Networks

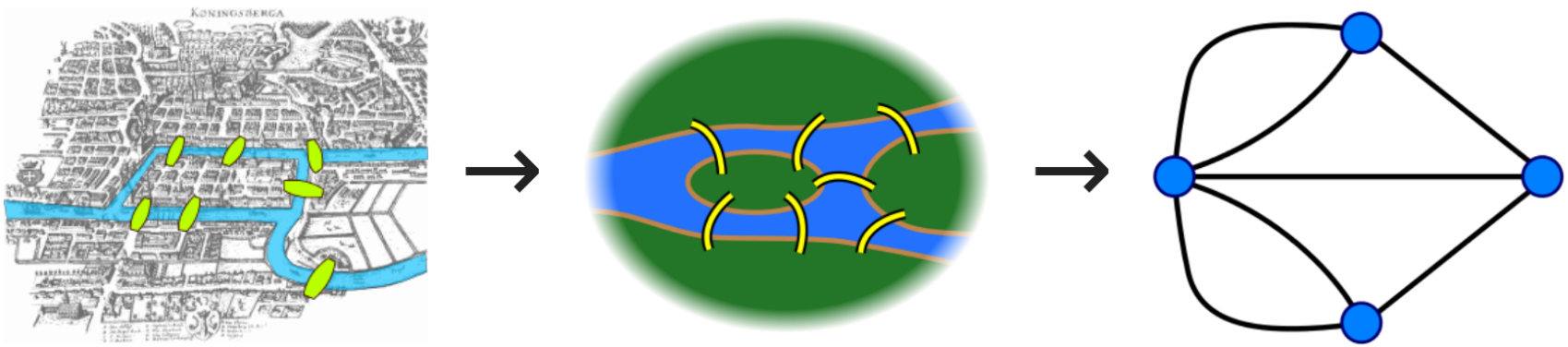
Arthur Charpentier

UQAM

Actuarial Summer School 2019

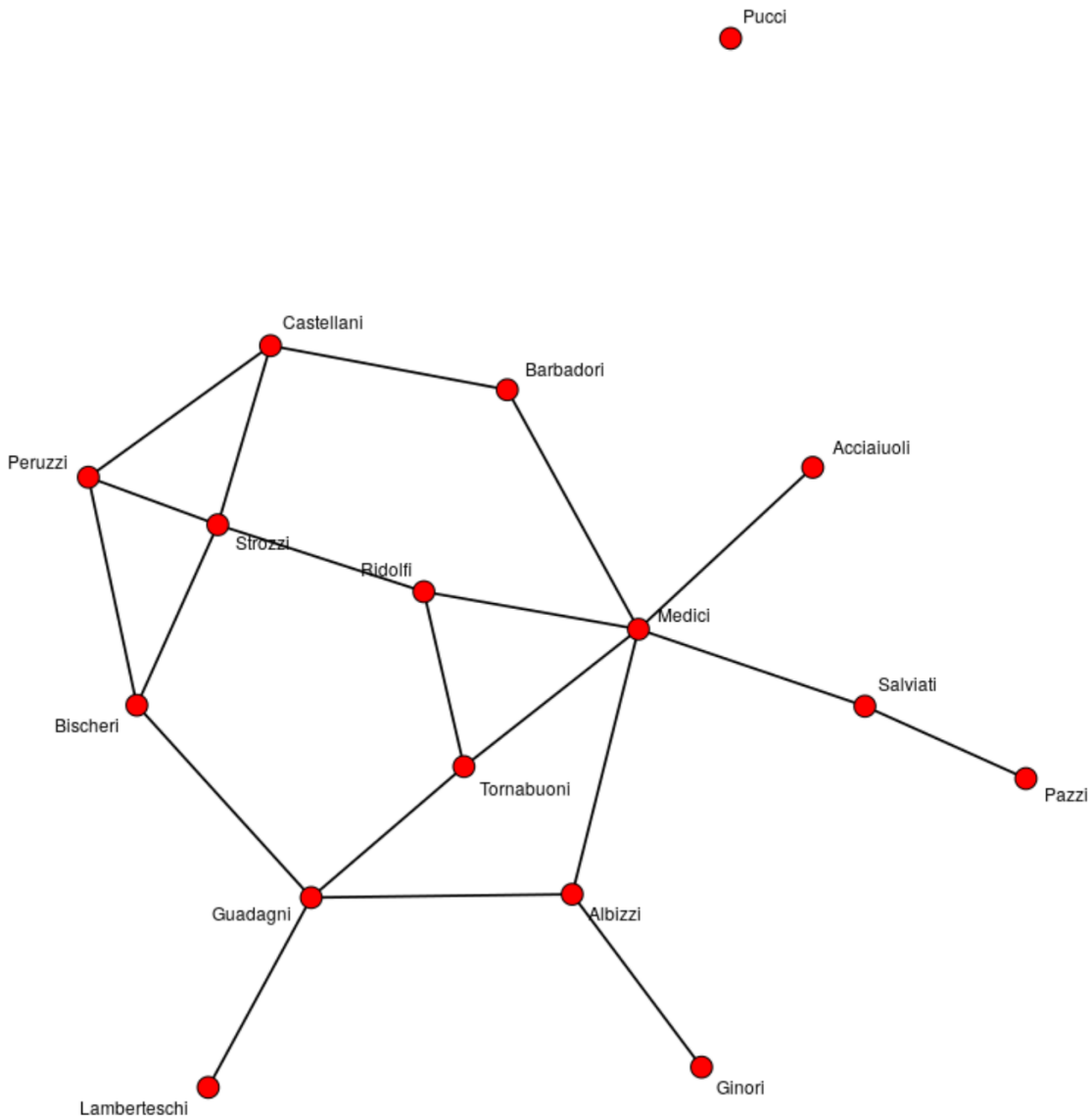
Königsberg's Seven Bridges

The path should cross over each of the seven bridges exactly once.



see also use of GPS to find the shortest path home

Florentine Marriages (1430's)



References

Briatte's Awesome Network Analysis, Adamic (2010, lecture notes), Easley & Kleinberg (2010, Networks, Crowds, and Markets: Reasoning About a Highly Connected World), Harary (1969, Graph Theory), Jackson (2008, Social and Economic Networks), Kolaczyk (2009, Statistical Analysis of Network Data: Methods and Models), Kolaczyk & Csárdi (2010, Statistical Analysis of Network Data with R) Newman (2004, The structure and function of complex networks), Newman (2010, Networks : An Introduction), Wasserman & Faust (1994) Social Network Analysis: Methods and Applications), West (1996). Introduction to Graph Theory

See also R packages, sna, network, igraph to create and manipulate networks, and ggnet2 or networkD3.

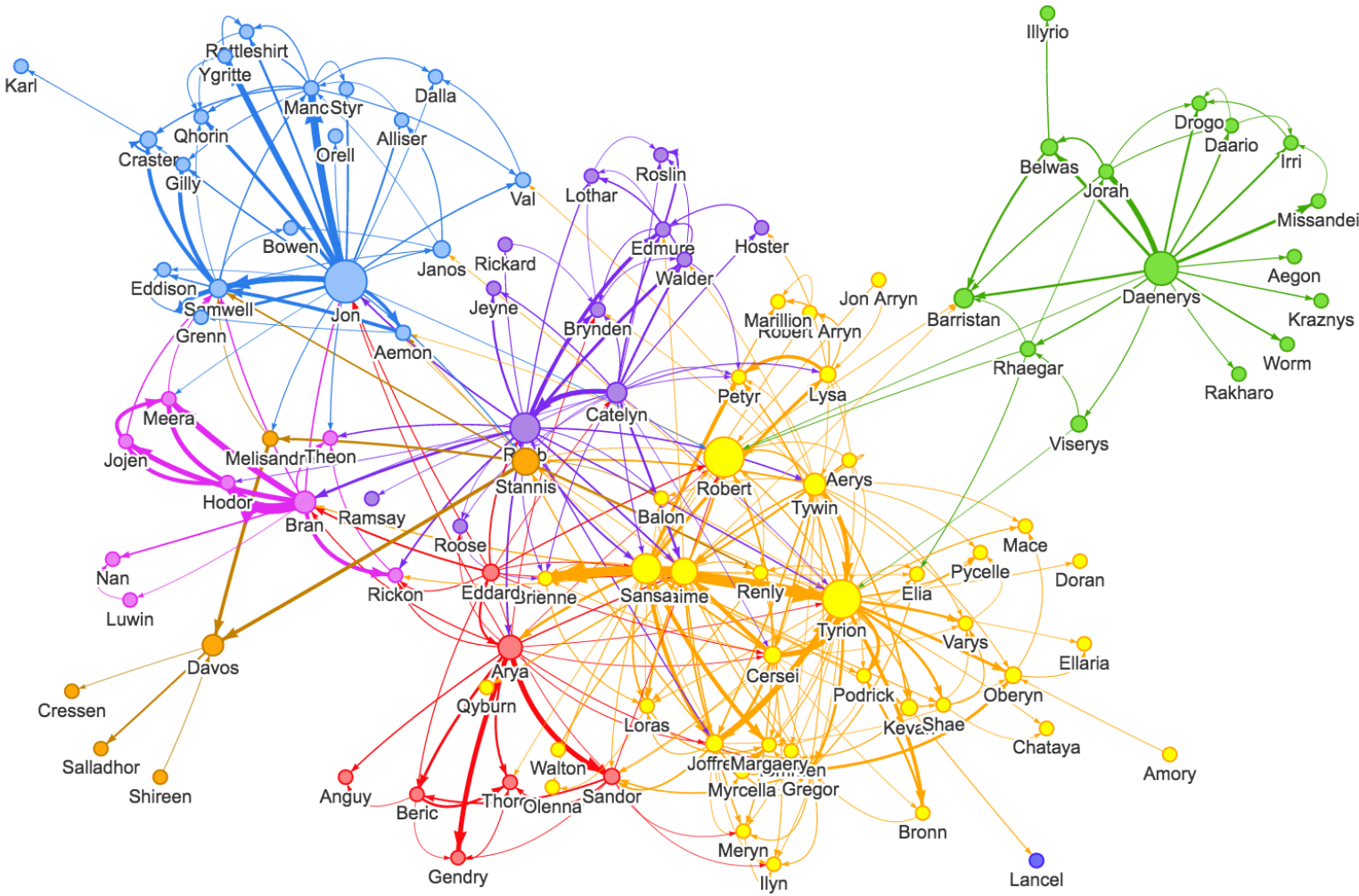
Network Representation

Let $V = \{1, \cdots, n_V\}$ denote either nodes, or vertices (n_V is the order). Let $E \in \{0,1\}^{n_V \times n_V}$ represents the relationships, through an adjacency matrix $\boldsymbol{A}$, $A_{i,j} = 1$ indicates a link - or edge - between i and j, or a collection of links $\{e_1, \cdots, e_{n_E}\}$. Let $n_E = |E|$ denote the number of edges, called size.
The degree $d(\cdot)$ of a vertice v is its number of incident edges.
A network is a pair $G = (V, E)$

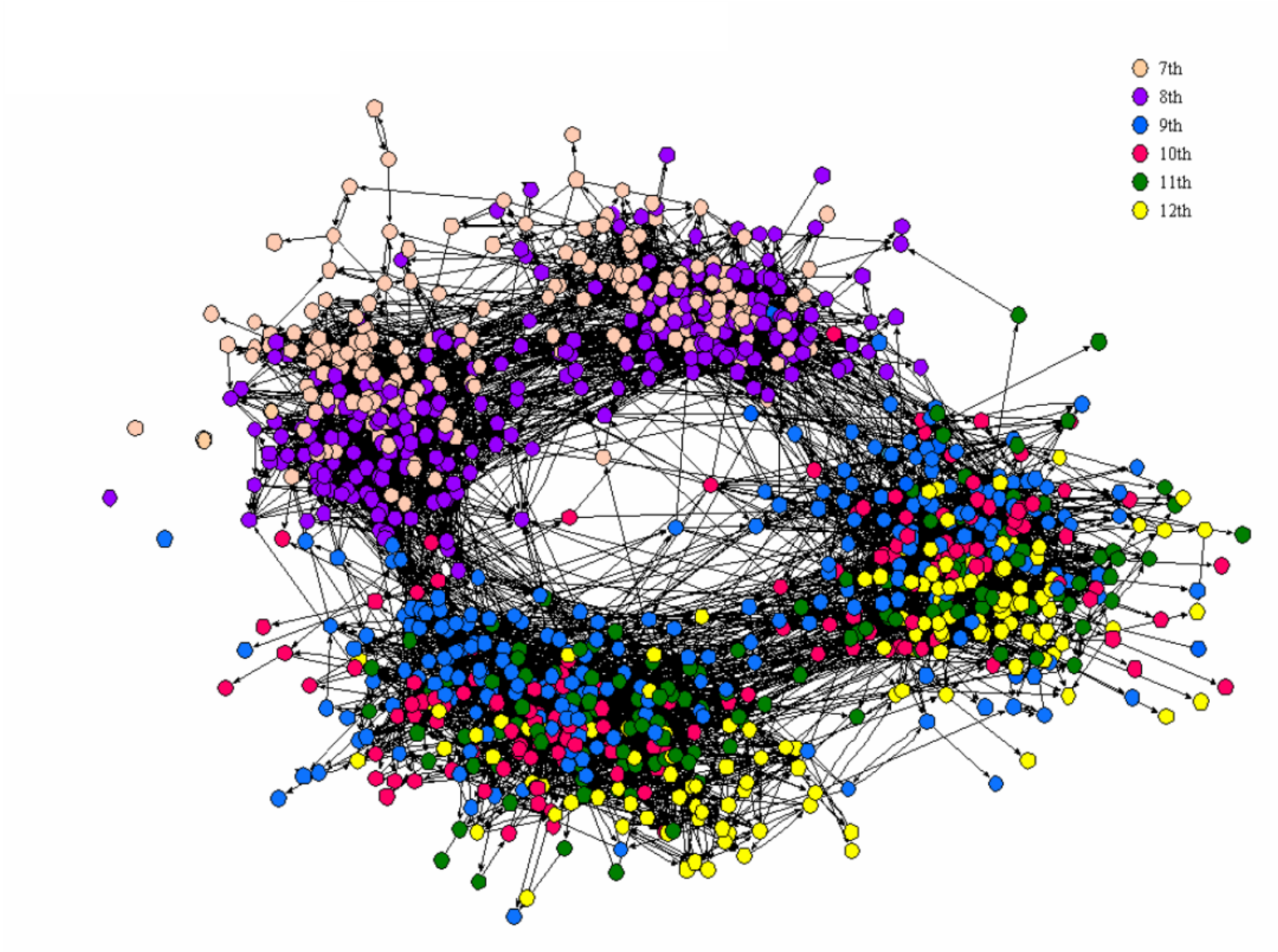
$G =$ Internet, $V =$ computers, $E =$ IP network adjacency
$G =$ World Wide Web, $V =$ web pages, $E =$ hyperlink
$G =$ Articles, $V =$ authors, $E =$ citations
$G =$ Friendship Network, $V =$ persons, $E =$ friendship
$G =$ Airport Network, $V =$ airports, $E =$ non-stop flight

Worldwide Airports (2017)

‘Friendship’ Network (2015)

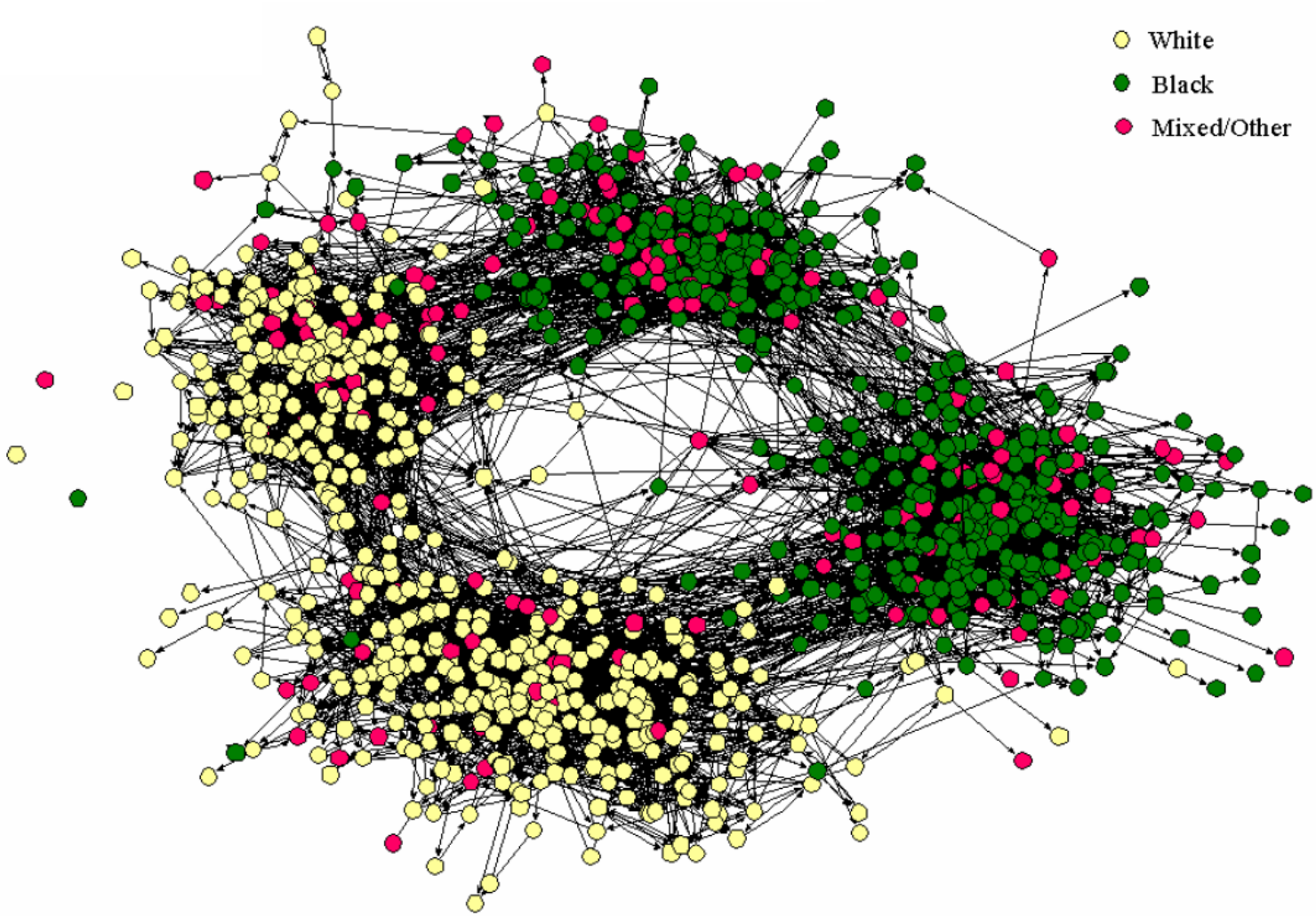


High School Friendship (2001)



from Moody (2001) Race, School Integration and Friendship

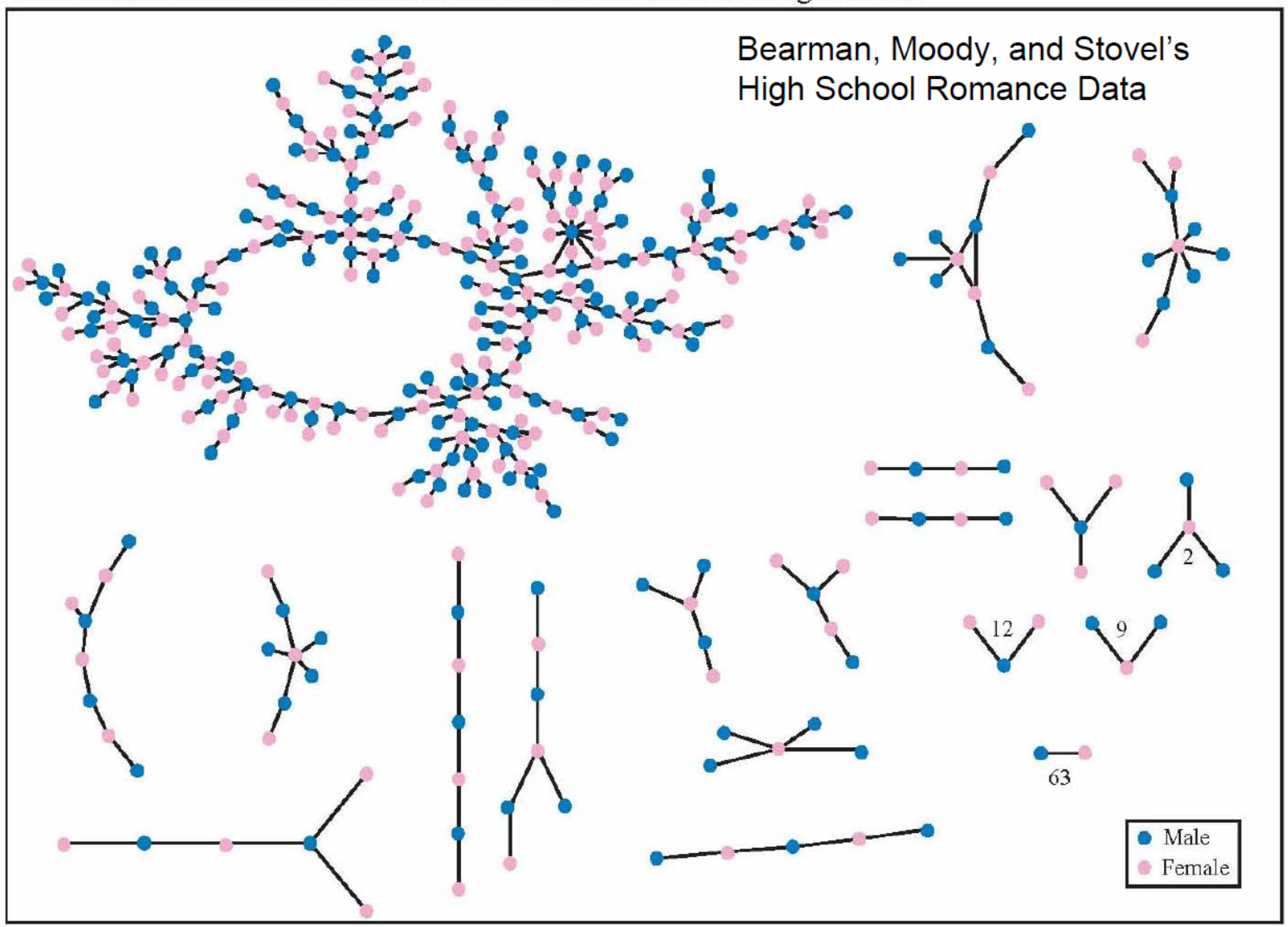
High School Friendship (2001)



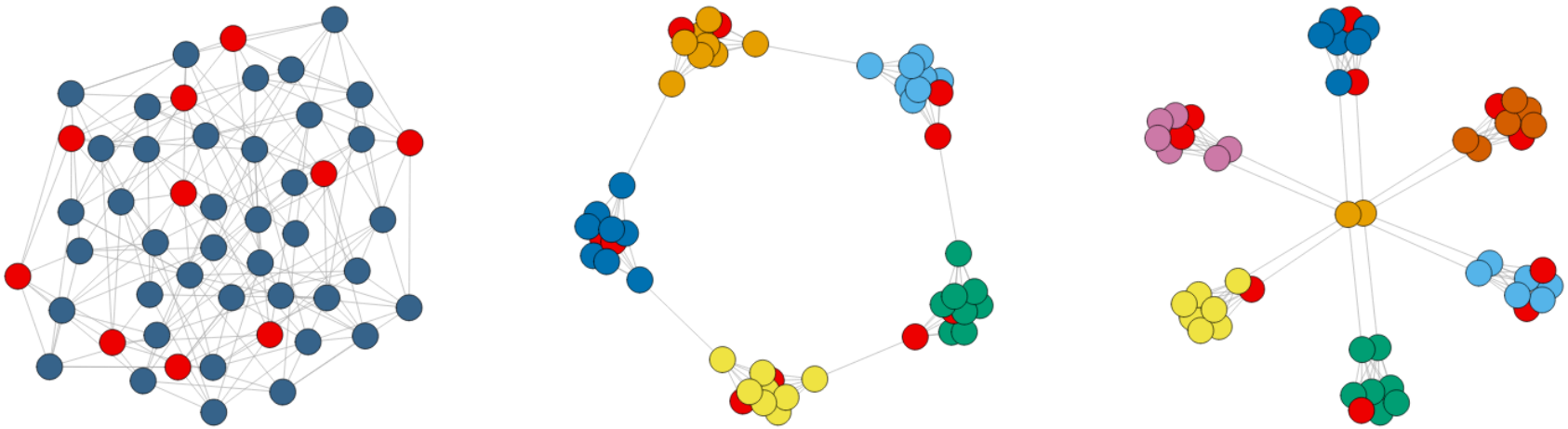
from Moody (2001) Race, School Integration and Friendship

Romantic and Sexual Relationships

The Structure of Romantic and Sexual Relations at "Jefferson High School"



Cyberisk in Network Models

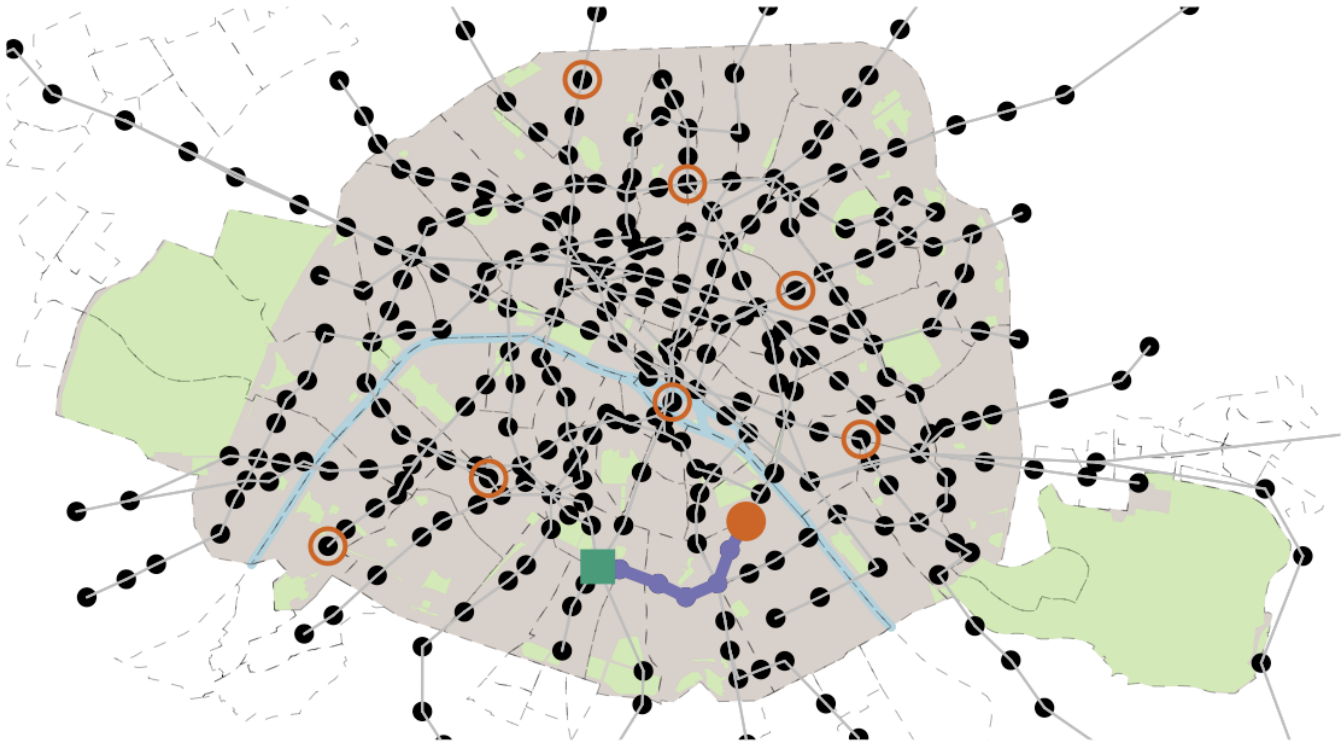


(A) **Homogeneous** (B) **Clustered** (C) **Star**

FIGURE 6. Infection scenario: The red nodes (3,5,13,15,23,25,33,35,43,45) are initially infected.

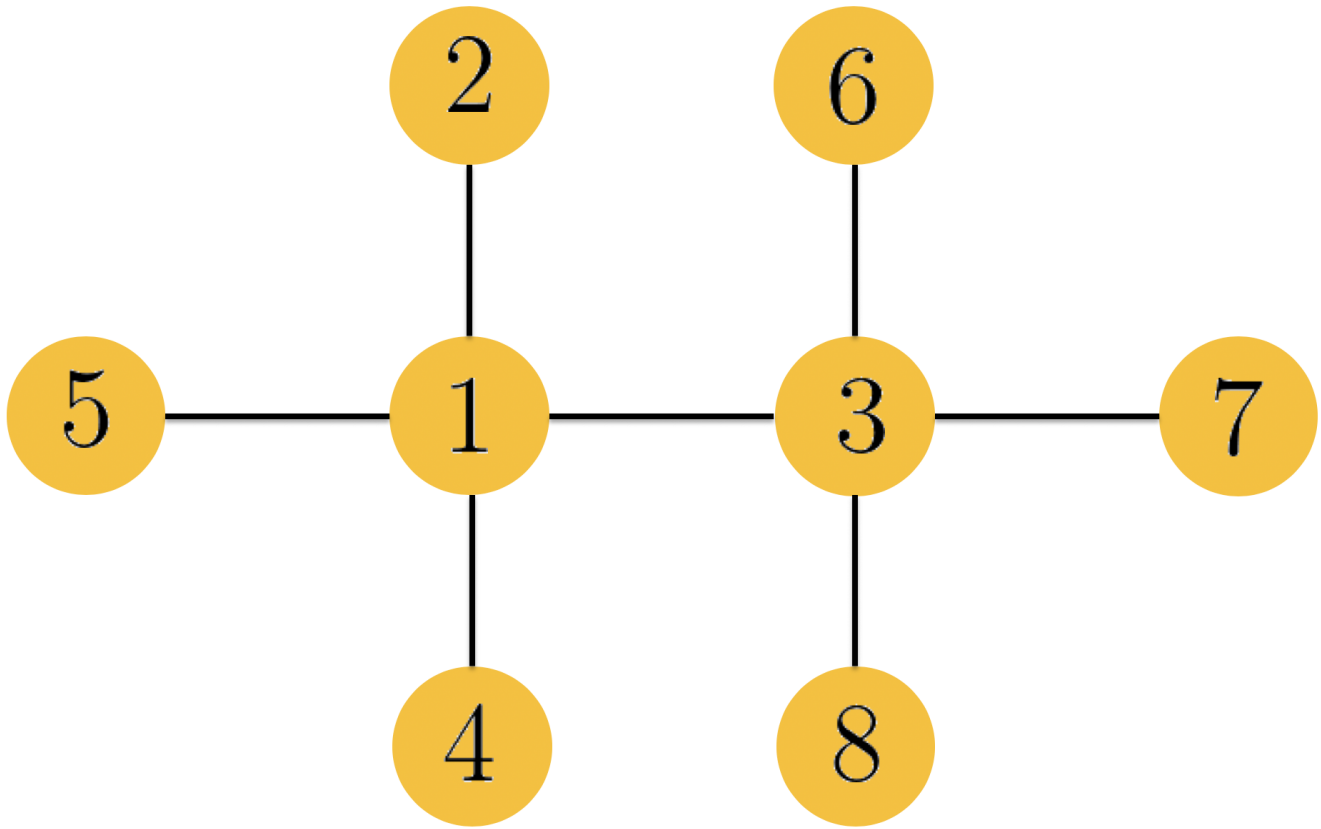
Fahrenwaldt *et al.* (2017, Pricing of cyber insurance contracts)

Shortest Path

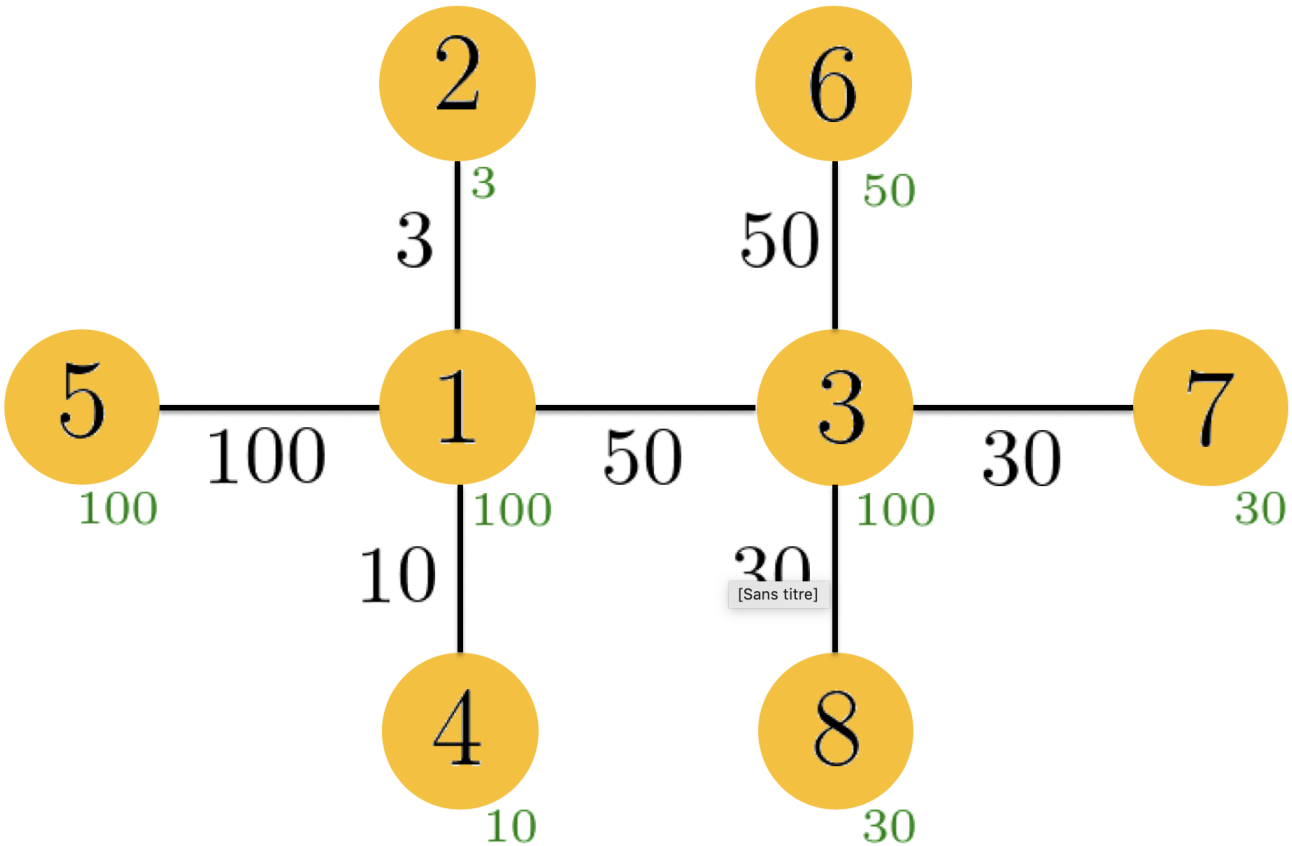


Charpentier *et al.* (2019, Optimal transport on large networks: a practitioner guide)

Networks & Insurance (P2P)



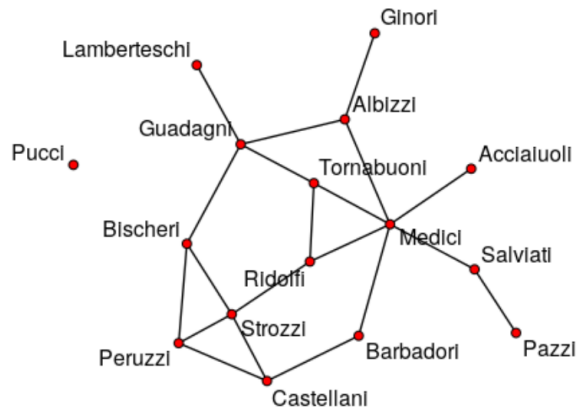
Networks & Insurance (P2P)



used by Inspeer, see https://wwwthe-digital-insurer.com

Networks with R

```
> library(network)
> data(flo)
> nflo<-network(flo,directed=FALSE)
> nflo
 Network attributes:
  vertices = 16
  directed = FALSE
  total edges= 20
> plot(nflo,displaylabels=TRUE,
    boxed.labels=FALSE)
```



Florentine Wedding Data. $n_V = 16$ vertices (families), $n_E = 20$ edges (weddings), from Padgett (1994, Marriage and Elite Structure in Renaissance Florence, 1282-1500)

Network Representation

Let $V = \{1, \cdots, n_V\}$ denote either nodes, or vertices (n_v is the order). Let $E \in \{0,1\}^{n_V \times n_V}$ represents the relationships, through an adjacency matrix $\boldsymbol{A}$, $A_{i,j} = 1$ indicates a link - or edge - between i and j, or a collection of links $\{e_1, \cdots, e_{n_E}\}$. Let $n_E = |E|$ denote the number of edges, called size.
The degree $d(\cdot)$ of a vertice v is its number of incident edges.
A network (or a graph) is a pair $G = (V, E)$
If $V_1 \subset V_2$ and $E_1 \subset E_2$, then (V_1, E_1) is a subgraph of (V_2, E_2).
Two vertices $u, v \in V$ are said to be adjacent (or connected) if they are joined by an edge in E.

Network Representation

One might consider some undirected network: consider e.g. adjacency matrix $\boldsymbol{A}$

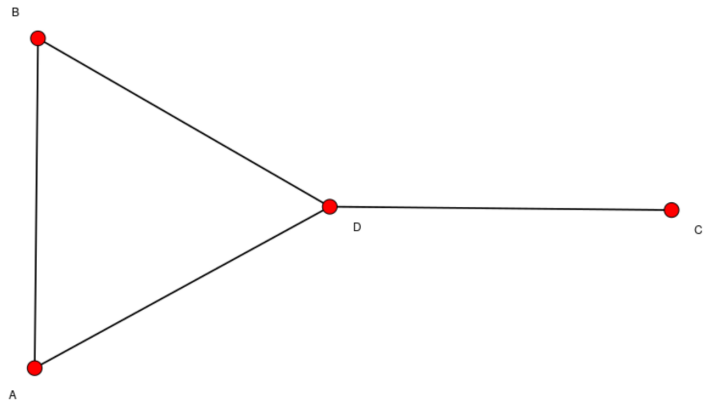
$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

with $n_V = 4$ vertices, and $n_E = 4$ edges (4 1's in the upper corner of the matrix).

There are no self-loops, i.e. $A_{i,i} = 0$.

Network Representation

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



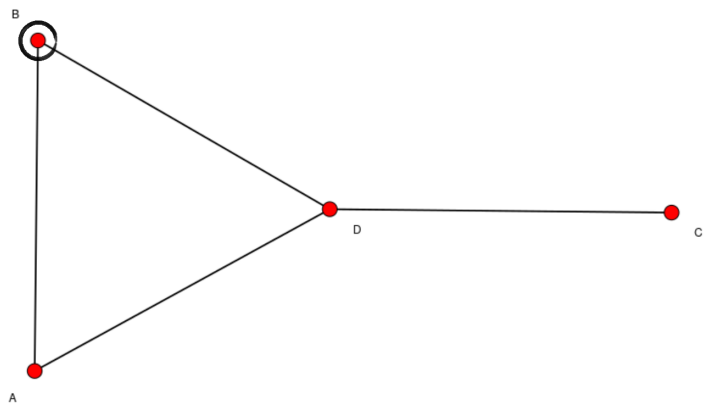
$A_{ij} = 1$ if and only if i and j are linked. The matrix is symmetric ($A_{ij} = A_{ji}$), the network is undirected.
Links are $E = \{1-2, 1-4, 2-4, 3-4\}$ - no need to mention $\{2-1, 4-1, 4-2, 4-3\}$ since undirected. Hence, $n_E = 4$.
Further, $d(1) = 2$, $d(2) = 2$, $d(3) = 1$ and $d(4) = 3$.

Network Representation

Row i contains list of vertices connected to vertice i.

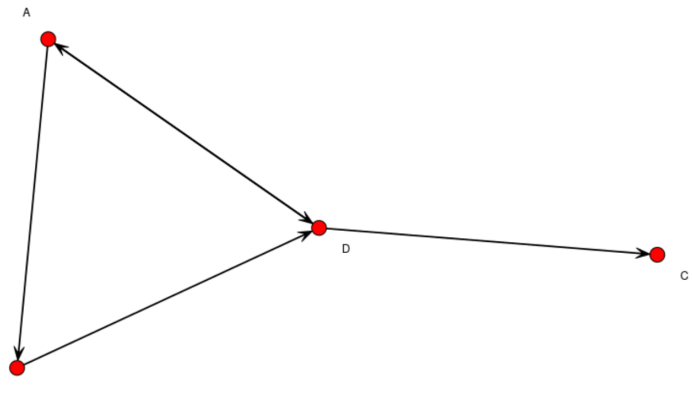
$$d(i) = \sum_{j=1}^{n_V} A_{i,j} = \boldsymbol{A}_{i,\cdot}^{\mathsf{T}} \mathbf{1}.$$

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



Network Representation

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

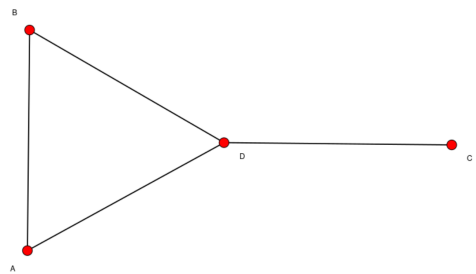


Here the network is directed (also called digraph).
Links are $E = \{(1,2),(1,4),(2,4),(4,1),(4,3)\}$ (the 5 arrows).
or $E = \{(1 \to 2),(1 \leftrightarrow 4),(2 \to 4),(3 \leftarrow 4)\}$
For edge $(1,2)$, 1 is the source node and 2 is the terminal node.
Vertices have in-degrees and out-degrees.

Network Representation

For the undirected graph, one can also consider the $n_V \times n_E$ incidence matrice $\boldsymbol{T}$

$$\boldsymbol{T} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} (ab) & (ad) & (bd) & (cd) \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$



Observe that $d(v) = \sum_{e=1}^{n_E} T_{v,e}$, i.e. $\boldsymbol{d} = \boldsymbol{T}\boldsymbol{1}$.

Further $\boldsymbol{T}\boldsymbol{T}^\top = \boldsymbol{D} + \boldsymbol{A}$ where $\boldsymbol{D} = \text{diag}[\boldsymbol{d}]$

Network Representation

Finally, the $n_V \times n_V$ matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is called the Laplacian of the graph. One can prove that for any $\boldsymbol{x} \in \mathbb{R}^{nv}$,

$$\boldsymbol{x}^{\mathsf{T}} \boldsymbol{L} \boldsymbol{x} = \sum_{(i,j) \in E} (x_i - x_j)^2$$

The normalized Laplacian is $\widetilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2}$. Note that eigenvalues of $\widetilde{\boldsymbol{L}}$ lie in interval $[0, 2]$.

Network Representation

The goal is to embed a combinatorial object, our network, (V, E) into a two-dimensional Euclidean space.
Clearly not unique...
The visualization of network $G = (V, E)$ is a mapping $\varphi : (V, E) \to \mathbb{R}^2$. No geometry in G
Multidimensional scaling (MDS) is commonly used for visualization. Given vertex distances $\boldsymbol{D} = [D_{i,j}]$ (shortest parth, e.g.), we want to find

$$\vec{\boldsymbol{z}}_i = (x_i, y_i), \text{ for } i \in V, \text{ so that } \|\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_j\| \sim D_{i,j}$$

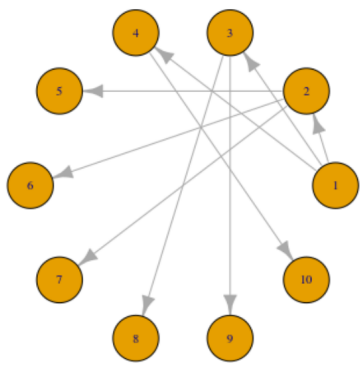
Consider

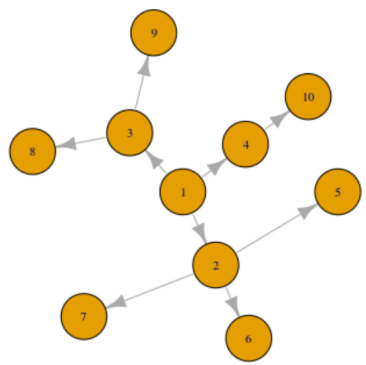
$$\min_{\vec{\boldsymbol{z}}_1, \cdots, \vec{\boldsymbol{z}}_{n_V}} \left\{ \left(D_{i,j} - \|\boldsymbol{z}_i - \boldsymbol{z}_j\| \right)^2 \right\}$$

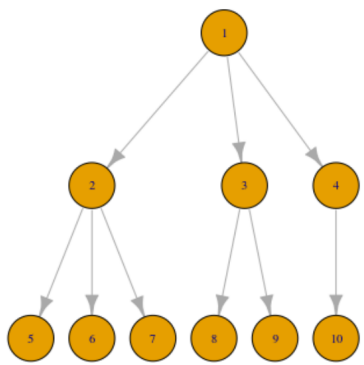
Possible to add some constraints, e.g. centralities

Network Representation

- convention : straight line segments
- aesthetics : minimal edge crossing
- aesthetics : relative placement of vertices, subgraphs, etc.

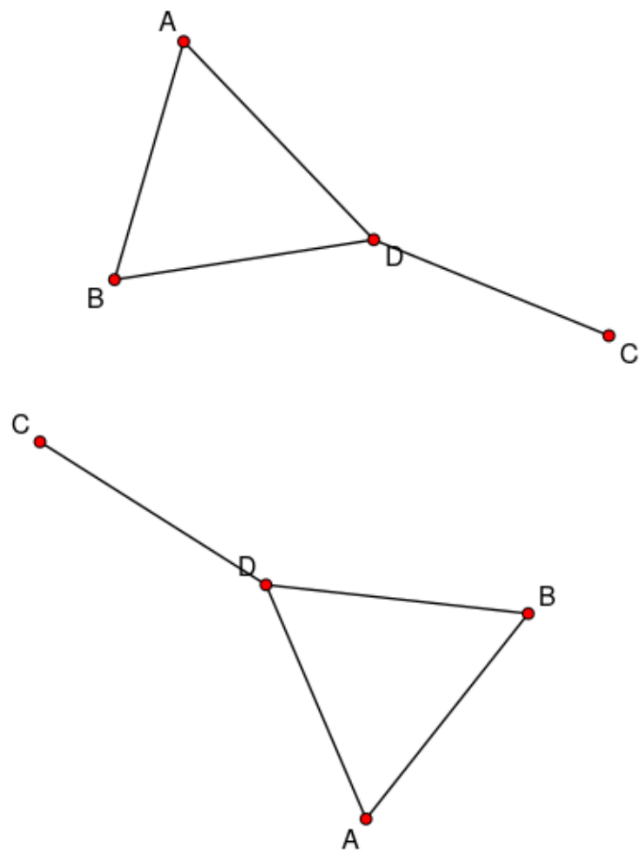






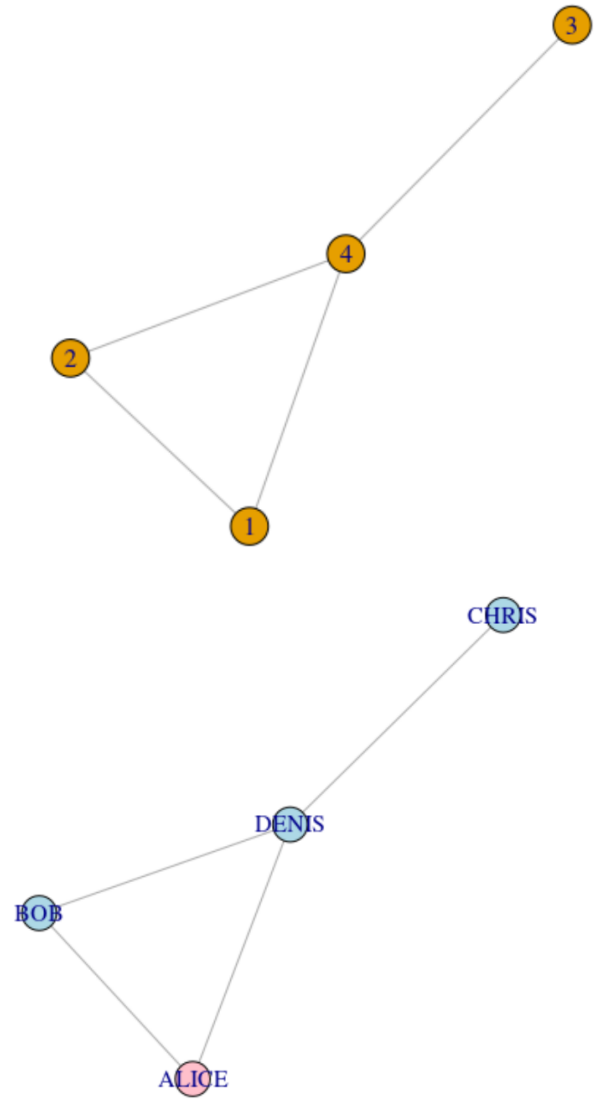
Networks with R

```
> library(network)
> m = matrix(0,4,4)
> colnames(m)=rownames(m)=LETTERS
    [1:4]
> m[1,2]=m[1,4]=m[2,1]=m[2,4]=m
    [3,4]=m[4,1]=m[4,2]=m[4,3]=1
> m
  A B C D
A 0 1 0 1
B 1 0 0 1
C 0 0 0 1
D 1 1 1 0
> nm=network(m,directed=FALSE)
> plot(nm,displaylabels=TRUE,boxed.
    labels=FALSE)
```



Networks with R

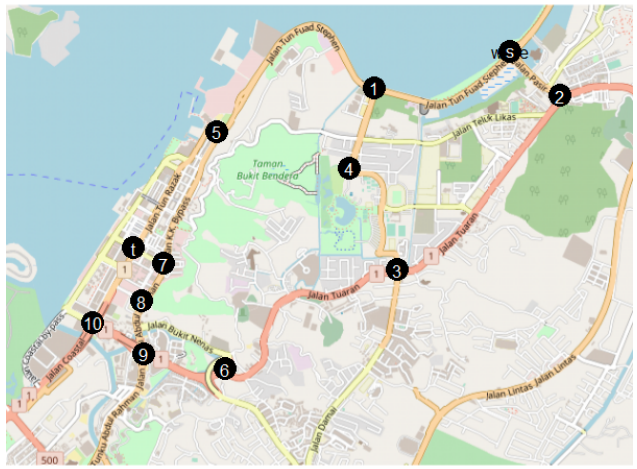
```
> library(igraph)
> g <- graph.formula
    (1-2,1-4,2-4,3-4)
> plot(g)
> V(g)
+ 4/4 vertices, named, from 3b97949
    :
[1] 1 2 4 3
> V(g)$label=c("ALICE","BOB","DENIS
    ","CHRIS")
> V(g)$color=c("pink",rep("light
    blue",3))
> E(g)
+ 4/4 edges from 3b97949 (vertex
    names):
[1] 1--2 1--4 2--4 4--3
> plot(g)
```



Spatial Data & Graphs

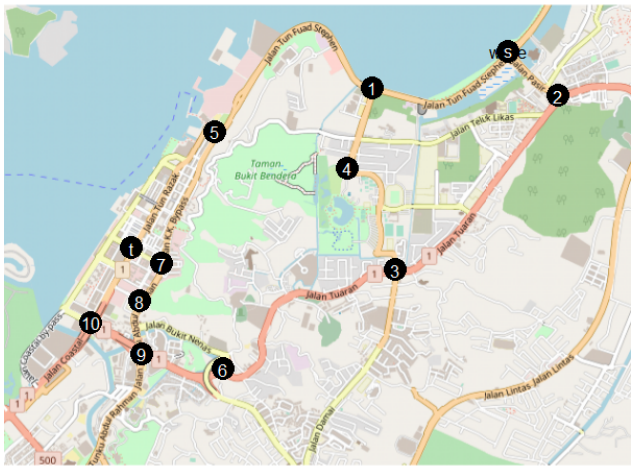
From Abdullah & Kien Hua (2017, The Application of the Shortest Path and Maximum Flow with Bottleneck in Traffic Flow)

```
location <- 'http://www.jistm.com/
    PDF/JISTM-2017-04-06-02.pdf'
m <- matrix(c(0, 5.995910,
    116.105520, 1, 5.992737,
    116.093718, 2, 5.992066,
    116.109883, 3, 5.976947,
    116.095760, 4, 5.985766,
    116.091580, 5, 5.988940,
    116.080112, 6, 5.968318,
    116.080764, 7, 5.977454,
    116.075460, 8, 5.974226,
    116.073604, 9, 5.969651,
    116.073753, 10, 5.972341,
    116.069270, 11, 5.978818,
    116.072880), 3, 12)
```



Spatial Data & Graphs

```
library(OpenStreetMap)
map <- openproj(openmap(c(lat=
    6.000, lon= 116.06), c(lat=
    5.960, lon= 116.12)))
plot(map)
points(t(m[3:2,]),col="black", pch
    =19, cex=3)
```

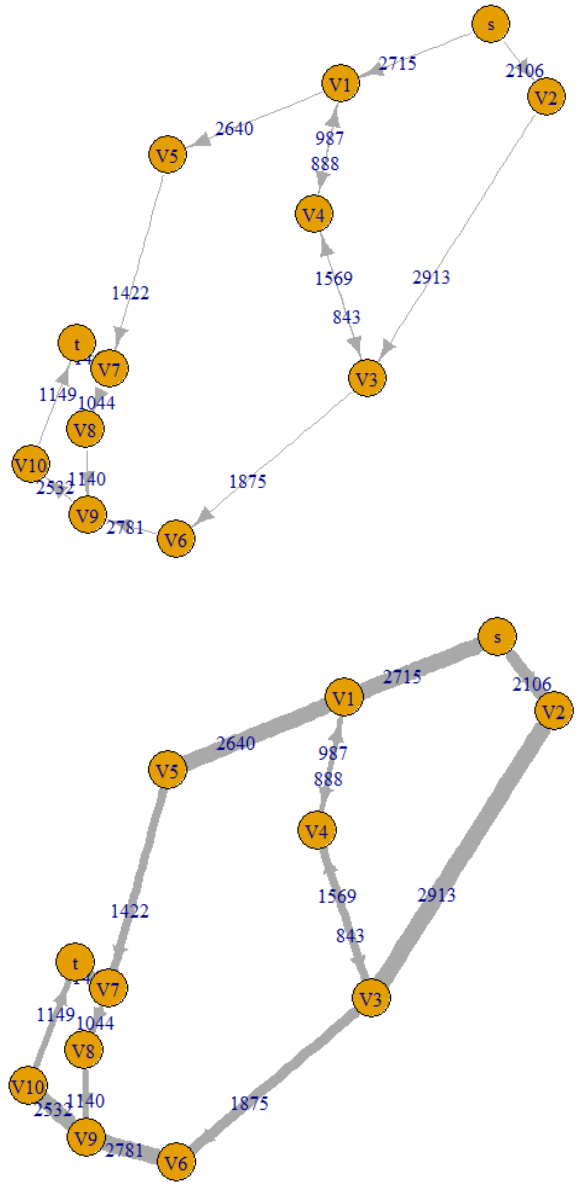



```
library(tabulizer)
out <- extract_tables(location)
B1 = as.data.frame(out[[2]])
B2 = as.data.frame(out[[3]])
E = data.frame(from=B1[3:20,"V3"],
    to=B1[3:20,"V4"])
E = E[-c(6,8),]
capacity = as.character(B2$V3[-1])
E$capacity = as.numeric(capacity)
```

Spatial Data & Graphs

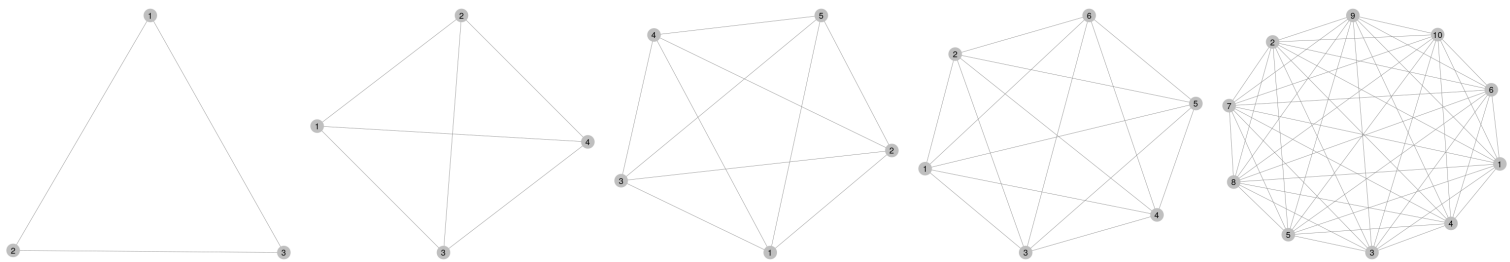
We can plot only the netgwork (without background map)

```
1 library(igraph)
2 g=graph_from_data_frame(E)
3 E(g)$label=E$capacity
4 plot(g, layout=as.matrix(B[,c("x","
      y")]))
5 plot(g, layout=as.matrix(B[,c("x","
      y")]), edge.width=E$capacity
      /200)
```

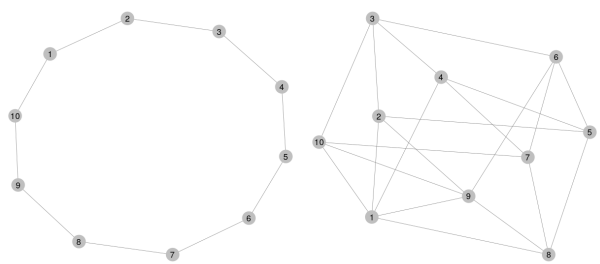


Special Graphs

A complete graph of size n has n vertices and $\frac{n(n-1)}{2}$ edges.
$A_{i,j} = \mathbf{1}_{i \neq j}$.



A regular graph is a graph in which every vertice has the same degree.



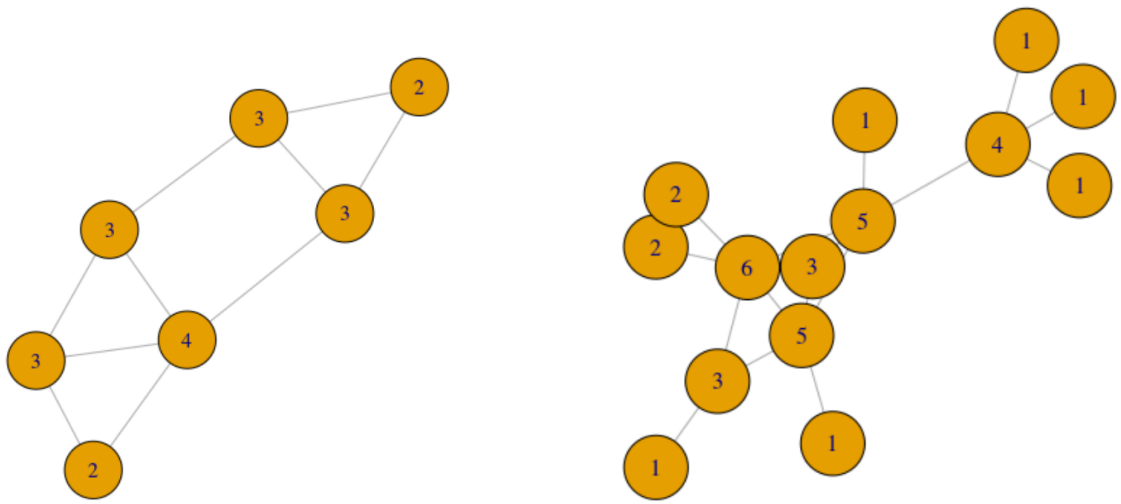
Random Graphs

The degree of a vertice v is $d(v)$, the number of vertices in V incident to v (i.e. the number of neighbors of v)
A network is said to be sparse if

$$n_E \ll \frac{n_V(n_V-1)}{2} \text{ or } \overline{d} = \frac{1}{n_V}\sum_{v=1}^{n_V} d_v \ll n_V - 1$$

Let f_δ denote the proportion of vertices with degree δ, and $\boldsymbol{f} = (f_\delta)$ the degree distribution, which is a summary of local connectivity across the graph.

Random Graphs

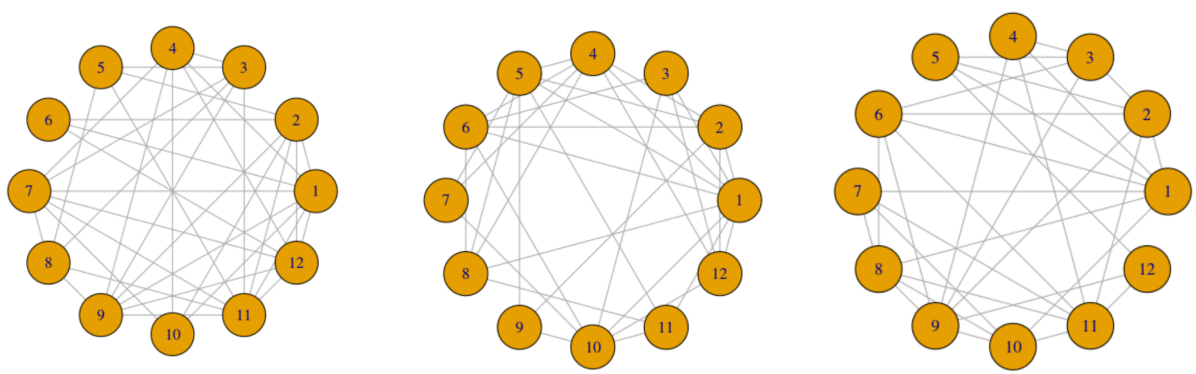


High-degree vertices are likely to be influential, central, prominent.

Networks with R

The Erdös-Renyi random graph model $G_{n,p}$ is an undirected graph with n vertices, such that edge (v, v') is present with probability p, independent of other edges.

```
library(igraph)
E = erdos.renyi.game(n=12, p.or.m=.5, type="gnp")
plot(E, layout=layout.circle)
# hist(igraph::degree(E))
```



Then $d(v)$ is a binomial distribution $\mathcal{B}(n_V - 1, p)$, i.e.

$$f_\delta = \mathbb{P}[d(v) = \delta] = \binom{n-1}{\delta} p^\delta (1-p)^{n_V - 1 - \delta}$$

For a large network ($n_V \to \infty$), $d(v) \sim \mathcal{N}(n_V p, n_V p(1-p))$ from the law of large numbers.
For a large netword with $p \sim \lambda / n_V$, $d(v) \sim \mathcal{P}(\lambda)$from the law of small numbers.

Random Graphs: Power Laws

Scale-free network have degree distribution with power-law tail
A scale free function $f(\cdot)$ satisfies $f(ax) = bf(x)$, $\forall x$, for some a, b
Ex Power-law functions $f(x) = x^{-\alpha}$ are scale-free

$$f(ax) = [ax]^{-\alpha} = a^{-\alpha} f(x) = bf(x), \text{ where } b = a^{-\alpha}$$

$\log f_\delta \sim c - \alpha \log \delta$ for some constant c
Power-law exponent (negative slope) is typically $\alpha \in [2, 3]$
The normalized power-law degree distribution is

$$f_\delta = \frac{\alpha - 1}{\delta_0} \left(\frac{\delta}{\delta_0} \right)^{-\alpha}, \text{ for } \delta \geq \delta_0.$$

More convenient to assume (for computation) that $\delta \in \mathbb{R}$ (instead of $\mathbb{N}$)

Random Graphs: Power Laws

For instance, the probability that a random node has degree exceeding 100 is

$$\mathbb{P}[f_\delta > 100] = \int_{100}^{\infty} \frac{\alpha - 1}{\delta_0} \left(\frac{x}{\delta_0} \right)^{-\alpha} dx = \left(\frac{100}{\delta_0} \right)^{1-\alpha}$$

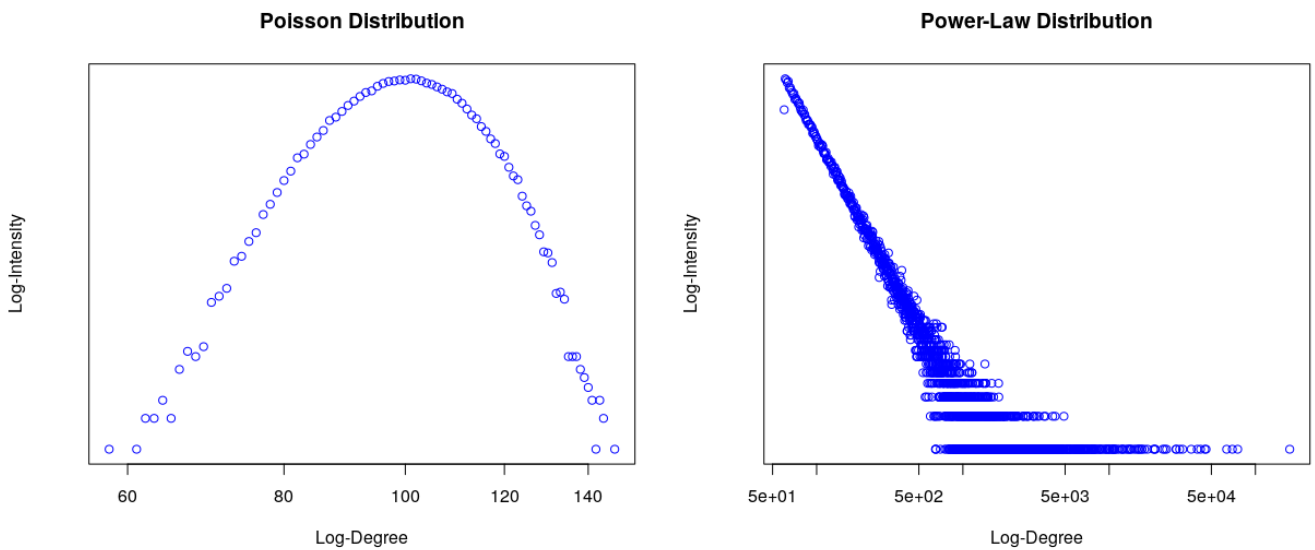
(which is also a power function, with index $\alpha - 1$).
Further

$$\mathbb{E}[f_\delta] = \frac{\alpha - 1}{\alpha - 2} \delta_0$$

but if $\alpha \in [2, 3]$, $\text{Var}[f_\delta] = \infty$.

Use log-log Pareto plots to visualize f_δ, either on log f_δ,

```
> dd.yeast <- table(degree(g))/n
> ind <- (dd.yeast != 0)
> d <- as.numeric(names(ind))
> plot(d[ind], dd.yeast[ind], log="xy")
```



... or on $\log \overline{F}_\delta$ (survival cumulative distribution)

```
> dd.yeast <- table(degree(g))/n
> ind <- (dd.yeast != 0)
> d <- as.numeric(names(ind))
> plot(d[ind], 1-cumsum(dd.yeast[ind]), log="xy")
```

(classical in extreme value theory: log-log Pareto plot)

Random Graphs: Power Laws

Natural to consider the linear least-squares (LS) estimator of c and α,

$$\min \left\{ \sum_{\delta} \left(\log \overline{F}_{\delta} - c + [\alpha - 1] \log \delta \right)^2 \right\}$$

Popular, but extremely noisy, biased (log transformations), and valid only when $\delta > \delta_0$

Here

$$f_{\delta} = \frac{\alpha - 1}{\delta_0} \left(\frac{\delta}{\delta_0} \right)^{-\alpha} \text{ for } \delta \geq \delta_0,$$

so that the log-likelihood function is (up to constants independent of α)

$$\log \mathcal{L}(\alpha) = \sum_{v=1}^{n_V} \log f_{\delta} \propto n_V \log[\alpha - 1] - \alpha \sum_{v=1}^{n_V} \log \left(\frac{\delta_v}{\delta_0} \right)$$

Random Graphs: Power Laws

and the maximum likelihood estimator is then

$$\widehat{\alpha} = 1 + \left(\frac{1}{n_V} \sum_{v=1}^{n_V} \log \left(\frac{\delta_v}{\delta_0} \right) \right)^{-1}$$

also called Hill estimator
Usually, we consider the k largest values (so-called tail)

$$\widehat{\alpha}_k = 1 + \left(\frac{1}{k} \sum_{i=0}^{k-1} \log \left(\frac{\delta_{n_V - i:n_V}}{\delta_{n_V - k:n_V}} \right) \right)^{-1}$$

Hill's plot is the graph of $\widehat{\alpha}_k$ as a function of k
in standard litterature on extreme value the so called α is $\alpha - 1$
from the network literature...

Preferential attachment model

Here also it is possible to derive a stochastic network representation.
Classical model for popularity, see Yule (1925, A Mathematical Theory of Evolution) or Merton (1968, The Matthew effect in science).

- with probability p, v' connects to v randomly (uniformly),
- with probability $1-p$, v' connects to v with probability proportional to the degree of v

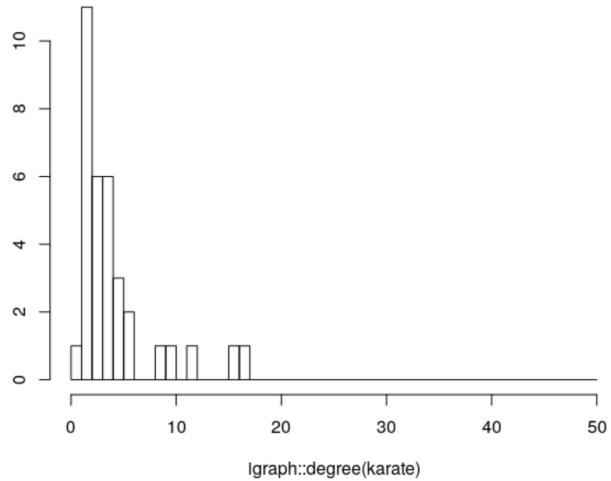
Preferential attachment model leads to rich-gets-richer dynamics
Degrees have a power-law distribution with tail exponent

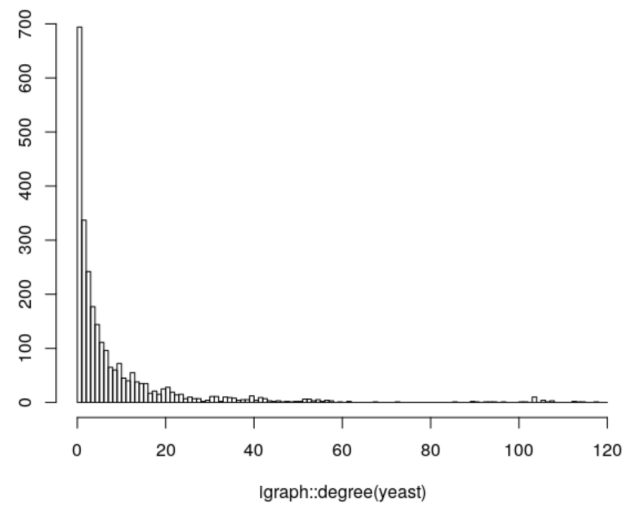
$$\alpha = 1 + \frac{1}{1-p}$$

$$f_\delta \sim \frac{1}{p}\left(\left(\frac{1-p}{p}\right)\delta + 1\right)^{-\left(1+\frac{1}{1-p}\right)}$$

Networks with R

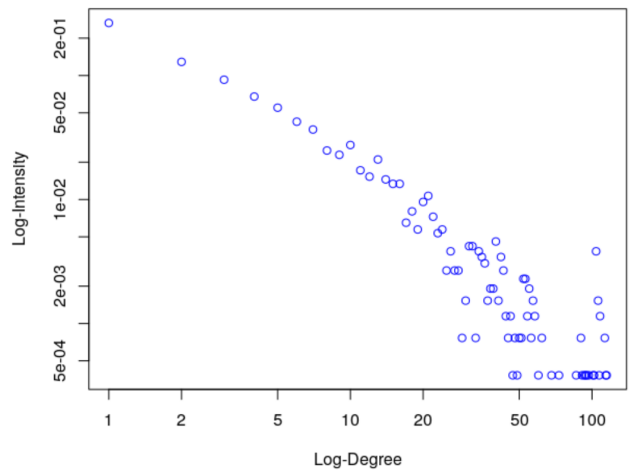
```
1 data(karate)
2 hist(igraph::degree(karate),breaks=
      seq(0,50))
3 data(yeast)
4 hist(igraph::degree(yeast),breaks=
    seq(0,120))
```





Networks with R

```
dd.yeast = degree.distribution(
    yeast)
d = 1:max(d.yeast)-1
ind = (dd.yeast != 0)
plot(d[ind], dd.yeast[ind], log="xy
    ", col="blue",xlab=c("Log-
    Degree"), ylab=c("Log-Intensity
    "))
```

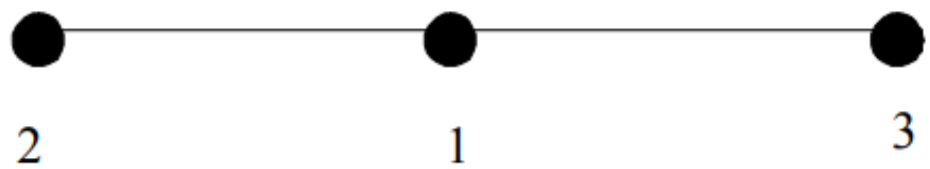


Paths in Networks

A walk of length k from vertice v_0 to vertice v_k is a sub-graph (V_p, E_p) with $V_p = \{v_0, v_1, \cdots, v_k\}$ and $E_p = \{(v_0, v_1), (v_1, v_2), \cdots, (v_{k-1}, v_k)\}$.
A trail is a walk without repeated edges
A path is a walk without repeated vertices, and therefore with different edges.
if $v_0 = v_k$, a walk is said to be closed. A cycle is a closed walk with different verticles (except $v_0 = v_k$), also called circuit.
A walk from v_0 to v_k contains a path from v_0 to v_k (remove subcycles).
If there is a walk from v to v', then v and v' are connected - or v' is reachable from v. A graph is connected if every vertex is reachable from every other.

Paths in Networks

Consider a network with 3 nodes and adjency matrix



$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
> A=matrix(c(0,1,1,1,0,0,1,0,0),3,3)
> A%*%A
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    1    1
[3,]    0    1    1
```

Paths in Networks

Proposition $[\boldsymbol{A}^k]_{i,j}$ is the number of walks of length k from i to j.
Indeed, there are 2 paths of length 2 from 1 to 1 : (1,2)+(2,1) or (1,3)+(3,1)
One can prove that

$$\boldsymbol{A}^{2k} = \begin{pmatrix} 2^k & 0 & 0 \\ 0 & 2^{k-1} & 2^{k-1} \\ 0 & 2^{k-1} & 2^{k-1} \end{pmatrix} \text{ and } \boldsymbol{A}^{2k+1} = \begin{pmatrix} 0 & 2^k & 2^k \\ 2^k & 0 & 0 \\ 2^k & 0 & 0 \end{pmatrix}$$

Exploration Algorithm: find the set of all vertices that can be reached by a walk from $v \in V$, denoted $\mathcal{C}(v)$.
Because of cycle properties, repeat exploration for vertices in $V \subset \mathcal{C}(v)$.

Connectivity in Networks

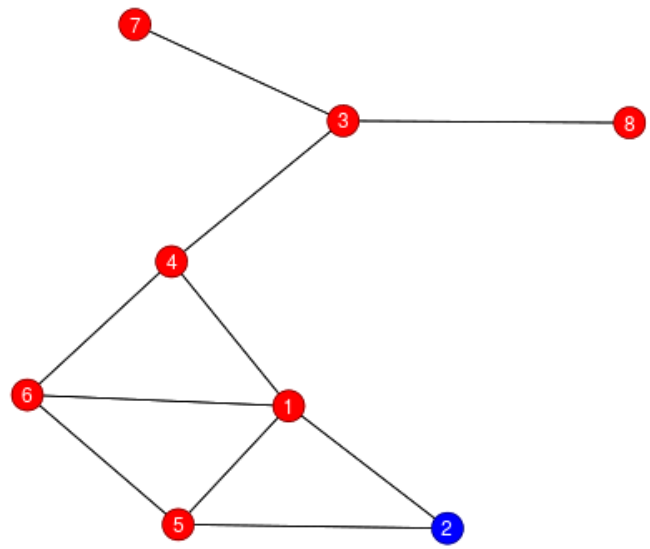
How to test for connectivity in a graph?
Idea : use adjacency list from a starting vertice s to explore

- Set $L = M = \{s\}$ and repeat while there are still nodes to explore ($L \neq \emptyset$)
- Pick $u \in L$
  - If there is an edge $(u, v) \in E$ with $v \in M$ then select one , and set $L = L \cup \{v\}$ and $M = M \cup \{v\}$
  - Otherwise prune, i.e. $L = L \backslash \{v\}$

Exploration is of order $2n_V$, each node is added and removed once.

Connectivity in Networks

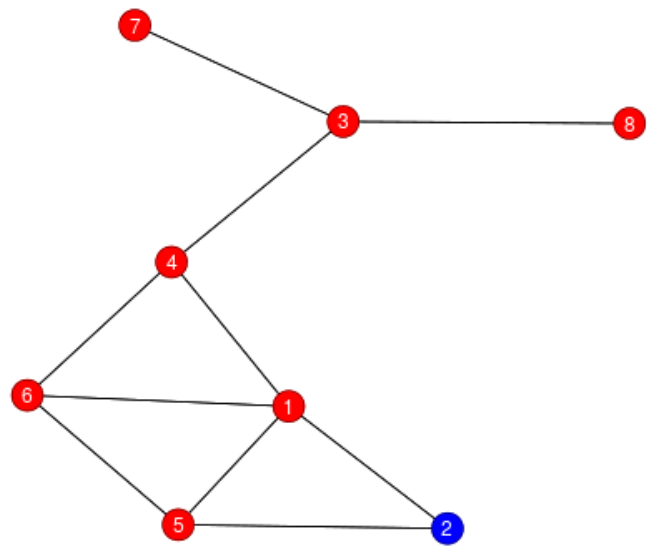
L	M
$\{2\}$	2
$\{2, 1\}$	1
$\{2, 1, 5\}$	5
$\{2, 1, 5, 6\}$	6
$\{1, 5, 6\}$	
$\{1, 5, 6, 4\}$	4
$\{5, 6, 4\}$	
$\{5, 4\}$	
$\{5, 4, 3\}$	3
$\{5, 3\}$	
$\{5, 3, 7\}$	7
$\{5, 3\}$	
$\{3\}$	
$\{3, 8\}$	8
$\{3\}$	
$\{\}$	



Connectivity in Networks (BFS)

Breadth-first search (BFS): u should be the first element of L

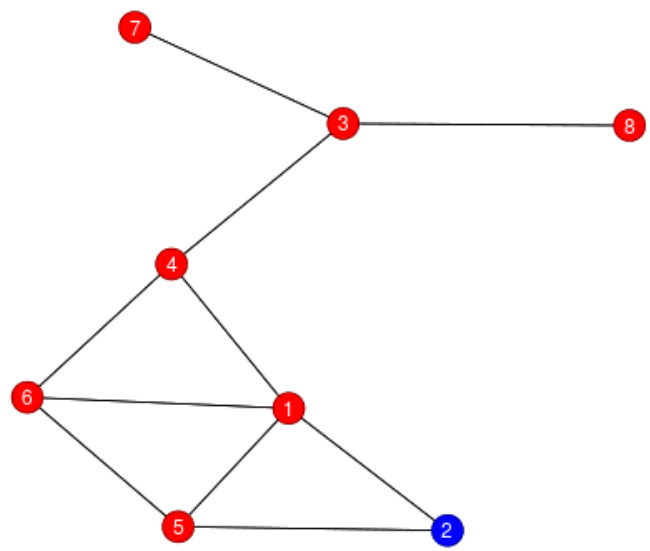
L	M
$\{2\}$	2
$\{2, 1\}$	1
$\{2, 1, 5\}$	5
$\{1, 5\}$	
$\{1, 5, 4\}$	4
$\{1, 5, 4, 6\}$	6
$\{5, 4, 6\}$	
$\{4, 6\}$	
$\{4, 6, 3\}$	3
$\{6, 3\}$	
$\{3\}$	
$\{3, 7\}$	7
$\{3, 7, 8\}$	8
$\{7, 8\}$	
$\{8\}$	
$\{\}$	



Connectivity in Networks (DFS)

Depth-first search (DFS): u should be the last element of L

L	M
$\{2\}$	2
$\{2, 1\}$	1
$\{2, 1, 4\}$	4
$\{2, 1, 4, 3\}$	3
$\{2, 1, 4, 3, 7\}$	7
$\{2, 1, 4, 3\}$	
$\{2, 1, 4, 3, 8\}$	8
$\{2, 1, 4, 3\}$	
$\{2, 1, 4\}$	3
$\{2, 1, 4, 6\}$	6
$\{2, 1, 4, 6, 5\}$	5
$\{2, 1, 4, 6\}$	
$\{2, 1, 4\}$	
$\{2, 1\}$	
$\{2\}$	
$\{\}$	



Connectivity in Networks

Another important concept is related to resilience of networks to the removal of some vertices.
As vertices are removed, shortest paths distances increase (can even be infinite when the network is disconnected).
Important in epidemiology and vaccination models.

Connectivity in Networks

The shortest path between two vertices (or nodes) in a graph is such that the number of its constituent edges is minimized.
This shortest path is often referred as the geodesic distance
The longest shortest path in a graph is the diameter of the graph
  more generally, the sum of the weights of its constituent edges is minimized.

Connectivity in Networks

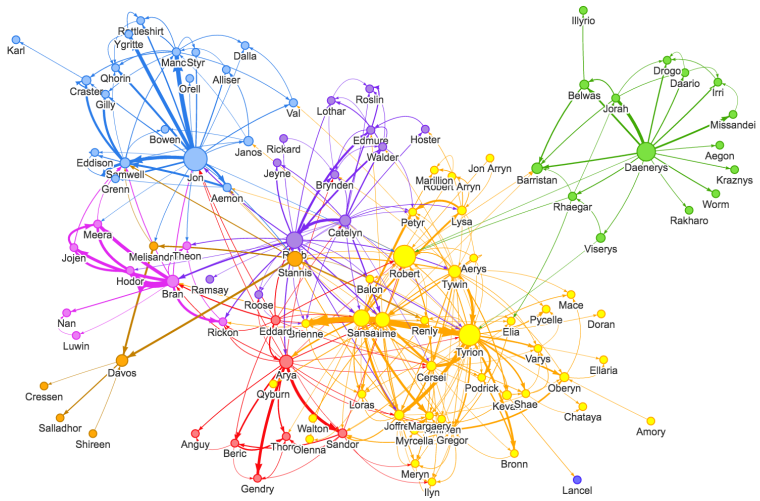
(initially designed for a digraph, with possible costs)
Start with $\mathcal{S} = \{v\}$, $\delta(v, v') = 1$ if $A_{v,v'} = 1$, ∞ otherwise.
Until $\mathcal{S} = V$: if there is $v' \notin \mathcal{S}$ such that $A_{u,v'} = 1$, $\forall u \in \mathcal{S}$, then $\mathcal{S} = V \cup \{v'\}$ and

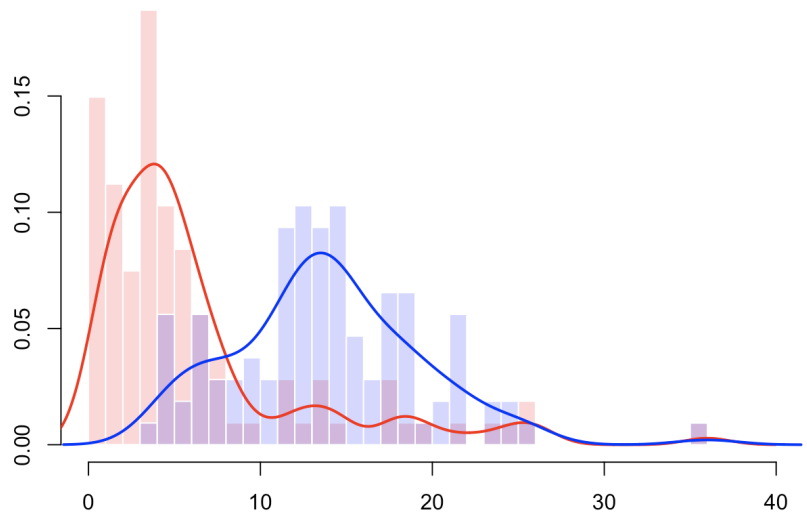
$$\delta(v') = \min\{\delta(v'), \delta(u) + 1\}$$

See Dijkstra's algorithm, or Floyd–Warshall algorithm

Friendship Paradox

"*The average degree of a friend is strictly greater than the average degree of a random node*", see Feld (1991) and Zuckerman & Jost (2001). See Game of Thrones network





Friendship Paradox

The average number of friends of a random person in the graph is

$$\mu = \frac{1}{n_V} \sum_{v \in V} d(v) = \frac{2n_E}{n_V} = \frac{1}{n_V} \|\boldsymbol{d}\|_1$$

where $\boldsymbol{d}$ is the vector of $d(v)$'s (or $\boldsymbol{d} = \boldsymbol{A}\mathbf{1}$).
The average number of friends that a typical friend has is

$$\frac{1}{n_V} \sum_{v \in V} \left(\frac{1}{d(v)} \sum_{v' : (v,v') \in E} d(v') \right)$$

Observe that each node v appears as a neighbour $d(v)$ times and each time it contributes to $d(v)$ neighbours. Since we simply count how many neighbours neighbours have, this average can also be written

$$\frac{1}{\|\boldsymbol{d}\|_1} \boldsymbol{d}^\top \boldsymbol{d}$$

Friendship Paradox

From Cauchy-Scharwz, for any $\boldsymbol{u}$ and $\boldsymbol{v}$, $\boldsymbol{u}^\top \boldsymbol{v} \leq \|\boldsymbol{u}\|_2 \cdot \|\boldsymbol{v}\|_2$. If $\boldsymbol{u} = \mathbf{1}$ and $\boldsymbol{v} = \boldsymbol{d}$, we can write

$$\mathbf{1}^\top \boldsymbol{d} = \|d\|_1 \leq \|\mathbf{1}\|_2 \cdot \|\boldsymbol{d}\|_2 = \sqrt{n_V} \cdot \sqrt{\boldsymbol{d}^\top \boldsymbol{d}}$$

After squaring and rearranging, we get

$$\frac{1}{\|\boldsymbol{d}\|_1} \boldsymbol{d}^\top \boldsymbol{d} \geq \frac{1}{n_V} \|\boldsymbol{d}\|_1$$

Alternatively, for a random variable D denoting the number of friends of a random individual, and D^2 is seen as number of friends of friends. Since $\text{Var}[D] = \mathbb{E}[D^2] - \mathbb{E}[D]^2$ we can write

$$\frac{\mathbb{E}[D^2]}{\mathbb{E}[D]} = \mathbb{E}[D] + \frac{\text{Var}[D]}{\mathbb{E}[D]} \geq \mathbb{E}[D]$$

because $\text{Var}[D] \geq 0$.

Friendship Paradox

"*You apply for a loan and your would-be lender somehow examines the credit ratings of your Facebook friends. If the average credit rating of these members is at least a minimum credit score, the lender continues to process the loan application. Otherwise, the loan application is rejected," the patent states.*" Bhattacharya (2015, Facebook patent: Your friends could help you get a loan - or not)

Consider some positive variable y observed at each node v. It can be the credit score of friends, or activity of people on twitter, as in Hodas *et al.* (2013, Friendship paradox redux: Your friends are more interesting than you)

Friendship Paradox

Using the extension of König-Huyghens,
$\text{Cov}[D, Y] = \mathbb{E}[DY] - \mathbb{E}[D]\mathbb{E}[Y]$, write $\boldsymbol{d}^\top \boldsymbol{y}$ as

$$\boldsymbol{d}^\top \boldsymbol{y} = \frac{\|\boldsymbol{d}\|_1 \|\boldsymbol{y}\|_1}{n_V} + n_V \text{Cov}[D, Y]$$

or

$$\frac{1}{\|\boldsymbol{d}\|_1} \boldsymbol{d}^\top \boldsymbol{y} = \frac{1}{n_V} \|\boldsymbol{y}\|_1 + \frac{n_v}{\|\boldsymbol{d}\|_1} \text{Cov}[D, Y]$$

This can be written also

$$\frac{\mathbb{E}[DY]}{\mathbb{E}[D]} = \mathbb{E}[Y] + \frac{\text{Cov}[D, Y]}{\mathbb{E}[D]}$$

Hence,

$$\frac{\mathbb{E}[DY]}{\mathbb{E}[D]} = \mathbb{E}[Y] \text{ if } X \perp Y \text{ and } \frac{\mathbb{E}[DY]}{\mathbb{E}[D]} > \mathbb{E}[Y] \text{ if } \text{Cov}[D, Y] > 0.$$

Friendship Paradox

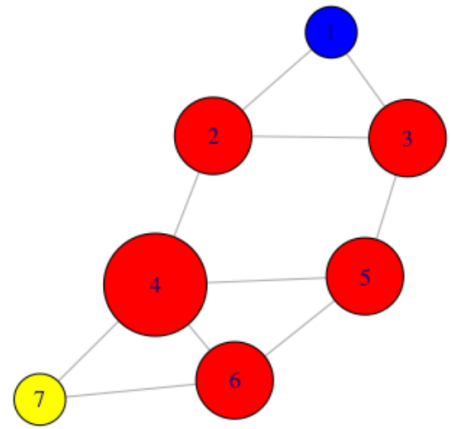
This concept is related to the *generalized friendship paradox with respect to quantity y* from Em & Jo (2014, Generalized friendship paradox in complex networks: The case of scientific collaboration) which states that it will arise if y is nonnegatively correlated with degree d, or Charpentier & Galichon (2019).

Networks & Centrality

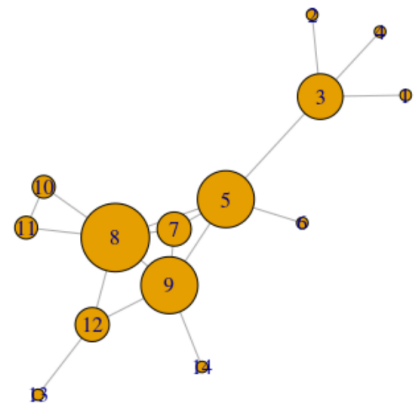
The goal is to understand who is important, based on their network position.
"*There is certainly no unanimity on exactly what centrality is or on its conceptual foundations, and there is little agreement on the proper procedure for its measurement*", Freeman (1979, Centrality in Social Networks)

Simple question such as *which vertice is central* can be replaced by more interesting ones, such as *what percentage of vertices is crucial to the network connectivity* ?
He or she who has many friends is most important, divided by the maximum possible n_V : for a vertice v, $C_D(v) = \frac{d(v)}{n_V}$

Networks with R

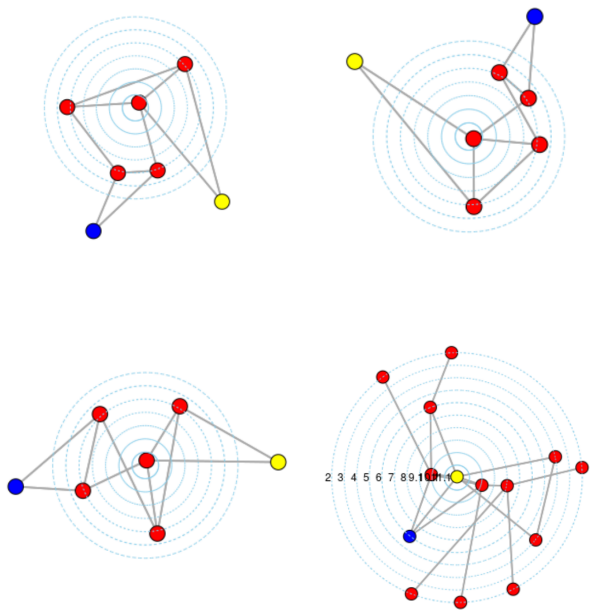



```
plot(g,vertex.size=degree(ng))
```



Networks with R

```
A = get.adjacency(g1, sparse=FALSE)
ng = network::as.network.matrix(A)
sna::gplot.target(ng, degree(ng),
    circ.col="skyblue",vertex.col=c
    ("blue", rep("red", 5), "yellow
    "), edge.col="darkgray")
degree(ng)
[1] 4 6 6 8 6 6 4
```



How much variation is there in the centrality scores among the nodes?
One can define a centralization index, using any dispersion measure (variance, Gini index).
Freeman's centralization is based to the distance to the maximum

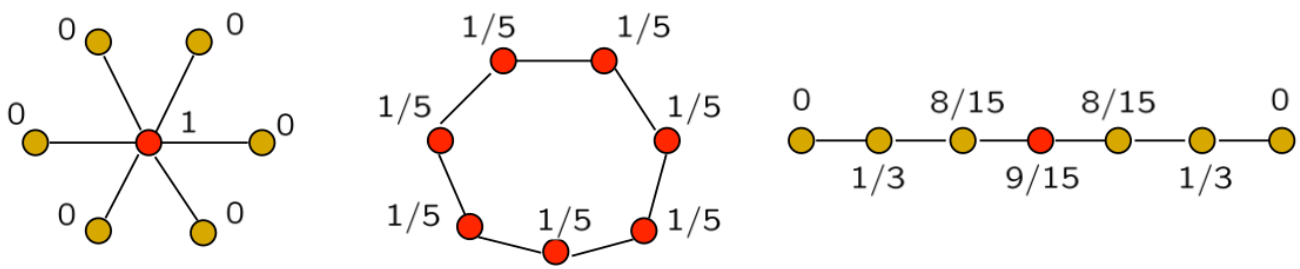
$$C_D = \frac{1}{(n_V - 1)(n_V - 2)} \sum_{v \in V} [d(v^\star) - d(v)]$$

where $v^\star = \text{argmax}\{d(v)\}$ (supposed to be unique).

How many pairs of individuals would have to go through you in order to reach one another in the minimum number of hops? The betweenness centrality of a node v is

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to t, and $\sigma_{st}(v)$ is the number of those paths that pass through v.



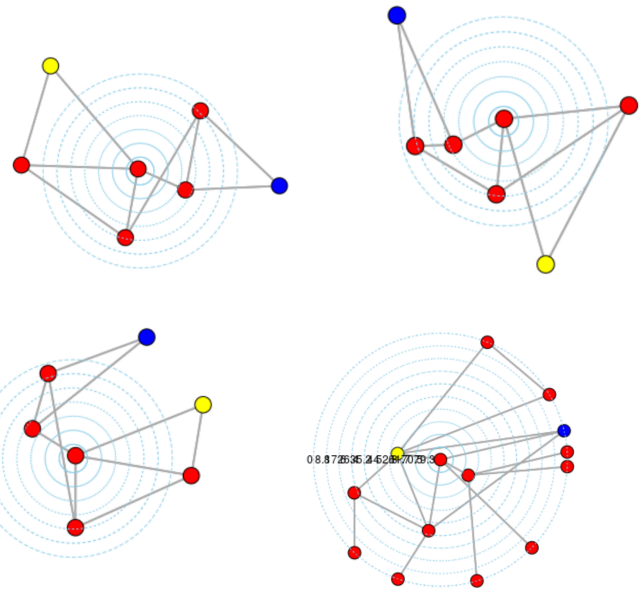
One possible normalization is $C_{B'}(v) = \dfrac{2C_B(v)}{(n_V - 1)(n_V - 2)}$ (number of pairs of vertices excluding the vertex itself), another one is $C_{\tilde{B}}(v) = \dfrac{C_B(v) - \min C_B}{\max C_B - \min C_B}$

Computation time can be very long, $O(n_V^3)$, but one can reach $O(n_V n_E)$, using Brandes (2001, A faster algorithm for betweenness centrality)

$v^\star = \text{argmax}\{C_B(v)\}$ is seen as the controler of information flow

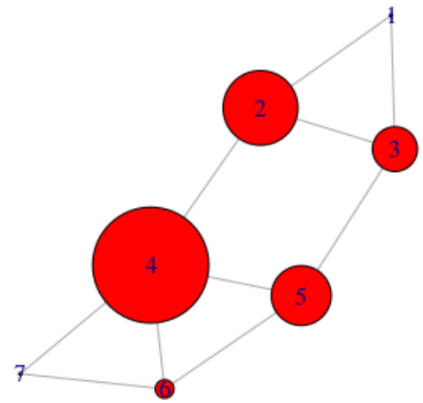
Networks with R

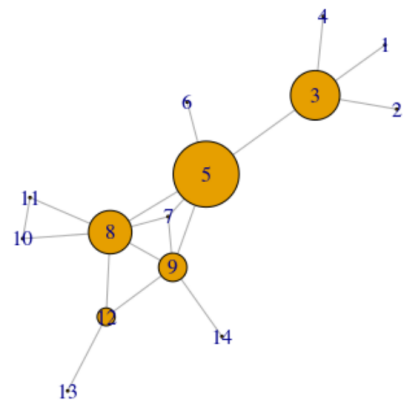
```
> A = get.adjacency(g1, sparse=
    FALSE)
> ng = network::as.network.matrix(A
    )
> sna::gplot.target(ng, betweenness
    (ng), circ.col="skyblue",vertex
    .col=c("blue", rep("red", 5), "
    yellow"), edge.col="darkgray")
> betweenness(ng)
[1]  0.000  6.667  4.000 10.333
    5.333  1.667  0.000
```



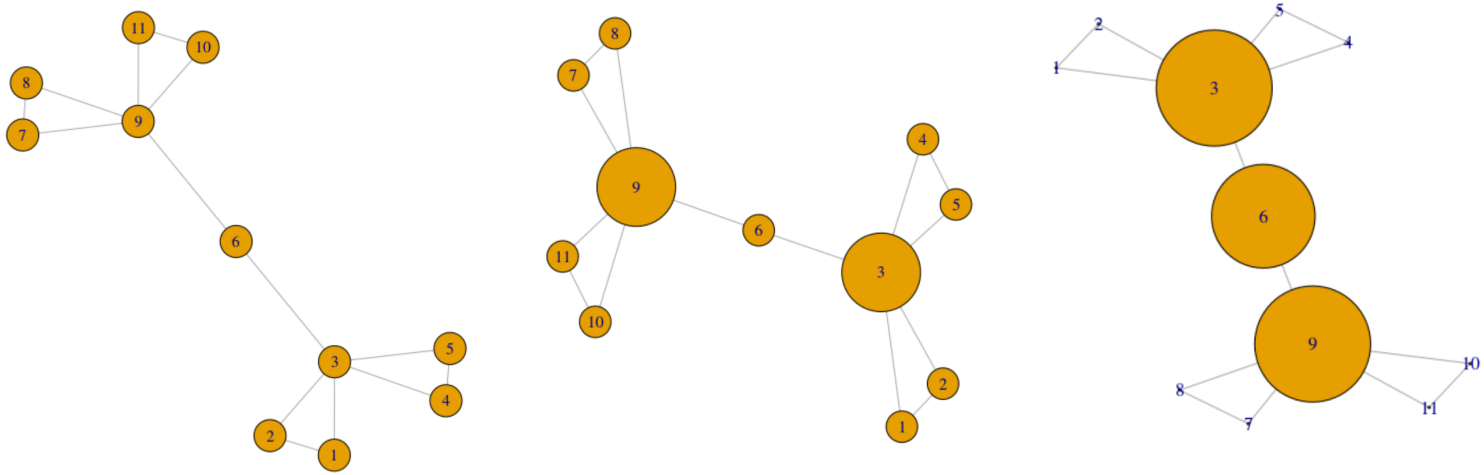
Networks with R

```
> plot(g,vertex.size = betweenness
    (ng))
```





Networks & Centrality

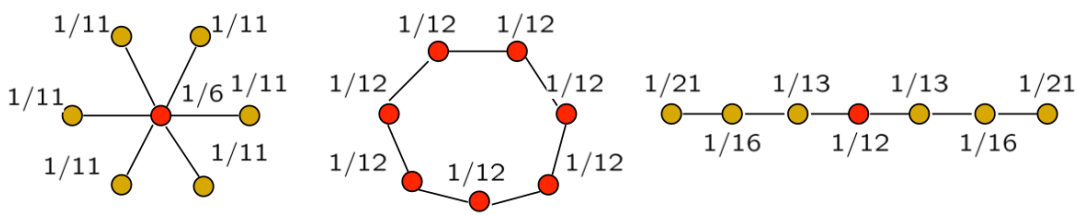


Closeness is based on the length of the average shortest path between a vertex and all vertices in the graph,

$$C_C(v) = \frac{1}{\sum_y \text{dist}(y, v)}.$$

where dist is the geodesic distance, and normalized Closeness Centrality is

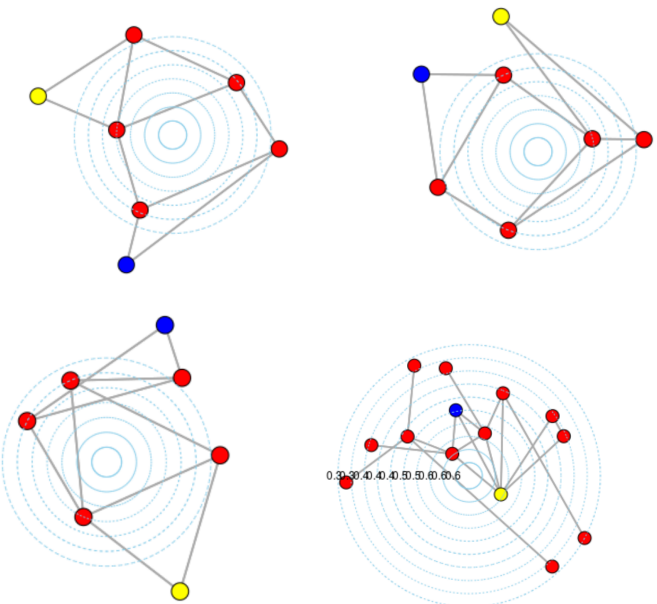
$$C_{C'}(v) = \frac{C_C(v)}{n_V} \text{ or } \frac{C_C(v)}{n_V - 1}$$



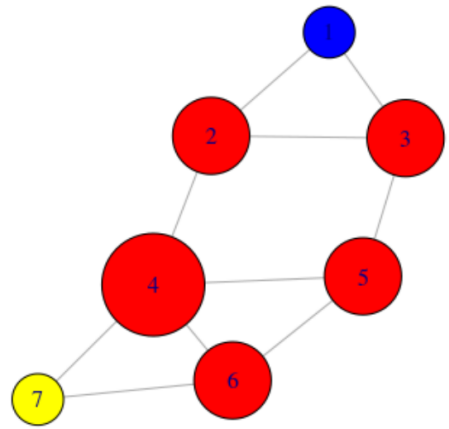
Computation time is $O(n_V^2 \log n_V + n_V n_E)$ on sparce network. $v^\star = \text{argmax}\{C_C(v)\}$ is the most approachable vertice.

Networks with R

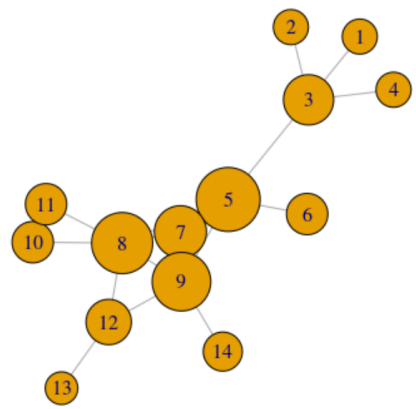
```
> A = get.adjacency(g1, sparse=
    FALSE)
> ng = network::as.network.matrix(A
    )
> sna::gplot.target(ng, closeness(
    ng), circ.col="skyblue",vertex.
    col=c("blue", rep("red", 5), "
    yellow"), edge.col="darkgray")
> closeness(ng)
[1] 0.5000 0.6667 0.6000 0.7500
    0.6667 0.6000 0.5000
```



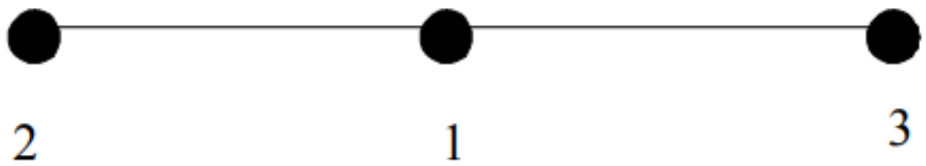
Networks with R




```
> plot(g,vertex.size=closeness(ng))
```



Consider a network with 3 nodes and adjacency matrix



$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
> A=matrix(c(0,1,1,1,0,0,1,0,0),3,3)
> A%*%A
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    1    1
[3,]    0    1    1
```

One can prove that

$$\boldsymbol{A}^{2k} = \begin{pmatrix} 2^k & 0 & 0 \\ 0 & 2^{k-1} & 2^{k-1} \\ 0 & 2^{k-1} & 2^{k-1} \end{pmatrix} \text{ and } \boldsymbol{A}^{2k+1} = \begin{pmatrix} 0 & 2^k & 2^k \\ 2^k & 0 & 0 \\ 2^k & 0 & 0 \end{pmatrix}$$

```
> eigen(A)
$values
[1]  1.414214  0.000000 -1.414214

$vectors
           [,1]       [,2]       [,3]
[1,] -0.7071068  0.0000000  0.7071068
[2,] -0.5000000 -0.7071068 -0.5000000
[3,] -0.5000000  0.7071068 -0.5000000
```

Eigenvalues are $\{\sqrt{2}, 0, -\sqrt{2}\}$. Eigenvector with positive components associated to the largest eigenvector (see Perron-Frobenius theorem) is

$$\boldsymbol{x} = \frac{1}{2}\left(\sqrt{2}, 1, 1\right)$$

```
> eigen_centrality(g)
$vector
        1         2         3
1.0000000 0.7071068 0.7071068

$value
[1] 1.414214
```

This eigenvalue centrality score is related to the following (recursive) equation

$$C_E(v) \propto \sum_{u \in M(v))} C_E(u) = \sum_{u \in E} A_{v,u} C_E(u)$$

where $M(v)$ is the set of neighbors of v, or equivalently

$$\boldsymbol{A}\boldsymbol{C}_E \propto \boldsymbol{C}_E$$

which is an eigenvector equation, $\boldsymbol{A}\boldsymbol{C}_E = \lambda \boldsymbol{C}_E$ for some λ.

Theorem [Perron–Frobenius] Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries, $a_{ij} > 0$ for $1 \leq i, j \leq n$, then

- There is a positive real number r, called the Perron–Frobenius eigenvalue, such that r is an eigenvalue of A and any other eigenvalue (possibly, complex) is strictly smaller than r in absolute value
- There exists an eigenvector $\boldsymbol{x}$ of $\boldsymbol{A}$ with eigenvalue r such that all components of $\boldsymbol{x}$ are positive
- There are no other positive eigenvectors except positive multiples of $\boldsymbol{x}$

Proof Meyer (2000, Matrix analysis and applied linear algebra) Hence, λ is Perron–Frobenius eigenvalue, and C_E is the (normalized) eigenvector

Other related measures can be considered.
Katz status index is a weighted count of all paths coming to the node, attenuated with factor β.
Define the vector of Bonacich centralities of parameter β, for $\boldsymbol{u} \in \mathbb{R}_+^{nv}$

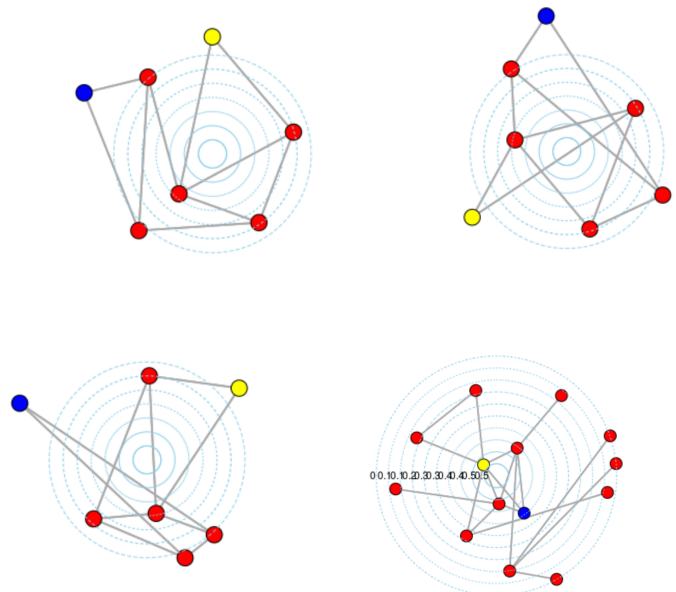
$$b_1(\boldsymbol{G}, \beta) = \sum_{h \geq 1} \beta^h \boldsymbol{A}^h \mathbf{1} = ([\mathbb{I} - \beta \boldsymbol{A}]^{-1} - \mathbb{I})\mathbf{1}$$

if β is smallest than the inverse of the largest eigenvalue, see see Bonacich (1987), or

$$b_1(\boldsymbol{G}, \beta) = \beta[\mathbb{I} - \beta \boldsymbol{A}]^{-1} \boldsymbol{A}\mathbf{1}$$

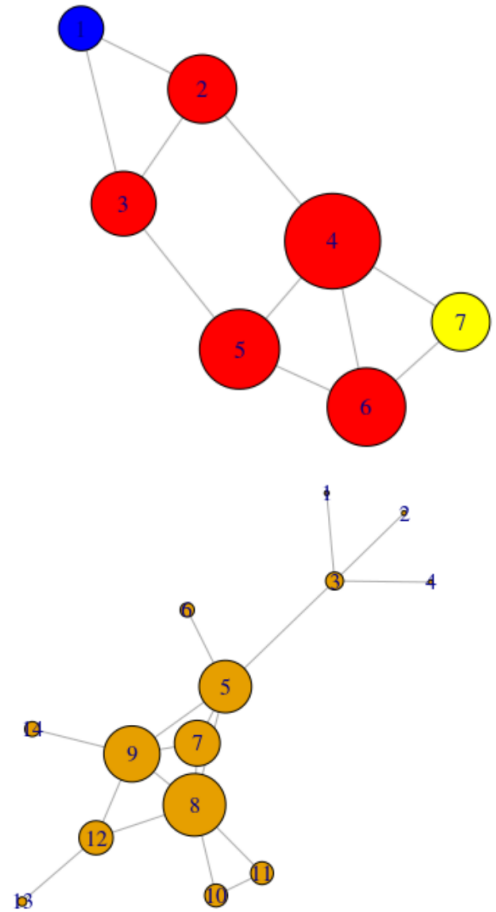
Networks with R

```
> A = get.adjacency(g1, sparse=
    FALSE)
> ng = network::as.network.matrix(A
    )
> sna::gplot.target(ng, evcent(ng),
     circ.col="skyblue",vertex.col=
    c("blue", rep("red", 5), "
    yellow"), edge.col="darkgray")
> evcent(ng)
[1] 0.2358069 0.3622484 0.3416056
    0.5038536 0.4215946 0.4129477
    0.3071490
```



Networks with R

```
1 > plot(g,vertex.size=evcent(ng))
```



Networks & Centrality

Centrality measures we've seen use the shortest path between pairs of vertices.
Katz centrality measures considers the total number of paths between pairs of vertices.

$$C_K(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n_V} \alpha^k (\boldsymbol{A}^k)_{ji}$$

with some attenuation coefficient α.
α has to be smaller than the inverse of the largest eigenvalue of the adjacency matrix $\boldsymbol{A}$.
One can also write $C_K = ([\mathbb{I} - \alpha \boldsymbol{A}^\mathsf{T}]^{-1} - \mathbb{I})\mathbf{1}$
Can be related to random walks on a graph : from vertice v, with probability α move to a neighbor of v, and with probability $1 - \alpha$ move to any vertice in V.

Score of a page is proportional to the sum of the scores of pages linked to it
Define $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$ where $\boldsymbol{D} = \text{diag}(\boldsymbol{d})$ is the diagonal matrix of degrees (out-degree for internet pages), and consider the following update rule

$$R^{(k)}(v) = \sum_{u \in M(v)} \frac{1}{d_u} R^{(k-1)}(u) \text{ i.e. } \boldsymbol{R}^{(k)} = \boldsymbol{P}^{\mathsf{T}} \boldsymbol{R}^{(k-1)}$$

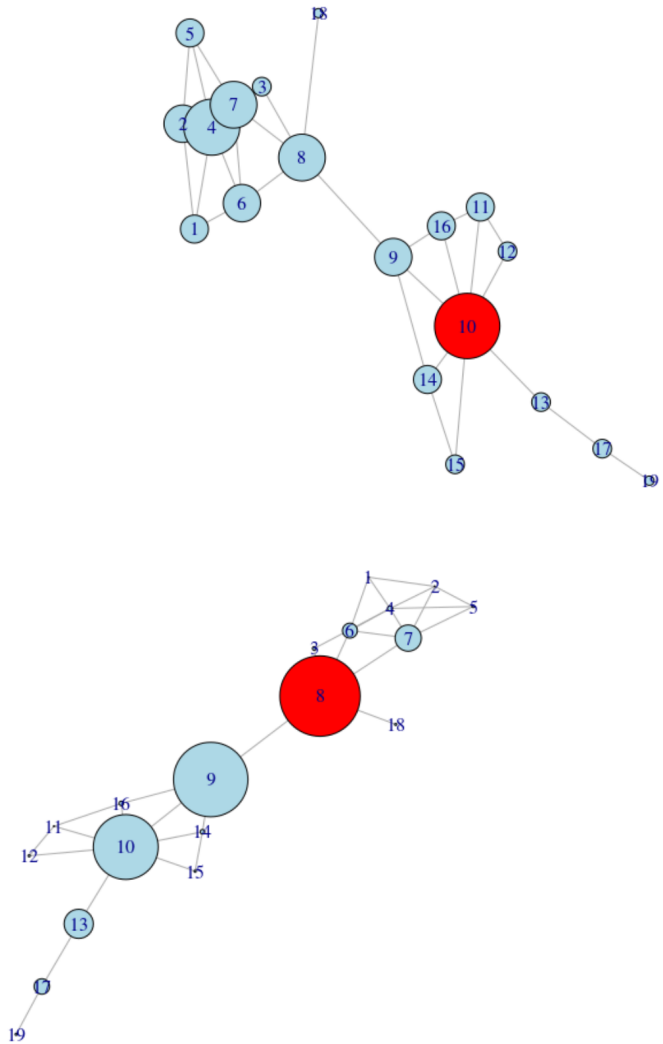
initialized with $\boldsymbol{R}^{(0)} = 1/N_V$. Or consider some scaled version

$$\boldsymbol{R}^{(k)} = \widetilde{\boldsymbol{P}}^{\mathsf{T}} \boldsymbol{R}^{(k-1)} \text{ where } \alpha \widetilde{\boldsymbol{P}} + (1-\alpha)\frac{\mathbf{1}\mathbf{1}^{\mathsf{T}}}{N_v}$$

where the scaling factor β is typically around 0.85
power iteration converges to the dominent eigenvector of $\widetilde{\boldsymbol{P}}$.

Networks with R

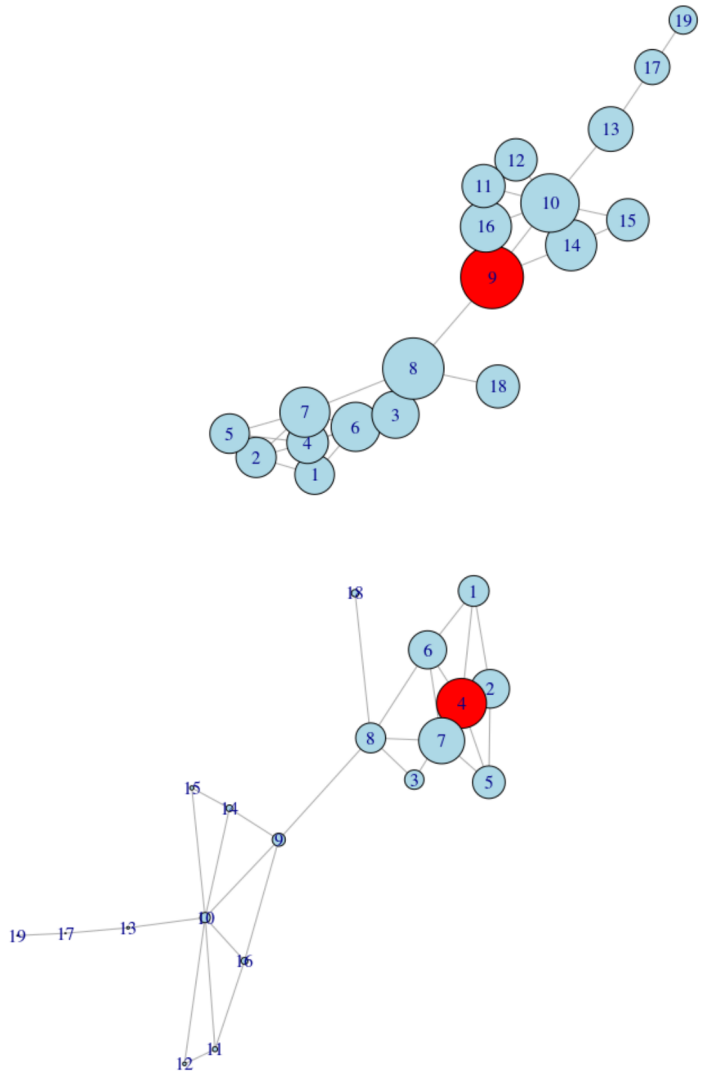
```
> A <- get.adjacency(g, sparse=
    FALSE)
> library(network)
> ng <- network::as.network.matrix(
    A)
> library(sna)
> which.max(sna::degree(ng))
[1] 13
> which.max(sna::betweenness(ng))
[1] 8
```



Networks with R

```
> A <- get.adjacency(g, sparse=
    FALSE)
> library(network)
> ng <- network::as.network.matrix(
    A)
> library(sna)
> which.max(sna::closeness(ng))
[1] 9
> which.max(sna::evcent(ng))
[1] 3
```

None is wrong, they just focus on different notions of 'importance'



Need a metric on the space of networks. Let $G_1 = (E_1, V)$ and $G_2 = (E_2, V)$

$$\Delta(G_1, G_2) = \sum_{e \in E_1 \cup E_2} |\mathbf{1}_{e \in E_1} - \mathbf{1}_{e \in E_2}|$$

A centrality measure C is said to be stable if for any $v \in V$ and any graphs G_1 and G_2,

$$|C^{G_1}(v) - C^{G_2}(v)| \leq \kappa \Delta(G_1, G_2)$$

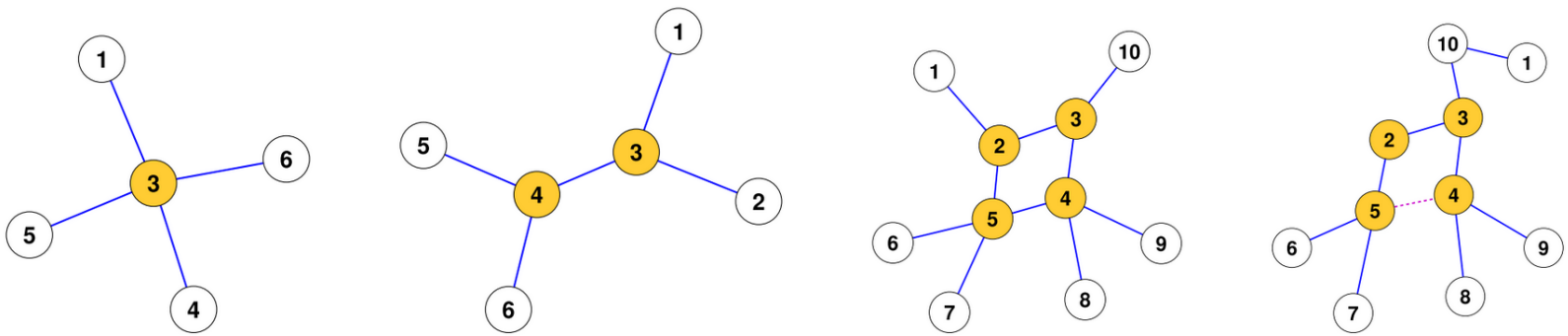
for some constant κ (see Lipschitz continuity).
Idea: the importance of nodes should be robust to small perturbations in the graph.
For instance, one can prove that for the degree

$$|C_D^{G_1}(v) - C_D^{G_2}(v)| \leq \Delta(G_1, G_2)$$

thus, degree centrality is a stable measure.

Centrality, Vulnerability & Security

See Worldwide Metro Networks, from Wang *et al.* (2017, Multi-criteria robustness analysis of metro networks)



(a) Rome. (b) Cairo and Marseille. (c) Montreal. (d) Toronto.

Defined several measures of robustness of metro networks around the world

Centrality, Vulnerability & Security

See Worldwide Metro Networks, from Wang *et al.* (2017, Multi-criteria robustness analysis of metro networks)

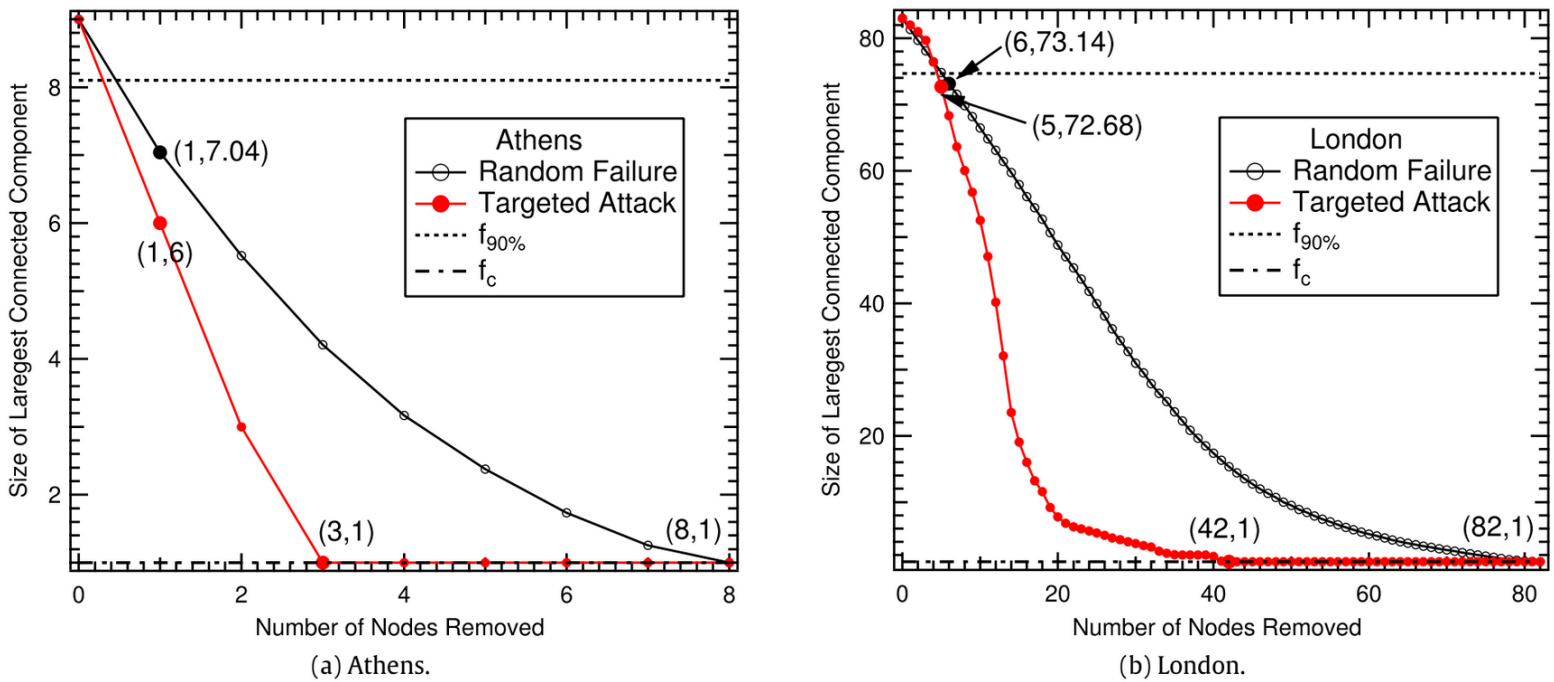


Fig. 4. Critical thresholds in metro networks under nodes removal.

Recall classical statistical tools: consider a sample of observations $\boldsymbol{x} = \{x_1, \cdots, x_n\}$ from i.i.d. random variables $X_1, \cdots, X_n$, with distribution F.
Assume that $F \in \mathcal{F} = \{F_\theta, \theta \in \Theta\}$
Consider some point estimator $\widehat{\theta} = s(x_1, \cdots, x_n)$
Estimator can be seen as a random variable $\widehat{\theta} = s(X_1, \cdots, X_n)$
The bias is $\mathbb{E}[\widehat{\theta}] - \theta$ and its standard-error $\text{Var}[\widehat{\theta}]^{1/2}$. The mean squarred error is

$$\text{mse}[\widehat{\theta}] = \mathbb{E}([\widehat{\theta} - \theta]^2) = \text{bias}[\widehat{\theta}]^2 + \text{Var}[\widehat{\theta}]$$

Inference & Networks

Survey sampling is a standard tool in socio-economic studies
Network sampling designs provide an alternative
see McCormick, He, Kolaczyk & Zheng (2012, Surveying hard-to-reach groups through sampled respondents in a social network)
Let $G = (V, E)$ denote a network, and let $\theta(G)$ denote a statistics of interest, e.g.

- the number of nodes n_V
- the number of connexions n_E
- the degree d_v of a node v

using a sampled subgraph of G is maybe not a great idea...

Inference & Networks

Measurements are usually on a portion of the complex population.
Here, we cannot observe $G = (V, E)$ but only $G^\star = (V^\star, E^\star)$, a subgraph of G.
Consider some statistics of interest on the graph, $\theta(G)$.
Can we consider the natural plug-in estimator $\widehat{\theta} = \theta(G^\star)$?
Consider the case where $\theta(G)$ is the average degree,
$\theta(G) = \frac{1}{n_V} \sum_{v \in V} d_v$. How could we sample ?

- sample n vertices $V_n^\star = v_1, \cdots, v_n$ (without replacement)
  - for each $v_k \in V_n^\star$ observe incident edges $(v_k, v) \in E$
  - observe edges only when vertices are in $V^\star$, $v_{k_1} \in V_n^\star$ and $v_{k_2} \in V_n^\star$.

Set $\widehat{\theta} = \frac{1}{n} \sum_{v \in V_n^\star} d_v^\star$

The first technique is fine, $\widehat{\theta} \sim \theta(G)$ but the second one under-estimate $\theta(G)$ (actually, $\widehat{\theta} \sim n\theta(G)/n_V$).
Necessary to incorporate effects of random sampling.
Consider there the mean μ of y_i's.
Let π_i denote the probability to have individual i, in the sample.

$$\mathbb{E}[\widehat{\mu}] = \frac{1}{n}\sum_{i=1}^{m} \pi_i y_i \neq \mu \text{ if } \pi_i \neq \frac{n}{m}$$

See Horvitz-Thompson's estimator (see Horvitz & Thompson (1952)): unequal probability sampling necessitates unequal weights when averaging : use inclusion probabilities as weights.

An unbiased estimator of μ is $\widetilde{\mu}_{HT} = \frac{1}{m} \sum_{i \in \text{sample}} \frac{y_i}{\pi_i}$,

$$\mathbb{E}[\widetilde{\mu}_{HT}] = \mathbb{E}\left[\frac{1}{m} \sum_{i=1}^{n} \frac{y_i}{\pi_i} \underbrace{\mathbf{1}(i \text{ is in sample})}_{=Z_i} \right] = \frac{1}{m} \sum_{i=1}^{n} \frac{y_i}{\pi_i} \underbrace{\mathbb{E}[Z_i]}_{=\pi_i} = \mu.$$

Further

$$\text{Var}[\widetilde{\mu}_{HT}] = \frac{1}{n^2} \sum_{i,j=1}^{n} y_i y_j \left(\frac{\pi_{i,j}}{\pi_i \pi_j} - 1 \right)$$

The main problem here is to compute π's. Furthermore

$$\text{Var}[\widetilde{\mu}] = \frac{1}{m^2} \sum_{i=1}^{m} \left(\frac{1 - \pi_i}{\pi_i^2} \right) y_i^2 + \frac{1}{m^2} \sum_{i \neq j=1}^{m} \left(\frac{\pi_{i,j} - \pi_i \pi_j}{\pi_i \pi_j} \right) y_i y_j$$

where $\pi_{i,j}$ is the probability to have i and j in the sample.

with simple random sampling (without replacement)

$$\pi_{i,j} = \frac{n(n-1)}{m(m-1)}.$$

but need to know m.
Use the mark and recapture estimator, see Chao *et al.* (2001, The applications of capture-recapture models to epidemiological data) to estimate m. Consider two samples $\mathcal{S}_1$ and $\mathcal{S}_2$.
Step 1: mark all units in sample $\mathcal{S}_1$ of size n_1
Step 2: create a sample $\mathcal{S}_2$ of size n_2

$$\widehat{m} = \frac{\text{Card}(\mathcal{S}_1) \cdot \text{Card}(\mathcal{S}_2)}{\text{Card}(\mathcal{S}_1 \cap \mathcal{S}_2)}$$

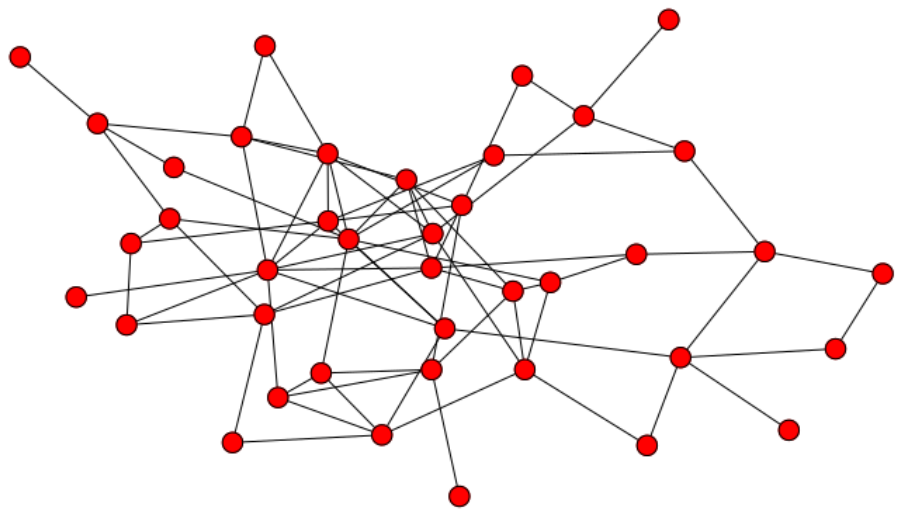
Complicated to estimate if the population is V (vertices) or E (edges).

Inference & Networks

Idea: take n vertices, set V_n, and observe all edges in the subgraph induced by V_n,

$$\pi_i = \frac{n}{n_V} \text{ and } \pi_{i,j} = \frac{n(n-1)}{n_v(n_V - 1)}$$

for vertices inclusion. See friendship paradox



Idea: take n edges, set E_n, and observe all vertices incident to edges in E_n,

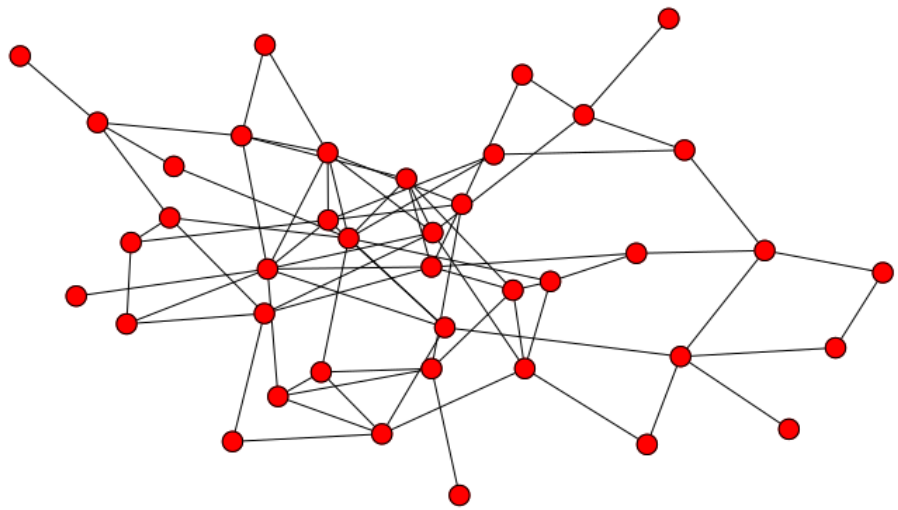
$$\pi_i = \frac{n}{n_V} \text{ and } \pi_{i,j} = \frac{n(n-1)}{n_v(n_V-1)}$$

See phone calls
Vertices inclusion probabilities are here

$$\pi_i = \mathbb{P}[\text{vertice } i \text{ is sampled}] = 1 - \frac{\binom{n_E - d_i}{n}}{n_E} n$$

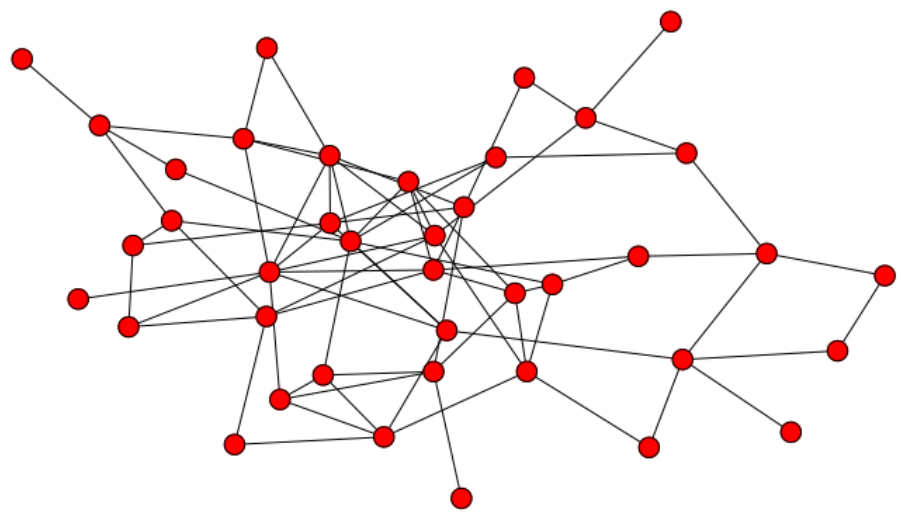
if $n \leq n_E - d_i$ (1 otherwise).

Inference & Networks



also called Snowball Sampling

Idea: start with an initial vertice sample V_0 and observe all incident edges, as well as all vertices sharing those edges. Iterate to have n edges.



Link Tracing

Idea: start with an initial vertice sample V_0 and observe *some* incident edges (called 'links'), as well as all vertices sharing those edges. Iterate to have n edges.

Path Sampling

Select a set of source edges, and target sources, and consider some shortest paths from each source to all targets. Called Traceroute sampling. One can prove that

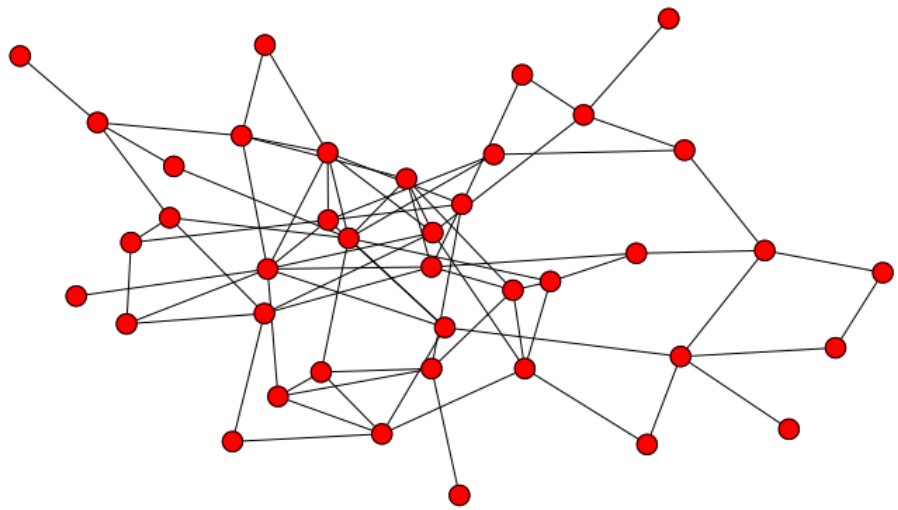
$$\pi_i \sim 1 - \left(1 - \frac{n_s}{n_E}\frac{n_t}{n_E}\right) \exp\left(-\frac{n_s}{n_E}\frac{n_t}{n_E} b_i\right)$$

and

$$\pi_{i,j} \sim 1 - \exp\left(-\frac{n_s}{n_E}\frac{n_t}{n_E} b_{i,j}\right)$$

where b_i is the betweness centrality of vertice i and $b_{i,j}$ is the betweness centrality of edge (i, j).

Inference & Networks

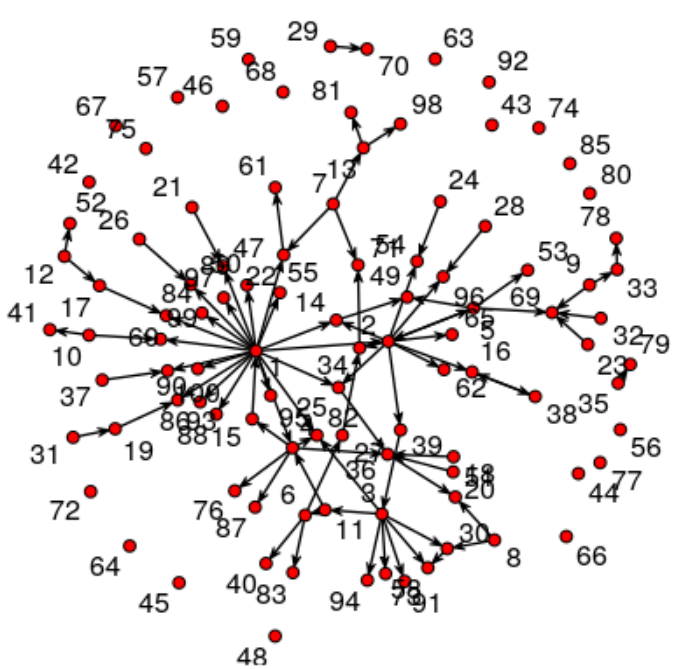


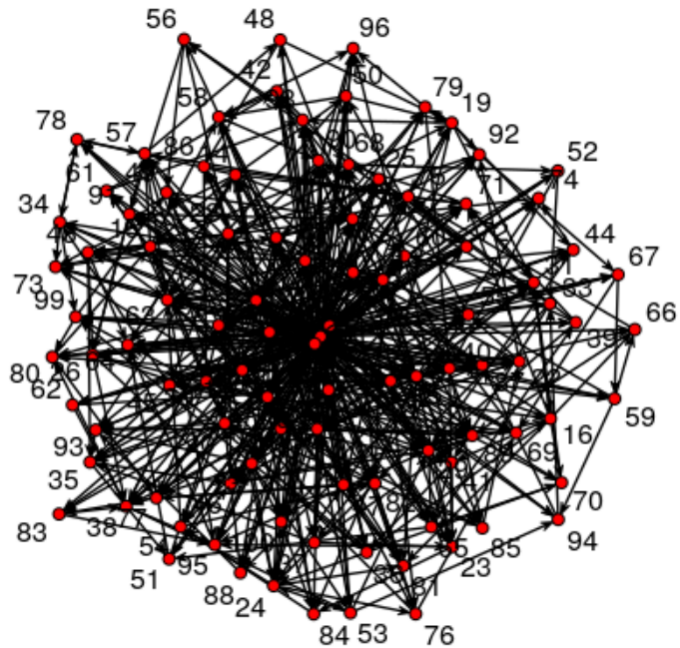
see Dall'Asta *et al.* (2006, Vulnerability of weighted networks) for more details.

Inference & Networks

emotion/behaviors of individuals being influenced by the group(s) to which they belong
see Christakis & Fowler (2009, Connected : The Surprising Power of Our Social Networks and How They Shape Our Lives)
challenge of distinguishing social contagion from ordinary homophily

Inference & Networks





Do friends of peoples tend to be friends together?
The cohesion can be related to density, clustering, connectivity or transitivity (*the friend of your friend is likely to be your friend*) etc.
The density of a sub-graph $H \subset G$ is defined as

$$\text{density}(H) = \frac{2n_{E_H}}{n_{V_H}(n_{V_H} - 1)}$$

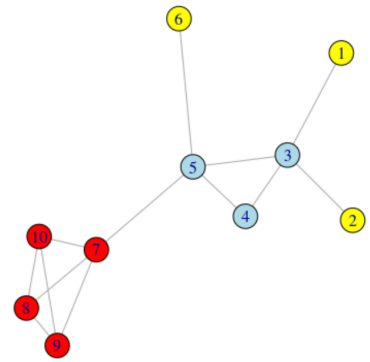
where $H = (V_H, E_H)$, which is the number of edges over the maximum number of edges (based on the number of vertices).

```
> graph.density(g1)
[1] 0.4761905
> graph.density(g2)
[1] 0.1978022
```

Usually, we consider for H the neighborhood of a vertice v.

Clusters & Networks

A clique is a complete subgraph of G.
Note that large cliques are rare, since a single missing edge destroys the property.
A sufficient condition to have a clique of size n is

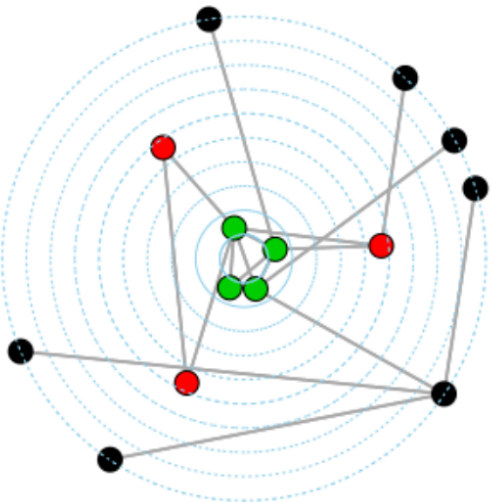


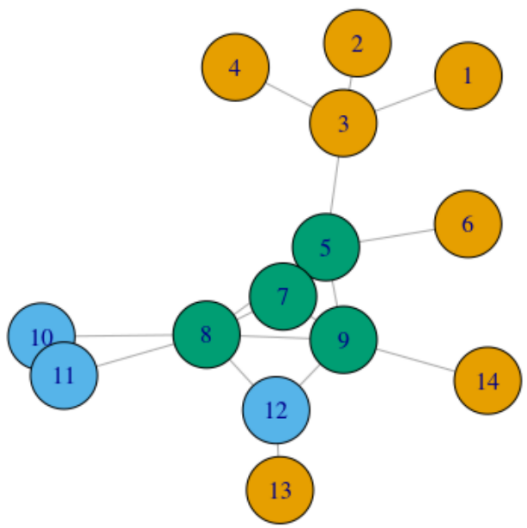
$$n_E > \frac{n_V^2}{2} \frac{n-2}{n-1}$$

A k-core $G^\star$ is a subgraph of G such that $d_v \star \geq k$ for all $k \in V^\star$, and $G^\star$ is maximal.
Here, degrees are computed on the subgraph $G^\star$.

Networks with R

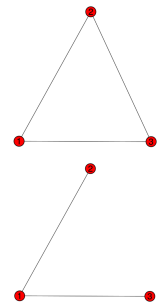
```
A = get.adjacency(g2, sparse=FALSE)
ty= network::as.network.matrix(A)
cores = graph.coreness(g2)
sna::gplot.target(ty, cores, circ.
    col="skyblue", usearrows =
    FALSE, vertex.col=cores, edge.
    col="darkgray")
V(g2)$color = cores
plot(g2)
```





Clusters & Networks

Note that clustering is related (heuristically) on cycles among 3 vertices (triangles).
$\tau_\Delta(v)$ is the number of triangles in G that contain v
$\tau_\wedge(v)$ is the number of connected triplets in G with two edges incident to v



One can define an aggregate local clustering coefficient,

$$\text{cluster}(v) = \frac{\tau_\Delta(v)}{\tau_\wedge(v)}$$

Let V_2 denote the subset $\{v \in V, d_v \geq 2\}$, then set

$$\text{cluster}(G) = \frac{1}{n_{V_2}} \sum_{v \in V_2} \text{cluster}(v)$$

the clustering coefficient, and a transitivity coefficient

$$\text{transitivity}(G) = \frac{\sum_{v \in V_2} \text{cluster}(v) \cdot \tau_\wedge(v)}{\sum_{v \in V_2} \tau_\wedge(v)}$$

```
> transitivity(g1)
[1] 0.45
> transitivity(g2)
[1] 0.3673469
```

One can also look for network partitioning, i.e. community detection

- vertices in a group should be well connected among themselves
- vertices in different groups should be well separated

Hierarchical clustering and spectral partitioning

Clusters & Networks

We seek a segmentation in natural subsets, i.e. a partition of the set of vertices.
A subset of vertices is said to be cohesive if

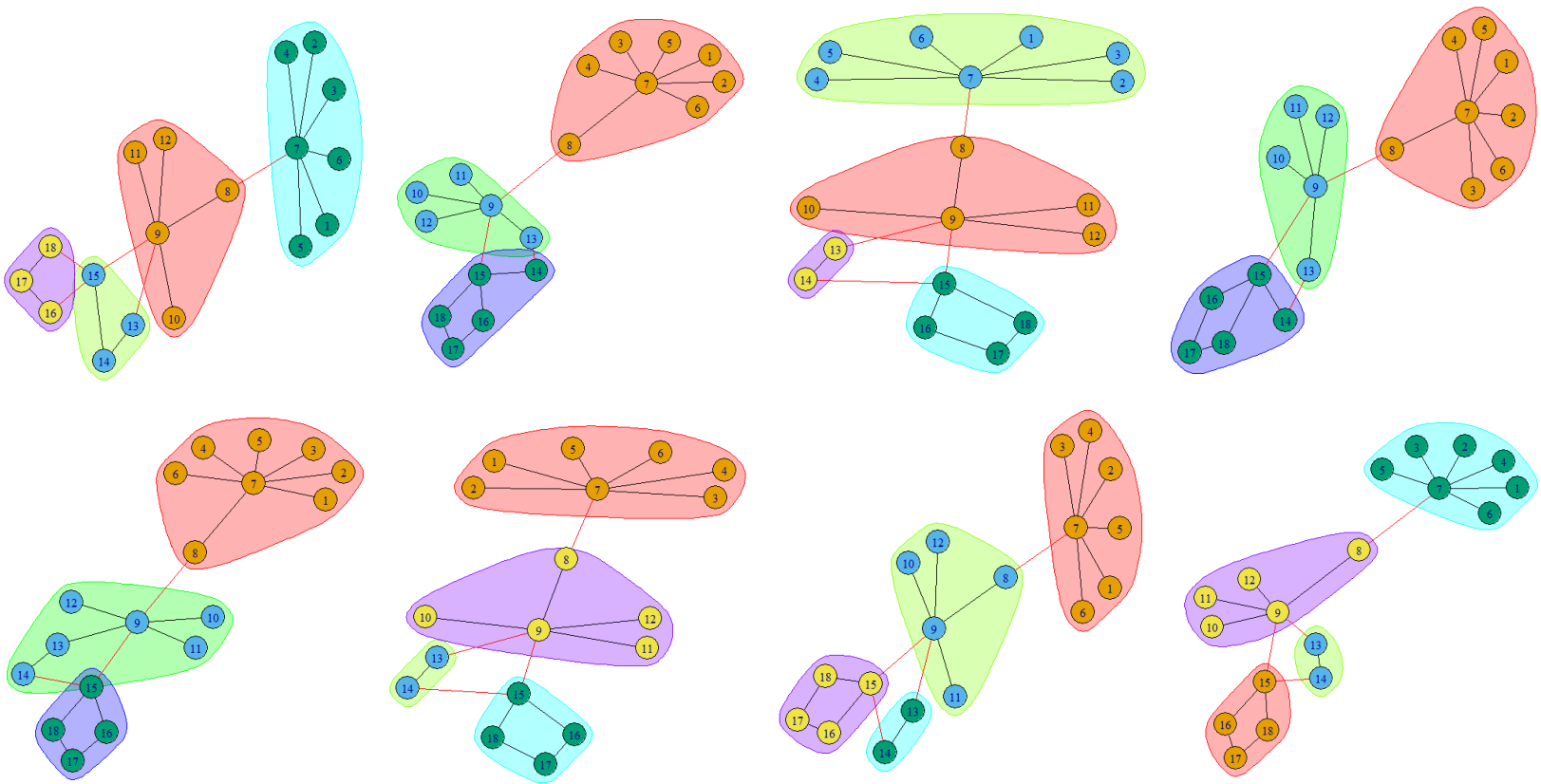
- vertices are well connected among themselves (within a subset)
- vertices are well separated from the remaining vertices (between subsets)

i.e. given a partition $\{C_1, \cdots, C_k\}$

- between: $E(C_k, C_{k'})$ set of connecting vertices in C_k to vertices in $C_{k'}$ is relatively small in size compared to
- within $E(C_k)$ and $E(C_{k'})$, edges connecting vertices within subsets

Application community detection in social media

Clusters & Networks



Clusters & Networks

Hierarchical clustering (greedy approach, since we iteratively modify partitions)
Can be either forward or backward,

- successive coarsening of partition, by merging
- successive refinement of partition, by splitting

with two extramal partitions, V and $\{\{v_1\}, \cdots, \{v_k\}\}$.
Dendogram-based representation.
(1) Need a cost measure to quantify (dis)similarity between pairs of vertices.
(2) Need to aggregate (dis)similarities on subsets C_k and $C_{k'}$

- classical single linkage
- common linkage

- Ward's method

Euclidean Similarity

$$ds_{i,j} = \sqrt{\sum_{k \neq i,j} (A_{i,k} - A_{j,k})^2}$$

Neighborhood-based Similarity

$$ds_{i,j} = \frac{\Delta_{i,j}}{d_{N_k} + d_{N_k}}$$

where $\Delta_{i,j}$ is the number of neighbors shared by i and j

Clusters & Networks

One can use modularity, from Newman (2004, Modularity and community structure in networks)
Heuristically, we choose a partition C that deviates most in modularity from what is expected at random from a collection of graphs with the same degree distribution as G.
Consider a partition $\mathcal{C} = \{C_1, \cdots, C_k\}$. Given two clusters i and j, define $f_{i,j}(\mathcal{C})$ the fraction of edges connecting vectives in C_i to vertices in C_j.
Modularity of $\mathcal{C}$ is

$$\text{mod}(\mathcal{C}) = \sum_k [f_{k,k}(\mathcal{C}) - f^\star_{k,k}]$$

where $f^\star_{k,k}$ is the expected value of $f_{k,k}$ under some random edge assignment.

Clusters & Networks

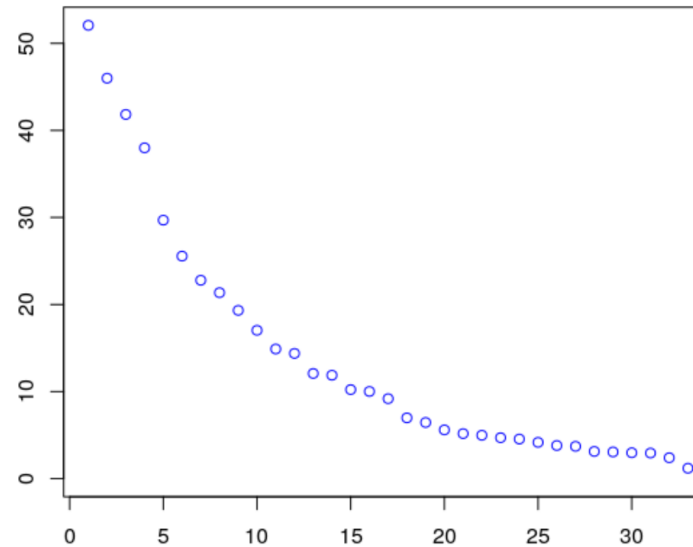
One can also consider spectral partitioning based on spectral (eigenvalues) of a matrix related to network G, e.g. adjacency matrix $\boldsymbol{A}$ or Laplacian matrix $\boldsymbol{L}$.
On $\boldsymbol{A}$, calculate n_V eigenvalues and eigenvectors,

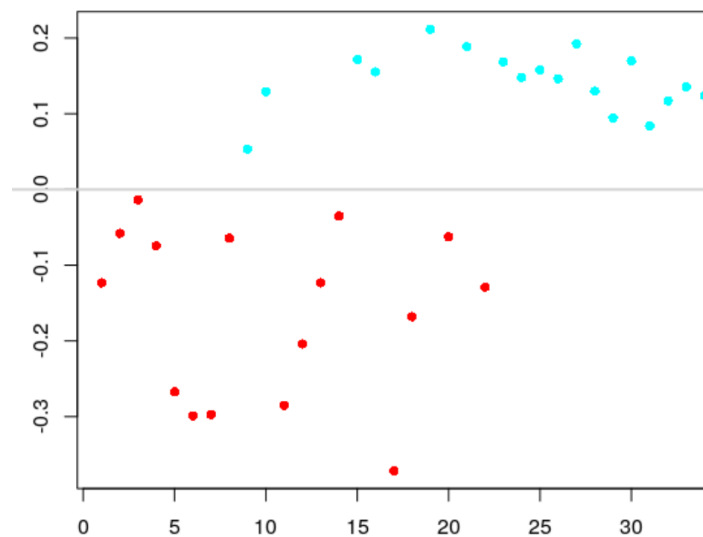
$$(\lambda_i, \boldsymbol{u}_i), \text{ with } \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n_V}.$$

Start with the largest (absolute value) eigenvalues,
vertices corresponding to particularly large positive or negative entries, in conjunction with their immediate neighbors, are declared to be a cluster
On $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, recall that $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{L}\boldsymbol{x} = \sum_{i,j}(x_i - x_j)^2$
The closer $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{L}\boldsymbol{x}$ is to zero, the more similar the elements of $\boldsymbol{x}$ at adjacent vertices in V. So the Laplacian provides some sense of the 'smoothness' of functions $\boldsymbol{x}$ on the network.

Networks with R

```
> k.lap <- graph.laplacian(karate)
> eigk <- eigen(k.lap)
> plot(eigk$values, col="blue")
> f.vec <- eigk$vectors[, 33]
>
> faction <- get.vertex.attribute(
    karate, "Faction")
> f.colors <- as.character(length(
    faction))
> f.colors[faction == 1] <- "red"
> f.colors[faction == 2] <- "cyan"
> plot(f.vec, pch=16, col=f.colors)
> abline(h=0, 0)
```





Walktrap

This algorithm finds densely connected subgraphs by performing random walks. The idea is that random walks will tend to stay inside communities instead of jumping to other communities.
see Pons & Latapy (2005)

Edge Betweenness

This algorithm is the Newman & Girvan (2003) algorithm. It is a divisive algorithm where at each step the edge with the highest betweenness is removed from the graph. For each division you can compute the modularity of the graph. At the end, choose to cut the dendrogram where the process gives you the highest value of modularity.

Clusters & Networks

Fastgreedy

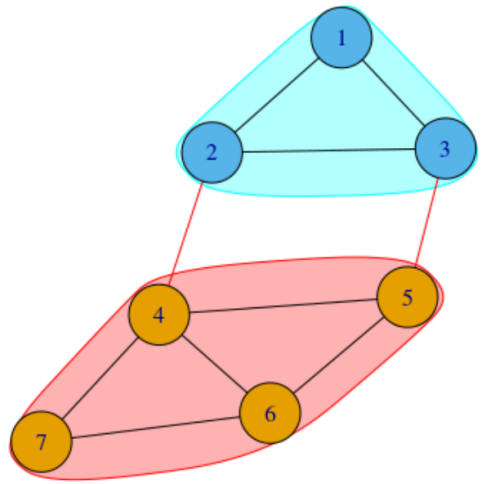
This algorithm is the Clauset, Newman & Moore (2004) algorithm. In this case the algorithm is agglomerative. At each step two groups merge. The merging is decided by optimising modularity. This is a fast algorithm, but has the disadvantage of being a greedy algorithm. Thus, is might not produce the best overall community partitioning, although I find it useful and accurate.

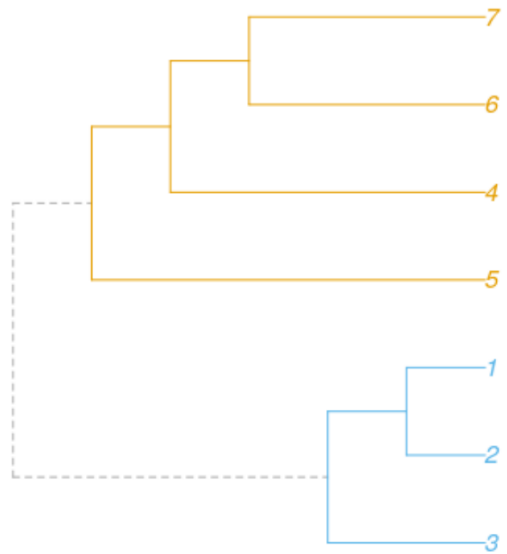
Spin-glass

This algorithm uses as spin-glass model (Sherrington & Kirkpatrick (1975)) and simulated annealing to find the communities inside a network. See also Reichardt & Bornholdt (2006)

Networks with R

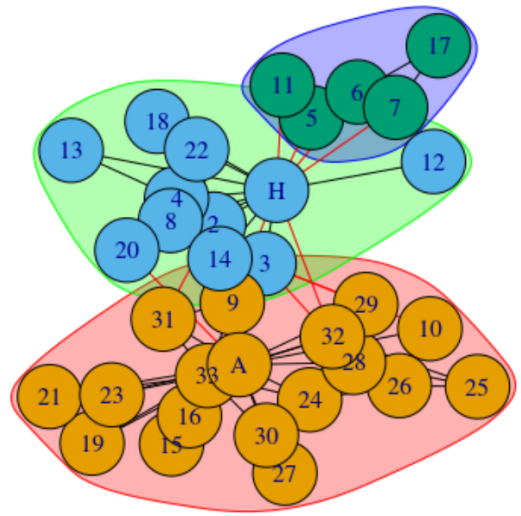
```
> kc = fastgreedy.community(g1)
> length(kc)
[1] 2
> sizes(kc)
Community sizes
1 2
4 3
> membership(kc)
1 2 3 4 5 6 7
2 2 2 1 1 1 1
> plot(kc,g1)
> dendPlot(kc,mode="phylo")
```

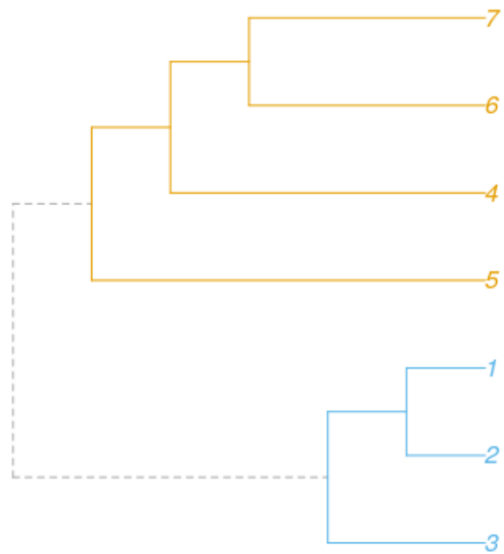




Networks with R

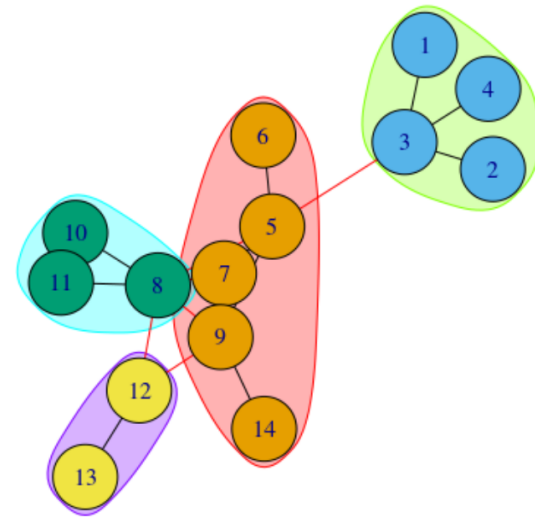
```
> kc <- fastgreedy.community(karate
    )
> length(kc)
[1] 3
> sizes(kc)
Community sizes
 1  2  3
18 11  5
> plot(kc,karate)
> dendPlot(kc,mode="phylo")
```

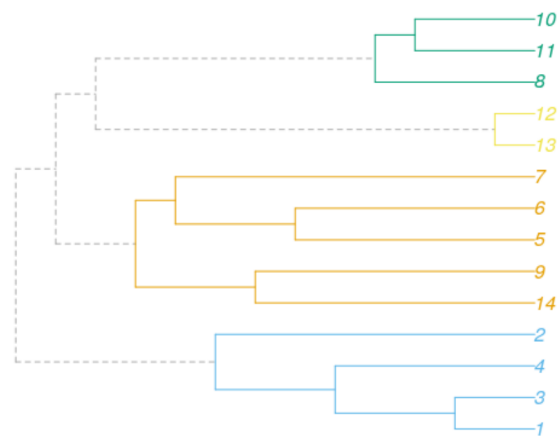




Networks with R

```
> kc <- fastgreedy.community(g2)
> length(kc)
[1] 4
> sizes(kc)
Community sizes
1 2 3 4
5 4 3 2
> membership(kc)
 1  3  2  4  5  6  7  8  9 10 11 12
    13 14
 2  2  2  2  1  1  1  3  1  3  3  4
     4  1
> plot(kc,g2)
> dendPlot(kc,mode="phylo")
```





Networks & Small World

"*I read somewhere that everybody on this planet is separated by only six other people. Six degrees of separation. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice. Fill in the names. . . . How every person is a new door, opening up into other worlds. Six degrees of separation between me and everyone else on this planet. But to find the right six people...*" Guare (1990, Six Degrees of Separation)

The small world property is mathematically intuitive: if the number of vertices within a given distance of a specific node growth exponentially with the distance, then the average path length increases a $\log n_V$. Formally, we talk here about the average path length

$$\bar{\ell} = \binom{n_V}{2}^{-1} \sum_{u \neq v} d(u, v) = O(\log n_V)$$

if $d_v = d$, if we reach everyone after k hops $n_V \sim d^k$ i.e. $k = O(\log n_V)$

Networks & Small World

"*Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of three and a half other people. The average distance we observe is 4.57, corresponding to 3.57 intermediaries or 'degrees of separation*"', see Bharat *et.* (1990, Three and a half degrees of separation)

An alternative is based on the harmonic mean, related to graph efficiency

$$\ell^{-1} = \frac{2}{n_V(n_V-1)} \sum_{i>j} d_{i,j}^{-1}$$

Networks & Homophily

As seen in the introduction, people have a strong tendancy to associate with equals, i.e. homophily or assortative mixing.
Consider some categories (partition) $\{C_1, \cdots, C_k\}$.
Let f_{ij} denote the fraction of edges joining vertices of categories C_i and C_j. Define

$$f_{i\cdot} = \sum_j f_{ij} \text{ and } f_{\cdot j} = \sum_i f_{ij}$$

The assortativity coefficient is

$$R_A = \frac{\sum_i f_{ii} - \sum_i f_{i\cdot} f_{\cdot i}}{1 - \sum_i f_{i\cdot} f_{\cdot i}}$$

see Newman (2003, Mixing patterns in networks)
Here $f_{i\cdot} f_{\cdot i}$ the the expected fraction of edges joining nodes in C_i.

Networks & Homophily

Perfect homophily means $R_A = R_A^+ = 1$, the upper bound while the lowe bound is

$$R_A^- = -\frac{\sum_i f_{i\cdot} f_{\cdot i}}{1 - \sum_i f_{i\cdot} f_{\cdot i}} > -1$$

An alternative is based on

$$R = \frac{\text{trace}(\boldsymbol{f}) - \|\boldsymbol{f}\|^2}{1 - \|\boldsymbol{f}\|^2}$$

Networks & Homophily

Students sitting in dining halls (men/women, and ethnic groups)

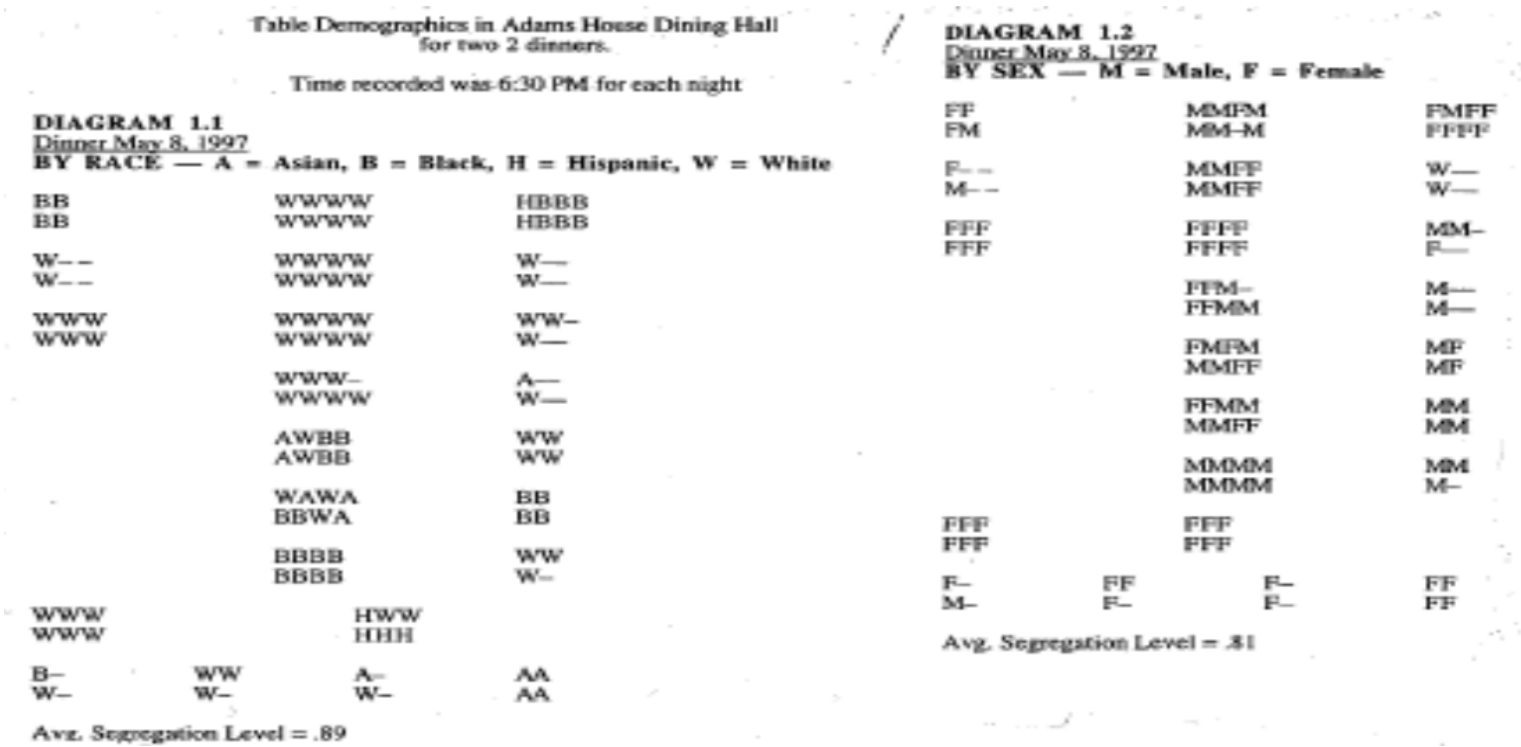


Table Demographics in Adams House Dining Hall
for two 2 dinners.

Time recorded was 6:30 PM for each night

DIAGRAM 1.1
Dinner May 8, 1997
BY RACE — A = Asian, B = Black, H = Hispanic, W = White

BB
BB

W– –
W– –

WWW
WWW

WWWW
WWWW

WWWW
WWWW

WWWW
WWWW

WWW–
WWWW

AWBB
AWBB

WAWA
BBWA

BBBB
BBBB

HBBB
HBBB

W—
W—

WW–
W—

A—
W—

WW
WW

BB
BB

WW
W–

WWW
WWW

HWW
HHH

B–
W–

WW
W–

A–
W–

AA
AA

Avg. Segregation Level = .89

DIAGRAM 1.2
Dinner May 8, 1997
BY SEX — M = Male, F = Female

FF
FM

F– –
M– –

FFF
FFF

MMFM
MM–M

MMFF
MMFF

FFFF
FFFF

FFM–
FFMM

FMFM
MMFF

FFMM
MMFF

MMMM
MMMM

MMFF
FFF

FMFF
FFFF

W—
W—

MM–
F—

M—
M—

MF
MF

MM
MM

MM
M–

FFF
FFF

F–
M–

FF
F–

F–
F–

FF
FF

Avg. Segregation Level = .81

Population divided in two types, each individuals wants at least a proportion $t \in [0, 1]$ of his neighbors with his own type (say 50%).

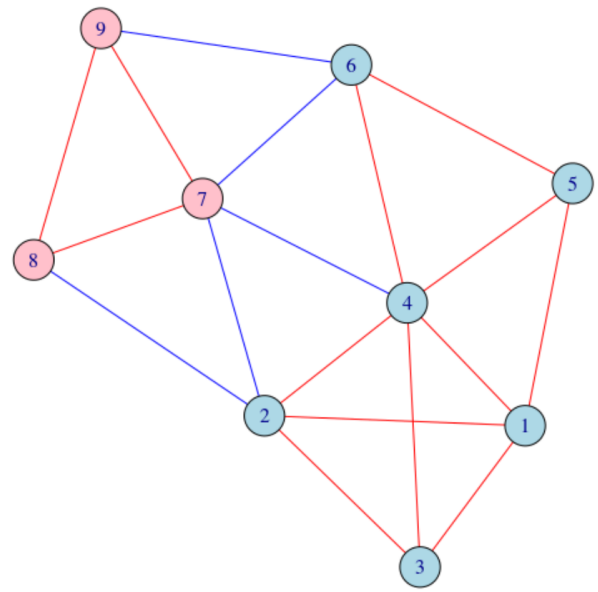
Networks & Homophily

Unsatisfied individuals move.
There is an externality from the move, with a cascading effect.
Need to define a neihbourhood, a preference function, a moving rule (consider either exchanges or empty space).

Schelling model generates segregation beyond what people really want, see Chapter 4 in Easley & Kleinberg (2010).
Fixed and unchanging characteristics (gender, ethnicity) can be highly correlated with mutable chaacteristics (location).

Networks & Homophily

Consider a netword (V, E), with n edges. Let x_e be an indicator variable if e is a cross edge (in blue on the graph).



$$\overline{x} = \frac{1}{n_E} \sum_{e \in E} x_e = \frac{5}{18}$$

Let p denote the proportion of one type (say pink). Here $p = 1/3$.
In a random graph, X_e's are i.id. random variables $\mathcal{B}(2p(1-p))$.
Use a standard t-test: $H_0 : \overline{X} > 2p(1-p)$ against
$H_0 : \overline{X} < 2p(1-p)$

```
> p=2/3
> ne=length(E(g))
> nce=5
> (x=c(rep(1,nce),rep(0,ne-nce)))
 [1] 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> t.test(x, alternative = "less", mu=2*p*(1-p))

  One Sample t-test

data:  x
t = -1.5342, df = 17, p-value = 0.07169
alternative hypothesis: true mean is less than
    0.4444444
95 percent confidence interval:
      -Inf 0.4667556
sample estimates:
mean of x
0.2777778
```