Social network analysis with R sna package

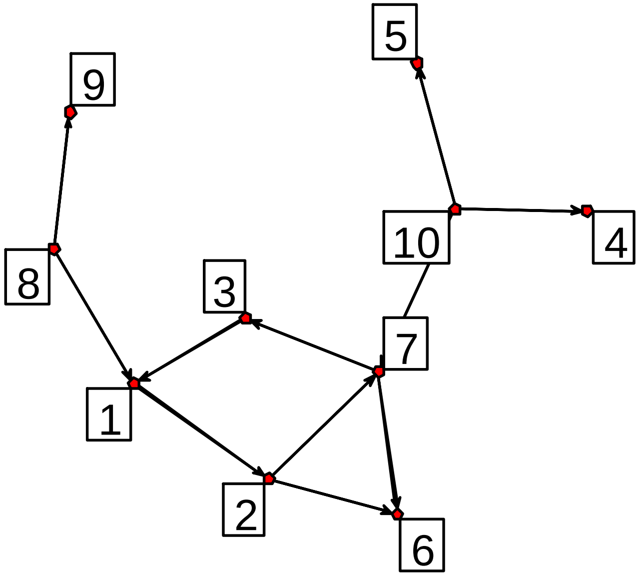
George Zhang

iResearch Consulting Group (China)
bird@iresearch.com.cn
birdzhangxiang@gmail.com

Social network (graph) definition

- G = (V,E)
  - Max edges = $\binom{N}{2}$
  - All possible E edge graphs = $\binom{\binom{N}{2}}{E}$
  - Linear graph(without parallel edges and slings)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	0	0	0
3	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	1	0	0	0	0
8	1	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	1	0	1	0	0	0



Erdös, P. and Rényi, A. (1960). “On the Evolution of Random Graphs.”

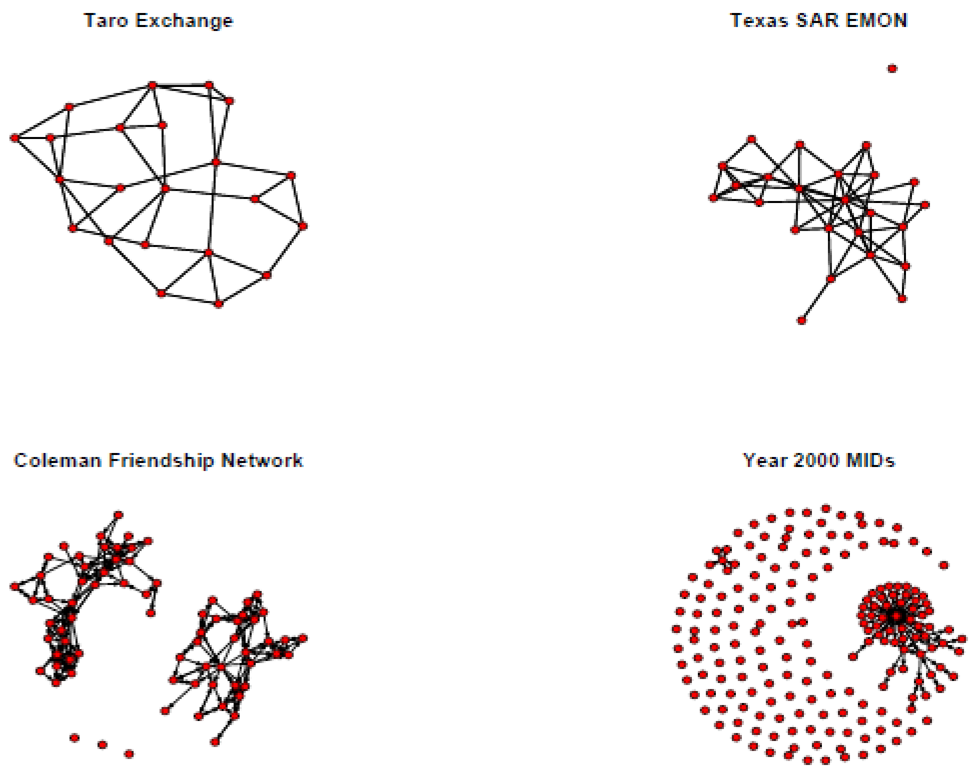
Different kinds of networks

- Random graphs
  - a graph that is generated by some random process
- Scale free network
  - whose degree distribution follows a power law
- Small world
  - most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops or steps

Differ by Graph index

- Degree distribution
- average node-to-node distance
  - average shortest path length
- clustering coefficient
  - Global, local

network examples



Butts, C.T. (2006). “Cycle Census Statistics for Exponential Random Graph Models.”

GLI-Graph level index

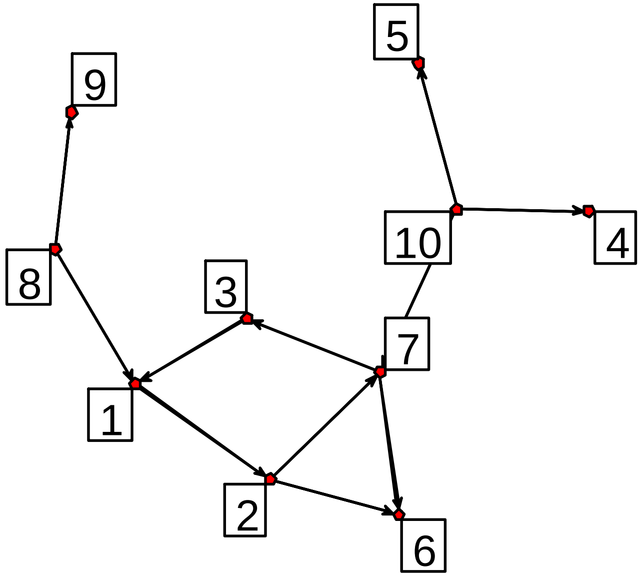
- Array statistic
  - Mean
  - Variance
  - Standard deviation
  - Skr
  - …
- Graph statistic
  - Degree
  - Density
  - Reciprocity
  - Centralization
  - …

Simple graph measurements

- Degree
  - Number of links to a vertex(indegree, outdegree…)
- Density
  - sum of tie values divided by the number of possible ties
- Reciprocity
  - the proportion of dyads which are symmetric
- Mutuality
  - the number of complete dyads
- Transtivity
  - the total number of transitive triads is computed

Example

- Degree
  - sum(g) = 11
- Density
  - gden(g) = 11/90 = 0.1222
- Reciprocity
  - grecip(g, measure="dyadic") = 0.7556
  - grecip(g,measure="edgewise") = 0
- Mutuality
  - mutuality(g) = 0
- Transtivity
  - gtrans(g) = 0.1111



Path and Cycle statistics

- kpath.census
- kcycle.census
  - dyad.census
  - Triad.census

Butts, C.T. (2006). “Cycle Census Statistics for Exponential Random Graph Models.”

Multi graph measurements

- Graph mean
  - In dichotomous case, graph mean corresponds to graph's density

$$\overline{\delta_H} = \frac{1}{|V_U|^2} \sum_{x=1}^{|V_U|} \sum_{y=1}^{|V_U|} \delta_H(x,y)$$

- Graph covariance
  - gcov/gscov

$$Cov(H_i,H_j) = \frac{1}{|V_U|^2} \sum_{x=1}^{|V_U|} \sum_{y=1}^{|V_U|} \left(\left(\delta_i(x,y) - \overline{\delta_{H_i}}\right) \left(\delta_j(x,y) - \overline{\delta_{H_j}}\right) \right)$$

- Graph correlatıon
  - gcor/gscor

$$\rho(H_i,H_j) = \frac{Cov(H_i,H_j)}{\sqrt{Var(H_i)\,Var(H_j)}}$$

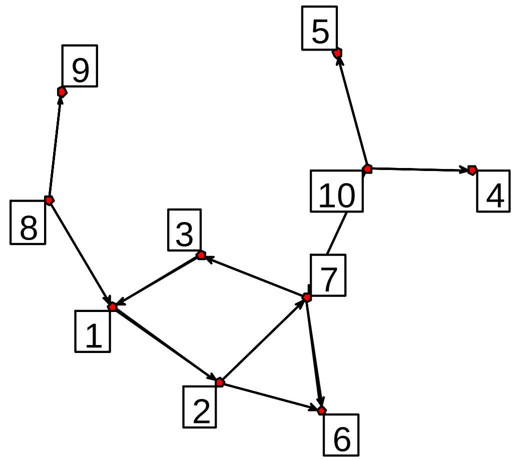
- Structural covariance
  - unlabeled graph

$$Cov_S(G_i,G_j|\mathcal{P}_i,\mathcal{P}_j) = \max_{L_a \in \mathcal{P}_i, L_b \in \mathcal{P}_j} Cov(L_a(G_i), L_b(G_j))$$

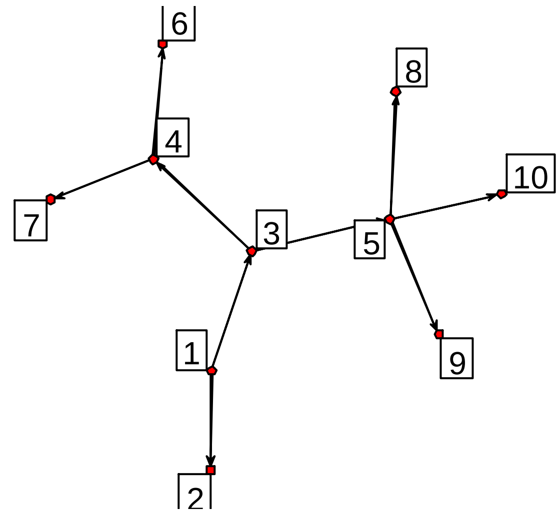
Butts, C.T., and Carley, K.M. (2001). "Multivariate Methods for Interstructural Analysis."

Example

- gcov(g1,g2) = -0.001123596
- gscov(g1,g2,exchange.list=1:10) = -0.001123596
- gscov(g1,g2)=0.04382022
  - unlabeled graph



- gcor(g1,g2) = -0.01130756
- gscor(g1,g2,exchange.list=1:10) = -0.01130756
- gscor(g1,g2) = 0.4409948
  - unlabeled graph



gcov

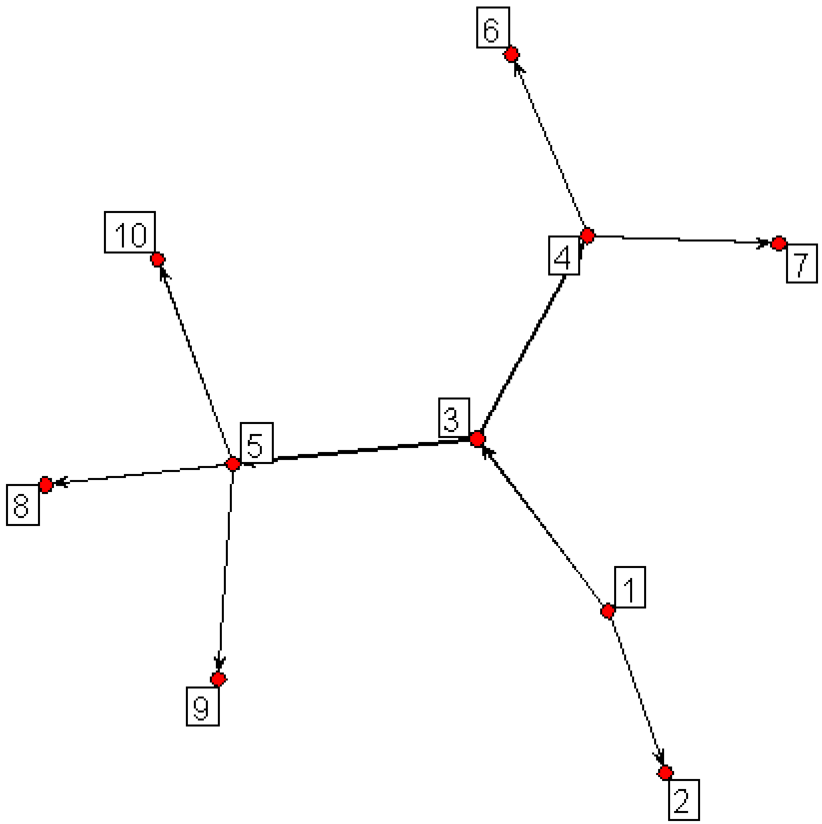
```
> v=10
> G=rgraph(v)
> H=rgraph(v)
> g1=G-sum(G)/(v*(v-1))
> g2=H-sum(H)/(v*(v-1))
> diag(g1)=0
> diag(g2)=0
> sum(g1*g2)/(v*(v-1)-1)
[1] -0.01947566
> gcov(G,H)
[1] -0.01947566
>
```

Measure of structure

- Connectedness $1-\left(\dfrac{V}{N(N-1)/2}\right)$
  - '0' for no edges
  - '1' for $\binom{N}{2}$ edges
- Hierarchy $1-\left(\dfrac{V}{MaxV}\right)$
  - '0' for all two-way links
  - '1' for all one-way links
- Efficiency $1-\left(\dfrac{V}{MaxV}\right)$
  - '0' for $\binom{N}{2}$ edgs
  - '1' for N-1 edges
- Least Upper Boundedness (lubness)
  - '0' for all vertex link into one
  - '1' for all outtree

Krackhardt,David.(1994)."Graph Theoretical Dimensionsof Informal Organizations."

Example



- Outtree
  - Connectedness=1
  - Hierarchy=1
  - Efficiency=1
  - Lubness=1

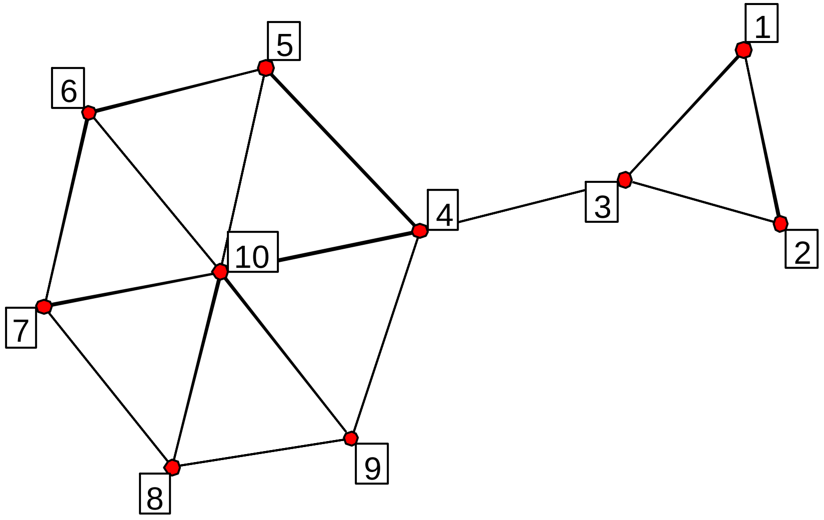
Graph centrality

- Degree
  - Number of links to a vertex(indegree, outdegree…)
- Betweenness
  - Number of shortest paths pass it
- Closeness
  - Length to all other vertices
- Centralization by 3 ways above
  - ‘0’ for all vertices has equal position(central score)
  - ‘1’ for 1 vertex be the center of the graph
- See also
  - evcent, bonpow, graphcent, infocent, prestige

Freeman,L.C.(1979). Centrality in Social Networks-Conceptual Clarification

Example

```
> centralization(g,degree,mode="graph")
[1] 0.1944444
> centralization(g,betweenness,mode="graph")
[1] 0.1026235
> centralization(g,closeness,mode="graph")
[1] 0
```



Mode="graph" means only consider indegree

Example

> centralization(g,degree,mode="graph")

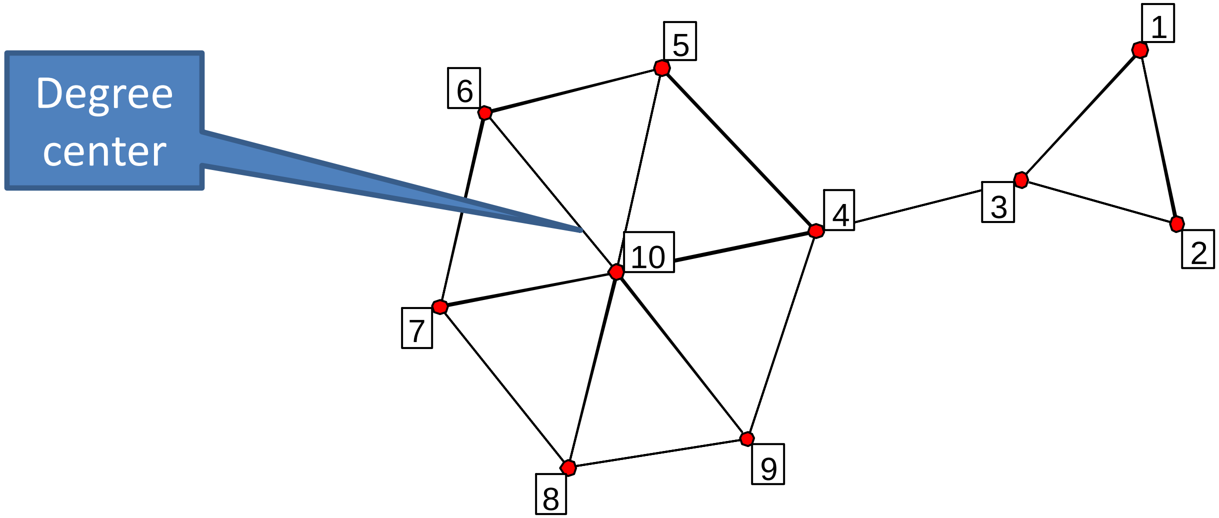
[1] 0.1944444

> centralization(g,betweenness,mode="graph")

[1] 0.1026235

> centralization(g,closeness,mode="graph")

[1] 0



Mode="graph" means only consider indegree

Example

> centralization(g,degree,mode="graph")

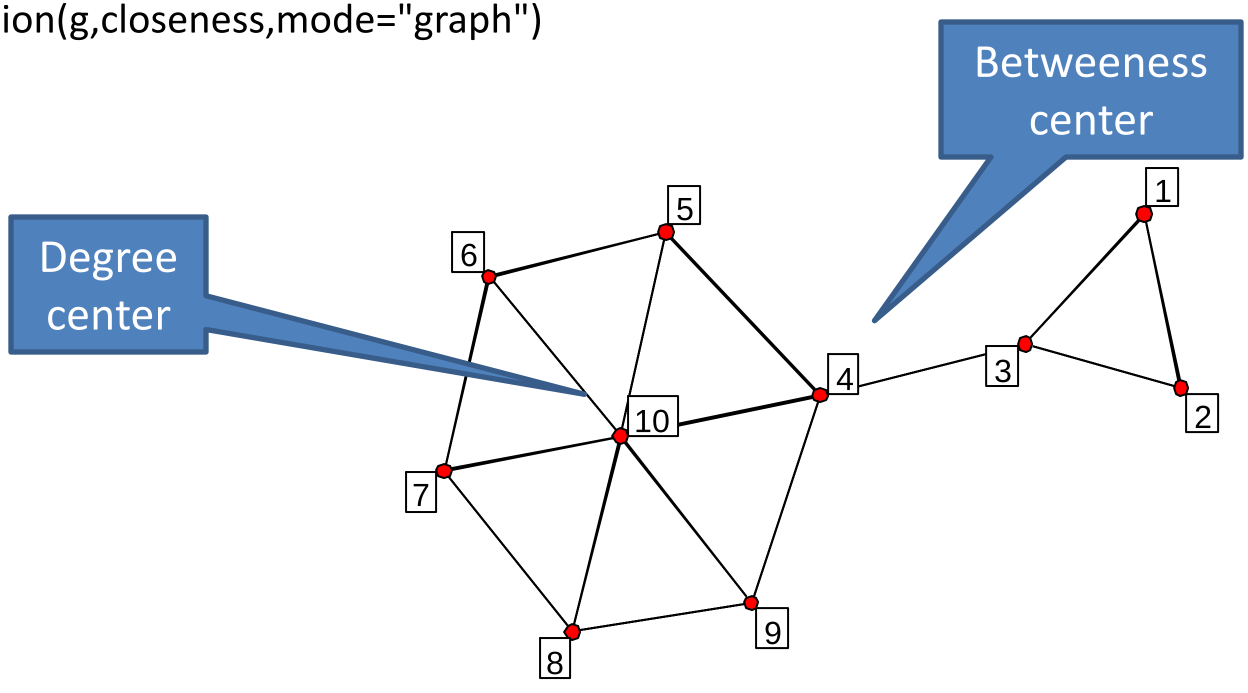
[1] 0.1944444

> centralization(g,betweenness,mode="graph")

[1] 0.1026235

> centralization(g,closeness,mode="graph")

[1] 0



Mode="graph" means only consider indegree

Example

> centralization(g,degree,mode="graph")

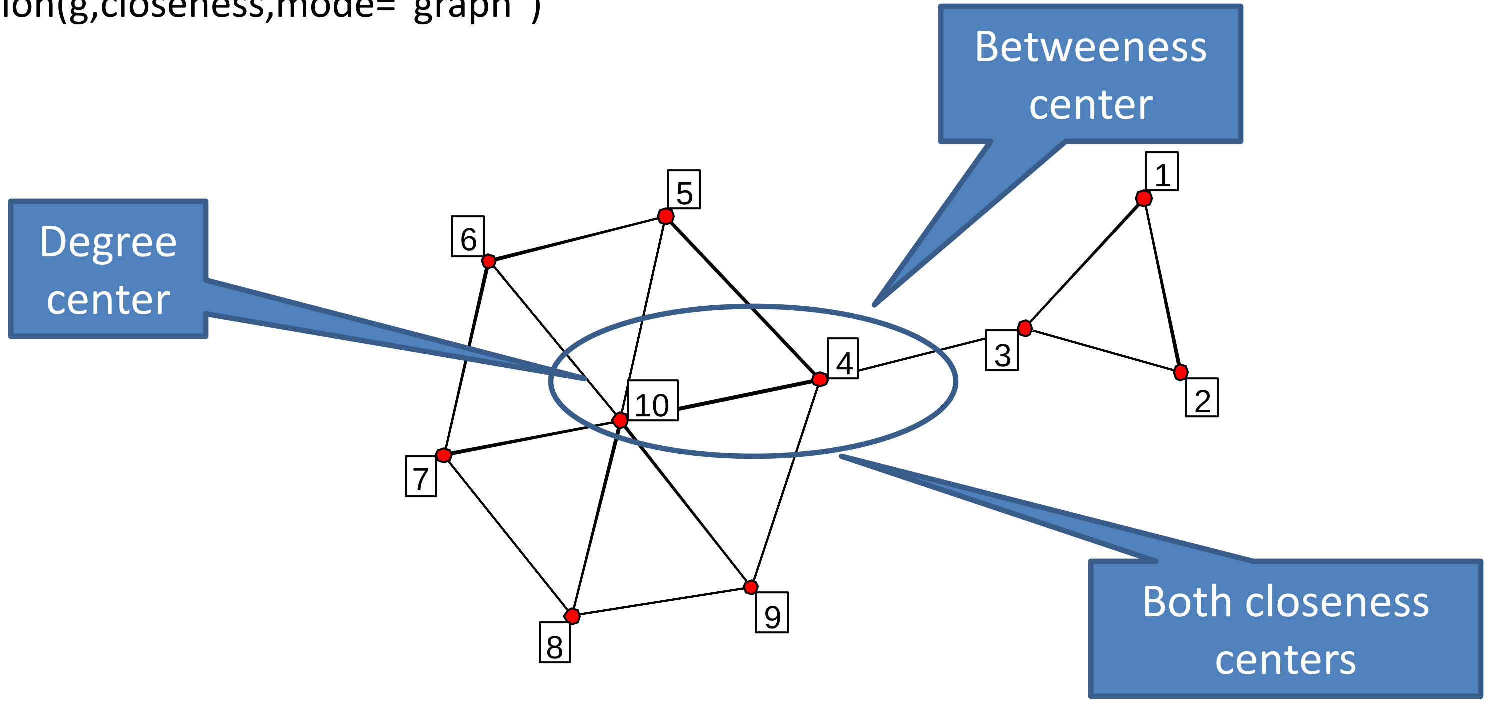
[1] 0.1944444

> centralization(g,betweenness,mode="graph")

[1] 0.1026235

> centralization(g,closeness,mode="graph")

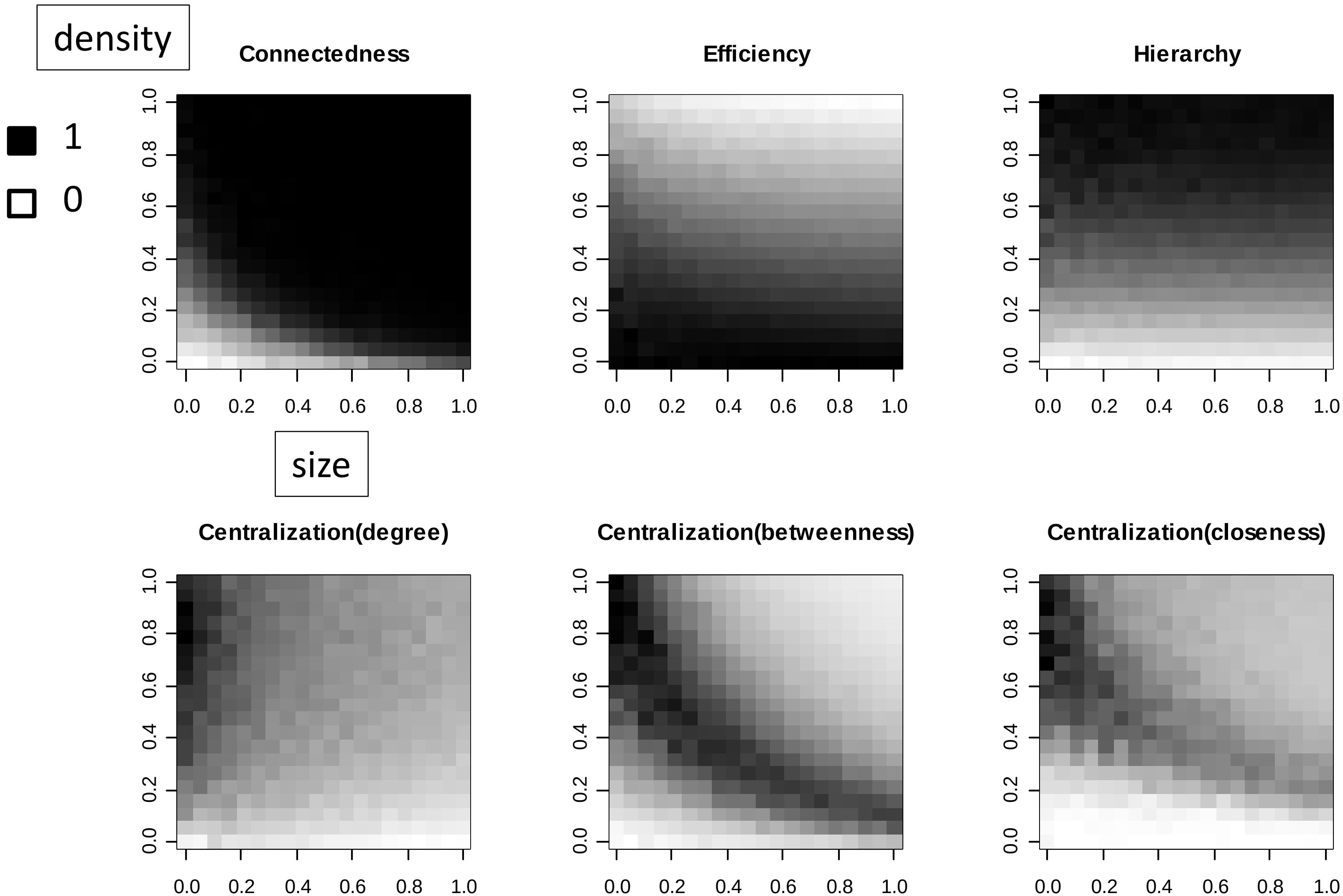
[1] 0



Mode="graph" means only consider indegree

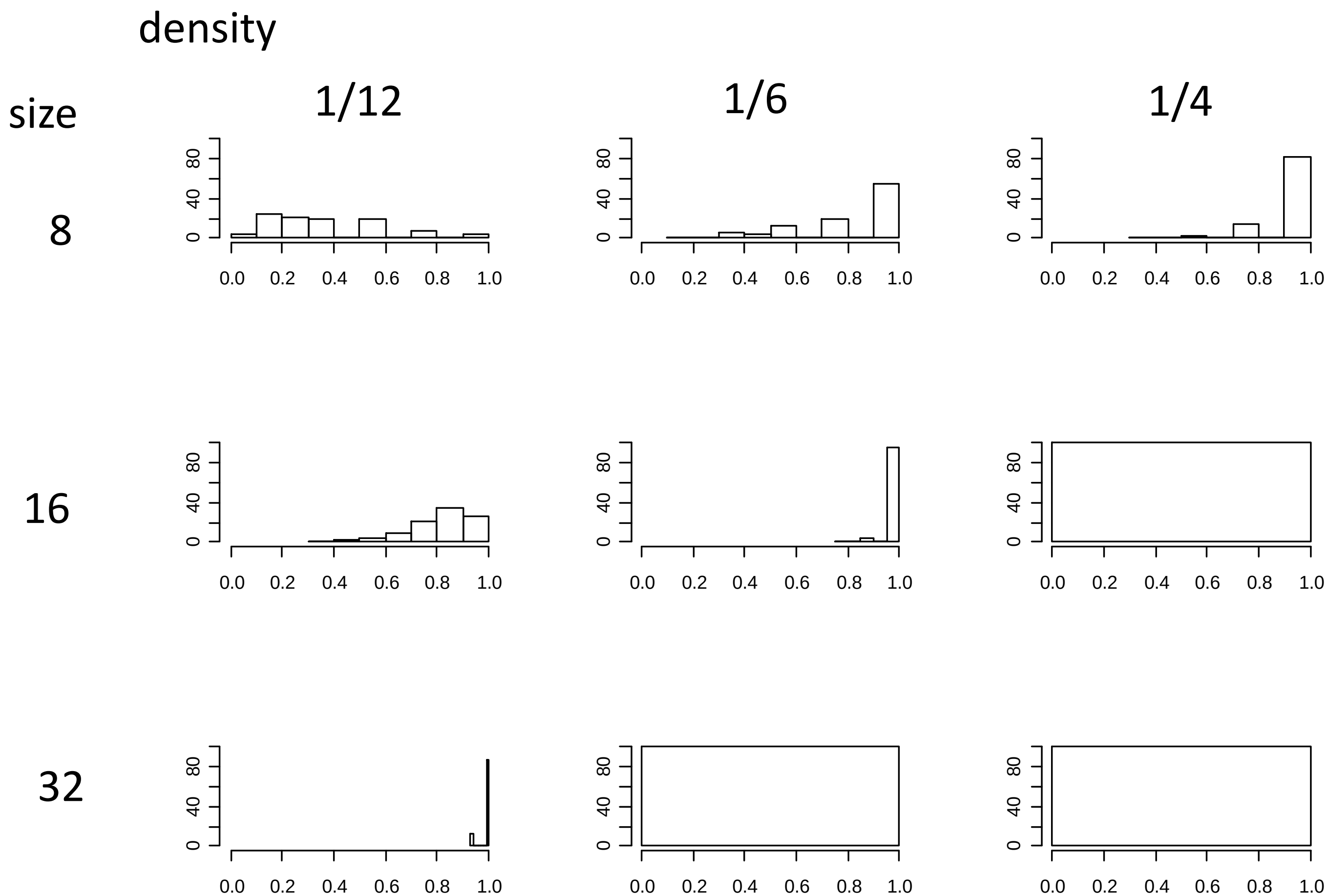
GLI relation

GLI map



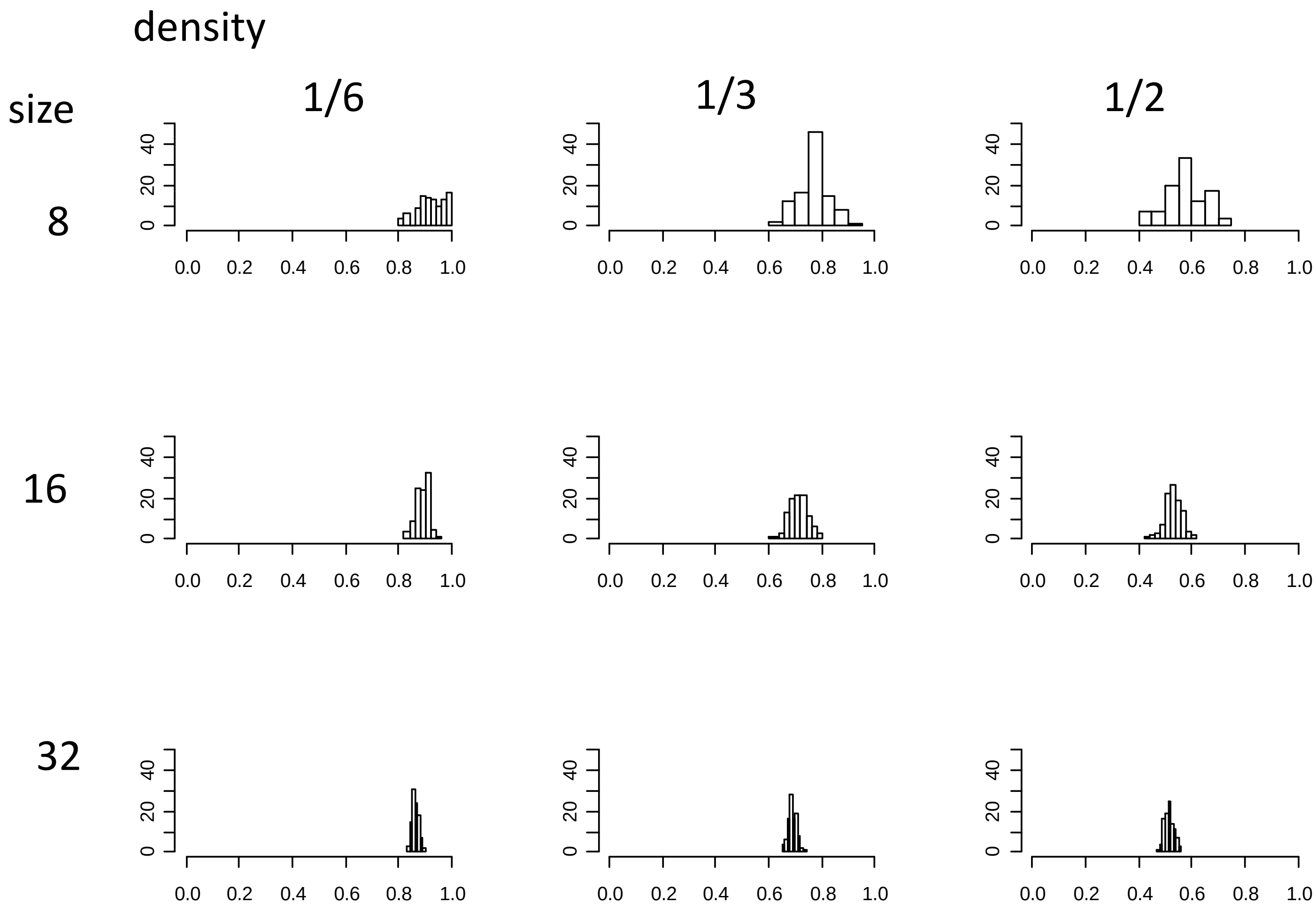
Anderson,B.S.;Butts ,C. T.;and Carley ,K. M.(1999).”The Interaction of Size and Density with Graph-Level Indices.”

connectedness distribution by graph size and density



Anderson,B.S.;Butts ,C. T.;and Carley ,K. M.(1999).”The Interaction of Size and Density with Graph-Level Indices.”

efficiency distribution by graph size and density



hierarchy distribution by graph size and density

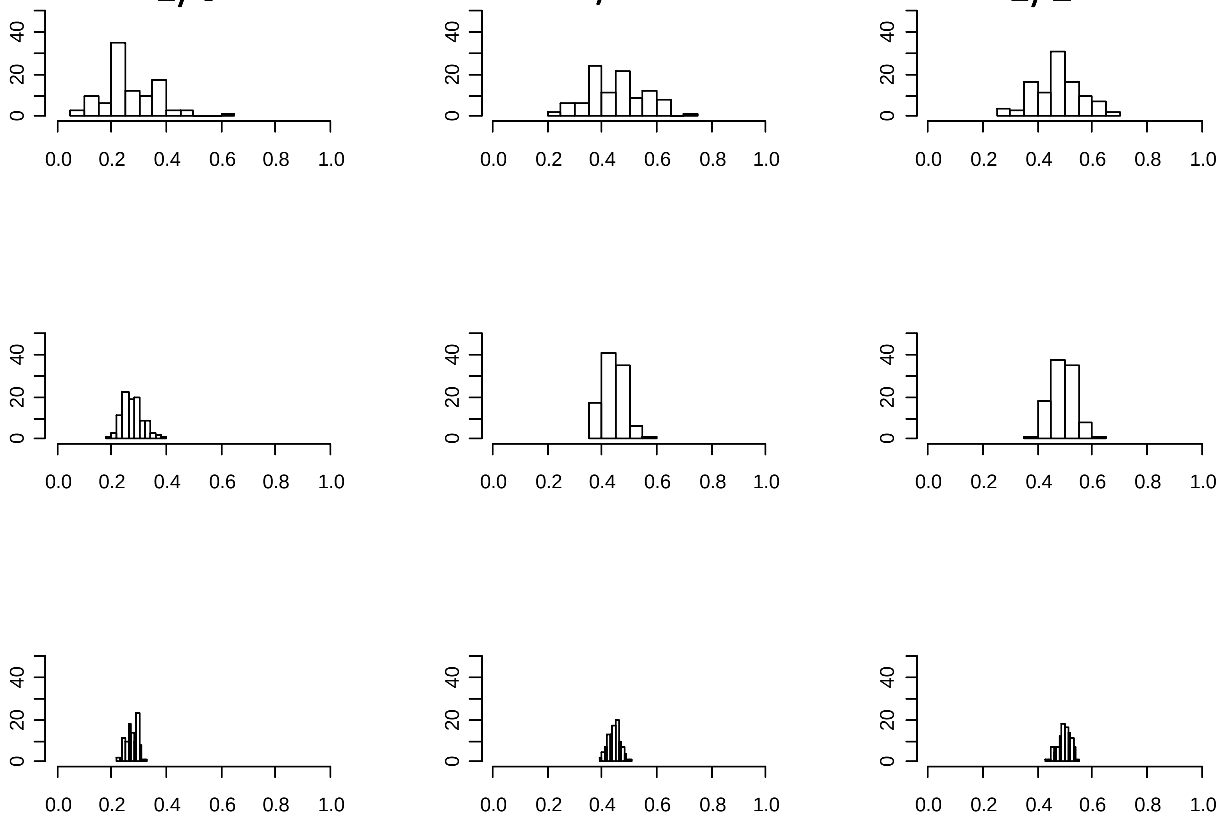
density

size | 1/6 | 1/3 | 1/2

8

16

32



GLI map R code

```
compare<-function(size,den)
{
 g=rgraph(n=size,m=100,tprob=den)
 gli1=apply(g,1,connectedness)
 gli2=apply(g,1,efficiency)
 gli3=apply(g,1,hierarchy)
 gli4=apply(g,1,function(x) centralization(x,degree))
 gli5=apply(g,1,function(x) centralization(x,betweenness))
 gli6=apply(g,1,function(x) centralization(x,closeness))
 x1=mean(gli1,na.rm=T)
 x2=mean(gli2,na.rm=T)
 x3=mean(gli3,na.rm=T)
 x4=mean(gli4,na.rm=T)
 x5=mean(gli5,na.rm=T)
 x6=mean(gli6,na.rm=T)
 return(c(x1,x2,x3,x4,x5,x6))
}
```

```
nx=20
ny=20
res=array(0,c(nx,ny,6))
size=5:26
den=seq(0.05,0.5,length.out=20)
for(i in 1:nx)
 for(j in 1:ny)
    res[i,j,]=compare(size[i],den[j])

#image(res,col=gray(1000:1/1000))

par(mfrow=c(2,3))
image(res[,,1],col=gray(1000:1/1000),main="Connectedness")
image(res[,,2],col=gray(1000:1/1000),main="Efficiency")
image(res[,,3],col=gray(1000:1/1000),main="Hierarchy")
image(res[,,4],col=gray(1000:1/1000),main="Centralization(degree)")
image(res[,,5],col=gray(1000:1/1000),main="Centralization(betweenness)")
image(res[,,6],col=gray(1000:1/1000),main="Centralization(closeness)")
```

GLI distribution R code

```
par(mfrow=c(3,3))
for(i in 1:3)
 for(j in 1:3)
  hist(centralization(rgraph(4*2^i,100,tprob=j/4),betweenness),main="",xlab="",ylab="",xlim=range(0:1),ylim=range(0:50))
  hist(centralization(rgraph(4*2^i,100,tprob=j/4),degree),main="",xlab="",ylab="",xlim=range(0:1),ylim=range(0:50))
  hist(hierarchy(rgraph(4*2^i,100,tprob=j/6)),main="",xlab="",ylab="",xlim=range(0:1),ylim=range(0:50))
  hist(efficiency(rgraph(4*2^i,100,tprob=j/6)),main="",xlab="",ylab="",xlim=range(0:1),ylim=range(0:50))
  hist(connectedness(rgraph(4*2^i,100,tprob=j/12)),main="",xlab="",ylab="",xlim=range(0:1),ylim=range(0:100))
```

Graph distance

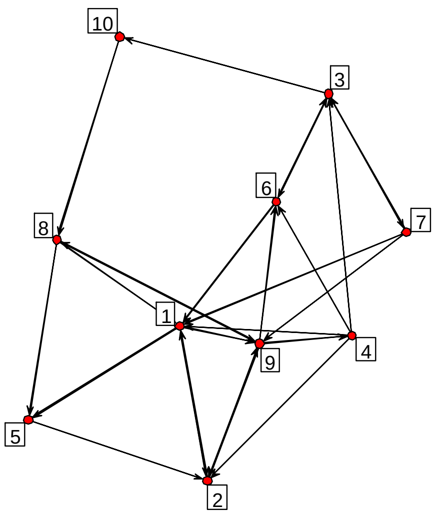
Clustering, MDS

Distance between graphs

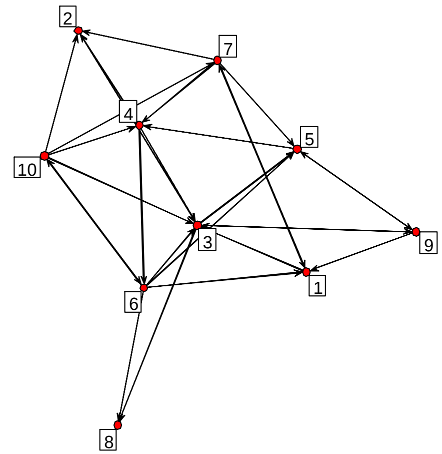
- Hamming(labeling) distance
  - $\left|\{e:(e \in E(G_1), e \notin E(G_2)) \Lambda (e \notin E(G_1), e \in E(G_2))\}\right|$
    number of addition/deletion operations required to turn the edge set of G1 into that of G2
  - 'hdist' for typical hamming distance matrix
- Structure distance
  - $d_S(G,H|L_G,L_H) = \min_{L_G,L_H} d(\ell(G),\ell(H))$
  - 'structdist' & 'sdmat' for structure distance with exchange.list of vertices

Butts, C.T., and Carley, K.M. (2001). "Multivariate Methods for Interstructural Analysis."

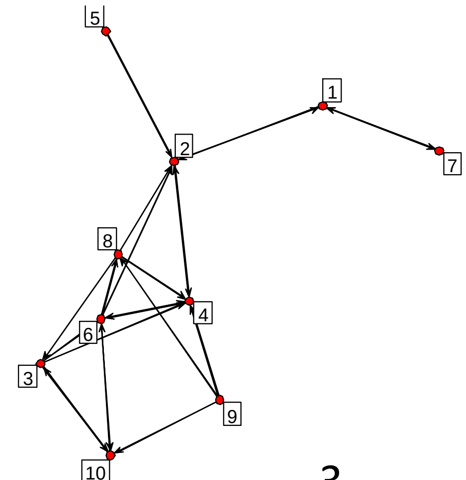
Example



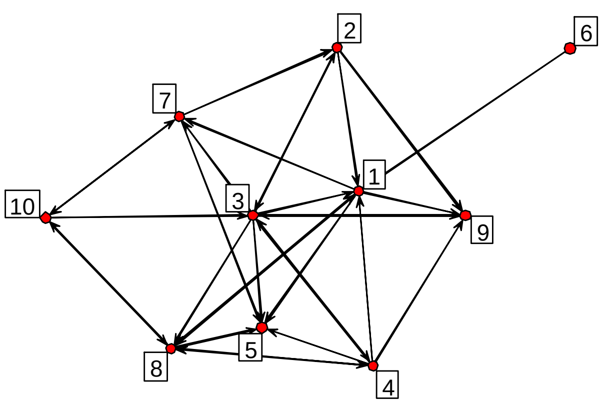
1



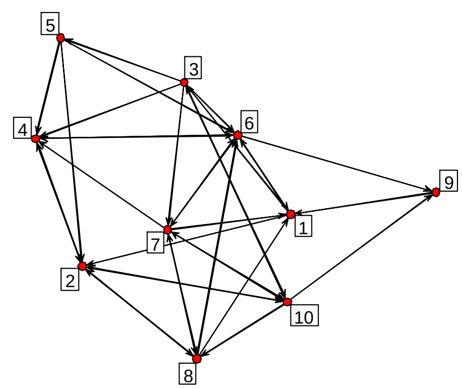
2



3



4



5

Example

hdist(g)

	1	2	3	4	5
1	0	44	29	35	39
2	44	0	35	35	39
3	29	35	0	44	34
4	35	35	44	0	48
5	39	39	34	48	0

sdmat(g)

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0	24	23	25	27
[2,]	24	0	25	27	29
[3,]	23	25	0	26	28
[4,]	25	27	26	0	28
[5,]	27	29	28	28	0

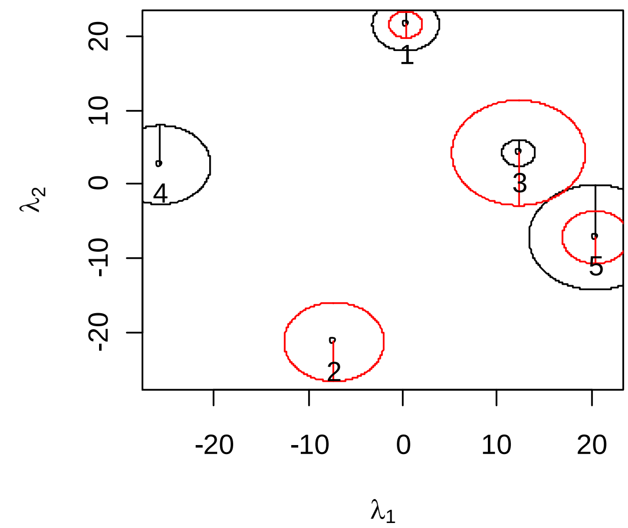
structdist(g)

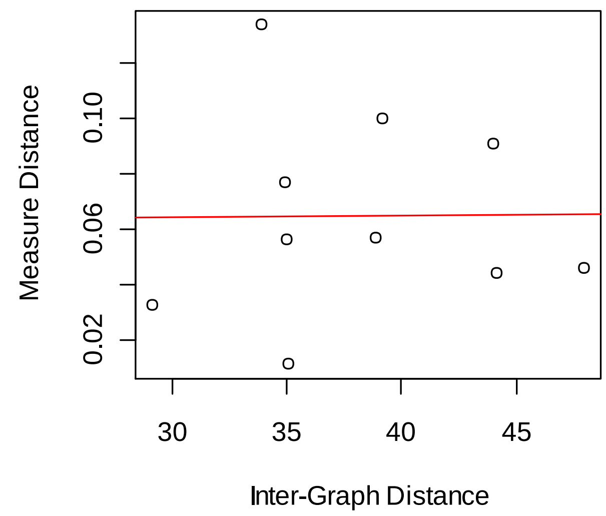
	1	2	3	4	5
1	0	22	21	23	25
2	22	0	21	21	23
3	21	21	0	20	24
4	23	23	20	0	20
5	25	23	22	20	0

Inter-Graph MDS

- ‘gdist.plotstats’
  - Plot by distances between graphs
  - Add graph level index as third or forth dimension

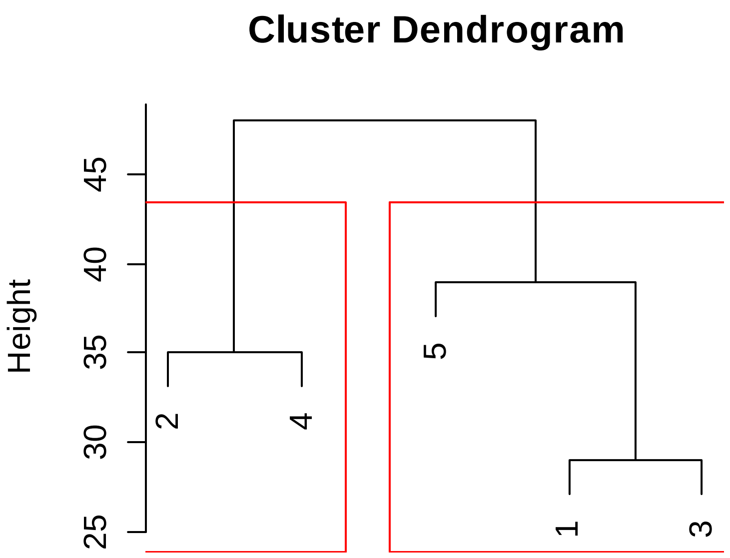
```
> g.h<-hdist(g)   #sample graph used before
> gdist.plotdiff(g.h,gden(g),lm.line=TRUE)
> gdist.plotstats(g.h,cbind(gden(g),grecip(g)))
```



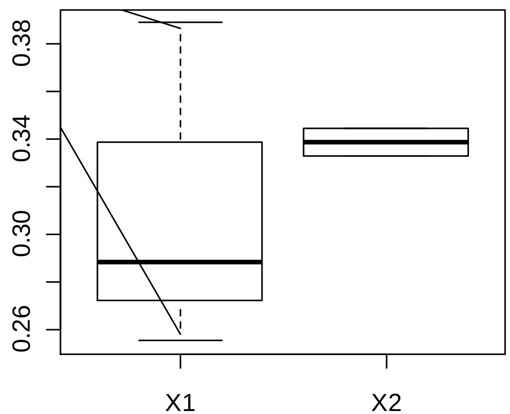


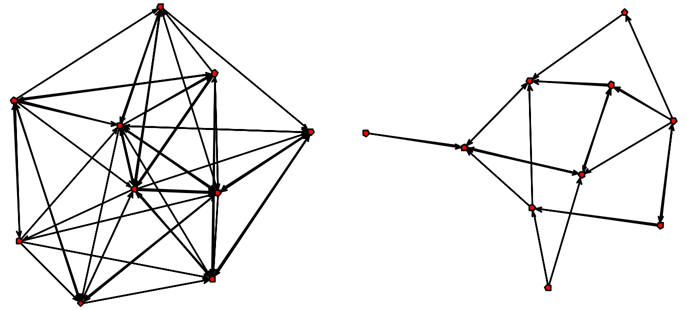
Graph clustering

- Use hamming distance
  - g.h=hdist(g)
  - g.c<-hclust(as.dist(g.h))
  - rect.hclust(g.c,2)
  - g.cg<-gclust.centralgraph(g.c,2,g)
  - gplot(g.cg[1,,])
  - gplot(g.cg[2,,])
  - gclust.boxstats(g.c,2,gden(g))



hc2 hc1





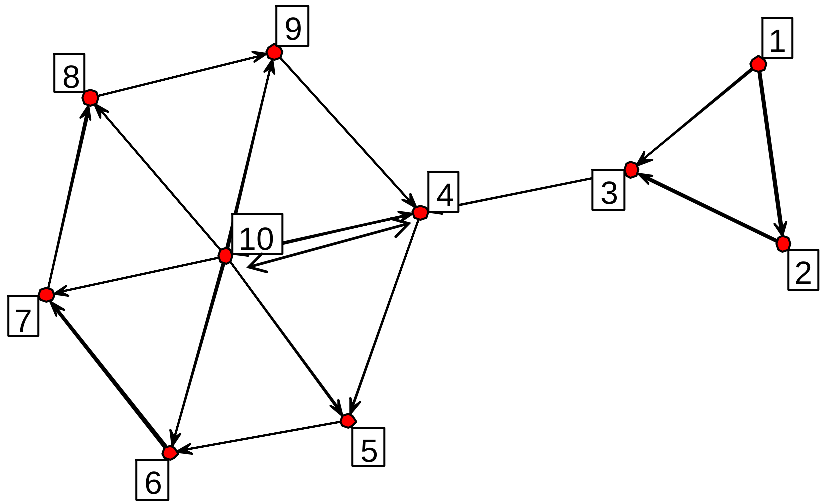
Distance between vertices

- Structural equivalence
  - ‘sedist’ with 4 methods:
    1. correlation: the product-moment correlation
    2. euclidean: the euclidean distance
    3. hamming: the Hamming distance
    4. gamma: the gamma correlation
- Path distance
  - ‘geodist’ with shortest path distance and the number of shortest pathes

Breiger, R.L.; Boorman, S.A.; and Arabie, P. (1975). “An Algorithm for Clustering Relational Data with Applications to Social Network Analysis and Comparison with Multidimensional Scaling.”

Brandes, U. (2000). “Faster Evaluation of Shortest-Path Based Centrality Indices.”

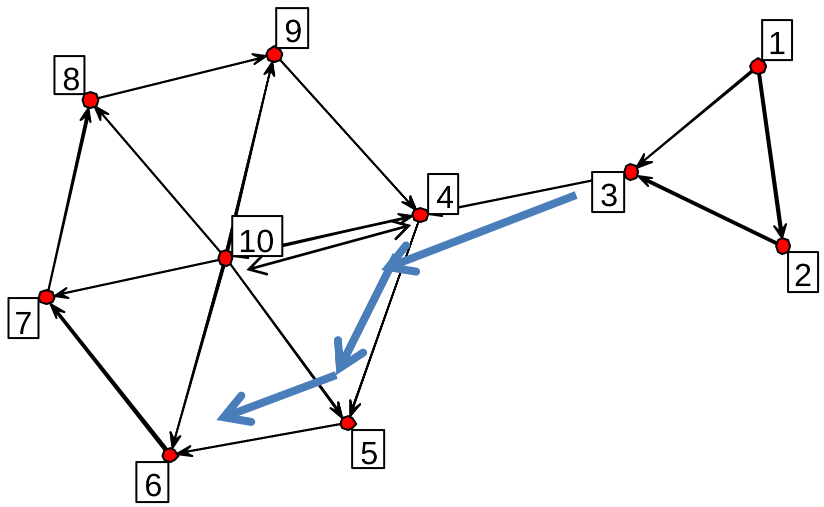
‘sedist’ Example



sedist(g) = sedist(g,mode="graph")

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	0	3	7	5	5	5	5	5	9
[2,]	0	0	1	7	5	5	5	5	5	9
[3,]	3	1	0	6	6	6	6	6	4	8
[4,]	7	7	6	0	4	6	6	6	4	6
[5,]	5	5	6	4	0	2	4	4	4	4
[6,]	5	5	6	6	2	0	2	4	4	6
[7,]	5	5	6	6	4	2	0	2	4	6
[8,]	5	5	6	6	4	4	2	0	2	6
[9,]	5	5	4	4	4	4	4	2	0	6
[10,]	9	9	8	6	4	6	6	6	6	0

'geodist' Example



geodist(g)

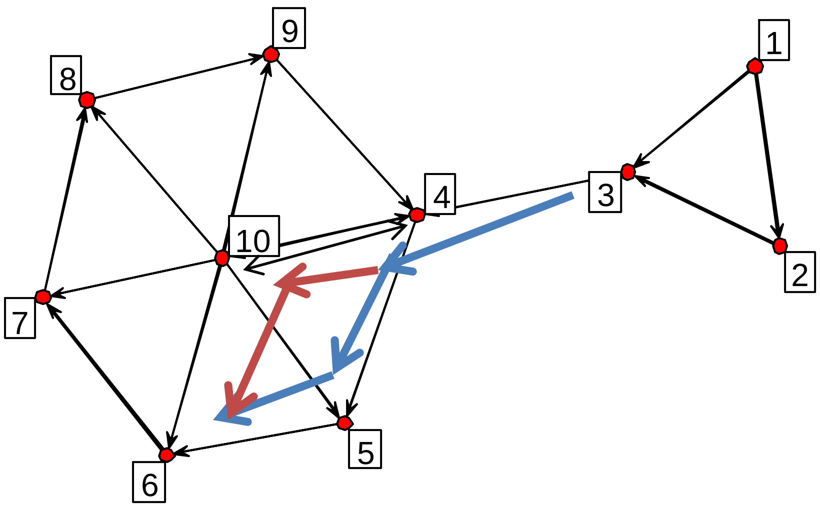
$counts

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	1	1	1	1	2	1	1	1	1
[2,]	0	1	1	1	1	2	1	1	1	1
[3,]	0	0	1	1	1	2	1	1	1	1
[4,]	0	0	0	1	1	2	1	1	1	1
[5,]	0	0	0	1	1	1	1	1	1	1
[6,]	0	0	0	1	1	1	1	1	1	1
[7,]	0	0	0	1	1	2	1	1	1	1
[8,]	0	0	0	1	1	2	1	1	1	1
[9,]	0	0	0	1	1	2	1	1	1	1
[10,]	0	0	0	1	1	1	1	1	1	1

$gdist

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	1	1	2	3	4	4	4	4	3
[2,]	Inf	0	1	2	3	4	4	4	4	3
[3,]	Inf	Inf	0	1	2	3	3	3	3	2
[4,]	Inf	Inf	Inf	0	1	2	2	2	2	1
[5,]	Inf	Inf	Inf	5	0	1	2	3	4	6
[6,]	Inf	Inf	Inf	4	5	0	1	2	3	5
[7,]	Inf	Inf	Inf	3	4	5	0	1	2	4
[8,]	Inf	Inf	Inf	2	3	4	4	0	1	3
[9,]	Inf	Inf	Inf	1	2	3	3	3	0	2
[10,]	Inf	Inf	Inf	1	1	1	1	1	1	0

‘geodist’ Example



geodist(g)

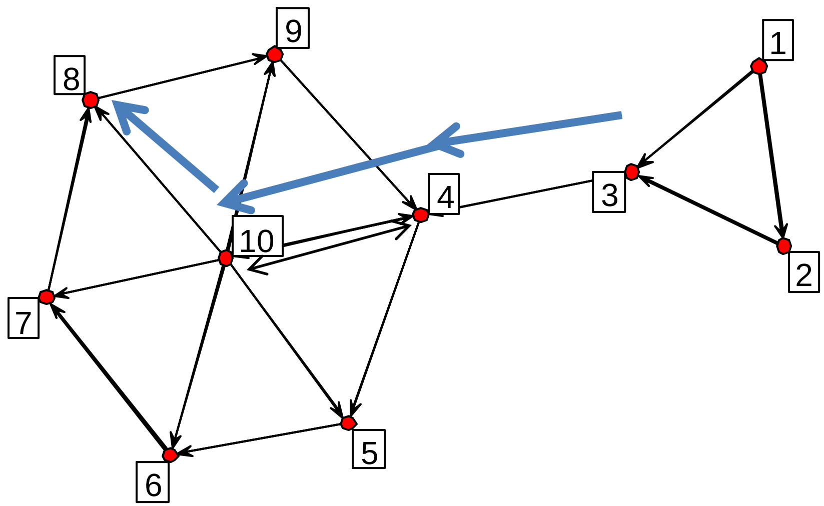
$counts

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	1	1	1	1	2	1	1	1	1
[2,]	0	1	1	1	1	2	1	1	1	1
[3,]	0	0	1	1	1	2	1	1	1	1
[4,]	0	0	0	1	1	2	1	1	1	1
[5,]	0	0	0	1	1	1	1	1	1	1
[6,]	0	0	0	1	1	1	1	1	1	1
[7,]	0	0	0	1	1	2	1	1	1	1
[8,]	0	0	0	1	1	2	1	1	1	1
[9,]	0	0	0	1	1	2	1	1	1	1
[10,]	0	0	0	1	1	1	1	1	1	1

$gdist

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	1	1	2	3	4	4	4	4	3
[2,]	Inf	0	1	2	3	4	4	4	4	3
[3,]	Inf	Inf	0	1	2	3	3	3	3	2
[4,]	Inf	Inf	Inf	0	1	2	2	2	2	1
[5,]	Inf	Inf	Inf	5	0	1	2	3	4	6
[6,]	Inf	Inf	Inf	4	5	0	1	2	3	5
[7,]	Inf	Inf	Inf	3	4	5	0	1	2	4
[8,]	Inf	Inf	Inf	2	3	4	4	0	1	3
[9,]	Inf	Inf	Inf	1	2	3	3	3	0	2
[10,]	Inf	Inf	Inf	1	1	1	1	1	1	0

‘geodist’ Example



geodist(g)

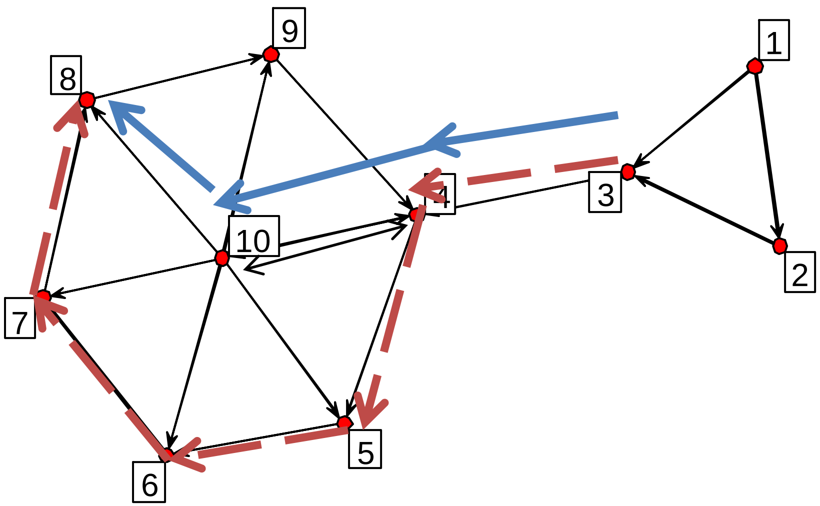
$counts

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]	[, 6]	[, 7]	[, 8]	[, 9]	[, 10]
[1,]	1	1	1	1	1	2	1	1	1	1
[2,]	0	1	1	1	1	2	1	1	1	1
[3,]	0	0	1	1	1	2	1	1	1	1
[4,]	0	0	0	1	1	2	1	1	1	1
[5,]	0	0	0	1	1	1	1	1	1	1
[6,]	0	0	0	1	1	1	1	1	1	1
[7,]	0	0	0	1	1	2	1	1	1	1
[8,]	0	0	0	1	1	2	1	1	1	1
[9,]	0	0	0	1	1	2	1	1	1	1
[10,]	0	0	0	1	1	1	1	1	1	1

$gdist

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]	[, 6]	[, 7]	[, 8]	[, 9]	[, 10]
[1,]	0	1	1	2	3	4	4	4	4	3
[2,]	Inf	0	1	2	3	4	4	4	4	3
[3,]	Inf	Inf	0	1	2	3	3	3	3	2
[4,]	Inf	Inf	Inf	0	1	2	2	2	2	1
[5,]	Inf	Inf	Inf	5	0	1	2	3	4	6
[6,]	Inf	Inf	Inf	4	5	0	1	2	3	5
[7,]	Inf	Inf	Inf	3	4	5	0	1	2	4
[8,]	Inf	Inf	Inf	2	3	4	4	0	1	3
[9,]	Inf	Inf	Inf	1	2	3	3	3	0	2
[10,]	Inf	Inf	Inf	1	1	1	1	1	1	0

‘geodist’ Example



geodist(g)

$counts

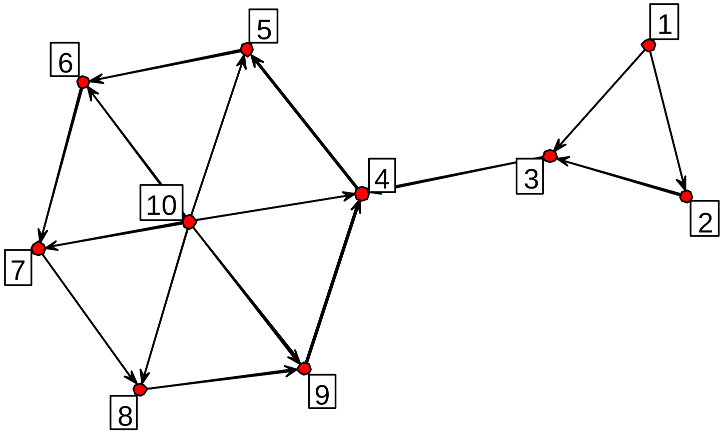
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	1	1	1	1	2	1	1	1	1
[2,]	0	1	1	1	1	2	1	1	1	1
[3,]	0	0	1	1	1	2	1	1	1	1
[4,]	0	0	0	1	1	2	1	1	1	1
[5,]	0	0	0	1	1	1	1	1	1	1
[6,]	0	0	0	1	1	1	1	1	1	1
[7,]	0	0	0	1	1	2	1	1	1	1
[8,]	0	0	0	1	1	2	1	1	1	1
[9,]	0	0	0	1	1	2	1	1	1	1
[10,]	0	0	0	1	1	1	1	1	1	1

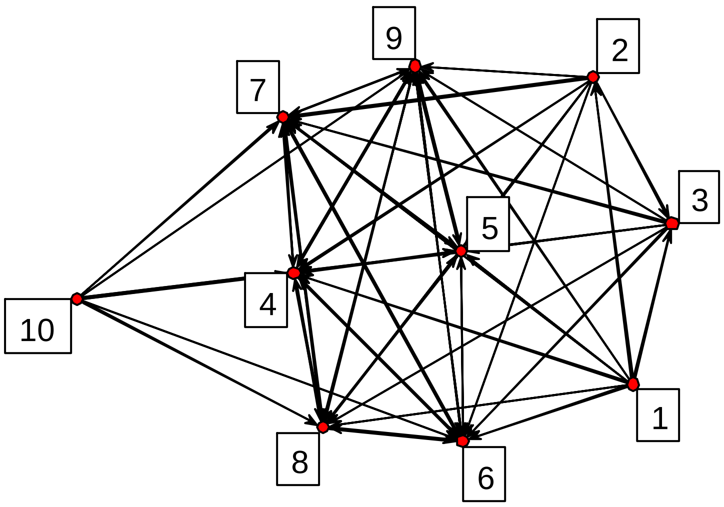
$gdist

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	1	1	2	3	4	4	4	4	3
[2,]	Inf	0	1	2	3	4	4	4	4	3
[3,]	Inf	Inf	0	1	2	3	3	3	3	2
[4,]	Inf	Inf	Inf	0	1	2	2	2	2	1
[5,]	Inf	Inf	Inf	5	0	1	2	3	4	6
[6,]	Inf	Inf	Inf	4	5	0	1	2	3	5
[7,]	Inf	Inf	Inf	3	4	5	0	1	2	4
[8,]	Inf	Inf	Inf	2	3	4	4	0	1	3
[9,]	Inf	Inf	Inf	1	2	3	3	3	0	2
[10,]	Inf	Inf	Inf	1	1	1	1	1	1	0

‘geodist’ reachability

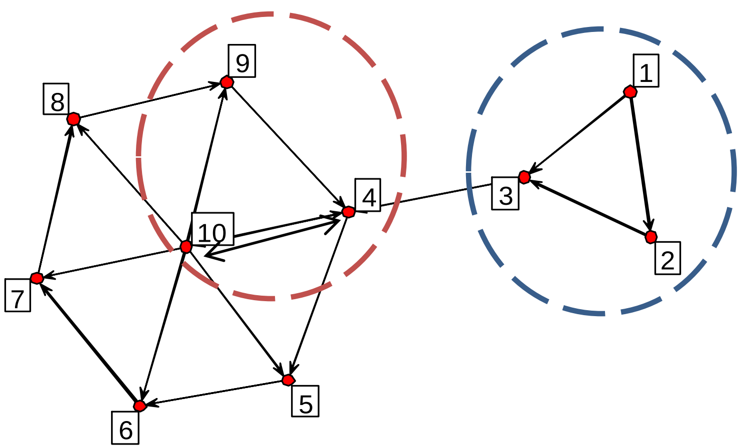
- gplot(reachability(g),label=1:10)

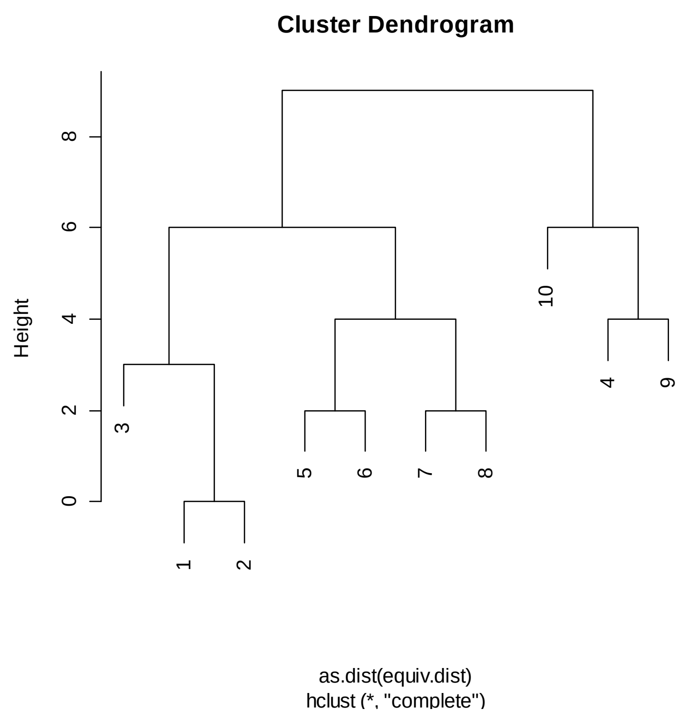




Graph vertices clustering by ‘sedist’

- General clustering methods
- ‘equiv.clust’ for vertices clustering by Structural equivalence(‘sedist’)

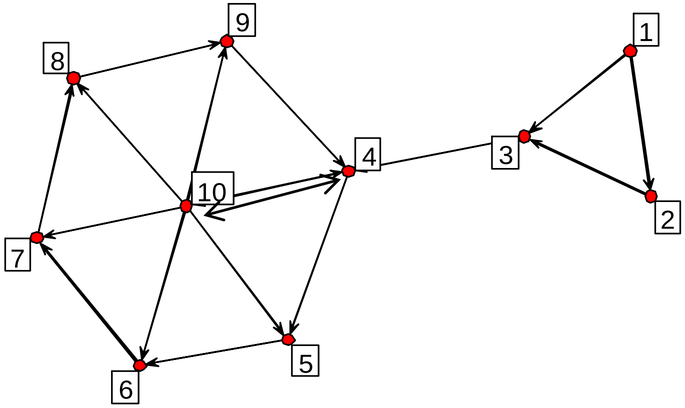


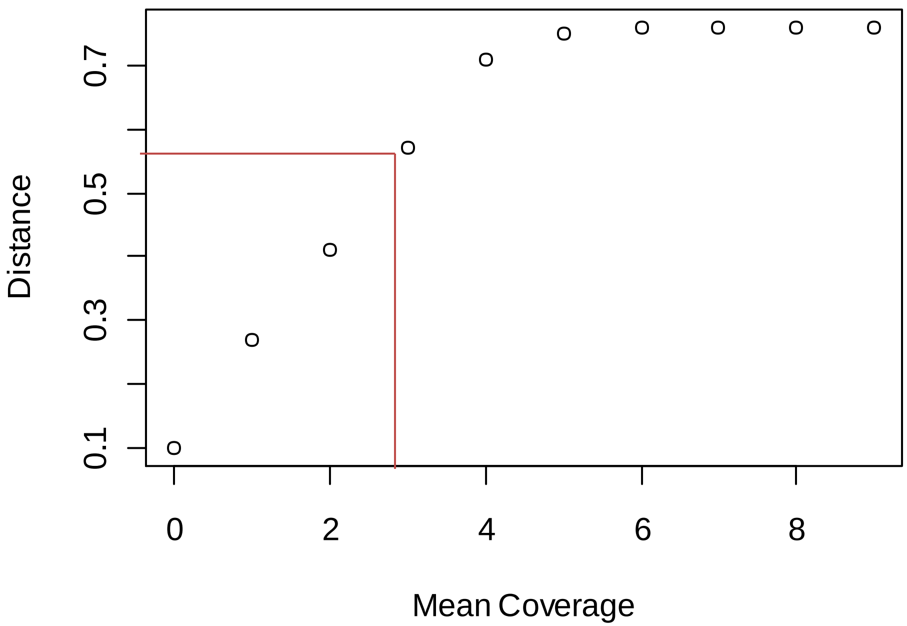


Graph structure by ‘geodist’

- structure.statistics

```
> ss<-structure.statistics(g)
> plot(0:9,ss,xlab="Mean Coverage",ylab="Distance")
```





Graph cov based function

Regression, principal component, canonical correlation

Multi graph measurements

- Graph mean
  - In dichotomous case, graph mean corresponds to graph's density

$$\overline{\delta_H} = \frac{1}{|V_U|^2} \sum_{x=1}^{|V_U|} \sum_{y=1}^{|V_U|} \delta_H(x,y)$$

- Graph covariance
  - gcov/gscov

$$Cov(H_i, H_j) = \frac{1}{|V_U|^2} \sum_{x=1}^{|V_U|} \sum_{y=1}^{|V_U|} \left(\left(\delta_i(x,y) - \overline{\delta_{H_i}} \right) \left(\delta_j(x,y) - \overline{\delta_{H_j}} \right) \right)$$

- Graph correlatıon
  - gcor/gscor

$$\rho(H_i, H_j) = \frac{Cov(H_i, H_j)}{\sqrt{Var(H_i) Var(H_j)}}$$

- Structural covariance
  - unlabeled graph

$$Cov_S(G_i, G_j | \mathcal{P}_i, \mathcal{P}_j) = \max_{L_a \in \mathcal{P}_i, L_b \in \mathcal{P}_j} Cov(L_a(G_i), L_b(G_j))$$

Butts, C.T., and Carley, K.M. (2001). “Multivariate Methods for Interstructural Analysis.”

Correlation statistic model

- Canonical correlation
  - netcancor
- Linear regression
  - netlm
- Logistic regression
  - netlogit
- Linear autocorrelation model
  - lnam
  - nacf

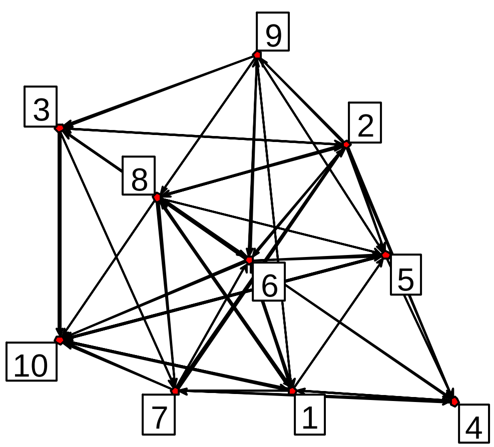
Random graph models

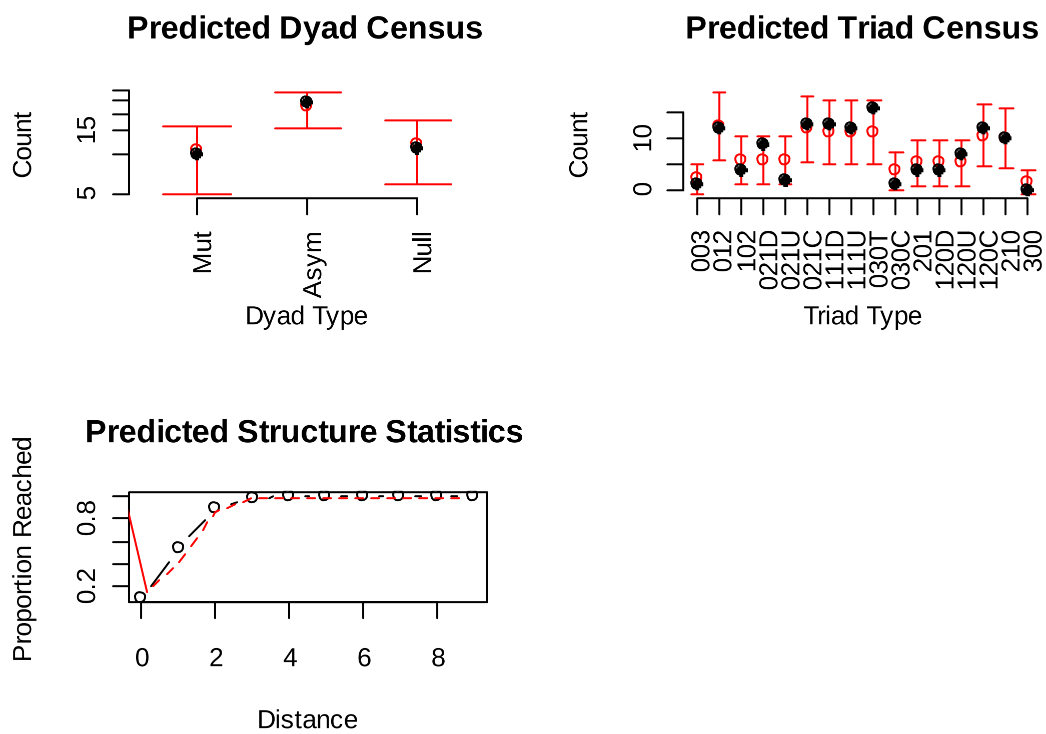
Graph evolution

- Random
- Biased
- 4 Phases

Biased net model

- graph generate: rgbn
- graph prediction: bn

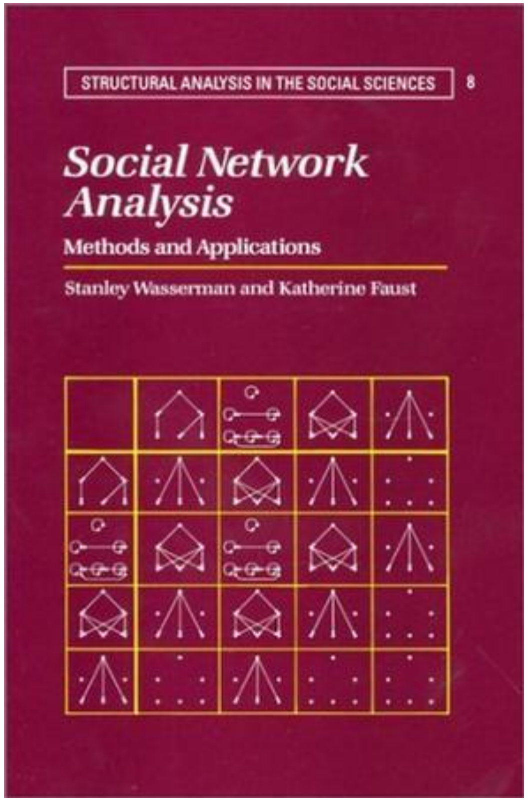




Graph statistic test

- cugtest
- qaptest

Thanks


STRUCTURAL ANALYSIS IN THE SOCIAL SCIENCES 8
Social Network Analysis
Methods and Applications
Stanley Wasserman and Katherine Faust

+