ADVANCED SOCIAL NETWORKS

UNIVERSITY OF TRENTO
A.Y. 2021/2022

CLASS BY
FILIP AGNEESSENS

CLASS NOTES BY
AURORA MARIA TUMMINELLO

Contents

Part I

Theory

Chapter 1

Introduction

1.1 Social Network Analysis

What is a social network?

It involves not only the existence of a network but also the social component (the presence of more entities interacting with each other).

Is Instagram a social network?

Yes, since it is a network that links people

Is the communication between students a social network?

It is since it represents the link between and among two or more students.

Is a supply chain a social network?

Each role inside the supply chain is linked with the previous and the following

Is the Tokyo train line a social network?

It is a network, but since it does not primarily involve people: it may link them, but mainly geographically and individually.

Is our brain a social network?

It is a network, but not necessarily a social one: it is supposed to be so whenever we interact with external factors.

Can words be part of a social network?

They can be if we trace what people say and link words based on the topic, the person, the place, the context etc. Remember that documents or conversations may be both the context and the node that may be linked to other documents or conversations, based on what we're studying (e.g. communication between two or more individuals, events, etc).

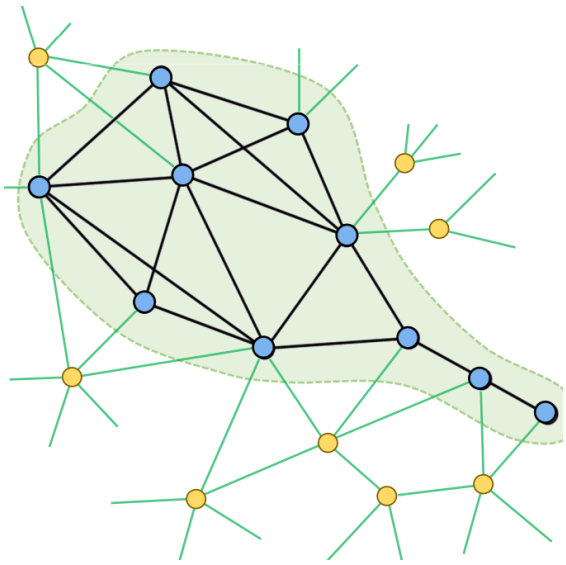
1.1.1 Definition

Networks are a way of thinking about social systems that focus our attention on the relationships among the entities that make up the system, which we call actors or nodes, whose categorical or quantitative traits are called *attributes*. These are the focus of **Network Science**.

The nodes in a network can be almost anything, although when we talk about social networks we normally expect the nodes to be active agents rather than, say, inanimate objects. Most often, nodes are individuals, but can also be collectivities. **Social Network Analysis** focuses on humans and how their actions and mechanisms impact the system of interest.

In a social network, we can find:

- **Nodes** or vertices, entities that are linked together and are studied;
- **Ties** or Edges, which link nodes with one another;
- **Boundary**, a delimited group where some nodes are part of it and some others don't. It's like the criteria of choosing a subset of data inside a much bigger group of nodes and links.



Nodes	Edges	Boundary
Students	Friends/Enemies	All students in a year
Protesters	Communicate	All protesters at an event
Organisations	Collaborate	All organisations in a field
Regions	People move between	All regions in Italy
Countries	Trade	All EU Countries
Words	Used together	All tweets on a specific topic

Table 1.1: Example of possible networks, specifying nodes, edges and the boundary.

1.1.2 Social Network Representation

A usual representation for networks is the **Adjacency matrix**, a square matrix that can be reciprocal (mirrored) or not (different if we swap columns and rows). Generally, self-loops do not make sense, therefore in certain situations, we can just ignore them by inserting a 0. Whenever there is a tie linking two nodes, we insert a 1.

Note that whenever the adjacency matrix is symmetric and squared with respect to the diagonal, we're looking at an undirected network, while if it is not so, we're looking at a directed network.

The value inside the adjacency matrix between a couple of nodes may assume values different from 0 and 1, e.g. the weight of the tie. By adding attributes (e.g. age, gender), we could modify the network representation based on a specific attribute (e.g. colour, size, shape).

A variant of the adjacency matrix A is the distance matrix, which holds the geodesic distances between pairs of nodes (i.e. how many steps are required to a to reach b).

Another representation for networks are graphs, defined as the set of nodes V and edges E.

1.2 Types of networks

1.2.1 Sociocentric or complete network analysis

A complete or **sociocentric network** is such that we detain data about the mutual relationship of every couple of nodes inside the considered network. Normally, an **ego network analysis** belongs to such type of networks, where there is a central node and we're interested in modelling all relationships entertained with that subject.

For instance, a LinkedIn ego network contains a central node and its connected contacts, which may be divided into different categories, based on the context (e.g. university, colleagues, fields of interest etc).

An extension to the ego network approach is to continue iterating and expanding the network, starting from a single node and iterating over its linked nodes (**Snowball sampling**).

1.2.2 Directed and undirected networks

Normally, friendships is an undirected network, since it is mutual, but not all people we define as such will agree with us, since they may have other priorities people to think to. Friendship requires frequent meetings, trust, long-term relationships. It is a concept that requires specific behaviours and expectations.

Undirected networks are such that relationships are mutual and both parts know they're linked.

Directed networks may be represented both by straight lines with two arrows or curved lines with two lines, pointing in opposite directions.

1.2.3 Multiple network relations

We could look at:

- different types of links (e.g. positive and negative), differentiated by the link or node colour;
- how relationships evolve among the same nodes;

1.2.4 Weighted networks

These types of networks have weights or values of the edge, where nodes or links may have different importances from each other.

1.2.5 Two-mode or Bipartite networks

In two-mode networks, there are two types of nodes, where each node is associated with the other type. It's an undirect way to look at connections between people that may be unconscious.

The two-mode network can be projected to a one-mode network, based on their common connections to the other type node.

1.3 Levels of analysis

- **Group Level**

  We look at the entire network, not focusing on a specific node, looking at communities or how far some nodes are from each other.

  Do well-connected networks tend to diffuse ideas faster?

- **Individual Level**

  Focusing on a single node, we look at its connections and position inside the network (e.g. centralization, density). Most node-level network properties are aggregations of dyad-level measurements, as when we count the number of ties that a node has.

Do actors with more friends tend to have stronger immune systems?

- **Dyadic Level**

  We study pairwise relations between actors and ask research questions. The dyad level is the fundamental unit of network data collection, and is the unit with the greatest frequency.

  Do pairs of actors with business ties tend to develop affective ties?

Chapter 2

Basic Measures

With respect to the considered level of analysis, there are some measures that can be computed for each of them. Focusing on centrality means focusing on a node's position in a network, at the individual level. It may be important for bridging other nodes or for the power derived by occupying that position.

2.1 Degree centrality

2.1.1 Directed Network

Consider a direct network, whose adjacency matrix is not symmetric. The in degree indicates the popularity of a node, while the out degree indicates its connection with other nodes.

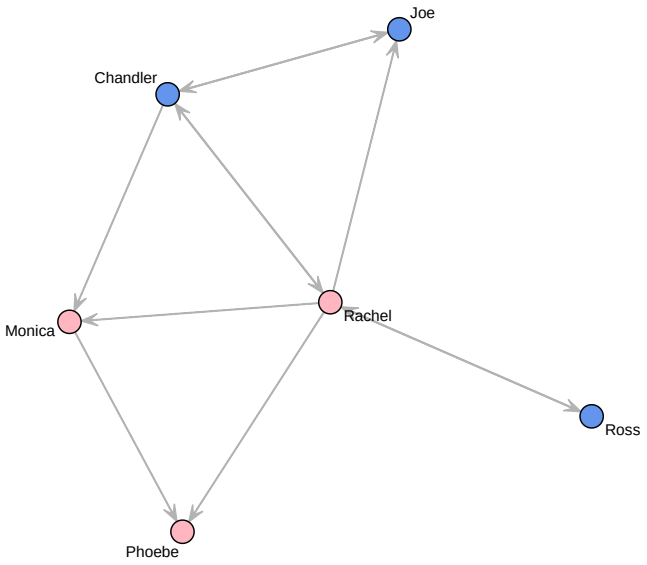


Figure 2.1: Friends directed network

This degree centrality can be normalized to get the proportion of outcoming/incoming connections for each node, by dividing the in/out-degree by the total number of people remaining ($n-1$, or the maximum possible connections that a node could have):

	Ross	*Phoebe*	*Rachel*	*Monica*	*Chandler*	*Joe*	**Out Degree**
Ross	0	0	1	0	0	0	1
Phoebe	0	0	0	0	0	0	0
Rachel	1	1	0	1	1	1	5
Monica	0	1	0	0	0	0	1
Chandler	0	0	1	1	0	1	3
Joe	0	0	0	0	1	0	1
In Degree	1	2	2	2	2	2	

Table 2.1: Nominations in Friends (directed relationship), showing for each person its out and in degree

	Ross	*Phoebe*	*Rachel*	*Monica*	*Chandler*	*Joe*	**Out Degree**
Ross	0	0	1	0	0	0	$1/5 = 0.2$
Phoebe	0	0	0	0	0	0	0
Rachel	1	1	0	1	1	1	$5/5 = 1$
Monica	0	1	0	0	0	0	$1/5 = 0.2$
Chandler	0	0	1	1	0	1	$3/5 = 0.6$
Joe	0	0	0	0	1	0	$1/5 = 0.2$
In Degree	$1/5 = 0.2$	$2/5 = 0.4$	$2/5 = 0.4$	$2/5 = 0.4$	$2/5 = 0.4$	$2/5 = 0.4$	

Table 2.2: Nominations in Friends (directed relationship), showing for each person its out and in degree

2.1.2 Undirected Network

In this case, the in-degree and the out-degree are the same, since we do not distinguish the type of connections. It can be considered as a double arrow pointing to both sides:

$$d_i = \sum_j x_{ij}$$

High degree centrality can be seen as higher risk of receiving whatever flows through the network, since these nodes are more visible and tend to be seen as important.

	Ned	**Marge**	**Homer**	**Abe**	**Maggie**	**Bart**	**Lisa**	**Krusty**	**Bob**	**Cecil**	**Degree**
Ned	0	1	1	1	0	1	0	0	0	0	4
Marge	1	0	1	0	1	0	1	0	0	0	4
Homer	1	1	0	1	1	1	1	0	0	0	6
Abe	1	0	1	0	0	1	0	0	0	0	3
Maggie	0	1	1	0	0	0	1	0	0	0	3
Bart	1	0	1	1	0	0	1	1	0	0	5
Lisa	0	1	1	0	1	1	0	1	0	0	5
Krusty	0	0	0	0	0	1	1	0	1	0	3
Bob	0	0	0	0	0	0	0	1	0	1	2
Cecil	0	0	0	0	0	0	0	0	1	0	1

Table 2.3: Adjacency matrix of the undirected network of The Simpsons.

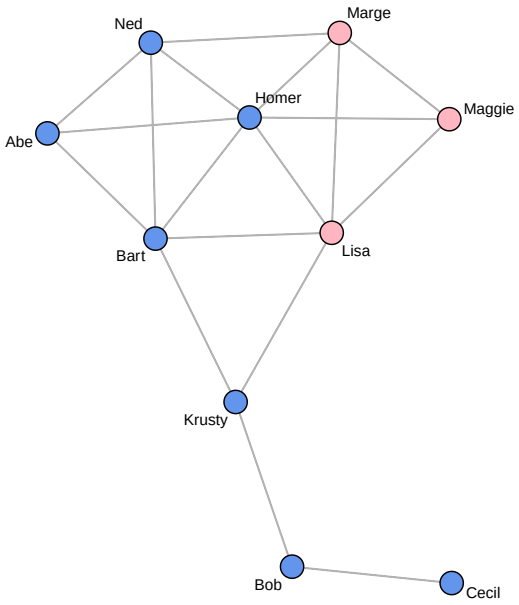


Figure 2.2: Simpsons Undirected Network

2.2 Density and average degree

The simplest measure of cohesion is density, which is the number of ties in the network, expressed as a proportion of the number possible.

2.2.1 Directed Network

$$Density = \frac{\text{Number of ties}}{\text{Number of possible ties}} = \frac{\text{Number of ties}}{n(n-1)}$$

The number of possible ties, without considering self-loops, equals $n(n-1)$. The number of ties equals the sum of the adjacency matrix.

For instance, by considering the Friends network, we can say that there are 11 ties and 6 people, so the density equals $D = 11/(6 \cdot 5) = 11/30 = 0.367$.

This number indicates the percentage of direct connections between nodes inside the network.

2.2.2 Undirected Network

In this case, the number of ties can be obtained by doubling the sum of half-matrix, since it is symmetric. The number of possible ties in this case is divided by two, since it is symmetrical and $(a, b) = (b, a)$:

$$Density = \frac{\text{Number of ties}}{\text{Number of possible ties}} = \frac{\text{Number of ties}}{\frac{n(n-1)}{2}}$$

In the Simpsons example, where there are 18 edges and 10 nodes, the density equals $D = 18/(10 \cdot 9/2) = 18/45 = 0.4$.

2.2.3 Average Degree

Since the density may assume a different meaning if we compare small and big size networks, the average degree is usually used in substitution to the density. It represents the average number of ties each node has.

Related to directed network, the average in-degree equals the sum of the in-degree or out-degree over the total number of nodes inside the network:

In the Friends example, the average in-degree is $11/6 = 1.833$, while the out-degree is $11/6 = 1.833$.

The normalized degree of each node is computed by dividing the in and out-degree by $n - 1$.

The normalized in degree and the out degree will give in the end the same sum. In particular, their mean equals the density of the entire network: in the density computation, total edges are divided by $n(n - 1)$, while in the normalized degree, everything is normalized by $n - 1$ and then the average requires another division over n,

2.2.4 Importance of density

The **density** is useful whenever we need to compare networks, whether same or different sizes. In particular, the bigger, the more important density becomes to evaluate its cohesiveness.

Notice that a bigger network makes it harder to have a bigger density than a smaller one. Comparing the density of networks of different sizes can be challenging. It would be better to compare average degrees but notice that **by increasing the network size and the average degree, the density reduces.**

2.3 Degree-centralization

Centralization is a question about how central some nodes are compared to other nodes and therefore can be considered as a characteristic of the network itself. It is related to the distribution of the centrality inside the network, so to the extend a network is dominated by a single node; while centrality is more related to the individual sphere of the nodes.

The least centralized network is a ring structure, where everyone is connected to the previous and following node (in this case the centralization is 0). The most centralized network is an ego one, where one node is connected to everyone and no one else is connected to nodes different from the ego (in this case the centralization measure is far from 0).

Centralization may indicate the status of some nodes inside a network.

2.3.1 Undirected Network

By taking the maximum degree inside the network, for each node, we subtract its degree to this maximum, in order to get its centralization. The centralization for the degree is then obtained by summing all these centralizations.

Note that the maximum number of connections we could build in a centralized network is $n - 1$. In a decentralized network, if everybody has the same number of connections, the centralization will always be 0, since the maximum will be the same for every node.

By considering the range between the least and the most centralized (i.e. $[0, 12]$), the centralization can be finally computed by dividing the centralization for the degree by the maximum centralization value possible.

Considering the three networks in the image, the centralization in the first one is $5/12$, *while the other two represent respectively the minimum and the maximum centralization obtainable.*

The general formula for the centralization is the following:

$$Centralization = \frac{\sum(\max - degree_i)}{(n-1)(n-2)}$$

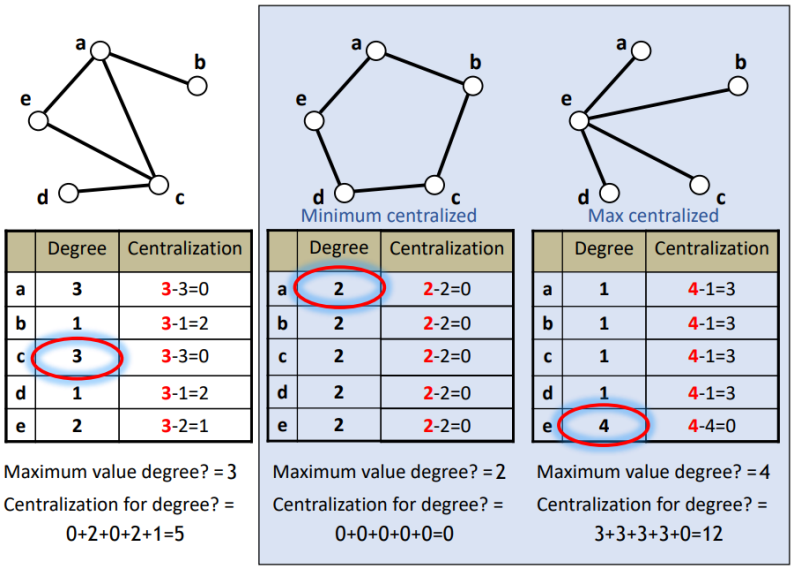


Figure 2.3: Three examples of centralization (random network, least and most centralized ones

2.3.2 Directed Network

In the case of a directed network, we detain both in and out-degree centralization.

$$In - Centralization = \frac{\sum(\max_{in} - \text{in-degree}_i)}{(n-1)^2}$$

$$Out - Centralization = \frac{\sum(\max_{out} - \text{out-degree}_i)}{(n-1)^2}$$

Still, if we consider Friends' network, we notice that related to the out-degree centralization, the network is kind of centralized on Rachel, which is talkative with everybody:

$$Out - C = \frac{(5-1)+(5-0)+(5-1)+(5-5)+(5-3)+(5-1)}{5^2} = \frac{19}{25} = 0.76$$

Related to the in-degree instead, the network is decentralized. By thinking about how is more popular than other characters, nobody is the true main star of the show, despite Ross is quite less cited by other people. This means that everybody is as popular as everybody else. It is not perfect yet actually, because of Ross.

$$In - C = \frac{(2-1)+5*(2-2)}{5^2} = \frac{1}{25} = 0.04$$

2.3.3 Experiment of Bavelas and Leavitt

How fast a group could make decisions together?

Bavelas and Leavitt conducted a series of experiments at the MIT. They aim to study the best communication structure for group problem-solving. Each subject was given a card, with 5 symbols from a set of 6. Each symbol appeared on 4 of the 5 cards, except for one. The group was asked to find the common symbol in the shortest time possible.

The subjects communicated by writing messages that could be passed through slots in the walls of the cubicles. Four structures were proposed:

- **ring structure:** no consistent pattern of organisations, since subjects sent messages only when receiving or work out the answer themselves;
- **straight line:** most of the time the answer was sent out by the individual in the most central position;
- **cross figure:** the peripheral sent information to the center, where answered arrived and were sent out.
- **t-figure:** the most central figure got all the information and sent the answer. Slower than the cross, but still working.

The most efficient structure is the T-figure. During the experiment, the person in the middle in the cross figure didn't understand the experiment at all, therefore the information was useless. It's the classical structure with the team leader that commands the actions to take. It actually depends on the leadership ability of the central node, otherwise there is no learning.

The efficiency is given by the velocity and availability to communicate. Simple things benefit from the decentralized problems, whereas complex problems benefit from the centralized option, since the divide et impera organisation makes everything more efficient.

Evaluation:

- satisfaction: ring structure;
- less errors: T-figure;
- like: ring-structure;
- at least one error: cross and T-figure

Aim: find a balance between efficiency and satisfaction.

Chapter 3

Cohesion measures

How can we compute whether a network is cohesive or not?

When we talk about cohesion, we're focusing on the **group level of analysis**.

3.1 Direct connections

3.1.1 Density

What proportion of ties is present? What proportion of pairs of nodes is directly connected?

It's the simplest measure of cohesion, the number of ties in the network, expressed as a proportion of the number possible.

Given an undirected network, its number of possible ties is $n(n-1)/2$:

$$Density = \frac{\text{Number of ties}}{\text{Number of possible ties}}$$

For instance, in this TBBT network, the density is 1/4 (7 actual ties, with 8 nodes, so $8*7/2 = 28$ possible ties).*

3.1.2 Average Degree

What is the average number of connections for nodes?

It is computed as the sum of all ties each node has, divided over the total number of nodes n.

The average degree of connections for nodes in the TBBT network is: $(1+2+1+1+2+2+2+2)/8 = 1.625$.

Remember that density does not tell us how these ties are distributed among the nodes.

3.2 Components

How many different groups of nodes exist that cannot reach each other (directly or indirectly)?

A component is a maximal set where every node can reach every other by some path. Maximal indicates that if you can add a node to the set without violating the condition that everyone can reach everyone, than it must be so. Weak components are a variant of components, where, if we disregard the direction of edges, we will find components; on the other side, strong components are such that even if we respect directions every node is connected to every other.

In the Big Bang Theory network, the components are two, as shown in the image.

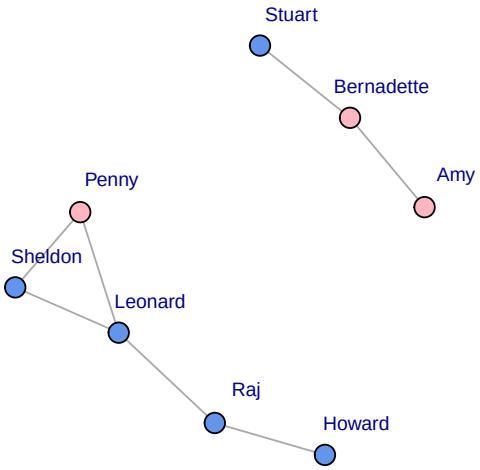


Figure 3.1: The Big Bang Theory network

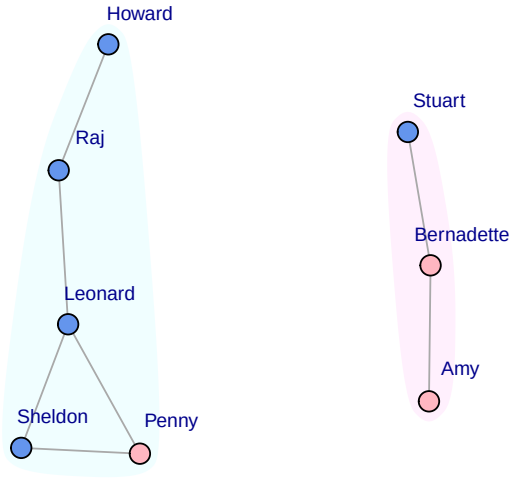


Figure 3.2: Components in the Big Bang Theory Network

Imagine that everyone has a piece of information that needs to be propagated. The more components, the less information everyone is reaching and the worse it is. The number of components of a network goes from 1 to the total number of nodes n. We should think of components as a way to spread creativity inside the network. All groups hold unique ideas and being one component allows for exposure to more ideas.

3.2.1 Component ratio

The component ratio is given by the number of components minus 1, over the maximum number of components minus 1:

$$CR = \frac{c-1}{n-1}$$

In TBBT example, the component ratio would be $(2-1)/(8-1) = 1/7$*, given the* 2 *components and* 8 *total nodes.*

1 means that all nodes are isolated, while 0 means there's one single component. In order to get cohesion, this component ratio can be subtracted to 1.

Notice that the component ratio does not tell us anything about the distribution of these components. The size of the largest component is a function of the network size, but it doesn't tell us anything about the smaller components. This happens because it is not very sensitive.

3.2.2 Connectedness and fragmentation

Connectedness and fragmentation are more sensitive than the component ratio instead. Connectedness is the proportion of pairs of nodes that can reach each other by a path of any length (i.e. proportion of pairs of nodes located in the same component).

The two networks below are similar to the largest components, but not the same.

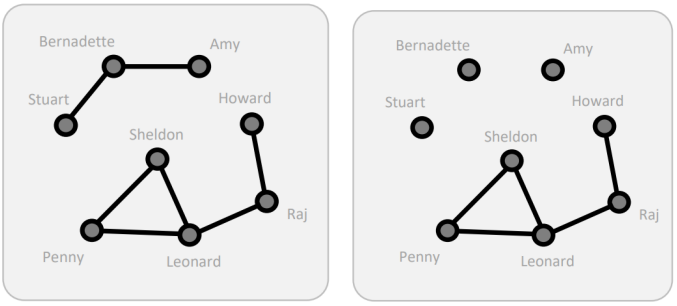


Figure 3.3: Examples of possible arrangements of the TBBT network

What proportion of pairs of nodes can reach each other directly or indirectly?

In the first example, Amy, Bernadette and Stuart reach each other $(3 \cdot 2)$*, whereas the others* 5 *are connected among them* $(5 \cdot 4)$ *, so* $(3 \cdot 2 + 5 \cdot 4)/(8 \cdot 7) = 0.464$*. In the second example, since the first three nodes do not reach each other, we have* $(0 + 0 + 0 + 5 * 4)/(8 * 7) = 0.357$*.*

This proportion is called **connectedness**, while its complement is called **fragmentation** (fragmentation $= 1 -$ connectedness). Notice that these measures do not inform about how long it takes to reach other nodes inside the same component:

$$Connectedness = \frac{\sum_{i=\text{number of nodes in ith component}} i(i-1)}{n(n-1)}$$

$$Fragmentation = 1 - Connectedness$$

3.3 Geodesic

3.3.1 Compactness and breadth

We can study the average distance between two nodes, based on cohesion measures. Remember that longer distances reduce information sharing since it is quite expensive to reach them. The lower the average distance, the easier it is to reach other nodes and therefore the more the network is cohesive and preferable. Unfortunately, average geodesic distance and its variants cannot be applied to disconnected graphs, were some distances are not defined (i.e. Inf).

For instance, the distance from Howard to Leonard is 2. If we want to compute the average distance between any couple of nodes inside the TBBT network, we need to consider disconnected nodes with inf distance, as shown in the geodesic distance matrix, which shows the minimum number of links to pass in order to reach a specific node.

```
geodist(tbbt)$gdist
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    1    2  Inf  Inf  Inf  Inf  Inf
## [2,]    1    0    1  Inf  Inf  Inf  Inf  Inf
## [3,]    2    1    0  Inf  Inf  Inf  Inf  Inf
## [4,]  Inf  Inf  Inf    0    1    1    2    3
## [5,]  Inf  Inf  Inf    1    0    1    2    3
## [6,]  Inf  Inf  Inf    1    1    0    1    2
## [7,]  Inf  Inf  Inf    2    2    1    0    1
## [8,]  Inf  Inf  Inf    3    3    2    1    0
```

By convention, Inf is expressed as 0 if we consider 1/geodesic distance, also called **reciprocal distance**:

```
round(1/(geodist(tbbt)$gdist),3)
```

```
##      [,1] [,2] [,3]  [,4]  [,5] [,6] [,7]  [,8]
## [1,]  Inf    1  0.5 0.000 0.000  0.0  0.0 0.000
## [2,]  1.0  Inf  1.0 0.000 0.000  0.0  0.0 0.000
## [3,]  0.5    1  Inf 0.000 0.000  0.0  0.0 0.000
## [4,]  0.0    0  0.0   Inf 1.000  1.0  0.5 0.333
## [5,]  0.0    0  0.0 1.000   Inf  1.0  0.5 0.333
## [6,]  0.0    0  0.0 1.000 1.000  Inf  1.0 0.500
## [7,]  0.0    0  0.0 0.500 0.500  1.0  Inf 1.000
## [8,]  0.0    0  0.0 0.333 0.333  0.5  1.0   Inf
```

Excluding the diagonal, the **compactness** is expressed as the average of the reciprocal distance. It weights the paths connecting nodes inversely by their length. **Breadth** is the complement of compactness (1 − compactness):

$$Compactness = \frac{\sum_{i \neq j}(1/d_{ij})}{n(n-1)}$$

$$Breadth = 1 - Compactness$$

3.3.2 K-step approach

The k-step approach tries to find *what proportion of pairs of nodes can reach each other in k steps.* It is a distance-based measure of cohesion.

For instance, given the geodesic distance, what is the proportion of nodes that can reach each other in less than 3 steps?

```
# Adjacency matrix of how many steps are necessary to reach other nodes
steps_3 = geodist(tbbt*tbbt*tbbt)$gdist
```

```r
# Counting the undirected edges that reach other nodes in less than 3 steps
less_then_3_steps = sum(steps_3 < 3 & steps_3 > 0)/2
n = 8
# Proportion of nodes that reach each other in less than 3 steps
less_then_3_steps/(n*(n-1)/2)
```

```
## [1] 0.3928571
```

3.4 Exercises

3.4.1 Comment on the results obtained

- Density

  Density assumes the same value for all the networks.

- Connectedness

  In networks composed of one single component, obviously, everyone is connected with everyone else, promoting easy information sharing.

  Within some components, for instance in network 6 there's a completely connected component between Stuart, Sheldon, Penny and Leonard, which reduces the cost of information sharing.

- Compactness

  The difference between the two networks with one component, we notice that in terms of compactness the fourth takes fewer steps to reach every other node in the network. If we consider the 2 steps reach in fact, we notice that it is higher in network 4 than in 3.

Considering the money to spend on the process of information sharing. Which network would be more helpful?

In this case, components play a key role in terms of reachability. So networks as 4 and 3 are preferable.

In terms of information sharing and studying for an exam to gain a certain competence?

Direct connections are more helpful than greater components. It's hard to transfer knowledge from multiple people to one.

When thinking to find a resource online?

It's probably better to request it from multiple people that suggest more courses as possible.

Chapter 4

Centrality

4.1 Degree Centrality

The aim of degree centrality is to find the node that reaches as many people as possible in one single step.

4.2 Closeness Centrality

How to discover who's the most central node in the network? If you have information that you want everyone in the group to have, and you can only give it to one person in the group, who would you give it to?

Every extra step needed to reach someone decreases accuracy and increases costs. A **geodesic** is the shortest path between two actors. Geodesics are not necessarily unique, there could be multiple paths of equally short length. A long geodesic distance implies that, even under the very best conditions, it would be a long time before something gets from one node to the other.

In the following subsections, two versions of the closeness centrality will be presented: Freeman's approach, which is about reaching everyone in the minimum number of steps; the reciprocal approach, which is about reaching the highest number possible in a few steps.

4.2.1 Freeman's closeness centrality

Freeman defines closeness centrality as the sum of geodesic distances from a node to all others. Large numbers indicate that a node is highly peripheral, while small numbers indicate that a node is more central. We start with a person that has the information and in one step we count how many of them have been reached starting from him/her.

Consider the following example where we start from Marge and try to reach for every other node in the network.

Some nodes may be equivalent, especially when they're interchangeable.

By considering the final table obtained by counting the necessary steps to reach any other node starting from a specific one, we obtain the table below. We can notice how Homer and Krusty are equivalent since they reach in the same number of steps all the nodes in the network, but Lisa and Bart minimize the cost with equivalent connections. On the other side, Cecil maximizes the path cost to reach all the other nodes.

These values can be normalized by computing the reciprocal, multiply for the minimum value possible: $n-1$, which equals 9 in the Simpsons' example.

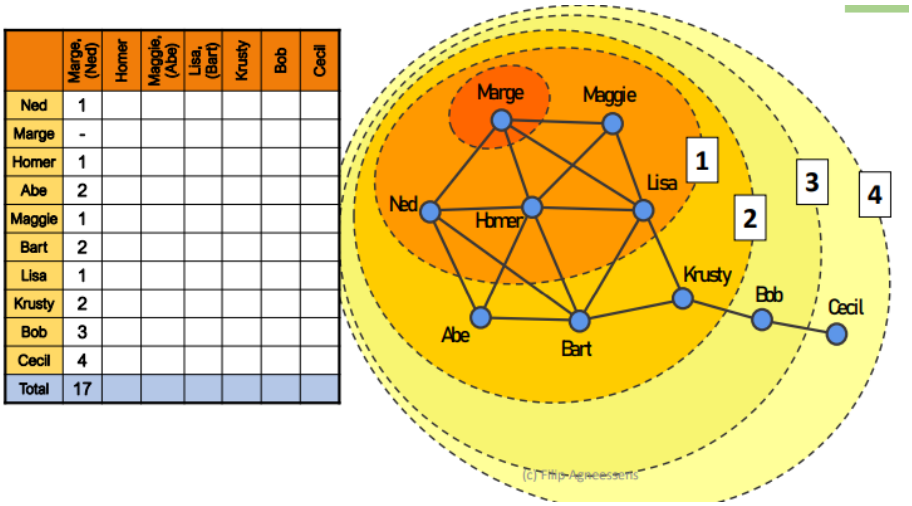


Figure 4.1: Example of Freeman Closeness centrality

	Marge, (Ned)	Homer	Maggie, (Abe)	Lisa, (Bart)	Krusty	Bob	Cecil
Ned	1	1	2	2	2	3	4
Marge	-	1	1	1	2	3	4
Homer	1	-	1	1	2	3	4
Abe	2	1	2	2	2	3	4
Maggie	1	1	-	1	2	3	4
Bart	2	1	2	1	1	2	3
Lisa	1	1	1	-	1	2	3
Krusty	2	2	2	1	-	1	2
Bob	3	3	3	2	1	-	1
Cecil	4	4	4	3	2	1	-
Total	17	15	18	14	15	21	29
Closeness	9/17	9/15	9/18	9/14	9/15	9/21	9/29
	0.53	0.60	0.50	0.64	0.60	0.42	0.31

Figure 4.2: Example of Freeman Closeness centrality

Name	Closeness
Ned	0.529
Marge	0.529
Homer	0.6
Abe	0.5
Maggie	0.5
Bart	0.643
Lisa	0.643
Krusty	0.6
Bob	0.429
Cecil	0.31

The lower the closeness, the less central a node is; the higher the closeness, the more central a node becomes. This can be computed through the following command on R:

```
library(sna)
Simpsons_n = as.matrix(read.csv("datasets/simpsons.csv",
                                row.names=1,
                                stringsAsFactors = F))
closeness = sna::closeness(Simpsons_n, gmode="graph")
```

What happens if we add multiple components?

The other nodes will never be reached, therefore it is counted as Inf (i.e. undefined), but it shouldn't be summed up to other steps, or every node will be equivalently central. Considering the code, we would obtain:

```
Simpsons_n2<-Simpsons_n
# Remove Cecil edges
Simpsons_n2[9,10]<-0
Simpsons_n2[10,9]<-0
sna::closeness(Simpsons_n2, gmode="graph")
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

Whoever we choose, will never reach everyone. But suppose we set a maximum cost for reaching a node, despite they're not connected and cannot reach each other. In this case, we can normalize as before and choose the highest value, remembering that still we cannot reach everyone.

4.2.2 Reciprocal closeness centrality

To overcome the limitation of the previous centrality measure whenever we have more components, we change the aim.

If you have information that you want as many people as possible in the group to have, and you can only give it to one person in the group, who would you give it to?

By using the reciprocal distances (1/geodesic distance), then unreachable nodes value $1/\infty$, therefore 0.

```
# Hand computing the reciprocal closeness
rec_geo = 1/geodist(Simpsons_n2)$gdist
rec_geo = replace(rec_geo,rec_geo == Inf,0)
n = nrow(rec_geo)-1
rec_clo = colSums(rec_geo)/n
```

Considering this centrality measure, the most central node is Homer, since there is not a single component anymore, but two separated ones and Lisa/Bart took advantage of the fact that they

Name	Closeness
Ned	0.6481481
Marge	0.6481481
Homer	0.7592593
Abe	0.5925926
Maggie	0.5925926
Bart	0.7222222
Lisa	0.7222222
Krusty	0.6111111
Bob	0.4074074
Cecil	0.0000000

could have reached both sides of the network, while now they can't.

	Marge, (Ned)	Homer	Maggie, (Abe)	Lisa, (Bart)	Krusty	Bob	Cecil
Ned	1/1	1/1	1/2	1/2	1/2	1/3	1/Inf
Marge	-	1/1	1/1	1/1	1/2	1/3	1/Inf
Homer	1/1	-	1/1	1/1	1/2	1/3	1/Inf
Abe	1/2	1/1	1/2	1/2	1/2	1/3	1/Inf
Maggie	1/1	1/1	-	1/1	1/2	1/3	1/Inf
Bart	1/2	1/1	1/2	1/1	1/1	1/2	1/Inf
Lisa	1/1	1/1	1/1	-	1/1	1/2	1/Inf
Krusty	1/2	1/2	1/2	1/1	-	1/1	1/Inf
Bob	1/3	1/3	1/3	1/2	1/1	-	1/Inf
Cecil	1/Inf	1/Inf	1/Inf	1/Inf	1/Inf	1/Inf	-
Total	5.833	6.833	5.333	6.5	5.5	3.667	0

Closeness	5.833 /9	6.833 /9	5.333/ 9	6.5 /9	5.5 /9	3.667 /9	0
	0.648	0.759	0.593	0.722	0.611	0.407	0

Figure 4.3: Reciprocal Closeness in the Simpsons network where Cecil has no edges

```
# Remove the link between Bob and the rest of the network
Simpsons_n2<-Simpsons_n
Simpsons_n2[8,9]<-0
Simpsons_n2[9,8]<-0

rec_geo = 1/geodist(Simpsons_n2)$gdist
rec_geo = replace(rec_geo,rec_geo == Inf,0)
n = nrow(rec_geo)-1
rec_clo = colSums(rec_geo)/n
```

By changing the network structure, separating both Bob and Cecil, we get different interpretations of this centrality measure: Bob reduces its closeness, while Cecil augments it for being connected to someone. Other nodes increase their closeness since they're more connected to the people inside their component.

Name	Closeness
Ned	0.6111111
Marge	0.6111111
Homer	0.7222222
Abe	0.5555556
Maggie	0.5555556
Bart	0.6666667
Lisa	0.6666667
Krusty	0.5000000
Bob	0.1111111
Cecil	0.1111111

	Marge, (Ned)	Homer	Maggie, (Abe)	Lisa, (Bart)	Krusty	Bob	Cecil
Ned	1/1	1/1	1/2	1/2	1/2	1/Inf	1/Inf
Marge	-	1/1	1/1	1/1	1/2	1/Inf	1/Inf
Homer	1/1	-	1/1	1/1	1/2	1/Inf	1/Inf
Abe	1/2	1/1	1/2	1/2	1/2	1/Inf	1/Inf
Maggie	1/1	1/1	-	1/1	1/2	1/Inf	1/Inf
Bart	1/2	1/1	1/2	1/1	1/1	1/Inf	1/Inf
Lisa	1/1	1/1	1/1	-	1/1	1/Inf	1/Inf
Krusty	1/2	1/2	1/2	1/1	-	1/Inf	1/Inf
Bob	1/Inf	1/Inf	1/Inf	1/Inf	1/Inf	-	1/1
Cecil	1/Inf	1/Inf	1/Inf	1/Inf	1/Inf	1/1	-
Total	*5.5*	*6.5*	*5*	*6*	*4.5*	*1*	*1*
Closeness	*5.5/9*	*6.5/9*	*5/9*	*6/9*	*4.5/9*	*1/9*	*1/9*

Figure 4.4: Reciprocal closeness with Bob and Cecil as separated component

```
Simpsons_n3<-Simpsons_n
Simpsons_n3[8,9]<-0
#Simpsons_n3[9,8]<-0

df = data.frame(
    "Name" = rownames(Simpsons_n),
    `Reciprocal Closeness Centrality` = sna::closeness(Simpsons_n3,
                                                       gmode="graph",
                                                       cmode="suminvundir"))
```

4.3 Betweenness Centrality

Who is important as an "in-between" person to transfer information?

This centrality is about the gatekeepers between two nodes that gain power from this situation, in terms of potential for controlling flows through the network. The more nodes in between, the more extra points are earned to connect other nodes. Betweenness centrality is a measure of how

Name	Reciprocal Closeness Centrality
Ned	0.6759259
Marge	0.6759259
Homer	0.7870370
Abe	0.6203704
Maggie	0.6203704
Bart	0.7592593
Lisa	0.7592593
Krusty	0.6666667
Bob	0.5185185
Cecil	0.3796296

often a node falls along the shortest path between two other nodes:

$$b_j = \frac{\sum_{i<k} g_{ijk}}{g_{ik}}$$

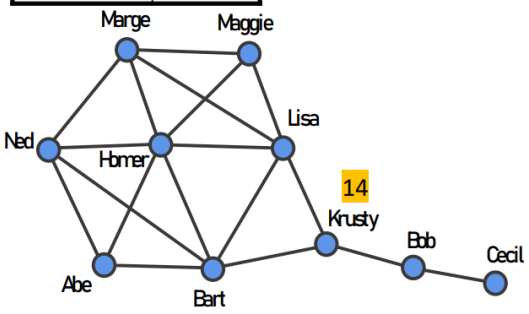
with g_{ijk} being the number of geodesic paths connecting i and k through j and g_{ik} the total number of geodesic paths connecting i and k. Whenever a node is an isolate or whenever every alter it has is connected to every other alter, the betweenness falls to 0.

Consider Lisa as the in-between person we're interested in. She's important whenever she connects all nodes except for Abe to the line composed of Krusty, Bob and Cecil.

When there are alternative paths to get from A to B, then the importance of every node in-between diminishes, because it is split among all in-between nodes (alternatives). For non-unique shortest paths, each actor gets 1/number of shortest paths.

Related to the Simpsons' example, we can compute the betweenness centrality for each character and then find out that Krusty has the highest betweenness centrality since without him there are no alternative paths for reaching Bob and Cecil and vice-versa.

Betweenness	Values?
Ned	0.833
Marge	0.833
Homer	3.667
Abe	0.000
Maggie	0.000
Bart	8.333
Lisa	8.333
Krusty	14.000
Bob	8.000
Cecil	0.000



In the first example, there are these three nodes that can be used as alternatives. In the second example, there are three paths from A to B, which comprehend in between nodes C, D, E, F and G.

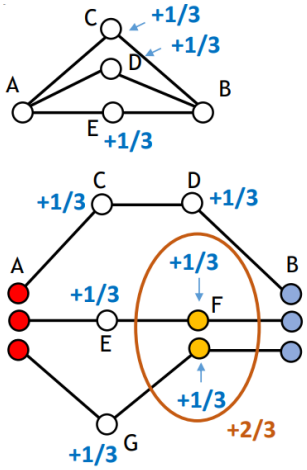


Table 4.1: Betweenness for Simpson's network

Name	Betweenness	Normalized Betweenness
Ned	0.833	0.023
Marge	0.833	0.023
Homer	3.667	0.102
Abe	0	0
Maggie	0	0
Bart	8.333	0.231
Lisa	8.333	0.231
Krusty	14	0.389
Bob	8	0.222
Cecil	0	0

The minimum value we can get is 0, while the maximum is the number of possible links we could have in a network $(n(n-1)/2)$, which equals $(9 \cdot 8)/2 = 36$ in the Simpsons' case. Normally, betweenness centrality is normalized according to the maximum value possible.

In the case of Krusty, its betweenness centrality is $14/36 = 0.389$ *which means that he's between two nodes shortest path nearly* 39% *of the time.*

4.4 Resource Dependence Theory

According to Emerson's power from a dependency framework work, A has power over B to the extent that B is dependent on A:

1. B needs specific resources from A, therefore can exert power over it until B needs the resources;
2. There are no alternatives for B to A: the power disappears by increasing the number of alternatives.

According to Daniel Brass's work about being in the right place, the dependent variable is influence. He looked at closeness and betweenness centrality when focusing on newspaper publishing company employees. In particular:

- closeness centrality is expressed as **access** or minimal distance between a focal actor and all other persons;
- betweenness centrality is expressed as **control** or the relative extent to which a focal actor falls on the shortest path between any two other persons.

Also, there were three levels of analysis: workgroup, department and organization. According to the result:

- when it comes to a workgroup, access to the whole resources is important. When it came to controlling, it is not important;
- control instead is more pivotal at the organizational level, whereas access to resources has no importance since it is strict.

4.5 Bonacich Beta Centrality

It is similar to closeness centrality because we worry about near connections with people that have a lot of connections.

Beta centrality not only refers to geodesic, as closeness does, but also to all possible walks. Flowing of information can be repeated and changed, such as attitude and beliefs. Whenever we reinforce the same links over and over again, we may risk entering inside an echo chamber, so the repetition of connections (back and forward) may affect the centrality of a node.

Closeness is a binary situation of connection, while Bonacich Beta Centrality is more on continuity and consistent connection between nodes. Bonacich based his idea on the fact that actors who have more connections are more likely to be powerful because they c1an directly affect more other actors.

Chapter 5

Matrices and Beta centrality

5.1 Matrix Algebra

5.1.1 Adding two matrices

The matrices X and Y indicate the ties for 2 different network relations (e.g. communication and friendship network) for the same set of 5 actors (a,b,c,d,e), shown in the image below:

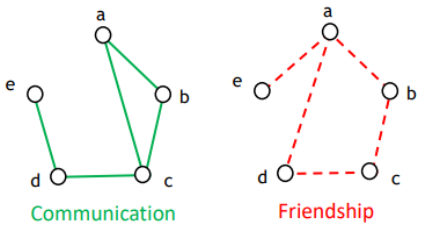


Figure 5.1: Two networks with same nodes

Suppose we want to sum $X + 2Y$ to get Z. Considering their adjacency matrix, we get:

$$X + 2Y = Z$$

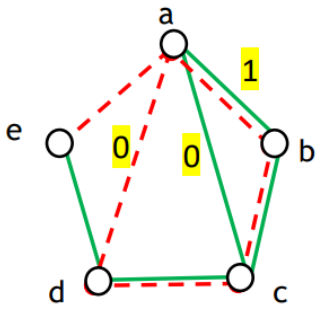
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} + 2 \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 3 & 1 & 2 & 2 \\ 3 & 0 & 3 & 0 & 0 \\ 1 & 3 & 0 & 3 & 0 \\ 2 & 0 & 3 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \end{pmatrix}$$

5.1.2 Cell Multiplication

Suppose we want to compute XY. Considering the adjacency matrices X and Y, we get:

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

This equals to the adjacency matrix of two-steps paths from nodes in X to nodes in Y.



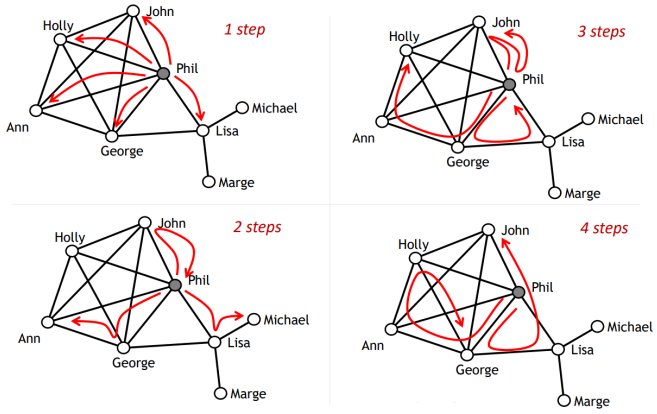
5.1.3 Matrix Multiplication of different matrices

Suppose the two matrices represent the flights and trains from a place to another. Then we can imagine the next scenarios:

- From point A to point A
- From point A to point B
- From point B to point B

5.2 Bonacich Beta Centrality

Imagine there can be a repeated influence, such that it spreads on two or more steps. The image below represents some of the one, two, three and four steps paths inside the network:



$$R = \sum_{k=1}^{\infty} (\beta^{k-1} X^k) = X + \beta X^2 + \beta^2 X^3 + ...$$

The terms in the Bonacich equation are powers of the adjacency matrix. The (i, j) cell of the kth power of an adjacency matrix gives the number of walks of length k from i to j. This formula gives the number of walks of all lengths that join every pair of nodes, weighted by the β values.

Imagine there can be a repeated influence and we want to capture popularity (i.e. being connected to many popular people). By looking at one single step, we get that node d has beta centrality $\beta * (1 + 1 + 1 + 3)$, which represent the number of nodes that neighbours can reach. We then add an additional term β^2, multiplied for the number of nodes each neighbour can reach in two steps.

If a node has a high beta centrality score, it means there are many short walks connecting the node to all others and therefore a strong possibility of influencing others (or being influenced).

Given the matrix X, we have the adjacency matrix of nodes A,B, C and D. We can compute all the two-steps adjacency paths and so on (i.e. $X^2, X^3, X^4, ...$). We could compute the beta centrality of a in the previous net by summing them up:

X	A	B	C	D		
A	0	1	1	0	2	
B	1	0	1	0	2	
C	1	1	0	1	3	
D	0	0	1	0	1	

X^2	A	B	C	D		
A	2	1	1	1	5	
B	1	2	1	1	5	
C	1	1	3	0	5	
D	1	1	0	1	3	

X^3	A	B	C	D		
A	2	3	4	1	10	
B	3	2	4	1	10	
C	4	4	2	3	13	
D	1	1	3	0	5	

X^4	A	B	C	D		
A	7	6	6	4	23	
B	6	7	6	4	23	
C	6	6	11	2	25	
D	4	4	2	3	13	

X^5	A	B	C	D		
A	12	13	17	6	48	
B	13	12	17	6	48	
C	17	17	14	11	59	
D	6	6	11	2	25	

$$c_B(A) = 2 + 5 + 10 + 23 + 48 + ...$$

Instead of just summing them, we could insert the beta β term which reduces the importance of the paths that necessitate of multiple steps: when β is small, only immediate friends matter, while for high β, the global network structure matters. If positive, nodes have higher centrality when they have edges to other central nodes, while if negative they have higher centrality when they have edges to less central nodes.

Note: When selecting negative β, centrality scores themselves can be negative, implying that an actor would be better off not having any connections. It is possible to think of situations in which this could occur but these are rare, and in those cases it is probably better to increase the value of β to eliminate negative scores.

$$c_B(i) = XI + \beta X^2 I + \beta^2 X^3 I + ...$$

By choosing, for example, $\beta = 0.1$, the beta centrality of A becomes:

$$c_B(A) = 1 * 2 + 0.1 * 5 + 0.01 * 10 + 0.001 * 23 + 0.0001 * 48 + ...$$

In the end, we get for each node (only considering a, b, c, d):

We could change the beta value to 0.2 and obtain:

```
library(igraph)
MAT4<-matrix(c(0, 1, 1, 0,
               1, 0, 1, 0,
               1, 1, 0, 1,
               0, 0, 1, 0),4,4)
```

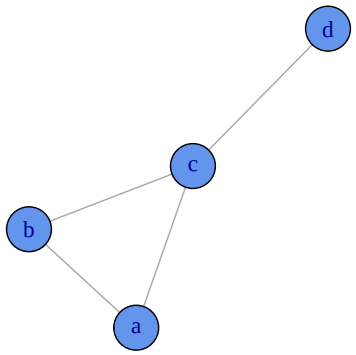
Node	Beta Centrality	Norm
A	0.9747	0.26
B	0.9747	0.26
C	1.3577	0.356
D	0.5065	0.133

Table 5.1: Beta Centrality with value 0.1

Node	Beta Centrality	Norm
A	0.9994	0.260
B	0.9994	0.260
C	1.3117	0.341
D	0.5309	0.138

Table 5.2: Beta centrality with value 0.2

```
MAT4i<-graph_from_adjacency_matrix(MAT4,
                                   mode=c("undirected"),
                                   diag=F)

V(MAT4i)$name = c("a","b","c","d")
# The network we're focusing on
plot(MAT4i,
     vertex.size = 30,
     vertex.color = "cornflowerblue",
     label.color = "white")
```




```
# Beta value = 0
bonpow(MAT4i, exponent=0)/sum(bonpow(MAT4i, exponent=0))
```

```
##     a     b     c     d
## 0.250 0.250 0.375 0.125
```

```
# Beta value = 0.1
bonpow(MAT4i, exponent=0.1)
```

```
##         a         b         c         d
## 0.9746799 0.9746799 1.3577419 0.5064930
```

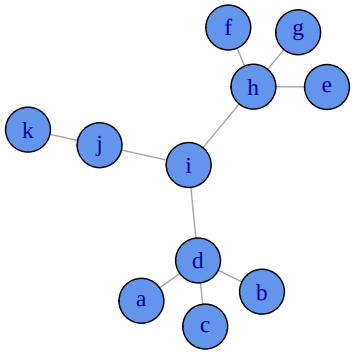
```
# Beta value = 0.1, with normalized values
bonpow(MAT4i, exponent=0.1)/sum(bonpow(MAT4i, exponent=0.1))
```

```
##         a         b         c         d
## 0.2555804 0.2555804 0.3560268 0.1328125
```

```
# Beta value = 0.2
bonpow(MAT4i, exponent=0.2)/sum(bonpow(MAT4i, exponent=0.2))
```

```
##         a         b         c         d
## 0.2601626 0.2601626 0.3414634 0.1382114
```

```
MAT11<-matrix(c(
        0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
        0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0,
        0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0),11,11)

# Creation of the graph
GRAPH11<-graph_from_adjacency_matrix(MAT11,mode=c("undirected"),diag= FALSE)

V(GRAPH11)$name = c("a","b","c","d","e","f","g","h","i","j","k")
# The network we're focusing on
plot(GRAPH11,
     vertex.size = 30,
     vertex.color = "cornflowerblue",
     label.color = "white")
```




```
bonpow(GRAPH11, exponent=.1)/sum(bonpow(GRAPH11, exponent=.1)) # Beta value = 0.1
```

```
##         a         b         c         d         e         f         g
## 0.05570762 0.05570762 0.05570762 0.18241829 0.05570762 0.05570762 0.05570762
##         h         i         j         k
## 0.18241829 0.15842859 0.09547577 0.04701336
```

```
bonpow(GRAPH11, exponent=.2)/sum(bonpow(GRAPH11, exponent=.2)) # Beta value = 0.2
```

```
##         a         b         c         d         e         f         g
## 0.05970982 0.05970982 0.05970982 0.17075893 0.05970982 0.05970982 0.05970982
##         h         i         j         k
## 0.17075893 0.16350446 0.09263393 0.04408482
```

```
bonpow(GRAPH11, exponent=.3)/sum(bonpow(GRAPH11, exponent=.3)) # Beta value = 0.3
```

```
##         a         b         c         d         e         f         g
## 0.06278200 0.06278200 0.06278200 0.16256285 0.06278200 0.06278200 0.06278200
##         h         i         j         k
## 0.16256285 0.16668815 0.09036999 0.04112415
```

```
bonpow(GRAPH11, exponent=.4)/sum(bonpow(GRAPH11, exponent=.4)) # Beta value = 0.4
```

```
##         a         b         c         d         e         f         g
## 0.06536114 0.06536114 0.06536114 0.15666328 0.06536114 0.06536114 0.06536114
##         h         i         j         k
## 0.15666328 0.16861648 0.08799593 0.03789420
```

```
bonpow(GRAPH11, exponent=.5)/sum(bonpow(GRAPH11, exponent=.5)) # Beta value = 0.5
```

```
##         a         b         c         d         e         f         g
## 0.06779661 0.06779661 0.06779661 0.15254237 0.06779661 0.06779661 0.06779661
##         h         i         j         k
## 0.15254237 0.16949153 0.08474576 0.03389831
```

```
bonpow(GRAPH11, exponent=0)/sum(bonpow(GRAPH11, exponent=0)) # Beta value = 0
```

```
##    a    b    c    d    e    f    g    h    i    j    k
## 0.05 0.05 0.05 0.20 0.05 0.05 0.05 0.20 0.15 0.10 0.05
```

```r
bonpow(GRAPH11,exponent=-.1)/sum(bonpow(GRAPH11,exponent=-.1)) # Beta value = -0.1
```

```
##         a         b         c         d         e         f         g         h
## 0.0408331 0.0408331 0.0408331 0.2293779 0.0408331 0.0408331 0.0408331 0.2293779
##         i         j         k
## 0.1345574 0.1087969 0.0528912
```

Chapter 6

Triads and structural holes

Aim: Focus on the individual level, in particular on triads (groups of three people).

6.1 Social Capital

According to Burt, *Social structure is a kind of **capital** that can create for certain individuals or groups a competitive advantage in pursuing their ends. Better connected people enjoy higher returns. In this definition, we can detect:

- **Social capital:** seen as social relations and networks;
- **Capital**: can be financial, human, cultural etc. We can define also who owns it;
- **Competitive Advantage**: it means that who detains it has some benefits; whenever there are no more benefits related to a relationship, we talk about social liabilities. Benefits tend to have a positive effect on the entire network (i.e. tautology).

Individuals benefits from social capital: *"friends, colleagues, and more general contacts through whom you receive opportunities to use your financial capital and human capital"* (Burt, 1992).

Groups benefit from social capital, where shared goals include the sharing of beliefs and norms: *"Social capital flows from the endowment of mutually respecting and trusting relationships which enable a group to pursue its shared goals more effectively than would otherwise be possible"* (Szreter, 2000).

Remember that higher returns do not necessarily involve an active investment. There could be some unintentional results: *"social capital depends on being a byproduct of activities engaged in for other purposes"* (Coleman, 1990). Also, Becker supposes that people choose social networks in ways that will maximize their utility, since returns may be actively created through social relations.

6.2 General Balance Theory

Imagine two people, Stan and Kyle, supposing that the first is into Trump and the second into Biden. Suppose they're friends but they have opposite opinions about politics. What could happen next? Try to convince each other and maintain the friendship or End the relationship;

In case relationships are ended whenever we meet differences, we talk about **social selection**. The issue here is that echo chambers may rise, leading to groups of people that share the same knowledge over and over again.

Instead, whenever we change our minds by interacting with others and dialoguing pacifically, we talk about **social influence or contagion**.

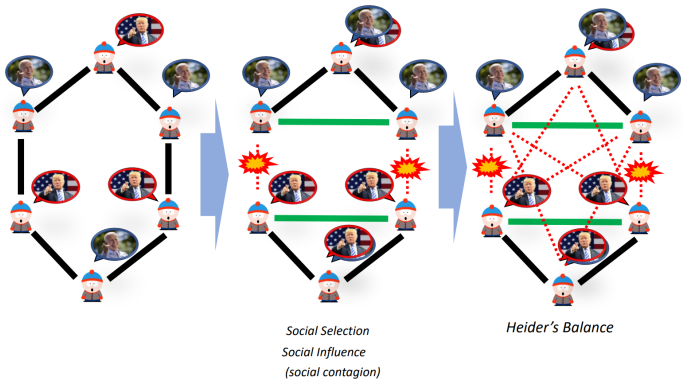
6.3 Heider's Balance Theory

It's about the attitude of people in groups of three. This theory considers triads instead of dyads, as in the previous general theory.

Suppose we're friends with someone that hates one of our friends. It may happen in one of the following cases to bring everything into balance:

- *End the relationship with one of the two;*
- *Make them get along.*

According to this theory is both possible social selection and social influence, such that opposition groups are more strongly differentiated from each other.



6.4 Granovetter's Strength of Weak Ties

> *Weak ties provide people with access to information and resources beyond those available in their own social circle; but strong ties have greater motivation to be of assistance and are typically more easily available. —- Granovetter, 1983*

In this theory, we do not focus on positive/negative ties, but on their strength. Let's think about forbidden triads, defined as those triads where there's a person that is strongly related to two people.

Proposition: If there are 2 strong ties between 3 people, then there will also be at least a weak tie between the others. Therefore, they know each other, despite not being good friends. This theory is based on a probabilistic assumption, therefore it may not happen all the time.

A **forbidden triad** is defined as a combination of two strong ties and a missing tie that links the missing two nodes.

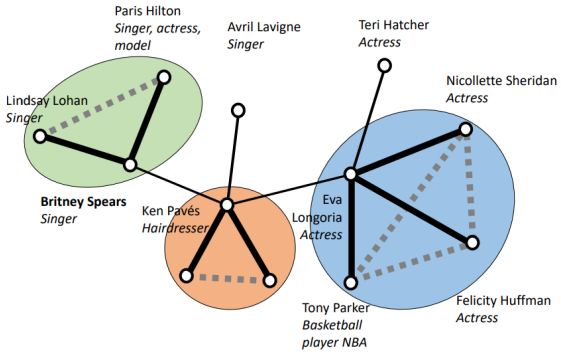
Consider the following example, we can detect three communities, with no structural bridge with strong ties. Weak ties are important because they cross social circles to get unique information. This means that information is spread between social circles via weak ties.

About the importance of acquaintances in social networks:

> *these clumps / [strong ties networks] could not, in fact, being connected to one another at all were it not for the existence of weak ties. —- Granovetter, 1973*

6.5 Simmel and triads

A relation between people in a dyad is of limited interest since it refers to two people. The social dynamics becomes evident when referring to three people: assuming equal individual power of (A, B and C), two actors (A and C) can join forces against a third party (B) and therefore build coalitions to force the third party (Simmel Theory).



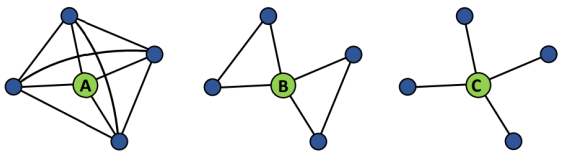
6.6 Coleman and closure

According to **Coleman's closure theory**, high density among people creates group pressure. Closure ensures that group norms are reinforced (joined control against freeriding, anti-social behaviour, etc.). This enforces trust and reduces costs of monitoring and transaction costs (tacit knowledge). In general, closure is a broader normative order within which the individual can optimise performance.

6.7 Burt's Structural Holes

Under a selfish and individualistic perspective, Burt's Structural Holes theory is applied.

By considering the three networks and nodes A, B and C, Coleman's closure theory states that A is in a better position because of the trust enabled by nodes in the network. Whereas Burt's theory prefers C since we maximize the relationships and resources received by nodes that must pass through C to communicate.



Betweenness centrality is relevant because it is about the flow of resources between people we're connected to. In this case, C has the highest betweenness centrality, while A has the lowest. We can summarize previous theories through 3 main arguements that Burt has used:

1. Simmel: No control over ego by others;
2. Betweenness: power overflow of information and resources;
3. Granovetter: access to unique information.

6.7.1 Gould and Fernandez's Brokerage Positions

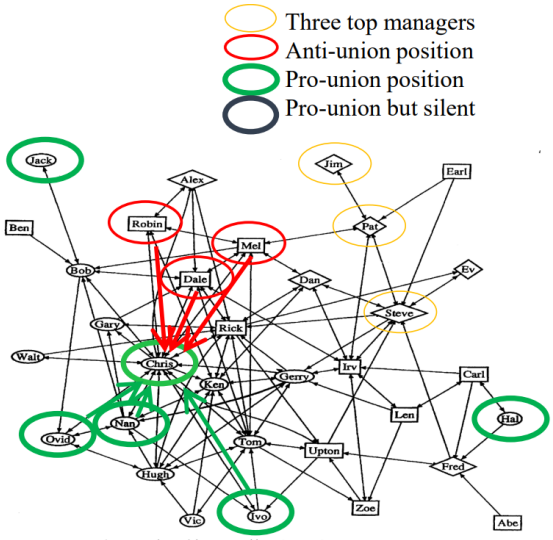
In this example, we look at gatekeepers, by focusing on different inputs and outputs. According to Gould and Fernandez, there are different brokerage positions:

- coordinator: interaction between members of the same group;
- gatekeeper: from an external group, it passes information to its group;
- representative: diffuses knowledge from its group to another;
- cosmopolitan: mediates as outsider members of the same group;
- liaison: interaction between different groups.

6.8 Krackhardt's Simmelian Ties

Krackhardt's disagrees with both Coleman and Burt's theories. According to Krackhardt's, the node in the middle has to satisfy expectations of both groups, which brings node B in the previous networks in a difficult situation. This means that B is in the worst position possible because of the social tension it may encapsulate.

In his example of a Silicon Valley company, he created a friendship network, establishing work position according to the color: manager, pro-union, anti-union and silent positions. In the example, Chris is the most central person and everybody will listen to him. Why? Because he belongs to the pro-union position community, but all the anti-union position workers are his friends. Instead of being at the centre of the attention, Chris became silent because of the closeness to two different groups.



6.9 Small worlds and key players

6.9.1 Six degrees of separation

Milgram's experiment in 1967 led us to the idea of six degrees of separation, according to which between any two people there is a path of length 6 or less.

The value of 6 was a result of an experiment by Stanley Milgram in which he distributed letters randomly to people in two US cities and asked them to pass them to a target, and, if they did not know the target, to pass them to someone who might. He found the average number of steps from originator to target was 6.4. A second example was the analysis of Microsoft Messenger in 2007 which at the time had 180 million nodes and 1.3 billion edges; it was calculated that the average geodesic path length was 5.5.

6.9.2 Small World Effect

According to Watts and Strogatz, in society, there's a high level of clustering, especially when the degrees of separation is that low. Whenever we have a lower geodesic distance within the same cluster, with few nodes, we may want to connect these small communities with some ties: in this case, the average distance reduces.

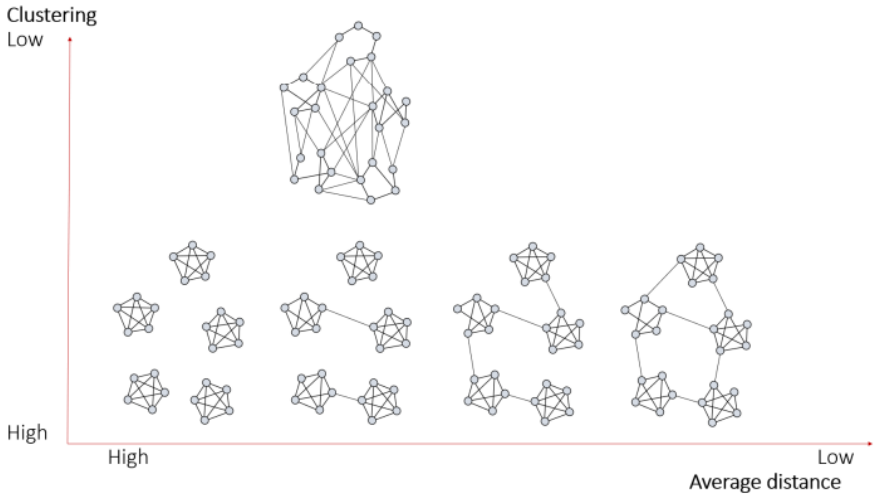
A few changes in where we place ties makes some connected clusters into a small world where anybody can reach anybody else in a few steps.

Human social systems are clumpy but also very compact, in the sense of having surprisingly short paths linking everyone to everyone else. The more transitivity there is in a network the longer

path distances tend to be. It only takes a few connections between clumps to shorten average path length considerably, so the class of networks that are both clumpy and have shorts paths is quite a bit bigger than initially thought.

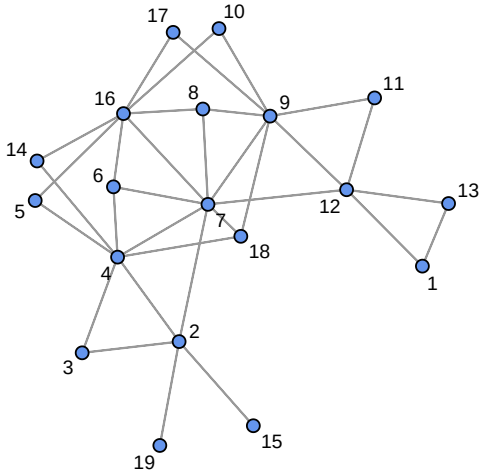
By testing the clustering coefficient, we find out that it is large relative to random graphs (in which the clustering coefficient will be very close to graph density) and the average distance approaches the average distance in random graphs (which is quite small).

By connecting the components with random bridges, by augmenting them, we lower the average distance in the network, while if no components are clearly detectable the clustering lowers.



6.9.3 Keyplayers

Information only reaches the direct contacts of a person and our aim is to reach as many people as possible. Suppose we consider the following network:



We want to reach as many people as possible and we can choose the persons to which we can communicate the information. Who would we choose?

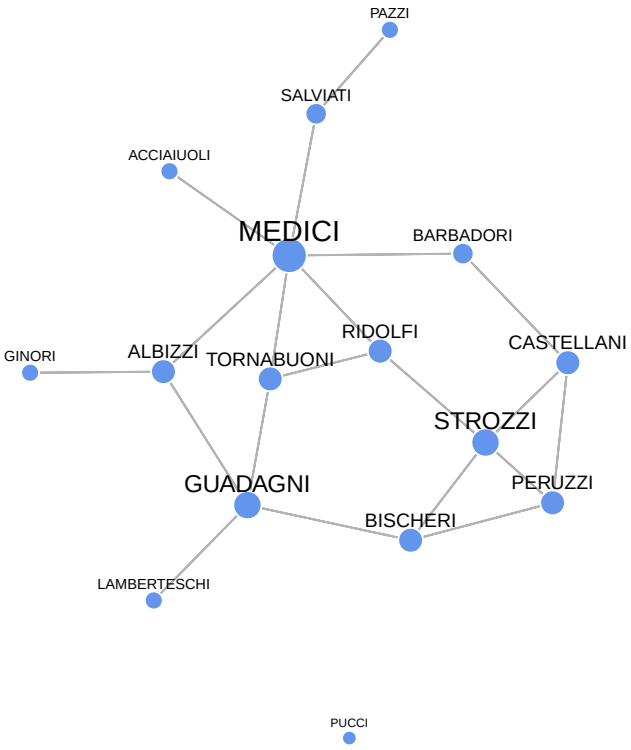
- 1 person: 7;
- 2 persons: $9, 4$;
- 3 persons: $12, 2, 16$, to reach everyone except for 18;
- 4 persons: $12, 4216, 18$.

We're trying to choose non redundant links to reach as many nodes as possible. What if we remove these nodes? Will we still be able to reach them all? Without 9, 7 and 4, we will have three clusters, guided by 12, 2 and 16.

Chapter 7

Metrics for structural holes

7.1 Ego density



Consider the Medici family, which has 6 alters. The number of ties between these alters is 1, while the maximum is $6 \cdot 5/2 = 15$. This means that the ego density is the ratio between the first value and the maximum: $1/15 = 0.067$:

$$\text{ego density} = \frac{\#\text{ ties between alters}}{\#\text{ maximum links between alters}}$$

We could repeat the computation for the remaining families:

For instance, the Peruzzi family has 3 *alters, between which there are* 2 *ties, while the maximum is* $3/(2/2) = 3$. *Therefore the ego density equals* $2/3 = 0.667$.

For instance, the Tornabuoni family has 3 neighbours and between them, there's a link, while the maximum is 3. The ego density equals $1/3 = 0.33$. *Same happens to Bischeri, Castellani, Ridolfi and Strozzi.*

7.2 Ego betweenness

The ego betweenness measures whether a node is within the shortest path between couples of nodes.

Consider the **Medici** Family. It has:

- 5 from Salviati to any other node;
- 5 from Acciaiuoli to any other;
- 5 from Albizzi to any other;
- 5 from Barbadori to any other;
- 4 from Tornabuoni to any other but Ridolfi;
- 4 from Ridolfi to any other but Tornabuoni.

In total, its ego betweenness is 28.

Consider the **Guadagni** family. They have 4 alters:

- from Lamberteschi to any other: 3;
- from Bischeri: 3;
- from Albizzi: 3;
- from Tornabuoni: 3;

In total, its ego betweenness is 12.

Consider the **Peruzzi** Family. They are NOT in the shortest path from Bischeri to Strozzi, neither from Castellani and Strozzi, but since there are two possible paths from Bischeni to Castellani and vice versa, the ego betweenness is 0.5 (1 if we sum up both versus).

Consider the **Strozzi** Family. It has:

- 0.5 from Bischeri to Castellani and vice versa (1)
- 1 from Bischeri/Castellani/Peruzzi to Ridolfi (3) and vice versa (3); In total, its ego betweenness is 7.

By summing the betweenness for every possible path we get the ego-betweenness of the single node.

In Florence's scenario, De Medici is the structural hole, while Pazzi, Pucci, Ginori and Acciaiuoli are the most closed families.

7.3 Constraint index

The constraint index is the opposite of structural holes, which indicate that we're limited by our network. With no constraint, nodes are free to do whatever they want.

$$c_i = \sum_{j=1}^{n} [c_{ij}] = \sum_{j=1}^{n} \left[p_{ij} + \sum_{k=1}^{n} p_{ik} p_{kj} \right]^2 \qquad \text{with} \quad k \neq j \neq i$$

Suppose a is the ego, in four different situations:

- all alters are connected with each other: if a truncates the relationship, there is no affection to theirs;
- same as the previous case, but lower number of alters: higher constraint index;
- no alter is connected to any other alter of a;
- same as the previous case, but lower number of alters, which reduces the constraint index;

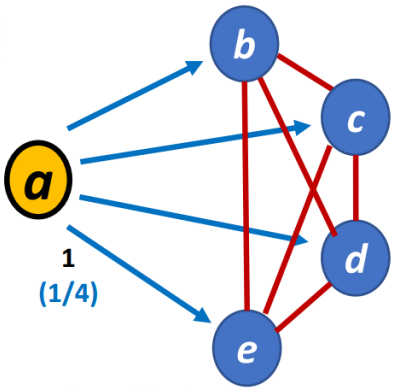
A lower value for the constraint index indicates a higher presence of structural holes. With lower connections, each of them is dangerous, since their loss implies the closure of nodes. If there are fewer friends that are connected with each other there's a higher constraint. Whenever a becomes central and regulates relationships between nodes, there's a structural node and the constraint falls.

7.3.1 Pilot Example

For each direct link starting from a, compute the probability of choosing that link plus the two-step paths that link a to the related node.

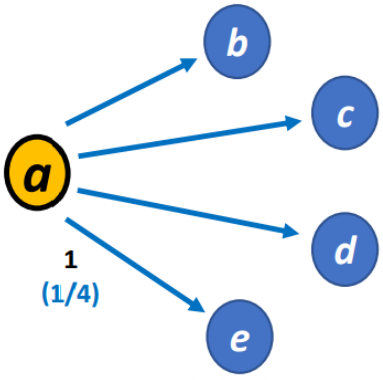
- Example 1: 1/4 probability per each node with within links

$$\begin{aligned} ab &= (p_{ab} + p_{ac}p_{cb} + p_{ad}p_{db} + p_{ae}p_{eb}) \\ &= (1/4 + 1/4 * 1/3 + 1/4 * 1/3 + 1/4 * 1/3)^2 = (1/2)^2 \\ &= ac = ad = ae \\ &\text{Constraint index} = ab + ac + ad + ae = 1 \end{aligned}$$



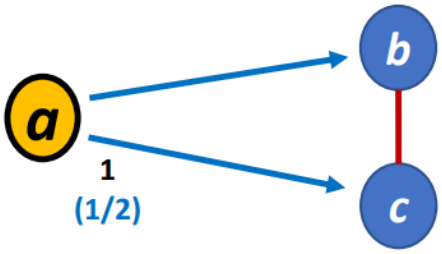
- Example 2: 1/4 probability per each node with no link

$$\begin{aligned} ab &= (p_{ab} + p_{ac}p_{cb} + p_{ad}p_{db} + p_{ae}p_{eb}) \\ &= (1/4 + 1/4 * 0 + 1/4 * 0 + 1/4 * 0)^2 = (1/4)^2 \\ &= ac = ad = ae \\ &\text{Constraint index} = ab + ac + ad + ae = 1/4 \end{aligned}$$



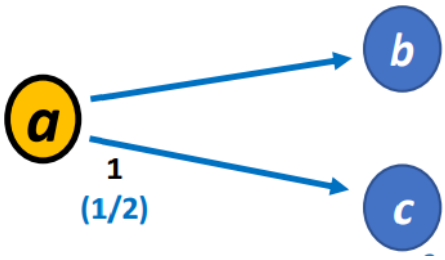
- Example 3: 1/2 probability per each node with link

$$
\begin{aligned}
ab &= (p_{ab} + p_{ac}p_{cb}) \\
&= (1/2 + 1/2 * 1) = 1 \\
&= ac \\
&\text{Constraint index} = ab + ac = 2
\end{aligned}
$$



- Example 3: 1/2 probability per each node with no link

$$
\begin{aligned}
ab &= (p_{ab} + p_{ac}p_{cb}) \\
&= (1/2 + 1/2 * 0) = 1/2 \\
&= ac \\
&\text{Constraint index} = ab + ac = 1
\end{aligned}
$$



Note: While igraph considers the probability $p_{ij} of i to reach j inside the entire network, the UCINET limits the constraint index probabilities to the ego's alters. For instance, suppose the ego is Carlo, which has Marco, Maria and Mirco as friends, all connected. In the UCINET network, $p_{Maria,Marco}$ *will be* 1/3*, while in igraph, supposing Maria has a friend named Clara, it would be* 1/4*.*

7.3.2 Florencian Families Example

```
# How to compute the constraint index
igraph::constraint(graph, nodes = V(graph), weights = NULL)
```

```
library(igraph)
PFM<-read.csv("datasets/Padgett_FlorentineFamilies_Marriage.csv",
                            stringsAsFactors=FALSE,
                            row.names=1)
PFM_i<-graph_from_adjacency_matrix(as.matrix(PFM),
                                   mode="undirected",
                                   diag=F)
constraint(PFM_i)
```

```
##   ACCIAIUOLI      ALBIZZI     BARBADORI      BISCHERI    CASTELLANI       GINORI
```

```
##    1.0000000    0.3333333    0.5000000    0.4822531    0.4822531    1.0000000
##     GUADAGNI LAMBERTESCHI       MEDICI        PAZZI      PERUZZI        PUCCI
##    0.2500000    1.0000000    0.2098765    1.0000000    0.6558642          NaN
##      RIDOLFI     SALVIATI      STROZZI   TORNABUONI
##    0.4598765    0.5000000    0.4583333    0.4598765
```

To know whether a spends the same amount of time between all of its neighbours, the strength of the relationship may be necessary. Otherwise, we will assume that its attention will be equally split among all alters.

Let's try to compute the index with some families:

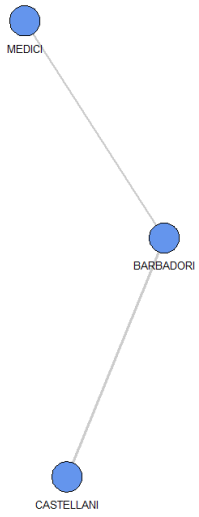
- Pazzi

  Pazzi only has 1 alter: Salviati. Therefore the constraint index will be equal to the term that counts the probability of going from Pazzi to Salviati (which equals $1^2 = 1$).

- Barbadori

  Barbadori only has two alters: Medici and Castellani:

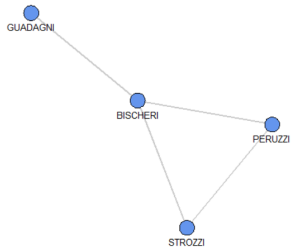
$$BM = (p_{BM} + p_{BC} * p_{CM})^2 = (1/2 + 1/2 * 0)^2 = 1/4$$
$$BC = (p_{BC} + p_{BM} * p_{MC})^2 = (1/2 + 1/2 * 0)^2 = 1/4$$
$$\text{Constraint index} = BM + BC = 1/2 = 0.5$$



- Bischeri

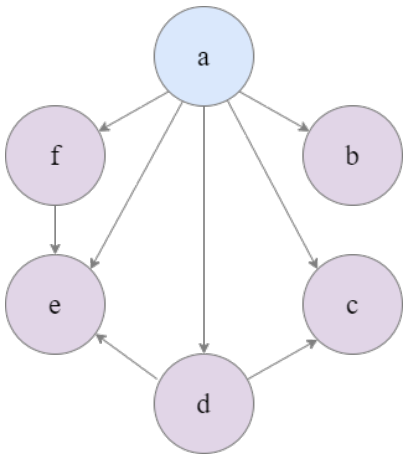
  Bischeri has three neighbours: Guadagni, Strozzi and Peruzzi:

$$\begin{aligned} BG &= (p_{BG} + p_{BP} * p_{PG} + p_{BS} * p_{SG})^2 \\ &= (1/3 + 1/3 * 0 + 1/3 * 0)^2 = 0.1111 \\ BS &= (p_{BS} + p_{BP} * p_{PS} + p_{BG} * p_{GS})^2 \\ &= (1/3 + 1/3 * 1/3 + 1/3 * 0)^2 = 0.1975 \\ BP &= (p_{BP} + p_{BG} * p_{GP} + p_{BS} * p_{SP})^2 \\ &= (1/3 + 1/3 * 0 + 1/3 * 1/4)^2 = 0.1736 \\ \text{Constraint index} &= BG + BS + BP = 0.4823 \end{aligned}$$



Note that this method of computing the constraint index is graph one, while UCINET automatically removes the starting edge that links the intermediary node to the starting ego node, leading to a greater or equal value for CI.

7.4 Exercise



Given node a as ego, compute the constraint index:

- ab

$$ab = (p_{ab} + p_{ac} * p_{cb})^2 = (2/11)^2 = (14/77)^2$$

- ac

$$\begin{aligned} ac &= p_{ab} * p_{bc} + p_{ac} + p_{ad} * p_{dc} + p_{ae} * p_{ec} + p_{af} * p_{fc} = \\ &= (2/11 * 0 + 5 + 1/11 * 5/7 + 1/11 * 0 + 2/11 * 0)^2 = \\ &= (5/11 + 5/77)^2 = (40/77)^2 \end{aligned}$$

- ad

$$\begin{aligned} ad &= p_{ab} * p_{bd} + p_{ac} * p_{cd} + p_{ad} + p_{ae} * p_{ed} + p_{af} * p_{fd} = \\ &= (2/11 * 0 + 5/11 * 0 + 1/11 + 1/11 * 0 + 2/11 * 0)^2 = (1/11)^2 = (7/77)^2 \end{aligned}$$

- ae

$$\begin{aligned} ad &= p_{ab} * p_{be} + p_{ac} * p_{ce} + p_{ad} * p_{de} + p_{ae} + p_{af} * p_{fe} = \\ &= (2/11 * 0 + 5/11 * 0 + 1/11 * 2/7 + 1/11 + 2/11 * 2/2)^2 = \\ &= (2/77 + 1/11 + 2/11)^2 = (23/77)^2 \end{aligned}$$

- af

$$\begin{aligned} af &= p_{ab} * p_{bf} + p_{ac} * p_{cf} + p_{ad} * p_{df} + p_{ae} * p_{ef} + p_{af} = \\ &= (2 * 0 + 5 * 0 + 1 * 0 + 1 * 0 + 2/11)^2 = (2/11)^2 = (14/77)^2 \end{aligned}$$

Seems like af and ab have the same constraint index, which is higher for ac and ae and lower for ad.

Chapter 8

Statistical Tests on Nodal Level

8.1 Classic Approach

In the classic approach, we would:

- correlate the attribute columns;
- significance tests, whose assumptions are violated by network data (e.g. independent observations, data distribution).

8.1.1 Correlation

The correlation coefficient measures the association between two interval or ratio variables. Consider two variables x_i and y_i. How are two continuous variables related? The correlation could be positive if increases, null if there is none and negative if decreases:

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}$$

where $\bar{x}, \bar{y}$ are the means and s_x, s_y are the standard deviations.

8.1.2 Significance test

It is based on the idea that a feature or variable provides some useful information about data.

1. **Sample**: make a statement about the relation between two variables for a well defined and finite population. Since it is difficult to collect information for the whole population, **randomly sampling** is a solution.
2. **Hypothesis**: Before looking at data, make a statement about the population H_0 that we will reject in favour of an alternative statement about the population H_A. What we aspire to do is to prove that H_0 is very unlikely.
3. **Probabilities**: Look at sample results. Assuming that the null hypothesis H_0 is true, how much chance do we have of obtaining a sample that has a correlation that is that extreme or more extreme (than we observed with the sample).
4. **Conclusion**: reject the null hypothesis H_0 in favour of the alternative hypothesis H_a if the probability of such an extreme or more extreme outcome is lower than 5% ($p < 0.05$).

8.2 Permutation Based Approach

This approach does not involve samples.

Imagine we have data about five people, knowing their age. We want to explore eventual correlations between out degree and age.

```
# Creating data and computing the correlation
OD = c(3,1,2,2,3) # Out degree
age = c(19,45,32,25,21) # Ages
cor.test(OD,age) # Correlation test
```

```
##
##  Pearson's product-moment correlation
##
## data:  OD and age
## t = -5.166, df = 3, p-value = 0.01407
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9966747 -0.4028377
## sample estimates:
##        cor
## -0.9481293
```

According to this test, with a p-value below 0.05, there is a negative correlation that explains that, by increasing the age, the number of out links decreases.

What happens if we permute nodes position? If we perform some permutations, what proportion of all these samples that could have come out would result in a correlation as large as the one we actually observed? What are the chances of observing such a large correlation even when the values of the variables are assigned independently of each other?

```
# Perform correlation test on permuted age
OD = c(3,1,2,2,3) # Out degree still the same
age = c(45,32,25,21,19) # Ages permuted
cor.test(OD,age)
```

```
##
##  Pearson's product-moment correlation
##
## data:  OD and age
## t = 0.17794, df = 3, p-value = 0.8701
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8573755  0.9030401
## sample estimates:
##       cor
## 0.1021936
```

If we permute ages, then the p-value rises, to reject our alternative hypothesis and accept the null one. A **permutation test** (also called re-randomization test) is an exact test, a type of statistical significance test in which the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under all possible rearrangements of the observed data points.

Reformulating, the permutation test calculates all the ways that the experiment could have come out given that scores were actually independent and counts the proportion of random assignments yielding a correlation as large as the one actually observed.

We can then compare the observed correlation against the distribution of correlations that one could obtain if the two variables were in fact independent of each other.

It's interesting to notice that by permuting ages, we will in the end get a normal distribution for our correlation.

Chapter 9

Two mode networks

9.1 Two-mode data

Whenever we focus on **two-mode networks**, we put effort on ties, not on actors, where there are two types of actors. As with 1-mode networks, we need to specify the boundary of the network:

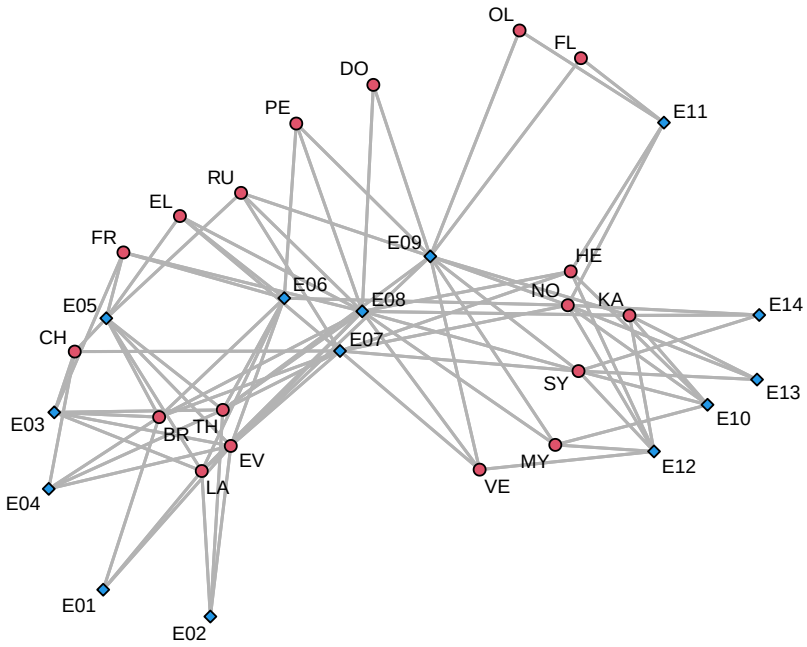
- defined by events to which actors participate, missing other events to which those actors attended;
- by actors, to include some events, missing the other actors that attended those events.

For instance, suppose we consider social movements consisting into actors and meetings or events or organizations to which they are affiliated.

In analyzing two-mode data, we typically make the assumption that attending the same event is either an indicator of an underlying social relationship between the actors or a potential opportunity for one to develop. It could also be a fortuity to notice the co-attendance of large events.

First, import data about the attendance of women to some events.

```
library(sna)
DSAm <- as.matrix(read.csv(
  "datasets/Davis_SouthernWomen_Attendance.csv",
  stringsAsFactors = FALSE, row.names=1))

DSAmS<-DSAm
#Change names in new object
rownames(DSAmS)<-substring(rownames(DSAm),1,2)

#Now draw the network
par(mar = c(0,0,0,0))
gplot(DSAmS,
      displaylabels=TRUE,
      usearrows=FALSE,
      edge.col = "grey70",
      gmode="twomode",
      vertex.cex = 0.5,
      label.cex = 0.7)
```



9.2 One-mode projections

A one-mode projection is another way of representing two-mode projections by inserting links between nodes if they share a common link toward a node. In order to compute it:

1. Get the adjacency matrix A;
2. Compute the transpose of this matrix $t(A)$;
3. Multiply the first matrix with the second one to compute two steps paths from a node to another $A \cdot t(A)$.

```
# One-mode projection

# Persons
DSAmS%*%t(DSAmS)
# Events
t(DSAmS)%*%DSAmS
```

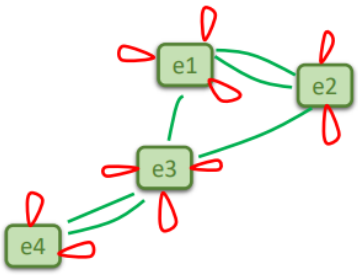
What is the best approach to represent the level of overlapping?

- Sum of unique events shared by all actors over the total number of events;
- Unique events shared over the sum of degrees of all actors;
- Unique events shared over the minimum degree of all actors;
- Unique events shared over the maximum degree of all actors.

With One-mode projections, we lose data. Check the example below.

9.3 Bipartite

One of the problems with converting the affiliation matrix to one-mode is that there can be a loss of information. It is possible to compute a two-mode network centrality using **bipartite one-mode**: according to this modality, we put together actors and events to form total nodes and fill



Transposed of M= M'

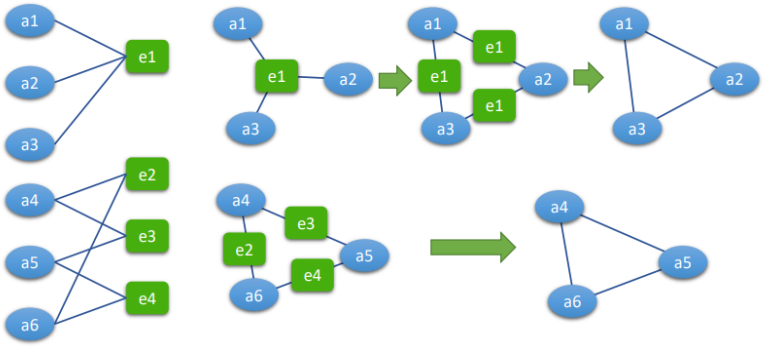
	a1	a2	a3	a4	a5
e1	1	1	1	0	0
e2	0	1	1	0	0
e3	0	0	1	1	0
e4	0	0	0	1	1

M

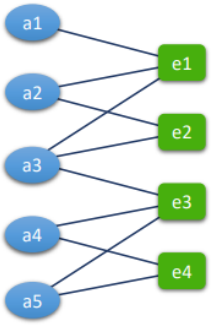
	e1	e2	e3	e4
a1	1	0	0	0
a2	1	1	0	0
a3	1	1	1	0
a4	0	0	1	1
a5	0	0	1	1

M' * M =

	e1	e2	e3	e4
e1	*3*	2	1	0
e2	2	*2*	1	0
e3	1	1	*3*	2
e4	0	0	2	*2*



the same-type nodes with 0s.



	e1	e2	e3	e4	
a1	1	0	0	0	1
a2	1	1	0	0	2
a3	1	1	1	0	3
a4	0	0	1	1	2
a5	0	0	1	1	2
	3	2	3	2	10

M* =

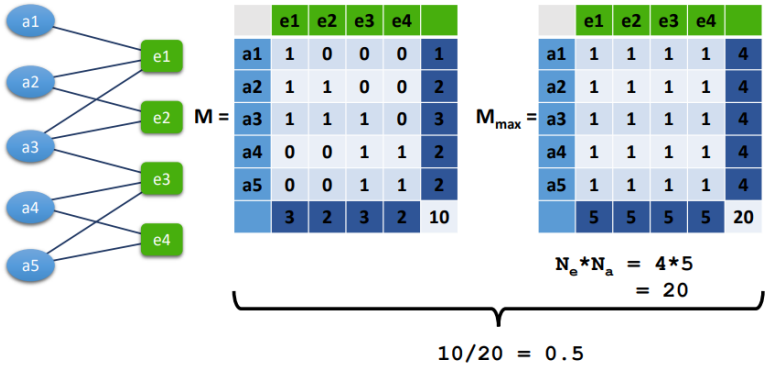
	a1	a2	a3	a4	a5	e1	e2	e3	e4	
a1	*0*	0	0	0	0	1	0	0	0	1
a2	0	*0*	0	0	0	1	1	0	0	2
a3	0	0	*0*	0	0	1	1	1	0	3
a4	0	0	0	*0*	0	0	0	1	1	2
a5	0	0	0	0	*0*	0	0	1	1	2
e1	1	1	1	0	0	*0*	0	0	0	3
e2	0	1	1	0	0	0	*0*	0	0	2
e3	0	0	1	1	1	0	0	*0*	0	3
e4	0	0	0	1	1	0	0	0	*0*	2
	1	2	3	2	2	3	2	3	2	*20*

9.4 Density

To compute the density of a two-mode network, compute the total number of edges and divide it for the maximum number of edges there could be.

```
# Getting the degree of a two-mode network
mean(DSAmS)
```

```
## [1] 0.3531746
```



9.5 Degree Centrality

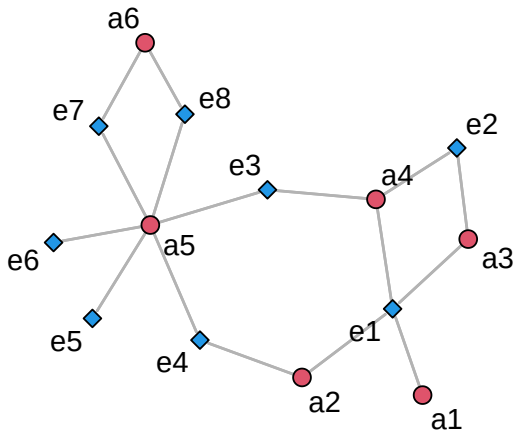


Figure 9.1: Scientists and relative written papers network

Consider a network with scientists and papers, where n_a defines the number of scientists and n_e defines the number of papers. It is possible to compute:

- *How many publications (E) does each person (A) have?*

$$A_1 = 1$$
$$A_2 = A_3 = A_6 = 2$$
$$A_4 = 3$$
$$A_5 = 6$$

```
rowSums(scientists_papers) # Scientists Degree
```

```
## a1 a2 a3 a4 a5 a6
##  1  2  2  3  6  2
```

```
colSums(scientists_papers) # Papers Degree
```

```
## e1 e2 e3 e4 e5 e6 e7 e8
##  4  2  2  2  1  1  2  2
```

- *In what proportion of all publications (E) was a person (A) involved?*

  Divide the previous number for the total number of publications n_e to obtain the proportion of collaboration in all papers.

```
# Proportion of scientists participation in papers
rowSums(scientists_papers)/NCOL(scientists_papers)
```

```
##    a1    a2    a3    a4    a5    a6
## 0.125 0.250 0.250 0.375 0.750 0.250
```

```
# Proportion of papers collaboration in scientific community
round(colSums(scientists_papers)/NROW(scientists_papers),3)
```

```
##    e1    e2    e3    e4    e5    e6    e7    e8
## 0.667 0.333 0.333 0.333 0.167 0.167 0.333 0.333
```

- *What is A's total contribution to all publications (E) taking into account the contribution of co-authors?*

  When computing the proportion, we consider the number of colleagues that collaborate on a specific paper. If a scientist works on more papers at a time (suppose A_2), we just sum up that proportion. In a short way, compute the proportion of people that worked in a paper and then compute row sums of those proportions.

```
# Get column sum = 1 and row sum equal
# the total contribution considering coauthors
t(t(scientists_papers)/colSums(scientists_papers))
```

```
##      e1  e2  e3  e4 e5 e6  e7  e8
## a1 0.25 0.0 0.0 0.0  0  0 0.0 0.0
## a2 0.25 0.0 0.0 0.5  0  0 0.0 0.0
## a3 0.25 0.5 0.0 0.0  0  0 0.0 0.0
## a4 0.25 0.5 0.5 0.0  0  0 0.0 0.0
## a5 0.00 0.0 0.5 0.5  1  1 0.5 0.5
## a6 0.00 0.0 0.0 0.0  0  0 0.5 0.5
```

```
# Total contribution to all papers by author
rowSums(t(t(scientists_papers)/colSums(scientists_papers)))
```

```
##   a1   a2   a3   a4   a5   a6
## 0.25 0.75 0.75 1.25 4.00 1.00
```

- *What is A's average contribution to all publications (E) taking into account the contribution of co-authors?*

Take the previous value and divide it for the total number of papers in which they collaborated (first column).

```
round((
  rowSums(
    t(
      t(
        scientists_papers)/colSums(scientists_papers))))/
    rowSums(scientists_papers),3)
```

```
##    a1    a2    a3    a4    a5    a6
## 0.250 0.375 0.375 0.417 0.667 0.500
```

Ego	Degree			
	Number of papers	Proportion of all papers	Total contribution	Average contribution
A1	1	0.13	.25	0.25/1
A2	2	0.25	.25+.5	0.75/2
A3	2	0.25	.25+.5	0.75/2
A4	3	0.38	.25+.5+.5	1.25/3
A5	**6**	**0.75**	**1+1+.5+.5+.5+.5**	**4.00/6**
A6	2	0.25	.5+.5	1.00/2

Figure 9.2: Contribution of each author

- *In total how many unique co-authors does A have?*
  1. Number of co-authors
     Just count the neighbours of the authors, independently from the event E, so just consider the one-way projection of the network. We will get the numbers below, noticing that A_2 and A_4 are the most central nodes in the network.
     A_1 3
     A_2 4
     A_3 3
     A_4 4
     A_5 3
     A_6 1
  2. Proportion of all authors
     Divide the number of co-authors by the maximum number obtainable ($N_a - 1 = 6 - 1 = 5$).
     A N P
     A_1 3 0.60
     A_2 4 0.80
     A_3 3 0.60
     A_4 4 0.80
     A_5 3 0.60
     A_6 1 0.20
  3. Number of multiple co-authors
     In total how many co-authors does A have on all papers (irrespective of whether they are the same or not, so counting multiple times same people if authors had multiple collaborations)?
     A N P NM
     A_1 3 0.60 3
     A_2 4 0.80 4

A_3	3	0.60	4
A_4	4	0.80	5
A_5	3	0.60	4
A_6	1	0.20	2

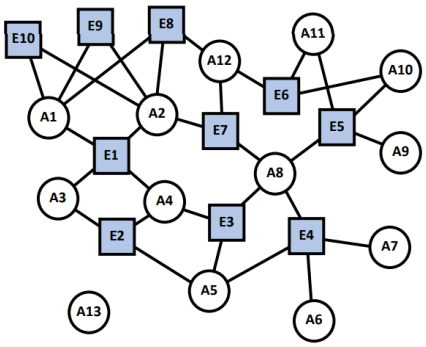
4. The average number of co-authors on a paper
   Divide the number of co-authors (point 1) by the number of papers per author (first column in the original table).

A	N	P	Nm	NP
A_1	3	0.60	3	3
A_2	4	0.80	4	2
A_3	3	0.60	4	2
A_4	4	0.80	5	1.33
A_5	3	0.60	4	0.67
A_6	1	0.20	2	1

   The last column may indicate that A_1, A_2, A_3 collaborate more, while A_5 tends to produce its own ideas.

9.6 Exercise

9.6.1 Exploration



Density: $31/(n_a \cdot n_e) = 31/130 = 0.24$

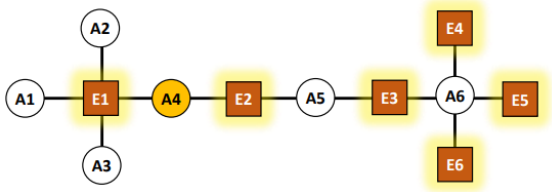
Which person has attended most parties? A_2

Which person has had the possibility to meet the most different people through parties? A_8

Which event is attended by the most people? E_1, E_4, E_5

9.7 Closeness centrality

The closeness centrality is computed as the sum of closenesses from actors to other actors and from actors to events, considering all edges in between.



The minimum distance in the first case (other) is n_e, while in the second case (own) is $2n_a - 2 = 2(n_a - 1)$. In total, by summing them, the minimum distance is $n_e + 2(n_a - 1)$.

	Closeness (distance)		
	other	own	all
A1	1+3+5+7+7+7 =30	2+2+2+4+6 =16	46
A2	1+3+5+7+7+7 =30	2+2+2+4+6 =16	46
A3	1+3+5+7+7+7 =30	2+2+2+4+6 =16	46
A4	1+1+3+5+5+5 =20	2+2+2+2+4 =12	32
A5	3+1+1+3+3+3 =14	4+4+4+2+2 =16	**30**
A6	**5+3+1+1+1+1 =12**	6+6+6+4+2 =24	36

9.8 Betweenness Centrality

We can focus on:

- how many times an actor is between two events;
- how many times an event is between two actors;
- how many times an actor is between two events/actors, mixing them;
- their sum, which is the total betweenness centrality.

It can be compared with the maximum of each term:

- $n_e(n_e-1)/2$
- $(n_a-1)(n_a-2)/2$
- $(n_a-1)(n_e-1)$
- $n_e(n_e-1)/2+(n_a-1)(n_a-2)/2+(n_a-1)(n_e-1)$

In a bipartite graph, the only way that a node can achieve the theoretical maximum is if it is the only member of its vertex set.

9.9 Structural Holes

It is about being between other people in a structural way such that without that bridge there would be a hole.

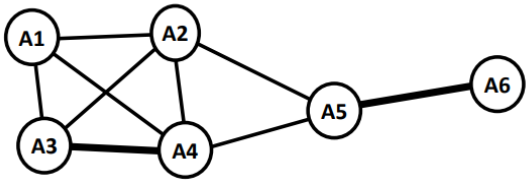
In a two-mode network, with a closed triadic structure, how should structural holes be defined?

Structural holes can be defined through **closure**, which is computed by the ratio of a number of co-authors with multiple counts over unique co-authors.

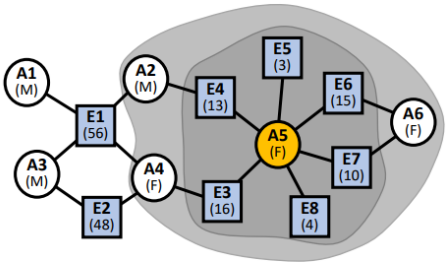
	Number of papers	Proportion of all papers	Number of co-authors	Number of co-authors (multiple count)	Closure
A1	1	0.13	3	3	3/3=1
A2	2	0.25	4	4	4/4=1
A3	2	0.25	3	4	1.33
A4	3	0.38	4	5	1.25
A5	6	**0.75**	3	4	1.33
A6	2	0.25	1	2	2/1=2

If we consider the problem of measuring closure/openness in a one-mode projection, we should consider the proportion between the actual edges from an alter to another over the maximum number of edges between a node's alters.

For instance, in this example, we would have the projection above, highlighting: $A_5 \to 1/3, A_2 = A_3, A_3 = A_4 = A_1 \to 1$.



9.10 Attribute-based measures



Ego	Alters (distance 1)	
	Total citations of papers	Average citations of papers
A1	56	56.0
A2	69	34.5
A3	104	52.0
A4	120	40.0
A5	61	12.2
A6	25	12.5

Suppose we work with a network of academics that published papers. Some attribute measures could be the total citations of papers and the average citations of papers, starting from the number of times a paper is cited and author gender.

What proportion of authors if female?

Seems like A_6 is connected only with females (i.e. A_5), while A_4 and A_1 are the authors that are mostly connected to men.

Ego	"Extended" alters (distance 2)		
	Proportion alters' female	Weighted proportion alters' female	Average proportion alters' female
A1	.33	.33	.33
A2	.50	.50	(1.00+.33)/2 =.66
A3	.33	.50	(1.00+.33)/2 =.66
A4	.25	.20	(1+0+0)/3=.33
A5	.67	.75	(1+1+1+0)/4=.75
A6	1.00	1.00	(1+1)/2=1

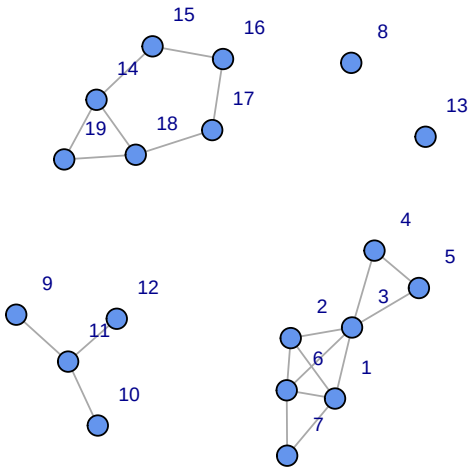
Chapter 10

Subgroups and Structural Equivalence

10.1 Subgroups

Embedded within a network there are often groups of actors who interact with each other to such an extent that they could be considered to be a separate entity. This group can be called "cohesive subgroup" or community. They may allow to discover patterns in people or why certain behaviour/information is spread around a group of nodes. Also, if there are shared norms and similar actions for group members, we may be able to replace cohesive group members by a single 'super-node'. This would reduce the complexity and size of the network and consequently aid visualization and analysis.

Let's start with the following example:



10.1.1 Components

Who would be able to "eventually" share information with each other?

A subset where everyone is directly or indirectly connected to everyone else is said **component**.

Note that if actors a and b are at least indirectly connected, and actor c is at least indirectly connected to b, then actor c is at least indirectly connected to a. Hence all actors belong to one and only one component.

10.1.2 Cliques

```
plot(network_i,
     vertex.label.dist=3,
     vertex.label.cex = 0.7,
     edge.arrow.size = 0.5,
     vertex.color = "cornflowerblue",
     displaylabels=T,
     vertex.size = 10,
     vertex.label.family = "Helvetica",
     mark.groups = list(c(14,18,19),c(1,2,3,6),c(1,7,6),c(3,4,5)),
     mark.col = NA,
     mark.border=c("red","blue", "orange","green"),
     margin = c(0,0,0,0),
     arrow.mode = 0)
```

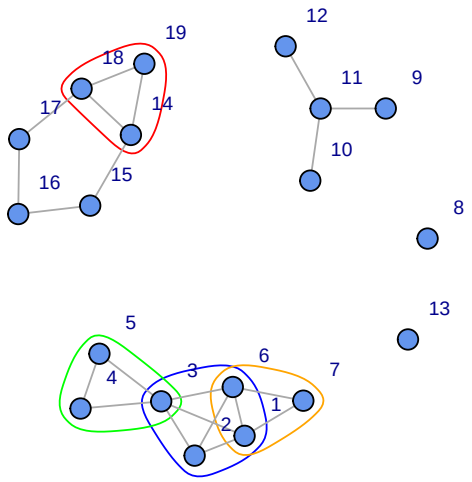


Figure 10.1: cliques

A subset where everyone is directly connected to everyone else in the subset is called a **clique**.

Note that both an actor and a dyad might belong to more than one clique. In order to limit the

overlapping of cliques, it is possible to increase the minimum size of a clique.

10.1.3 K-cliques

A subset where everyone is connected to everyone else in the subset by at most K steps is called K-cliques.

By considering the previous network, the 2-cliques are:

- $14, 15, 17, 18, 19$;
- $14, 15, 16, 17, 18$;
- $9, 10, 11, 12$;
- $1, 2, 3, 6, 7$;
- $1, 2, 3, 6, 4, 5$;

```
plot(network_i,
     vertex.label.dist=3,
     vertex.label.cex = 0.7,
     edge.arrow.size = 0.5,
     vertex.color = "cornflowerblue",
     displaylabels=T,
     vertex.size = 10,
     vertex.label.family = "Helvetica",
     mark.groups = list(c(14, 15, 17, 18, 19),c(14, 15, 16, 17, 18),
                        c(9, 10, 11, 12), c(1, 2, 3, 6, 7), c(1, 2, 3, 6, 4, 5)),
     mark.col = NA,
     mark.border=c("red","orange","green","blue","violet"),
     margin = c(0,0,0,0),
     arrow.mode = 0)
```

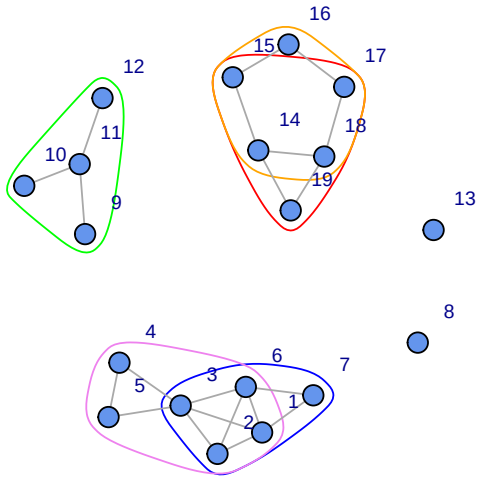


Figure 10.2: 2 cliques

Note that actors might become cliques through others that are not members themselves. For instance, if we consider the 2-clique composed by $K_2 = \{14, 15, 17, 18, 19\}$ thanks to 16.

By considering instead 3-cliques, we obtain:

- $14, 15, 16, 17, 18, 19$;
- $1, 2, 3, 4, 5, 6, 7$;
- $9, 10, 11, 12$.

If $K = N - 1$, then we get a component, while if $K = 1$, we get a normal clique.

```
# Compute the cliques
# Returns number of cliques per each node
clique.census(network, mode = "graph")
```

```
## $clique.count
##   Agg 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
## 1   2 0 0 0 0 0 0 0 1 0  0  0  0  1  0  0  0  0  0  0
## 2   7 0 0 0 0 0 0 0 0 1  1  3  1  0  1  2  2  2  1  0
## 3   3 1 0 1 1 1 1 1 0 0  0  0  0  0  1  0  0  0  1  1
## 4   1 1 1 1 0 0 1 0 0 0  0  0  0  0  0  0  0  0  0  0
##
## $cliques
## $cliques[[1]]
## $cliques[[1]][[1]]
## [1] 13
##
## $cliques[[1]][[2]]
## [1] 8
##
##
## $cliques[[2]]
## $cliques[[2]][[1]]
## [1] 17 18
##
## $cliques[[2]][[2]]
## [1] 16 17
##
## $cliques[[2]][[3]]
## [1] 15 16
##
## $cliques[[2]][[4]]
## [1] 14 15
##
## $cliques[[2]][[5]]
## [1] 11 12
##
## $cliques[[2]][[6]]
## [1] 10 11
##
## $cliques[[2]][[7]]
## [1]  9 11
##
##
## $cliques[[3]]
## $cliques[[3]][[1]]
## [1] 14 18 19
##
```

```
## $cliques[[3]][[2]]
## [1] 1 6 7
##
## $cliques[[3]][[3]]
## [1] 3 4 5
##
##
## $cliques[[4]]
## $cliques[[4]][[1]]
## [1] 1 2 3 6
```

10.1.4 K-clans

A K-clique where everyone is connected to everyone else in the subset by at most K steps through its own members is called K-clan. This means that actors become cliques only through cliques members.

Some 2-clans in the network are:

- 14, 15, 18, 19;
- 14, 17, 18, 19;
- 9, 10, 11, 12;
- 1, 2, 3, 6, 7;
- 1, 2, 3, 6, 4, 5.

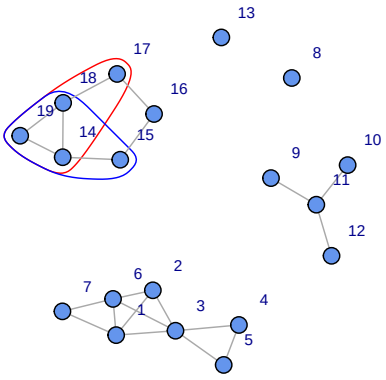


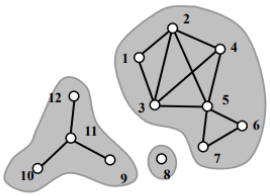
Figure 10.3: Example of 4-clan

10.2 Equivalence

10.2.1 Core periphery

If we try to represent the network with an adjacency matrix, we would notice that cliques are evident by 1s zones. In order to get a completely filled by 1s zone, there should be a direct path between all nodes. Based on the size of these blocks, we can distinguish the core from the periphery. In fact, the periphery will have zeros zones, if not connected to anyone, or 1 and 0s zones, while the core will have 1s zones.

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	1	1	0	0	0	0	0	0	0	0	0
2	1	-	1	1	1	0	0	0	0	0	0	0
3	1	1	-	1	1	0	0	0	0	0	0	0
4	0	1	1	-	1	0	0	0	0	0	0	0
5	0	1	1	1	-	1	1	0	0	0	0	0
6	0	0	0	0	1	-	1	0	0	0	0	0
7	0	0	0	0	1	1	-	0	0	0	0	0
8	0	0	0	0	0	0	0	-	0	0	0	0
9	0	0	0	0	0	0	0	0	-	0	1	0
10	0	0	0	0	0	0	0	0	0	-	1	0
11	0	0	0	0	0	0	0	0	1	1	-	1
12	0	0	0	0	0	0	0	0	0	0	1	-



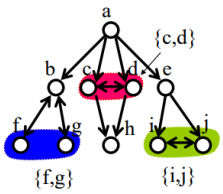
10.2.2 Structural equivalence

In social network analysis, two nodes are considered **structurally equivalent** if they have the same neighborhoods – they are connected to the same others. Structural equivalence is related to the roles we play in the network and to the fact that people that play the same role in the network act similarly.

Structural equivalence examines the direct connections of an actor to other actors in the network. Two actors are structurally equivalent if they send ties to the same third parties, and receive ties from the same third parties. They do not need to have a direct tie to each other to be equivalent.

It is a useful concept to identify similarities in attitudes and behaviours of nodes in the network and they tend to show homogeneity, as actors in the same cohesive subgroup. This similarity may be due to environment influence, due to their role or their shared neighbours. Remember that persons adapt to their social environments, and therefore actors with similar social environments will tend to have certain similarities.

For instance, in the network below, we know that there are some nodes that have the same behaviours: $\{f, g\}, \{c, d\}, \{i, j\}$.



	a	b	c	d	e	f	g	h	i	j
a	-	1	1	1	1	0	0	0	0	0
b	0	-	0	0	0	1	1	0	0	0
c	0	0	-	1	0	0	0	1	0	0
d	0	0	1	-	0	0	0	1	0	0
e	0	0	0	0	-	0	0	0	1	1
f	0	1	0	0	0	-	0	0	0	0
g	0	1	0	0	0	0	-	0	0	0
h	0	0	0	0	0	0	0	-	0	0
i	0	0	0	0	1	0	0	0	-	1
j	0	0	0	0	1	0	0	0	1	-

10.2.3 Regular equivalence

Regular equivalence is a restriction of structural equivalence according to which actors need to be connected to the same actors to be equivalent. Therefore, two actors are **regular equivalent** if they are equally related to equivalent others. Two nodes are said to play the same role (i.e., are regularly equivalent) if they have ties to the same roles.

For instance, we can distinguish two types of nodes: $\{b, c, d, e\}, \{f, g, h, i, j\}$.

10.2.4 Blockmodeling

Blockmodel (sometimes also **block model**) in blockmodeling (part of network science) is defined as a multitude of structures, which are obtained with:

- **identification of all vertices** (e.g., units, nodes) within a cluster and at the same time representing each cluster as a vertex, from which vertices for another graph can be constructed;

- **combination of all the links** (ties), represented in a block as a single link between positions, while at the same time constructing one tie for each block. In a case, when there are no ties in a block, there will be no ties between the two positions, that define the block.

The aim of blockmodeling is to produce a simplified or reduced matrix. We first arrange the rows and columns of the adjacency matrix so that structurally equivalent actors are grouped together. This grouping induces blocks within the matrix. For pure structural equivalence, the blocks are either all ones or all zeros and are called 1-blocks and $0-blocks$, respectively.

Some types of blocks can be:

- complete: filled with 1s;
- null: filled with 0s;
- regular: one covered rows and columns;
- row-dominant or col-dominant, whenever we can distinguish a block of nodes and links as a cluster (there exists at least one row/column with all 1s);
- row-functional or col-functional: all starting nodes link to some other nodes and vice-versa, if there is just one 1 in each row/column;
- row-regular or col-regular: each row/column is covered;

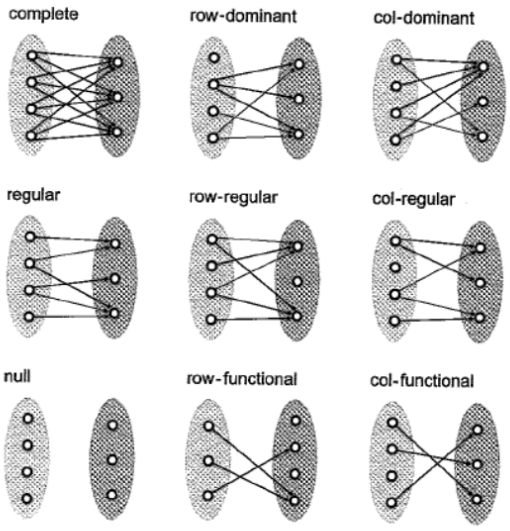


Figure 7.1. Nine types of connection between clusters.

Figure 10.4: Nine types of connection between clusters

10.3 Structural Equivalence on R

We can treat matrices as proximity measure and apply classification and clustering techniques.

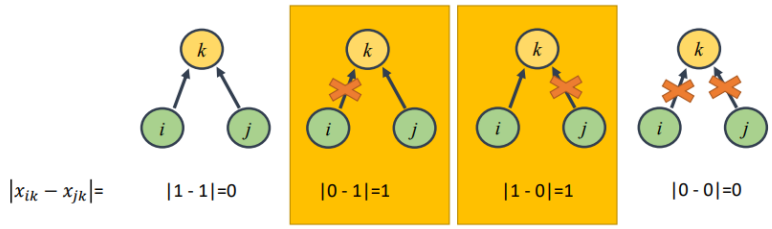
10.3.1 Hamming Structural Equivalence

$$\text{Hamming }_{ij} = \sum_{k}^{N} \left(|x_{ik} - x_{jk}| + |x_{ki} - x_{kj}| \right)$$

Suppose there are two nodes i, j and we want to compute the Hamming distance between the link of these two nodes with k:

- if both i, j are linked to k, then $|1-1| = 0$;
- if both i, j are not linked to k, then $|0-0| = 0$;
- if only one of them is linked to k, then $|1-0| = 1$.

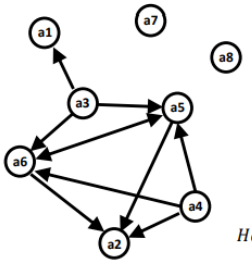
Note that this example covers only the situations with direct links from i, j to k, but also the vice-versa must be considered.




```
# Constructing the distance matrix with the Hamming distance
network_2_SEH<-sedist(network_2, method="hamming")
network_2_SEH
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    4    2    4    4    4    1    1
## [2,]    4    0    6    4    2    2    3    3
## [3,]    2    6    0    2    4    4    3    3
## [4,]    4    4    2    0    2    2    3    3
## [5,]    4    2    4    2    0    0    5    5
## [6,]    4    2    4    2    0    0    5    5
## [7,]    1    3    3    3    5    5    0    0
## [8,]    1    3    3    3    5    5    0    0
```

	a1	a2	a3	a4	a5	a6	a7	a8
a1	0	0	0	0	0	0	0	0
a2	0	0	0	0	0	0	0	0
a3	1	0	0	0	1	1	0	0
a4	0	1	0	0	1	1	0	0
a5	0	1	0	0	0	1	0	0
a6	0	1	0	0	1	0	0	0
a7	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0

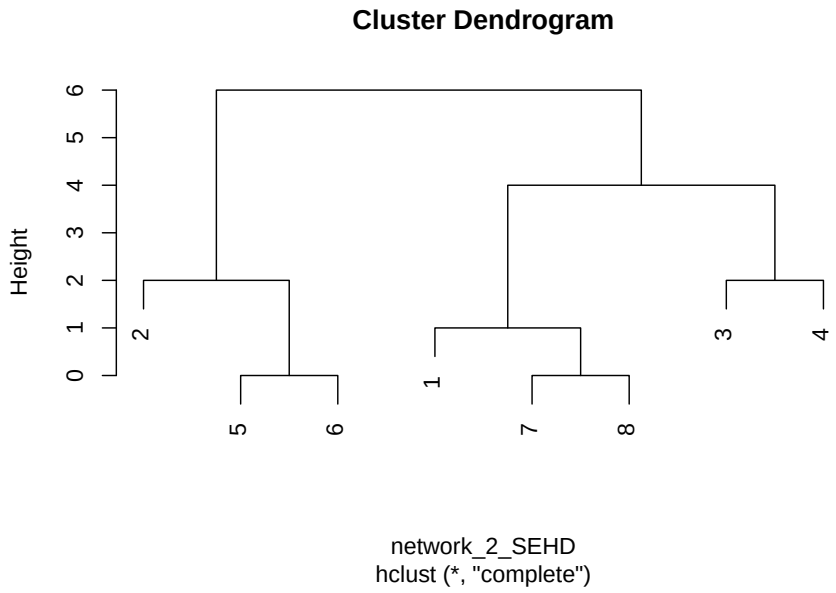


$$Hamming_{ij} = \sum_{k}^{N} (|x_{ik} - x_{jk}| + |x_{ki} - x_{kj}|)$$

	a1	a2	a3	a4	a5	a6	a7	a8
a1	0	4	2	4	4	4	1	1
a2	4	0	6	4	2	2	3	3
a3	2	6	0	2	4	4	3	3
a4	4	4	2	0	2	2	3	3
a5	4	2	4	2	0	0	5	5
a6	4	2	4	2	0	0	5	5
a7	1	3	3	3	5	5	0	0
a8	1	3	3	3	5	5	0	0

Figure 10.5: Same distance matrix obtained in the chunk

```
# Clustering
network_2_SEHD<-as.dist(network_2_SEH)

# Clustering on the Humming distance
network_2_SEHD_HC<-hclust(network_2_SEHD,method="complete")
plot(network_2_SEHD_HC)
```



10.3.2 Euclidean Structural Equivalence

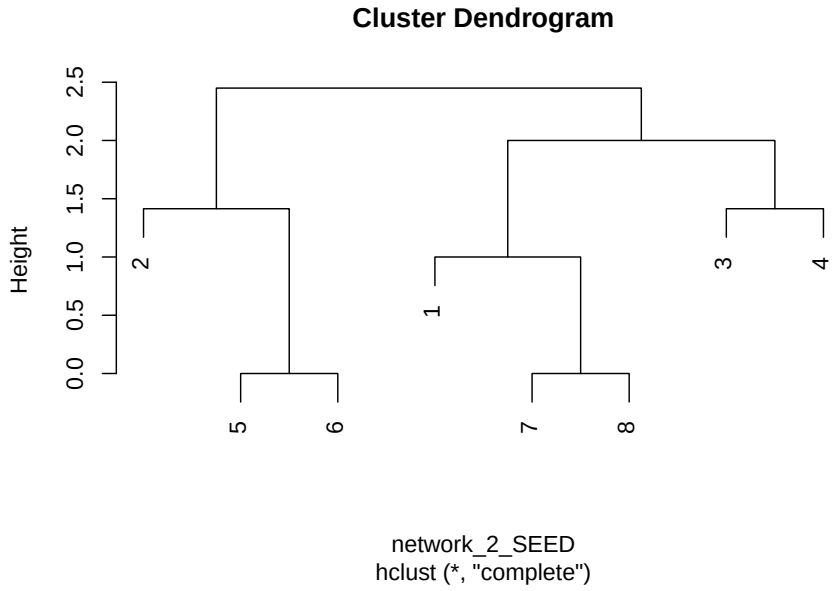
In this case, a pair of structurally equivalent actors would yield a distance of zero. Values close to zero would indicate that the actors involved are nearly structurally equivalent.

$$Euclidean_{ij} = \sqrt{\sum_{k}^{N} \left((x_{ik} - xjk)^2 + (x_{ki} - x_{kj})^2 \right)}$$

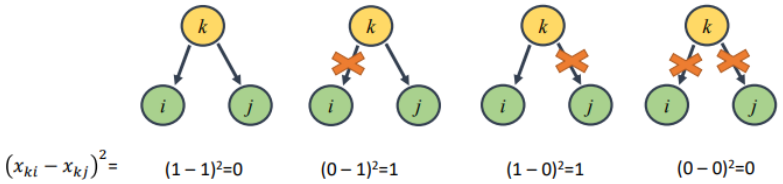
```
network_2_SEE<-sedist(network_2, method="euclidean")
network_2_SEE
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] 0.000000 2.000000 1.414214 2.000000 2.000000 2.000000 1.000000 1.000000
## [2,] 2.000000 0.000000 2.449490 2.000000 1.414214 1.414214 1.732051 1.732051
## [3,] 1.414214 2.449490 0.000000 1.414214 2.000000 2.000000 1.732051 1.732051
## [4,] 2.000000 2.000000 1.414214 0.000000 1.414214 1.414214 1.732051 1.732051
## [5,] 2.000000 1.414214 2.000000 1.414214 0.000000 0.000000 2.236068 2.236068
## [6,] 2.000000 1.414214 2.000000 1.414214 0.000000 0.000000 2.236068 2.236068
## [7,] 1.000000 1.732051 1.732051 1.732051 2.236068 2.236068 0.000000 0.000000
## [8,] 1.000000 1.732051 1.732051 1.732051 2.236068 2.236068 0.000000 0.000000
```

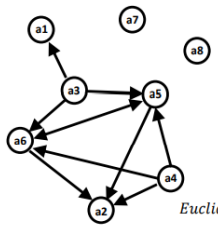
```
network_2_SEED<-as.dist(network_2_SEE)
network_2_SEED_HC<-hclust(network_2_SEED,method="complete")
plot(network_2_SEED_HC)
```



Suppose having two nodes i, j pointing to k. We aim to compute the euclidean distance between x_{ik} and x_{jk}.



	a1	a2	a3	a4	a5	a6	a7	a8
a1	0	0	0	0	0	0	0	0
a2	0	0	0	0	0	0	0	0
a3	1	0	0	0	1	1	0	0
a4	0	1	0	0	1	1	0	0
a5	0	1	0	0	0	1	0	0
a6	0	1	0	0	1	0	0	0
a7	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0



$$Euclidean_{ij} = \sqrt{\sum_{k}^{N} \left((x_{ik} - x_{jk})^2 + (x_{ki} - x_{kj})^2 \right)}$$

	a1	a2	a3	a4	a5	a6	a7	a8
a1	0.00	2.00	1.41	2.00	2.00	2.00	1.00	1.00
a2	2.00	0.00	2.45	2.00	1.41	1.41	1.73	1.73
a3	1.41	2.45	0.00	1.41	2.00	2.00	1.73	1.73
a4	2.00	2.00	1.41	0.00	1.41	1.41	1.73	1.73
a5	2.00	1.41	2.00	1.41	0.00	0.00	2.24	2.24
a6	2.00	1.41	2.00	1.41	0.00	0.00	2.24	2.24
a7	1.00	1.73	1.73	1.73	2.24	2.24	0.00	0.00
a8	1.00	1.73	1.73	1.73	2.24	2.24	0.00	0.00

Figure 10.6: Example of distance matrix computed with Euclidean distance, as shown in the chunk above

Chapter 11

Reciprocity and transitivity

11.1 Dyads

11.1.1 Dyad census

With **dyad census**, we count how many mutual, asymmetric and null dyads there are in the network. By looking at this network, there are:

- mutual dyads: 4;
- asymmetric dyads: 4;
- null dyads: $(6 \cdot 5)/2 - 8 = 7$ (removing all links, counting mutual dyads one time).

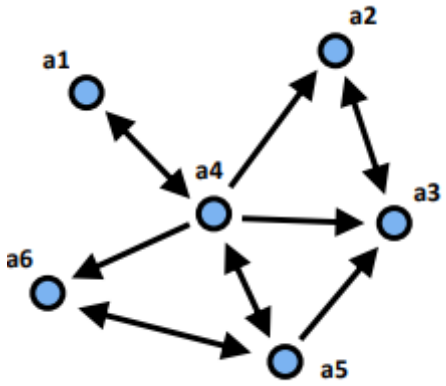


Figure 11.1: Example of directed network

Get the total dyads available as $n(n-1)/2$ and then remove the total amount of dyads actually existing (sum mutual and asymmetric's).

Given the density, what would be the expected dyad census?

- Mutual dyads: $d * d$;
- Asymmetric dyads: $2d * (1-d)$;
- Null dyads: $(1-d)(1-d)$;

In total, if all dyads existed in the network, the density would be:

$$\begin{aligned} d \cdot d + 2d \cdot (1-d) + (1-d)(1-d) &= d \cdot (d+1-d) + (1-d) \cdot (d+1-d) \\ &= d + 1 - d = 1 \end{aligned}$$

11.1.2 Reciprocity

Is there more reciprocity in the network than by chance?

There are two indexes for computing reciprocity:

- Compute the number of mutual dyads in the network and compare it for the total amount of possible mutual dyads ($2 \cdot mutual + asymmetric$):

$$\frac{2 \cdot M}{2 \cdot M + A}$$

  In the previous example, we would get $2 \cdot 4/(2 \cdot 4 + 4) = 8/12 = 0.667$.

  Suppose that the previous network doesn't have reciprocal edges, then reciprocity would be $0/(0+8) = 0$, because the index of reciprocity would be $2 \cdot 0/(2 \cdot 0 + A) = 0$. If instead all edges are mutual, then the highest reachable value is 1, because $2 \cdot M/(2 \cdot M + A) = 2M/(2M) = 1$.

  Note that UCINET adds the number of symmetric pairs as well as the reciprocates ties, including the case where there is a reciprocated zero in the adjacency matrix.

- Considering the density, we would obtain the reciprocity index by:

$$\frac{2 \cdot M}{2 \cdot M + A} = \frac{2d \cdot d}{2d \cdot d + 2d \cdot (1-d)} = \frac{d}{d + (1-d)} = d$$

11.1.3 On R

```
mat6<-matrix(c(0,0,0,1,0,0,
               0,0,1,0,0,0,
               0,1,0,0,0,0,
               1,1,1,0,1,1,
               0,0,1,1,0,1,
               0,0,0,0,1,0),6,6, byrow=T)

# Dyad Census
sna::dyad.census(mat6, g=NULL)
```

```
##      Mut Asym Null
## [1,]   4    4    7
```

```
# Reciprocity index
# Like edgewise but multiplied for m/(m+a)
sna::grecip(mat6,measure="dyadic")
```

```
##       Mut
## 0.7333333
```

```
# Reciprocity index
# Edgewise, 2m/(2m+a)
sna::grecip(mat6,measure="edgewise")
```

```
##       Mut
## 0.6666667
```

```
# Density
sna::gden(mat6)
```

```
## [1] 0.4
```

11.2 Triads

11.2.1 Triad Census

According to the Davis and Leinhardt triad census, there are 16 ways to arrange a triad in directed networks. The `sna` library on R helps to count how many of them there are in a network. The triads are labelled using the MAN convention:

- Mutuals, dyads with reciprocated ties;
- Asymmetrics, dyads with unreciprocated ties;
- Nulls, dyads with no tie.

For instance,

- *003: 3 non-null, 0 mutual and asymmetric;*
- *111D: 1 mutual, 1 asymmetric, 1 non-dyad*
- *021D: 0 mutual, 2 asymmetric, 1 null dyads.*

In addition, letters indicate:

- D: down arrows;
- U: up arrows;
- C: cycle;
- T: transitivity.

```
mat6<-matrix(c(0,0,0,1,0,0,
               0,0,1,0,0,0,
               0,1,0,0,0,0,
               1,1,1,0,1,1,
               0,0,1,1,0,1,
               0,0,0,0,1,0),6,6, byrow=T)

# Triad census
sna::triad.census(mat6)
```

```
##      003 012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
## [1,]   3   1   4    2    0    0    1    5    0    0   1    1    1    0   1   0
```

By considering the code output and the triad census legend, we could focus for instance on 111U triad and try to detect them:

- a_1, a_4, a_6
- a_1, a_4, a_2
- a_1, a_4, a_3
- a_2, a_4, a_5
- a_3, a_5, a_6

11.2.2 Transitivity

In order to know whether a network is transitive, we can compare the transitivity computed through `gtrans()` with the density of the network. Transitivity is computed as the ratio between a dyad and the number of dyads between these two nodes plus the number of tryads there are:

$$Transitivity = \frac{\sum_{i,j,k} x_{ij}x_{jk}x_{ki}}{\sum_{i,j,k} x_{ij}x_{jk}}$$

To simplify, we count the proportion of triads in which $A \to B \to C \implies A \to C$ over the total triads that go from A to B and from B to C.

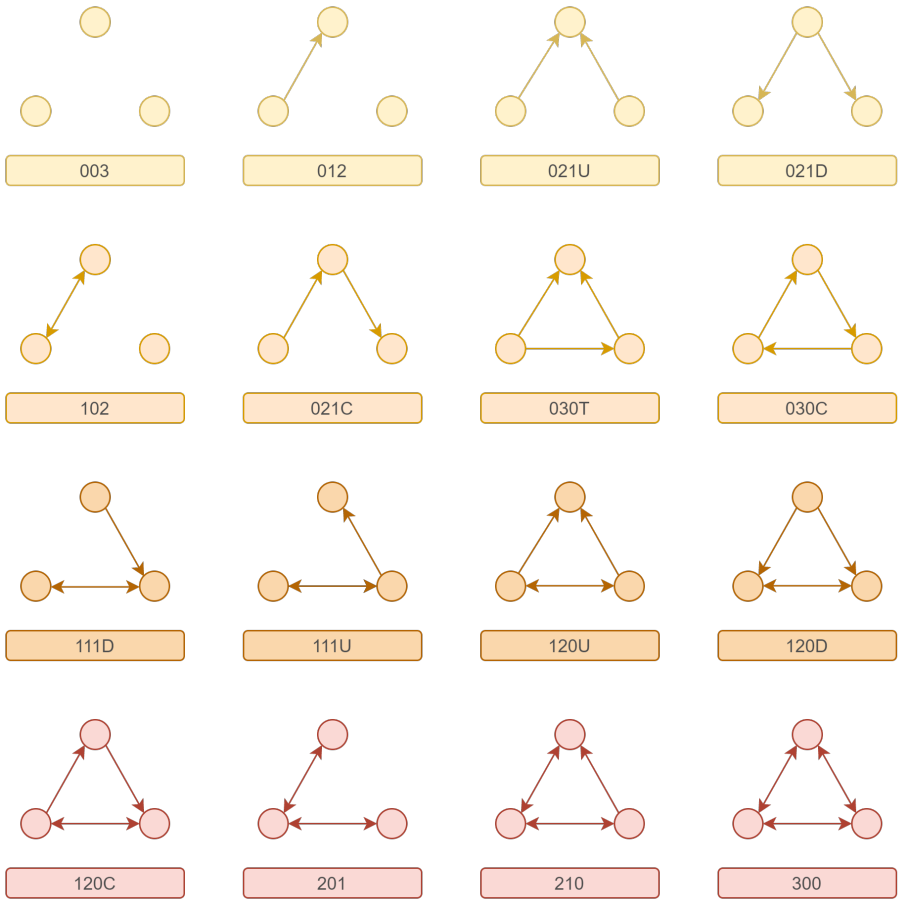


Figure 11.2: MAN Nomenclature

```
# Transitivty
sna::gtrans(mat6)
```

```
## [1] 0.4375
```

```
# Density
sna::gden(mat6)
```

```
## [1] 0.4
```

```
# Also density
sum(mat6)/(nrow(mat6)*(nrow(mat6)-1))
```

```
## [1] 0.4
```

Watts and Strogatz proposed a measure for undirected networks called **clustering coefficient** to capture the extent to which a network had areas of high and low density:

- measure the density of ties in each node's ego network;
- average the qauntity across all nodes to get the overall clustering coefficient.

If the clustering coefficient is weighted by the number of pairs of nodes in every ego's network it equals the transitivity coefficient. Watts and Strogatz use this coefficient to define small-world networks.

Chapter 12

ERGMs

12.1 Introduction

Suppose we collect data about 7 females and 6 males. Why should we end up observing a specific network instead of another?

The probability of observing the network x equals:

$$P(X = x) = \frac{exp(\theta' z(x))}{\kappa(\theta)}$$

Consider the network as the aggregated result of **tie-based decisions** among pairs of nodes. Everyone behaviours according to the same set of rules, constructing a network with a specific structure (i.e. **homophily effect**).

These tie-based decisions are driven by a **combination of social forces** which are not directly observable, therefore the observed network is useful to get an idea of how they're made. We only observe the aggregated outcome of the combination of hidden forces. The aggregation of these dyadic outcomes will contain some randomness.

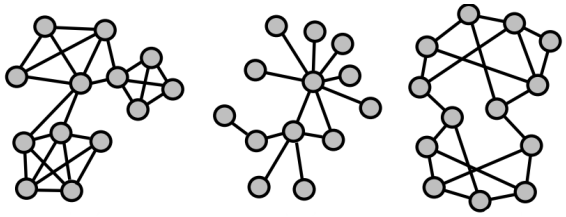
$$\begin{aligned}\Pr\left(X_{ij} = 1 \mid \mathbf{X}_{ij}^{c}\right) &= \frac{\Pr\left(\mathbf{X} = \mathbf{x}_{ij}^{+}\right)}{\Pr\left(\mathbf{X} = \mathbf{x}_{ij}^{+}\right) + \Pr\left(\mathbf{X} = \mathbf{x}_{ij}^{-}\right)} \\ &= \frac{\exp\left\{\theta' \mathbf{z}\left(\mathbf{x}_{ij}^{+}\right)\right\}}{\exp\left\{\theta' \mathbf{z}\left(\mathbf{x}_{ij}^{+}\right)\right\} + \exp\left\{\theta' \mathbf{z}\left(\mathbf{x}_{ij}^{-}\right)\right\}}\end{aligned}$$

As Borgatti et al. states in Analyzing Social Networks, these models are related to more general linear models of standard statistics but deal with the fact that we cannot assume independence of observations. Conditional dependence is related to the fact that if two edges share a common vertex, then they are dependent, conditional on the rest of the graph. These models are called Markov random graphs models.

12.1.1 Density - Is there an overall tendency to build or refrain from building ties?

We could refer to this question with the density. Consider that connections may require time and energy, but also provide positive benefits.

By looking at the following networks, we can see three different networks. The number of ties is respectively $25, 13$ *and* 20*. The density is computed as the number of ties over the total number of ties there could be (i.e.* 78*). Therefore, the density equals* $0.32, 0.167, 0.256$*.*



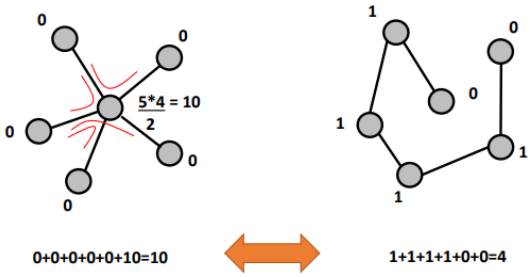
If we observe a low number of ties in our network (given network size), then the underlying force was probably one that tended not to build ties, rather than build ties. However, there also might be a high tendency to build ties in a network and nevertheless we might still end up with a low number of realized ties.

12.1.2 Centralization - Is there a tendency for popular people to obtain more ties?

If so, popular nodes might be more attractive, otherwise, there are limits to the level of variation in degree. According to the **Matthew effect**, the rich get richer and the poor get poorer, also called preferential attachment.

In order to answer this question, we should focus on the number of 2-stars as a measure of centralization. Consider two networks that have roughly the same density and the same number of nodes.

For instance, by considering the two networks below, despite both networks having similar density, the 2-stars is very different.



If we observe a high number of two-stars in our network given the number of ties, then the underlying force was probably one that tended to prefer to build ties to nodes with a high degree.

In the previous 3 networks, the 2-stars measure is $77, 40$ *and* 43.

12.1.3 Clustering - Is there a tendency for triadic closure?

If not, open structures provide access to unique alters; au contraire, triadic closure is easy and provides a feeling of safety. By looking at the number of closed triads, we could answer this question.

In the previous 3 networks, the number of closed triads is $17, 1$ *and* 1.

If we observe a large number of closed triads in our network, given the number of ties and 2-stars, then the underlying force was probably one that tended to prefer to build ties between nodes that have common connections.

12.2 How to get values for these forces

In order to figure out the values above, count specific properties in the observed network to provide information about local social forces that determines the structure of the network.

12.2.1 Introduction

Logit models are expressed in the following form:

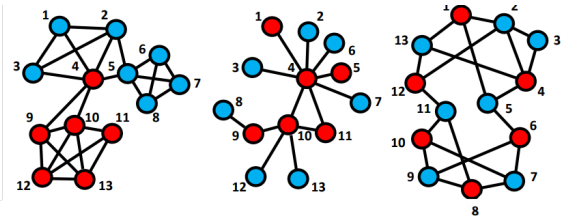
$$Pr(X = x) = \frac{exp(\theta' z(x))}{\kappa(\theta)}$$

- θ is a vector of model parameters;
- $z(x)$ is a vector of network statistics.

They come from the exponential form, in which the probability function depends on an exponential function of a linear combination of network statistics.

Suppose we want to know whether a tie emerges because of an indirect connection based on two other ties. We cannot simply perform a logit regression because this reasoning could be applied to all the ties in the triad. That's why MCMC (Markov Chain Monte Carlo) are necessary.

12.3 Notal attributes



Suppose we detain a network whose nodes have attributes (e.g. gender-based):

- *Is there a tendency for red to be connected to red and blue to blue?*

  We could compute it by looking at the number of **homophilous** ties. It's the tendency to be connected to nodes similar to us. If there's high homophily, the same types are easier to connect to, otherwise, different types hold complementary resources. By looking at the three networks, we know that homophilous ties are:

  - $\#H = 21$ homophily
  - $\#H = 6$ neutrality
  - $\#H = 1$ heterophily

- *Is there a tendency for males (red nodes) to have more ties than females (blue nodes)?*

  Being of different types may change the behaviour of nodes? In the three examples, we know that:

  - red is slightly more connected;
  - red nodes are more connected;
  - all nodes are almost equally connected.

  The difference can be detected through the **difference in average degrees for red and blue:**

  - reds: 26/5, blues: 24/6;
  - reds: 19/5, blues: 7/6;
  - reds: 19/5, blues: 21/6.

12.3.1 Example with MPNet

Through MPNet it's possible to run ERGMs to test the multiple effects at the same time. It provides edges, 2-star, triangles (closed triads), gender activity (i.e. the distinction between males and females) and gender interaction (whether the gender difference can be also found on the number of ties):

- the first network has particular significance for the 2-stars, triangles and gender interaction;
- the second network has particular significance just for stars and triangles;
- the third one is just for the 2-stars.

12.4 Pseudolikelihood

Rather than consider the probability of observing the graph, the pseudo-likelihood:

$$P\left(X = X_{\text{obs}}\right) = \frac{e^{\theta_L \ \#L_{\text{obs}} + \theta_S \ \#S_{\text{obs}} + \theta_T \ \#T_{\text{obs}}}}{\sum_{k=1}^{\text{All}} \left[e^{\theta_L \#L_k + \theta_S \ \#S_k + \theta_T \ \#T_k}\right]}$$

Consider the presence or absence of a tie between actors i and j as a function of the rest of the graph. Given the rest of the graph, what is the likelihood of x_{ij} being present?

$$\begin{aligned} P(x_{ij} = 1 \mid X_{ij}^C) &= \frac{P\left(X = X_{obs}^{i} = 1\right)}{P\left(X = X_{obs}^{i=0}\right) + P\left(X = X_{obs}^{i=1}\right)} \\ &= \frac{\frac{e^{\theta_L \ \#L_{\text{obs}}^{ij=1} + \theta_S \ \#S_{\text{obs}}^{ij=1} + \theta_T \ \#T_{\text{obs}}^{ij=1}}}{\sum_{k=1}^{\text{All}} \left[e^{\theta_L \#L_k + \theta_S \ \#S_k + \theta_T \ \#T_k}\right]}}{\frac{e^{\theta_L \ \#L_{\text{obs}}^{ij=0} + \theta_S \ \#S_{\text{obs}}^{ij=0} + \theta_T \ \#T_{\text{obs}}^{ij=0}}}{\sum_{k=1}^{\text{All}} \left[e^{\theta_L \#L_k + \theta_S \ \#S_k + \theta_T \ \#T_k}\right]} + \frac{e^{\theta_L \ \#L_{\text{obs}}^{ij=1} + \theta_S \ \#S_{\text{obs}}^{ij=1} + \theta_T \ \#T_{\text{obs}}^{ij=1}}}{\sum_{k=1}^{\text{All}} \left[e^{\theta_L \#L_k + \theta_S \ \#S_k + \theta_T \ \#T_k}\right]}} \end{aligned}$$

$$P(X_{ij} = 1 | X_{ij}^c) = \frac{e^{\theta_L \ \#L_{\text{obs}}^{ij=1} + \theta_S \ \#S_{\text{obs}}^{ij=1} + \theta_T \ \#T_{\text{obs}}^{ij=1}}}{e^{\theta_L \ \#L_{\text{obs}}^{ij=0} + \theta_S \ \#S_{\text{obs}}^{ij=0} + \theta_T \ \#T_{\text{obs}}^{ij=0}} + e^{\theta_L \ \#L_{\text{obs}}^{ij=1} + \theta_S \ \#S_{\text{obs}}^{ij=1} + \theta_T \ \#T_{\text{obs}}^{ij=1}}}$$

The same formula can be defined for $X_{ij} = 0$.

$$\begin{aligned} \frac{P\left(X_{i,j} = 1 \mid X_{i,j}^c\right)}{P\left(X_{i,j} = 0 \mid X_{i,j}^c\right)} &= \frac{e^{\theta_L \ \# \ L_{obs}^{ij=1} + \theta_S \ \# \ S_{obs}^{ij=1} + \theta_T \ \# \ T_{obs}^{ij=1}}}{e^{\theta_L \ \# \ L_{\text{obs}}^{ij=0} + \theta_S \ \# \ S_{\text{obs}}^{ij=0} + \theta_T \ \# \ T_{\text{obs}}^{ij=0}}} \\ &= e^{\left[\theta_L \ \#L_{\text{obs}}^{ij=1} + \theta_S \ \#S_{\text{obs}}^{ij=1} + \theta_T \ \#T_{\text{obs}}^{ij=1}\right] - \left[\theta_L \ \#L_{\text{obs}}^{ij=0} + \theta_S \ \#S_{\text{obs}}^{ij=0} + \theta_T \ \#T_{\text{obs}}^{ij=0}\right]} \\ &= e^{\theta_L \left[\#L_{obs}^{ij=1} - \#L_{obs}^{ij=0}\right] + \theta_S \left[\#S_{obs}^{ij=1} - \#S_{obs}^{ij=0}\right] + \theta_T \left[\#T_{obs}^{ij=1} - \#T_{obs}^{ij=0}\right]} \end{aligned}$$

By considering the logarithm, we get a simple logistic regression equation. Does the local surrounding predict the presence or absence of a tie between i and j?

- θ_L: what is the overall chance of a tie? Through density
- θ_S: if actors i and j have more connections, does this increase the chance of a tie between i and j? Through 2-star
- θ_T: if actors i and j share the same others, does this increase or decrease the chance of a tie between i and j? Triangle

12.4.1 Problems

Classic regression assumes independent cases. Pseudolikelihood estimates are known to be wrong:

- Overestimation of significance;
- Often overestimation of effects and sometimes completely wrong.

The alternative is to approximate MLE through the MCMC algorithm.

12.5 MCMC

12.5.1 Introduction

We observed a network, and assume that it is the result of a combination of forces, e.g.:

- an overall tendency to build ties (or not);
- a preference to build ties to others with high degree (or not);
- and a tendency for closure (or not).

These forces are captured by the parameters $\theta_L, \theta_S, \theta_T$:

$$P(X = X_{\text{obs}}) = \frac{e^{\theta_L \#L_{\text{obs}} + \theta_S \#S_{\text{obs}} + \theta_T \#T_{\text{obs}}}}{\sum_{k=1}^{\text{All}} [e^{\theta_L \#L_k + \theta_S \#S_k + \theta_T \#T_k}]}$$

Any values for these parameters could lead to a particular network, but some are more likely to generate it. We aim to find those that maximize the chance of observing the graph. To approximate the MLE values for these three parameters, Markov Chain Monte Carlo will be used.

12.5.2 Logic

This approach is based on the Markov Chains. We can simulate a network through random graphs:

1. Start with any network with a specific number of nodes;
2. Select a random dyad at a time;
3. Update the selected dyad using a formula, based on the current state in the surrounding only:

$$P\left(X_{ij}^{t+1} = 1 \mid x_{ij}^{C,t}\right) = \frac{e^{\theta_L \left[\#\ L_t^{ij=1} - \#\ L_t^{ij=0}\right] + \theta_S \left[\#\ S_t^{ij=1} - \#\ S_t^{ij=0}\right] + \theta_T \left[\#\ T_t^{ij=1} - \#\ T_t^{ij=0}\right]}}{1 + e^{\theta_L \left[\#\ L_t^{ij=1} - \#\ L_t^{ij=0}\right] + \theta_S \left[\#\ S_t^{ij=1} - \#\ s_t^{ij=0}\right] + \theta_T \left[\#T_t^{ij=1} - \#T_t^{ij=0}\right]}}$$

4. If we use the same rules for a long enough time, the Markov chain forgets about history (forgets the starting network), i.e., only takes into account the rules of the process/game and the networks generated will tend to go towards a stable "equilibrium" (i.e. similar networks). Time after time, the algorithm will show a different characteristic of the network;
5. If we use the estimates that truly reflect the forces which are most likely to have led to our observed network, then the process will generate networks that are similar to our observed network. That is provided we run this long enough so it forgets about the start = burn-in. Of course in reality we do not know the values for the estimates but we can search for them in this way.

Chapter 13

Christakis and Valente

13.1 Christakis

13.1.1 What is clustering for Christakis?

A cluster is a group of people who are connected and are similar in characteristics.

For instance, Christakis detects obese and non-obese people.

13.1.2 What are 3 potential types of reasons for finding clustering of obese and non-obese people?

- Induction effect or Social Contagion: spread from a person to a person

  If your friends gain weight, you gain weight, and you then, in turn, impact other friends. Because we conform and we change our idea about what acceptable body size is.

  If your friends smoke, you might start smoking.

- Homophily effect: birds of a feather flock together

  Form a tie because we have a similar body size, to begin with, or similar interest in exercise (etc.) that might predict our body size.

  Friendship emerging as a result of smoking behaviour.

- Confounding factor: it confounds our ability to figure out what's going on

  Something happening in a cluster's environment (e.g. a gym opening, which makes all members of the cluster lose weight, or a fast-food restaurant opening, which makes all members gain weight).

  Environment makes both of us start/stop smoking (like an anti-smoking advertisement, or new laws that impacts one group but not another).

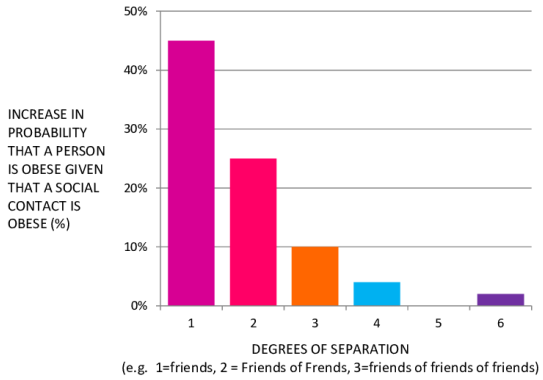
Some examples that can be applied to these three reasons are:

- movies preferences;
- happiness;
- smiling;
- eating fast food.

13.1.3 What is the meaning of 3 degrees of separation?

The plot here shows, on the Y-axis, the increase in the probability that a person is obese given that a social contact of theirs is obese and, on the X-axis, the degrees of separation between the two

people. According to Christakis, there are three degrees of separation that hold for many traits, in particular, obesity:



- If your friends are obese you have about a 45% higher likelihood of being obese yourself;
- if your friend's friends are obese you have about a 25% higher likelihood;
- if your friend's friend's friends are obese you have about a 10% higher likelihood;
- up to 3 degrees of separation is there a real relation between your body size and that person's body size (not really any more at 4 degrees away).

13.1.4 What is a multi-centric epidemic?

Obesity is a **multicentric epidemic** since there is no patient zero of this epidemic. In general, lots of people are doing things at the same time and if we focus on a certain phenomenon, we may notice that there may be a key player, but no central node that indicates the starting node.

For instance, the 2008 crash was due to a single bank, therefore it was a uni-centric epidemic.

For instance, the unemployment crisis is instead a multicentric epidemic, since there is no starting node for this phenomenon.

Also, the coronavirus had a patient zero that started to diffuse the pandemic. However, inside each new country, there's a single patient zero, therefore it is a multi-centric epidemic.

13.1.5 What might be potential mechanisms for the spread of behaviour?

Alter's appearance or behaviour changes:

- Ego's behaviour

  For instance, related to smoking, seeing other people smoke makes us smoke.

  Related to mask-wearing, we notice other people wearing masks, friends specifically, and we change our behaviour accordingly.

- Ego's expectations or perceptions of norms.

  For instance, related to smoking, we are convinced by arguments from others about what is normal and expected. Life is short and there are so many other dangers, you should worry about smoking.

  Related to mask-wearing, we notice contagion graphs, indications in shops and public places and people may tell us to wear the mask and that we can enter only if we wear it. That's how our behaviour changes.

13.2 Valente

This speak is about the adoption of new ideas or practices that flows through interpersonal contacts. In particular, he speaks about:

- models of the diffusion of innovations;
- network models for diffusion (Network models of change);
- network interventions (Network models for a change).

13.2.1 What is the classic idea of diffusion?

Studying how networks influence behaviour moves us from theories about networks to network theory. New ideas and practices originate enter communities from some external source (e.g. mass media, labour exchanges, cosmopolitan contact, technical shifts). Adoption of the new idea or practice then flows through **interpersonal contact networks**.

Diffusion happens out of interpersonal contact, persuasion and communication modelling. Diffusion occurs over time, usually following a trend studied in the Economy about the adoption of new products or technology, where there are nearly adopters, then diffusion and in the end the large common use/adoption of a behaviour.

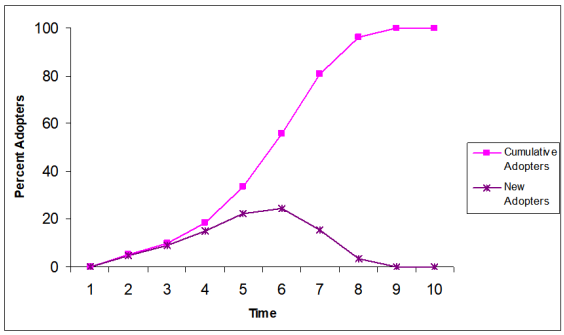


Figure 13.1: Stages of adoption over time

Valente created mathematical models of the diffusion of innovations in different situations: for external and internal influence and when the external communications are mixed with person-to-person contact. However, it is far more interesting to study the processes that occur amongst people in the network.

Therefore, Valente created a hypothetical community of people and the percentage of diffusion when adopters persuade non-adopters at a rate of one per cent. When this happens, we get the S-shape curve described above. But this is wrong since there's no network in this. Once we start to see there's clustering in the network, we notice that it all goes in a different way. In a random way, we get the S-shape, in a real network, there are some moments where the diffusion speeds up or slows down, deviating from the model.

13.2.2 What types of network influence weights are there according to Valente?

Valente also describes the generative engine for understanding how networks influence behaviour.

1. **Direct influence**

   When you're exposed to people with different behaviours, you are more likely to change your behaviour. Not only actual use but also perceived use influences behaviour.

   For instance, if in a network of people there are non-users and users and they interact with each other it is likely that all of them become users.

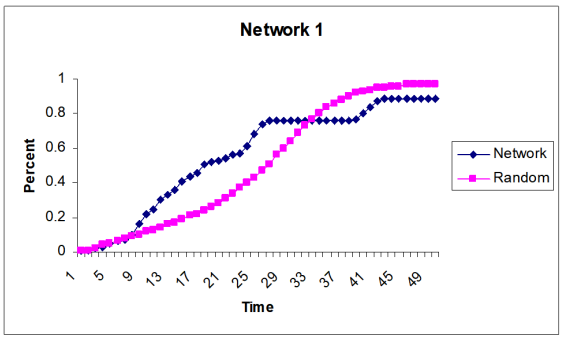
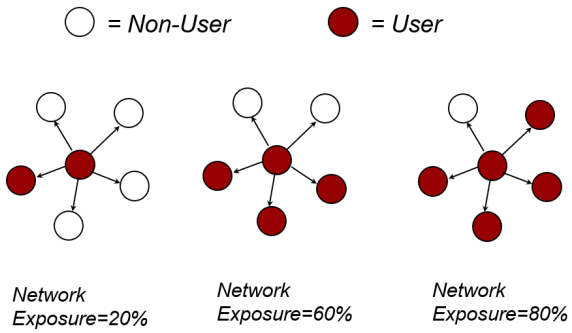


Figure 13.2: S-curve to describe random behaviour adoption



2. **Indirect ties**

   Indirect exposures may matter in influencing others, but there's kinda debate if there's really an effect and how does it work. It varies by network and behaviour type. We need some quality data study to make any statement about the role of indirect influences.

3. **Structural equivalent ties**

   On the other hand, structural equivalence can matter according to Burt. We might be influenced by other people too who we're not connected, but occupy the same position in the network and therefore we tend to monitor their behaviour and situation. In some cases, competition might be there (e.g. compete for the attention of a friend) or not (e.g. occupy friendship positions and be influenced one another through direct communication).

4. **Tie strength**

   Stronger ties exert a stronger influence on ego than weaker ties. They have more influence on the ego's behaviour than those he/she sees less frequently since they have a less emotional relationship with them.

   Weak ties are important for information spread and rumours, but when it comes to behavioural change, strong ties make the difference.

5. **Simmelian ties**

   Empirically, simmelian ties also affect people. Suppose A and B are connected together and they have a stronger influence on the ego with respect to C or F which are not connected to any other node in the ego network. Therefore, the social environment matters a lot in terms of the amount of influence that alters can exert on egos.

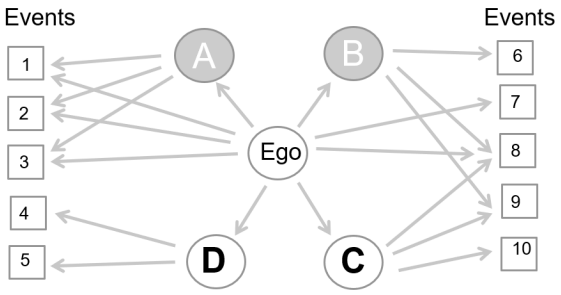
   If all my friends are friends with each other, their behaviours are more likely to influence me than if my friends are not friends with one another, because their decisions, behaviours and communications are all reinforcing one another rather than being disparate.

6. **Density weighted or Centrality Measures**

We can weigh our influences by using centrality measures, which also allows us to understand who's the ego in the network and if this fact is associated with his/her traits. The most used centrality measures are degree (in and out), closest and between centrality. There are over 100 centrality measures to use.

7. **Degree weighted**
8. **Joint participation**

   Whenever two or more nodes in the network participate jointly in an event, the influence on one another on its behavioural change is stronger than if they participate individually in the event.



9. **Attribute weighted**

   In diffusion theory there's the thought that homophily is associated with a greater social influence: if we're similar, the flow of information and communication is easier and more frequent.

   There may be some behaviours that are actually attribute specific, therefore it is more likely that the influence would occur when people are similar on the attribute.

10. **Thresholds**

    Some people are willing to adopt a new behavior when a minority or none of their network partners are willing to (e.g. early adopters that discover all the new features and they communicate interesting facts inside the network). Other people wait until their network is filled with users before they're willing to try something (e.g. late adopters that need to adapt inside the network in order to homologate and gain benefits).

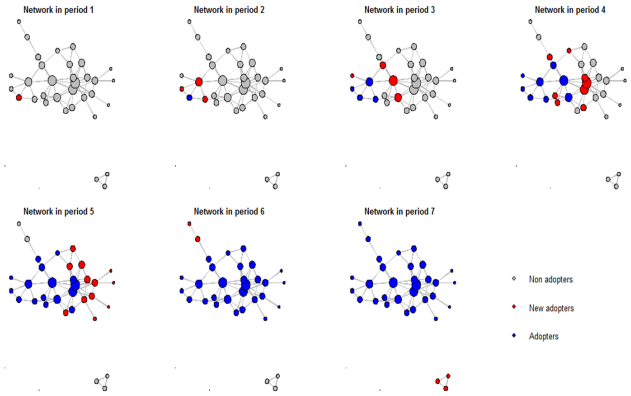
    People late in the diffusion process are late because of their position in the network, which gives them access to the behaviour or the information very late in the process. It still may happen to be connected to early adopters despite being late adopters.

    According to Valente's Threshold model, in the x-axis there's the time in which a node adopts a behaviour, while the y-axis indicates the percentage of people that adopts a behaviour, i.e. threshold. For each time, a node adopts the behaviour. The high threshold people only get affected if people around are affected by the behaviour, while low threshold people are called early adopters and do not need to be influenced to adopt a behaviour. Also, we can distinguish between early and late adopters, while low and high threshold relies on the affection of other people.

13.2.3 Network Diffusion

It is possible to use a `NetdiffuseR` package to track the diffusion of information. In particular, as shown in the image below, in red there are new adopters, while in blue continuing users and in grey users that do not use or have a behaviour.

What happens over time during the diffusion process?



- External influence (e.g. media, advertisement, travel), whose effect decreases over time. As more people adopt, we do not worry about external influence;
- Selection, which is very high in the beginning , because in the beginning nobody does nothing in my network. Over time, as the community fills with adopters, we don't need to make changes in the network in order to have people being consistent with the ego's behaviour;
- Influence should increase over time: as the behaviour becomes normative, as there are more adopters, I'm more likely to change my behaviour;
- Opinion leaders, whose trend varies, when the behaviour is normative and compatible with the community. When it is not culturally compatible, opinion leaders delay their adoption.

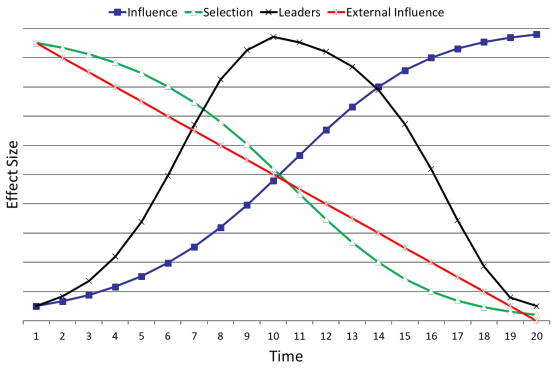


Figure 13.3: Phenomena that happen over time during the diffusion process and their development

Chapter 14

Ego's Network

We will go through the process of designing and conducting a social network study.

En ego network is the part of a network that involves a particular node we are focusing on, which we call ego. This network consists of ego, the nodes ego is connected to (alters), and usually the ties between ego's alters.

14.1 Finding a research topic

A research topic can be considered good if interesting, relevant, has not been answered or researched before or is measurable.

For instance, we could focus on people who live abroad.

14.2 Identify a research question

Related to people who live abroad, some research questions could be

How do people experience moving home and why? What factors make it good or bad for them? How do Hungarian returning migrants experience their return and what are the main factors that affect their experiences?

Note that if the research question doesn't allow you to restrict the set of alters that a respondent could name, then it is better to use a personal-network research design. In addition, whenever we consider a sociological group, we automatically restrict the actors in our network.

14.3 Think about relevant theories

Is there a theory that directly addresses or answers our question?

Are there any sociological theories or concepts that help us think about why some people are happier in general or cope better in certain situations? Maybe something you learnt in class?

14.3.1 Coleman - Closure

Let's consider the social capital theories. James Coleman in 1988 focuses on the function of social capital, which according to the economic sociology is based on a rational choice approach: it attempts to introduce social structure into his analysis by employing social capital and social norms to explain how individual rational actions translate into systematic collective action.

It is not a single entity, but a variety of entities, having two characteristics in common: they all consist of some aspect of social structure, and they facilitate certain actions of individuals who

are within the structure. There are three forms of social capital: *obligations and expectations*; *information channels*; *social norms*.

A key concept is about **closure or bonding capital**, which facilitates obligations and expectations and social norms and takes the form of trust, social norms of reciprocity and cooperation.

14.3.2 Burt and Granovetter - Bridging

According to Burt and Granovetter, Social capital is is a resource that is embedded in social networks and is accessible to individuals through their social connections. Information is the key resource in social networks.

People with strong ties come from the same social circle thus have access to the same information. However, weakly related ties are members of other social groups and have access to different, non-redundant information. The key concept is bridging or the presence of weak ties.

14.3.3 Nan Lin

According to Nan Lin, social capital can be defined as resources embedded in a social structure which are accessed and/or mobilized in purpose action.

It cuts through the debate of bridging vs closure: differentiates two different kinds of returns to social capital, that result from different mechanisms:

- Instrumental returns to social capital, which arises from bridging capital and enables individuals to access resources they previous did not have (e.g. money, wealth, reputation, power);
- Expressive returns to social capital, which arises from bonding capital (closure) and enables individuals to preserve previously existing resources (e.g. physical health, mental health, life satisfaction).

14.3.4 Social Support

Social Support is closely related to Social Capital, being an inherently relationship-based concept. It highlights the assistive nature of personal relationships. It can be defined as people are surrounded by a variety of social ties, which provide them with different supportive resources.

Different forms of social support (to capture the diversity) are:

- Emotional support: someone you can talk to when you have a problem;
- Instrumental or practical support: someone can help you with your groceries when you brake your leg. Can also include information;
- Financial support: someone can lend you some money when you are bankrupt;
- Social companion: someone you can spend your free-time with, for example grab a beer with after class.

14.4 Data collection

There are two major kinds of network research designs:

- **Socio-centric or whole network design**

  The focus is on the ties among all pairs of nodes in a given set. We get a single and complete network, but the cost of the design increases with the network size.

  It captures the position of individuals within the social structure. It can measure indirect or absent ties.

- **Ego-centric or personal network design**

  We sample to obtain a set of respondents and then collect from each respondent (ego) the list of people (alters) they are connected to, the nature of the ties connecting them, characteristics

of these alters, and the respondent's perceptions of the ties among the alters. Whenever we capture egos ties, we reach their alters.

It also captures the position of individuals within the social structure but its limited to direct ties. It simplifies the boundaries of the network, making more manageable to collect data.

14.5 From theory to practice

- **Socio centric**

  Requirements:

  It requires the identification of a boundary inside a population. It's important for individuals' position within the network. As a result, it is focused on a social domain. It requires the collection of all ties of all nodes within the population to every other node within the population.

  Privacy:

  There are however ethical concerns due to the lack of anonymity, otherwise it is not possible to construct the whole network.

  External validity:

  We cannot generalize our assumptions for all networks, but just for the whole one we're studying, since each one of them may have different dynamics.

  Let's say we make a network study of your network class here and conclude that the tallest person has the most central position. Does it hold for your other classes too? And for other universities? Also for undergraduate students? For other network classes at other universities? Since we didn't sample students from all Italian (or European) universities, we cannot generalize.

- **Ego centric**

  Requirements:

  Each person lives in a personal community that's unique to them (i.e. social context). This social context affects individuals in multiple ways. We want to collect data from the individual's (ego) perspective to understand their social context.

  We don't have to artifically define the boundaries of the network – not restricted to a group. It can also span multiple social domains. It's typically easier to collect these data instead of the whole network. Each person nominated by default has a connection to the interviewed individual. Connection between nominated individuals may or may not be collected.

  Privacy:

  Respondents' identity can be kept anonymous and also the nominated people (they can use pseudonyms).

  External validity:

  When egos are randomly selected from a wider population we may be able to generalise our findings.

  For instance, a random selection of MA students from all italian universities are interviewed about their friends both within and outside their university.

14.6 Select the right network approach

In order to choose between the socio or the ego-centric approach we must consider 2 main criteria:

- Which one suits better for assessing our theoretical approach or answering our research question?
- Practical considerations involved in data collection.

14.6.1 Ego-centric network

The ego-centric network, also called **ego-networks**, is a network where the person whose network is collected is called ego. The person who are nominated by the ego are called alters.

In order to collect data for ego-networks:

1. Name generator: who are the members of the network? who are the ego alters? Remember to set a time frame and to limit the size of the network. generally, it consists of a series of open-ended questions designed to generate the names of people;
2. Name interpreter: what alter attributes are relevant? We ask the respondent about each name that came up in the generator, requiring attributes and qualities of ego's relationship with this alter;
3. Name interrelater: how do we define who knows whom in the network? is it weighted or not? We ask the respondent about ties between the alters. Notice that the ego may have limited knowledge of the ties among alters and the task can be time-consuming.

14.6.2 Generating names of ego-networks

The aim is to develop a list of distinct names that we can systematically ask the respondent about.

1. Exchange based name generators: elicit ties to provide access to particular resources;
2. Content based name generators: interaction between partners with particular attributes, roles or shared experience;
3. Affect based name generators: elicit intimate alters that are mostly influential;
4. Interaction based name generators: measure the sociability or social isolation.

14.6.3 How to collect data?

- Questionnaires
- Interviews
- Participant-aided sociograms;
- Target method.

Part II

Laboratory

Chapter 15

Visualization

15.1 Prerequisites

Before starting to visualize networks, it is necessary to install the package for Social Network Analysis, called `sna`:

```
library(sna)
```

15.2 Modifying the layout on KHF network

15.2.1 Random layout

Let's plot the nodes randomly in the two-dimensional space, with the KHF dataset.

Notice that in order to build a network, we need an adjacency matrix, that's why the csv is converted into a matrix structure. Then, in order to symmetrize it and make it undirected, it is multiplied for its transpose, to make a self-multiplication, in order to get an undirected network from a directed one (remember that if there's a link going in both sides, we can insert an undirected link between these two vertices).

For instance, consider the following directed adjacency matrix:

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

By computing its transpose and perform the self multiplication we obtain the undirected network (symmetric and with links only whenever the link goes on both sides in the original directed network):

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

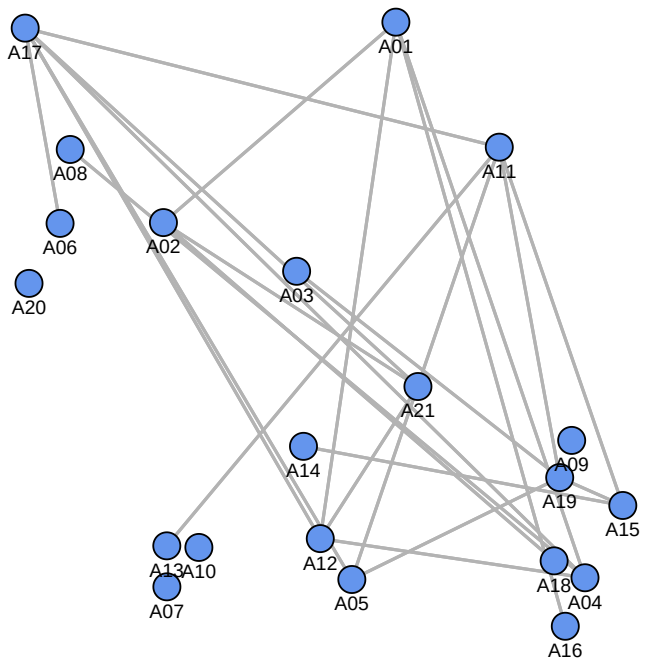
In the end it is turned into a network object.

```
# 1. Read in the friendship network for Krackhardt's high-tech managers:
KHF <- as.matrix(read.csv("datasets/Krackhardt_HighTech_Friendship.csv",
                          stringsAsFactors=FALSE, row.names=1))


# 2. Symmetrize the matrix using the minimum approach, so a friendship tie
```

```
# is only considered if both agree they are friends:
KHFS <- KHF*t(KHF)

# 3. Turn this into a "network" object:
KHFSn<-as.network(KHFS, directed=F)
```

To save the plot as a tiff file (in the default folder) use the `tiff()` function, which opens an image file, and finish with `dev.off()`, which closes the image file for changes.

```
# 4. Create and save visualization
par(mar=c(0,0,0,0)) # Margin deletion
gplot(KHFSn,
      gmode="graph",       # undirected network
      mode="random",       # random plotting of nodes
      jitter=F,            # do not allow nodes to be "jittered"
      edge.col="grey70",   # set color of ties
      vertex.col="cornflowerblue",    # set color of nodes
      displaylabels=T,     # indicate that labels should be included
      label.pos=1,         # indicate that labels should be given below points
      label.cex=.7)        # indicate the size of the labels (1 is default)
```



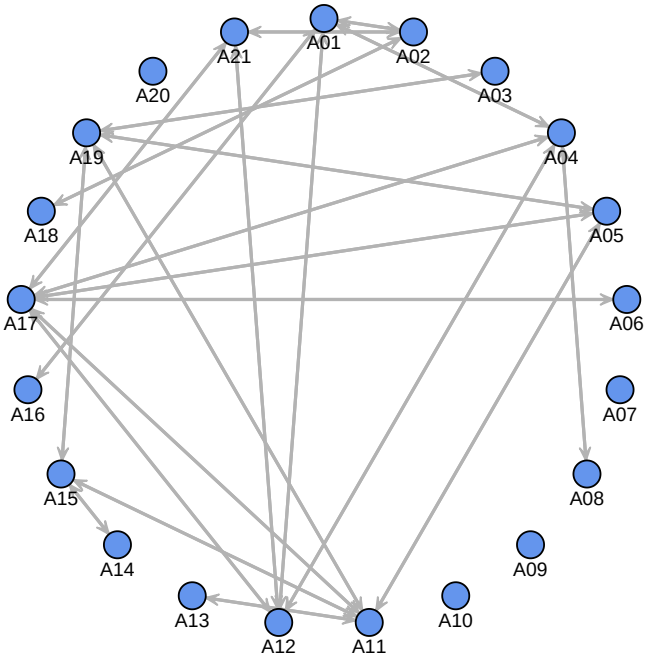
*`gmode="graph"` is used for **undirected** networks, while `gmode="digraph` is used for **directed** networks. If the input graph is undirected but the selected gmode is for directed networks, then arrows on both sides will be visualized. Also, most of the times the margin deletion is not necessary.*

The use of random position may be risky because of links overlapping and too close nodes.

15.2.2 Circle Layout

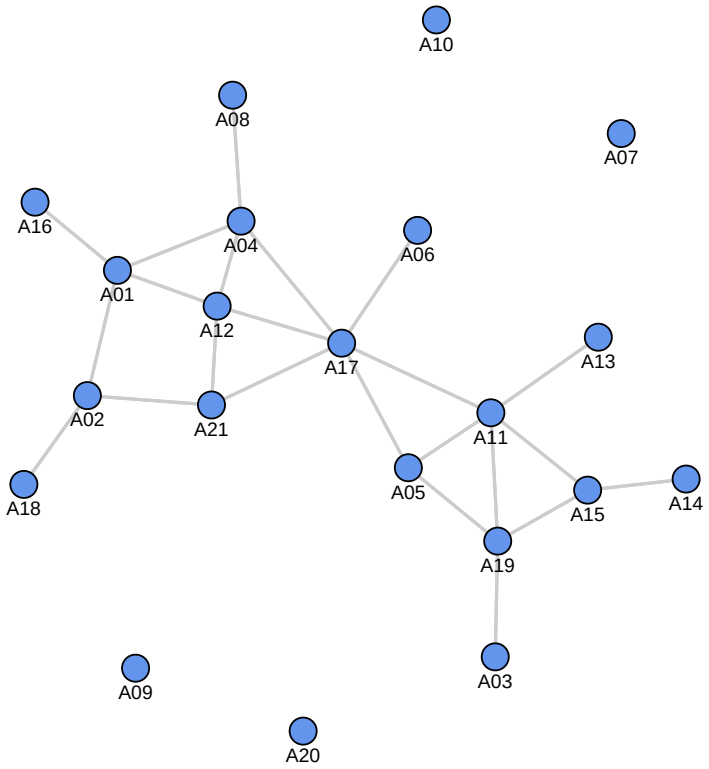
For a circle layout, it is necessary to change `mode` into `"circle"`.

```
par(mar=c(0,0,0,0))
gplot(KHFSn,
      gmode="digraph",
      mode="circle",
      jitter=F,
      edge.col="grey70",
      vertex.col="cornflowerblue",
      displaylabels=T,
      label.pos=1,
      label.cex=.7,
      arrowhead.cex = 0.5)
```



15.2.3 FR Layout

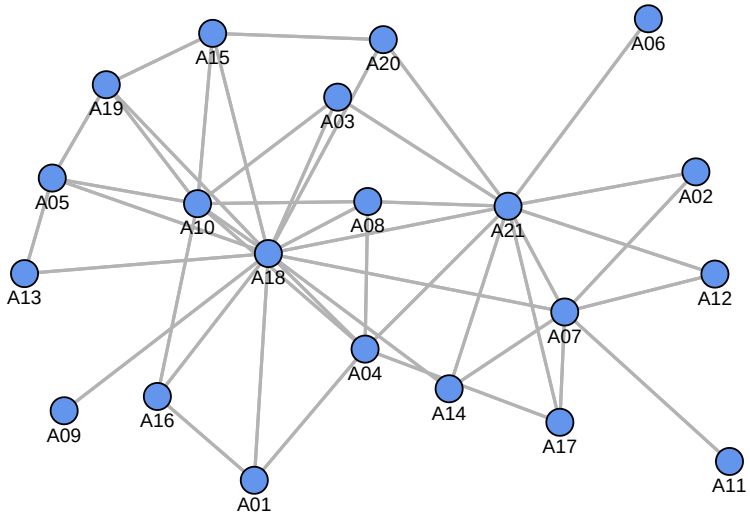
```
par(mar=c(0,0,0,0))
gplot(KHFSn,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=FALSE,
      #ties
      edge.col="grey80",
      #nodes
      vertex.col="cornflowerblue",
      vertex.cex = 1,
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



15.2.4 KHA Network

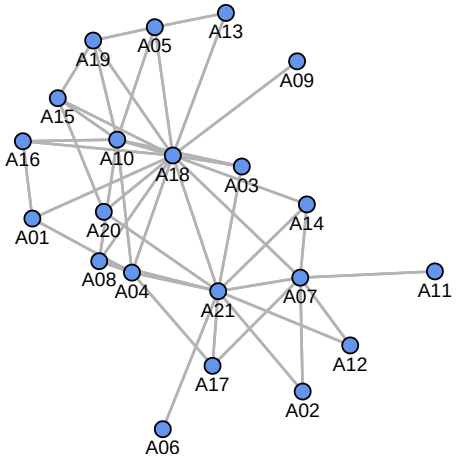
The following network is denser than the previous, since we multiply the matrix for its transpose, indicating

```
# Read in the advice network for Krackhardt's high-tech managers:
KHA<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Advice.csv",
                        stringsAsFactors=FALSE, row.names=1))

# Symmetrize the matrix using the minimum approach, so an advice tie
# is only considered if both ask each other for advice:
KHAS<-KHA*t(KHA)

# Turn this into a "network" object:
KHASn<-as.network(KHAS, directed=T)

par(mar=c(0,0,0,0))
gplot(KHASn,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=F,
      #ties
      edge.col="grey70",
      #nodes
      vertex.col="cornflowerblue",
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



We could also build a function to plot a network:

```
plot_network = function(network){
    gplot(network,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=F,
      #ties
      edge.col="grey70",
      #nodes
      vertex.col="cornflowerblue",
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
}

# Example of use
plot_network(KHASn)
```



15.3 Layout based on 2 continuous attributes

```
# Read data and convert it into matrix
WPJ <- as.matrix(read.csv("datasets/Wolfe_Primates_JointPresence.csv",
                          stringsAsFactors=FALSE, row.names=1))

# Read attributes
WPAttr <- read.csv("datasets/Wolfe_Primates_Attributes.csv",
                   stringsAsFactors=FALSE, row.names=1)

# Rescale attributes so they range from 0 to 1
WP.AGE <- (WPAttr$AGE-min(WPAttr$AGE))/(max(WPAttr$AGE)-min(WPAttr$AGE))
WP.RANK <- (max(WPAttr$RANK)-WPAttr$RANK)/(max(WPAttr$RANK)-min(WPAttr$RANK))

# Turn the 2 continuous variables into coordinates
WPCOORD <- matrix(c(WP.AGE,WP.RANK),
                  dim(WPJ)[1],2)

# Consider drawing only "relatively strong" ties
WPJ6W <- (WPJ>5)*(WPJ-5)
# Only consider a tie in tie matrix, but only
# if major than 5 and then rescale it by removing 5
```

In the next step, we're going to make darker ties for those links where primates interacted more with each other (from grey40 to grey90).

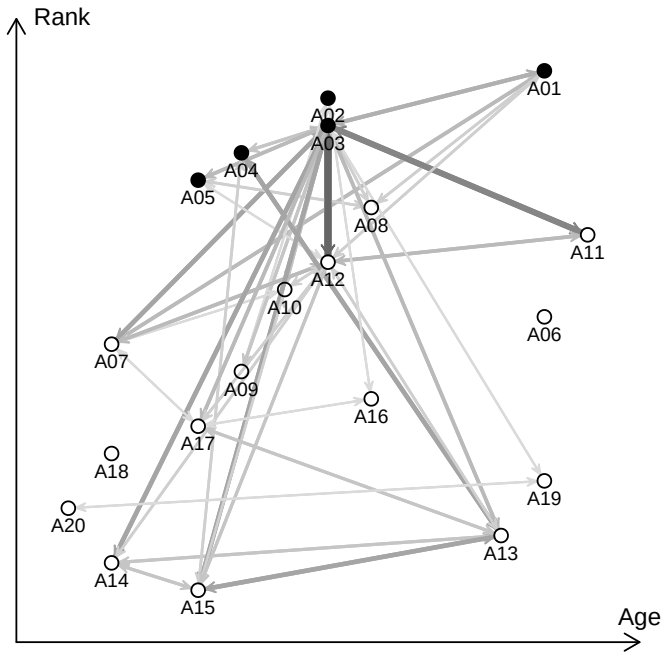
```
# Create colors (greyscales) for ties
WPJ6W.Color<-WPJ6W
col.grey<-colorRampPalette(c("grey90", "grey40"))
col.greyN<-col.grey(max(WPJ6W)+1)
```

```
for (k in (1:max(WPJ6W)))
{
    WPJ6W.Color[WPJ6W==k]<-col.greyN[k+1]
}
```

Now let's start drawing, remembering that:

- coordinates of nodes are given by their rank and their age (attributes): `coord=WPCOORD`;
- the edge color is given by the mapping color based on their number of links: `edge.col = WPJ6W.Color`;
- the color of the vertex indicates the gender of the primate (white for females, black for males): `vertex.col = (WPAttr$GENDER=="male")*9`;
- the added arrows allows us to visualize the cartesian plan.

```
par(mar=c(0,0,0,0))
gplot(WPJ6W*1.1,
      gmode="digraph",
      #layout
      coord=WPCOORD, # using the coordinates to place primates
      jitter=F,
      #ties
      edge.col=WPJ6W.Color,
      edge.lwd=.3,
      #nodes
      vertex.col=(WPAttr$GENDER=="male")*9,
      vertex.cex=0.5,
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7,
      arrowhead.cex = 0.4)

# Add arrows
arrows(-.1, -.1, 1.1, -.1, length = 0.1)
arrows(-.1, -.1, -.1, 1.1, length = 0.1)
text(-.1, 1.1, labels="Rank",
     cex=0.8, pos=4)
text(1.1,-.1, labels="Age",
     cex=0.8, pos=3)
```



15.4 Layout where points are grouped based on nominal attribute

First of all, we're going to import the datasets and convert them into networks as usual:

```
# 1. Convert datasets into networks
## Convert the KHA dataset into a network
KHA<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Advice.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHAS<-KHA*t(KHA)
KHASn<-as.network(KHAS, directed=F)

## Convert the KHF dataset into a network
KHF<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Friendship.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHFS<-KHF*t(KHF)
KHFSn<-as.network(KHFS, directed=F)
```

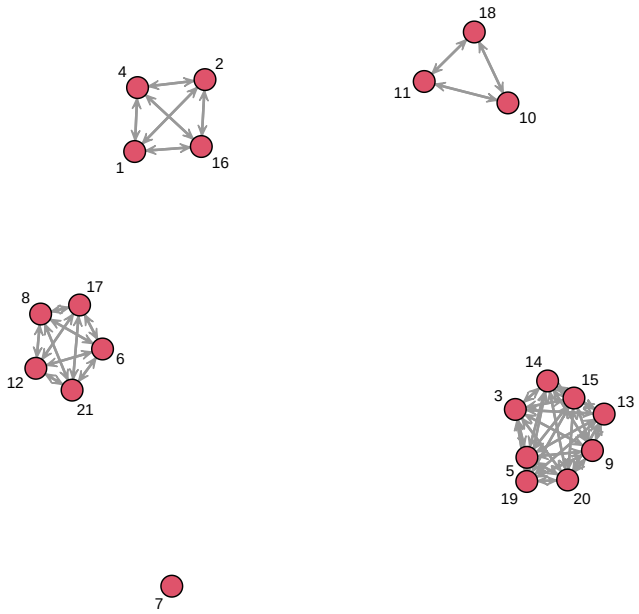
Then, we can create a network with ties within departments to define positions. By importing the KHA dataset and maintaining its attribute, we will maintain the departments and convert them to matrix. In the end, we maintain only those links where two subjects belong to the same department (`DEPS==t(DEPS)`).

```
# 2. Create a network with ties within department to define positions
KHAttr<-read.csv("datasets/Krackhardt_HighTech_Attributes.csv",
                 stringsAsFactors=FALSE, row.names=1)
DEPS<-matrix(KHAttr$DEPT,nrow(KHA),nrow(KHA))
DEPSIM<-DEPS==t(DEPS)
```

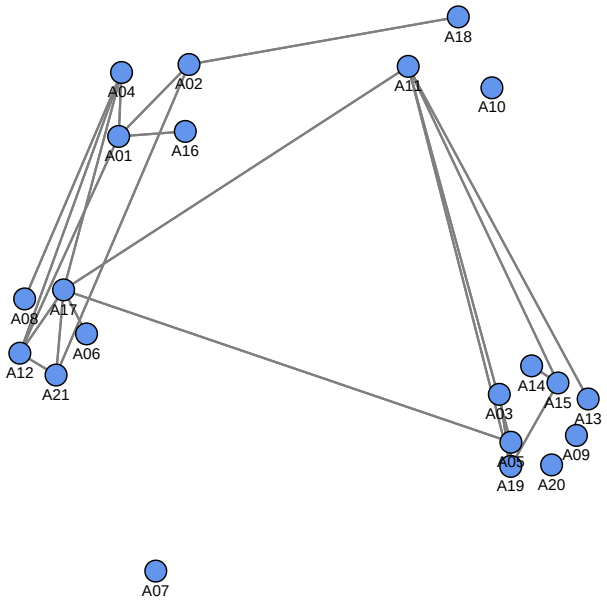
We can get nodes coordinates by plotting them based on their department similarity. Coords may change from an execution to another, but groupings will be the same. In this way, we will place

similar nodes next to each other.

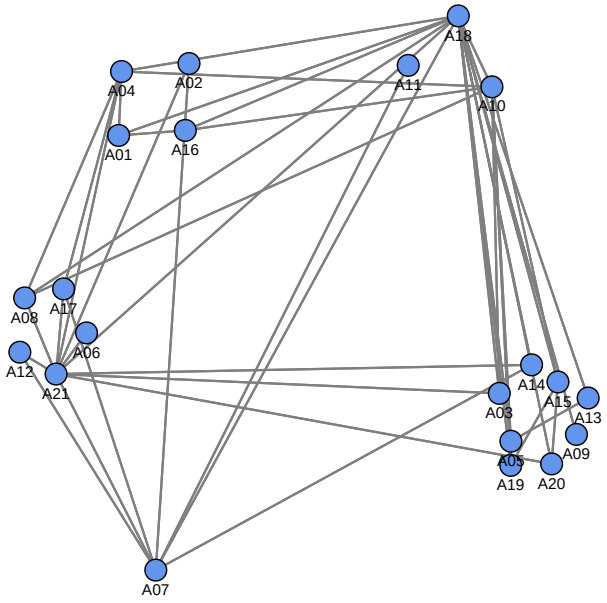
```
# 3. Get coordinates
par(mar = c(0,0,0,0))
DEP_COORD<-gplot(DEPSIM,
                 displaylabels=T,
                 vertex.cex = 1,
                 label.cex = 0.7,
                 arrowhead.cex = 0.5,
                 edge.col = "grey60")
```




```
# 4. Draw the friendship network using
#    the coordinates got from nominal friendship
par(mar=c(0,0,0,0))
gplot(KHFSn,
      gmode="graph",
      #layout
      coord=DEP_COORD,
      jitter=F,
      #ties
      edge.col="grey50",
      edge.lwd=.7,
      #nodes
      vertex.col="cornflowerblue",
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```




```
par(mar=c(0,0,0,0))
gplot(KHASn,
      gmode="graph",
      #layout
      coord=DEP_COORD,
      jitter=F,
      #ties
      edge.col="grey50",
      edge.lwd=.7,
      #nodes
      vertex.col="cornflowerblue",
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```

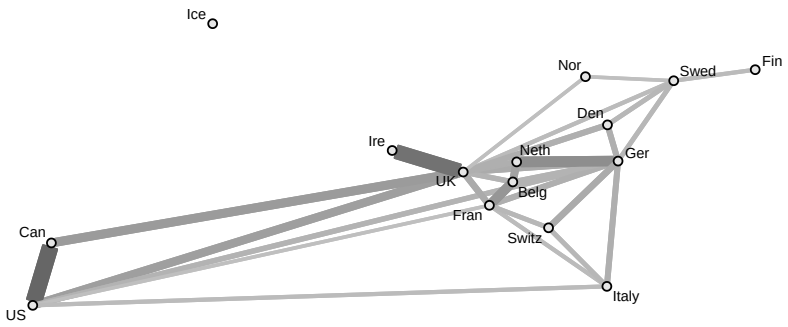


15.5 Geography - International trade among countries in 1928

In this case, nodes are positioned based on the geographic location of their capitals. Canada and the US have been slightly moved to the West to make the trade between European countries more visible.

```
# 1. Dataset conversion into network
STE<-as.matrix(read.csv("datasets/Savage_TransactionFlows_ExportsPerc.csv",
                        stringsAsFactors=FALSE, row.names=1))
STAttr<-read.csv("datasets/Savage_TransactionFlows_Attributes.csv",
                 stringsAsFactors=FALSE, row.names=1)

# 2. Consider Latitude and Longitude
LA<-STAttr$Latitude
LO<-STAttr$Longitude

# 3. Move Canada and US to the west
LO[2:3]<-LO[2:3]+40

# 4. Symmetrized using product
STES<-(STE*t(STE))
##  To plot only stronger trades (more than 19 trades)
STESGT20<-round(((STES>19)*(STES-19))^.5)

# 5. Create colors (greyscale) for ties
STESGT20.Color<-STESGT20
col.grey<-colorRampPalette(c("grey80", "grey40"))
col.greyN<-col.grey(max(STESGT20)+1)
for (k in (1:max(STESGT20)))
{
    STESGT20.Color[STESGT20==k]<-col.greyN[k+1]
}
```

```
# 6. Visualizing the map
par(mar=c(0,0,0,0))
gplot(STESGT20,
      gmode="graph",
      coord=cbind(LO,LA), # Using geographical coordinates to map cities
      jitter=F,
      edge.col=STESGT20.Color,
      edge.lwd=0.4,
      vertex.col="grey90",
      vertex.cex=0.3,
      displaylabels=T,
      label.cex=.5)
```



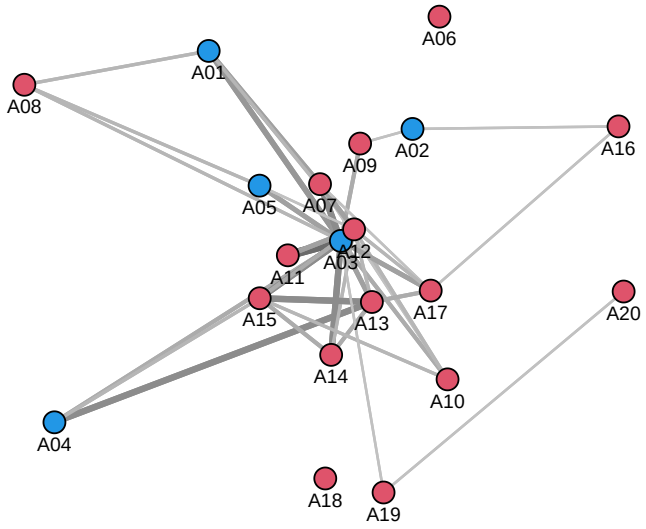
15.6 Multidimensional scaling

15.6.1 MDS on Primates

We can apply the MDS in order to better plot nodes inside the network, by using the default function in R `cmdscale()`.

```
# 1. Rescale values using MDS
WPJS<-log(WPJ+1)
WPJ_CoordMDS<-cmdscale(max(WPJS)*1.001-WPJS, eig=TRUE, k = 2)

# 2. Consider drawing only "relatively strong" ties
WPJ6W<-(WPJ>5)*(WPJ-5)

# 3. Create color for ties
WPJ6W.Color<-WPJ6W
col.grey<-colorRampPalette(c("grey80", "grey30"))
col.greyN<-col.grey(max(WPJ6W)+1)
for (k in (1:max(WPJ6W)))
{
    WPJ6W.Color[WPJ6W==k]<-col.greyN[k+1]
}

# 4. Draw
par(mar=c(0,0,0,0))
gplot(WPJ6W*1.1,
      gmode="graph",
      #layout
      coord=WPJ_CoordMDS$points,
      jitter=F,
```

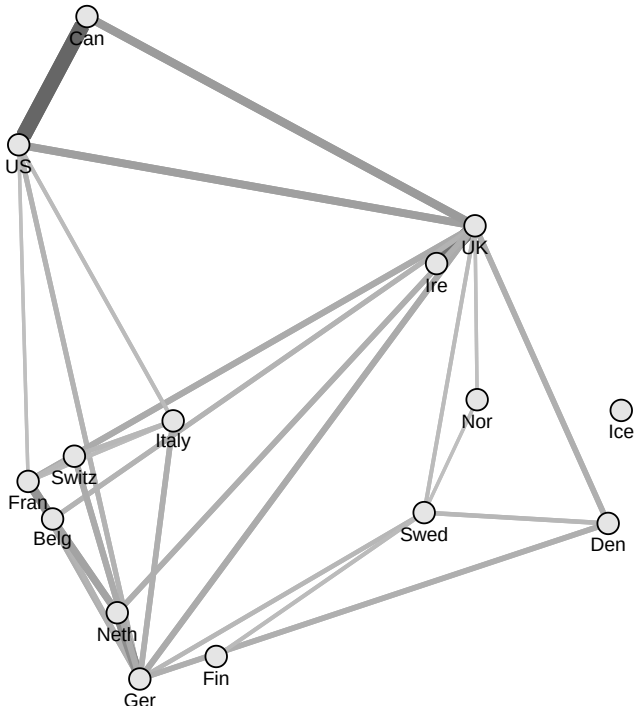
```
      edge.col=WPJ6W.Color,
      edge.lwd=0.5,
      #nodes
      vertex.col=ifelse(WPAttr$GENDER=="male",4,2),
      vertex.cex=0.8,
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



15.6.2 MDS on valued trade

```
STE<-as.matrix(read.csv("datasets/Savage_TransactionFlows_ExportsPerc.csv",
                        stringsAsFactors=FALSE, row.names=1))
# Symmetrized using product
STES<-(STE*t(STE))
# To plot only stronger trade
STESGT20<-round(((STES>19)*(STES-19))^.5)

# Create colors (greyscale) for ties
STESGT20.Color<-STESGT20
col.grey<-colorRampPalette(c("grey80", "grey40"))
col.greyN<-col.grey(max(STESGT20)+1)
for (k in (1:max(STESGT20)))
{
    STESGT20.Color[STESGT20==k]<-col.greyN[k+1]
}

STESGT20_Coord<-cmdscale((max(STE)*1.002-STE)^2, eig=TRUE, k = 2)
#STESGT20_Coord<-cmdscale(STE-STE, eig=TRUE, k = 2)
```

```
par(mar=c(1,1,1,1))
gplot(STESGT20,
      gmode="graph",
      #layout
      coord=STESGT20_Coord$points,
      jitter=F,
      #ties
      edge.col=STESGT20.Color,
      edge.lwd=0.4,
      #nodes
      vertex.col="grey90",
      vertex.cex=0.8,
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



15.6.3 MDS on geodesic distances

```
KHA<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Advice.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHAS<-KHA*t(KHA)
KHASn<-as.network(KHAS, directed=F)
GeoD<-geodist(KHASn)
max(GeoD$gdist)
```

```
## [1] 3
```

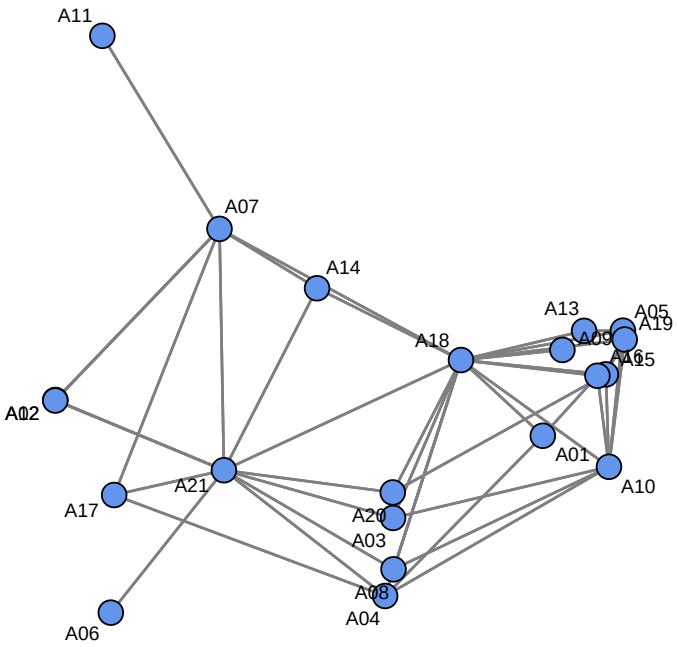
```
#GeoD$gdist[GeoD$counts==0]<--999
#GeoD$gdist[GeoD$gdist==-999]<-max(GeoD$gdist)
```

```r
GeoD_Coord<-cmdscale(GeoD$gdist, eig=TRUE, k=2)
GeoD_Coord
```

```
## $points
##               [,1]        [,2]
##  [1,]  0.76162888 -0.15192804
##  [2,] -1.51428017  0.01452473
##  [3,]  0.06243284 -0.53328907
##  [4,]  0.02556773 -0.89875609
##  [5,]  1.13565391  0.34041487
##  [6,] -1.25508843 -0.97588178
##  [7,] -0.74784913  0.81617695
##  [8,]  0.06471364 -0.77547617
##  [9,]  0.85297735  0.24843971
## [10,]  1.07052904 -0.29602193
## [11,] -1.29460998  1.71553243
## [12,] -1.51428017  0.01452473
## [13,]  0.95428481  0.34017104
## [14,] -0.29225683  0.53688623
## [15,]  1.05597933  0.13644745
## [16,]  1.01641735  0.12773099
## [17,] -1.23959938 -0.42884357
## [18,]  0.37975961  0.20110852
## [19,]  1.14398272  0.29717063
## [20,]  0.06113491 -0.41675189
## [21,] -0.72709801 -0.31217975
##
## $eig
##  [1]  1.881496e+01  7.566499e+00  5.471410e+00  3.744726e+00  3.436953e+00
##  [6]  2.613479e+00  2.377477e+00  2.000000e+00  2.000000e+00  2.000000e+00
## [11]  1.109910e+00  7.348856e-01  5.306488e-01  1.185842e-01  2.664535e-15
## [16] -1.212696e-01 -2.706349e-01 -6.477894e-01 -1.307601e+00 -2.646835e+00
## [21] -3.715879e+00
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.4308616 0.5023171
```

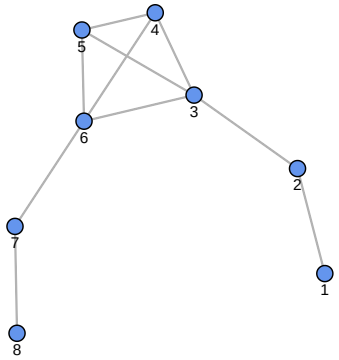
```r
par(mar=c(1,1,1,1))
gplot(KHASn,
      gmode="graph",
      #layout
      coord=GeoD_Coord$points,
      jitter=T,
      #ties
      edge.col="grey50",
      edge.lwd=.7,
      #nodes
      vertex.col="cornflowerblue",
      #labels
      displaylabels=T,
```

```
      label.cex=.7)
```

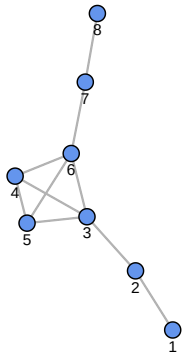


15.7 KamadaKawai and FR layout KHFS

```
mat8<-matrix(c(0),8,8)
mat8[1,2]<-1
mat8[2,3]<-1
mat8[3,4:6]<-1
mat8[4,5:6]<-1
mat8[5,6]<-1
mat8[6,7]<-1
mat8[7,8]<-1
mat8s<-(mat8+t(mat8))>0

gplot(mat8,
      gmode="graph",
      jitter=FALSE,
      edge.col="grey70",
      vertex.col="cornflowerblue",
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```




```
gplot(mat8,
      gmode="graph",
      mode="kamadakawai",
      jitter=FALSE,
      edge.col="grey70",
      vertex.col="cornflowerblue",
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```

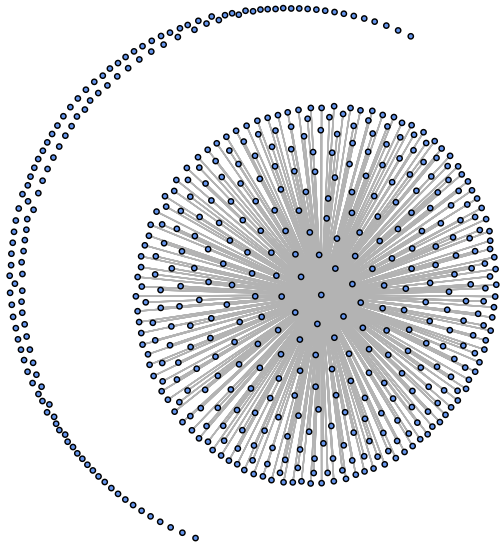


15.7.2 Scientists collaborations

In the next chunk, the aim is to separated into two clusters those scientists that have more than 3 years of experience in their studies from those who do not. As can be seen from the plot, two groups are obtained: one with no edges and the other where every node is connected with the

center.

```
BS504Cx<-as.matrix(read.csv("datasets/Borgatti_Scientists504_Attributes.csv",
                            stringsAsFactors=FALSE, row.names=1))
BS504C<-(BS504Cx>3)

par(mar=c(0,0,0,0))
gplot(BS504C,
      gmode="graph",
      #layout
      mode="kamadakawai",
      jitter=F,
      #ties
      edge.col="grey70",
      edge.lwd=.1,
      #nodes
      vertex.col="cornflowerblue",
      vertex.cex=.75)
```

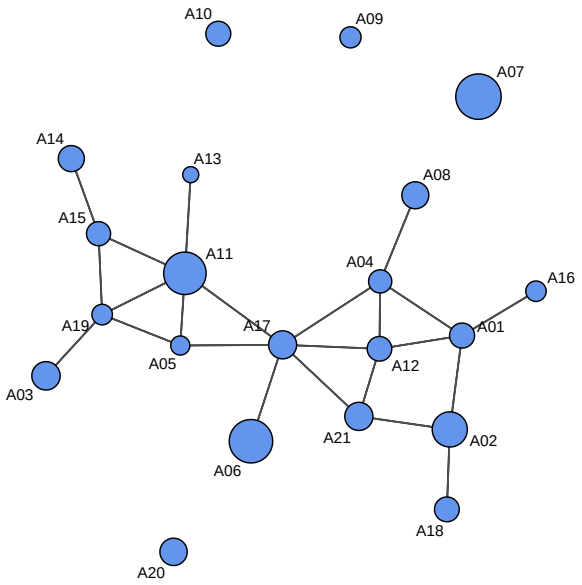


15.7.3 Nodal attributes on KHF

Let's consider scientists' tenure. If we aim to plot the variation in terms of tenure among scientists, size can be a choice:

```
KHF<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Friendship.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHFS<-KHF*t(KHF)
KHAttr<-read.csv("datasets/Krackhardt_HighTech_Attributes.csv",
                 stringsAsFactors=FALSE, row.names=1)

par(mar=c(1,1,1,1))
gplot(KHFS,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=F,
      #ties
```

```
      edge.col="grey30",
      edge.lwd=.4,
      #nodes
      vertex.col="cornflowerblue",
      vertex.cex=KHAttr$TENURE/20+.8,
      #labels
      displaylabels=TRUE,
      label.cex=.7)
```

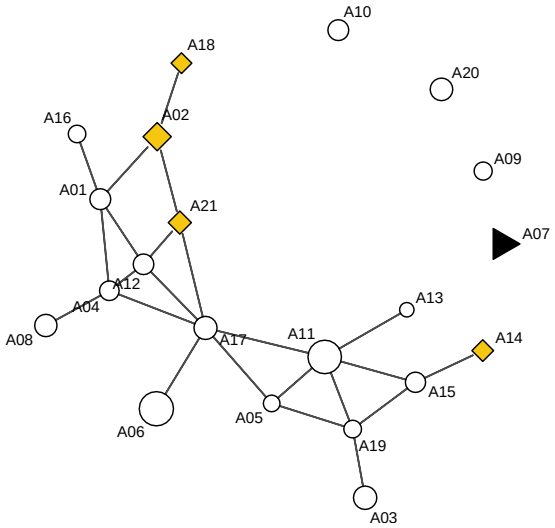


Now, the same plot is proposed again with:

- level as shape and colour;
- tenure as size.

```
# type sides of nodes
KHF<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Friendship.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHFS<-KHF*t(KHF)
KHAttr<-read.csv("datasets/Krackhardt_HighTech_Attributes.csv",
                 stringsAsFactors=FALSE, row.names=1)

par(mar=c(2,2,2,2))
gplot(KHFS,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=F,
      #ties
      edge.col="grey30",
      edge.lwd=.4,
      #nodes
      vertex.col=((KHAttr$LEVEL==1)*1+
                  (KHAttr$LEVEL==2)*7+
                  (KHAttr$LEVEL==3)*0),
      # Define symbols
      vertex.sides=((KHAttr$LEVEL==1)*3+
```

```
                    (KHAttr$LEVEL==2)*4+
                    (KHAttr$LEVEL==3)*20),
     vertex.cex=.8+KHAttr$TENURE/25,
     #labels
     displaylabels=TRUE,
     label.cex=.7)
```

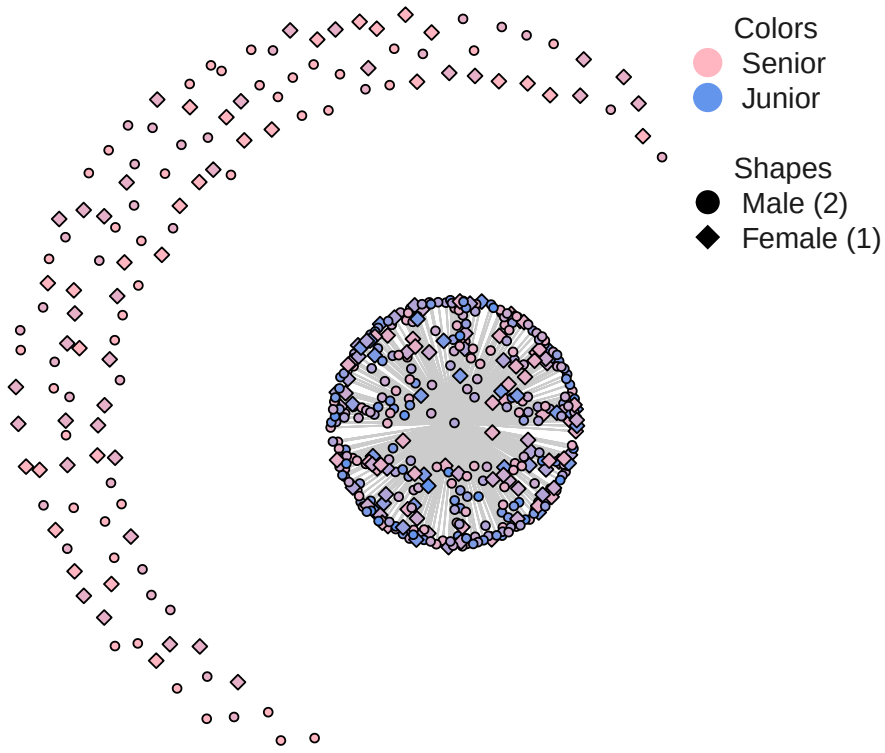


15.7.4 Scientists collaboration (more complex)

```
# Gender and tenure with legend
BS504Cx<-as.matrix(read.csv("datasets/Borgatti_Scientists504_Attributes.csv",
                            stringsAsFactors=FALSE, row.names=1))
BS504C<-(BS504Cx>3)

BS504Attr<-read.csv("datasets/Borgatti_Scientists504_Attributes.csv",
                    stringsAsFactors=FALSE, row.names=1)
```

```
# Get the years
Tenure<-(BS504Attr$Years)

# Normalize the tenure by the maximum and consider the integer number of years
TenureA<-(round(10*(Tenure)/max((Tenure))))

# Create a color palette
colfunc <- colorRampPalette(c("lightpink", "cornflowerblue"))
COLWB<-colfunc(7)

# Assign colors to different tenures
TenureA2<-TenureA
TenureA2[TenureA==0]<-COLWB[1]
TenureA2[TenureA==1]<-COLWB[2]
TenureA2[TenureA==2]<-COLWB[3]
TenureA2[TenureA==3]<-COLWB[4]
TenureA2[TenureA==4]<-COLWB[5]
TenureA2[TenureA %in% c(5,6)]<-COLWB[6]
```

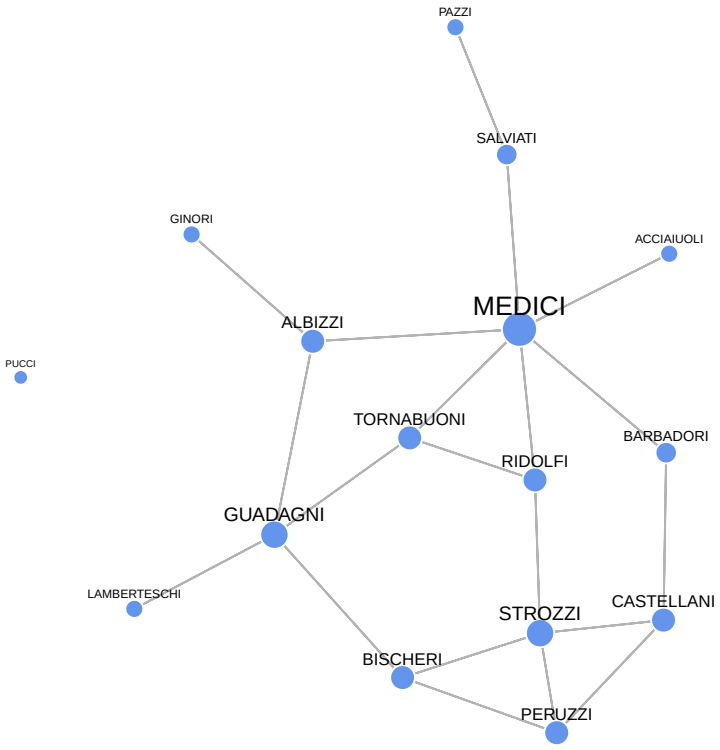
```
TenureA2[TenureA %in% c(7,8,9,10)]<-COLWB[7]

# Plot
par(mar=c(0,0,0,0))
gplot(BS504C,
      gmode="graph",
      jitter = F,
      edge.col="grey80",
      edge.lwd=.2,
      vertex.col=TenureA2,
      vertex.cex=1+.7*(2-BS504Attr$Sex),
      vertex.sides=(BS504Attr$Sex-1)*47+4)

legend("topright",
       legend = c("Colors"," Senior", " Junior",
                  " ","Shapes", " Male (2)", " Female (1)"),
       col = c("white","lightpink","cornflowerblue",
               "white","white","black","black"),
       bty = "n", pch =c(19,19,19,19,19,19,18),
       pt.cex = c(0,2.1,2.1,0,0,1.7,1.9),
       cex = 1,
       text.col = "grey10",
       horiz = F ,
       inset = c(0.01))
```



15.7.5 Nodal attributes on PFM

```
PFM<-as.matrix(read.csv("datasets/Padgett_FlorentineFamilies_Marriage.csv",
                        stringsAsFactors=FALSE, row.names=1))
```

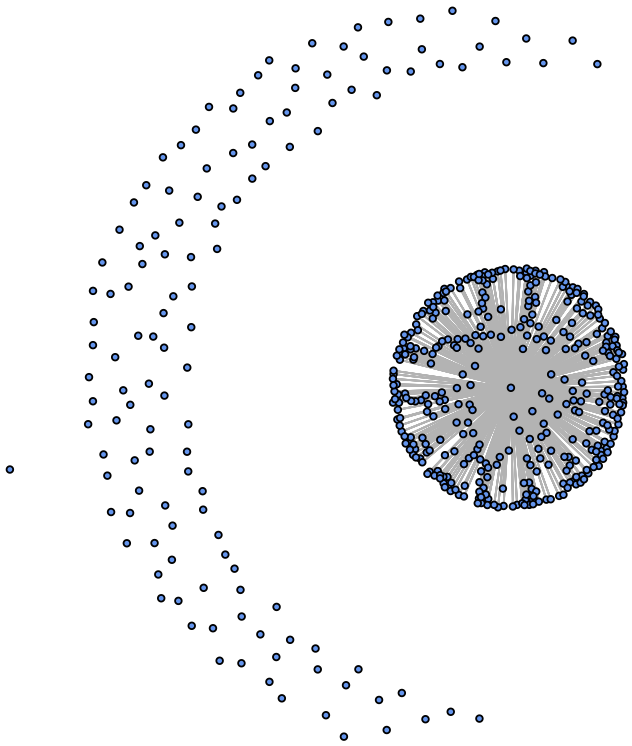
```
PFMn<-as.network(PFM, directed=F)

# Compute the degree of the network
DPFM<-sna::degree(PFMn)

# Plot the network, where the size relies on the degree of every node
par(mar=c(0,0,0,0))
gplot(PFMn,
      gmode="graph",
      jitter=F,
      edge.col="grey70",
      edge.lwd=.1,
      vertex.col="cornflowerblue",
      vertex.cex=DPFM/17+.5,
      displaylabels=TRUE,
      label.pos=3,
      label.cex=DPFM/20+.3,
      vertex.border="white")
```



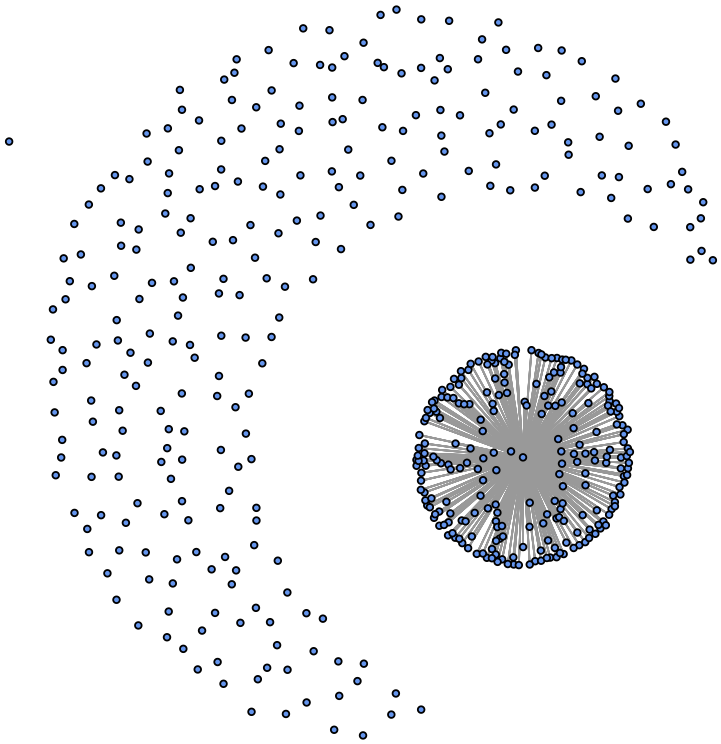
15.8 Tie strength on Scientists collaborations

```
BS504C<-as.matrix(read.csv("datasets/Borgatti_Scientists504_Attributes.csv",
                           stringsAsFactors=FALSE, row.names=1))

# Filtering different years of experience
BS504CGT3<-(BS504Cx>3)
BS504CGT5<-(BS504Cx>5)
BS504CGT7<-(BS504Cx>7)
BS504CGT9<-(BS504Cx>9)
```

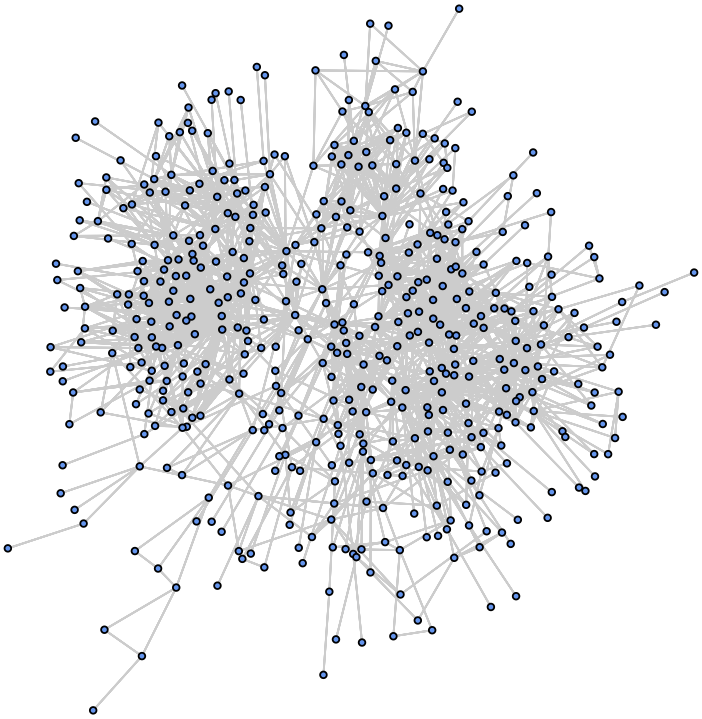
```
# 3 years of experience
par(mar=c(0,0,0,0))
gplot(BS504CGT3,
      gmode="graph",
      jitter=F,
      edge.col="grey70",
      edge.lwd=.1,
      vertex.col="cornflowerblue",
      vertex.cex=.75)
```



```
# 9 years of experience
par(mar=c(0,0,0,0))
gplot(BS504CGT9,
      gmode="graph",
      jitter=F,
      edge.col="grey60",
      edge.lwd=.1,
      vertex.col="cornflowerblue",
      vertex.cex=.75)
```

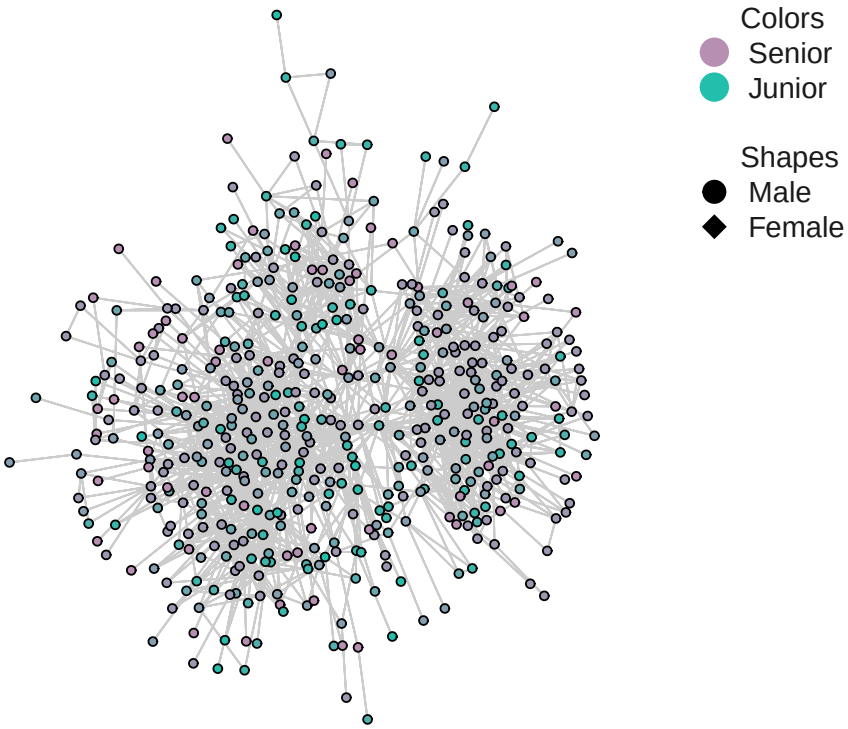


```
library(sna)

# SCIENTISTS COLLABORATION
BS504Cx<-as.matrix(read.csv("datasets/Borgatti_Scientists504_Collaboration.csv",
                            stringsAsFactors=FALSE, row.names=1))

# Pick only those who have at least 3 collabs
BS504C<-(BS504Cx>3)

# Plotting the collaborations
par(mar=c(0,0,0,0))
gplot(BS504C,
      gmode="graph",
      mode="kamadakawai",
      jitter=F,
      edge.col="grey80",
      edge.lwd=.1,
      vertex.col="cornflowerblue",
      vertex.cex=.75)
```



```
# Inserting attributes
BS504Attr<-read.csv("datasets/Borgatti_Scientists504_Attributes.csv",
                    stringsAsFactors=FALSE, row.names=1)

Tenure<-(BS504Attr$Years) # Years attribute
TenureA<-(round(10*(Tenure)/max((Tenure)))) # Normalizing years
colfunc <- colorRampPalette(c("#B78FB3", "#22BFAC")) # color palette

COLWB<-colfunc(7) # 7 colors
vertex_color = ifelse(TenureA %in% c(5,6),COLWB[6],
                      ifelse(TenureA>=7, COLWB[7],COLWB[TenureA+1]))

par(mar=c(0,0,0,0))
gplot(BS504C,
      gmode="graph",
      mode="kamadakawai",
      jitter = F,
      edge.col="grey80",
      edge.lwd=.2,
      vertex.col=vertex_color, # color
      vertex.cex=1,#1+.7*(2-BS504Attr$Sex),
      vertex.sides=BS504Attr$Sex-1*47+3)

legend("topright",
       legend = c("Colors"," Senior", " Junior", " ","Shapes", " Male", " Female"),
       col = c("white","#B78FB3","#22BFAC","white","white","black","black"),
       bty = "n",
       pch =c(19,19,19,19,19,19,18),
       pt.cex = c(0,2.1,2.1,0,0,1.7,1.9),
       cex = 1,
       text.col = "grey10",
```

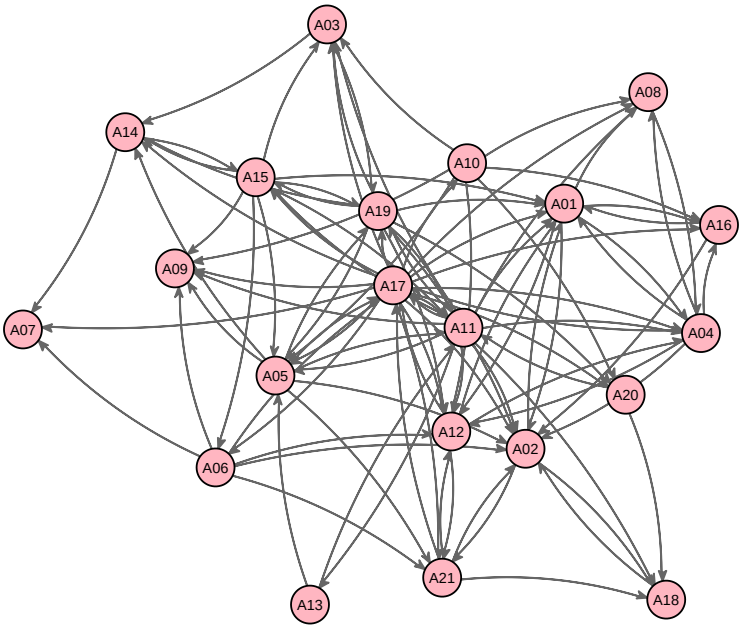
```
       horiz = F ,
       inset = c(0.01))
```



15.10 Directed networks

```
KHF<-as.matrix(read.csv("datasets/Krackhardt_HighTech_Friendship.csv",
                        stringsAsFactors=FALSE, row.names=1))
KHFn<-as.network(KHF, directed=T)
```

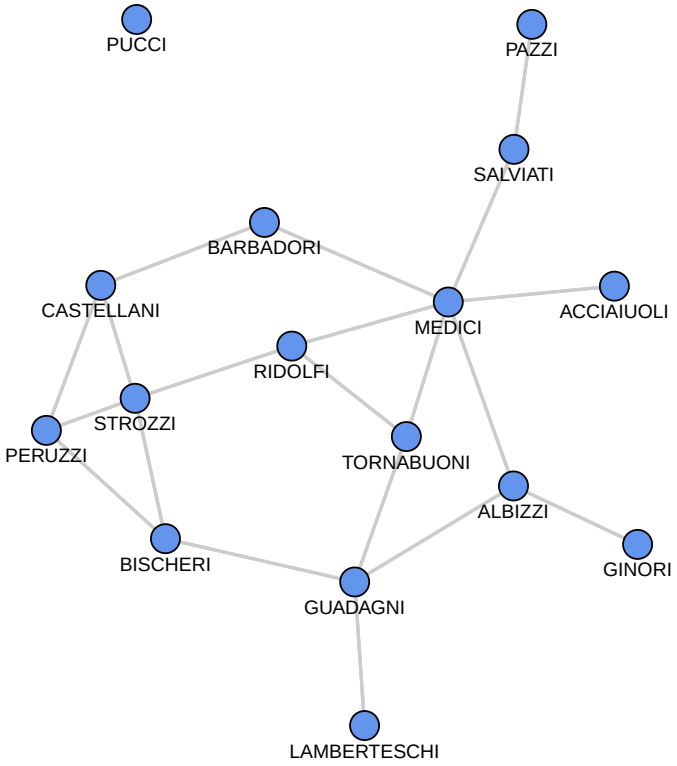
```
# Plotting points based on their coordinates
par(mar=c(0,0,0,0))
gplot(KHFn,
      gmode="digraph",
      coord=COOKHF,
      jitter=F,
      edge.col="grey40",
      edge.lwd=0.2,
      usecurve=T,
      edge.curve=.04,
      arrowhead.cex=.4,
      vertex.col="lightpink",
      vertex.cex=1.4,
      displaylabels=T,
      label.pos=5,
      label.cex=.5)
```


A03
A08
A14
A10
A15
A01
A19
A16
A09
A17
A07
A11
A04
A05
A20
A12
A02
A06
A21
A18
A13

Chapter 16

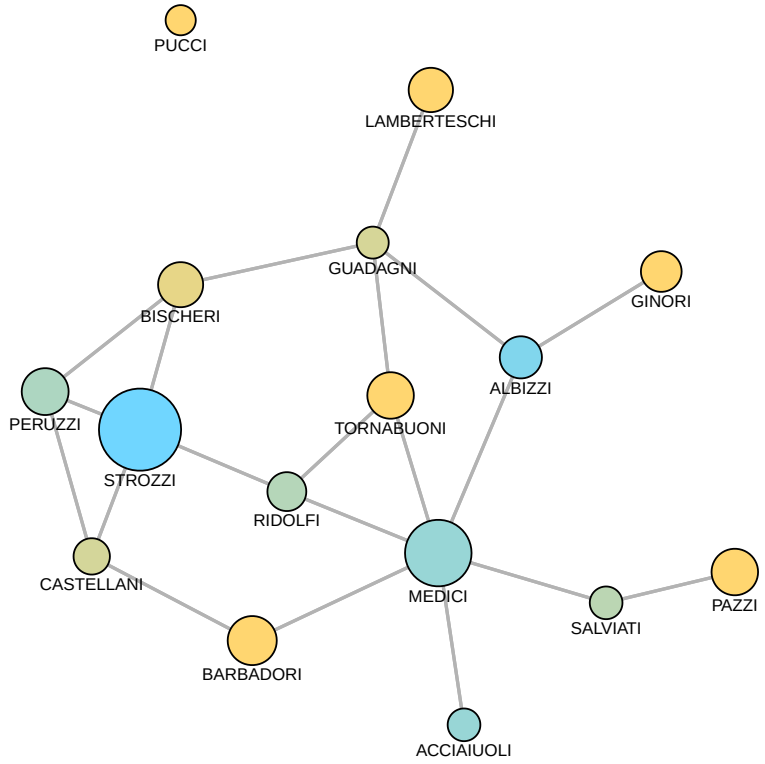
Centrality Measures

```
library(sna)
# FLORENTINE FAMILIES
PFM<-as.matrix(read.csv("datasets/Padgett_FlorentineFamilies_Marriage.csv",
                        stringsAsFactors=FALSE, row.names=1))
PFMn<-as.network(PFM, directed=F)
par(mar=c(0,0,0,0))
gplot(PFMn,
      gmode="graph",
      jitter=F,
      edge.col="grey80",
      vertex.cex = 1,
      vertex.col="cornflowerblue",
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



The following plot considers the family's wealth, represented both by size and colour variation.

```
# Add attributes file and use wealth to change size of nodes (vertex.cex)
PFA<-read.csv("datasets/Padgett_FlorentineFfamilies_Attributes.csv",
               stringsAsFactors=FALSE, row.names=1)

# Color palette
n_colors<-max(PFA$Priors)-min(PFA$Priors)+1
colfunc <- colorRampPalette(c("#FFD670", "#70D6FF"))
colors<-colfunc(n_colors)

par(mar=c(0,0,0,0))
gplot(PFMn,
      gmode="graph",
      jitter=F,
      edge.col="grey70",
      vertex.col=colors[PFA$Priors+1],
      vertex.cex=PFA$Wealth/80+1,
      displaylabels=T,
      mar = c(0,0,0,0),
      label.pos=1,
      label.cex=.6)
```



First, we will compute some centrality measures, such as degree, betweenness, Bonacich beta centrality and the constraint index.

```
# Calculate centrality measures
## Degree
sna::degree(PFMn, gmode="graph")
```

```
##  [1] 1 3 2 3 3 1 4 1 6 1 3 0 3 2 4 3
```

```
## Betweenness centrality
sna::betweenness(PFMn, gmode="graph", cmode="undirected")
```

```
##  [1]  0.000000 19.333333  8.500000  9.500000  5.000000  0.000000 23.166667
##  [8]  0.000000 47.500000  0.000000  2.000000  0.000000 10.333333 13.000000
## [15]  9.333333  8.333333
```

```
## Bonacich Power Centrality
sna::bonpow(PFMn, gmode="graph", exponent=0)
```

```
##   ACCIAIUOLI      ALBIZZI    BARBADORI     BISCHERI   CASTELLANI       GINORI
##    0.3455474    1.0366421    0.6910947    1.0366421    1.0366421    0.3455474
##     GUADAGNI LAMBERTESCHI       MEDICI        PAZZI      PERUZZI        PUCCI
##    1.3821895    0.3455474    2.0732842    0.3455474    1.0366421    0.0000000
##      RIDOLFI     SALVIATI      STROZZI   TORNABUONI
##    1.0366421    0.6910947    1.3821895    1.0366421
```

```
## Constraint index
library(igraph)
PFM_i<-graph_from_adjacency_matrix(as.matrix(PFM),mode="undirected", diag=F)
constraint(PFM_i)
```

```
##   ACCIAIUOLI      ALBIZZI    BARBADORI     BISCHERI   CASTELLANI       GINORI
##    1.0000000    0.3333333    0.5000000    0.4822531    0.4822531    1.0000000
##     GUADAGNI LAMBERTESCHI       MEDICI        PAZZI      PERUZZI        PUCCI
```

```
##    0.2500000    1.0000000    0.2098765    1.0000000    0.6558642          NaN
##      RIDOLFI      SALVIATI       STROZZI   TORNABUONI
##    0.4598765    0.5000000    0.4583333    0.4598765
```

Then, some statistical tests are applied to evaluate the correlation between degree, wealth and the number of priors for each family.

```
# Statistical tests for significance
PFMn.Deg<-sna::degree(PFMn, gmode="graph")

cor.test(PFMn.Deg,PFA$Priors) # Number of families vs Priors
```

```
##
##  Pearson's product-moment correlation
##
## data:  PFMn.Deg and PFA$Priors
## t = 2.4519, df = 14, p-value = 0.02794
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.07193331 0.82079537
## sample estimates:
##       cor
## 0.5480948
```

```
cor.test(PFMn.Deg,PFA$Wealth) # Number or families vs Wealth
```

```
##
##  Pearson's product-moment correlation
##
## data:  PFMn.Deg and PFA$Wealth
## t = 2.5225, df = 14, p-value = 0.02438
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08755017 0.82585806
## sample estimates:
##       cor
## 0.5589956
```

```
cor.test(PFA$Priors,PFA$Wealth) # Priors vs Wealth
```

```
##
##  Pearson's product-moment correlation
##
## data:  PFA$Priors and PFA$Wealth
## t = 1.7401, df = 14, p-value = 0.1038
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09356713  0.75878748
## sample estimates:
##       cor
## 0.4216972
```

Also, linear models can be used, paying attention to the fact that we detain a few data and the model may not be accurated and general.

```
# Linear Models
## One of them
M1<-lm(PFMn.Deg~PFA$Priors)
summary(M1)
```

```
##
## Call:
## lm(formula = PFMn.Deg ~ PFA$Priors)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36244 -0.69127 -0.02179  0.70515  2.63756
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.67341    0.46887   3.569  0.00308 **
## PFA$Priors   0.03187    0.01300   2.452  0.02794 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.303 on 14 degrees of freedom
## Multiple R-squared:  0.3004, Adjusted R-squared:  0.2504
## F-statistic: 6.012 on 1 and 14 DF,  p-value: 0.02794
```

```
## Both of them
M2<-lm(PFMn.Deg~PFA$Priors+PFA$Wealth)
summary(M2)
```

```
##
## Call:
## lm(formula = PFMn.Deg ~ PFA$Priors + PFA$Wealth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5677 -0.9423 -0.1498  0.8269  2.1718
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.234029   0.507067   2.434   0.0301 *
## PFA$Priors  0.022091   0.013414   1.647   0.1235
## PFA$Wealth  0.016282   0.009419   1.729   0.1075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.22 on 13 degrees of freedom
## Multiple R-squared:  0.4312, Adjusted R-squared:  0.3436
## F-statistic: 4.927 on 2 and 13 DF,  p-value: 0.02555
```

Since there's no right parameter to set as exponent, we can iterate over some possible values and choose the one that gives the wanted results: if negative, we repulse other people; if positive, we aim to attract people and see if our connections propagate over the network. If close to 0, we just consider direct links.

```
# Bonacich Centrality
exponents = c(seq(-0.5,0.5,0.1))
exp = c()
bonpow_results = c()
for(e in exponents){
   exp = append(exp, rep(e,16))
   bonpow_results = append(bonpow_results,
                           sna::bonpow(PFA, gmode="graph", exponent=e))
}
```

Table 16.1: Three families Bonacich Beta centrality varying exponent

Exponent	Bonpow	Family
-0.5	3.212	GUADAGNI
0	-0.49	GUADAGNI
0.5	0.68	GUADAGNI
-0.5	-1.191	MEDICI
0	0.492	MEDICI
0.5	0.39	MEDICI
-0.5	0.492	PUCCI
0	0.39	PUCCI
0.5	3.212	PUCCI

Chapter 17

Statistics about network

17.1 Florentine Families

In the following chunk, Florentine Families data is imported first as matrix and then as undirected network inside `PFMn`. Also, `PFA` contains the attributes for each family.

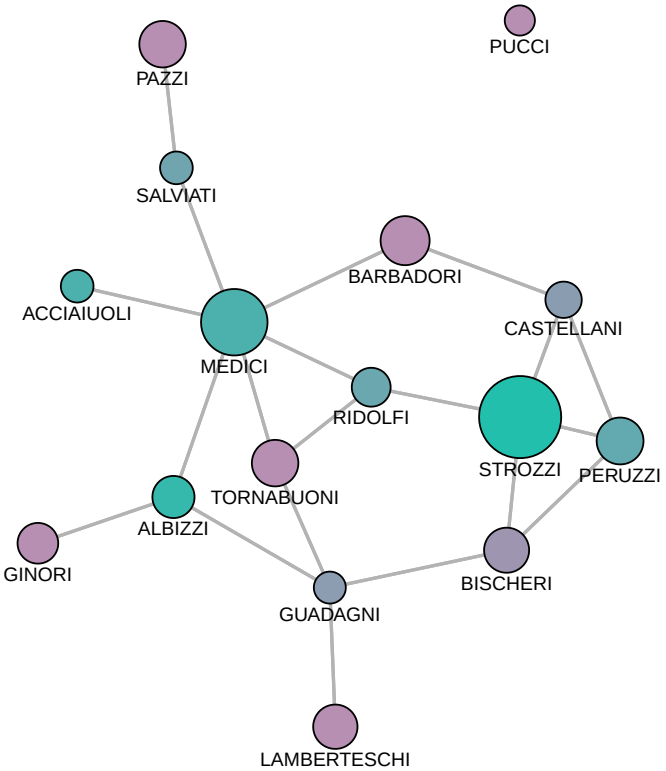
```
# Libraries import
library(sna)
library(tidyverse)

# Data import
PFM<-as.matrix(read.csv("datasets/Padgett_FlorentineFamilies_Marriage.csv",
                        stringsAsFactors=FALSE, row.names=1))
PFA<-read.csv("datasets/Padgett_FlorentineFamilies_Attributes.csv",
              stringsAsFactors=FALSE, row.names=1)

# Converting the matrix into network
PFMn<-as.network(PFM, directed=F)

summary(PFMn, print.adj = FALSE)
```

```
## Network attributes:
##   vertices = 16
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##  total edges = 20
##    missing edges = 0
##    non-missing edges = 20
##  density = 0.1666667
##
## Vertex attributes:
##   vertex.names:
##    character valued attribute
##    16 valid vertex names
##
## No edge attributes
```

Then the families' links are shown in the following network, mapping colours according to the priors (i.e. number of times a family member became part of the Florence councel) and size based

on families' wealth (i.e. the greener, the richer; the purpler, the poorer).

```
# Working on nodes colors...
## Defining the number of colors necessary to map wealth
n_colors<-max(PFA$Priors)-min(PFA$Priors)+1

## Create a function to generate a palette
colfunc <- colorRampPalette(c("#B78FB3", "#22BFAC"))

## Generate range
vertex_color<-colfunc(n_colors)[PFA$Priors-min(PFA$Priors)+1]

par(mar = c(0,0,0,0))
# Plotting the families based on their wealth
gplot(PFMn,
      gmode="graph",
      #layout
      mode="fruchtermanreingold",
      jitter=F,
      #ties
      edge.col="grey70",
      #nodes
      vertex.col=vertex_color,
      vertex.cex=PFA$Wealth/80+1,
      #labels
      displaylabels=T,
      label.pos=1,
      label.cex=.7)
```



Then, the degree for each family is shown, indicating that the most powerful family is the De Medici, followed by Guadagni and Strozzi. Since they are not direct alters, probably they were competing for the power in Florence.

Table 17.1: Florentine Families Degree

Degree	Family
1	ACCIAIUOLI
3	ALBIZZI
2	BARBADORI
3	BISCHERI
3	CASTELLANI
1	GINORI
4	GUADAGNI
1	LAMBERTESCHI
6	MEDICI
1	PAZZI
3	PERUZZI
0	PUCCI
3	RIDOLFI
2	SALVIATI
4	STROZZI
3	TORNABUONI

```
# Degree
PFMn.Deg<-sna::degree(PFMn, gmode="graph")
```

17.2 Classic approach

Let's start with the classic approach, computing the correlation between a family degree and its number of priors.

```
## Correlation test between degree and priors
cor.test(PFMn.Deg,PFA$Priors) # positive correlation
```

```
##
##  Pearson's product-moment correlation
##
## data:  PFMn.Deg and PFA$Priors
## t = 2.4519, df = 14, p-value = 0.02794
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.07193331 0.82079537
## sample estimates:
##       cor
## 0.5480948
```

The p-value is below 0.05, therefore the Pearson correlation is significative and, as the score indicates, positive. Let's try with a linear model where the number of priors is the predictor and the degree is the predicted value:

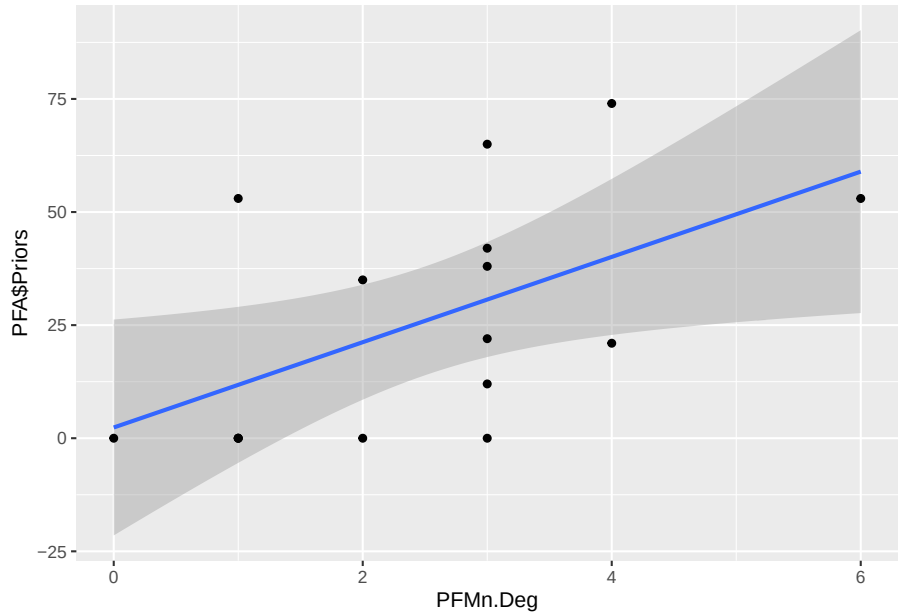
```
## Linear Model
M1<-lm(PFMn.Deg~PFA$Priors)
summary(M1)
```

```
##
## Call:
```

```
## lm(formula = PFMn.Deg ~ PFA$Priors)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36244 -0.69127 -0.02179  0.70515  2.63756
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.67341    0.46887   3.569  0.00308 **
## PFA$Priors   0.03187    0.01300   2.452  0.02794 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.303 on 14 degrees of freedom
## Multiple R-squared:  0.3004, Adjusted R-squared:  0.2504
## F-statistic: 6.012 on 1 and 14 DF,  p-value: 0.02794
```

Priors is a meaningful feature to use to predict the degree, as the p-values suggests, but both the R-squared and the Adjusted R-squared indicate the low quality of the model. By plotting the fitted line over data points we may notice that there is a high variability on the extremes of the degree, while most of the families degree lies on the middle (i.e. around 3).

```
library(ggplot2)

ggplot()+
  geom_smooth(aes(PFMn.Deg,PFA$Priors), method = lm)+
  geom_point(aes(PFMn.Deg,PFA$Priors))
```



17.3 Permutation based approach

Since the permutatio based approach is based on making samples of the original data by permutating them, the function `sample()` comes into help.

```
## Data preparation
PFA.P<-PFA$Priors
sample(PFA.P) # Sample of priors with no replacement
```

```
##  [1]  0 35  0 53 38 65 22 74  0  0  0 12 42 53 21  0
```

First, we build an empty matrix with 1,000 values, since we're going to perform 1,000 permutations. Inside the loop, a sample of priors data is computed, in order to perfom the correlation test on the degree and the permuted priors data.

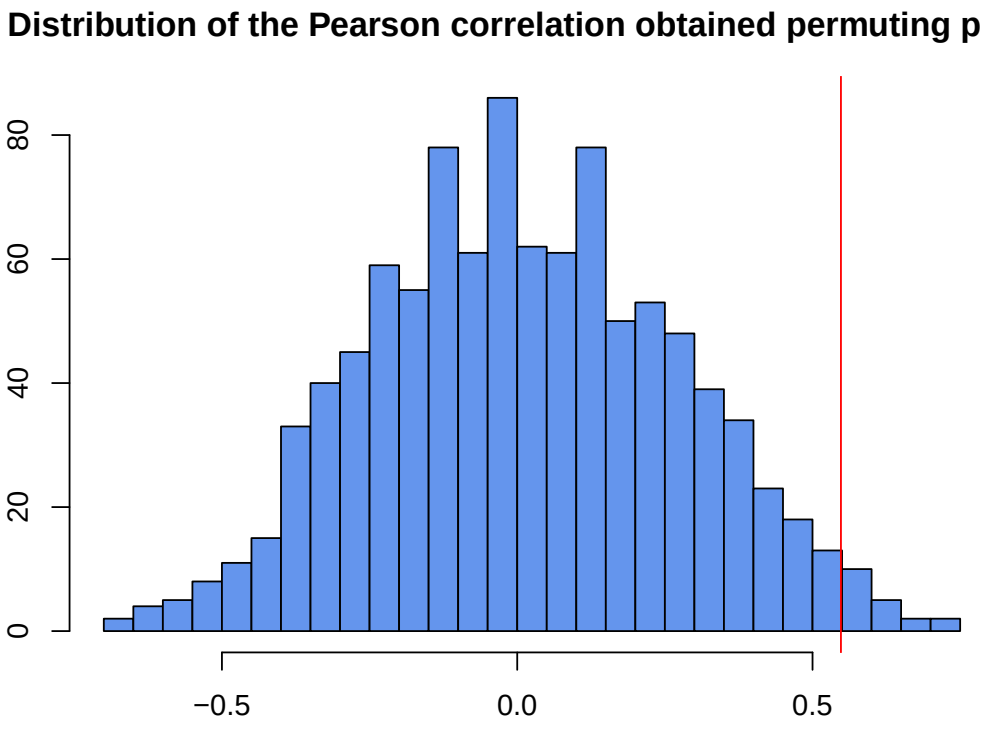
```
sample_matrix<-matrix(NA,1000,1) # Initializing the matrix

for (k in c(1:1000))
{
  PFA.P_PERM<-sample(PFA.P) # Permutation of Priors
  # Compute correlation between degree and Priors permuted
  sample_matrix[k,1]<-cor(PFMn.Deg,PFA.P_PERM)
}
```

We can inspect the matrix filled with correlation values to notice some basic statistics. Remember that originally we got 0.5480948 as Pearson correlation value, which is above the 3rd quantile, but below the maximum correlation reached.

```
## Information of the Matrix with correlations
summary(sample_matrix)
```

```
##        V1
##  Min.   :-0.6814845
##  1st Qu.:-0.1684472
##  Median :-0.0008551
##  Mean   : 0.0128909
##  3rd Qu.: 0.1992295
##  Max.   : 0.7430490
```

This plot shows the distribution of the Pearson correlation obtained permuting the priors 1,000 times. We can notice that the mean is below the 0, while the score we obtained before is around 0.5, in the tail of the distribution.



We can compute the probability of obtaining that correlation score by computing:

- mean correlation above the real value;
- mean correlation below the real value.

```
## Sum
(corRealValue<-cor(PFMn.Deg,PFA$Priors)) # Actual correlation with no sampling
```

```
## [1] 0.5480948
```

```
sum(sample_matrix>=corRealValue)/1000 # Mean correlation above actual cor
```

```
## [1] 0.019
```

```
sum(sample_matrix<=-corRealValue)/1000 # Mean correlation below actual cor
```

```
## [1] 0.011
```

```
# Summing them
sum(sample_matrix>=corRealValue)/1000 + sum(sample_matrix<=-corRealValue)/1000
```

```
## [1] 0.03
```

The same thing can be done by considering the absolute value of the correlation score, in order to get the probability of getting such score in both tails.

```
# Consider the absolute number of correlation
sum(sample_matrix>=abs(corRealValue))/1000
```

```
## [1] 0.019
```

```
sum(sample_matrix<=-abs(corRealValue))/1000
```

```
## [1] 0.011
```

```
sum(sample_matrix>=abs(corRealValue))/1000 +
  sum(sample_matrix<=-abs(corRealValue))/1000
```

```
## [1] 0.03
```

If we approximate the distribution to a Gaussian, we could use `pnorm()` to get the probability for the Pearson coefficient to be equal or higher than the real coefficient.

```
# Positive correlation Probability
pnorm(corRealValue, mean = mean(sample_matrix),
      sd = sd(sample_matrix), lower.tail = F)
```

```
## [1] 0.01983379
```

```
# Negative correlatio probability
pnorm(-corRealValue, mean = mean(sample_matrix),
      sd = sd(sample_matrix), lower.tail = T)
```

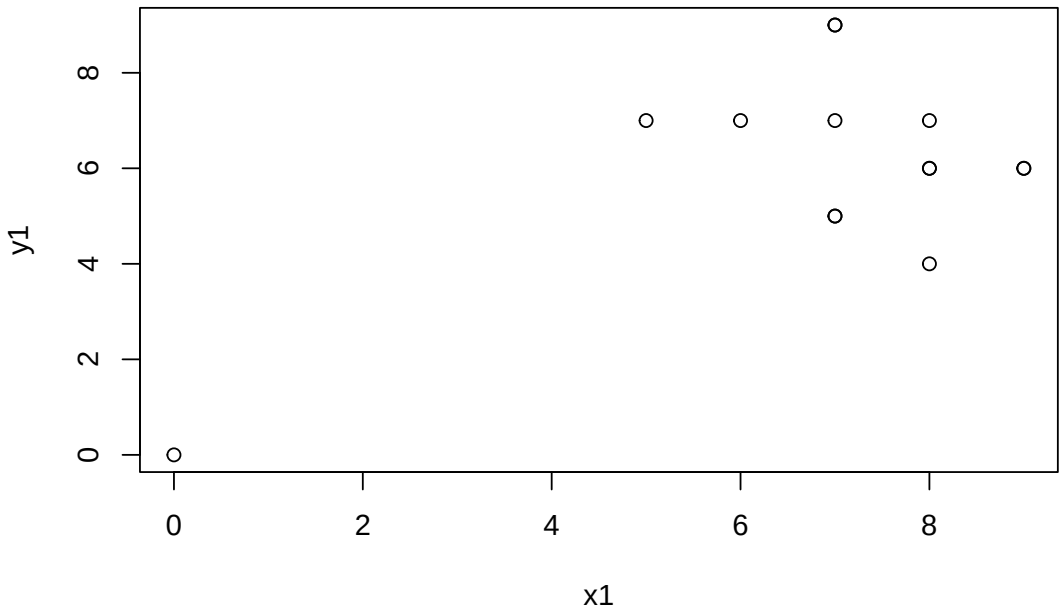
```
## [1] 0.0155304
```

```
# Both positive and negative probability
pnorm(corRealValue, mean = mean(sample_matrix),
      sd = sd(sample_matrix), lower.tail = F)+
  pnorm(-corRealValue, mean = mean(sample_matrix),
        sd = sd(sample_matrix), lower.tail = T)
```

```
## [1] 0.03536419
```

Suppose now we detain new data x_1, y_1 as follows, with a cluster in the upper part of the plot and a single data point in the lower left part.

```r
# SPECIFIC DATA SAMPLING
x1<-c(7,8,6,7,0,8,9,7,8,5,9,7,8,7)
y1<-c(5,6,7,9,0,4,6,5,6,7,6,7,7,9)
plot(x1,y1)
```

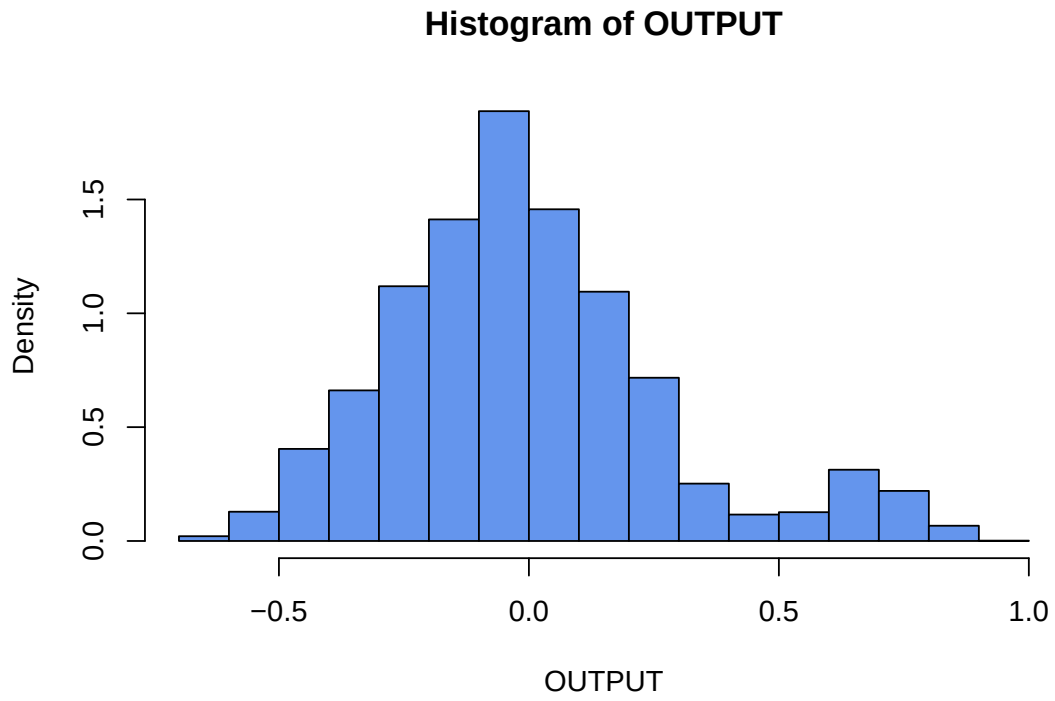



```r
cor.test(x1,y1)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x1 and y1
## t = 2.5049, df = 12, p-value = 0.02767
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08035754 0.85173373
## sample estimates:
##       cor
## 0.5859543
```

The correlation is 0.5859543 with p-value 0.0276672. Verify if the relationship between x_1 and y_1 is independent or not through a permutation based approach that iterates $20,000$ times.

```r
# Initializing output as matrix that does not follow a
# normal distribution through permutation based approaches
OUTPUT<-matrix(NA,20000,1)
for (k in c(1:20000))
{
  x1_PERM<-sample(x1)
  OUTPUT[k,1]<-cor(y1,x1_PERM)
}
hist(OUTPUT, nclass=20, prob=T, col = "cornflowerblue")
```



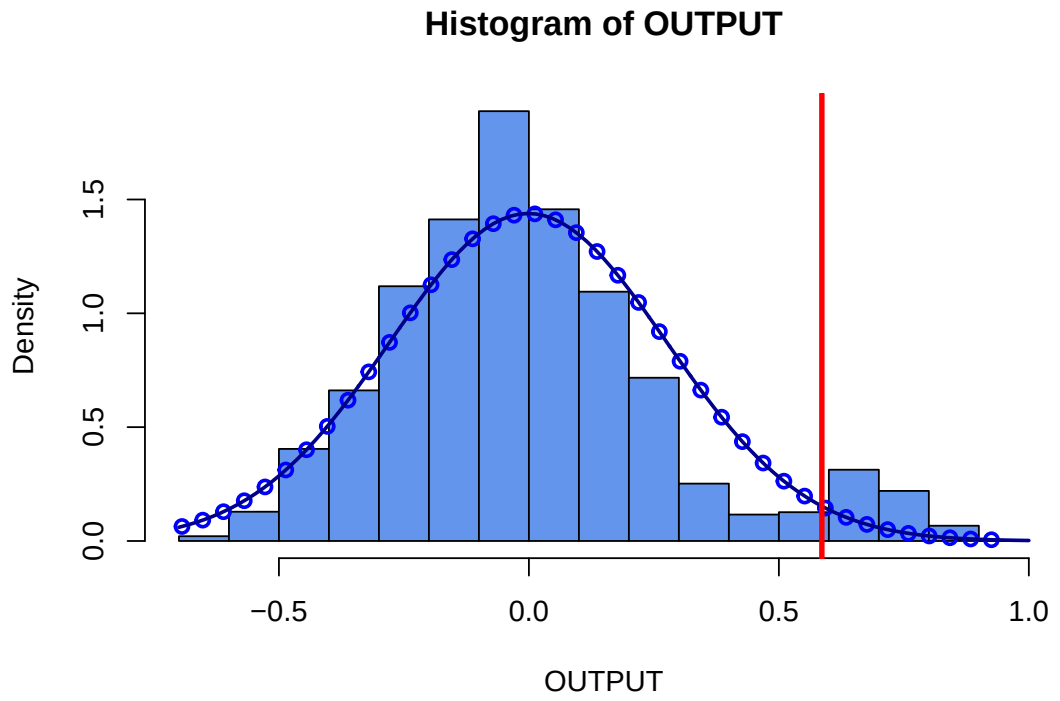
We could inspect the 95% confidence interval of these permutations:

```
## 95%-CI
cbind(mean(OUTPUT)+sd(OUTPUT)*1.96, mean(OUTPUT)-sd(OUTPUT)*1.96)
```

```
##            [,1]       [,2]
## [1,] 0.5422184 -0.5449739
```

A second approach considers x as the sequence of 40 numbers that go from the minimum to the maximum correlation value obtained in the previous permutation example. f is the normal distribution obtained considering its mean and standard deviation. The reproduced histogram is similar to the previous one.

```
## 2 approaches
x <- seq(min(OUTPUT), max(OUTPUT), length = 40)
f <- dnorm(x, mean = mean(OUTPUT), sd = sd(OUTPUT))

# Plotting the Gaussian distribution
hist(OUTPUT, nclass=20, prob=T, col="cornflowerblue")
lines(x, f, col = "blue", lwd = 2, type = "b")
curve(dnorm(x, mean=mean(OUTPUT), sd=sd(OUTPUT)),
      col="darkblue", lwd=2, add=TRUE)
abline(v=cor(x1,y1), lwd=3, col="red")
```




```
sum(OUTPUT>=cor(x1,y1))/20000
```

```
## [1] 0.0633
```

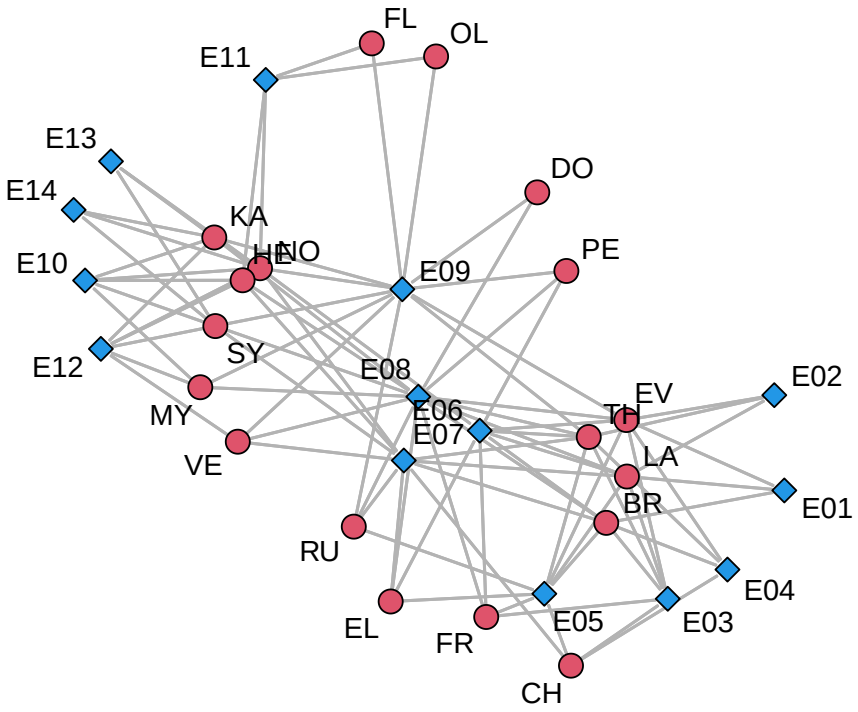
After $20,000$ permutations we can see that our correlation coefficient can be get in the top 6.33% of the correlation scores obtainable.

Chapter 18

Two-mode projections

In two-mode networks, rows and columns of the adjacency matrix represent different types of nodes. In this particular example, rows represent women attending to events (i.e. columns).

```
library(sna)

# Importing data
DSAm <- as.matrix(read.csv(
    "datasets/Davis_SouthernWomen_Attendance.csv",
    stringsAsFactors = FALSE, row.names=1))


# Limit letters to the first two letters
DSAmS<-DSAm
#Change names in new object
rownames(DSAmS)<-substring(rownames(DSAm),1,2)

par(mar = c(0,0,0,0))
# Now draw the network
gplot(DSAmS, displaylabels=TRUE,
      edge.col = "grey70",
      usearrows=FALSE, gmode="twomode")
```



In order to get one-mode projections, matrix multiplication is necessary, by computing the transpose and ordering the terms of the multiplication according to the nodes we're focusing on.

```
DSAmS%*%t(DSAmS) # Women
```

```
##    EV LA TH BR CH FR EL PE RU VE MY KA SY NO HE DO OL FL
## EV  8  6  7  6  3  4  3  3  3  2  2  2  2  2  1  2  1  1
## LA  6  7  6  6  3  4  4  2  3  2  1  1  2  2  2  1  0  0
## TH  7  6  8  6  4  4  4  3  4  3  2  2  3  3  2  2  1  1
## BR  6  6  6  7  4  4  4  2  3  2  1  1  2  2  2  1  0  0
## CH  3  3  4  4  4  2  2  0  2  1  0  0  1  1  1  0  0  0
## FR  4  4  4  4  2  4  3  2  2  1  1  1  1  1  1  1  0  0
## EL  3  4  4  4  2  3  4  2  3  2  1  1  2  2  2  1  0  0
## PE  3  2  3  2  0  2  2  3  2  2  2  2  2  2  1  2  1  1
## RU  3  3  4  3  2  2  3  2  4  3  2  2  3  2  2  2  1  1
## VE  2  2  3  2  1  1  2  2  3  4  3  3  4  3  3  2  1  1
## MY  2  1  2  1  0  1  1  2  2  3  4  4  4  3  3  2  1  1
## KA  2  1  2  1  0  1  1  2  2  3  4  6  6  5  3  2  1  1
## SY  2  2  3  2  1  1  2  2  3  4  4  6  7  6  4  2  1  1
## NO  2  2  3  2  1  1  2  2  2  3  3  5  6  8  4  1  2  2
## HE  1  2  2  2  1  1  2  1  2  3  3  3  4  4  5  1  1  1
## DO  2  1  2  1  0  1  1  2  2  2  2  2  2  1  1  2  1  1
## OL  1  0  1  0  0  0  0  1  1  1  1  1  1  2  1  1  2  2
## FL  1  0  1  0  0  0  0  1  1  1  1  1  1  2  1  1  2  2
```

```
t(DSAmS)%*%DSAmS # Events
```

```
##     E01 E02 E03 E04 E05 E06 E07 E08 E09 E10 E11 E12 E13 E14
## E01   3   2   3   2   3   3   2   3   1   0   0   0   0   0
## E02   2   3   3   2   3   3   2   3   2   0   0   0   0   0
## E03   3   3   6   4   6   5   4   5   2   0   0   0   0   0
## E04   2   2   4   4   4   3   3   3   2   0   0   0   0   0
```

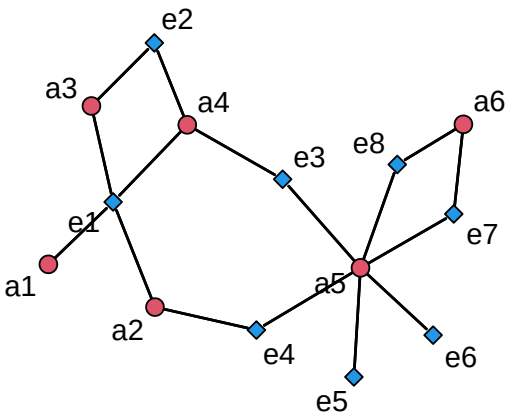
```
## E05   3   3   6   4   8   6   6   7   3   0   0   0   0   0
## E06   3   3   5   3   6   8   5   7   4   1   1   1   1   1
## E07   2   2   4   3   6   5  10   8   5   3   2   4   2   2
## E08   3   3   5   3   7   7   8  14   9   4   1   5   2   2
## E09   1   2   2   2   3   4   5   9  12   4   3   5   3   3
## E10   0   0   0   0   0   1   3   4   4   5   2   5   3   3
## E11   0   0   0   0   0   1   2   1   3   2   4   2   1   1
## E12   0   0   0   0   0   1   4   5   5   5   2   6   3   3
## E13   0   0   0   0   0   1   2   2   3   3   1   3   3   3
## E14   0   0   0   0   0   1   2   2   3   3   1   3   3   3
```

```
mean(DSAmS) # Degree
```

```
## [1] 0.3531746
```

Considering instead scientists and papers, we get a less complex network where *a*s represent scientists and *e*s represent papers.

```
scientists_papers<-matrix(c(1,0,0,0,0,0,0,0,
                            1,0,0,1,0,0,0,0,
                            1,1,0,0,0,0,0,0,
                            1,1,1,0,0,0,0,0,
                            0,0,1,1,1,1,1,1,
                            0,0,0,0,0,0,1,1),
                          6,8,byrow=TRUE)
rownames(scientists_papers)<-c("a1","a2","a3","a4","a5","a6")
colnames(scientists_papers)<-c("e1","e2","e3","e4","e5","e6","e7","e8")

gplot(scientists_papers, displaylabels=TRUE, usearrows=FALSE, gmode="twomode")
```




```
# Number of papers per scientist
rowSums(scientists_papers)
```

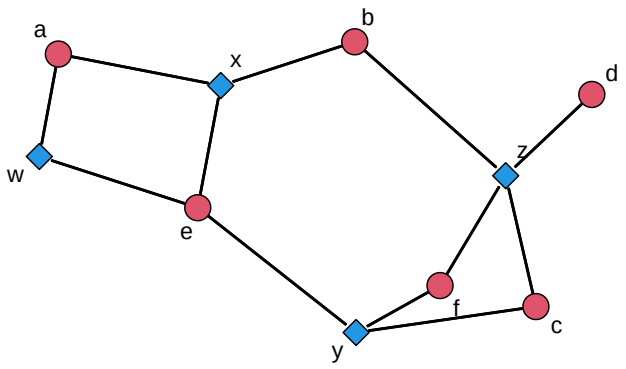
```
## a1 a2 a3 a4 a5 a6
##  1  2  2  3  6  2
```

```
# Authors per each paper
colSums(scientists_papers)
```

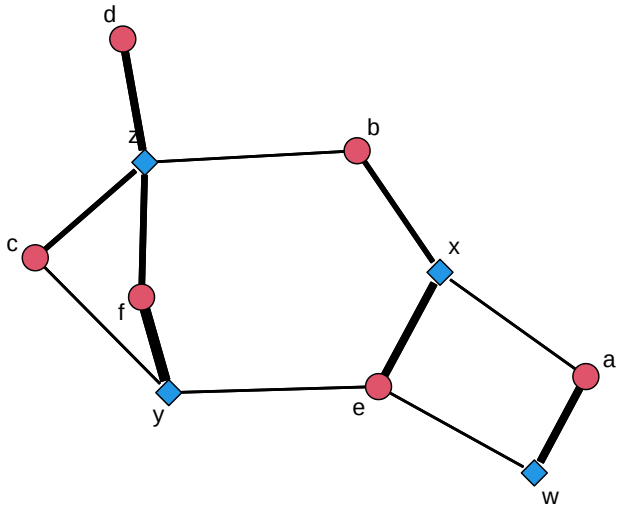
```
## e1 e2 e3 e4 e5 e6 e7 e8
##  4  2  2  2  1  1  2  2
```

The following example instead of inserting 1s and 0s contains weights, which represent the strength of the link between a node and another.

```
MATRIX2V<-matrix(c(5,1,0,0,
                   0,3,0,1,
                   0,0,1,3,
                   0,0,0,5,
                   1,5,1,0,
                   0,0,7,4),6,4, byrow=TRUE)
rownames(MATRIX2V)<-c("a","b","c","d","e","f")
colnames(MATRIX2V)<-c("w","x","y","z")
par(mar=c(0,0,0,0))
gplot(MATRIX2V, displaylabels=TRUE,
      usearrows=FALSE, gmode="twomode")
```




```
# Weighted plot
par(mar=c(0,0,0,0))
gplot(MATRIX2V, displaylabels=TRUE,
      usearrows=FALSE,
      gmode="twomode",
      edge.lwd=1)
```




```
# Centrality Measures about Authors
# Number of papers per author
rowSums(scientists_papers)
```

```
## a1 a2 a3 a4 a5 a6
##  1  2  2  3  6  2
```

```
# Normalized number of papers per author
rowSums(scientists_papers)/NCOL(scientists_papers)
```

```
##    a1    a2    a3    a4    a5    a6
## 0.125 0.250 0.250 0.375 0.750 0.250
```

```
# Centrality Measures about Papers
# Number of authors per paper
colSums(scientists_papers)
```

```
## e1 e2 e3 e4 e5 e6 e7 e8
##  4  2  2  2  1  1  2  2
```

```
# Normalized number of authors per paper
colSums(scientists_papers)/NROW(scientists_papers)
```

```
##        e1        e2        e3        e4        e5        e6        e7        e8
## 0.6666667 0.3333333 0.3333333 0.3333333 0.1666667 0.1666667 0.3333333 0.3333333
```

```
# Proportion of participation of authors for each paper
t(scientists_papers)/colSums(scientists_papers)
```

```
##      a1   a2   a3   a4  a5  a6
## e1 0.25 0.25 0.25 0.25 0.0 0.0
## e2 0.00 0.00 0.50 0.50 0.0 0.0
## e3 0.00 0.00 0.00 0.50 0.5 0.0
## e4 0.00 0.50 0.00 0.00 0.5 0.0
## e5 0.00 0.00 0.00 0.00 1.0 0.0
## e6 0.00 0.00 0.00 0.00 1.0 0.0
## e7 0.00 0.00 0.00 0.00 0.5 0.5
## e8 0.00 0.00 0.00 0.00 0.5 0.5
```

```r
# Same but transposed
t(t(scientists_papers)/colSums(scientists_papers))
```

```
##      e1  e2  e3  e4 e5 e6  e7  e8
## a1 0.25 0.0 0.0 0.0  0  0 0.0 0.0
## a2 0.25 0.0 0.0 0.5  0  0 0.0 0.0
## a3 0.25 0.5 0.0 0.0  0  0 0.0 0.0
## a4 0.25 0.5 0.5 0.0  0  0 0.0 0.0
## a5 0.00 0.0 0.5 0.5  1  1 0.5 0.5
## a6 0.00 0.0 0.0 0.0  0  0 0.5 0.5
```

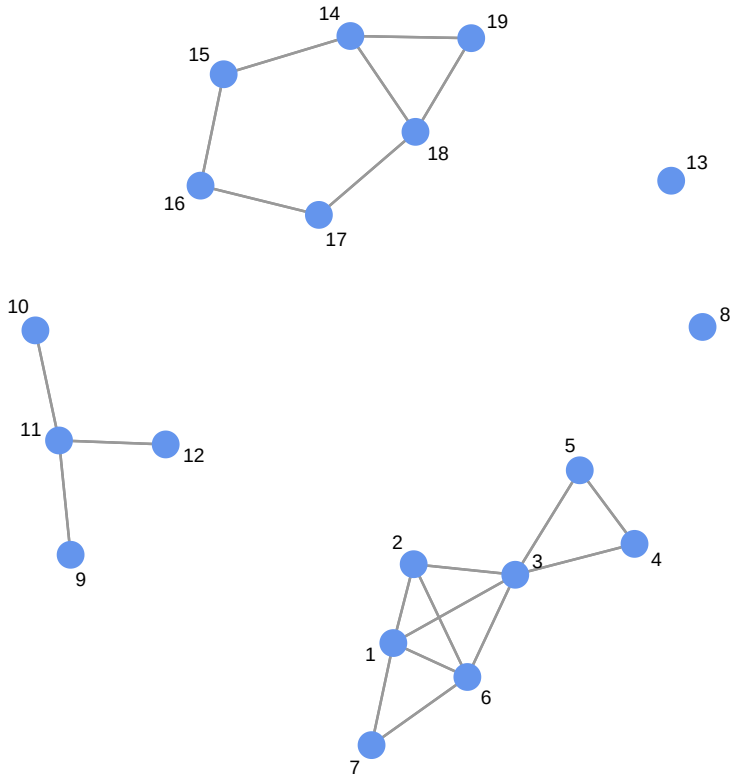
```r
# Percentage of participation into papers
rowSums(t(t(scientists_papers)/colSums(scientists_papers)))
```

```
##   a1   a2   a3   a4   a5   a6
## 0.25 0.75 0.75 1.25 4.00 1.00
```

Chapter 19

Subgroups and equivalence

Suppose we focus on the following network:

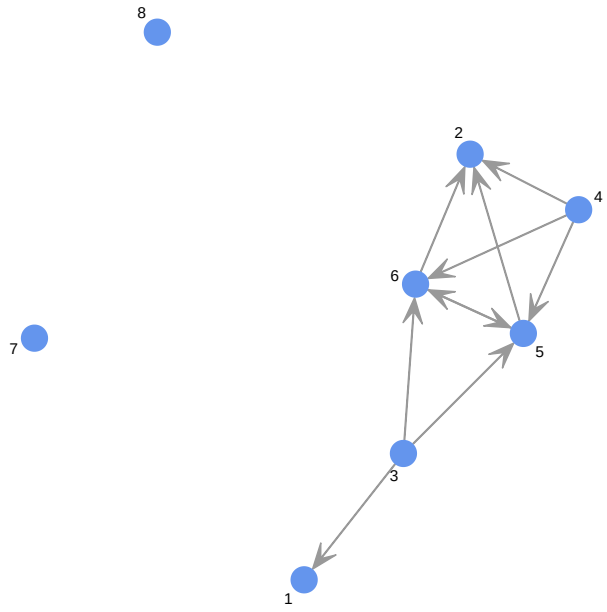


We can compute the census of the cliques inside this network. In particular, we can count per each node and per each k-clique how many cliques it belongs to.

```
clique.census(network, mode = "graph")$clique.count
```

```
##   Agg v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16 v17 v18 v19
## 1   2  0  0  0  0  0  0  0  1  0   0   0   0   1   0   0   0   0   0   0
## 2   7  0  0  0  0  0  0  0  0  1   1   3   1   0   1   2   2   2   1   0
## 3   3  1  0  1  1  1  1  1  0  0   0   0   0   0   1   0   0   0   1   1
## 4   1  1  1  1  0  0  1  0  0  0   0   0   0   0   0   0   0   0   0   0
```

Consider on the other side this network:

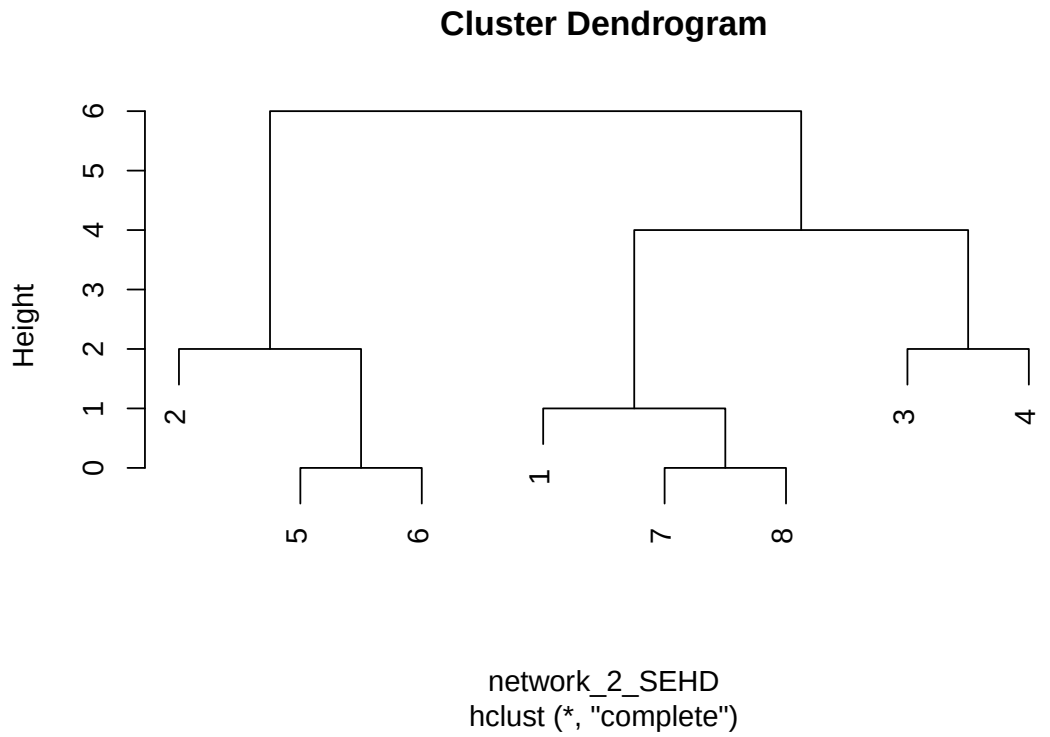


Then the Hamming distance can be computed. Look at 7 and 8: they are both isolated and their Hamming distance amounts to 0. It basically represents how many edges nodes do not share with the other node. For instance, 3 has 3 links, while 5 has 5 links. They share a common link to 8 and they are linked, therefore their distance is $1 + 3 = 4$.

```
# Hamming Structural Equivalence
network_2_SEH<-sedist(network_2, method="hamming")
network_2_SEH
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    4    2    4    4    4    1    1
## [2,]    4    0    6    4    2    2    3    3
## [3,]    2    6    0    2    4    4    3    3
## [4,]    4    4    2    0    2    2    3    3
## [5,]    4    2    4    2    0    0    5    5
## [6,]    4    2    4    2    0    0    5    5
## [7,]    1    3    3    3    5    5    0    0
## [8,]    1    3    3    3    5    5    0    0
```

```
## Clustering on hamming distance
network_2_SEHD<-as.dist(network_2_SEH)
network_2_SEHD_HC<-hclust(network_2_SEHD,method="complete")
plot(network_2_SEHD_HC)
```

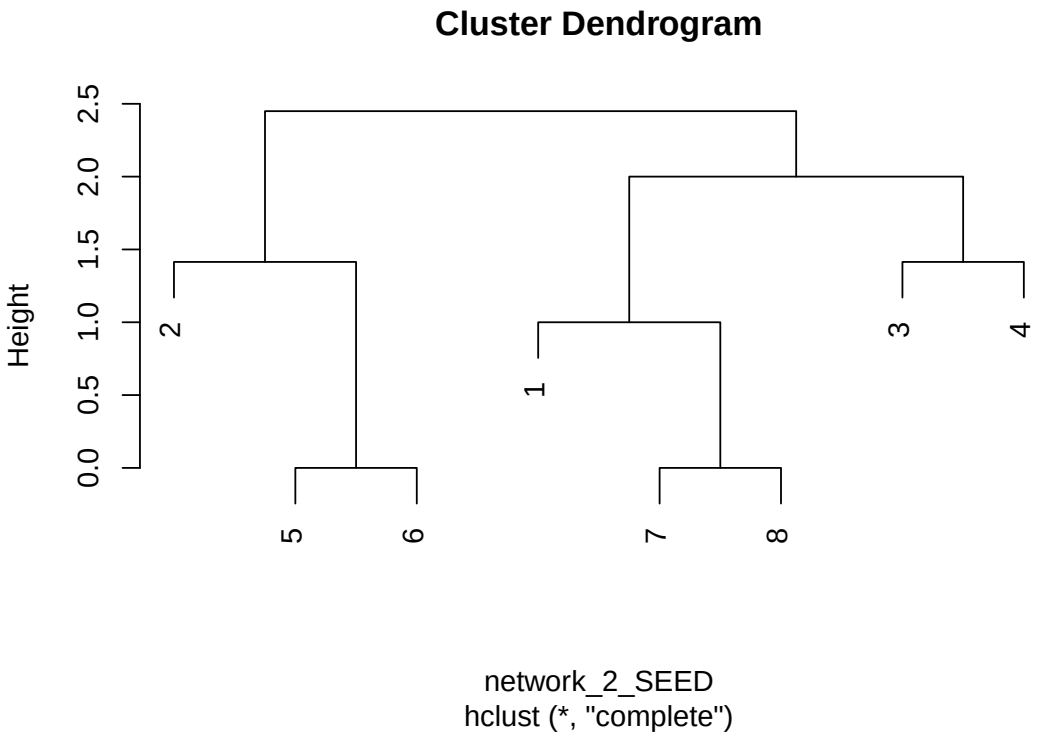


Same happens with the euclidean distance, which instead of absolute values considers the squared ones.

```
# Euclidean Structural Equivalence
network_2_SEE<-sedist(network_2, method="euclidean")
network_2_SEE
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] 0.000000 2.000000 1.414214 2.000000 2.000000 2.000000 1.000000 1.000000
## [2,] 2.000000 0.000000 2.449490 2.000000 1.414214 1.414214 1.732051 1.732051
## [3,] 1.414214 2.449490 0.000000 1.414214 2.000000 2.000000 1.732051 1.732051
## [4,] 2.000000 2.000000 1.414214 0.000000 1.414214 1.414214 1.732051 1.732051
## [5,] 2.000000 1.414214 2.000000 1.414214 0.000000 0.000000 2.236068 2.236068
## [6,] 2.000000 1.414214 2.000000 1.414214 0.000000 0.000000 2.236068 2.236068
## [7,] 1.000000 1.732051 1.732051 1.732051 2.236068 2.236068 0.000000 0.000000
## [8,] 1.000000 1.732051 1.732051 1.732051 2.236068 2.236068 0.000000 0.000000
```

```
## Clustering on euclidean distance
network_2_SEED<-as.dist(network_2_SEE)
network_2_SEED_HC<-hclust(network_2_SEED,method="complete")
plot(network_2_SEED_HC)
```


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
2.5
2
5
6
1
7
8
3
4
network_2_SEED
hclust (*, "complete")

Chapter 20

Dyads and triads

20.1 Dyads

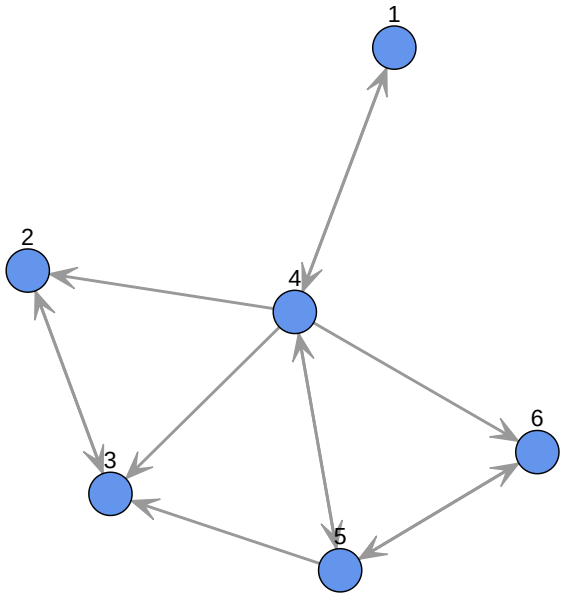


Figure 20.1: Example of a directed network

The `dyad.census()` function returns the number of mutual, asymmetric and null edges.

```
# Dyad Census
sna::dyad.census(mat6, g=NULL)
```

```
##      Mut Asym Null
## [1,]   4    4    7
```

In order to compute the reciprocity, there are two variants:

- the first one is the ratio of mutuals M and null edges N over the total amount of edges possible $(M + A + N)$:

```
sna::grecip(mat6,measure="dyadic")
```

```
##       Mut
## 0.7333333
```

- the second version just considers mutuals and asymmetric (i.e. $2M/(2M + A)$):

```
sna::grecip(mat6,measure="edgewise")
```

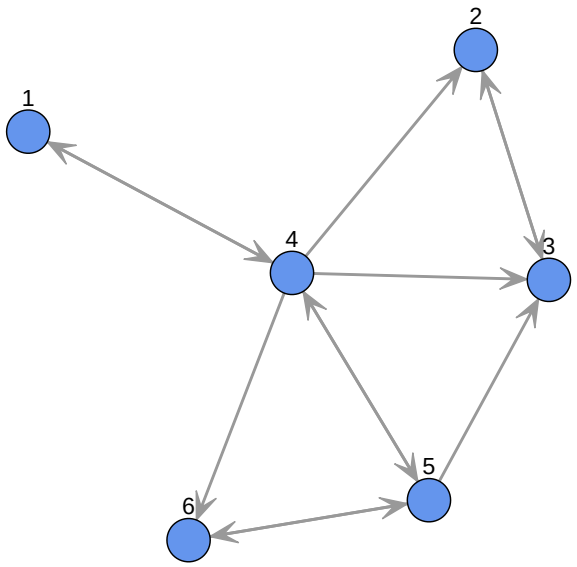
```
##       Mut
## 0.6666667
```

Also, igraph allows to compute the reciprocity but only with the second modality ($2M/(2M + A)$):

```
# With igraph
library(igraph)
mat6i<-graph_from_adjacency_matrix(mat6,mode=c("directed"),diag=F)
igraph::reciprocity(mat6i)
```

```
## [1] 0.6666667
```

20.2 Triads



Let's change network and compute the triad census. The function `triad.census()` will return the number of triads per type according to the MAN nomenclature.

```
# Triad census
sna::triad.census(mat6)
```

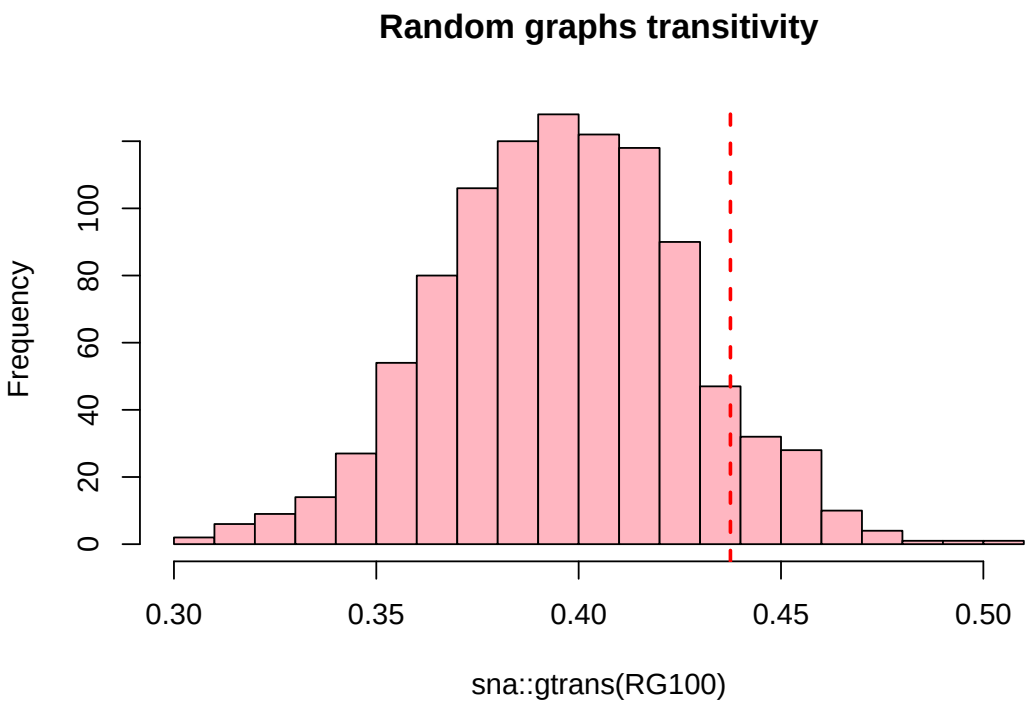
```
##      003 012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
## [1,]   3   1   4    2    0    0    1    5    0    0   1    1    1    0   1   0
```

For instance, there are 3 null edges ($(6, 1), (1, 2), (2, 6)$) and 1 210 $(4, 5, 6)$. The transitivity amounts to 0.4375, which is highly significant. We can test it creating 1,000 permutations of the graph and computing the transitivity of each of them. The probability of forming an edge equals the density of the original matrix.

```
sna::gtrans(mat6)
```

```
## [1] 0.4375
```

```
RG100<-sna::rgraph(20,1000,tprob=sna::gden(mat6))
hist(sna::gtrans(RG100), breaks = 20, col="lightpink",
     main = "Random graphs transitivity")
abline(v = sna::gtrans(mat6), col = "red", lty = 2, lwd = 2)
```


Random graphs transitivity
Frequency
0
20
40
60
80
100
0.30
0.35
0.40
0.45
0.50
sna::gtrans(RG100)