
## [13,]   Inf  Inf    1  Inf  Inf  Inf  Inf    9    3   Inf     1   Inf
## [14,]   Inf  Inf    1  Inf  Inf  Inf  Inf    7    5   Inf     3   Inf
## [15,]   Inf  Inf  Inf  Inf  Inf    1    2  Inf  Inf   Inf   Inf   Inf
## [16,]   Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf   Inf   Inf
## [17,]   Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf   Inf   Inf
## [18,]   Inf  Inf  Inf  Inf  Inf    1    2  Inf  Inf   Inf   Inf   Inf
```

Cliques and community clusters

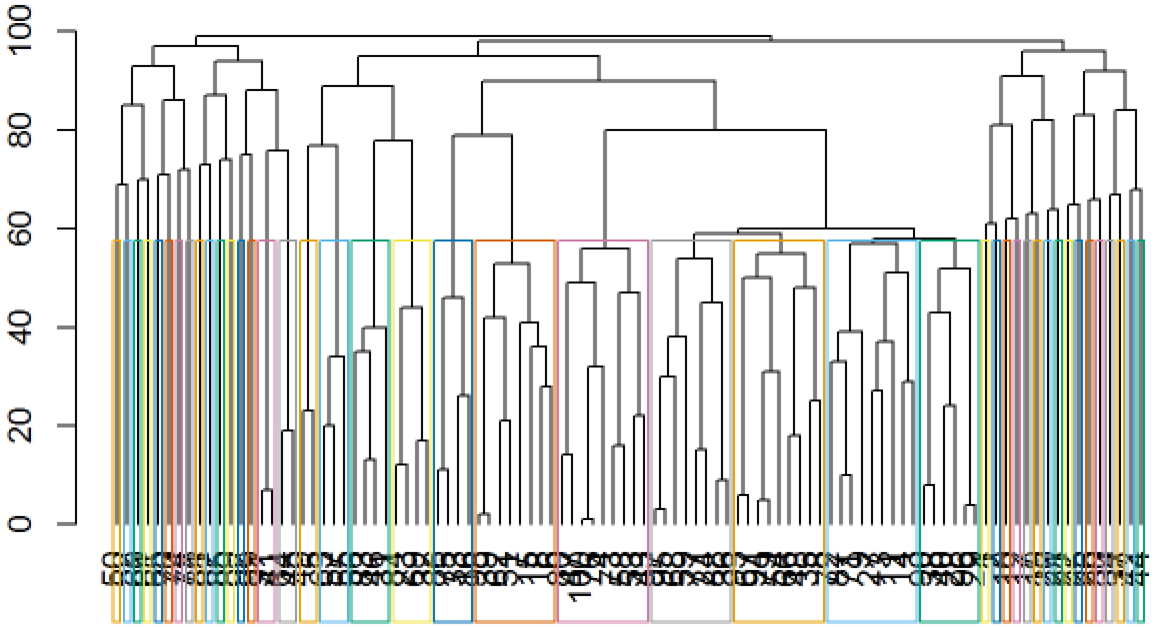
- Get the number of cliques and the number of persons in each clique
- Plot a cluster analysis of the communities (using the hclust/hierarchical clustering method)



```
sapply(cliques(er), length) # clique sizes
```

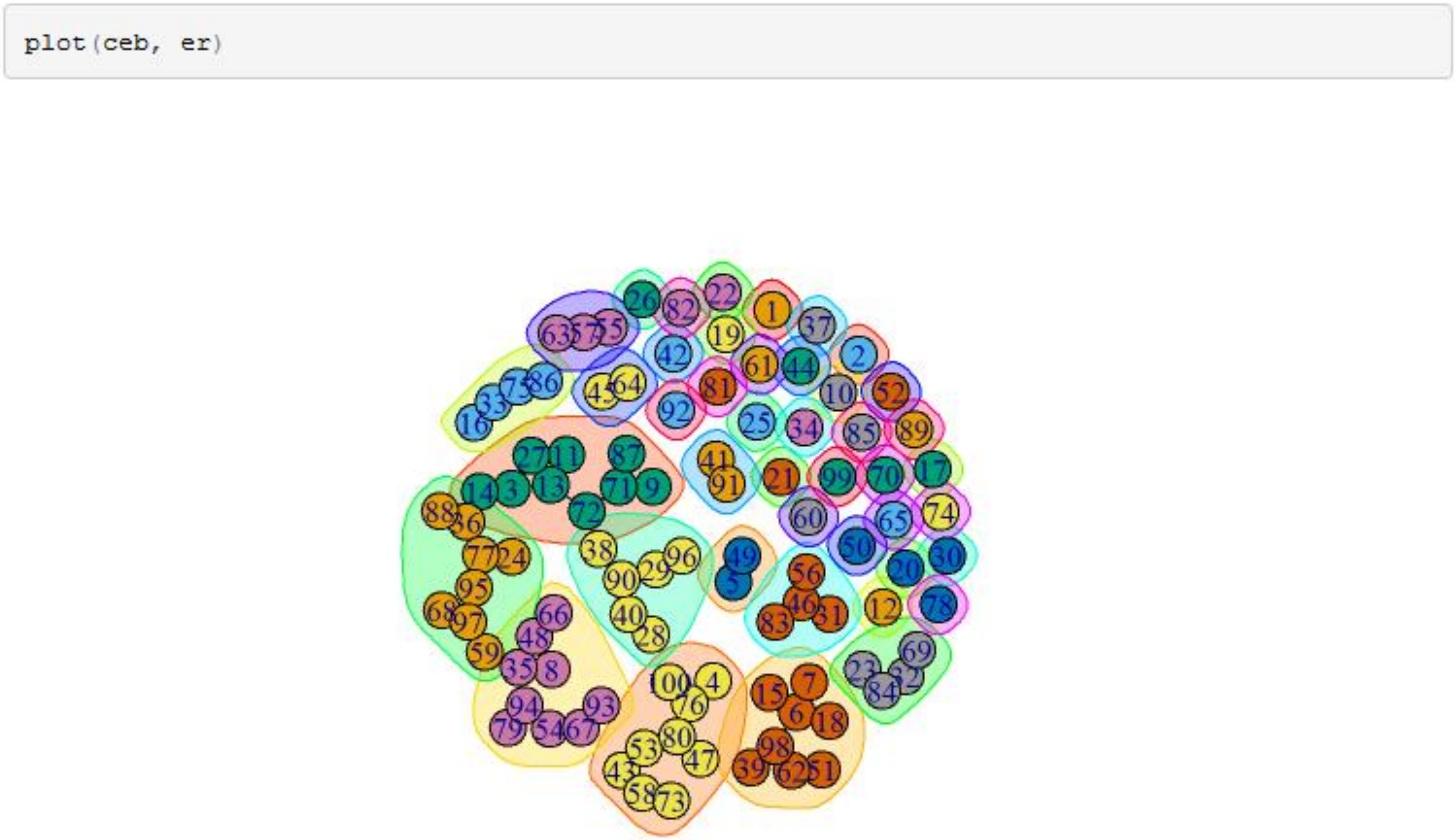
```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 2 1 2 2
##  [71] 2 1 2 2 1 2 1 2 1 2 1 2 1 2 2 2 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 2 2 1
## [106] 2 1 2 1 2 1 2 2 1 2 2 2 1 2 2 1 2 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2
## [141] 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2
```

```
ceb <- cluster_edge_betweenness(er)
dendPlot(ceb, mode="hclust")
```



Community clustering

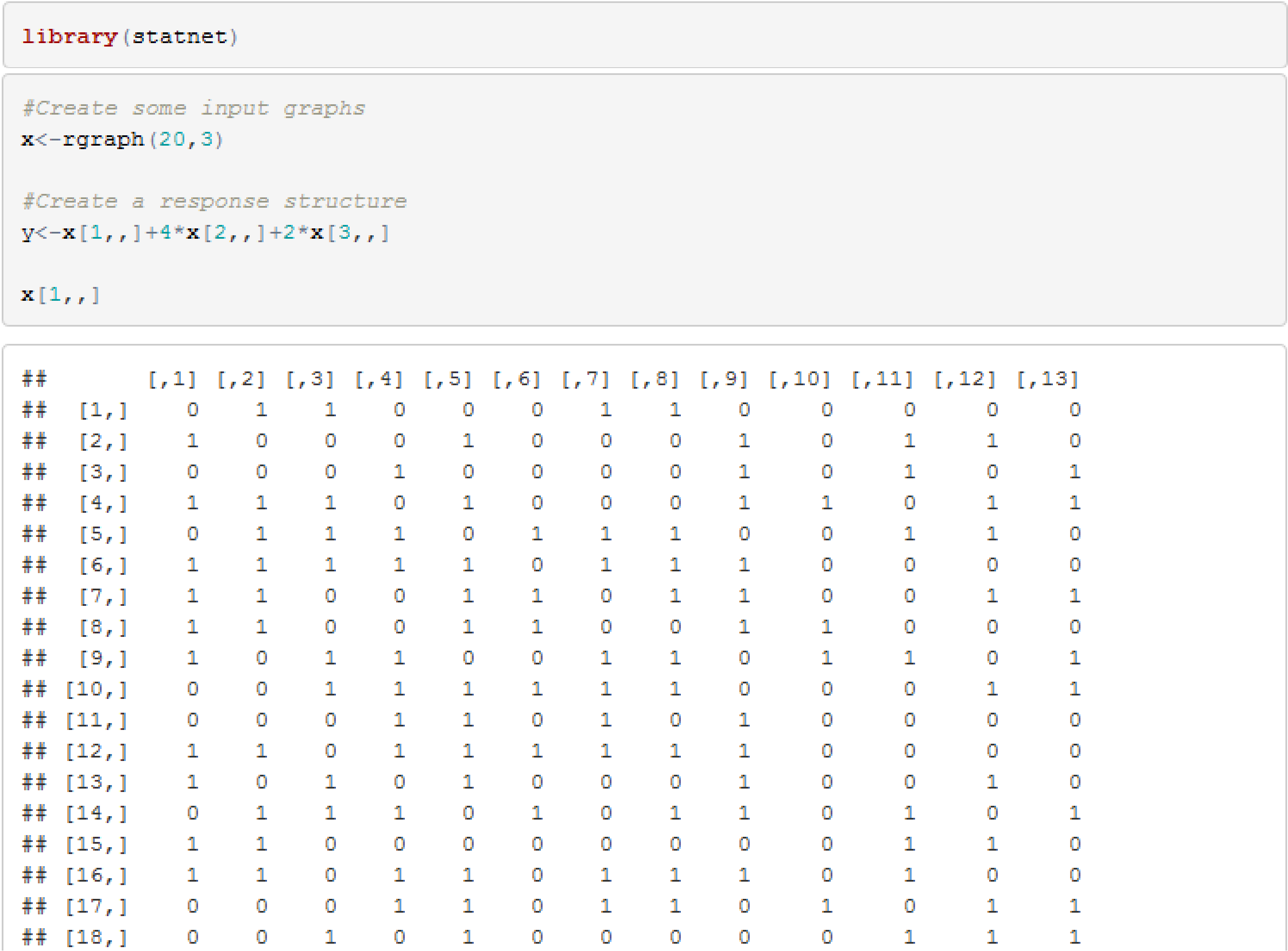
- Plot the results of the clustering method to see the communities as defined by the hculst algorithm



```
plot(ceb, er)
```

QAP Regression

- Create three “x” matrices, each with different patterns of associations
- Create “y” as a composite of the x values
- Look at the first x matrix of assoications



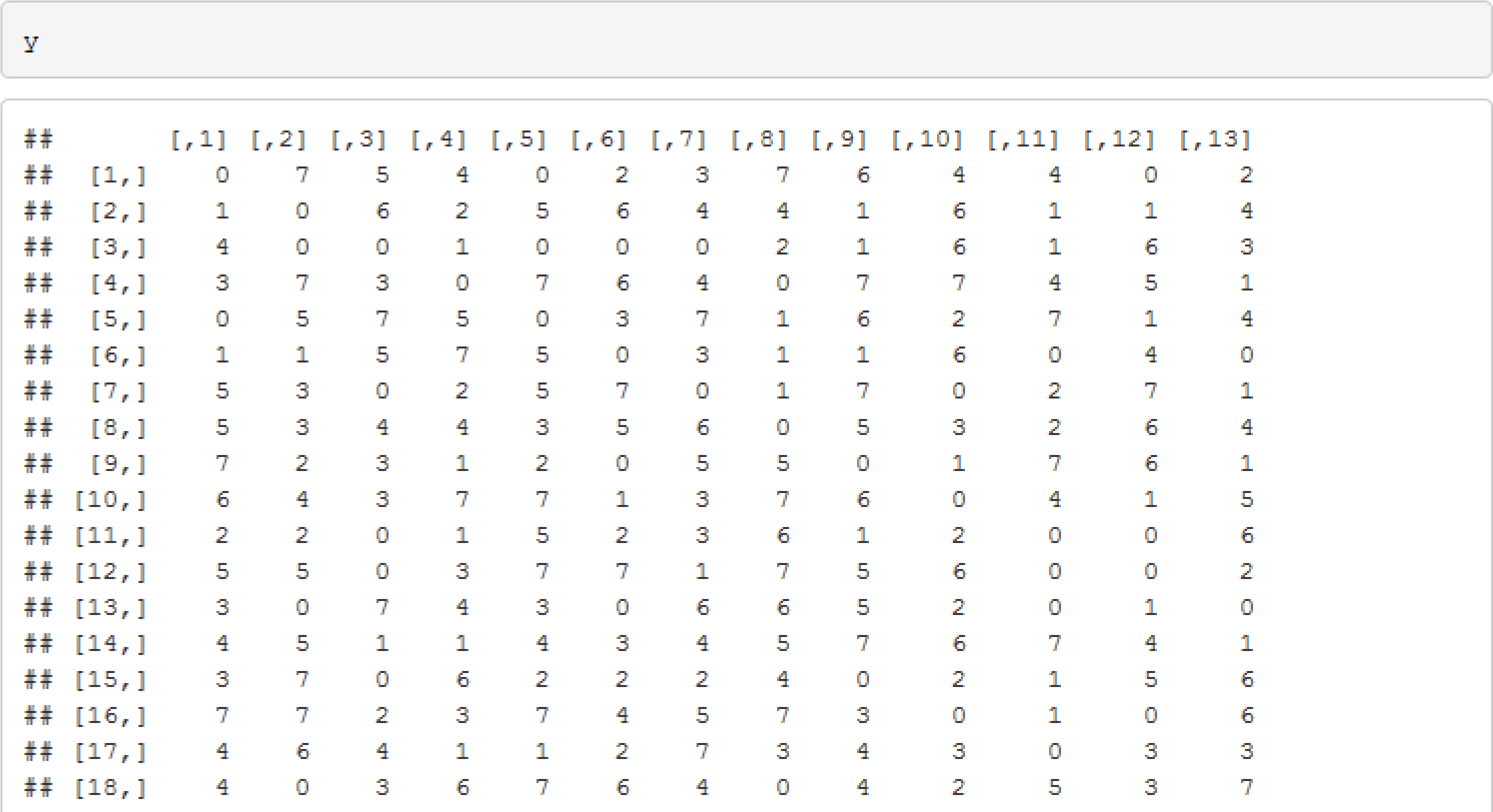
```
library(statnet)
```

```
#Create some input graphs
x<-rgraph(20,3)

#Create a response structure
y<-x[1,,]+4*x[2,,]+2*x[3,,]

x[1,,]
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    1    1    0    0    0    1    1    0     0     0     0     0
##  [2,]    1    0    0    0    1    0    0    0    1     0     1     1     0
##  [3,]    0    0    0    1    0    0    0    0    1     0     1     0     1
##  [4,]    1    1    1    0    1    0    0    0    1     1     0     1     1
##  [5,]    0    1    1    1    0    1    1    1    0     0     1     1     0
##  [6,]    1    1    1    1    1    0    1    1    1     0     0     0     0
##  [7,]    1    1    0    0    1    1    0    1    1     0     0     1     1
##  [8,]    1    1    0    0    1    1    0    0    1     1     0     0     0
##  [9,]    1    0    1    1    0    0    1    1    0     1     1     0     1
## [10,]    0    0    1    1    1    1    1    1    0     0     0     1     1
## [11,]    0    0    0    1    1    0    1    0    1     0     0     0     0
## [12,]    1    1    0    1    1    1    1    1    1     0     0     0     0
## [13,]    1    0    1    0    1    0    0    0    1     0     0     1     0
## [14,]    0    1    1    1    0    1    0    1    1     0     1     0     1
## [15,]    1    1    0    0    0    0    0    0    0     0     1     1     0
## [16,]    1    1    0    1    1    0    1    1    1     0     1     0     0
## [17,]    0    0    0    1    1    0    1    1    0     1     0     1     1
## [18,]    0    0    1    0    1    0    0    0    0     0     1     1     1
```

QAP Regression

- Look at the resulting Y values

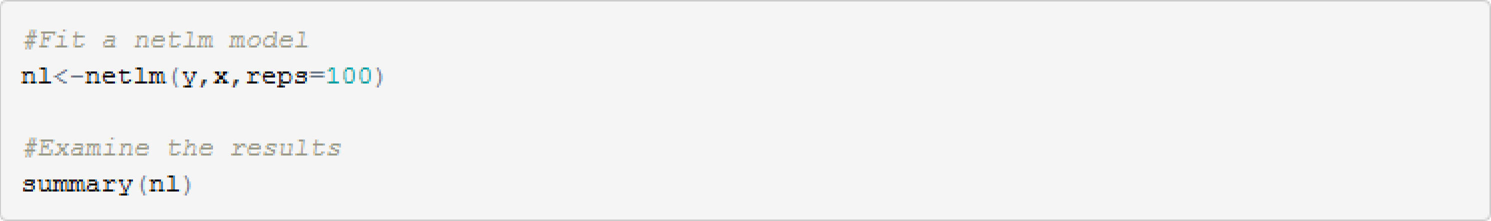


```
y
```

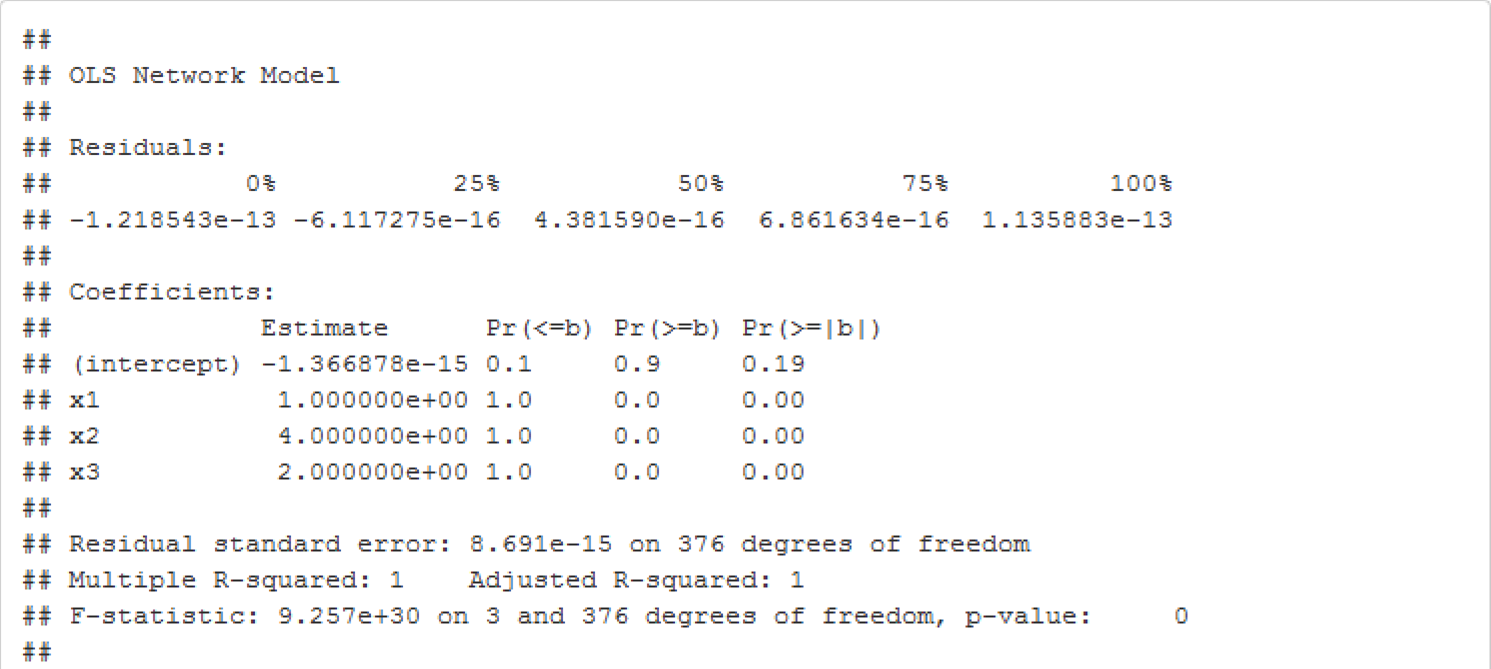
```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    7    5    4    0    2    3    7    6     4     4     0     2
##  [2,]    1    0    6    2    5    6    4    4    1     6     1     1     4
##  [3,]    4    0    0    1    0    0    0    2    1     6     1     6     3
##  [4,]    3    7    3    0    7    6    4    0    7     7     4     5     1
##  [5,]    0    5    7    5    0    3    7    1    6     2     7     1     4
##  [6,]    1    1    5    7    5    0    3    1    1     6     0     4     0
##  [7,]    5    3    0    2    5    7    0    1    7     0     2     7     1
##  [8,]    5    3    4    4    3    5    6    0    5     3     2     6     4
##  [9,]    7    2    3    1    2    0    5    5    0     1     7     6     1
## [10,]    6    4    3    7    7    1    3    7    6     0     4     1     5
## [11,]    2    2    0    1    5    2    3    6    1     2     0     0     6
## [12,]    5    5    0    3    7    7    1    7    5     6     0     0     2
## [13,]    3    0    7    4    3    0    6    6    5     2     0     1     0
## [14,]    4    5    1    1    4    3    4    5    7     6     7     4     1
## [15,]    3    7    0    6    2    2    2    4    0     2     1     5     6
## [16,]    7    7    2    3    7    4    5    7    3     0     1     0     6
## [17,]    4    6    4    1    1    2    7    3    4     3     0     3     3
## [18,]    4    0    3    6    7    6    4    0    4     2     5     3     7
```

QAP regression

- Run QAP regression predicting the Y values from the x matrices
- Use 100 replications for the permutation test



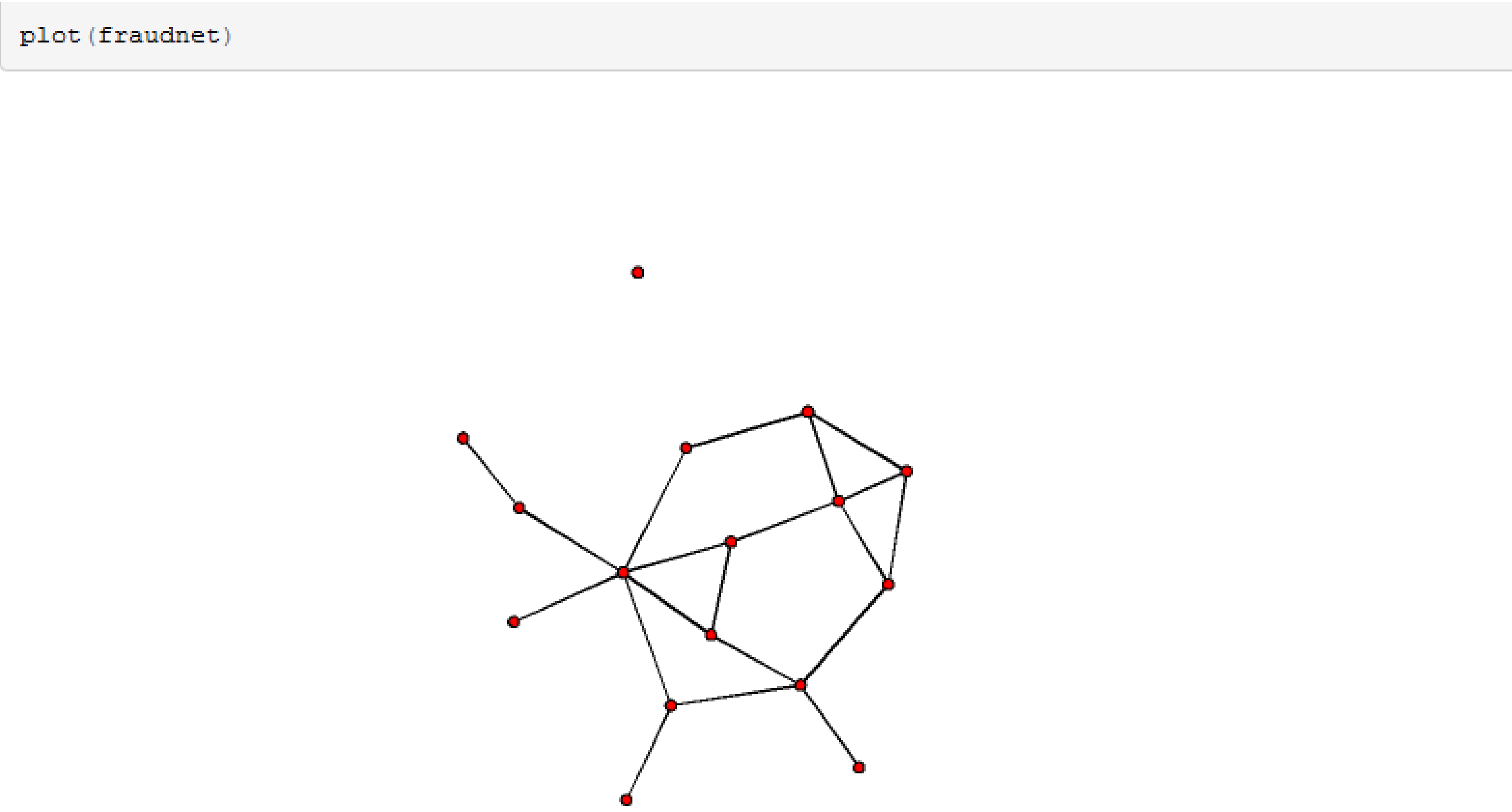
```
#Fit a netlm model
nl<-netlm(y,x,reps=100)

#Examine the results
summary(nl)
```



```
##
## OLS Network Model
##
## Residuals:
##             0%           25%           50%           75%          100%
## -1.218543e-13 -6.117275e-16  4.381590e-16  6.861634e-16  1.135883e-13
##
## Coefficients:
##             Estimate      Pr(<=b) Pr(>=b) Pr(>=|b|)
## (intercept) -1.366878e-15 0.1     0.9     0.19
## x1           1.000000e+00 1.0     0.0     0.00
## x2           4.000000e+00 1.0     0.0     0.00
## x3           2.000000e+00 1.0     0.0     0.00
##
## Residual standard error: 8.691e-15 on 376 degrees of freedom
## Multiple R-squared: 1    Adjusted R-squared: 1
## F-statistic: 9.257e+30 on 3 and 376 degrees of freedom, p-value:     0
##
```

ERGM example

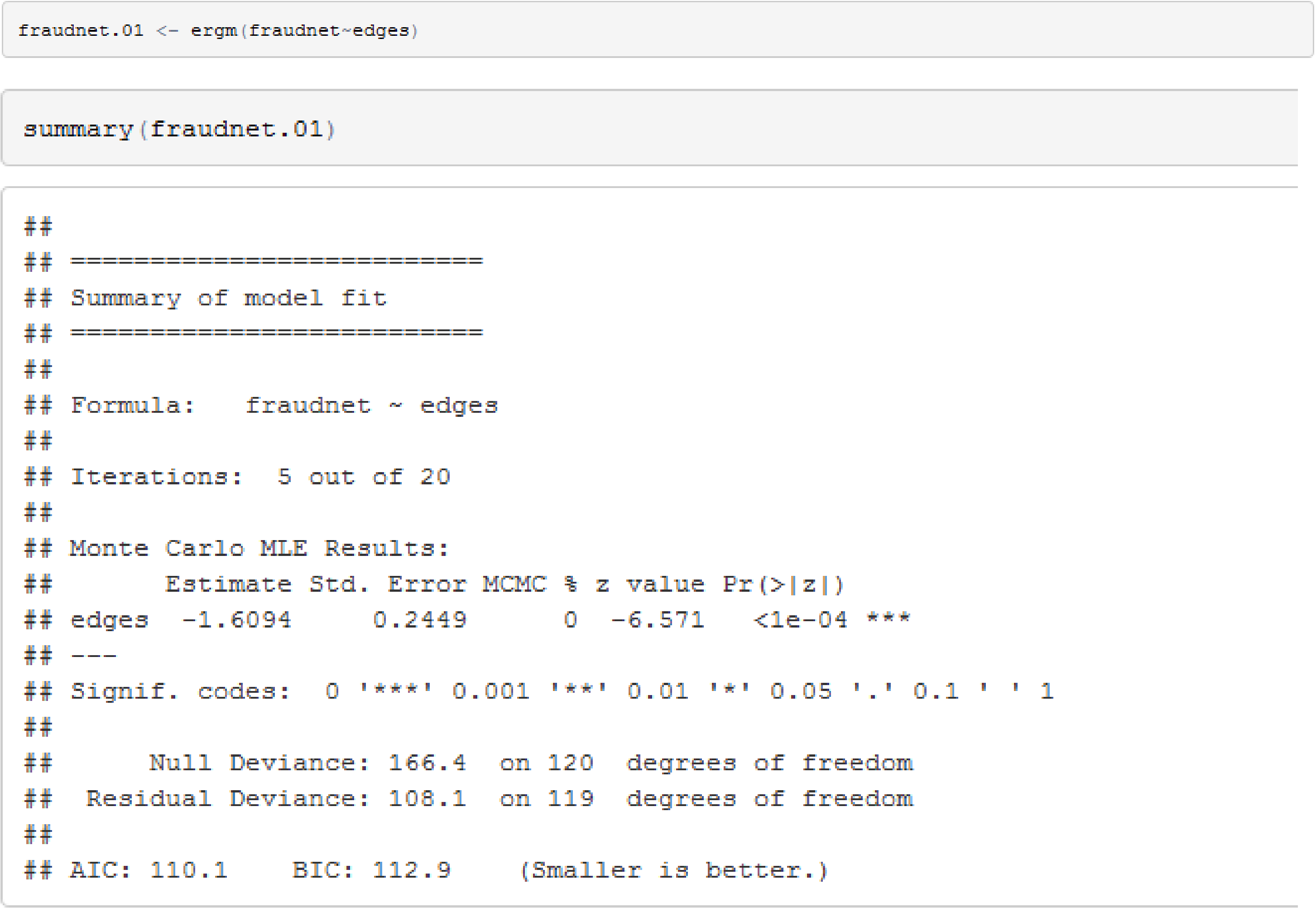
- This is a toy network linking individuals based on test security analyses



```
plot(fraudnet)
```

ERGM – Basic relationships

- This is a baseline model asking whether ties are more probable than not
- Similar to simple logistic regression, like an intercept



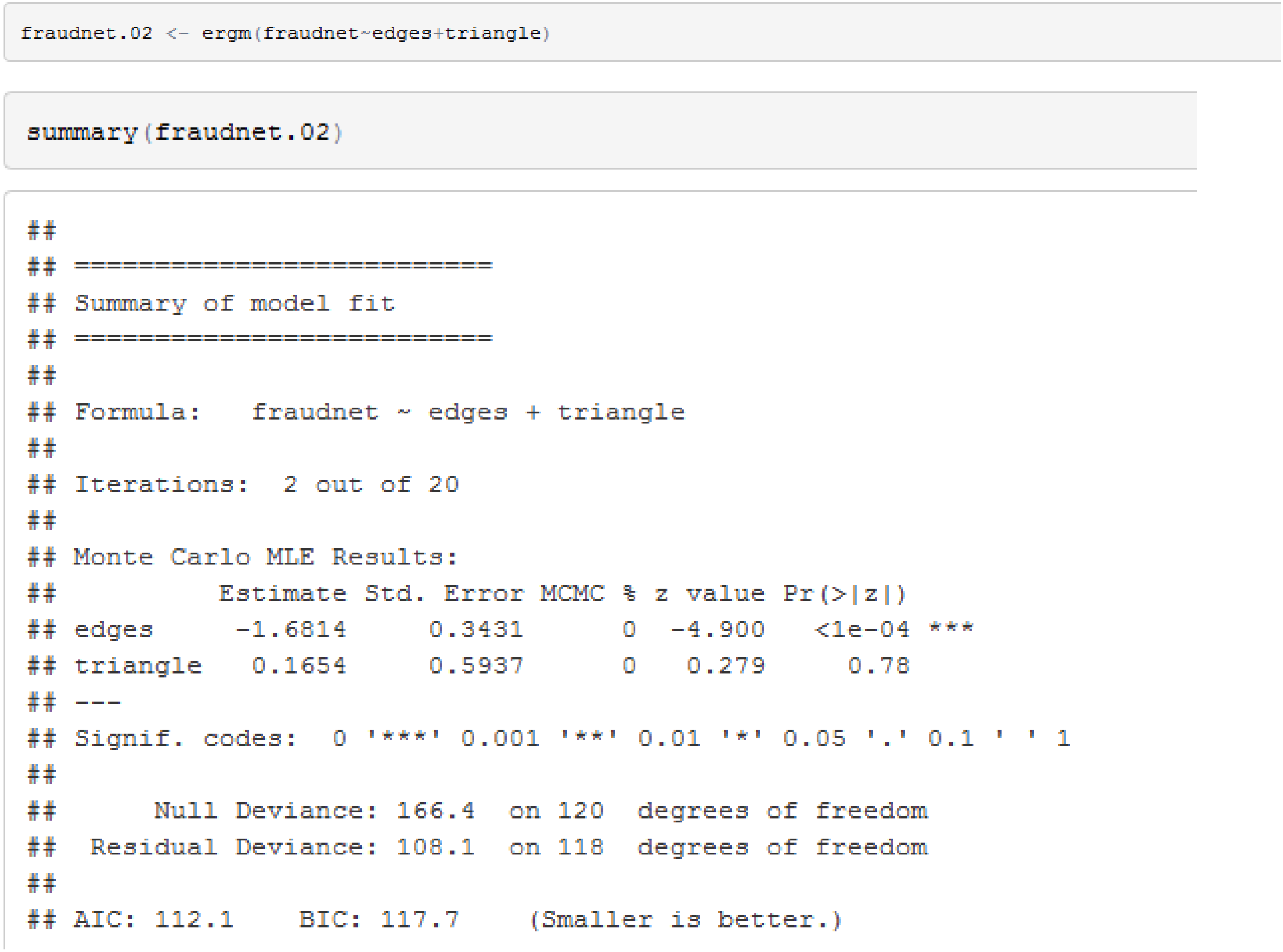
```
fraudnet.01 <- ergm(fraudnet~edges)
```

```
summary(fraudnet.01)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   fraudnet ~ edges
##
## Iterations:  5 out of 20
##
## Monte Carlo MLE Results:
##       Estimate Std. Error MCMC % z value Pr(>|z|)
## edges  -1.6094     0.2449      0  -6.571   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 166.4  on 120  degrees of freedom
##  Residual Deviance: 108.1  on 119  degrees of freedom
##
## AIC: 110.1    BIC: 112.9    (Smaller is better.)
```

ERGM – Assessing relationships

- Now we add closed relationships like cliques
- The triangle term is not significant, suggesting we don’t have more triangle terms than expected by chance

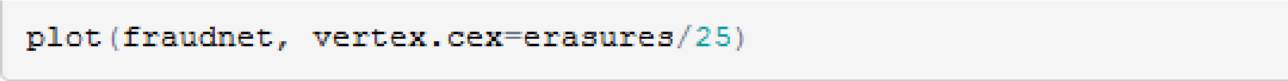


```
fraudnet.02 <- ergm(fraudnet~edges+triangle)
```

```
summary(fraudnet.02)
```

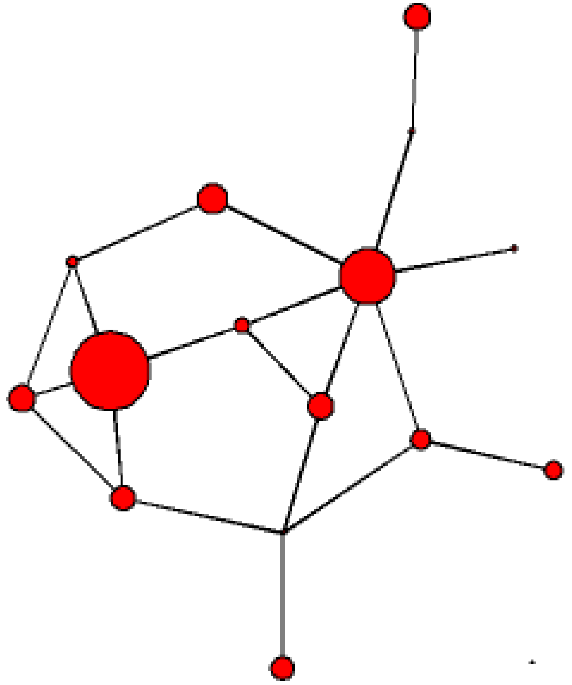
```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   fraudnet ~ edges + triangle
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##          Estimate Std. Error MCMC % z value Pr(>|z|)
## edges     -1.6814     0.3431      0  -4.900   <1e-04 ***
## triangle   0.1654     0.5937      0   0.279     0.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 166.4  on 120  degrees of freedom
##  Residual Deviance: 108.1  on 118  degrees of freedom
##
## AIC: 112.1    BIC: 117.7    (Smaller is better.)
```

ERGM – Actor size manipulation



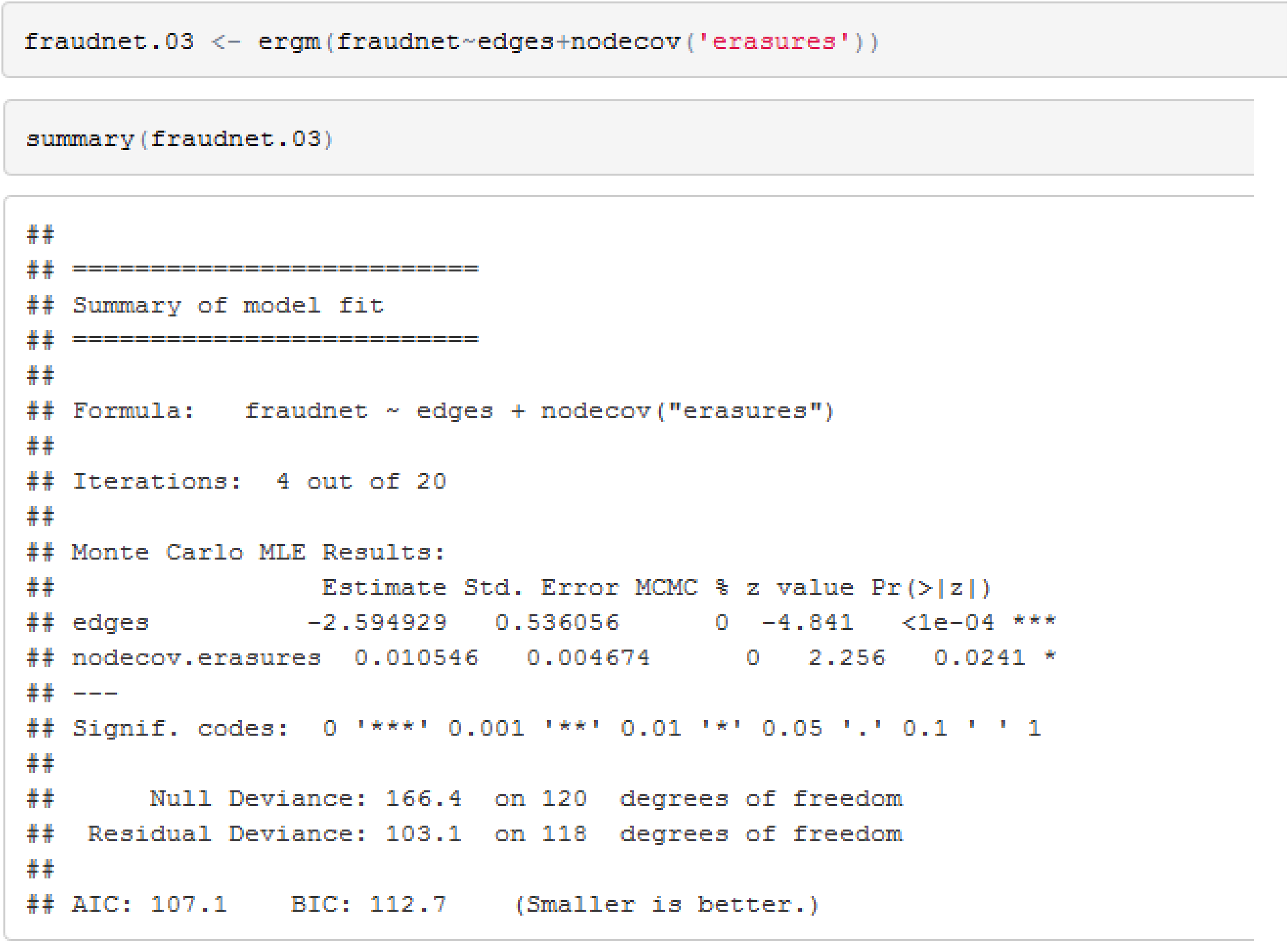
```
plot(fraudnet, vertex.cex=erasures/25)
```

- The size of the actor indicates the number of unusual erasures for that individual
- Do these erasures have a relationship with ties, or are they independent?



ERGM – Add actor characteristics

- Erasures are useful for predicting ties between actors



```
fraudnet.03 <- ergm(fraudnet~edges+nodecov('erasures'))
```

```
summary(fraudnet.03)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   fraudnet ~ edges + nodecov("erasures")
##
## Iterations:  4 out of 20
##
## Monte Carlo MLE Results:
##                 Estimate Std. Error MCMC % z value Pr(>|z|)
## edges          -2.594929   0.536056      0  -4.841   <1e-04 ***
## nodecov.erasures  0.010546   0.004674      0   2.256   0.0241 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 166.4  on 120  degrees of freedom
##  Residual Deviance: 103.1  on 118  degrees of freedom
##
## AIC: 107.1    BIC: 112.7    (Smaller is better.)
```

Thank you!

JGrochowalski@collegeboard.org

Bibliography

Lay and technical overviews of statistical network theory

- Ahuja, R. K. (2017). Network flows: theory, algorithms, and applications. Pearson Education.
- Borgatti, S. P., Everett, M. G., & Johnson, J. C. (2018). *Analyzing social networks*. Sage.
- Crane, H. (2018). Probabilistic foundations of statistical network analysis. Chapman & Hall.
- Krackhardt, D. (1988). Predicting with networks: Nonparametric multiple regression analysis of dyadic data. *Social networks*, *10*(4), 359-381.
- Lusher, D., Koskinen, J., & Robins, G. (Eds.). (2013). Exponential random graph models for social networks: Theory, methods, and applications. Cambridge University Press.