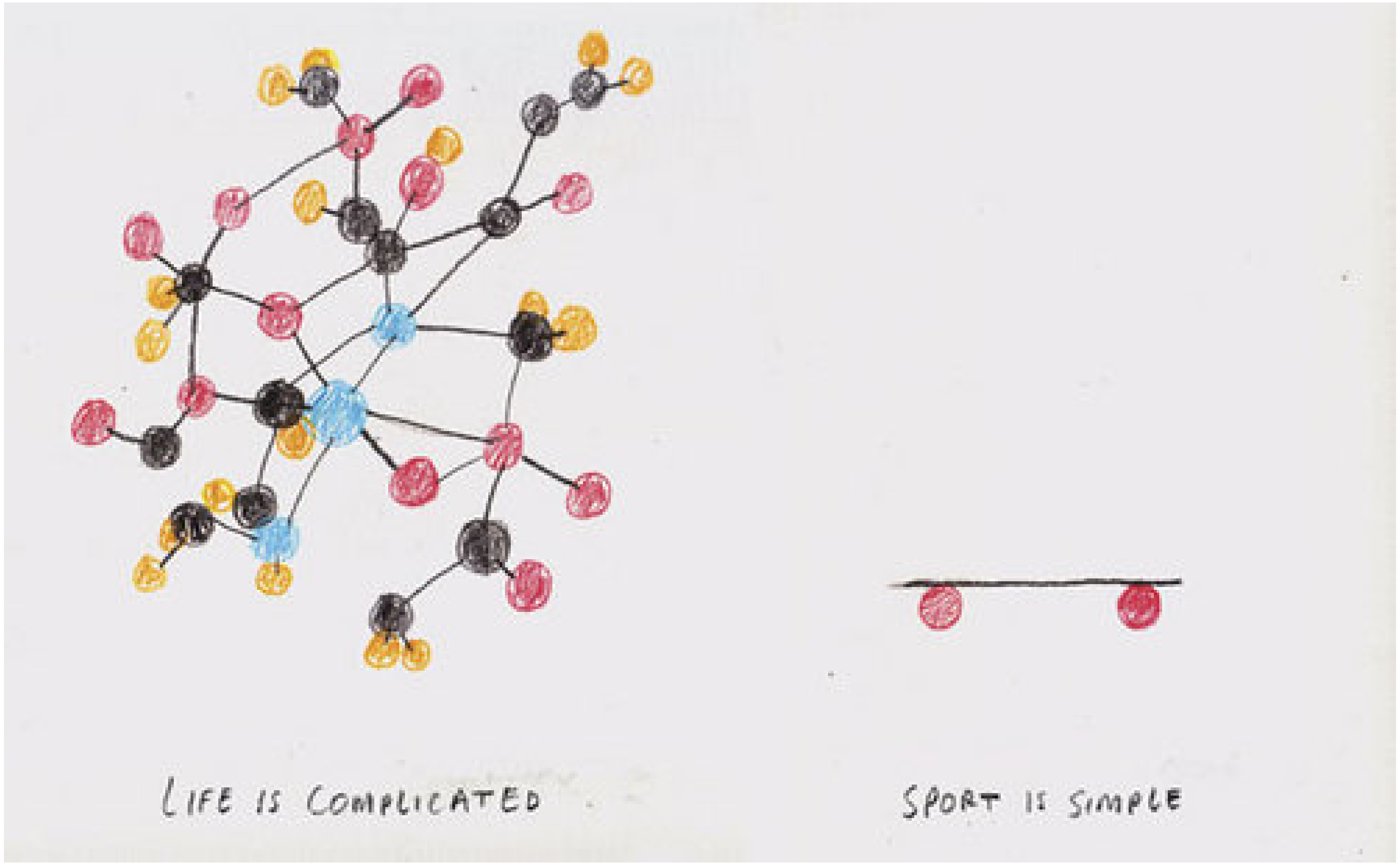
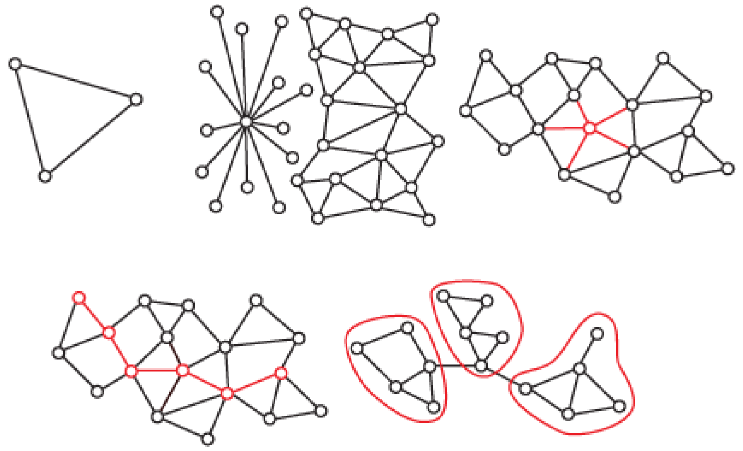

LIFE IS COMPLICATED
SPORT IS SIMPLE

Arthur Charpentier

arthur.charpentier@univ-rennes1.fr

https://freakonometrics.github.io/

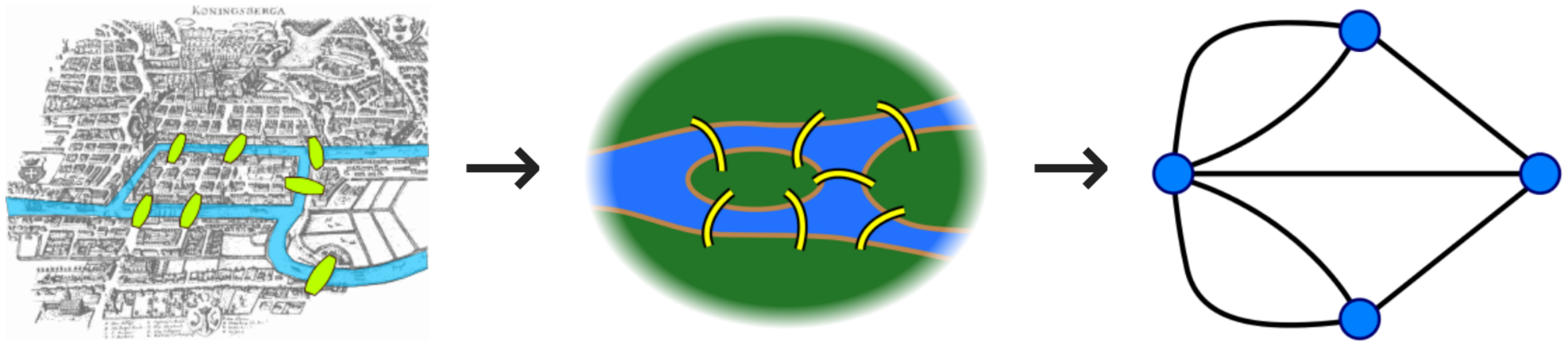
Université Rennes 1, 2017



Introduction to Graphs & Networks

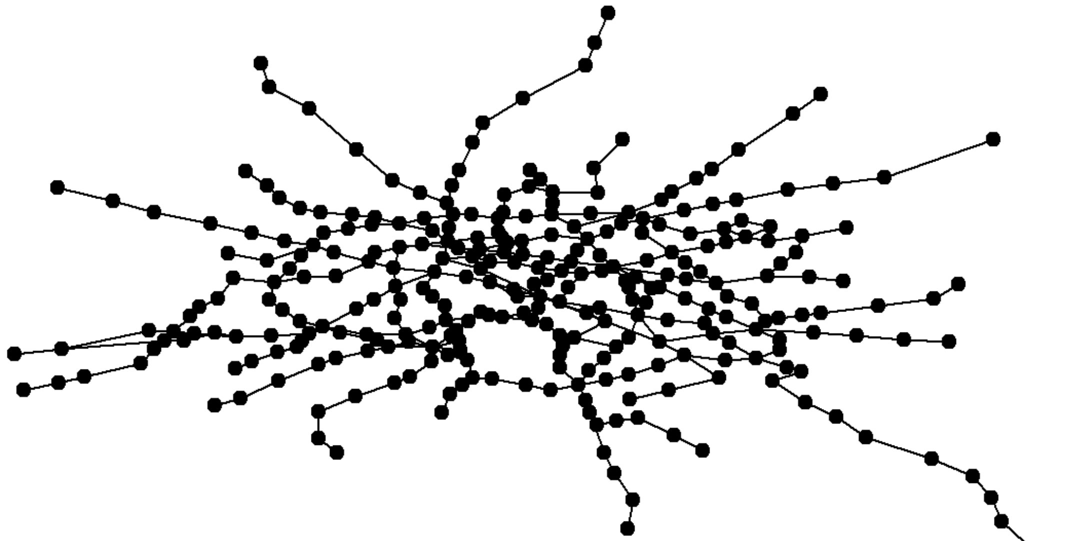
Example: Seven Köningsberg Bridges

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

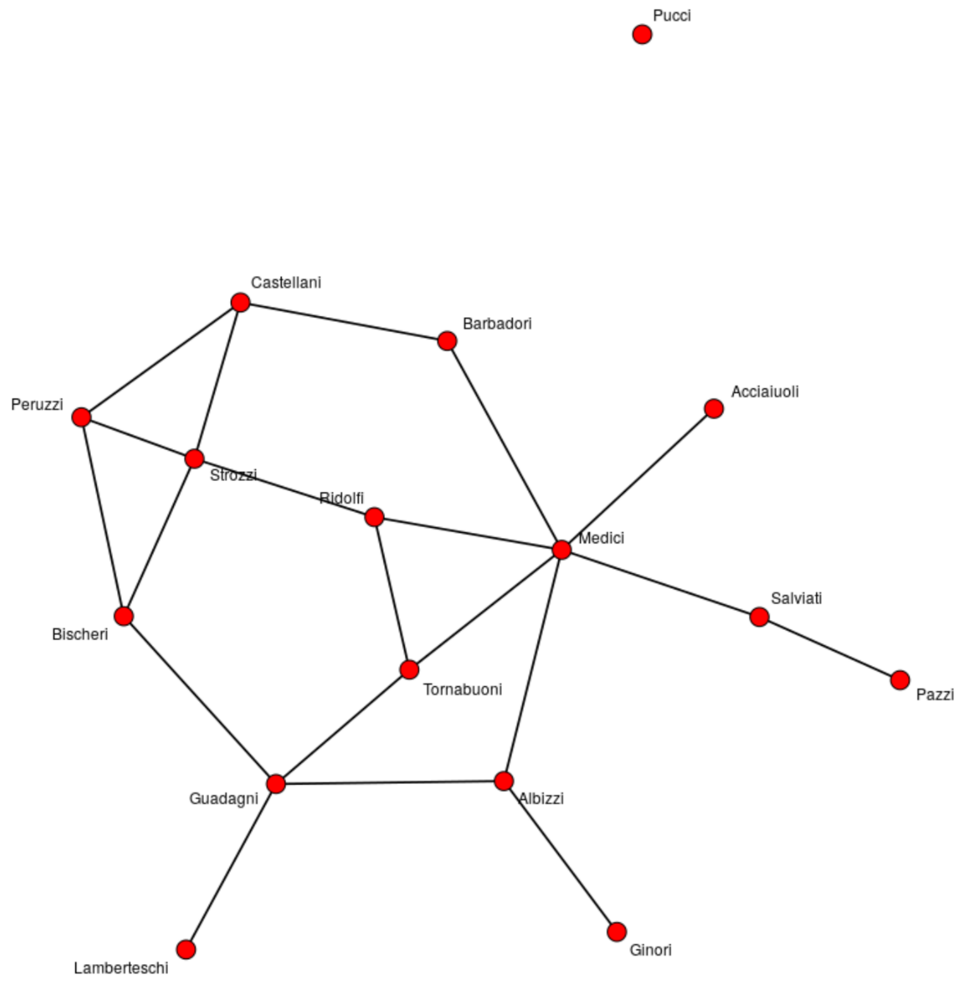


see also use of GPS to find the shortest path home (discussed in the course on transportation).

Example: Metro in Paris (2017)



Example: Florentine Marriages (1430's)



References

Briatte, F. Awesome Network Analysis, https://github.com/briatte/

Lada Adamic's Lecture notes

Easley, D. & Kleinberg, J. (2010) Networks, Crowds, and Markets: Reasoning About a Highly Connected World Cambridge University Press

Harary , F. (1969) Graph Theory

Jackson, M. (2008). Social and Economic Networks

Kolaczyk, E.D. (2009). Statistical Analysis of Network Data: Methods and Models. Springer Verlag.

Kolaczyk, E.D. & Csárdi, G. (2010). Statistical Analysis of Network Data with R. Springer Verlag.

References

Newman, M. (2004) The structure and function of complex networks

Newman, M. (2010) Networks : An Introduction. Oxford University Press

Wasserman, S. & Faust, K. (1994) Social Network Analysis : Methods and Applications

West, D.B. (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 (V, E)

Examples: G = Internet, V = computers, E = IP network adjacency

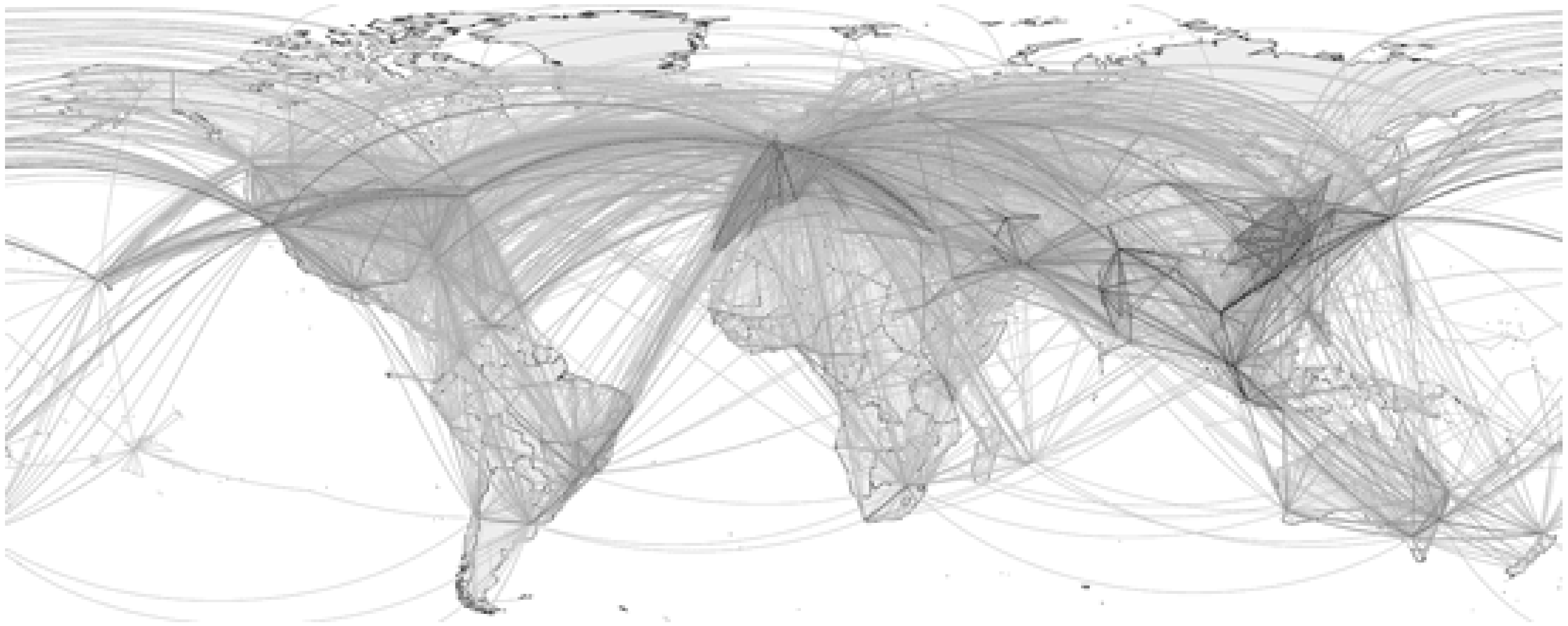
G = World Wide Web, V = web pages, E = hyperlink

G = Articles, V = authors, E = citations

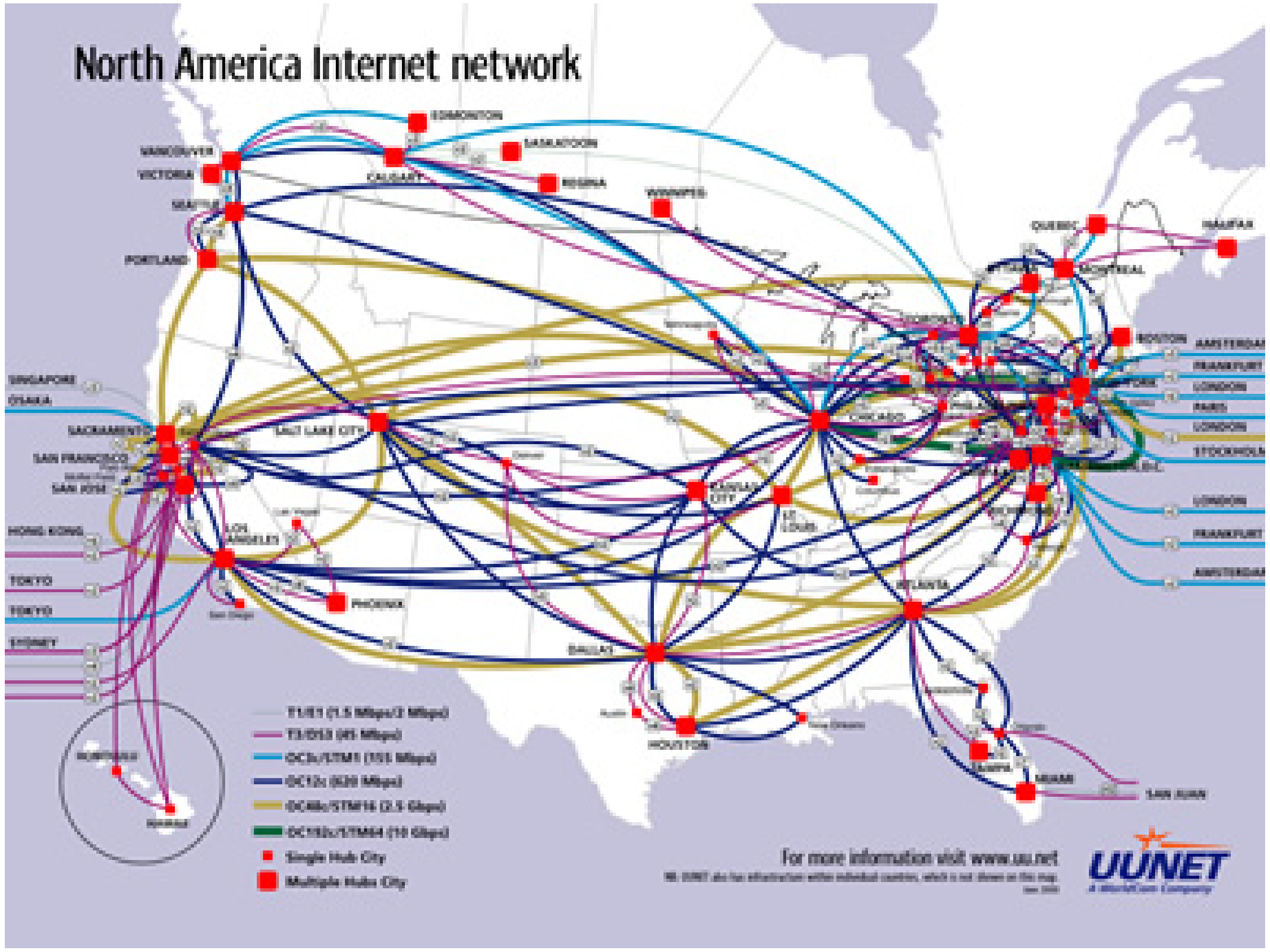
G = Friendship Network, V = persons, E = friendship

G = Airport Network, V = airports, E = non-stop flight

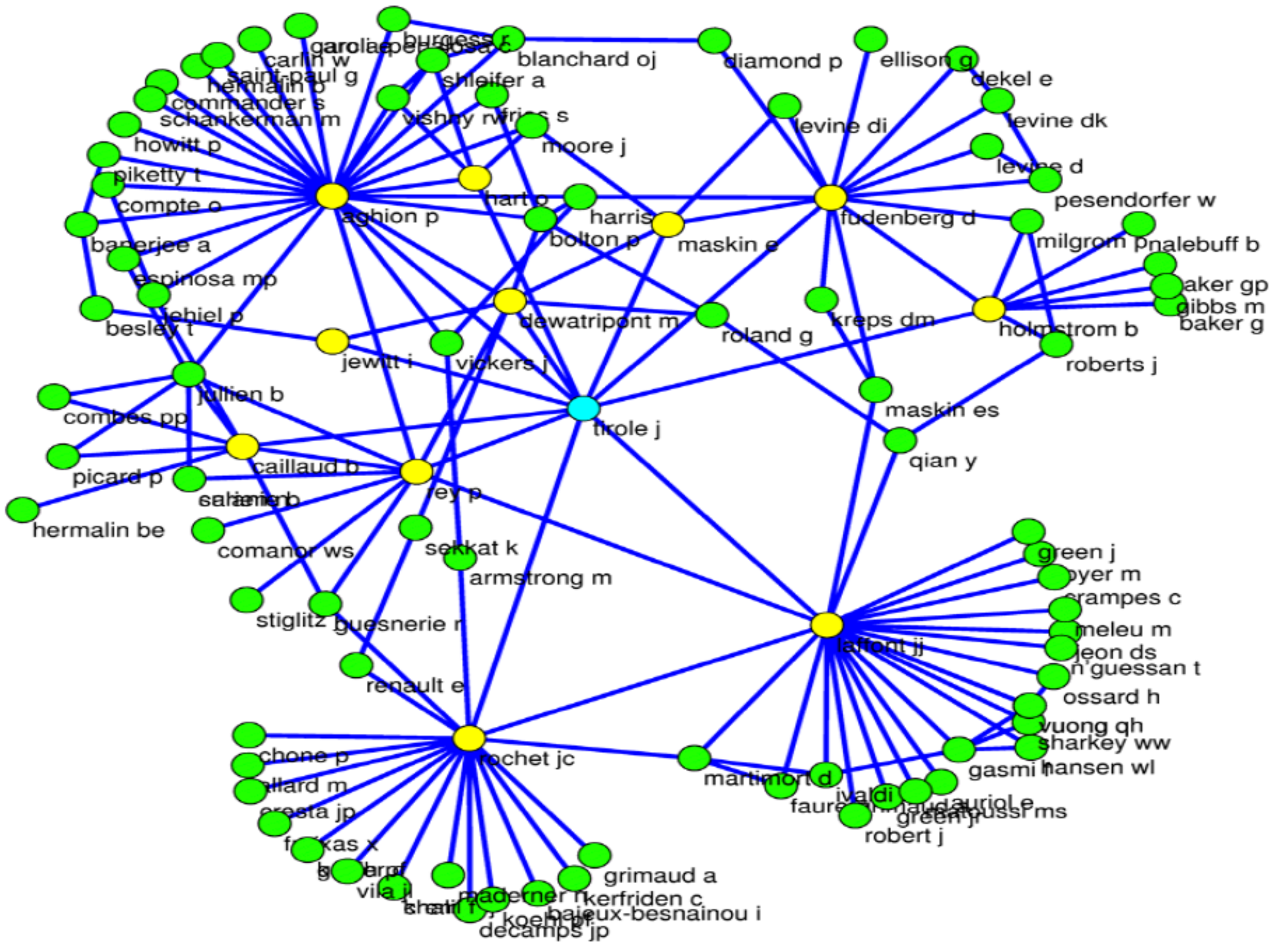
Example: Worldwide Airports (2017)



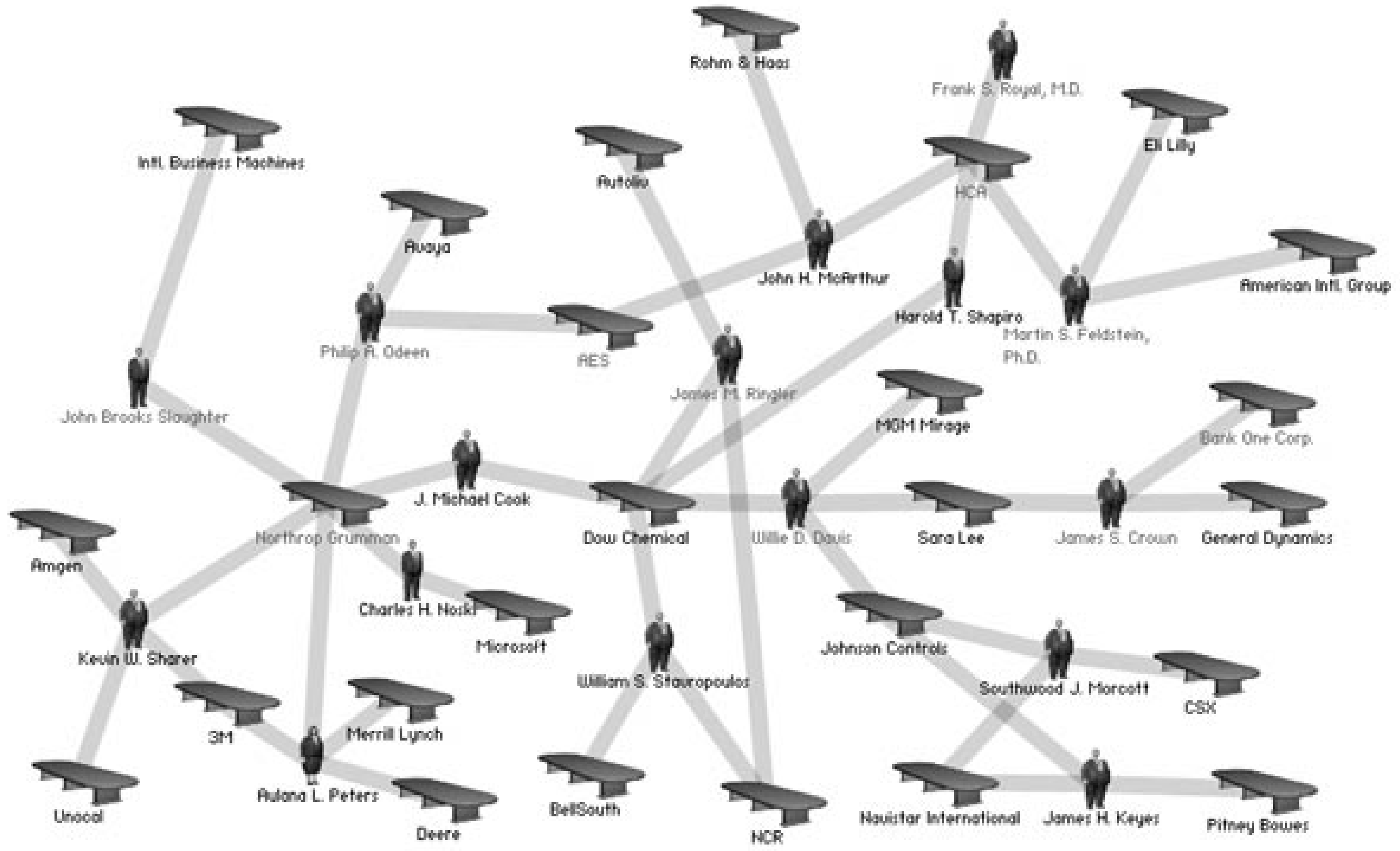
Example: Internet Network (2015)



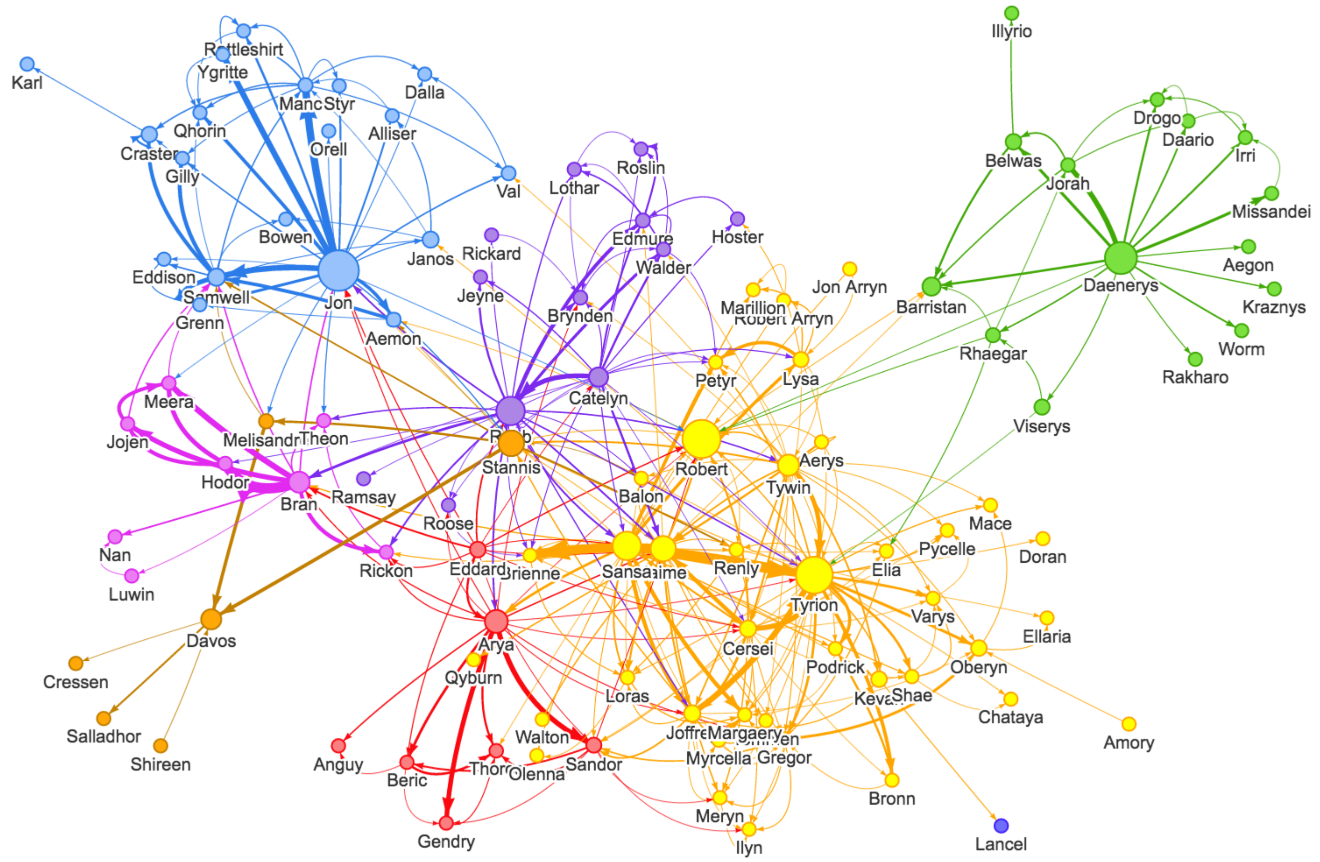
Example: Scientific Publications - Jean Tirole (1993)



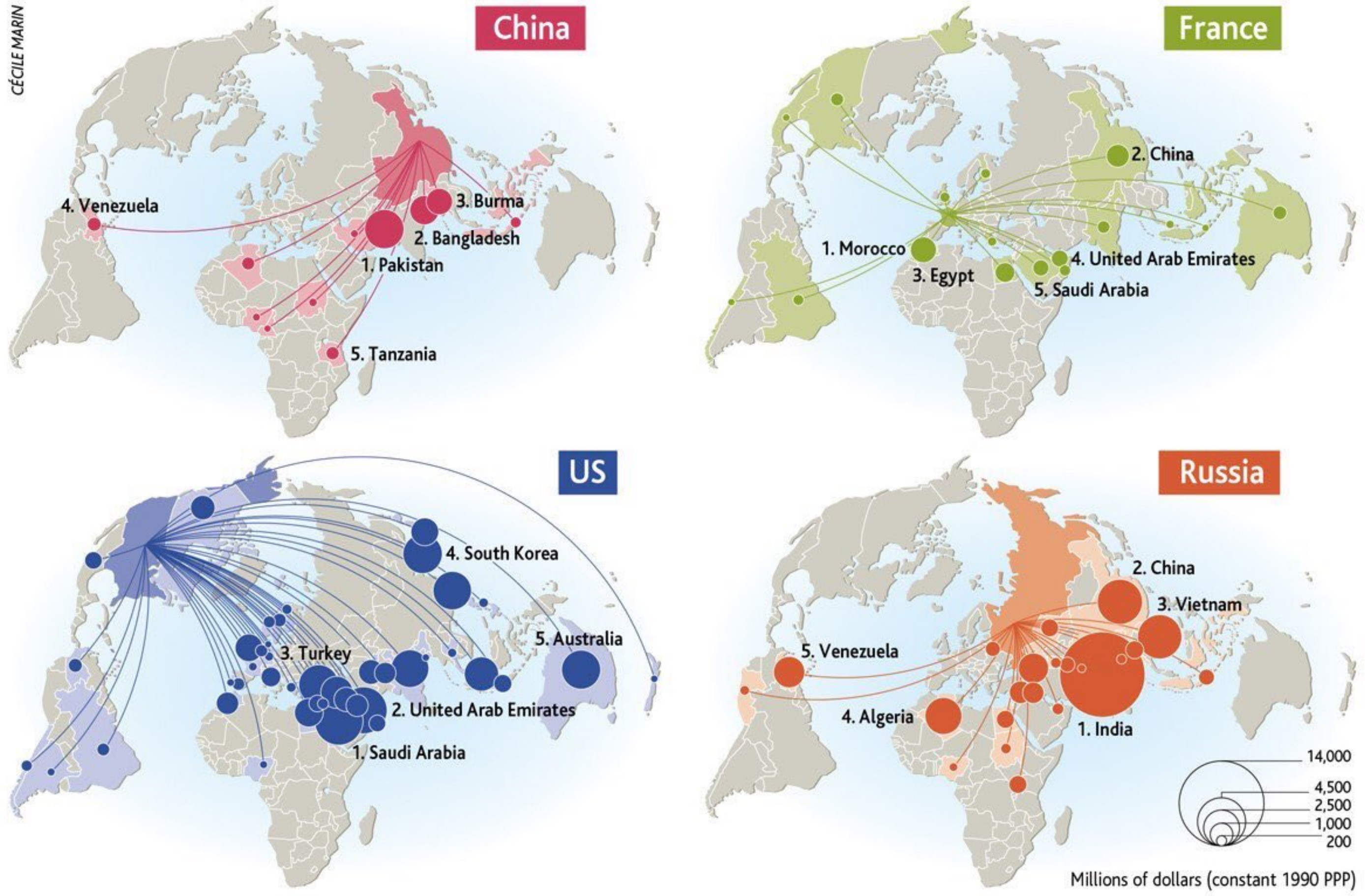
Example: CEO & Corporate Boards (2015)



Example: 'Friendship' Network (2015)



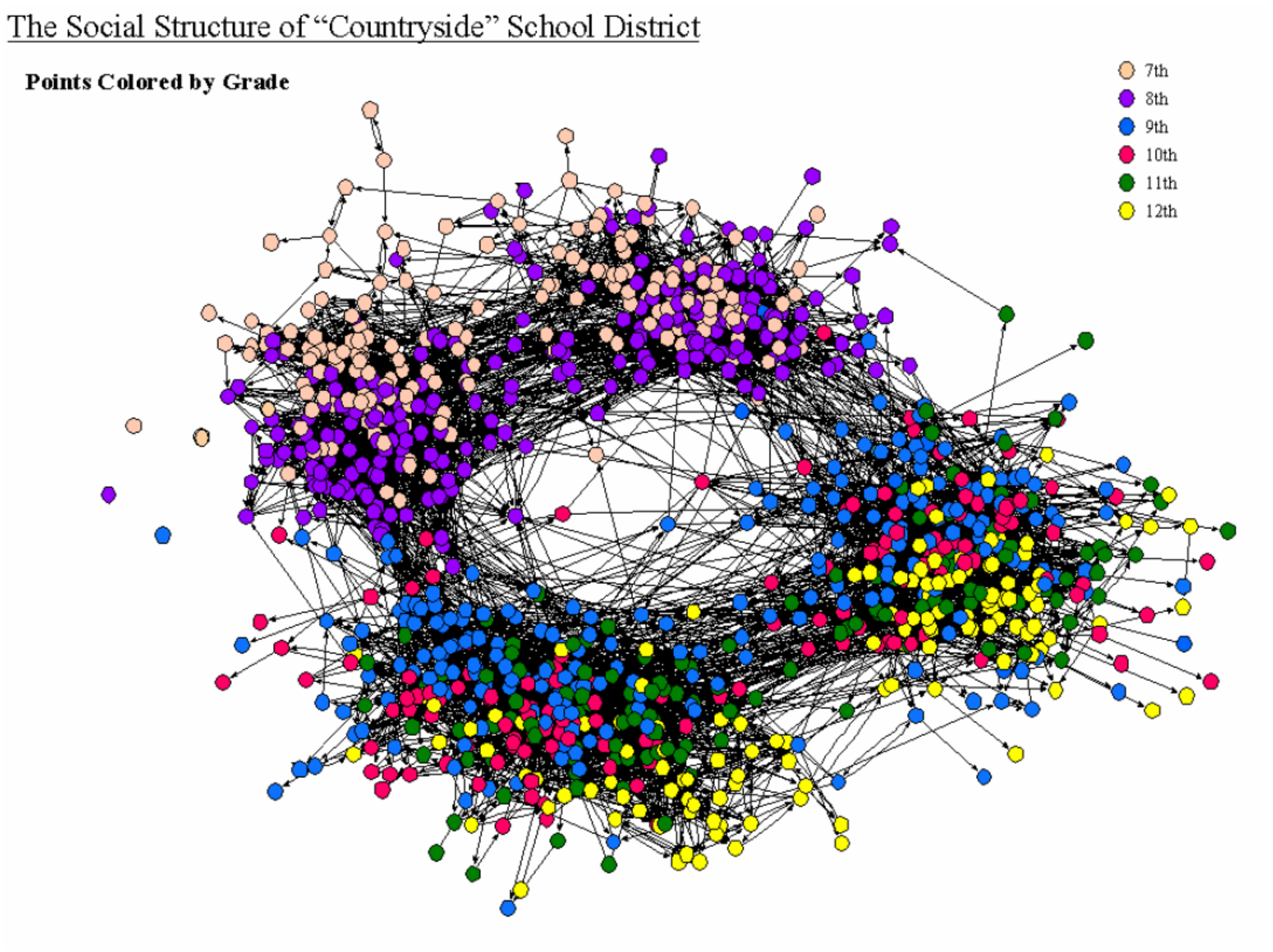
Example: Weapons Exporters (2017)



Total arms exports 2011-5 (in excess of $100m)

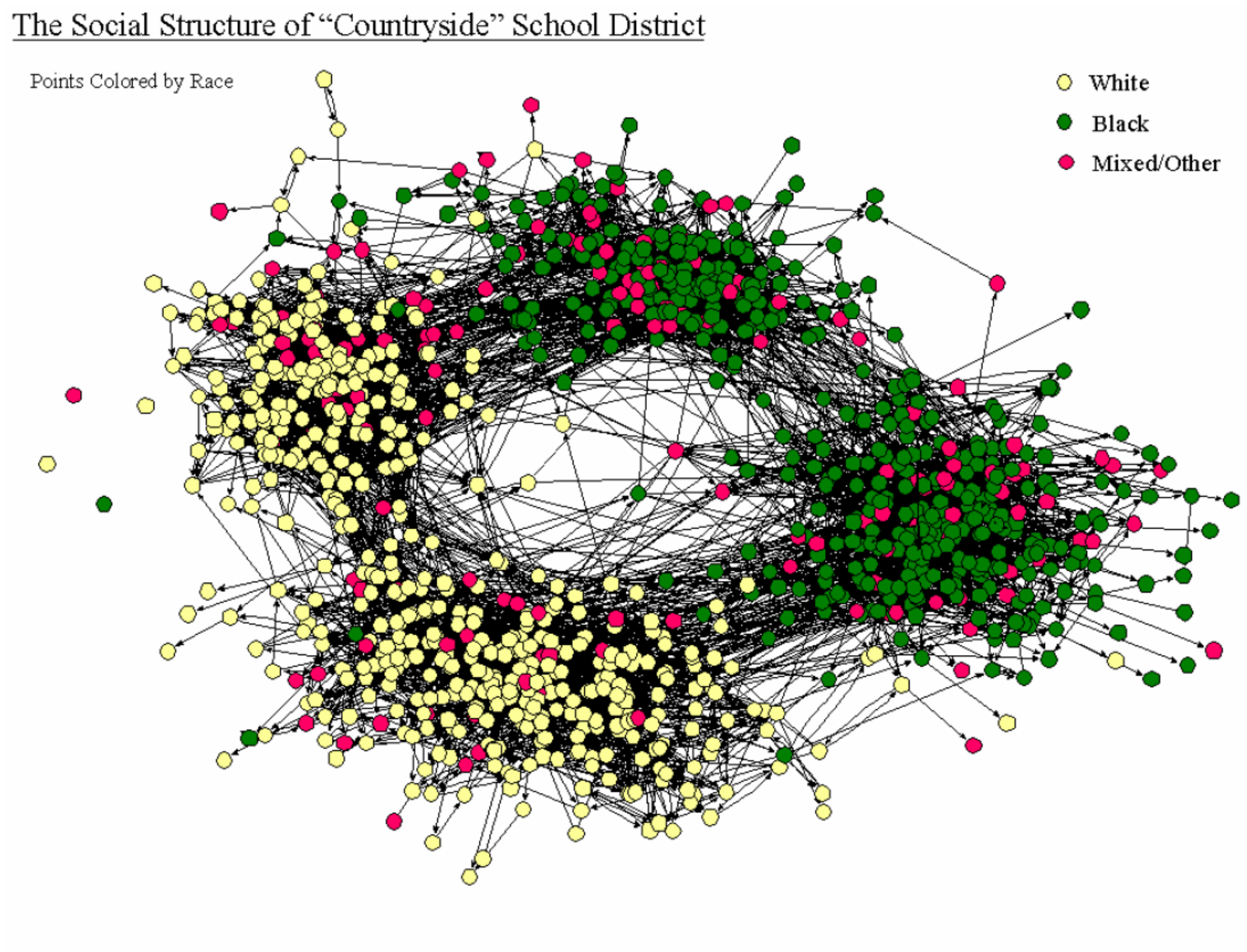
Source: Sipri, Arms Transfers Database, 2016

Example: High School Friendship (2001)



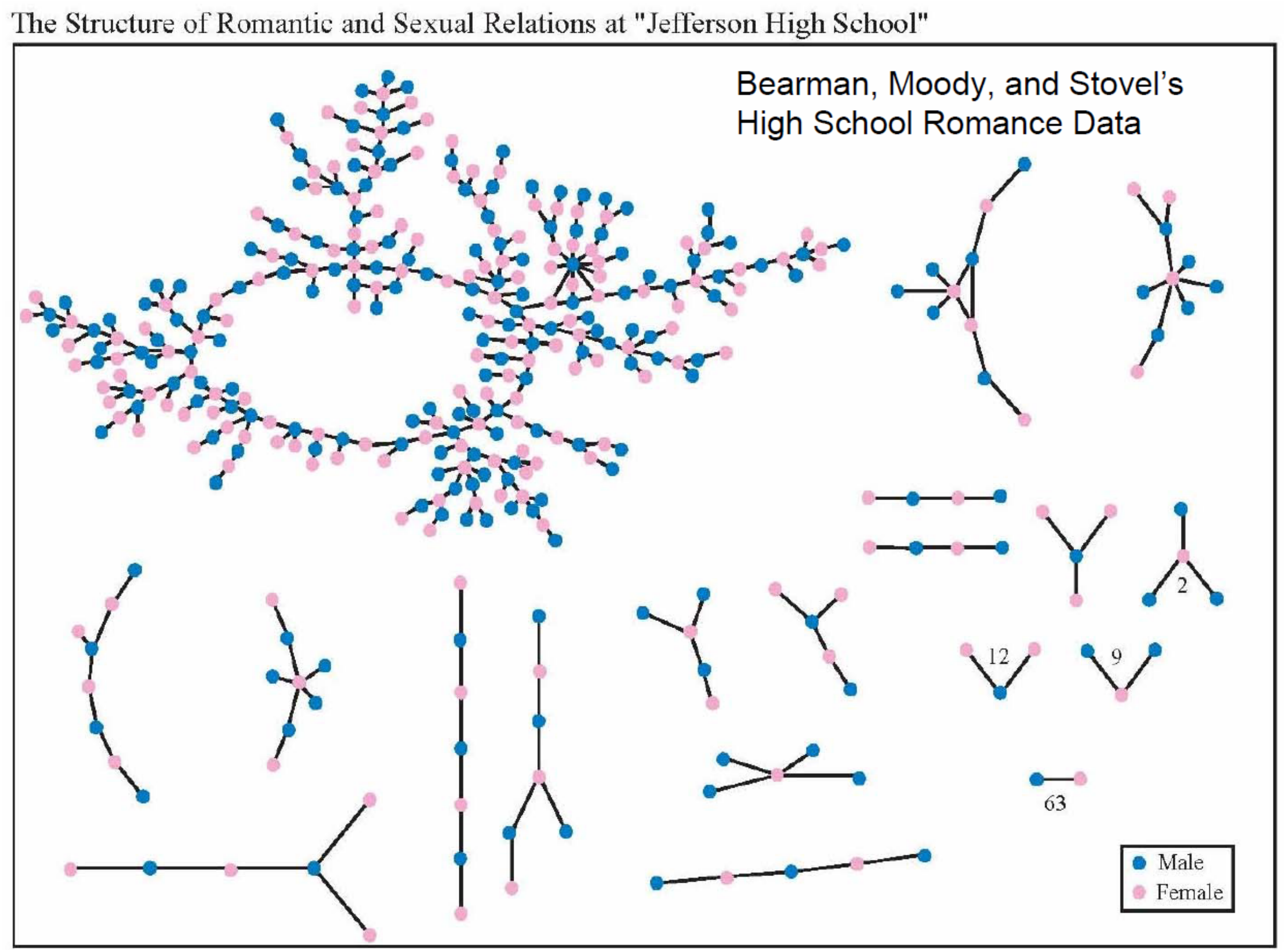
from Moody (2001) Race, School Integration and Friendship Segregation in America

Example: High School Friendship (2001)



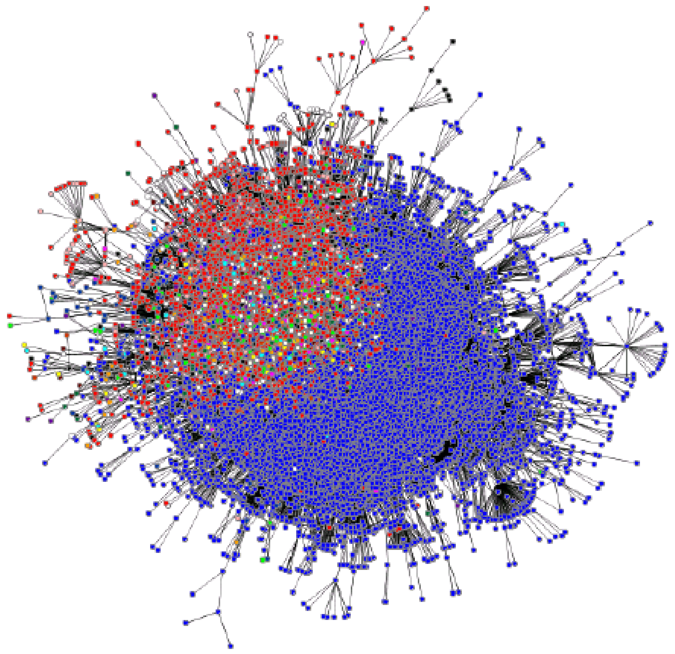
from Moody (2001) Race, School Integration and Friendship Segregation in America

Example: Romantic and Sexual Relationships


The Structure of Romantic and Sexual Relations at "Jefferson High School"
Bearman, Moody, and Stovel's High School Romance Data
2
12
9
63
Male
Female

See Bearman, Moody & Stovel (2004) Chains of Affection: The Structure of Adolescent Romantic and Sexual Networks

Example: Blog Network



Applications of Networks

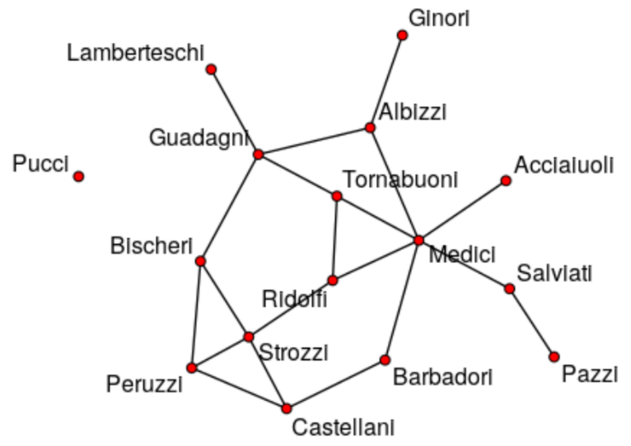
Amazon's *Customers Who Bought This Item Also Bought*, see Yang & Leskovec (2012) Defining and Evaluating Network Communities based on Ground-truth.

"*Network was collected by crawling Amazon website. It is based on Customers Who Bought This Item Also Bought feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j.*"

Ultimately yielded a graph of $n_V = 334,863$ vertices and $n_E = 925,872$ edges.

Networks Used in those Lectures

```
> 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)
```

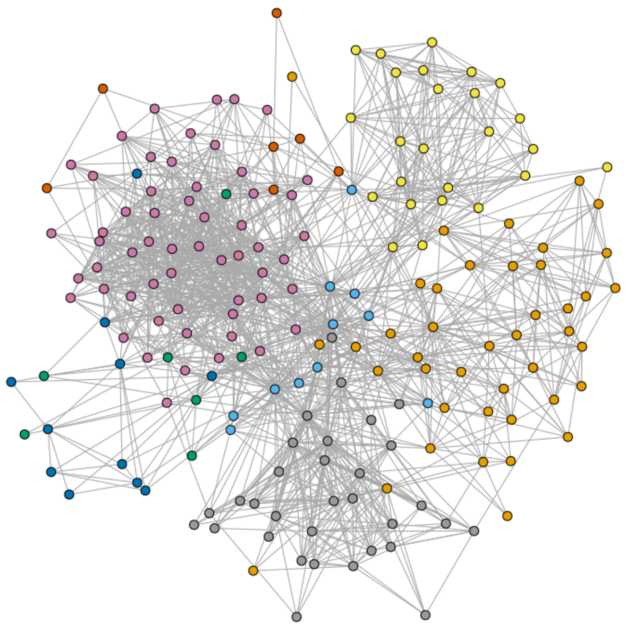


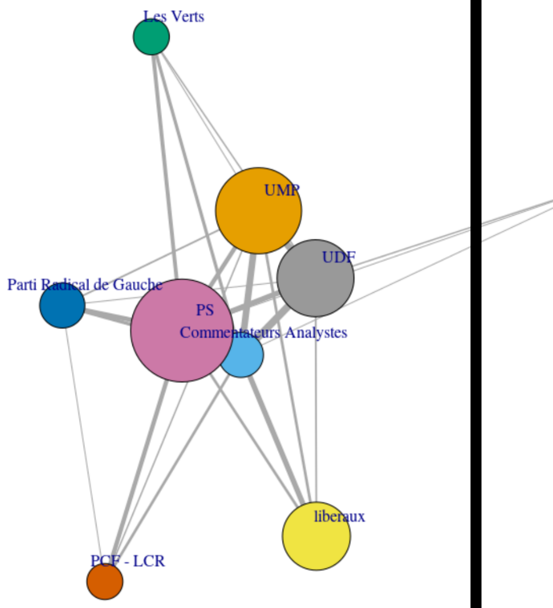
Florentine Wedding Data. $n_V = 16$ vertices (families), $n_E = 20$ edges (weddings).

From Padgett (1994). Marriage and Elite Structure in Renaissance Florence, 1282-1500

Networks Used in those Lectures

```
> library(sand)
> summary(fblog)
IGRAPH NA UN-- 192 1431 --
+ attr: name (v/c),
   PolParty (v/c)
> plot(fblog)
```

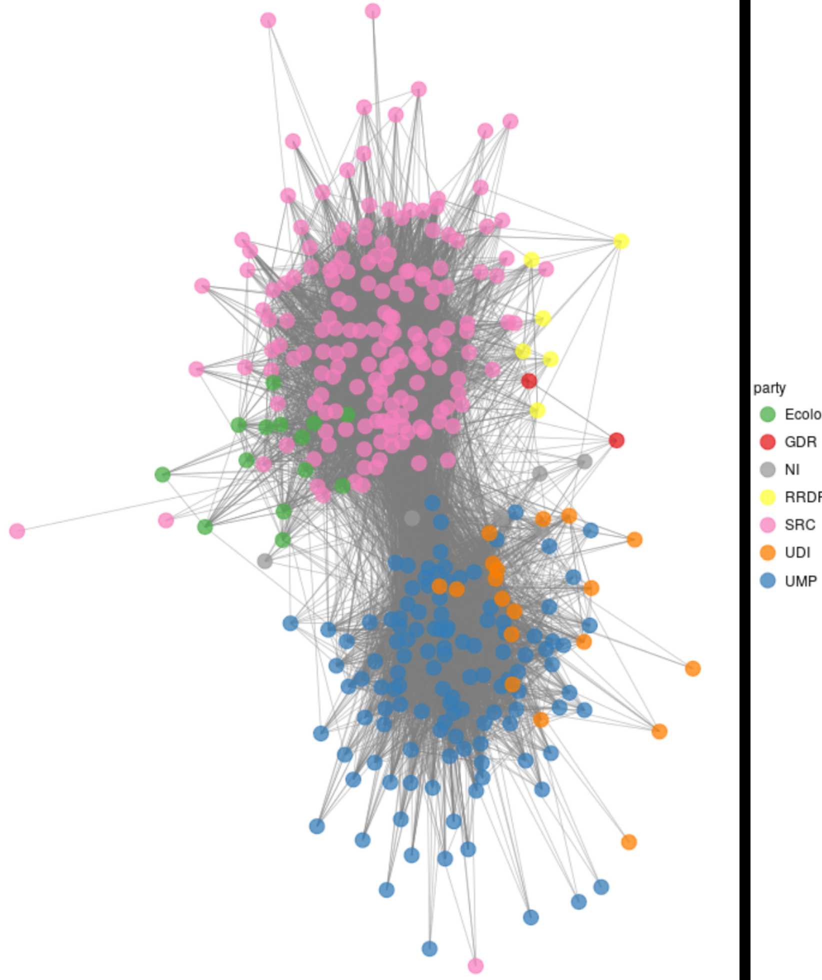




Political blogs, in France. $n_V = 192$ vertices (blogs), $n_E = 1431$ edges (hyperlinks).

Networks Used in those Lectures

```
> library(network)
> r = "https://raw.githubusercontent.com/
    briatte/ggnet/master/"
> v = read.csv(paste0(r, "inst/extdata/nodes
    .tsv"), sep = "\t")
> e = read.csv(paste0(r, "inst/extdata/
    network.tsv"), sep = "\t")
> net = network(e, directed = TRUE)
> plot(net)
```



Member of Parlement, on Twitter.

$n_V = 339$ vertices (twitter accounts - MPs),

$n_E = 12869$ edges (following).

From François Briattes's https://briatte.github.io/ggnet/.

Networks Used in those Lectures

```
> library(igraphdata)
> data(karate)
> plot(karate)
```

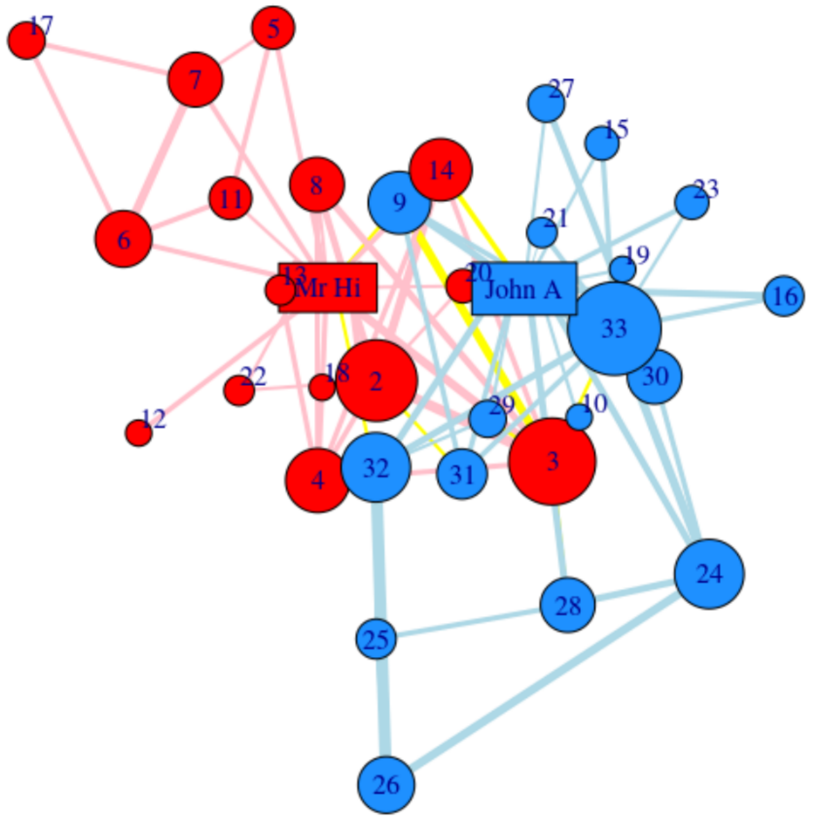
Zachary's karate club network.

$n_V = 34$ vertices (members),

$n_E = 78$ edges (social interraction).

2 specific members

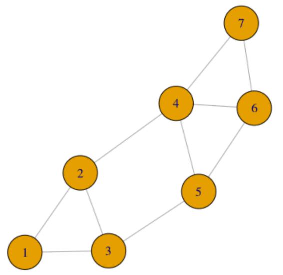
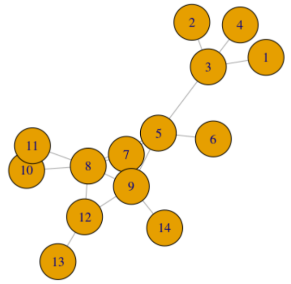
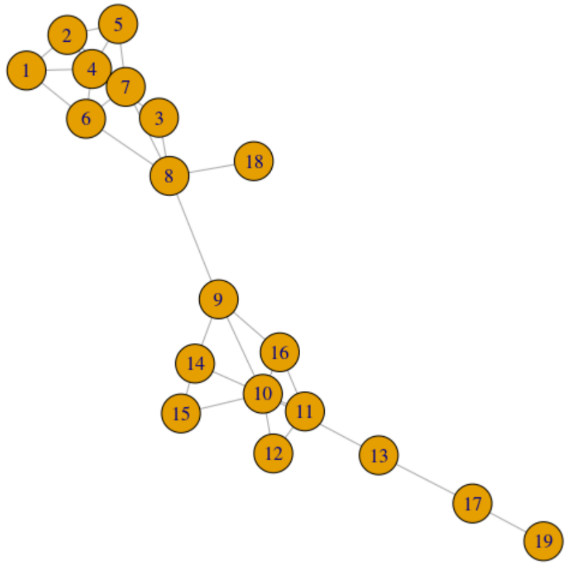
- Mr Hi
- John A



From Zachary (1977) An information flow model for conflict and fission in small groups.

Networks Used in those Lectures

```
> library(igraphdata)
> g1 = graph.formula(1-2, 1-3, 2-3, 2-4,
    3-5, 4-5, 4-6, 4-7, 5-6, 6-7)
> g2 = graph.formula(1-3, 2-3, 3-4, 3-5,
    5-6, 5-7, 7-8, 8-9, 9-5, 9-7, 5-8, 8-10,
     10-11, 11-8, 8-12, 9-12, 12-13, 9-14)
> g3 = graph.formula(1-2, 1-4, 1-6, 2-4,
    2-5, 2-7, 3-4, 3-8, 4-5, 4-7, 4-6, 5-7,
    6-7, 6-8, 7-8, 8-9, 8-18, 9-14, 9-16,
    9-10, 10-14, 10-16, 10-13, 10-12, 10-11,
     10-15, 11-12, 11-16, 13-17, 14-15,
    17-19)
> plot(g1)
> plot(g2)
> plot(g3)
```

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 $\boldsymbol{A}$

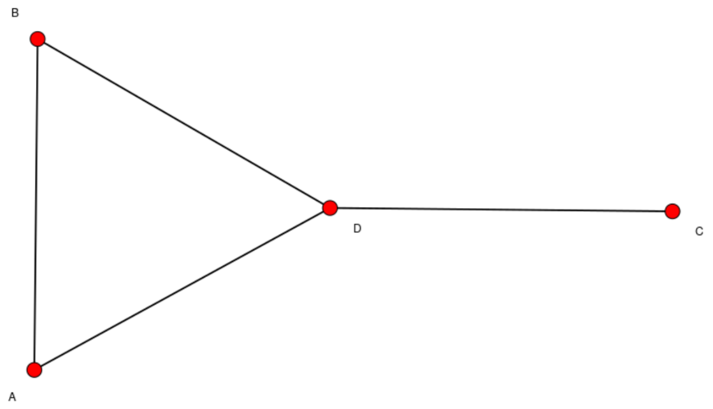
$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

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

Remark 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 $E = \{12, 14, 24, 34\}$ - no need to mention $\{21, 41, 42, 43\}$ since undirected. Hence, $n_E = 4$.

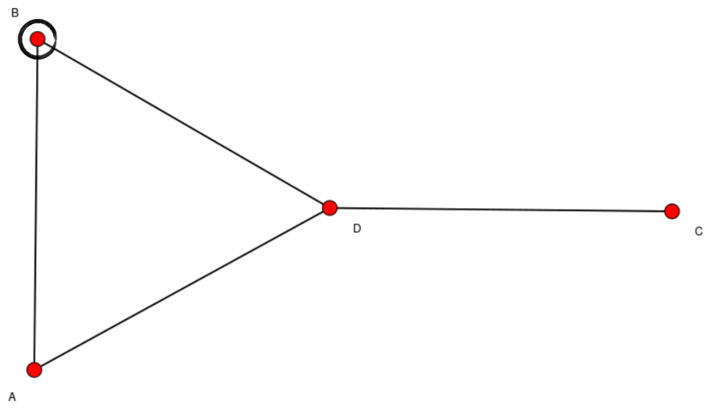
Further, $d(1) = 2$, $d(2) = 2$, $d(3) = 1$ and $d(4) = 3$.

Network Representation

Row i contains list of vertices connected to vertice i.

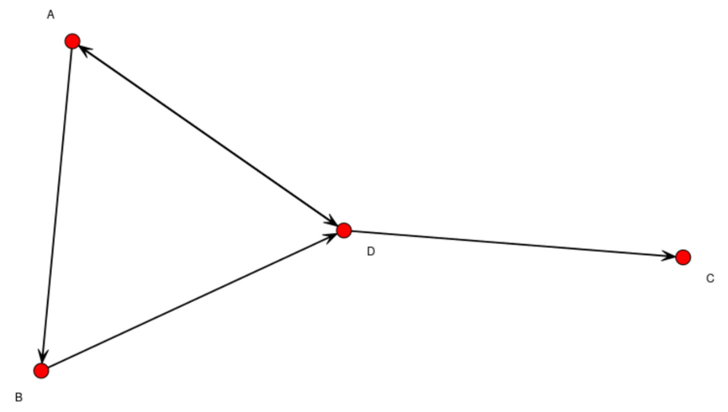
$$d(i) = \sum_{j=1}^{n_V} A_{i,j} = \boldsymbol{A}_{i,\cdot}^{\mathsf{T}} \mathbf{1}.$$

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



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).

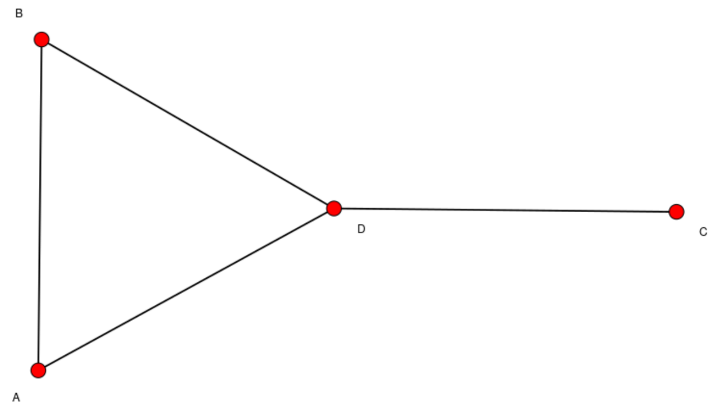
or $E = \{(1 \rightarrow 2),(1 \leftrightarrow 4),(2 \rightarrow 4),(3 \leftarrow 4)\}$

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. $\boldsymbol{d} = \boldsymbol{T}\mathbf{1}$.

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

Networks with R

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 $\boldsymbol{x} \in \mathbb{R}^{n_V}$,

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

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 $[0, 2]$.

Networks with R

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.

Idea: given vertex distances $\boldsymbol{D} = [D_{i,j}]$ (shortest parth, e.g.), we want to find

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

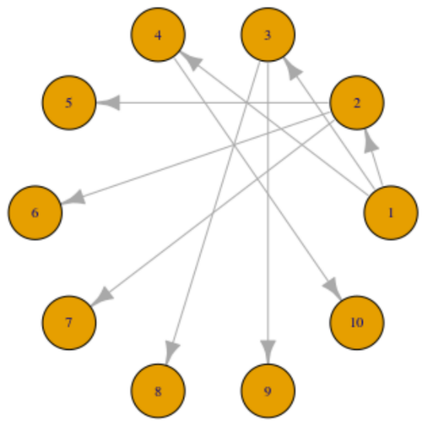
Consider

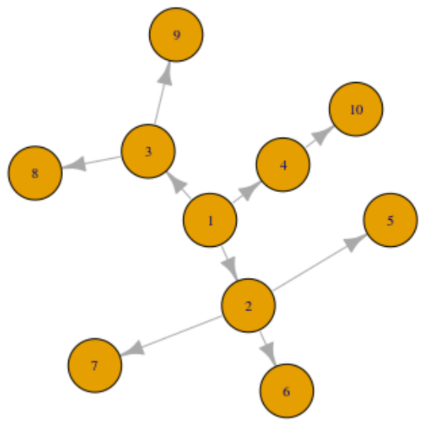
$$\min_{\vec{\boldsymbol{z}}_1,\cdots,\vec{\boldsymbol{z}}_{n_V}} \left\{\left(D_{i,j} - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|\right)^2\right\}$$

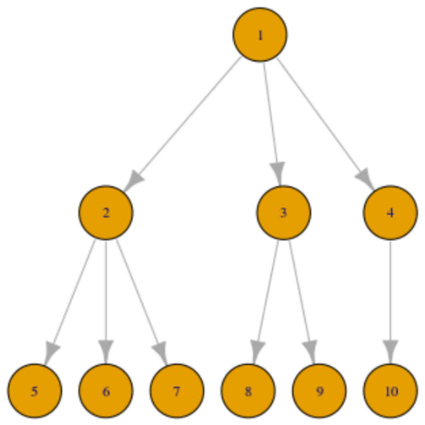
Possible to add some constraints, e.g. centralities

Networks with R

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

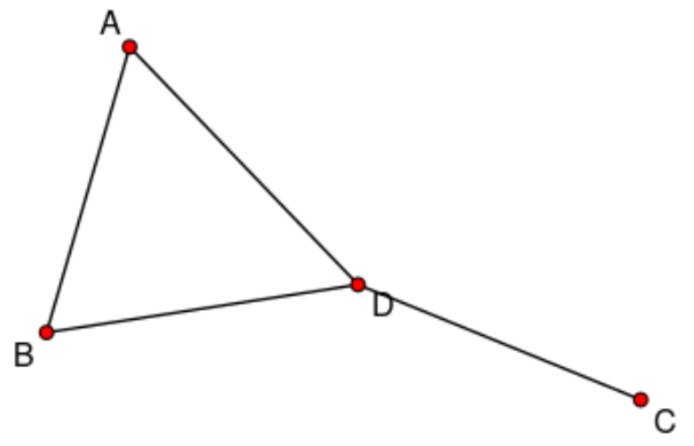


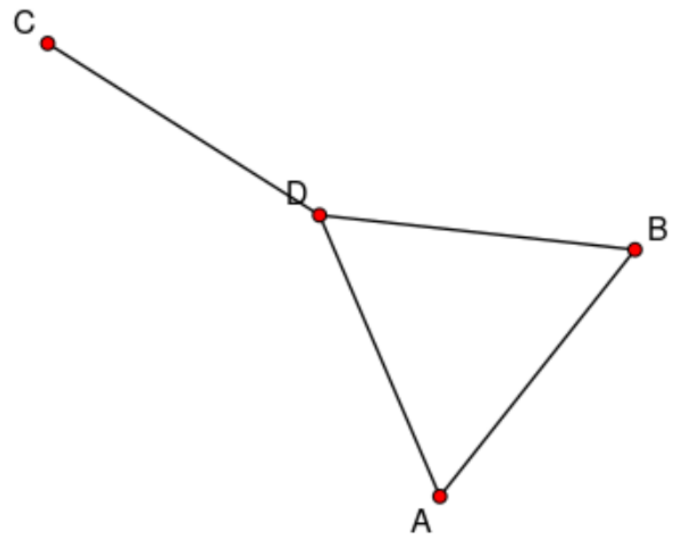




Networks with R

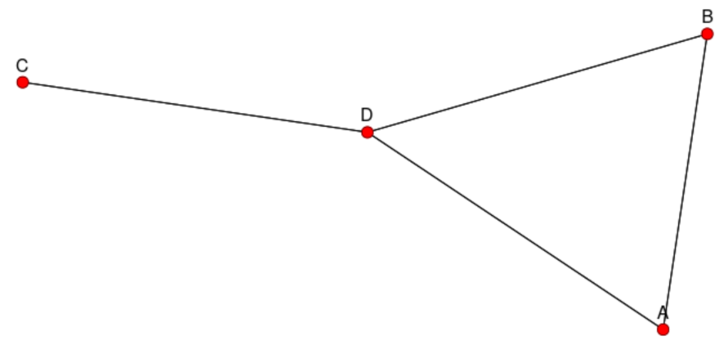
```
> library(network)
> m = matrix(0,4,4)
> colnames(m)=rownames(m)=LETTERS[1:4]
> m[1,2]=m[1,4]=m[2,1]=m[2,4]=m[3,4]=m
    [4,1]=m[4,2]=m[4,3]=1
> m
  A B C D
A 0 1 0 1
B 1 0 0 1
C 0 0 0 1
D 1 1 1 0
> nm=network(m,directed=FALSE)
> plot(nm,displaylabels=TRUE,boxed.
    labels=FALSE)
```





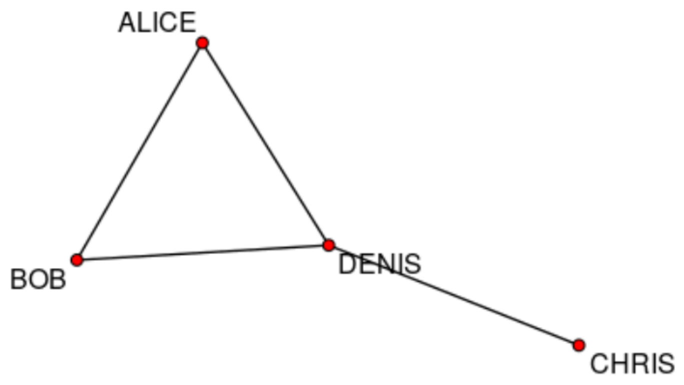
Networks with R

```
> pm=plot(nm,displaylabels=TRUE,boxed.
    labels=FALSE)
> plot(pm,col="white",xlab="",ylab="",
    axes=FALSE,ylim=c(min(pm[,2]),.4+
    max(pm[,2])))
> text(pm[,1],pm[,2],colnames(g),pos=3)
> for(i in 1:nrow(g)){
+   for(j in 1:nrow(g)){
+     if(g[i,j]) segments(pm[i,1],pm[i
    ,2],pm[j,1],pm[j,2])
+   }}
> points(pm[,1],pm[,2],col="red",cex
    =1.4,pch=19)
```



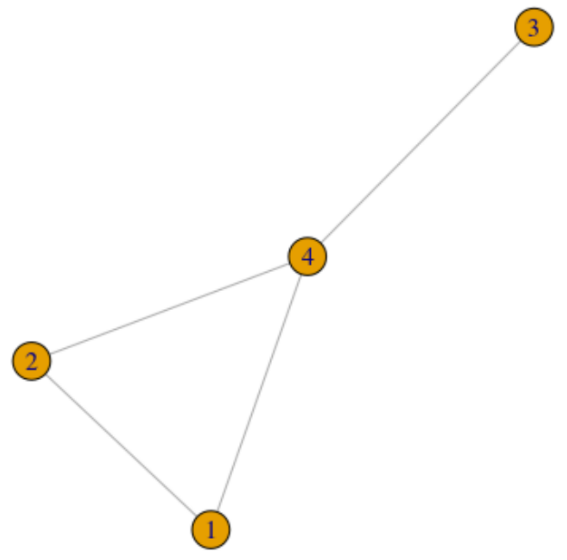
Networks with R

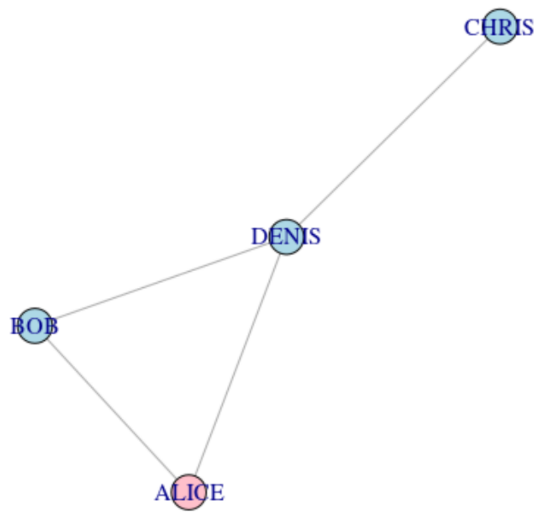
```
> network.vertex.names(nm)
[1] "A" "B" "C" "D"
> network.vertex.names(nm)=c("ALICE","
    BOB","CHRIS","DENIS")
> plot(nm,displaylabels=TRUE,boxed.
    labels=FALSE)
```



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)
```

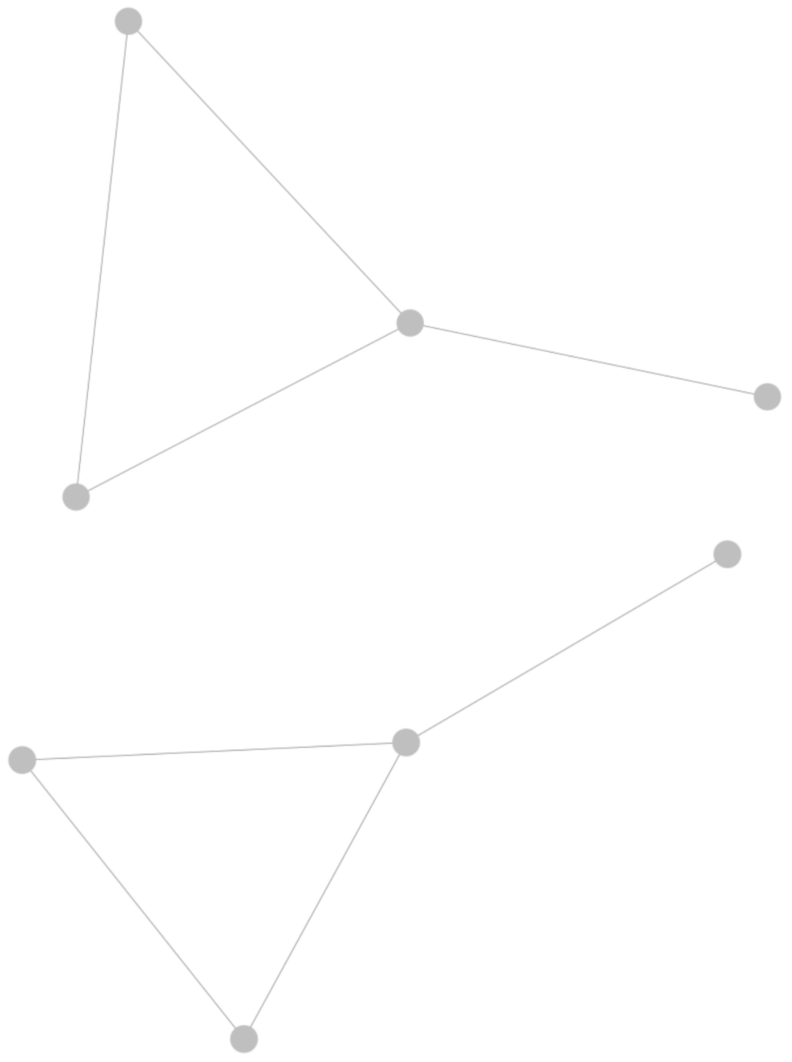




Networks with R

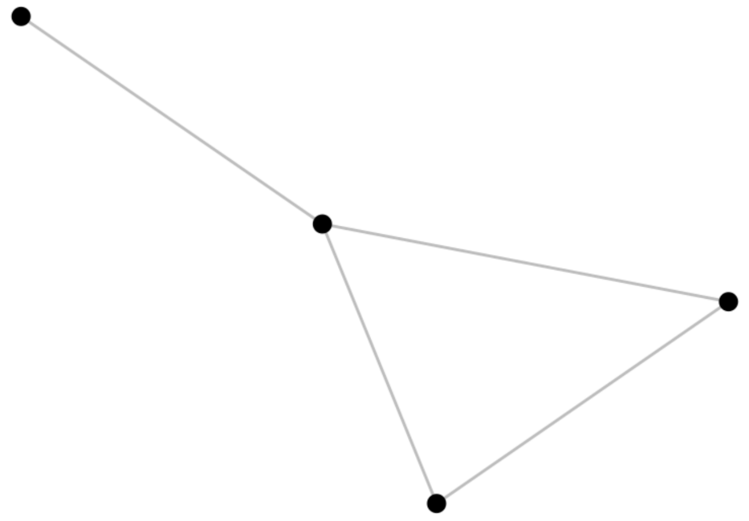
It is also possible to use some ggplot2 graphs
see aide mémoire

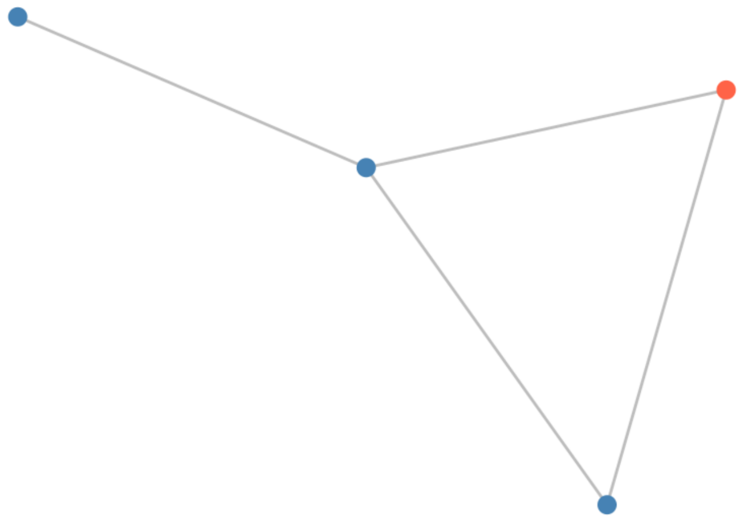
```
> library(GGally)
> library(ggnet)
> library(ggplot2)
> library(sna)
> ggnet2(nm)
> ggnet2(nm,mode="kamadakawai")
> ggnet2(nm,mode="circle")
> ggnet2(nm,mode="target")
```



Networks with R

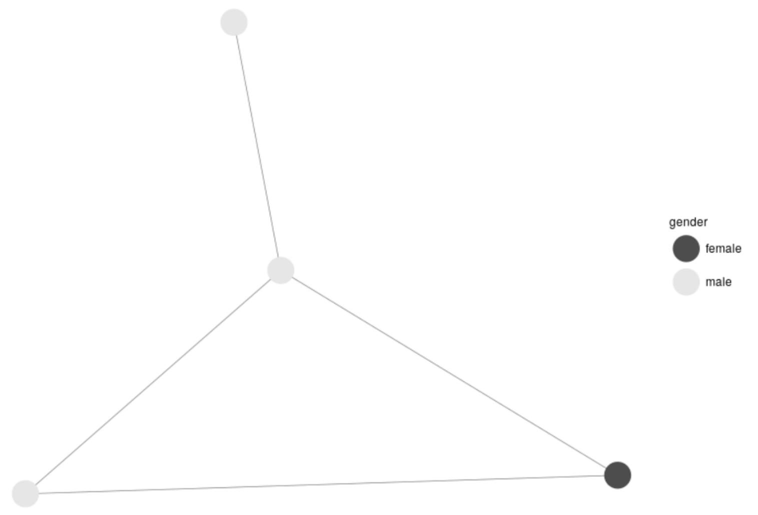
```
> ggnet2(nm,node.size=6,node.color="
    black",edge.size=1,edge.color="grey
    ")
> ggnet2(nm,node.size=6,color=c("tomato
    ",rep("steelblue",3)),edge.size=1,
    edge.color="grey")
```

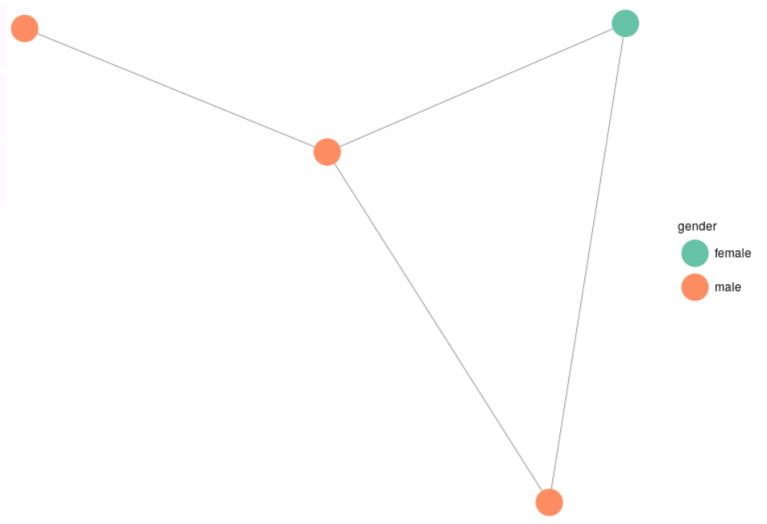




Networks with R

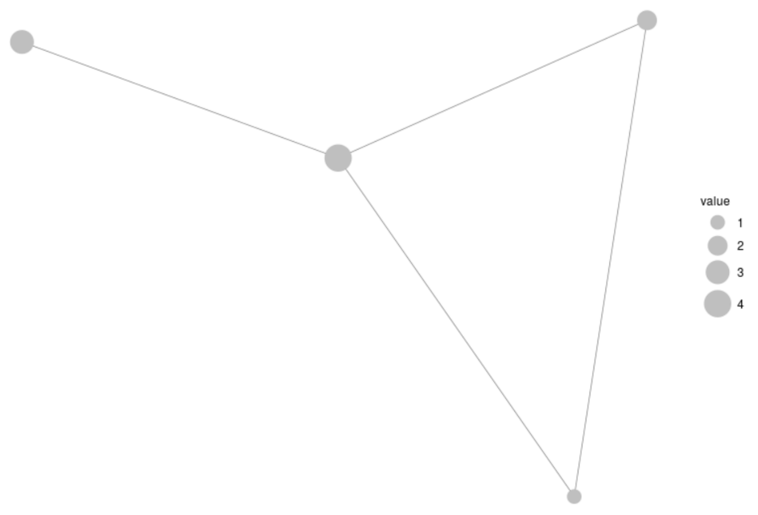
```
> ?"%v%"
> nm %v% "gender" = c("female","male","
    male","male")
> ggnet2(nm,color="gender")
> ggnet2(nm,color="gender", palette="
    Set2")
```

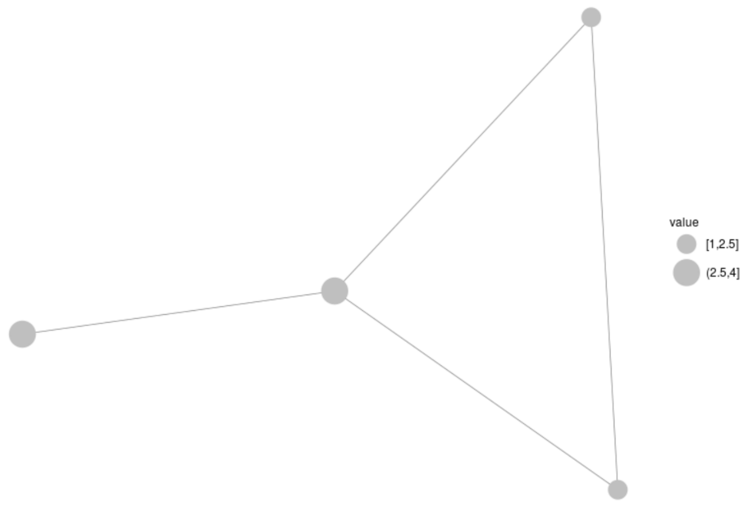




Networks with R

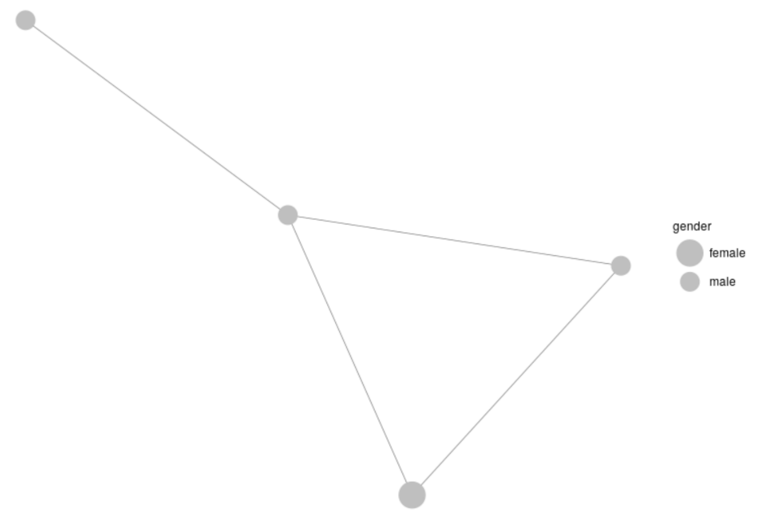
```
> nm %v% "value" = c(2,1,3,4)
> ggnet2(nm,size="value")
> ggnet2(nm,size="value",size.cut=2)
```

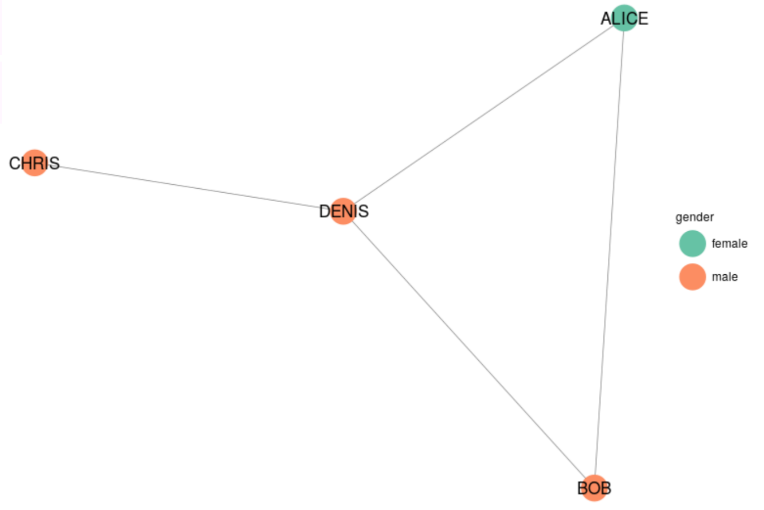




Networks with R

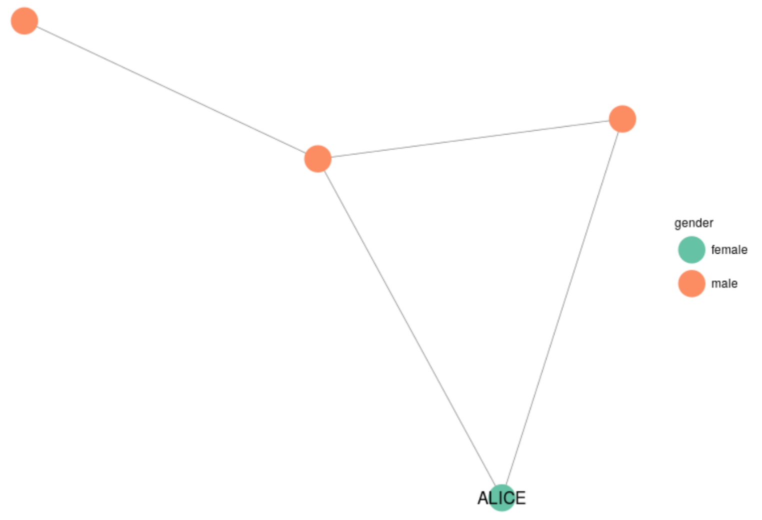
```
> ggnet2(nm,size="gender",size.palette=
    c("female"=2,"male"=1))
> ggnet2(nm,label=TRUE,color="gender",
    label.color="black",palette="Set2")
```

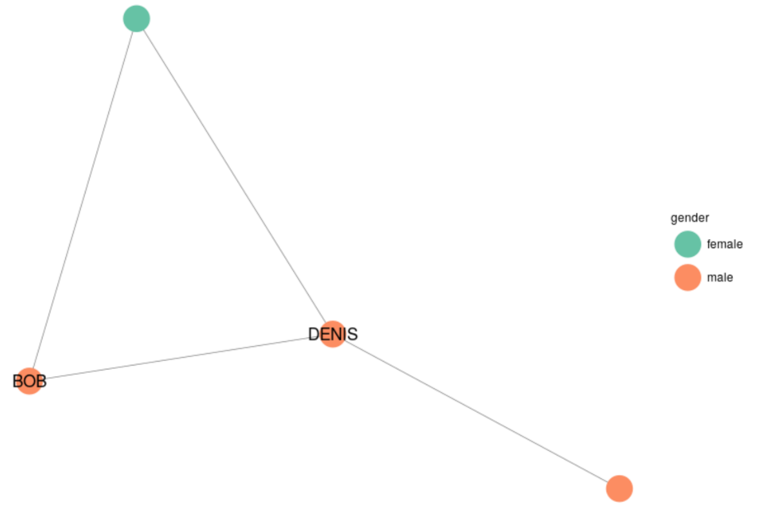




Networks with R

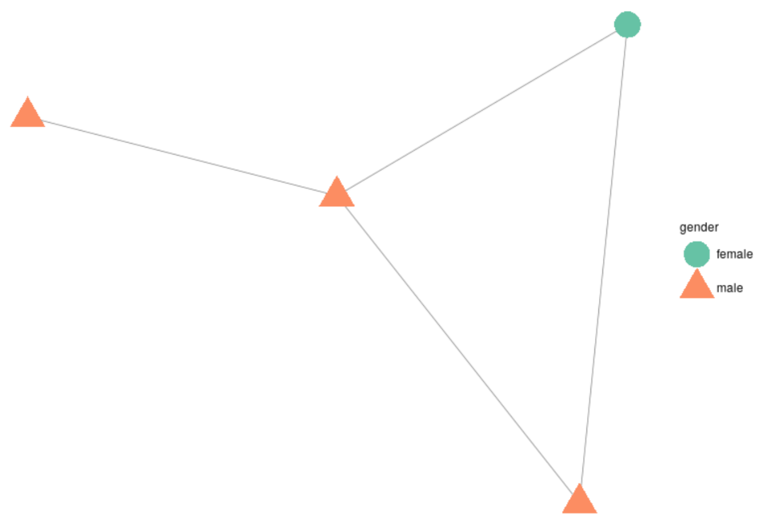
```
> ggnet2(nm,label="ALICE",color="gender
    ",label.color="black",palette="Set2
    ")
> ggnet2(nm,label=c("BOB","DENIS"),
    color="gender",label.color="black",
    palette="Set2")
)
```





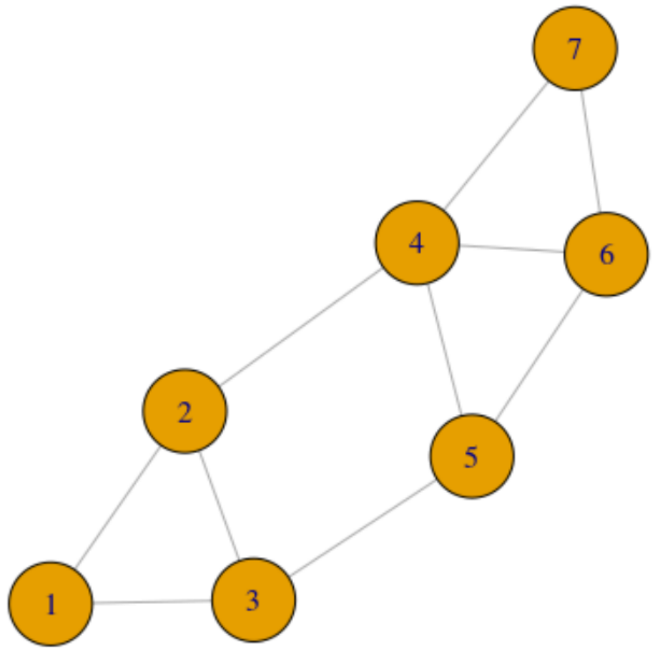
Networks with R

```
> ggnet2(nm,color="gender",palette="
    Set2",shape="gender")
```



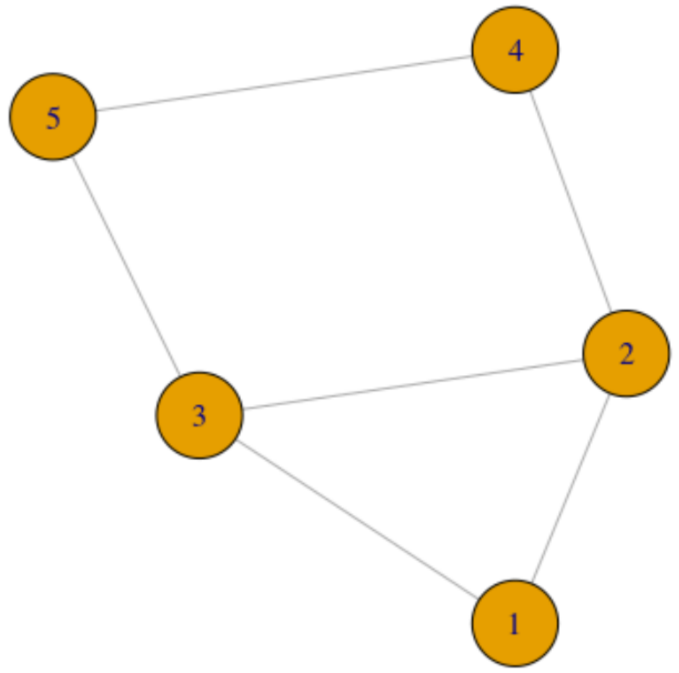
Networks with R

```
> library(igraph)
> g1 = graph.formula(1-2, 1-3, 2-3,
    2-4, 3-5, 4-5, 4-6, 4-7, 5-6, 6-7)
> V(g1)
+ 7/7 vertices, named, from 13ca0c3:
[1] 1 2 3 4 5 6 7
> E(g1)
+ 10/10 edges from 13ca0c3 (vertex
    names):
 [1] 1--2 1--3 2--3 2--4 3--5 4--5 4--6
     4--7 5--6 6--7
 > plot(g1)
```

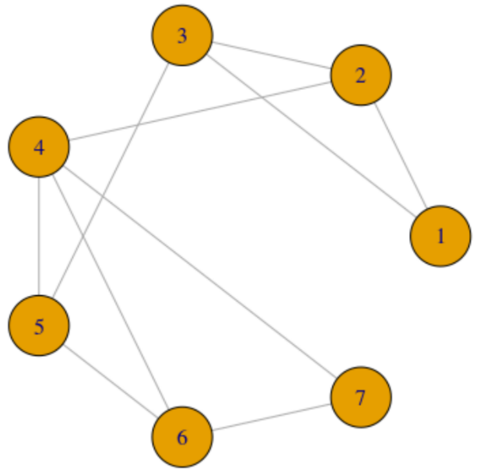


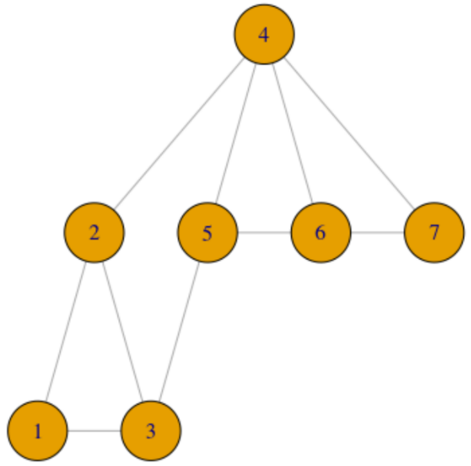
Networks with R

```
> h <- induced.subgraph(g1, 1:5)
> plot(h)
> h1 <- h
> h2 <- graph.formula(4-6, 4-7, 5-6,
    6-7)
> g1 <- graph.union(h1,h2)
```



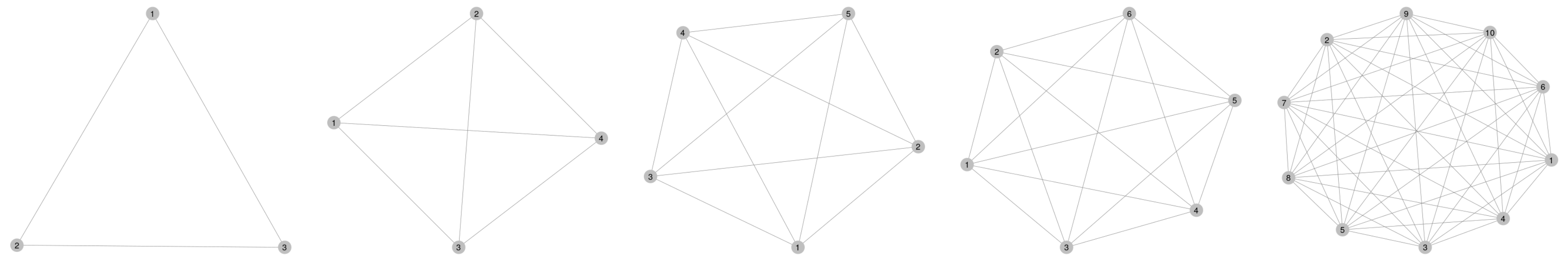
```
> plot(g1, layout=layout.circle)
> plot(g1, layout=layout.reingold.tilford)
> neighbors(g1,4)
+ 4/7 vertices, named:
[1] 2 5 6 7
> igraph::degree(g1)
1 2 3 4 5 6 7
2 3 3 4 3 3 2
> clusters(g1)
$membership
1 2 3 4 5 6 7
1 1 1 1 1 1 1
$csize
[1] 7
> igraph::is.connected(g1)
[1] TRUE
```



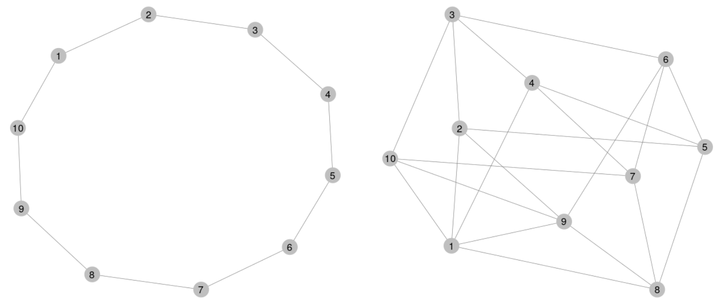


Special Graphs

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



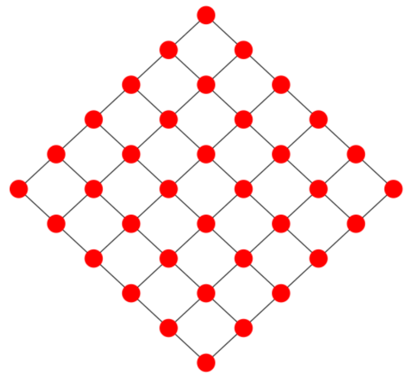
A regular graph is a graph in which every vertice has the same degree.

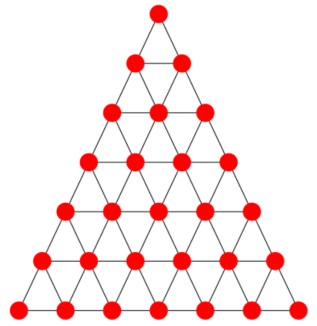


Special Graphs

A tree is a connected graph with no cycles (with parent/children, ancestor/descendant, root/branches/leaves etc)

The networks on the right are called a lattice, or grid graph.





Other Types of Graphs : Bipartite Graph

More complex graphs can be generated, to visualize bipartite networks, with e.g. players (rows) and state of nature (columns), where the matrix is a payoff matrix.

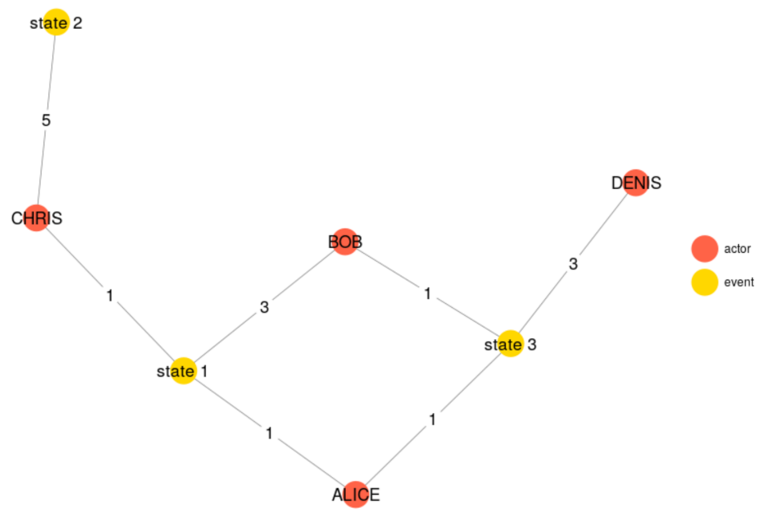
Alternative definition: a bipartite graph is one where $V = V_1 \cup V_2$ such that there are no edges betweens vertices V_1 and V_2.

Note that from $(V_1 \cup V_2, E)$ one can define one-mode projections induced graphs (V_1, E_1) and (V_2, E_2)

Remark this topic will be discussed more in the second part.

Other Types of Graphs : Bipartite Graph

```
> outcome=matrix(c
    (1,3,1,0,0,0,5,0,1,1,0,3),4,3)
> colnames(outcome)=paste("state",1:3)
> rownames(outcome)=c("ALICE","BOB","
    CHRIS","DENIS")
> outcome
        state 1 state 2 state 3
ALICE         1       0       1
BOB           3       0       1
CHRIS         1       5       0
DENIS         0       0       3
> outc=network(outcome,matrix.type="
    bipartite",names.eval="payoff")
> ggnet2(outc,color="mode",label=TRUE,
    edge.label="payoff")
```



Random Networks (and Statistical Aspects)

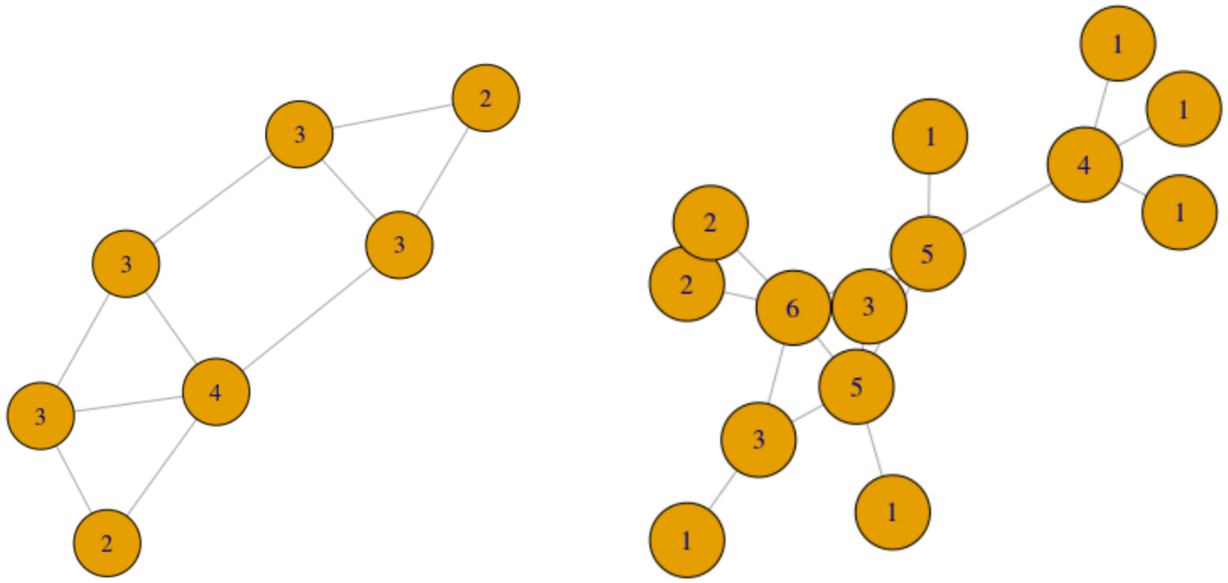
The degree of a vertice v is $d(v)$, the number of vertices in V incident to v (i.e. the number of neighbors of v)

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 Networks (and Statistical Aspects)

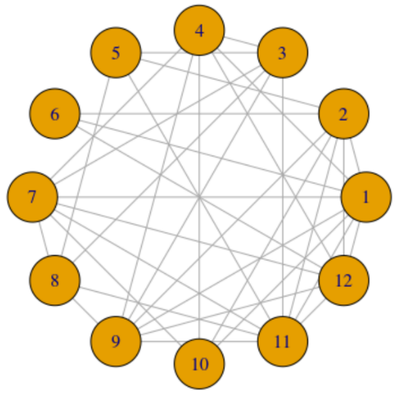


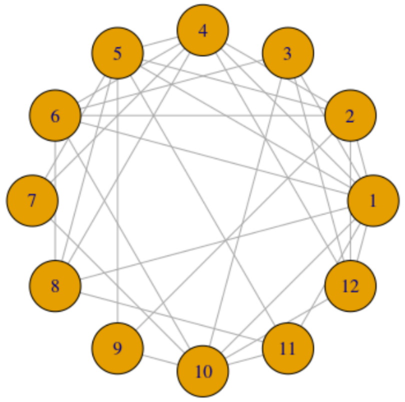
High-degree vertices are likely to be influential, central, prominent.

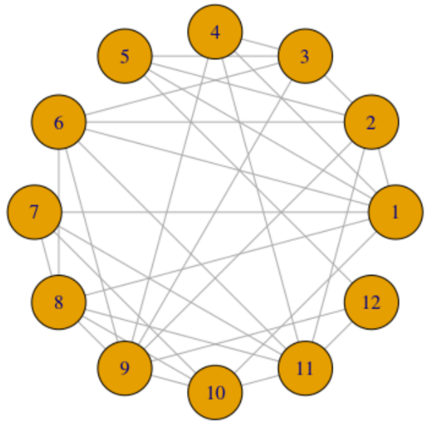
Random Graph Model

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))
```







Random Graph Model

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.

Scale-free network and Power-Law

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 $\mathbb{N}$)

Scale-free network and Power-Law

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

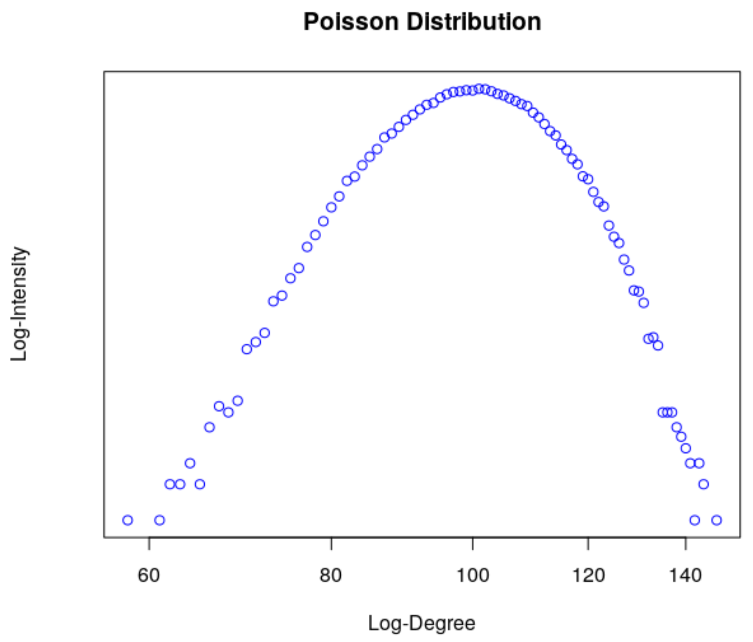
$$\mathbb{E}[f_\delta] = \frac{\alpha-1}{\alpha-2}\delta_0$$

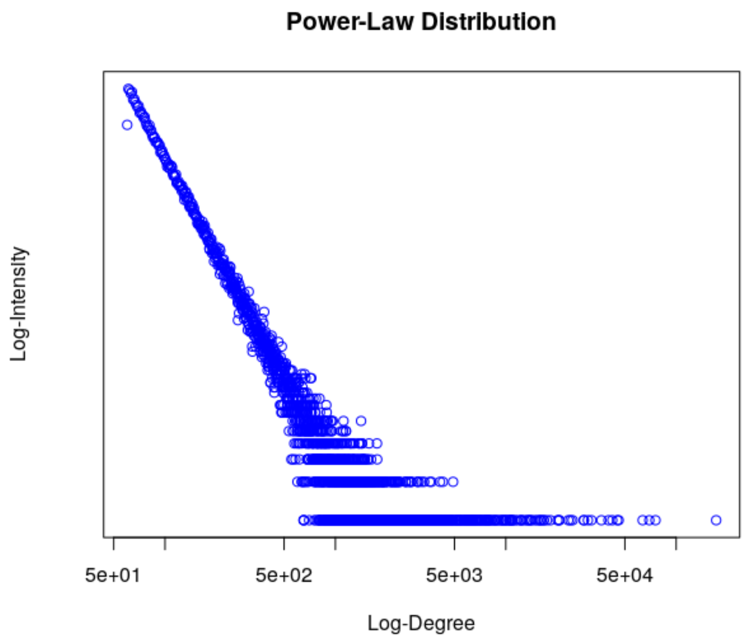
but if $\alpha \in [2,3]$, $\text{Var}[f_\delta] = \infty$.

Scale-free network and Power-Law

Use log-log Pareto plots to visualize f_δ, either on $\log f_\delta$,

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

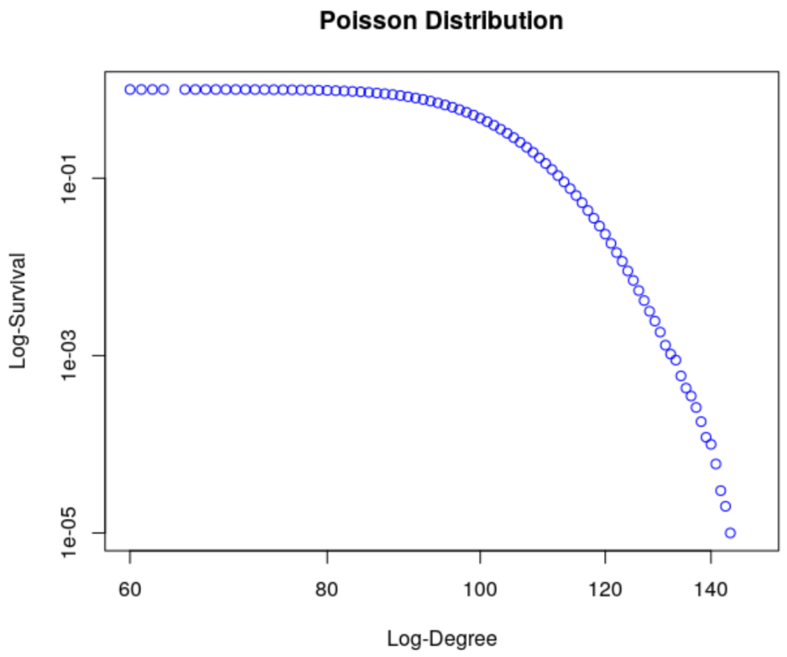


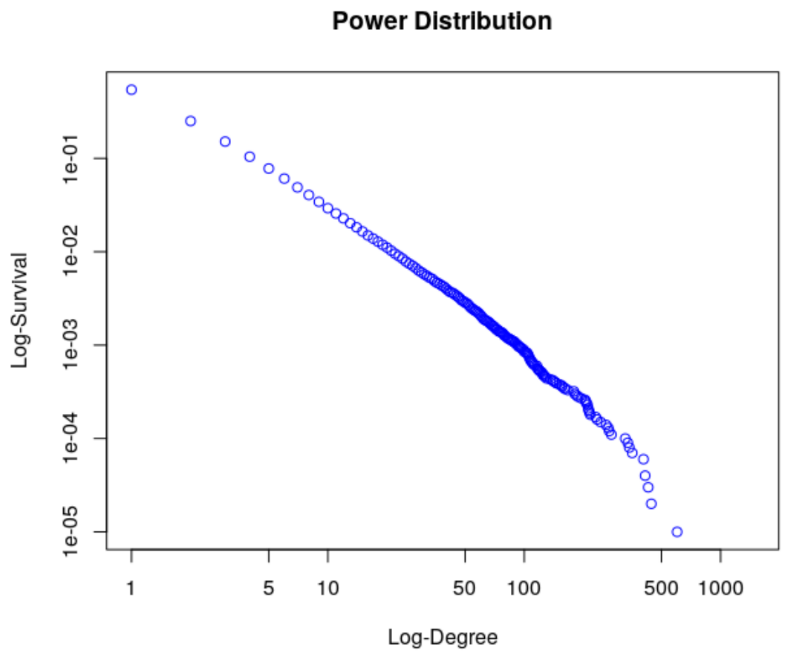


Scale-free network and Power-Law

... 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")
```





Scale-free network and Power-Law

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

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

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)

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

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)

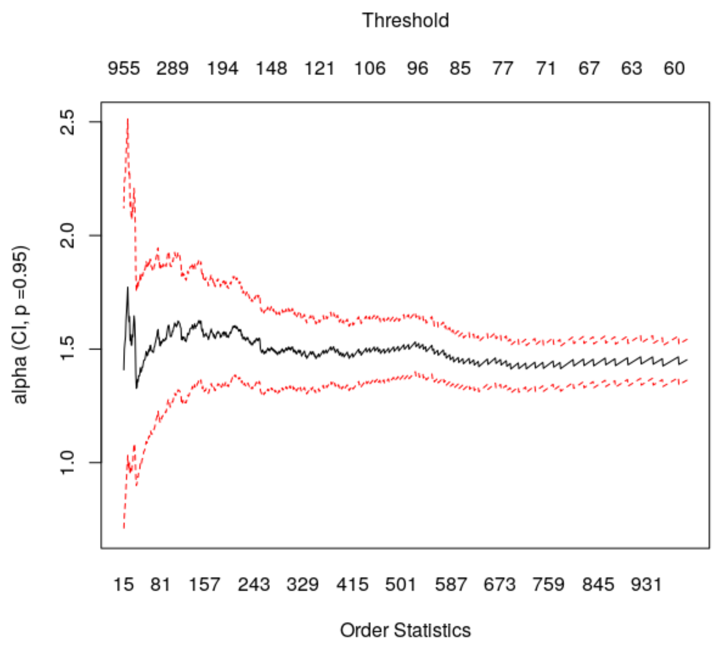
$$\widehat{\alpha}_k = 1 + \left(\frac{1}{k} \sum_{i=0}^{k-1} \log\left(\frac{\delta_{n_V - i:n_V}}{\delta_{n_V - k:n_V}} \right) \right)^{-1}$$

Hill's plot is the graph of $\widehat{\alpha}_k$ as a function of k

Remark in standard litterature on extreme value the so called α is $\alpha - 1$ from the network litterature...

Scale-free network and Power-Law

```
> library(evir)
> hill(degree(g))
```



Preferential attachment model and Power-Law

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

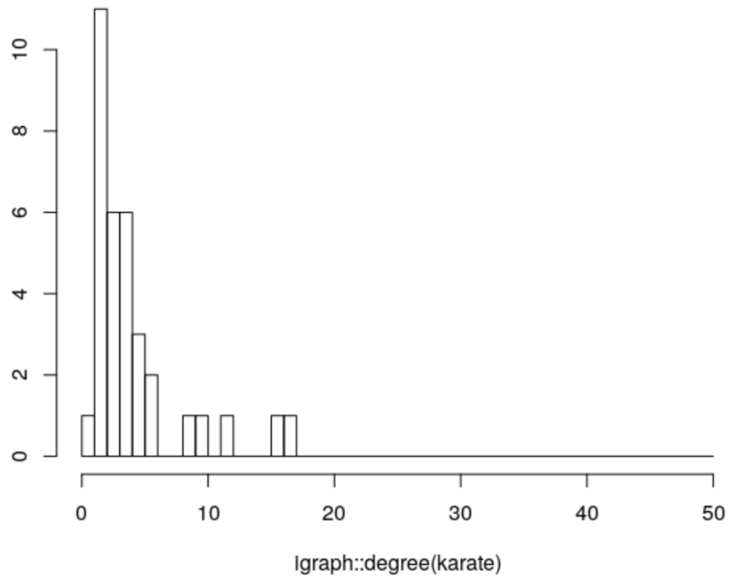
Preferential attachment model leads to rich-gets-richer dynamics

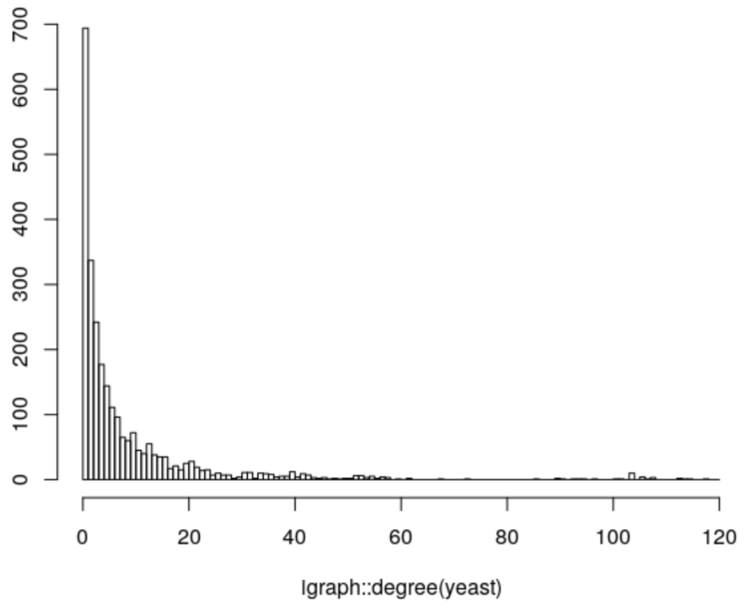
Degrees have a power-law distribution with tail exponent $\alpha = 1 + \dfrac{1}{1-p}$

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

Degrees In Networks and Power Law

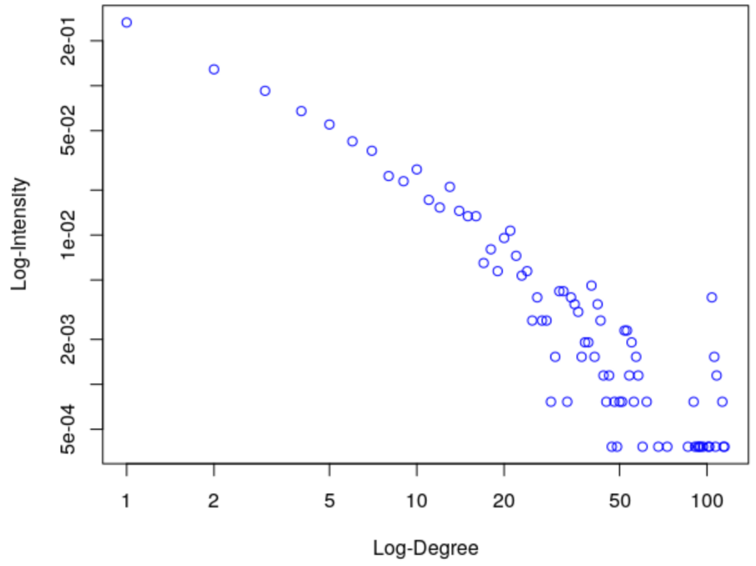
```
> hist(igraph::degree(karate),breaks=
     seq(0,50))
> data(yeast)
> hist(igraph::degree(yeast),breaks=
     seq(0,120))
```





Degrees In Networks and Power Law

```
> 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.

Remark 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.

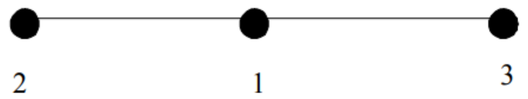
A walk from v_0 to v_k contains a path from v_0 to v_k (remove subcycles).

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.

For a digraph, it is said to be strongly connected if every vertice v' is reachable from every other vertice v via a directed walk.

Paths In Networks

Consider a network with 3 nodes and adjency matrix



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

```
> 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?

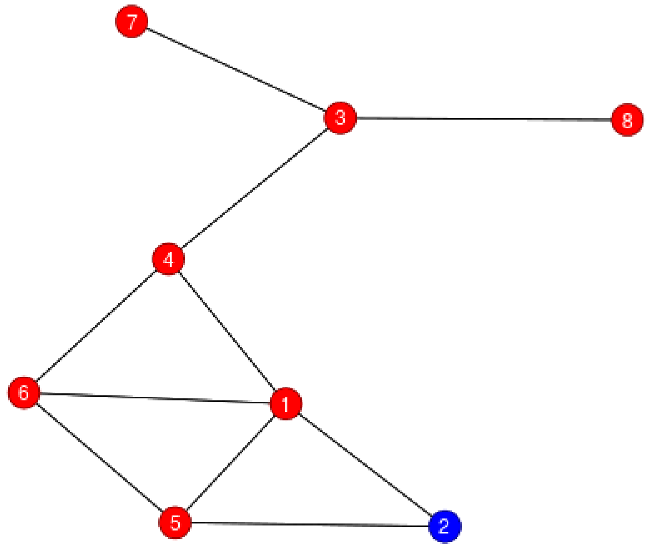
Idea : use adjacency list from a starting vertice s to explore

- 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
$\{3\}$	
$\{\}$	



click to visualize the construction

Connectivity In Networks

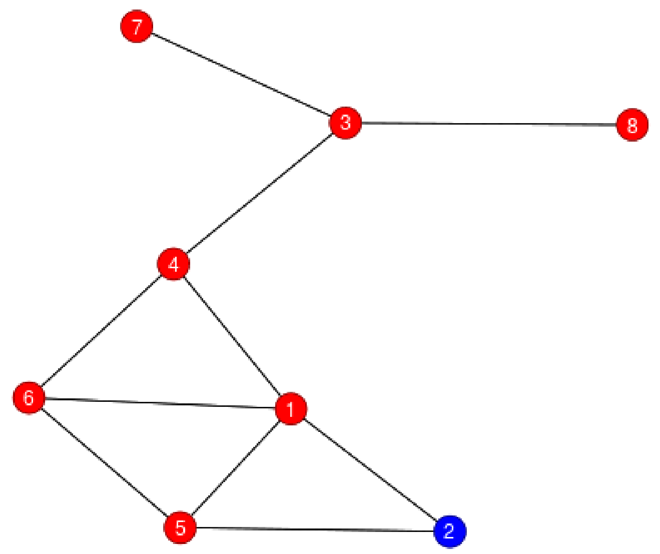
This algorithm can be improved,

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

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

Connectivity In Networks: BFS

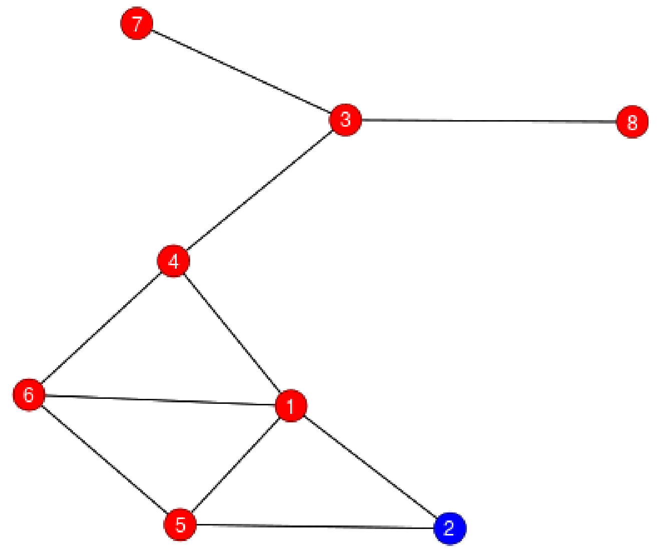
L	M
$\{2\}$	2
$\{2, 1\}$	1
$\{2, 1, 5\}$	5
$\{1, 5\}$	
$\{1, 5, 4\}$	4
$\{1, 5, 4, 6\}$	6
$\{5, 4, 6\}$	
$\{4, 6\}$	
$\{4, 6, 3\}$	3
$\{6, 3\}$	
$\{3\}$	
$\{3, 7\}$	7
$\{3, 7, 8\}$	8
$\{7, 8\}$	
$\{8\}$	
$\{\}$	



click to visualize the construction

Connectivity In Networks: DFS

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
$\{2, 1, 4, 6\}$	
$\{2, 1, 4\}$	
$\{2, 1\}$	
$\{2\}$	
$\{\}$	



click to visualize the construction

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 longuest shortest path in a graph is the diameter of the graph

more generally, the sum of the weights of its constituent edges is minimized.

Shortest Path and Dijkstra's algorithm

(initially designed for a digraph, with possible costs)

Start with $\mathcal{S} = \{v\}$, $\delta(v, v') = 1$ if $A_{v,v'} = 1$, ∞ otherwise.

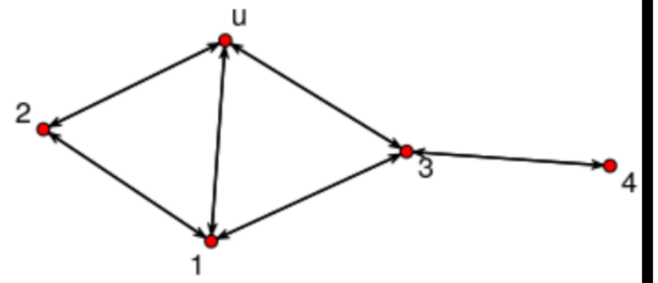
Until $\mathcal{S} = V$: if there is $v' \notin \mathcal{S}$ such that $A_{u,v'} = 1$, $\forall u \in \mathcal{S}$, then $\mathcal{S} = V \cup \{v'\}$ and

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

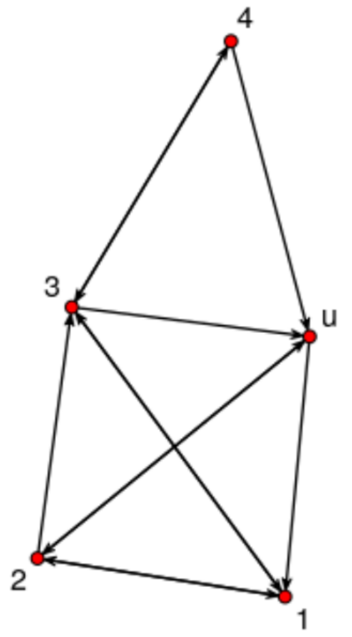
Shortest Path and Dijkstra's algorithm

Example

step	$\mathcal{S}$	$d(u)$	$d(1)$	$d(2)$	$d(3)$	$d(4)$
0	$\{u\}$	0	1	1	1	∞
1	$\{u,1,2,3\}$	0	1	1	1	2



step	$\mathcal{S}$	$d(u)$	$d(1)$	$d(2)$	$d(3)$	$d(4)$
0	$\{u\}$	0	1	1	∞	∞
1	$\{u,1,2\}$	0	1	1	2	∞
2	$\{u,1,2,3\}$	0	1	1	2	3



Shortest Path and Dijkstra's algorithm

```
> m["u",1:4]
1 2 3 4
1 1 0 0
> (m %*% m)["u",1:4]
1 2 3 4
1 1 2 0
> (m %*% m %*% m)["u",1:4]
1 2 3 4
4 2 2 2
```

Proposition Dijkstra's algorithm find the shortest path in $O(n^2)$ time, from u to any other vectice V.

see also Floyd–Warshall algorithm

Friendship Paradox

People typically have fewer friends than their friends

Consider a vertex $v \in V$, in the undirected graph $G = (V, E)$, and let $d(v)$ denote the number of edges touching it (i.e. v has $d(v)$ friends).

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

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

The average number of friends that a typical friend has is

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

But

$$\sum_{v \in V} \left(\frac{1}{d(v)} \sum_{v' \in E_v} d(v') \right) = \sum_{v,v' \in G} \left(\frac{d(v')}{d(v)} + \frac{d(v)}{d(v')} \right)$$

Friendship Paradox

$$= \sum_{v,v'\in G}\left(\frac{d(v')^2+d(v)^2}{d(v)d(v')}\right) = \sum_{v,v'\in G}\left(\frac{(d(v')-d(v))^2}{d(v)d(v')}+2\right) > \sum_{v,v'\in G}(2) = \sum_{v\in V} d(v)$$

Thus,

$$\frac{1}{n_V}\sum_{v\in V}\left(\frac{1}{d(v)}\sum_{v'\in E_v} d(v')\right) > \frac{1}{n_V}\sum_{v\in V} d(v)$$

Remark This can be related to the variance decomposition $\text{Var}[X]=\mathbb{E}[X^2]-\mathbb{E}[X]^2$ thus

$$\frac{\mathbb{E}[X^2]}{\mathbb{E}[X]} = \mathbb{E}[X] + \frac{\text{Var}[X]}{\mathbb{E}[X]} > \mathbb{E}[X]$$

(Jensen inequality).

See Feld (1991) and Zuckerman & Jost (2001)

Centrality In Networks : Degree Centrality

The goal is to understand who is important, based on their network position.

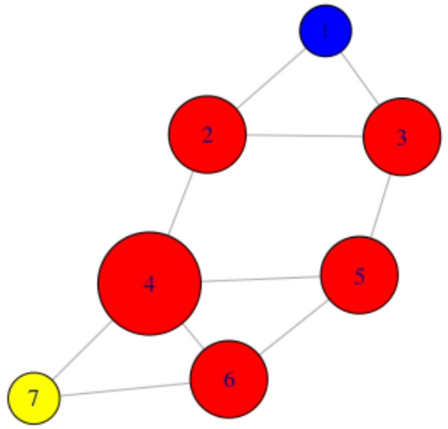
"There is certainly no unanimity on exactly what centrality is or on its conceptual foundations, and there is little agreement on the proper procedure for its measurement" Freeman (1979)

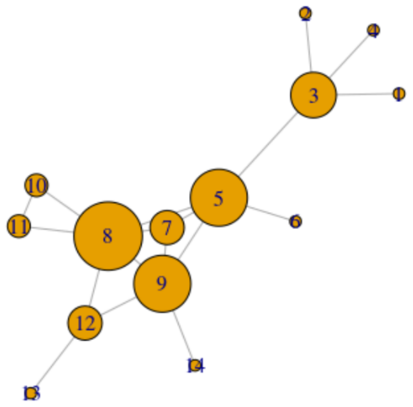
Remark simple question such as *which vertice is central* can be replaced by more interesting ones, such as *what percentage of vertices is crucial to the network connectivity* ?

He or she who has many friends is most important, divided by the maximum possible n_V : for a vertice v, $C_D(v) = \dfrac{d(v)}{n_V}$

Centrality In Networks : Degree

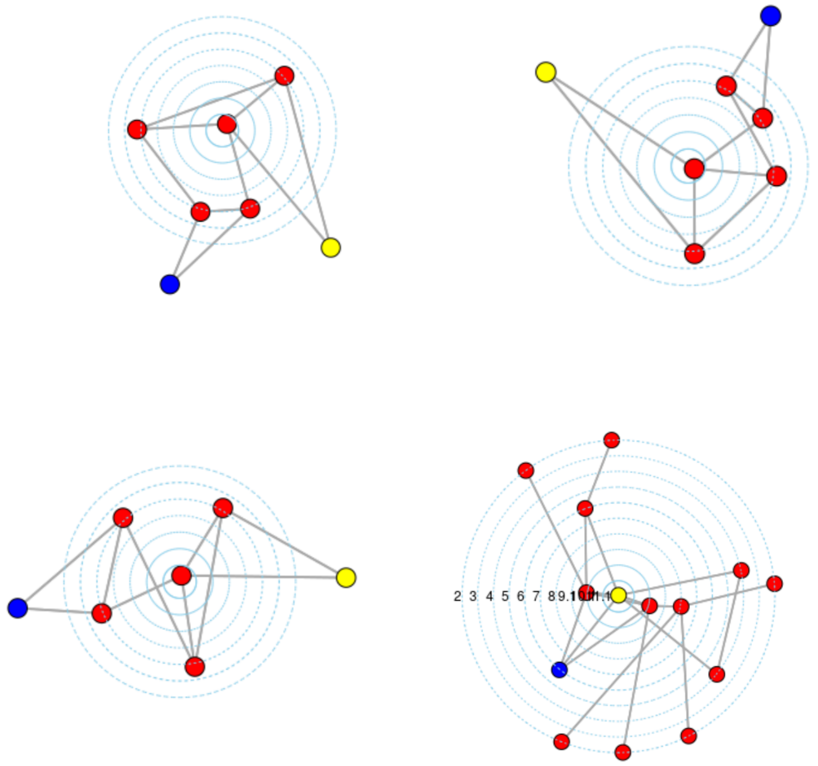
```
> plot(g,vertex.size=degree(ng))
```





Centrality In Networks : Degree

```
> 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
```



Centrality In Networks : Heterogeneity

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).

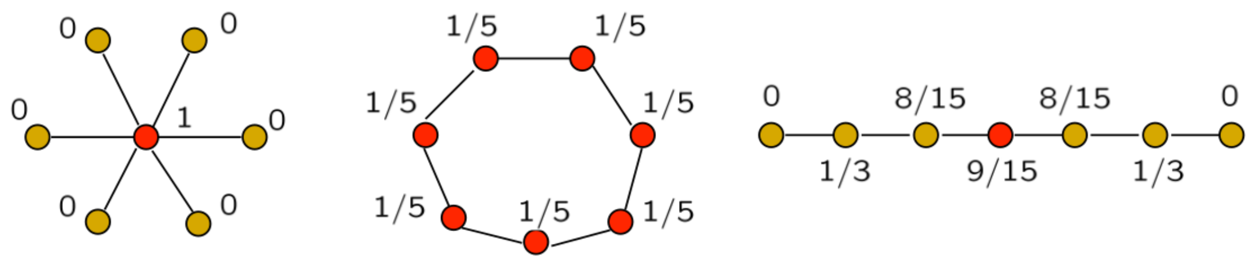
Centrality In Networks : Betweenness

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.



Centrality In Networks : Betweenness

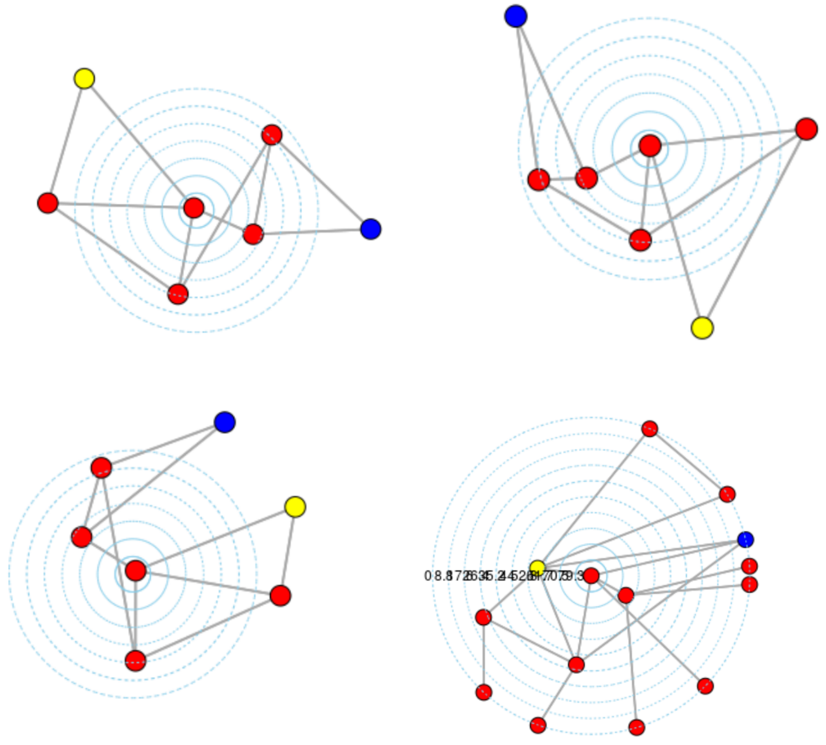
One possible normalization is $C_{B'}(v) = \dfrac{2C_B(v)}{(n_V-1)(n_V-2)}$ (number of pairs of vertices excluding the vertex itself), another one is $C_{\tilde{B}}(v) = \dfrac{C_B(v) - \min C_B}{\max C_B - \min C_B}$

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

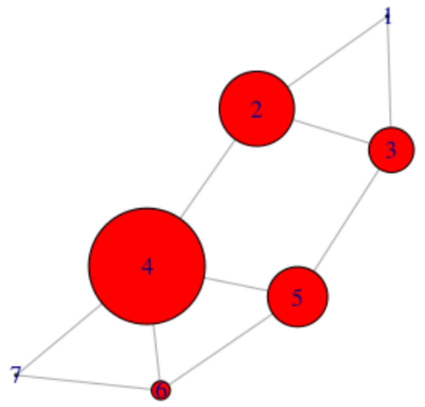
$v^\star = \text{argmax}\{C_B(v)\}$ is seen as the controler of information flow

Centrality In Networks : Betweenness

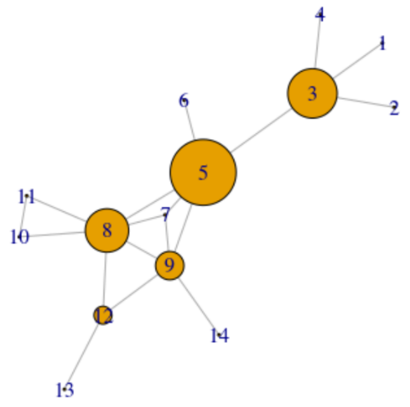
```
> A = get.adjacency(g1, sparse=FALSE)
> ng = network::as.network.matrix(A)
> sna::gplot.target(ng, betweenness(ng)
    , circ.col="skyblue",vertex.col=c("
   blue", rep("red", 5), "yellow"),
   edge.col="darkgray")
> betweenness(ng)
[1]  0.000  6.667  4.000 10.333  5.333
     1.667  0.000
```



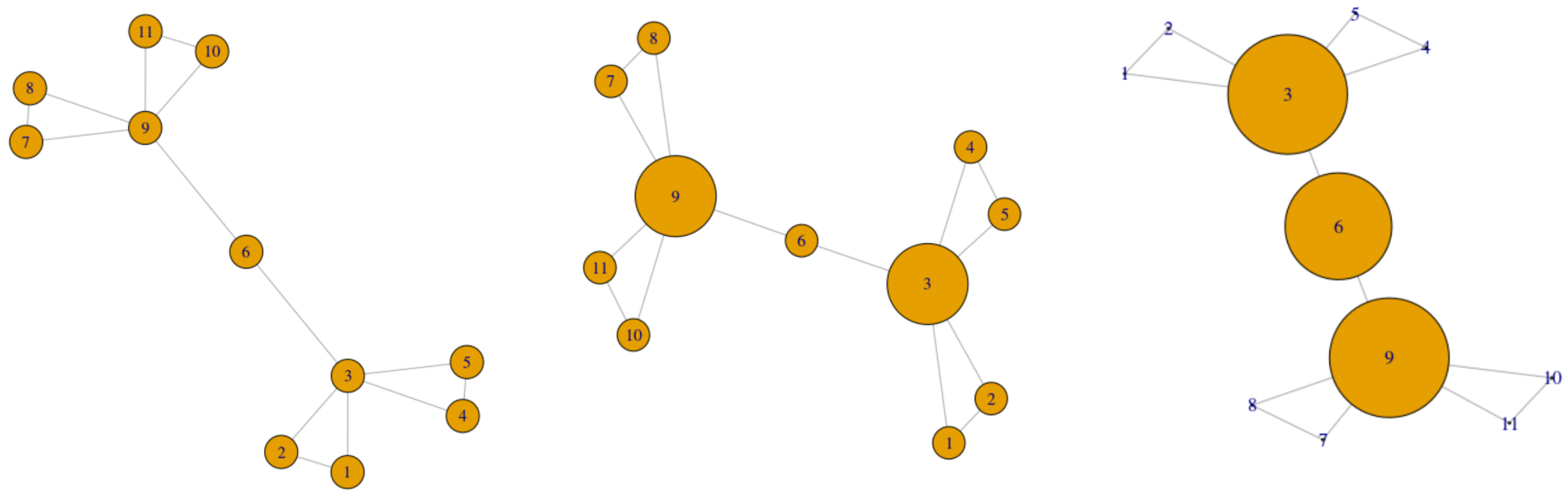
Centrality In Networks : Betweenness



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



Centrality In Networks : Degree vs. Betweenness



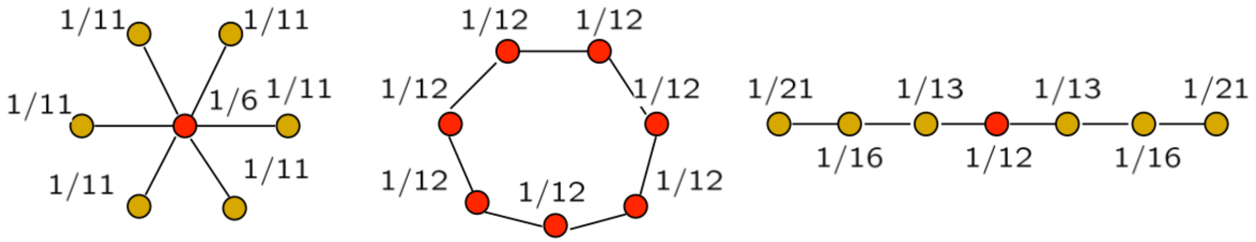
Centrality In Networks : Closeness

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

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

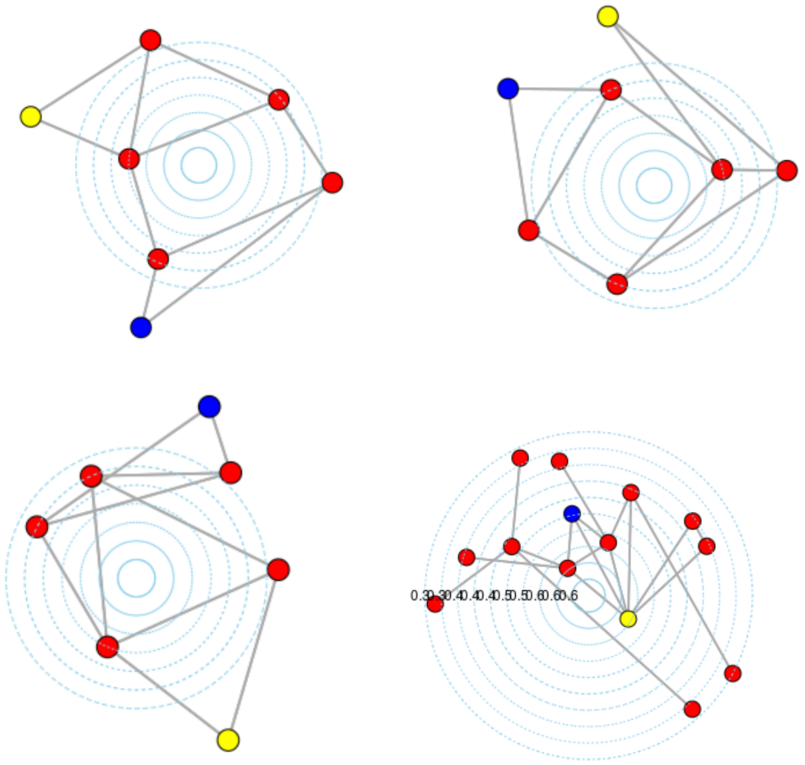


Computation time is $O(n_V^2 \log n_V + n_V n_E)$ on sparce network.

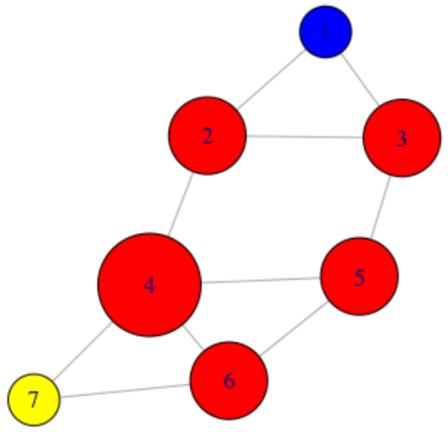
$v^\star = \text{argmax}\{C_C(v)\}$ is the most approachable vertice.

Centrality In Networks : Closeness

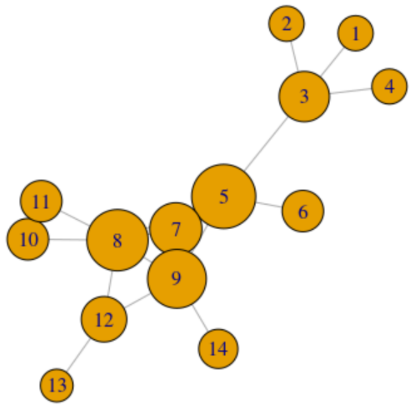
```
> A = get.adjacency(g1, sparse=FALSE)
> ng = network::as.network.matrix(A)
> sna::gplot.target(ng, closeness(ng),
    circ.col="skyblue",vertex.col=c("
    blue", rep("red", 5), "yellow"),
    edge.col="darkgray")
> closeness(ng)
[1] 0.5000 0.6667 0.6000 0.7500 0.6667
    0.6000 0.5000
```



Centrality In Networks : Closeness

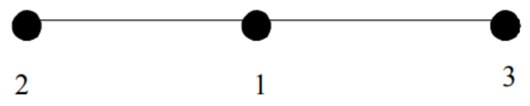


```
> plot(g,vertex.size=closeness(ng))
```



Centrality In Networks : Eigenvector Centrality

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
```

Centrality In Networks : Eigenvector Centrality

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

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

```
> 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
```

Centrality In Networks : Eigenvector Centrality

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
```

Centrality In Networks : Eigenvector Centrality

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.

Centrality In Networks : Eigenvector Centrality

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_R is the (normalized) eigenvector

Centrality In Networks : Eigenvector Centrality

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 $\boldsymbol{u} \in \mathbb{R}_+^{n_V}$

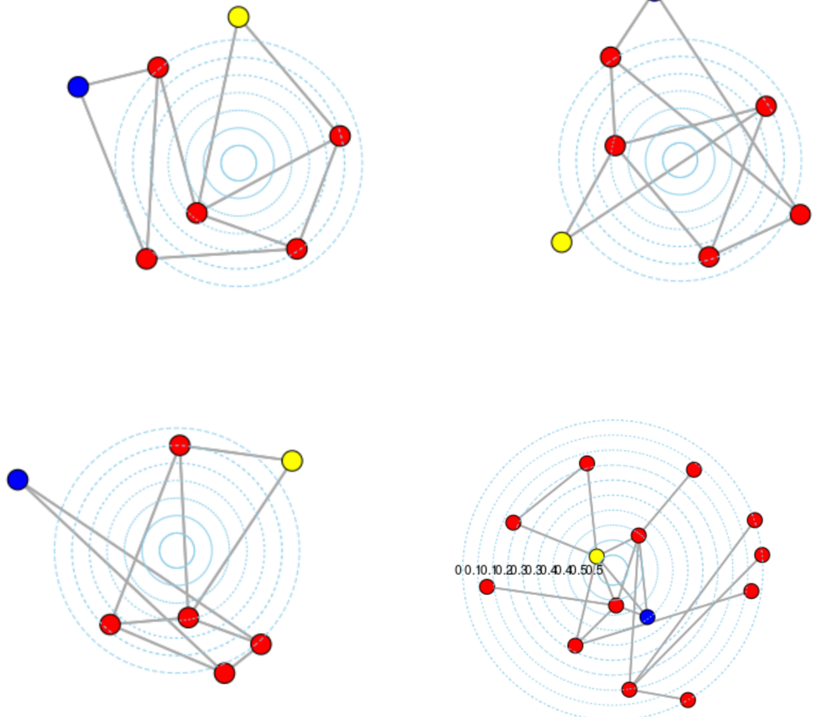
$$b_{\mathbf{1}}(\boldsymbol{G},\beta) = \sum_{h\geq 1} \beta^h \boldsymbol{A}^h \mathbf{1} = ([\mathbb{I} - \beta\boldsymbol{A}]^{-1} - \mathbb{I})\mathbf{1}$$

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

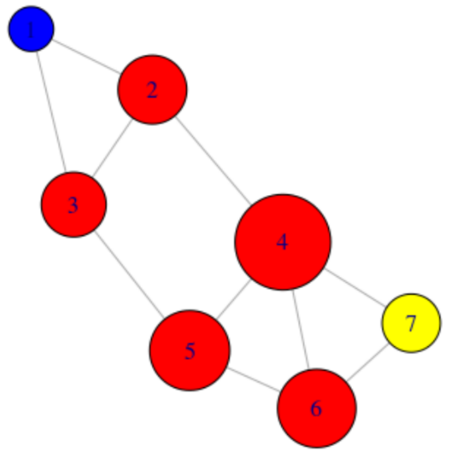
$$b_{\mathbf{1}}(\boldsymbol{G},\beta) = \beta[\mathbb{I} - \beta\boldsymbol{A}]^{-1}\boldsymbol{A}\mathbf{1}$$

Centrality In Networks : Eigenvector Centrality

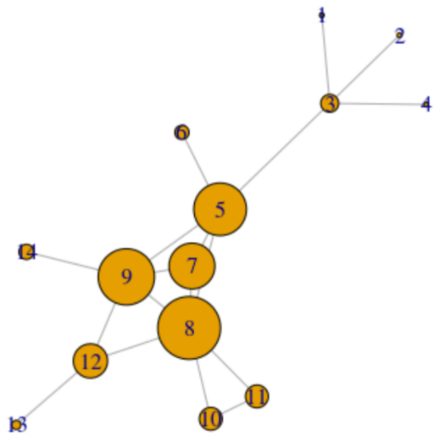
```
> A = get.adjacency(g1, sparse=FALSE)
> ng = network::as.network.matrix(A)
> sna::gplot.target(ng, evcent(ng), circ.col="skyblue",vertex.col=c("blue", rep("red", 5), "yellow"), edge.col="darkgray")
> evcent(ng)
[1] 0.2358069 0.3622484 0.3416056
    0.5038536 0.4215946 0.4129477
    0.3071490
```



Centrality In Networks : Eigenvector Centrality



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



Centrality In Networks : Katz 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.

Remark 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.

Centrality In Networks : Google page rank

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

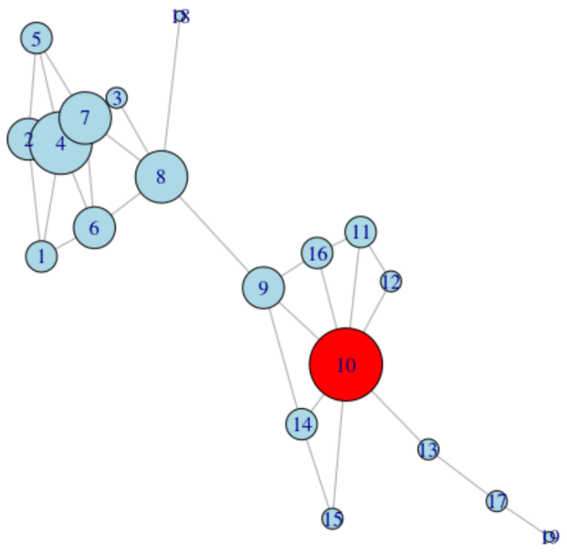
$$\boldsymbol{R}^{(k)} = \widetilde{\boldsymbol{P}}^{\mathsf{T}} \boldsymbol{R}^{(k-1)} \text{ where } \alpha \widetilde{\boldsymbol{P}} + (1-\alpha)\frac{\boldsymbol{1}\boldsymbol{1}^{\mathsf{T}}}{N_v}$$

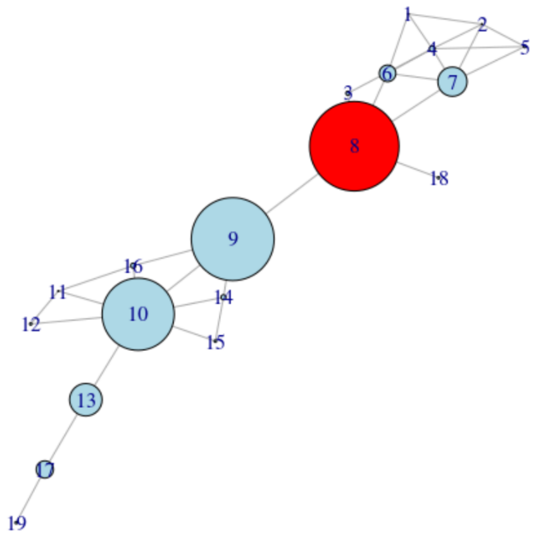
where the scaling factor β is typically around 0.85

Remark power iteration converges to the dominent eigenvector of $\widetilde{\boldsymbol{P}}$.

Centrality In Networks : Comparison

```
> 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
```

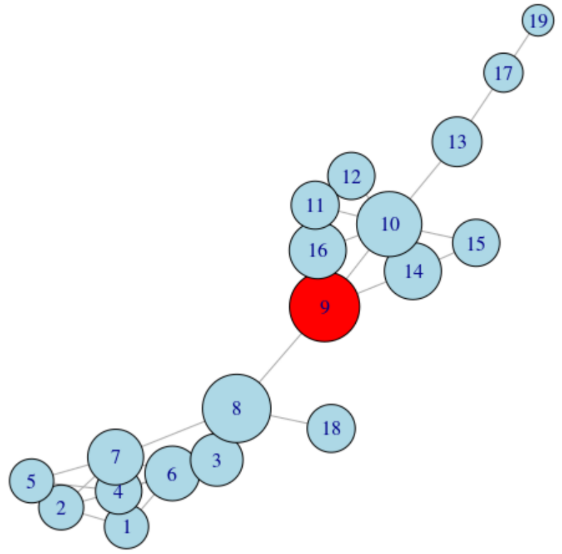


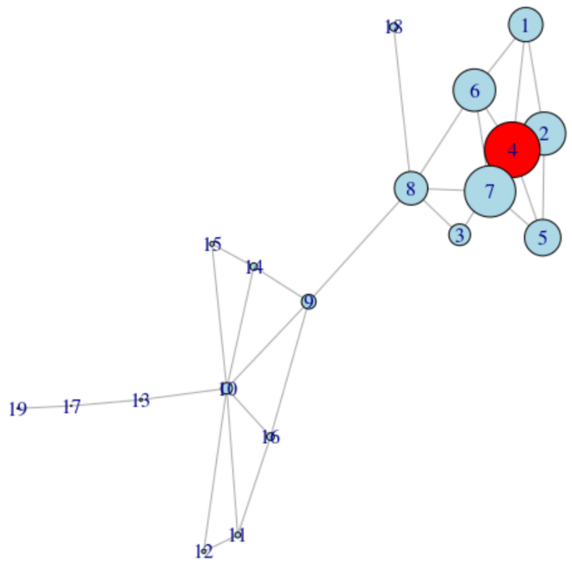


Centrality In Networks : Comparison

```
> 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'





Network Stability

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

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

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.

Nevertheless, the betweeness centrality is not a stable measure

Inference and Networks

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 squared error is

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

Sampling Within Graphs

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...

Sampling Within Graphs

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, $\theta(G) = \dfrac{1}{n_V}\displaystyle\sum_{v\in V} d_v$. 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 $\widehat{\theta} = \dfrac{1}{n}\displaystyle\sum_{v\in V_n^\star} d_v^\star$

Sampling Within Graphs

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.

Sampling Within Graphs

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

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

Further

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

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

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

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

Sampling Within Graphs

Example 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)) 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

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

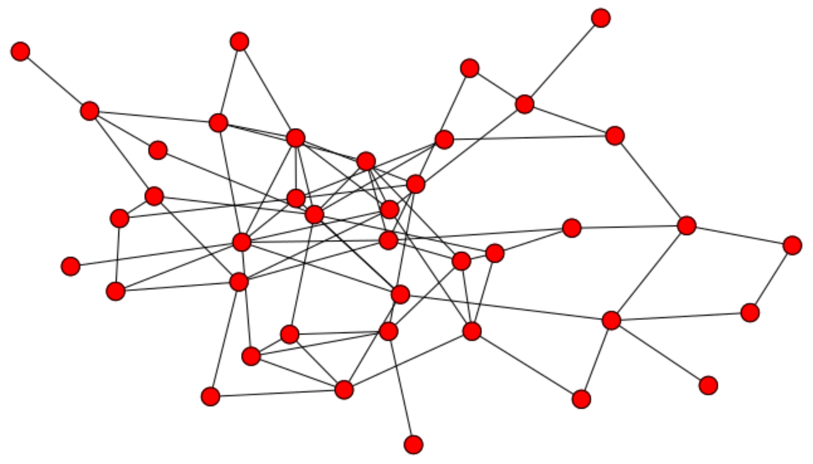
Complicated to estimate if the population is V (vertices) or E (edges).

Sampling Within Graphs : Induced Subgraph Sampling

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



click to visualize the construction

Sampling Within Graphs : Incident Subgraph Sampling

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

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

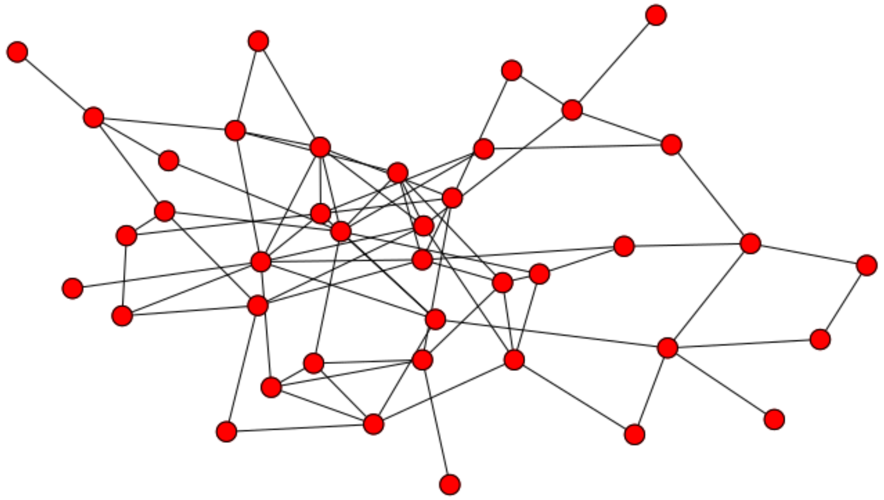
See phone calls

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).

Sampling Within Graphs : Incident Subgraph Sampling

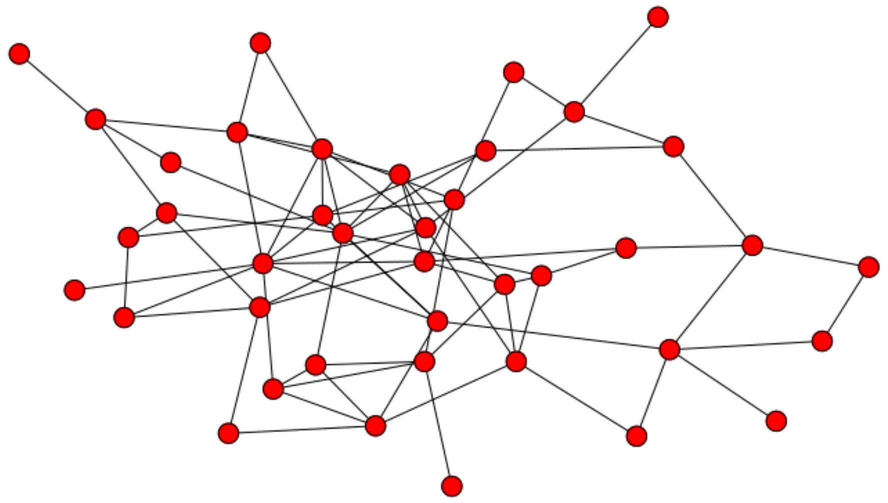


click to visualize the construction

Sampling Within Graphs : Star Sampling

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.



click to visualize the construction

Sampling Within Graphs

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

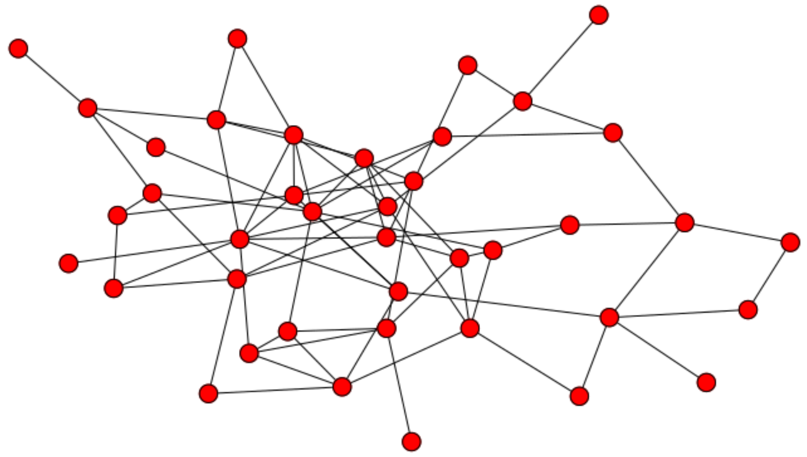
$$\pi_i \sim 1 - \left(1 - \frac{n_s}{n_E}\frac{n_t}{n_E}\right)\exp\left(-\frac{n_s}{n_E}\frac{n_t}{n_E}b_i\right)$$

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).

Sampling Within Graphs



click to visualize the construction

see Dall'Asta *et al.* (2006) for more details.

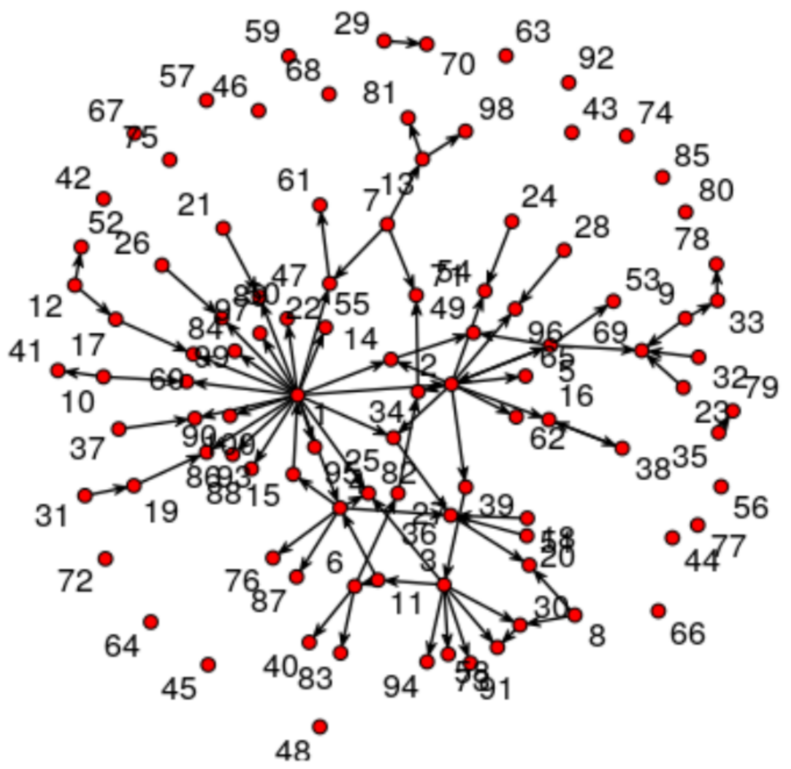
Contagion Within Graphs

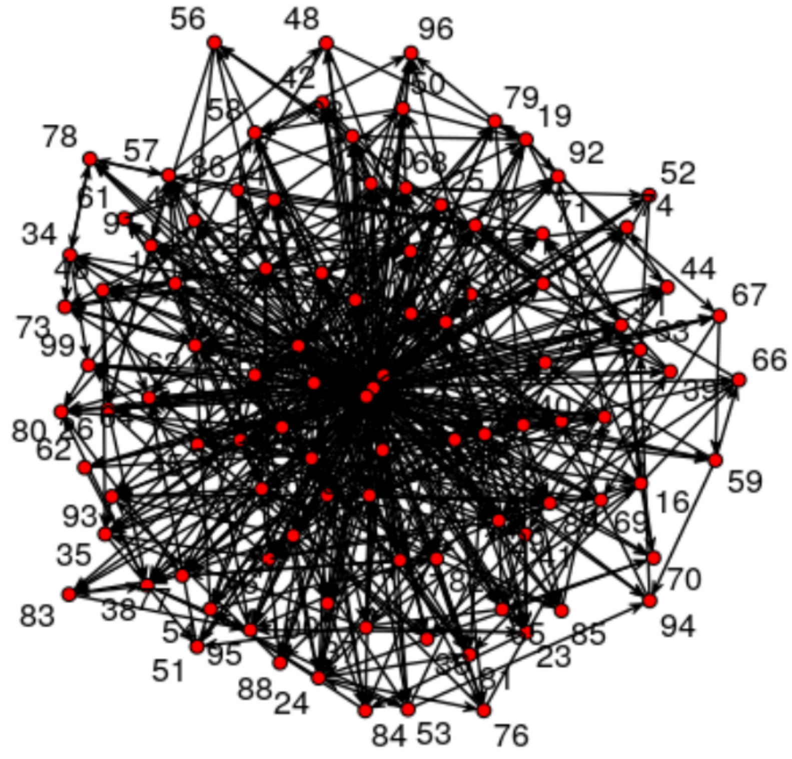
emotion/behaviors of individuals being influenced by the group(s) to which they belong

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

Network Cohesion





Network Cohesion

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

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

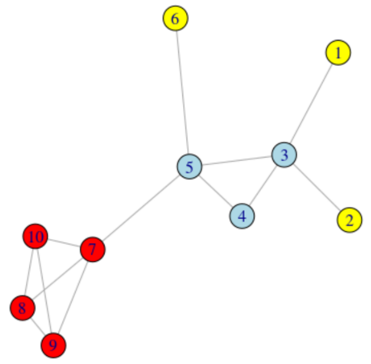
where $H = (V_H, E_H)$, which is the number of edges over the maximum number of edges (based on the number of vertices).

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

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

Network Cohesion

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



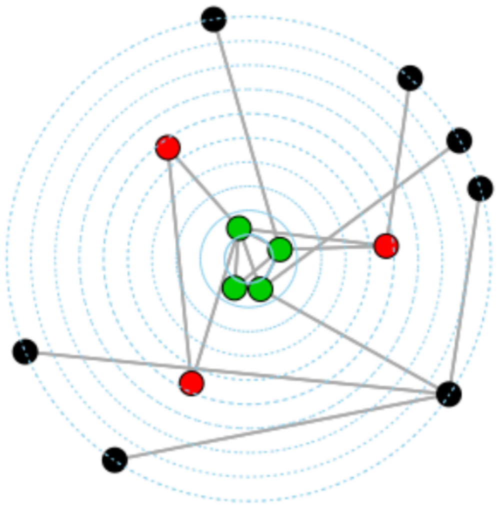
$$n_E > \frac{n_V^2}{2}\frac{n-2}{n-1}$$

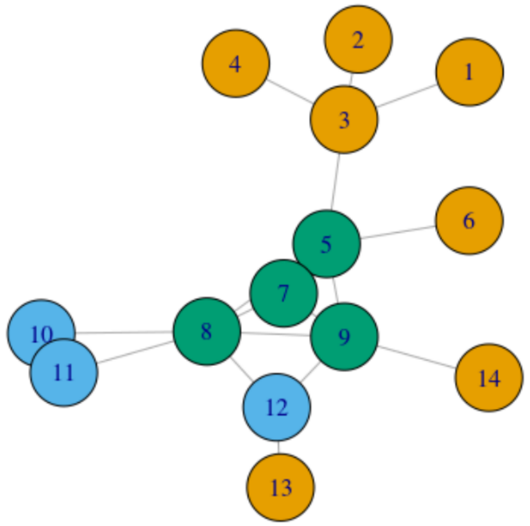
A k-core $G^\star$ is a subgraph of G such that $d_{v^\star} \geq k$ for all $k \in V^\star$, and $G^\star$ is maximal.

Here, degrees are computed on the subgraph $G^\star$.

Network Cohesion

```
> 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)
```



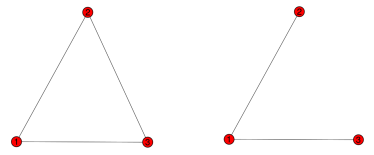


Network Cohesion

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

$\tau_\Lambda(v)$ is the number of connected triplets in G with two edges incident to v



One can define an aggregate local clustering coefficient, $\text{cluster}(v) = \dfrac{\tau_\Delta(v)}{\tau_\Lambda(v)}$

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

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

Network Cohesion

```
> 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
- vectices in different groups should be well separated

Hieractical clustering and spectral partitioning

Partitioning 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

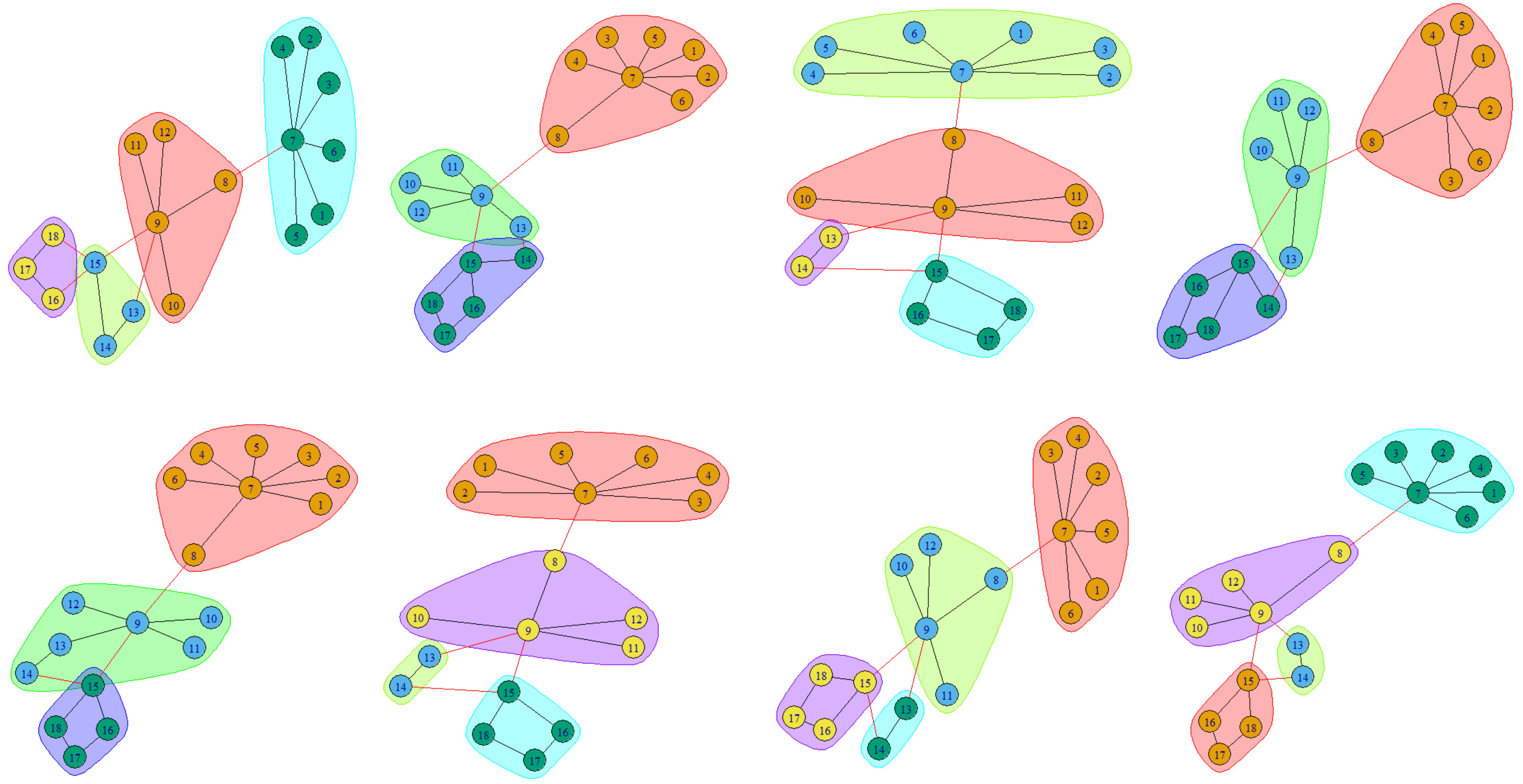
- vertices are well connected among themselves (within a subset)
- vertices are well separated from the remaining vertices (between subsets)

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 $E(C_{k'})$, eddges connecting vertices within subsets

Application community detection in social media

Clusters Within Graphs



Clusters Within Graphs

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

Clusters Within Graphs

- 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 Within Graphs

One can use modularity, from Newman (2004)

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 Within Graphs

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,

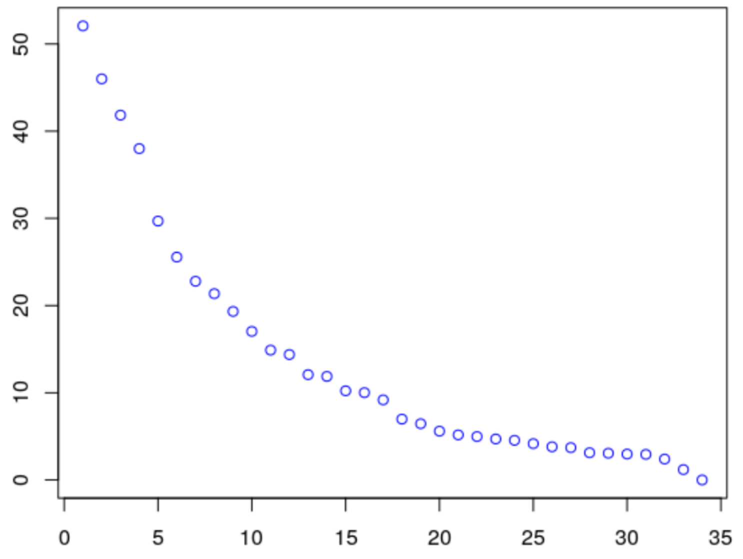
vertices corresponding to particularly large positive or negative entries, in conjunction with their immediate neighbors, are declared to be a cluster

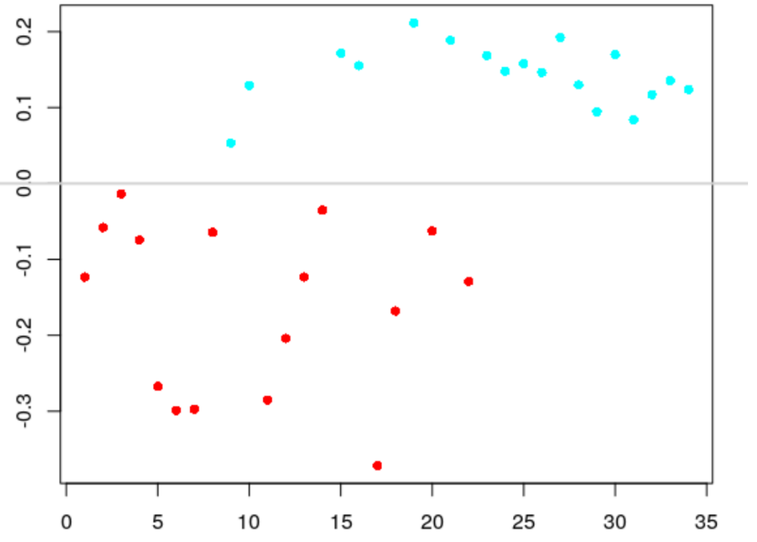
On $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, recall that $\boldsymbol{x}^\mathsf{T}\boldsymbol{L}\boldsymbol{x} = \sum_{i,j}(x_i - x_j)^2$

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.

Spectral Partitionning

```
> 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)
```





Clusters Within Graphs

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 Within Graphs

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.

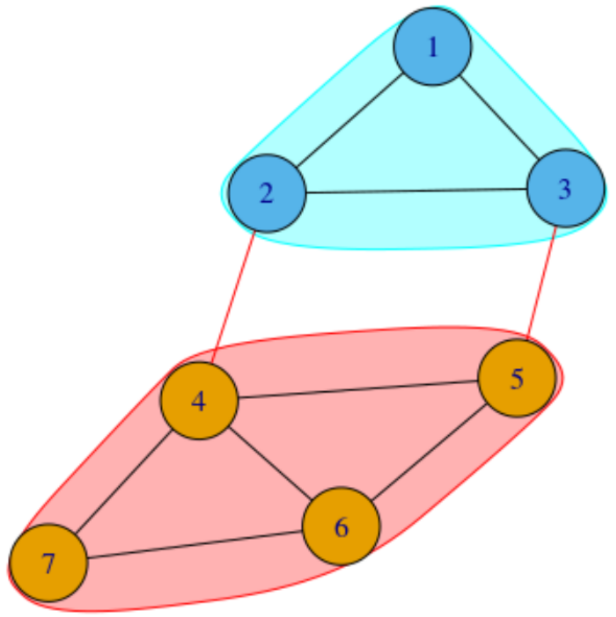
Spin-glass

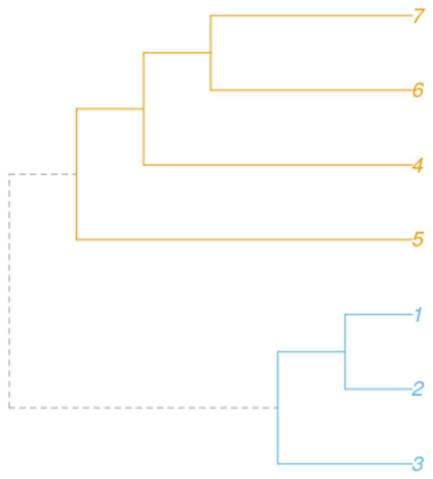
This algorithm uses as spin-glass model (Sherrington & Kirkpatrick (1975)) and simulated annealing to find the communities inside a network.

See Reichardt & Bornholdt (2006)

Clusters Within Graphs

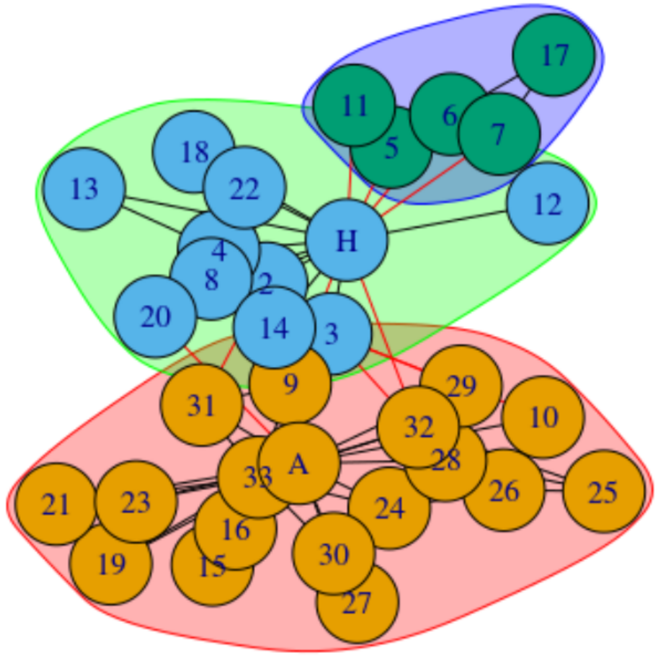
```
> kc = fastgreedy.community(g1)
> length(kc)
[1] 2
> sizes(kc)
Community sizes
1 2
4 3
> membership(kc)
1 2 3 4 5 6 7
2 2 2 1 1 1 1
> plot(kc,g1)
> dendPlot(kc,mode="phylo")
```

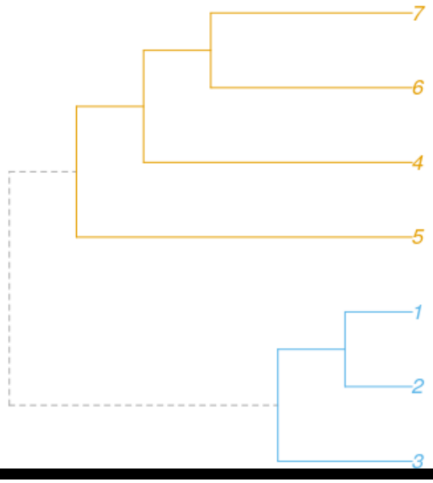




Clusters Within Graphs

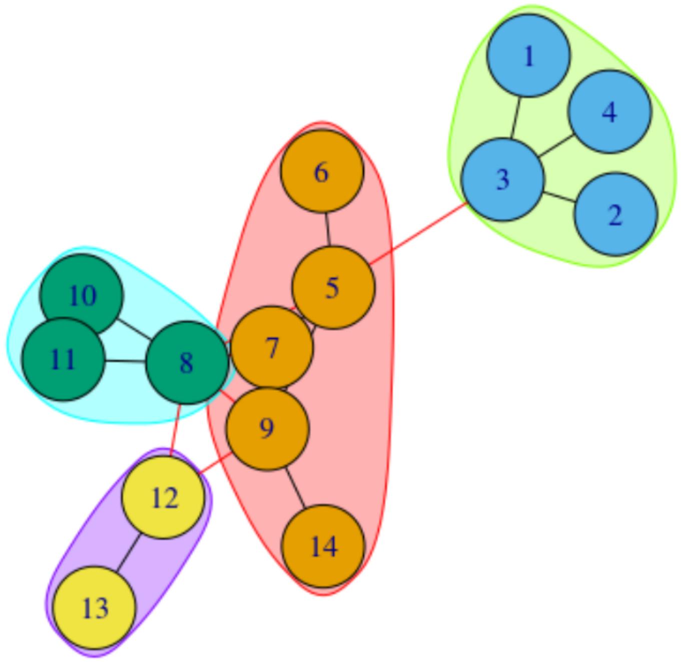
```
> kc <- fastgreedy.community(karate)
> length(kc)
[1] 3
> sizes(kc)
Community sizes
 1  2  3
18 11  5
> plot(kc,karate)
> dendPlot(kc,mode="phylo")
```

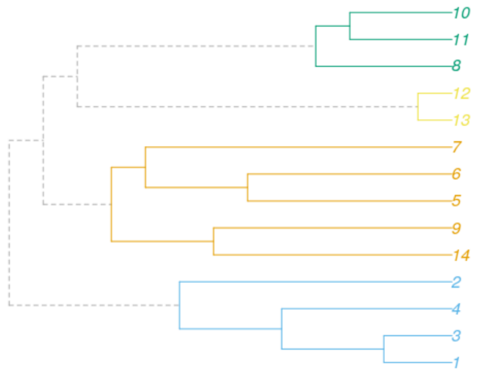




Clusters Within Graphs

```
> 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")
```





Small Word Property

"*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

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

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

Small Word Property

"*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

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

Homophility and Mixing

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.

Homophility and Mixing

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

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

Homophily and Schelling Model

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

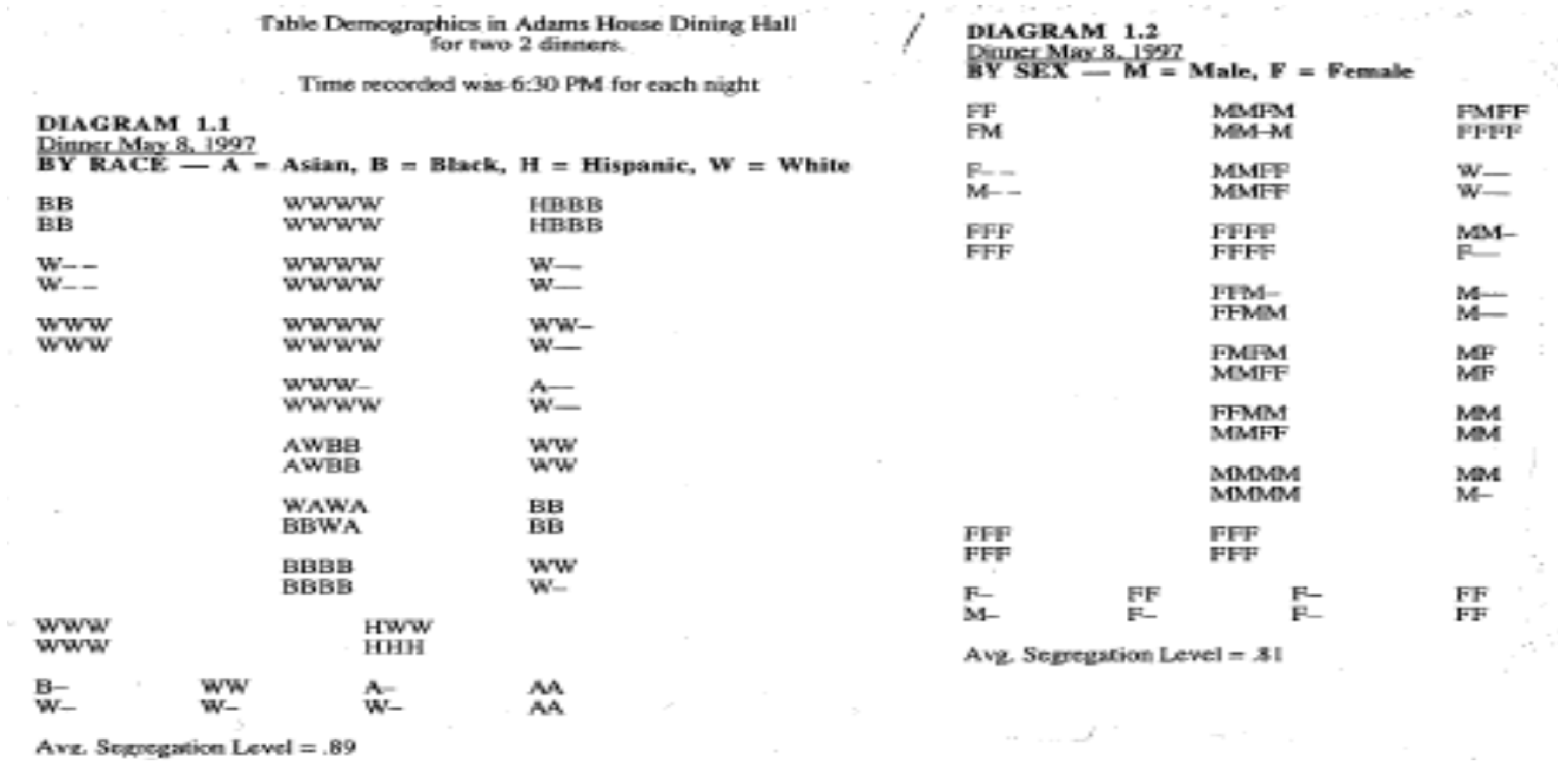


Table Demographics in Adams House Dining Hall
for two 2 dinners.

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              WWWW            HBBB
BB              WWWW            HBBB

W--             WWWW            W—
W--             WWWW            W—

WWW             WWWW            WW-
WWW             WWWW            W—

                WWW-            A—
                WWWW            W—

                AWBB            WW
                AWBB            WW

                WAWA            BB
                BBWA            BB

                BBBB            WW
                BBBB            W-

WWW                     HWW
WWW                     HHH

B-        WW            A-      AA
W-        W-            W-      AA

Avg. Segregation Level = .89
```

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

FF              MMFM            FMFF
FM              MM-M            FFFF

F--             MMFF            W—
M--             MMFF            W—

FFF             FFFF            MM-
FFF             FFFF            F—

                FFM-            M—
                FFMM            M—

                FMFM            MF
                MMFF            MF

                FFMM            MM
                MMFF            MM

                MMMM            MM
                MMMM            M-

FFF             FFF
FFF             FFF

F-        FF        F-          FF
M-        F-        F-          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%).

Homophily and Schelling Model 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).

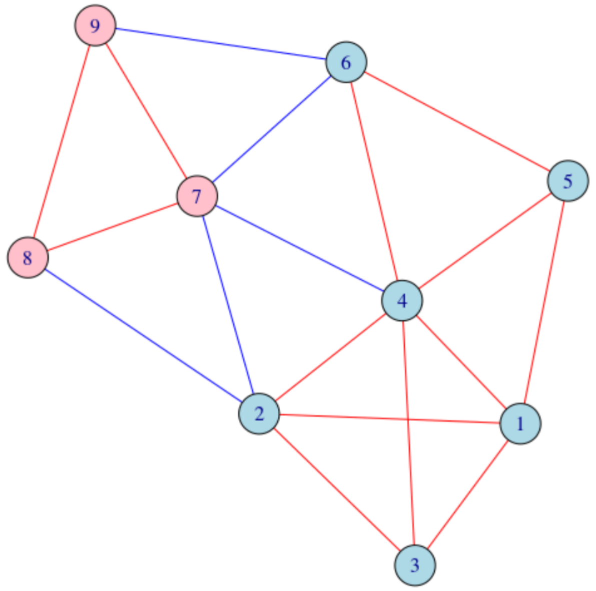
Schelling model generates segregation beyond what people really want, see Chapter 4 in Easley & Kleinberg (2010).

Fixed and unchanging characteristics (gender, ethnicity) can be highly correlated with mutable chaacteristics (location).

Testing for 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 $H_0 : \overline{X} < 2p(1-p)$

Testing for Homophily

```
> 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
> 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
```

Moving Along a Network: Markov Chains

Consider a network (V, E) with n vertices and set $\mathcal{I} = \{1, 2, \cdots, n\}$.

Edges where related to the $n \times n$ adjency matrix $\boldsymbol{A} = [A_{i,j}]$ with $A_{i,j} \in \{0, 1\}$.

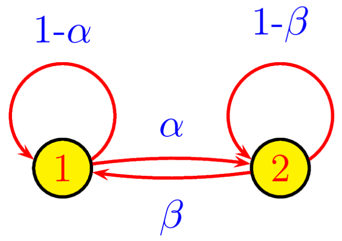
$\boldsymbol{A}_i = [A_{i,j}]$ is the vector of indicators of connexions between vertice i and j.

Instead, consider a transition probability vector $\boldsymbol{P}_i \in \mathcal{S}^{n-1}$ where

$$\mathcal{S}^{n-1} = \left\{ \boldsymbol{u} \in \mathbb{R}_+^n, \text{ such that } \sum_{i=1}^n u_i = 1 \right\}$$

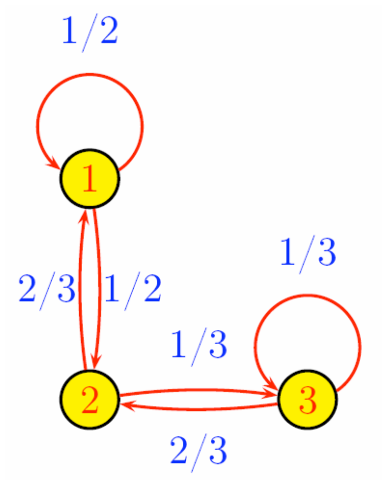
Matrix $\boldsymbol{P} = [p_{i,j}, i, j \in \mathcal{I}]$ is a stochastic matrix, or transition matrix, if each row $\boldsymbol{P}_i = [p_{i,j}, j \in \mathcal{I}]$ is a probability measure on $\mathcal{I}$.

Matrix $\boldsymbol{P} = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$ is a stochastic matrix on the set of vertices $\{1, 2\}$ if $\alpha, \beta \in [0, 1]$.

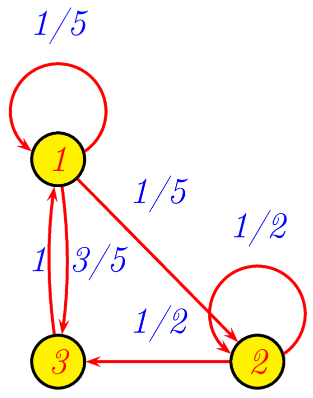


Moving Along a Network: Markov Chains

Matrix $\boldsymbol{P} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 2/3 & 0 & 1/3 \\ 0 & 2/3 & 1/3 \end{pmatrix}$ is a stochastic matrix on the set of vertices $\{1,2,3\}$.

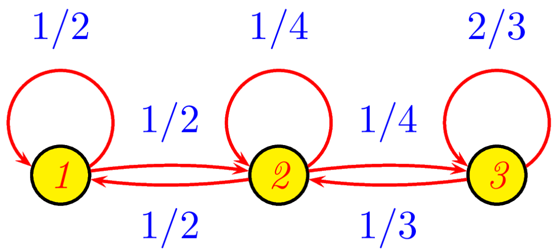


Matrix $\boldsymbol{P} = \begin{pmatrix} 1/5 & 1/5 & 3/5 \\ 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \end{pmatrix}$ is a stochastic matrix



Moving Along a Network: Markov Chains

Matrix $\boldsymbol{P} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/4 & 1/4 \\ 0 & 1/3 & 2/3 \end{pmatrix}$ is a stochastic matrix



Consider a sequence of random variables $(X_n)_{n\in\mathbb{N}}$ on $V = \mathcal{I}$.

This sequence is a Markov Chain if for all $n \geq 1$, and any $i_0, i_1, \ldots, i_{n-1}, i_n, i_{n+1}$ such that $\mathbb{P}(X_0 = i_0, \ldots, X_n = i_n) > 0$, we have

$$\mathbb{P}(X_{n+1} = i_{n+1} | X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \mathbb{P}(X_{n+1} = i_{n+1} | X_n = i_n),$$

so-called memoryless property.

A Markov Chain $(X_n)_{n\in\mathbb{N}}$ is said to be homogeneous if $\mathbb{P}(X_{n+1} = i_{n+1} | X_n = i_n)$ does not depend on n.

Moving Along a Network: Markov Chains

A sequence of random variable on $\mathcal{I}$, $(X_n)_{n\in\mathbb{N}}$ is an homogenous Markov chain with initial probability $\boldsymbol{\lambda}$ (on $\mathcal{I}$) with transition matrix $\boldsymbol{P} = [p_{i,j}]$ if

- X_0 has distribution $\boldsymbol{\lambda}$, i.e. $\mathbb{P}(X_0 = i) = \lambda_i$,
- the probability that $X_{n+1} = j$ given $X_n = i$ is $p_{i,j}$, i.e. $\mathbb{P}(X_{n+1} = j|X_n = i) = p_{i,j}$.

$\boldsymbol{P}$ is the matrix of one-step-ahead transition probabilities.

$\boldsymbol{P}^k$ is the matrix of k-step-ahead transition probabilities.

Hence, if $(X_n)_{n\in\mathbb{N}}$ is a Markov chain with initial distribution $\boldsymbol{\lambda}$ and transition matrix $\mathbb{P}$, then for $n, k \in \mathbb{N}$,

- $\mathbb{P}(X_n = i) = (\boldsymbol{\lambda}\boldsymbol{P}^n)_i$
- $\mathbb{P}(X_{n+k} = j|X_n = i) = p^{(k)}_{i,j} = [\boldsymbol{P}^k]_{i,j}$

Moving Along a Network: Markov Chains

Similarly, one can prove that for any $h = 1, ..., k-1$,

$$\mathbb{P}(X_{n+k}=j|X_n=i)=\sum_{v\in V}\mathbb{P}(X_{n+k}=j|X_{n+h}=v)\times\mathbb{P}(X_{n+h}=v|X_n=i),$$

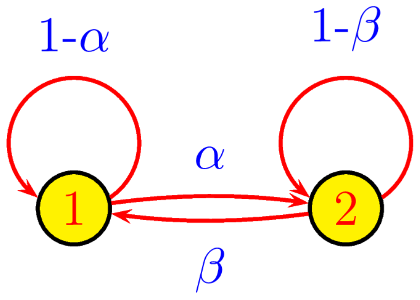
since $\boldsymbol{P}^k=\boldsymbol{P}^h\times\boldsymbol{P}^{k-h}$, or

$$p_{i,j}^{(k)}=\sum_{v\in V}p_{i,v}^{(h)}\times p_{v,j}^{(k-n)}.$$

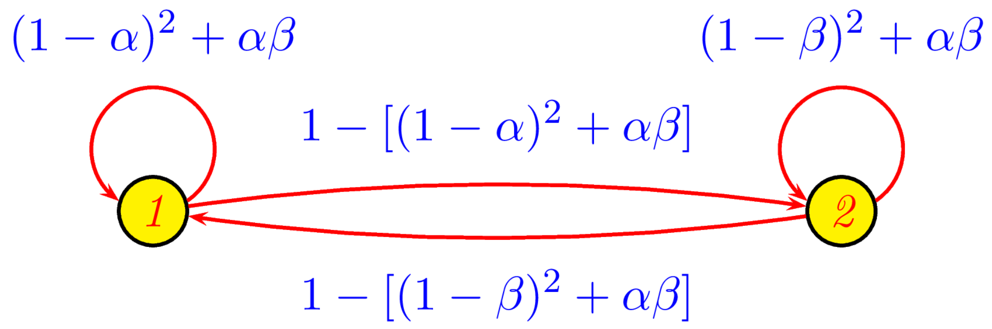
which are the so-called Chapman-Kolmogorov equations.

Moving Along a Network: Markov Chains

Consider transition matrix $\boldsymbol{P} = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$



Then $\boldsymbol{P}^2$ yields


$(1-\alpha)^2+\alpha\beta$
$(1-\beta)^2+\alpha\beta$
$1-[(1-\alpha)^2+\alpha\beta]$
$1-[(1-\beta)^2+\alpha\beta]$
1
2

Moving Along a Network: Markov Chains

Consider a Markov Chain $(X_n)_{n\in\mathbb{N}}$ with initial distribution $\boldsymbol{\lambda}$, and transition matrix $\boldsymbol{P}$, then j is reacheable from i if $\mathbb{P}(X_{n+k}=j|X_n=i)>0$ for some $k\in\mathbb{N}$, denoted$i\rightarrow j$. We will say that i and j are connected, denoted $i\leftrightarrow j$ if $i\rightarrow j$ and $j\rightarrow i$.

Note that $i\rightarrow j$ if there is $k\in\mathbb{N}$ such that $p_{i,j}^{(k)}>0$.

Note that if $i\rightarrow j$ and $j\rightarrow k$, then $i\rightarrow k$.

The set of connected states is related to connected subgraphs.

A class $\mathcal{C}\in\mathcal{I}$ is closed if $i\in\mathcal{C}$ and $i\rightarrow j$ implies that $j\in\mathcal{C}$.

If $\mathcal{I}$ is closed, the chain is irreductible.

A vertice i is said to be absorbing if $\{i\}$ is closed (and $p_{i,i}=1$).

Moving Along a Network: Markov Chains

Consider a Markov chain $(X_n)_{n\in\mathbb{N}}$ with initial distribution $\boldsymbol{\lambda}$ and transition matrix $\boldsymbol{P}$. Given $\mathcal{A}\subset\mathcal{I}$, define $\tau_{\mathcal{A}}$ as

$$\tau_{\mathcal{A}}(\omega)=\inf\{n\geq 0, X_n(\omega)\in\mathcal{A}\}$$

i.e. the time before reaching $\mathcal{A}$ for the first time.

Define

$$p_{\mathcal{A}|i}=\mathbb{P}(\tau_{\mathcal{A}}<\infty|X_0=i).$$

The average time before reaching $\mathcal{A}$ from vertice i is

$$e_{\mathcal{A}|i}=\mathbb{E}(\tau_{\mathcal{A}}|X_0=i)=\sum_{n<\infty}n\mathbb{P}(\tau_{\mathcal{A}}=n)+\infty\mathbb{P}(\tau_{\mathcal{A}}=\infty).$$

Moving Along a Network: Markov Chains

Observe that $\boldsymbol{p}_{\mathcal{A}} = (p_{\mathcal{A}|i})_{i\in I}$ is the (positive) minimal solution of

$$\begin{cases} p_{\mathcal{A}|i} = 1 & \text{if } i \in \mathcal{A} \\ p_{\mathcal{A}|i} = \sum_{j\in I} p_{i,j} \times p_{\mathcal{A}|j} & \text{if } i \notin \mathcal{A} \end{cases}$$

Observe that $\boldsymbol{e}_{\mathcal{A}} = (e_{\mathcal{A}|i})_{i\in I}$ is the (positive) minimal solution of

$$\begin{cases} e_{\mathcal{A}|i} = 0 & \text{if } i \in \mathcal{A} \\ e_{\mathcal{A}|i} = 1 + \sum_{j\notin \mathcal{A}} p_{i,j} \times e_{\mathcal{A}|j} & \text{if } i \notin \mathcal{A} \end{cases}$$

Moving Along a Network: Markov Chains

Let (X_n^x) denote a Markov Chain starting from vertice $x \in \mathcal{I}$. x is said to be

1. transient for $\boldsymbol{P}$ if $\mathbb{P}(T_x < \infty) < 1$,
2. recurrent for $\boldsymbol{P}$ if $\mathbb{P}(T_x < \infty) = 1$.

A distribution $\boldsymbol{\lambda}$ on $\mathcal{I}$ est dite invariant for $\boldsymbol{P}$ if $\boldsymbol{\lambda}\boldsymbol{P} = \boldsymbol{\lambda}$ or $\boldsymbol{\lambda} = \boldsymbol{P}\boldsymbol{\lambda}$.

Let $\boldsymbol{P}$ denote an irreductible stochastic matrix. $\boldsymbol{P}$ has an invariant measure if and only if the is a reccurent vertice.

Let $\boldsymbol{P}$ be an irreductible stochastic matrix, with invariant measure $\boldsymbol{\pi}$. Let $(X_n)_{n\geq 0}$ denote some Markov Chain with transition matrix $\boldsymbol{P}$, and some initial distribution $\boldsymbol{\lambda}$, then

$$\mathbb{P}(X_n = j) \to \pi_j \text{ as } n \to \infty, \text{ for all } j.$$

Moving Along a Network: Markov Chains

```
> alpha=.2; beta= .3
> P = matrix(c(1-alpha,beta,alpha,1-beta),2,2)
> P
      [,1] [,2]
[1,]   0.8  0.2
[2,]   0.3  0.7
> P%*%P%*%P%*%P%*%P%*%P%*%P%*%P%*%P%*%P%*%P%*%P
          [,1]      [,2]
[1,] 0.6000977 0.3999023
[2,] 0.5998535 0.4001465
```

Here the limiting distribution is proportional to $(3, 2)$

Moving Along a Network: Markov Chains

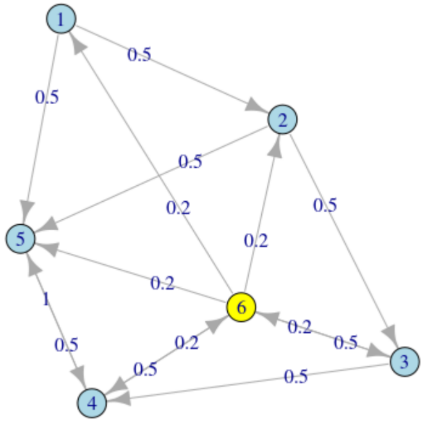
Since $\boldsymbol{\pi} = \boldsymbol{P}\boldsymbol{\pi}$, $\boldsymbol{\pi}$ is an eigenvector of $\boldsymbol{P}$ associated with eigenvalue 1

```
> eigen(t(P))
$values
[1] 1.0 0.5

$vectors
          [,1]       [,2]
[1,] 0.8320503 -0.7071068
[2,] 0.5547002  0.7071068
> eigen(t(P))$vector[,1]/(sum(eigen(t(P))$vector[,1]))
[1] 0.6 0.4
```

Here normalization is $\|\boldsymbol{u}\|_2 = u_1^2 + u_2^2 = 1$ while for probabilities we want $\|\boldsymbol{u}\|_1 = u_1 + u_2 = 1$.

Markov Chains and Internet Web pages



Start with a web page u, chosen randomly.
Then visit randomly (with equal probabilities) pages in the neighborhood of u (i.e. pages linked from u).
E.g. $u = 6$ and consider 5 neighbors.

$$p_{u,v} = \mathbb{P}[X_{n+1} = v | X_n = u] = \frac{1}{d_u} \text{ if } A_{i,j} = 1$$

and 0 otherwise.

The rank of vertice u is the average number of visits on the random walk to u.

$$R_u = \lim_{h\to\infty} \frac{1}{h} \sum_{k=1}^{h} \mathbf{1}(X_{n+k} = u)$$

and consider $\boldsymbol{R} = (R_1, \cdots, R_{n_V})$

Markov Chains and Internet Web pages

In practice, (X_n) starts from $X_0 = u$, and set

$$\widehat{R}_u^{(n)} = \frac{1}{n}\sum_{k=1}^{h}\mathbf{1}(X_k = u)$$

such that $\widehat{R}_u^{(n)} \to \widehat{R}_u$ as $n \to \infty$. Can it be used to find $\text{argmax}\{R_i\}$?

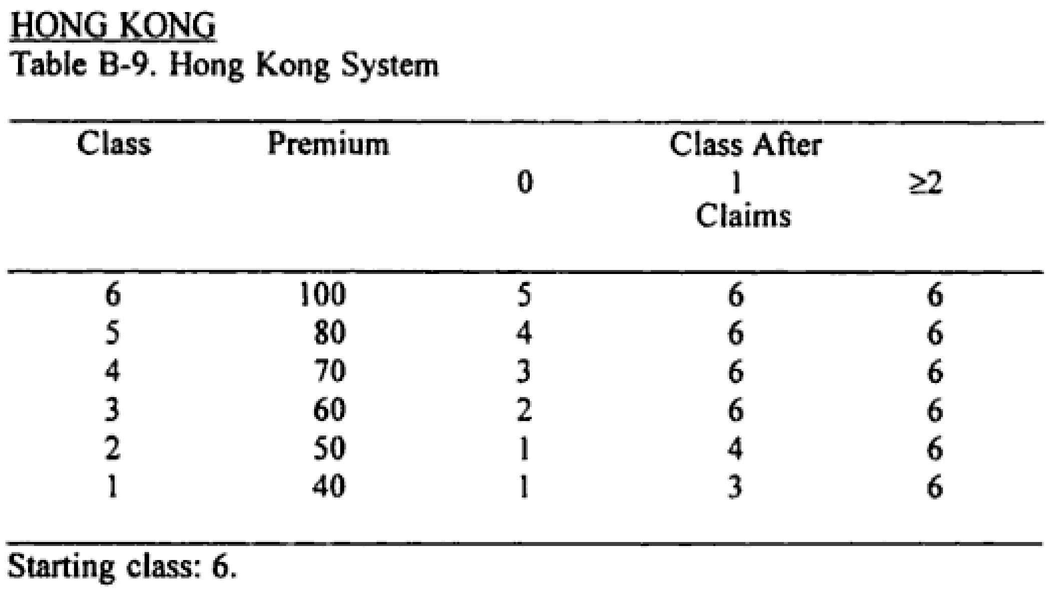
Let $\boldsymbol{P}$ be the transition matrix of the Markov Chain. The stationary measure $\boldsymbol{\pi}$ satisfies

$$\pi_u = \sum_{v\in V} p_{u,v}\pi_v, \text{ with } \|\boldsymbol{\pi}\| = 1$$

Hence, $\boldsymbol{\pi} \propto \boldsymbol{R}$.

Markov Chains and Bonus-Malus

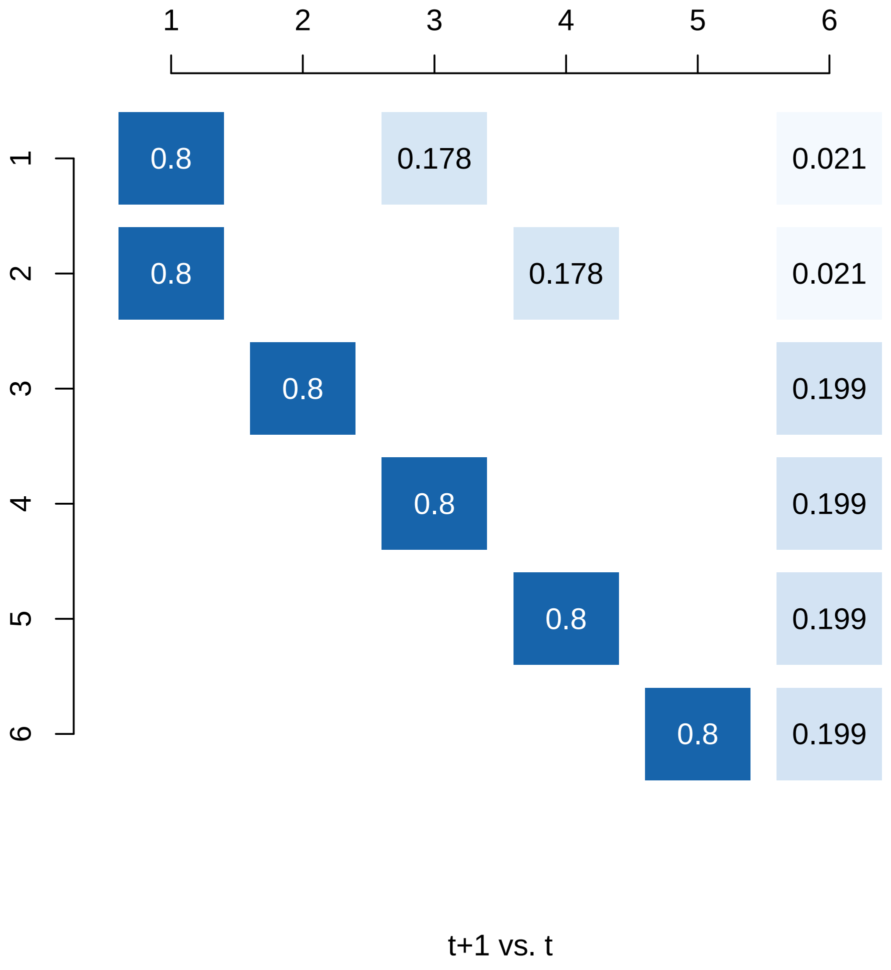
Example no-claim bonus, see Lemaire (1995).



HONG KONG
Table B-9. Hong Kong System

Class	Premium	Class After 0 Claims	Class After 1 Claims	Class After ≥2 Claims
6	100	5	6	6
5	80	4	6	6
4	70	3	6	6
3	60	2	6	6
2	50	1	4	6
1	40	1	3	6

Starting class: 6.



Assume that the probability to claim a loss is 20%.

Weights of Edges

So far, we have considered in the adjency matrix $A_{i,j} \in \{0, 1\}$. We can consider weights, to characterize strength of ties

- qualitative, $\{$weak, strong$\}$ friends
- quantitative (continuous) $\omega_{i,j} \in \mathbb{R}_+$

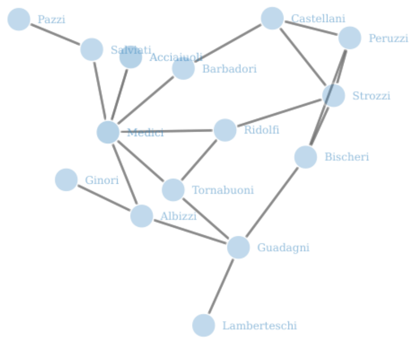
... to be continued.

Practicals: the Padgett Florentine Families

1. Import the data and visualize the network

Remark Use

```
> library(networkD3)
> df=as.data.frame(ends(g, E(g)))
> names(df)=c("src","target")
> simpleNetwork(df)
```



to get a D3.js visualization

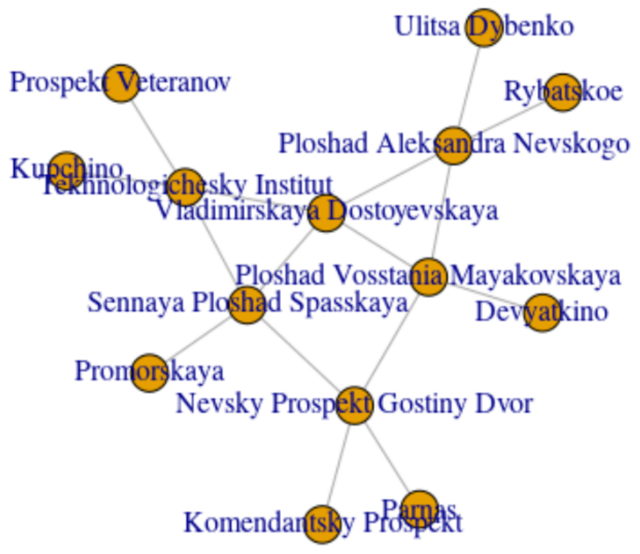
2. Centrality Measures on the network

3. Comparing Centrality Measures with a Hierarchical Classification

Practicals: Robustness of Metro Networks

1. Import metro data from Adjacency.xls

```
> library(xlsx)
> E=read.xlsx(loc,"StPetersburg")
> n=nrow(E)
> nom=as.character(E[3:(n-2),1])
> Adj=E[3:(n-2),(4:ncol(E)-1)]
> Adj[is.na(Adj)]=0
> Adj=as.matrix(Adj)
> colnames(Adj)=rownames(Adj)=nom
```


Ulitsa Dybenko
Prospekt Veteranov
Rybatskoe
Ploshad Aleksandra Nevskogo
Kupchino
Tekhnologichesky Institut
Vladimirskaya Dostoyevskaya
Ploshad Vosstania Mayakovskaya
Sennaya Ploshad Spasskaya
Devyatkino
Promorskaya
Nevsky Prospekt Gostiny Dvor
Komendantsky Prospekt
Parnas

2. Compute Centrality Measures

3. Compute Robustness Measures from The complexity and robustness of metro networks