Day 3: Descriptive Analysis of Network Graph Characteristics

Sacha Epskamp

University of Amsterdam
Department of Psychological Methods

16-10-2013

Notation

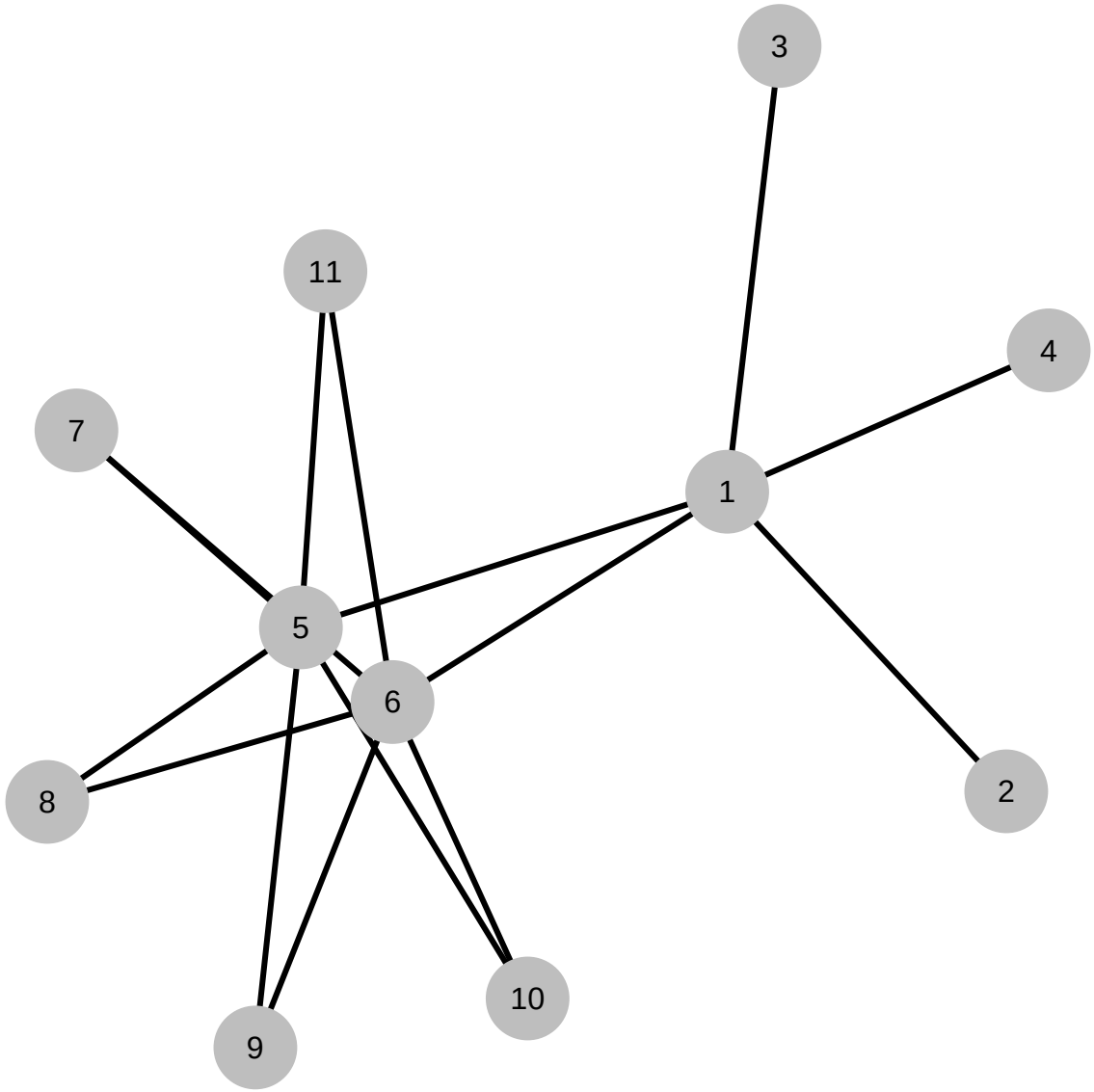
In these sheets we analyze mostly the unweighted, undirected simple graph G:

$$G = (V, E)$$

With $|N|$ nodes and $|E|$ edges, encoded using $|N| \times |N|$ adjacency matrix $\boldsymbol{A}$:

$$a_{vu} = a_{uv} = \begin{cases} 1 & \text{if } \{v, u\} \in E \\ 0 & \text{Otherwise} \end{cases}$$

```r
# Toy adjacency matrix:
A <- matrix(0, 11, 11)
A[1, 2:6] <- 1
A[5, 7:11] <- 1
A[6, 7:11] <- 1
A <- pmax(A, t(A))
# Create graph in igraph:
library("igraph")
G <- graph.adjacency(A, mode = "undirected")
# Or via qgraph:
library("qgraph")
G <- as.igraph(qgraph(A, DoNotPlot = TRUE))
```


3
11
4
7
1
5
6
8
2
10
9

Shortest Path length

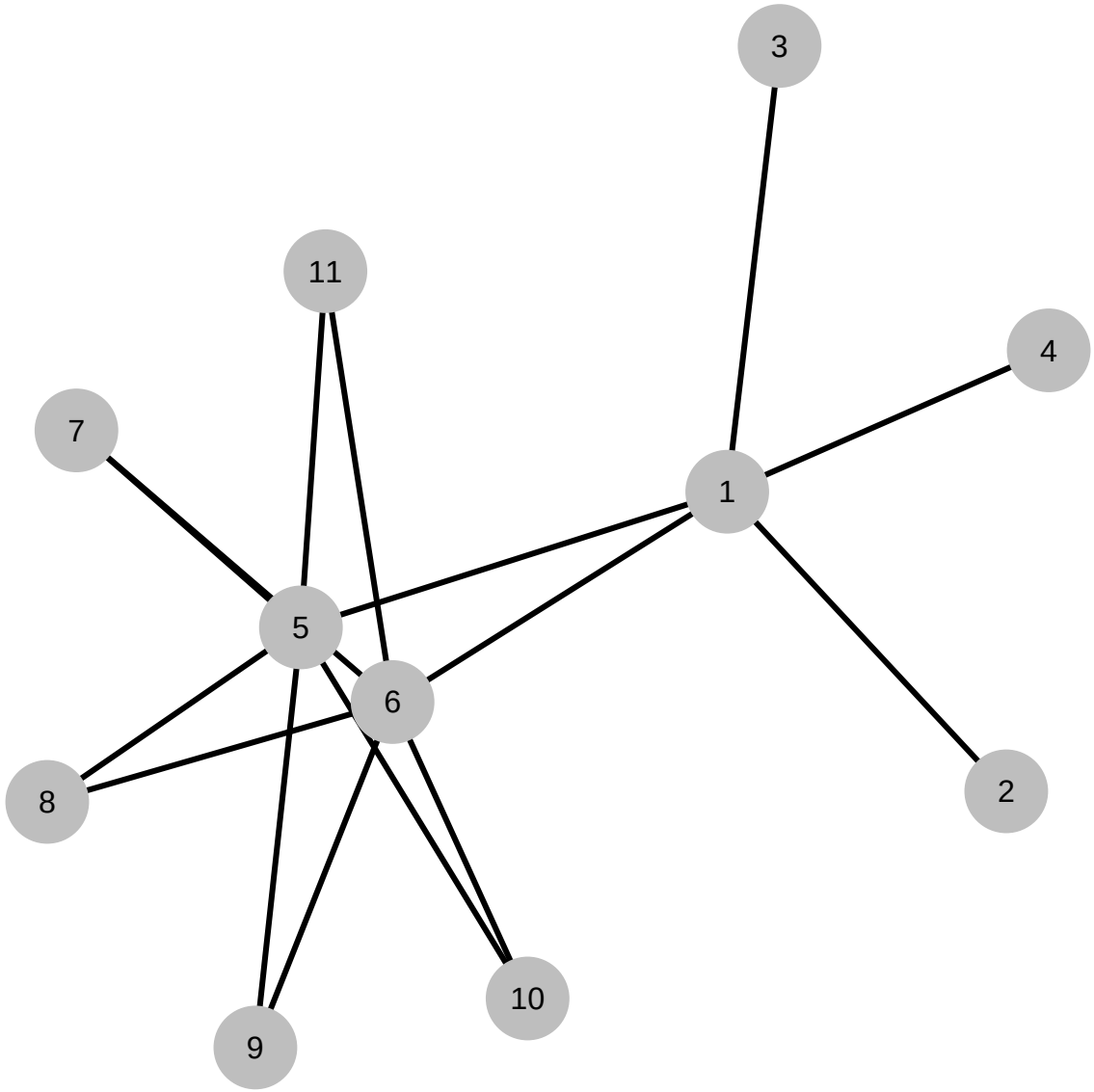
The shortest path length between nodes v and u, $\mathrm{dist}(v, u)$ is defined in an unweighted graph as the minimum number of steps you need to take from node v to node u:

$$\mathrm{dist}(v, u) = \min\left(a_{vx} + \ldots + a_{yu}\right)$$

- Can be computed using Dijkstra's algorithm (Dijkstra, 1959) with weights fixed to 1.
- Commonly refered to as the *shortest path length* or *geodesic distance*

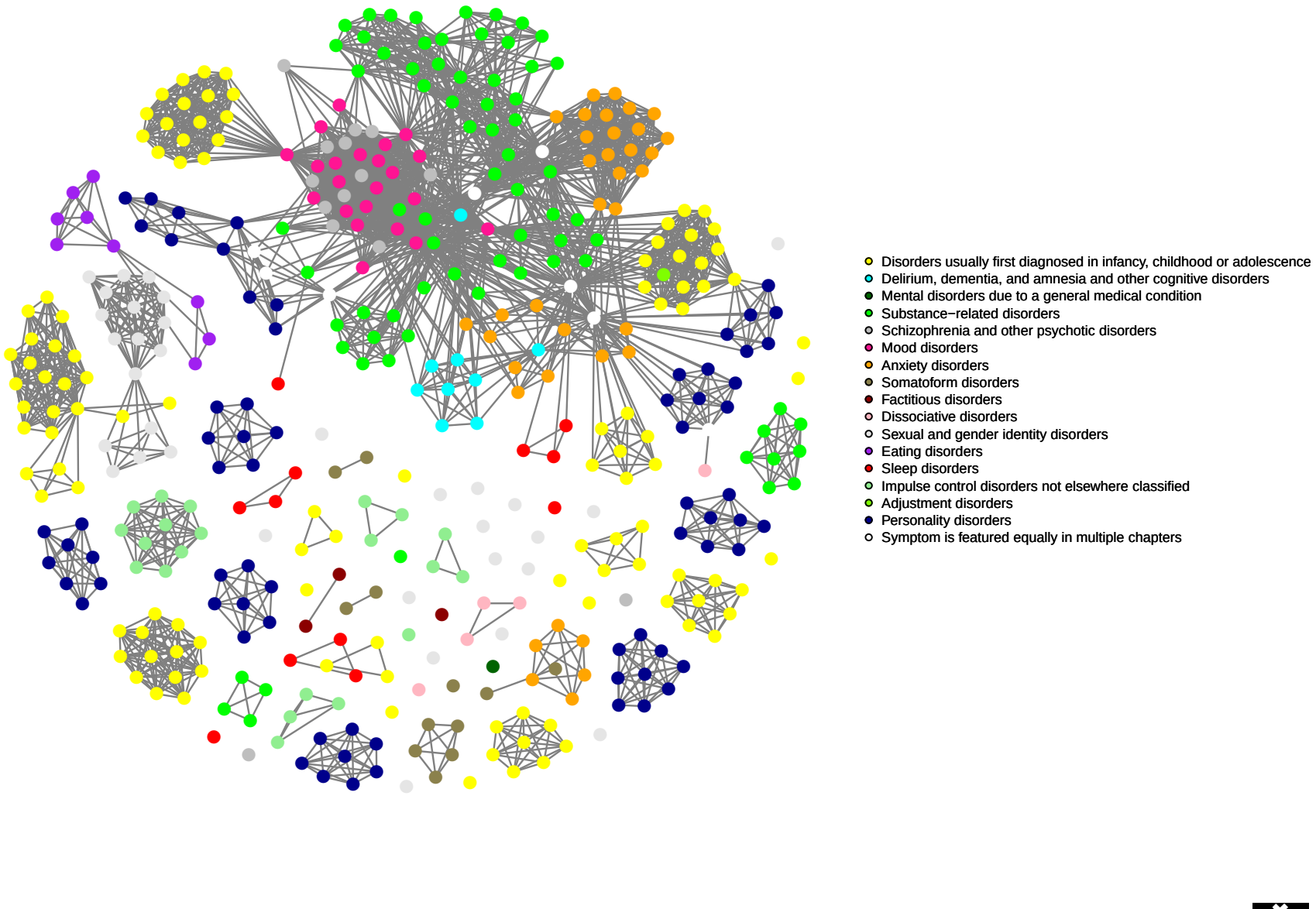
The mean shortest path length is called the *average shortest path length* (APL) and is an important measure for how well connected a graph is:

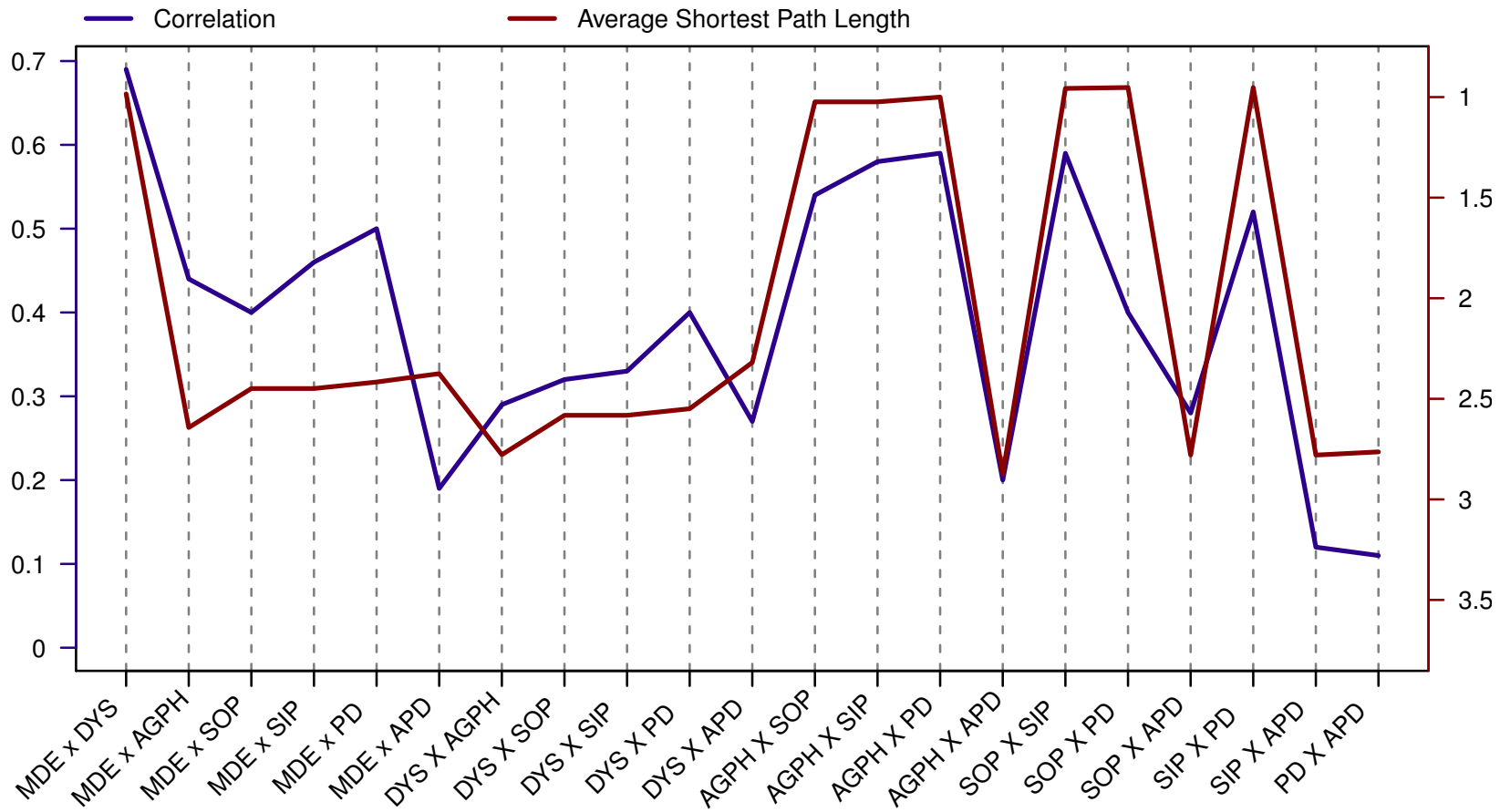
$$\mathrm{APL}(G) = \frac{\sum_{v,u} \mathrm{dist}(v, u)}{|N|\left(|N| - 1\right)/2}$$


3
11
4
7
1
5
6
8
2
10
9

```
shortest.paths(G)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    1    1    1    1    1    2    2    2     2
##  [2,]    1    0    2    2    2    2    3    3    3     3
##  [3,]    1    2    0    2    2    2    3    3    3     3
##  [4,]    1    2    2    0    2    2    3    3    3     3
##  [5,]    1    2    2    2    0    2    1    1    1     1
##  [6,]    1    2    2    2    2    0    1    1    1     1
##  [7,]    2    3    3    3    1    1    0    2    2     2
##  [8,]    2    3    3    3    1    1    2    0    2     2
##  [9,]    2    3    3    3    1    1    2    2    0     2
## [10,]    2    3    3    3    1    1    2    2    2     0
## [11,]    2    3    3    3    1    1    2    2    2     2
##       [,11]
##  [1,]     2
##  [2,]     3
##  [3,]     3
##  [4,]     3
##  [5,]     1
##  [6,]     1
##  [7,]     2
##  [8,]     2
##  [9,]     2
```

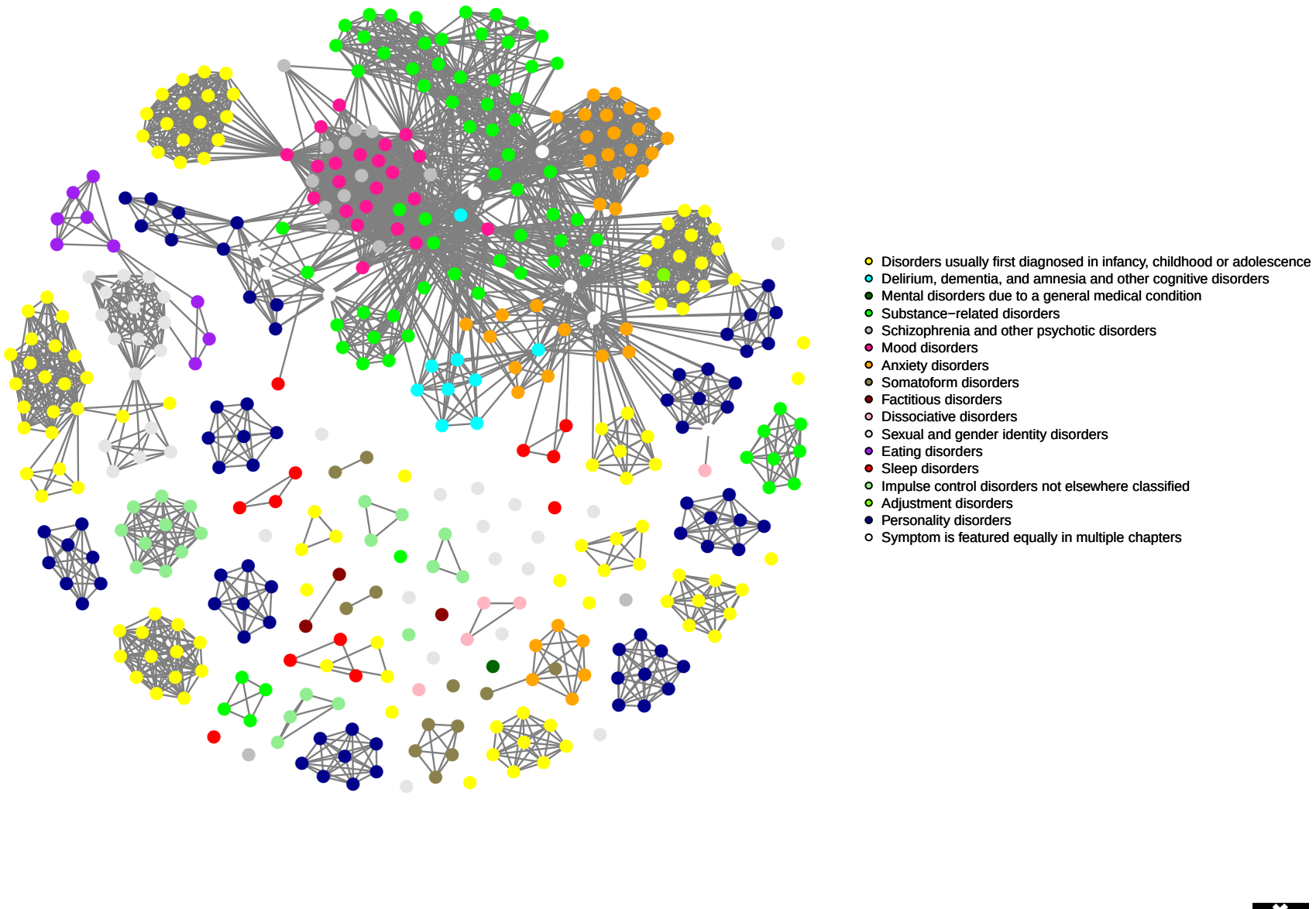

Disorders usually first diagnosed in infancy, childhood or adolescence
Delirium, dementia, and amnesia and other cognitive disorders
Mental disorders due to a general medical condition
Substance−related disorders
Schizophrenia and other psychotic disorders
Mood disorders
Anxiety disorders
Somatoform disorders
Factitious disorders
Dissociative disorders
Sexual and gender identity disorders
Eating disorders
Sleep disorders
Impulse control disorders not elsewhere classified
Adjustment disorders
Personality disorders
Symptom is featured equally in multiple chapters


Correlation
Average Shortest Path Length
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
1.5
2
2.5
3
3.5
MDE x DYS
MDE x AGPH
MDE x SOP
MDE x SIP
MDE x PD
MDE x APD
DYS X AGPH
DYS X SOP
DYS X SIP
DYS X PD
DYS X APD
AGPH X SOP
AGPH X SIP
AGPH X PD
AGPH X APD
SOP X SIP
SOP X PD
SOP X APD
SIP X PD
SIP X APD
PD X APD

Vertex centrality

Centrality measures assign numeric values to the importance of nodes in the graph and answer the question “what is the most central node?”.

- Degree
- Closeness
- Betweenness
- Eigenvector centrality

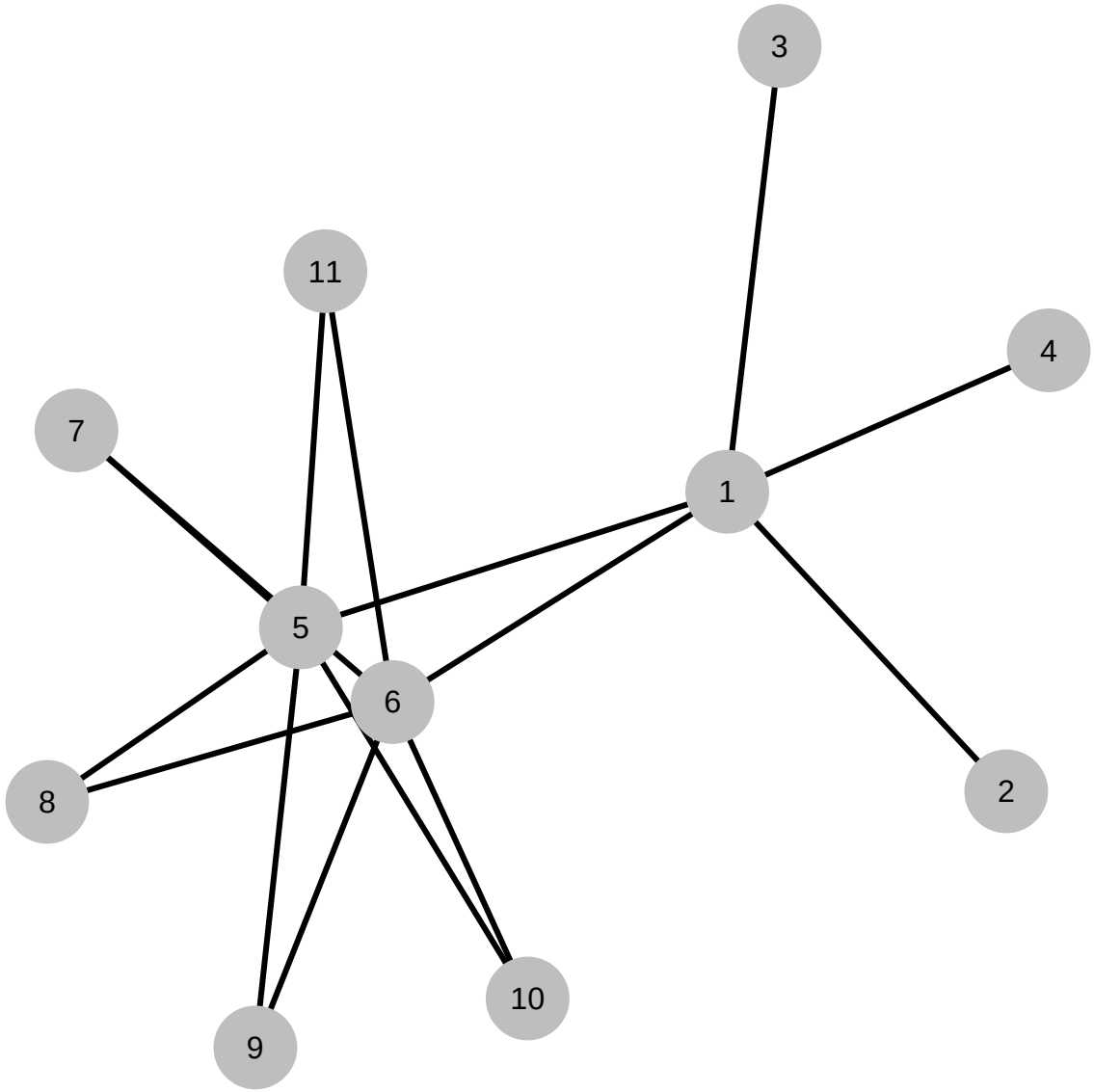

Disorders usually first diagnosed in infancy, childhood or adolescence
Delirium, dementia, and amnesia and other cognitive disorders
Mental disorders due to a general medical condition
Substance−related disorders
Schizophrenia and other psychotic disorders
Mood disorders
Anxiety disorders
Somatoform disorders
Factitious disorders
Dissociative disorders
Sexual and gender identity disorders
Eating disorders
Sleep disorders
Impulse control disorders not elsewhere classified
Adjustment disorders
Personality disorders
Symptom is featured equally in multiple chapters

Degree

The *degree* of node v, $C_D(v)$ is simply the number of edges connected to node v, which we can compute by either summing over row v or column v of $\boldsymbol{A}$:

$$C_D(v) = \sum_{i=1}^{|N|} a_{iv} = \sum_{j=1}^{|N|} a_{vj}$$

In the book the notation used for the degree is d_v.


3
11
4
7
1
5
6
8
2
10
9

```r
degree(G)
```

```
##  [1] 5 1 1 1 6 6 2 2 2 2 2
```

Degree distribution

The *degree distribution*, f_d, gives the probability that a node in G has degree d:

$$f_d = \mathbb{P}(C_D(v) = d)$$

For a given graph the observed degree distribution can simply be computed by dividing the number of nodes that have degree d with the total number of nodes:

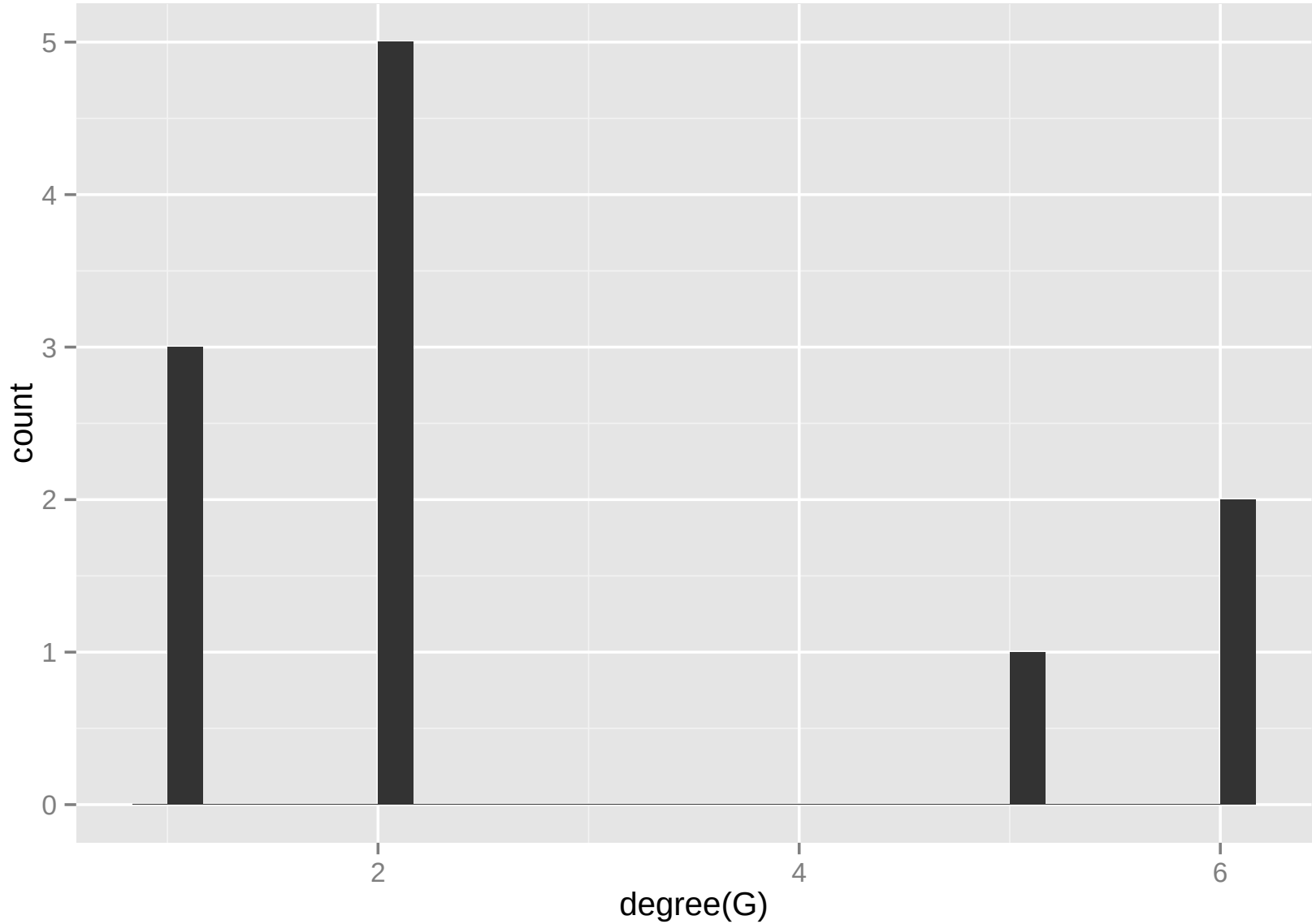
$$f_d = \frac{\text{\# of nodes with degree } d}{|N|}$$

and can easilly be represented with an histogram.

```r
table(degree(G))/vcount(G)
```

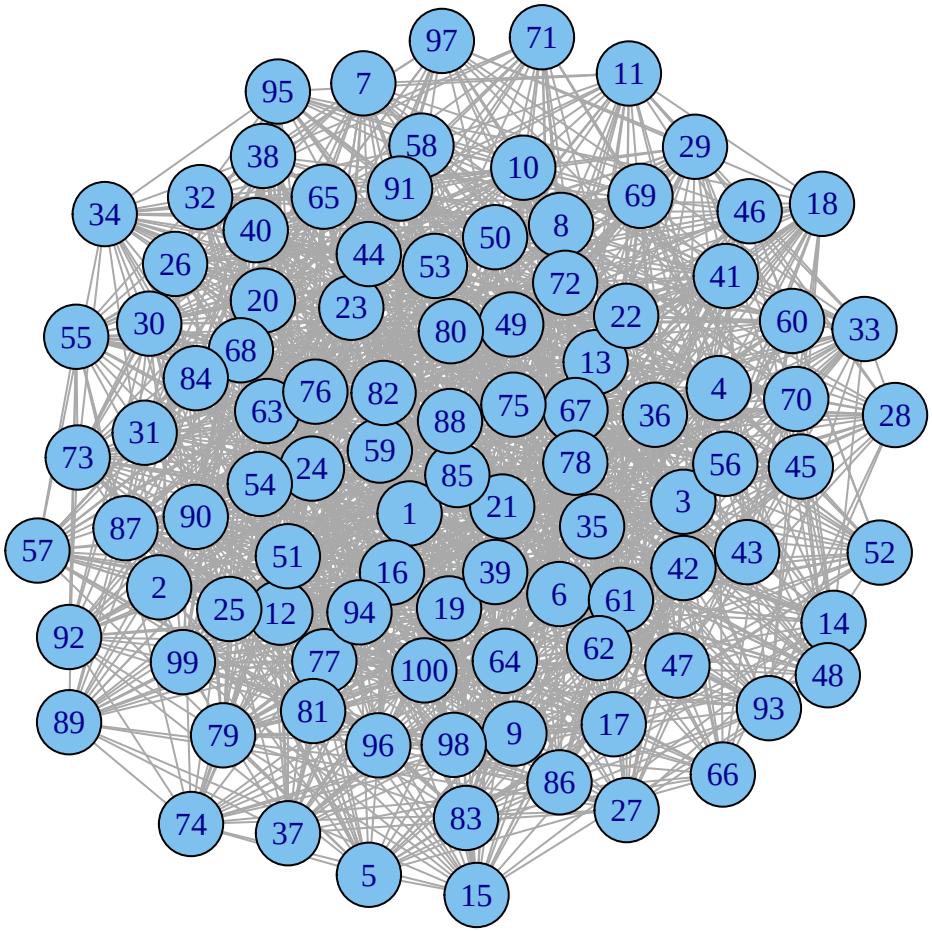
```
##
##       1       2       5       6
## 0.27273 0.45455 0.09091 0.18182
```

```
library("ggplot2")
qplot(degree(G), geom = "histogram")
```



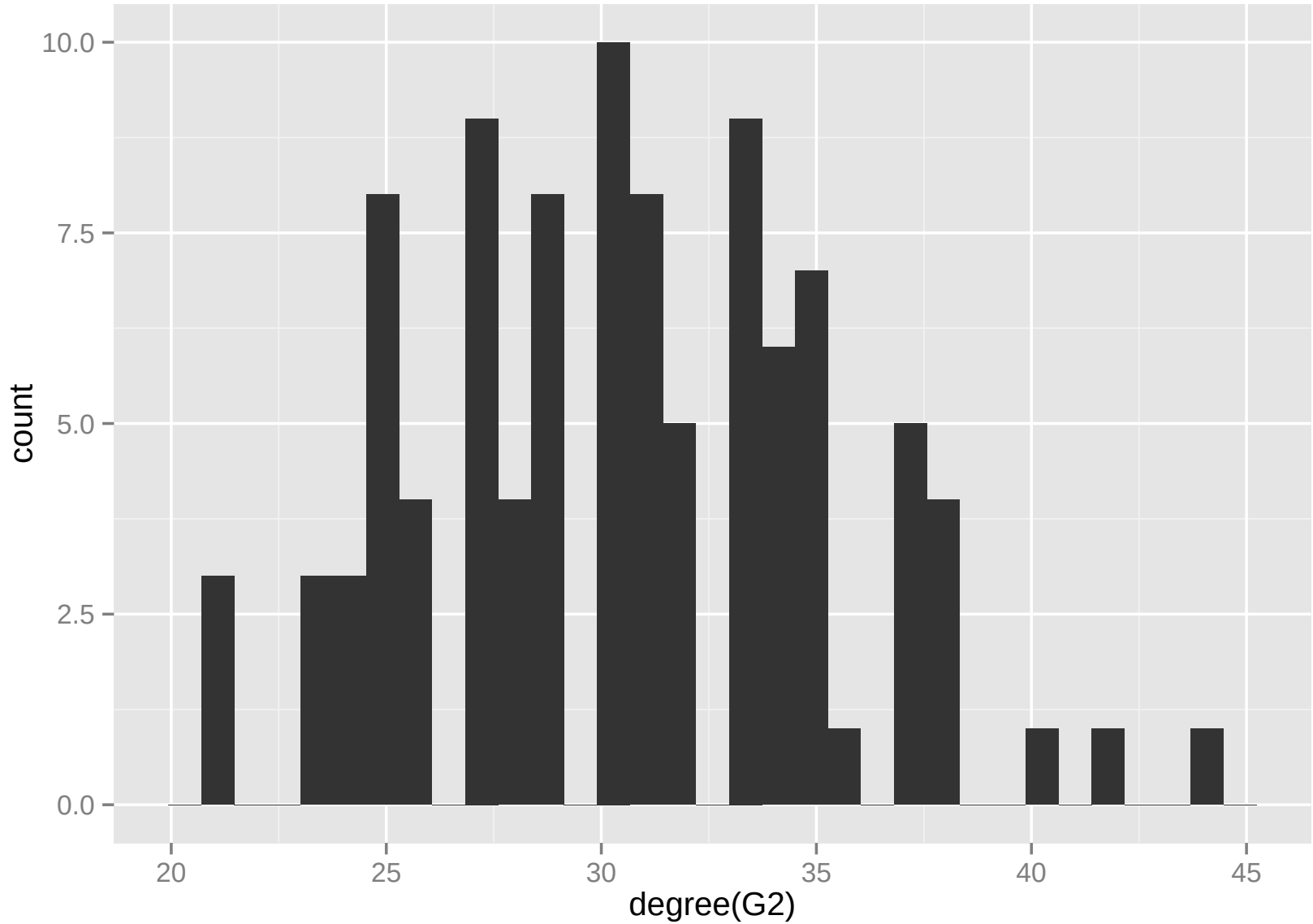
Random graph

```
G2 <- erdos.renyi.game(100, 0.3)
plot(G2)
```



Random graph

```
qplot(degree(G2), geom = "histogram")
```



Random graph

A random graph is a graph in which each edge is present with probability p. The degree distribution of a random graph follows a binomial distribution:

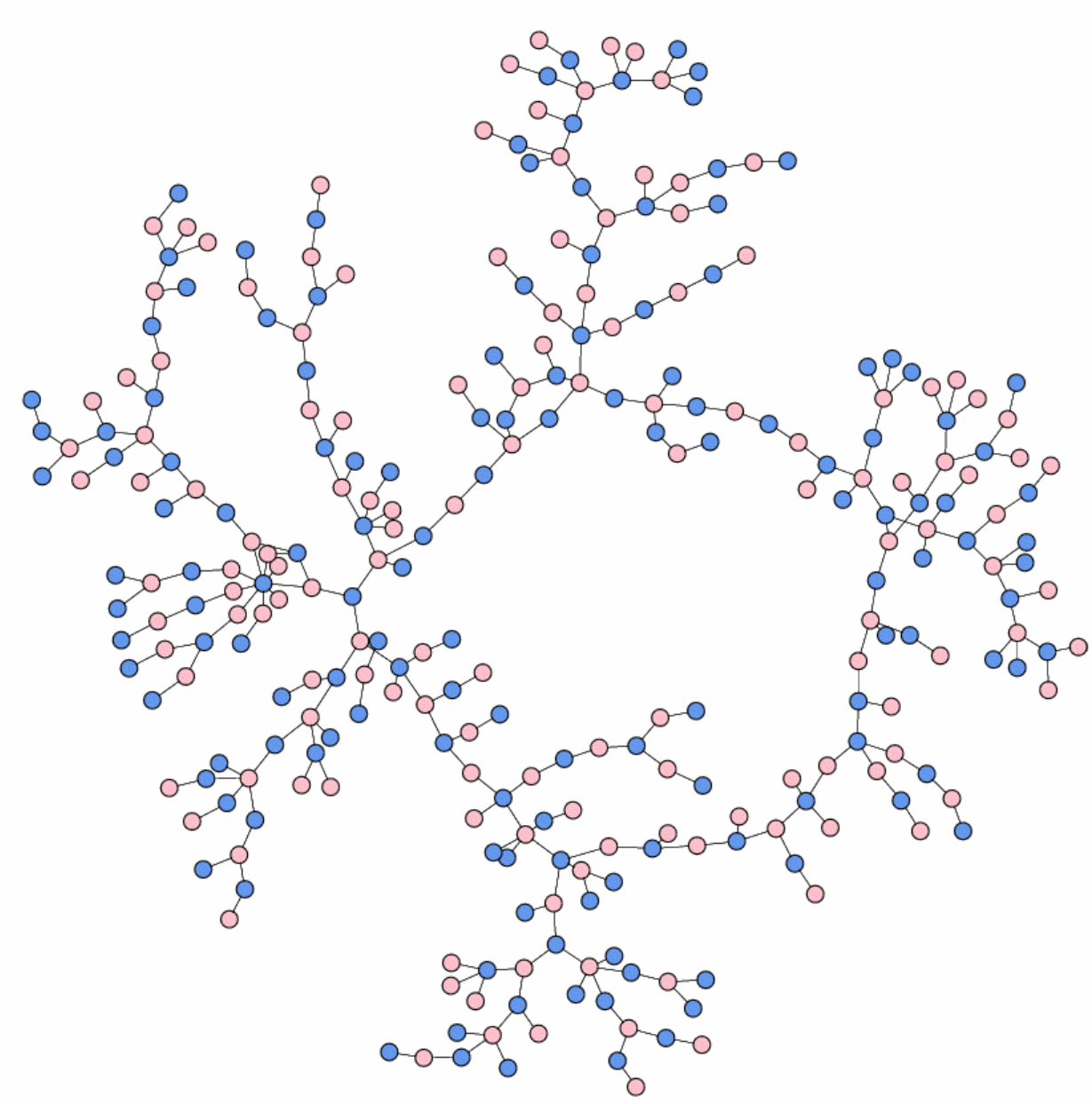
$$f_d = \binom{|N| - 1}{d} p^d (1 - p)^{|N|-1-d}$$

Or Poisson for large graphs:

$$f_d = \frac{\rho^d e^{-\rho}}{d!}$$

where $\rho = |N|\, p$

High School dating



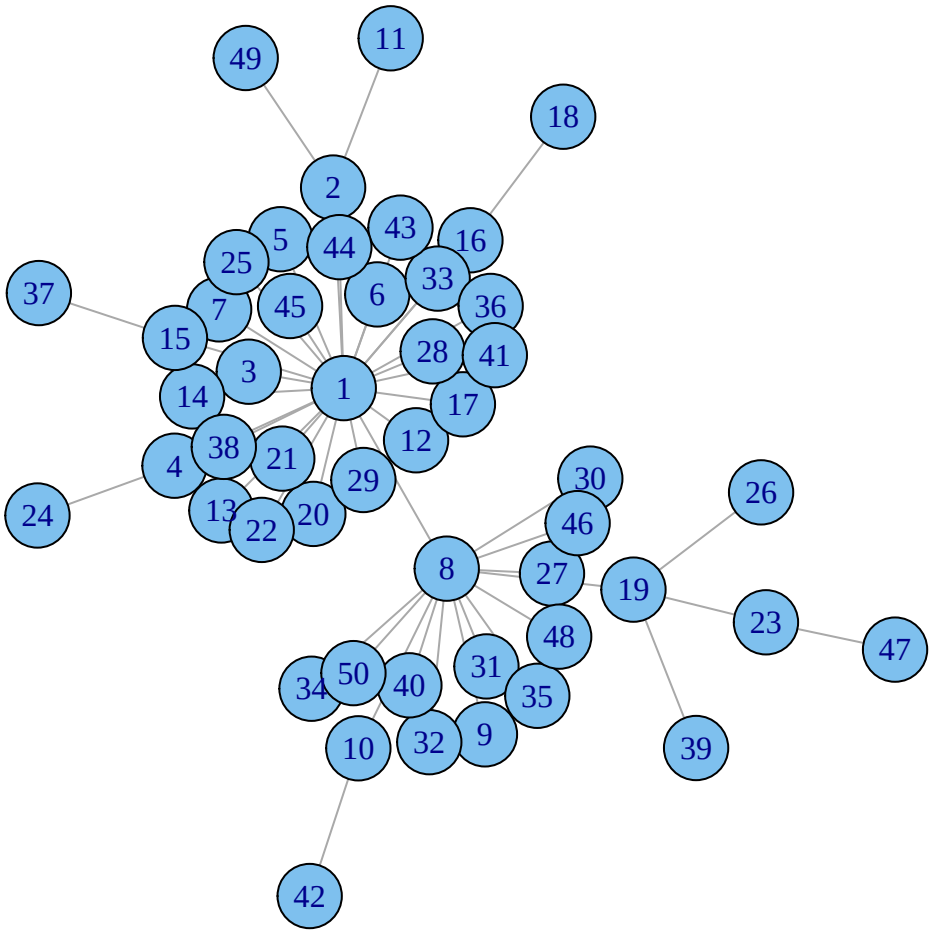
scale-free networks

- Many natural networks do not have a binomial or poisson degree distribution
- These graphs usually have many nodes with a very low degree and few nodes with a very high degree
- These are termed *scale-free networks*, and for these networks a power-law holds approximatly true at least in part.

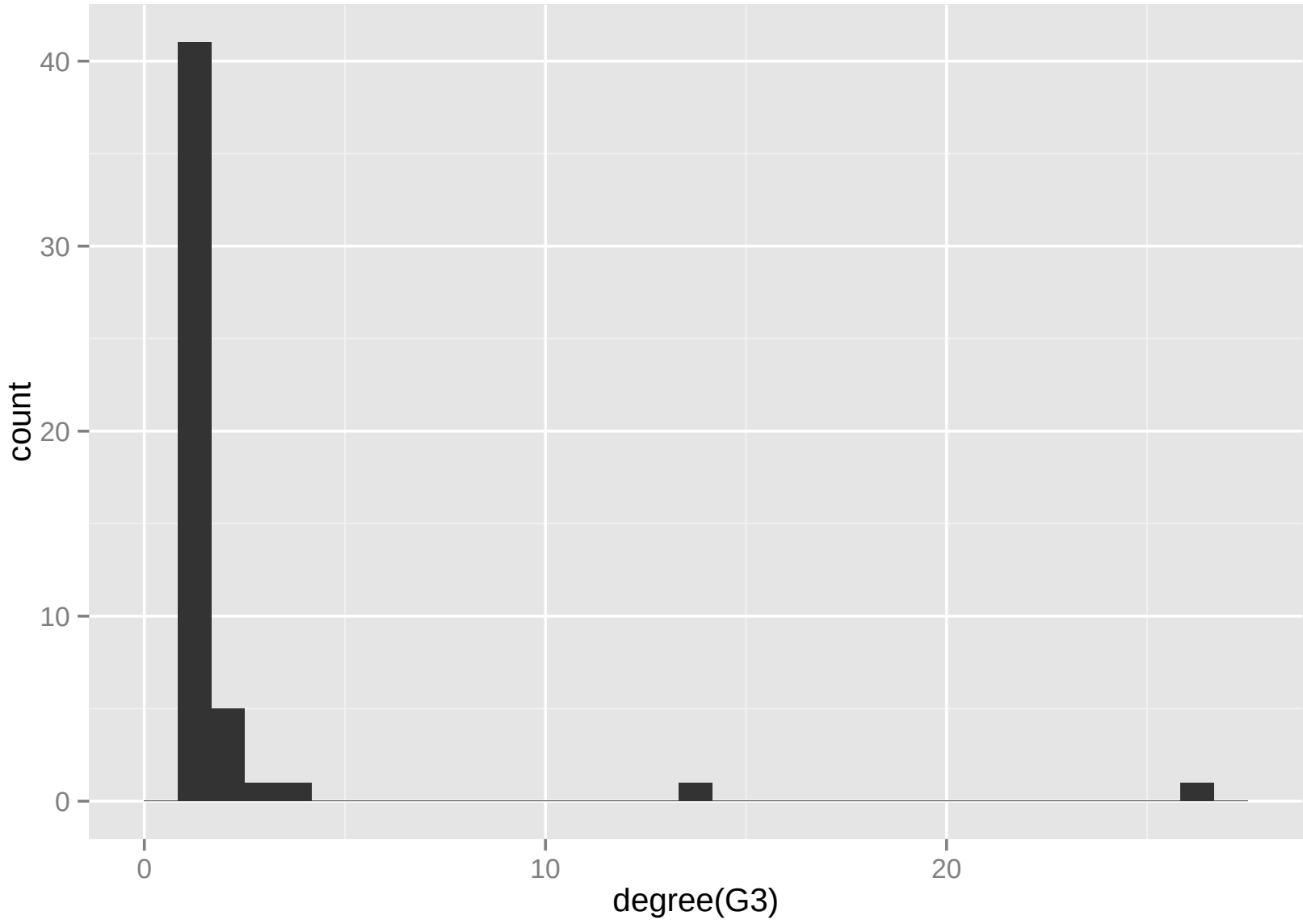
$$f_d \propto d^{-\alpha}$$

Preferential Attachment

```r
G3 <- barabasi.game(50, 1.2, directed = FALSE)
plot(G3)
```



```
qplot(degree(G3), geom = "histogram")
```



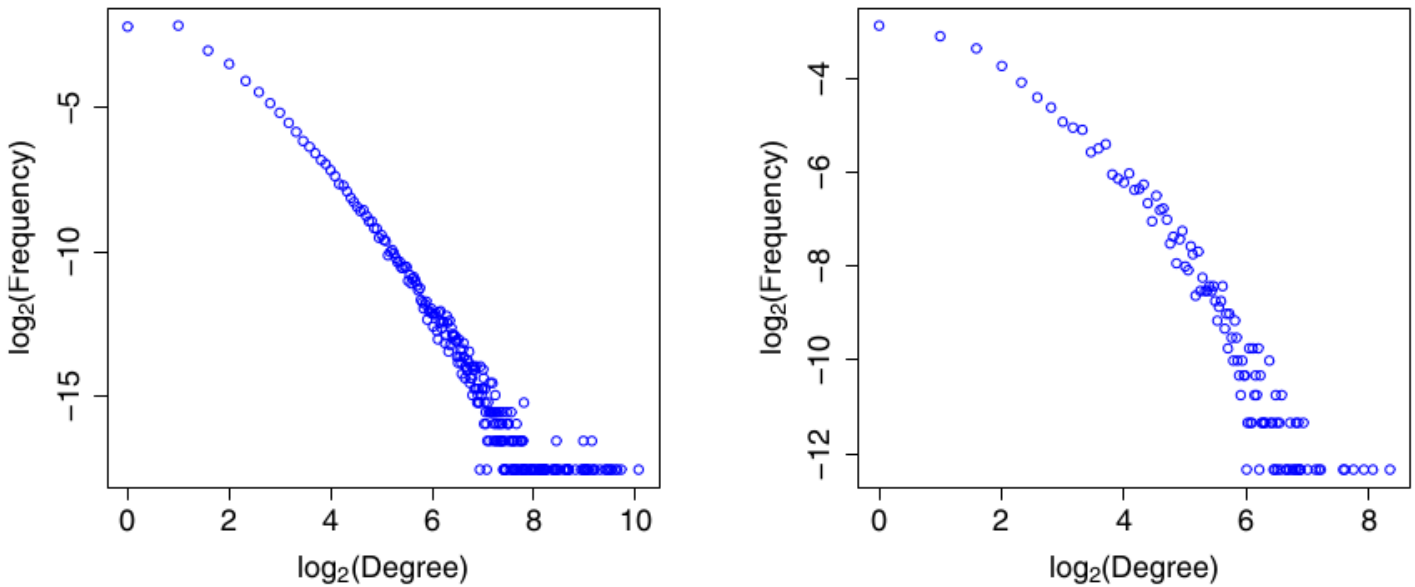
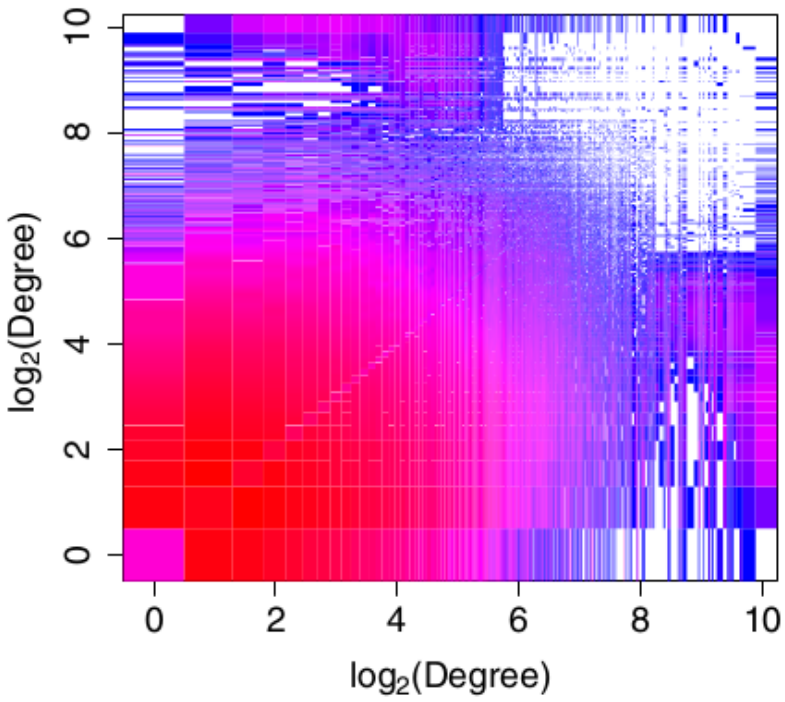
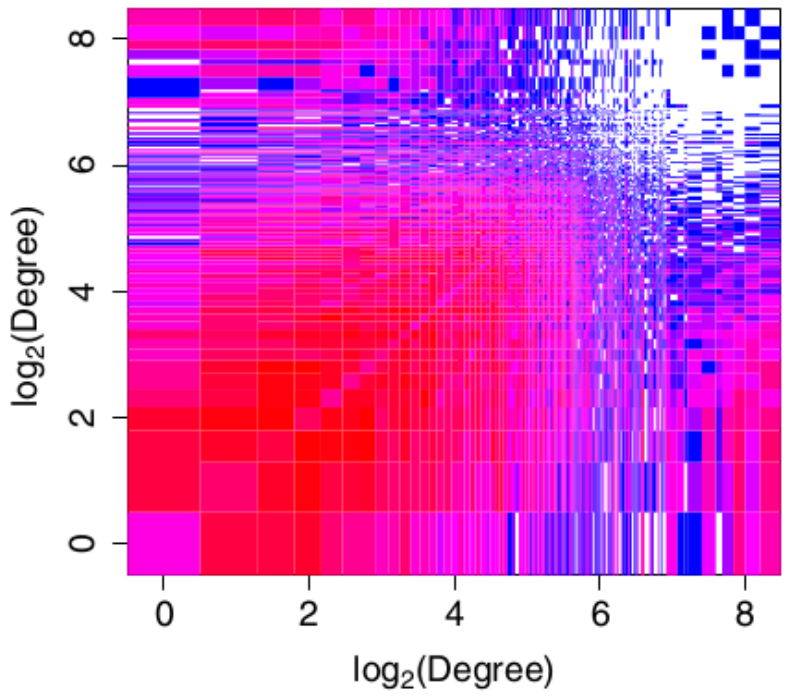


Fig. 4.1 Degree distributions. Left: the router-level Internet network graph described in Section 3.5.2. Right: the network of measured interactions among proteins in *S. cerevisiae* (yeast), as of January 2007. In each plot, both x- and y-axes are in base-2 logarithmic scale.

Degree correlation





Degree correlation

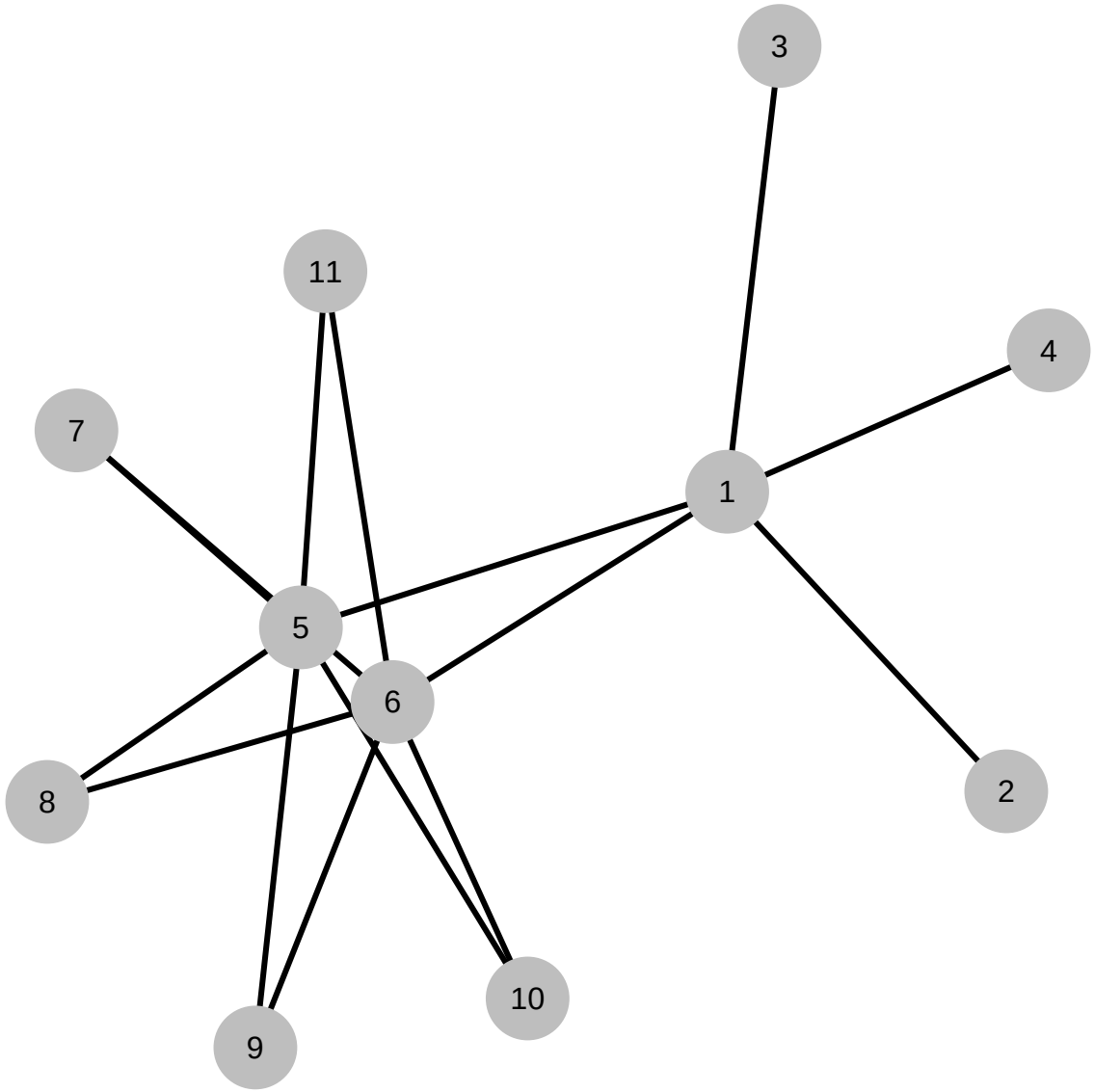
```
assortativity.degree(G)

## [1] -0.75
```

Closeness

Closeness $C_C(v)$ defines that a node is central if it is 'close' to other nodes. This can be computed by taking the inverse of the sum of all path lengths going from node v to all other nodes:

$$C_C(v) = \frac{1}{\sum_{i=1}^{|N|} \text{dist}(v, i)}$$

This is only an interesting measure for fully connected graphs or components.


3
11
4
7
1
5
6
8
2
10
9

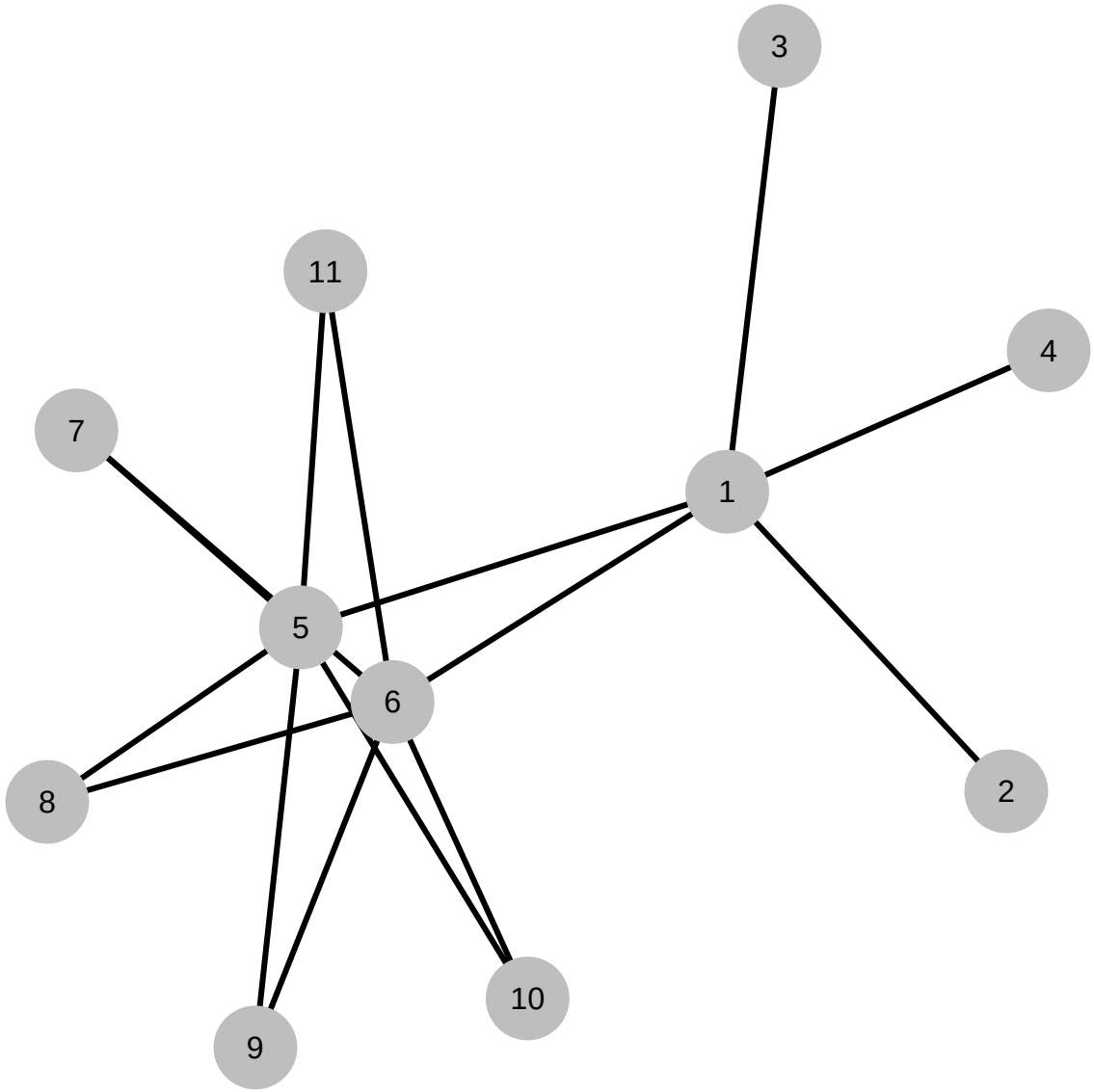
```
closeness(G)

##  [1] 0.06667 0.04167 0.04167 0.04167 0.07143 0.07143 0.0476
##  [8] 0.04762 0.04762 0.04762 0.04762
```

Betweenness

Betweenness of node v is defined as the sum of proportions of the number of shortest paths between all pairs of nodes that go through node v:

$$C_B(i) = \sum_{i \neq j \neq k \in V}^{n} \frac{\sigma(i, j \mid v)}{\sigma(i, j)}$$

Where $\sigma(i, j)$ is the total number of shortest paths between any two nodes and $\sigma(i, j \mid v)$ the amount of those paths that go through v.


3
11
4
7
1
5
6
8
2
10
9

```
betweenness(G)

##  [1] 24.1667  0.0000  0.0000  0.0000 15.0000 15.0000  0.166
##  [8]  0.1667  0.1667  0.1667  0.1667
```

Eigenvector centrality

Eigenvector centrality states that a node if central if its neighboors are central, and is recursive:

$$C_{EI}(v) = \alpha \sum_{\{u,v\} \in E} C_{EI}(u)$$

Since $\boldsymbol{A}$ contains only zeroes and ones we can write this as:

$$C_{EI}(v) = \alpha \sum_{i=1}^{|N|} a_{iv} C_{EI}(i)$$

Which we can write as a matrix equation:

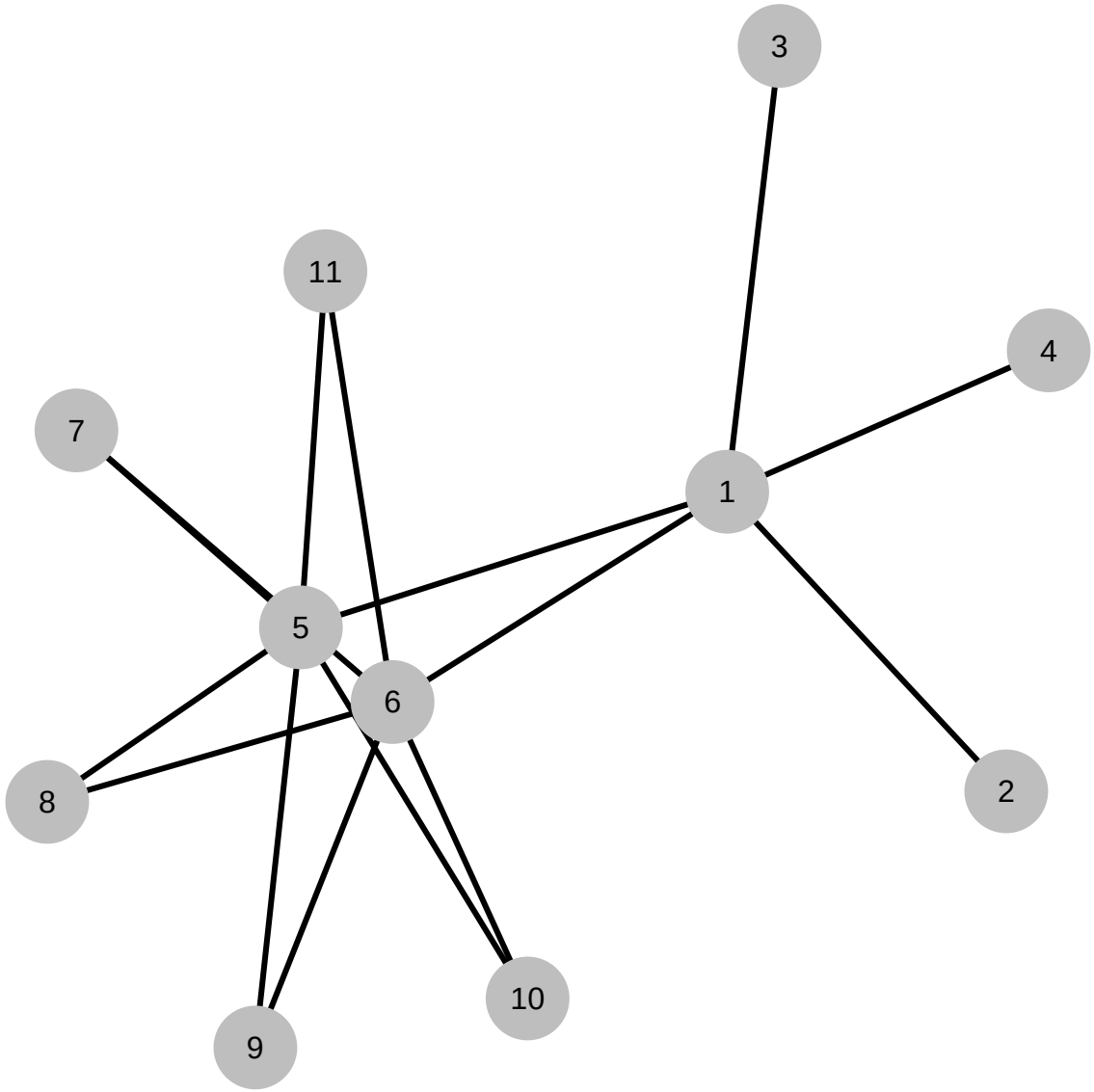
$$\boldsymbol{C_{EI}} = \alpha \boldsymbol{A C_{EI}}$$

Eigenvector centrality

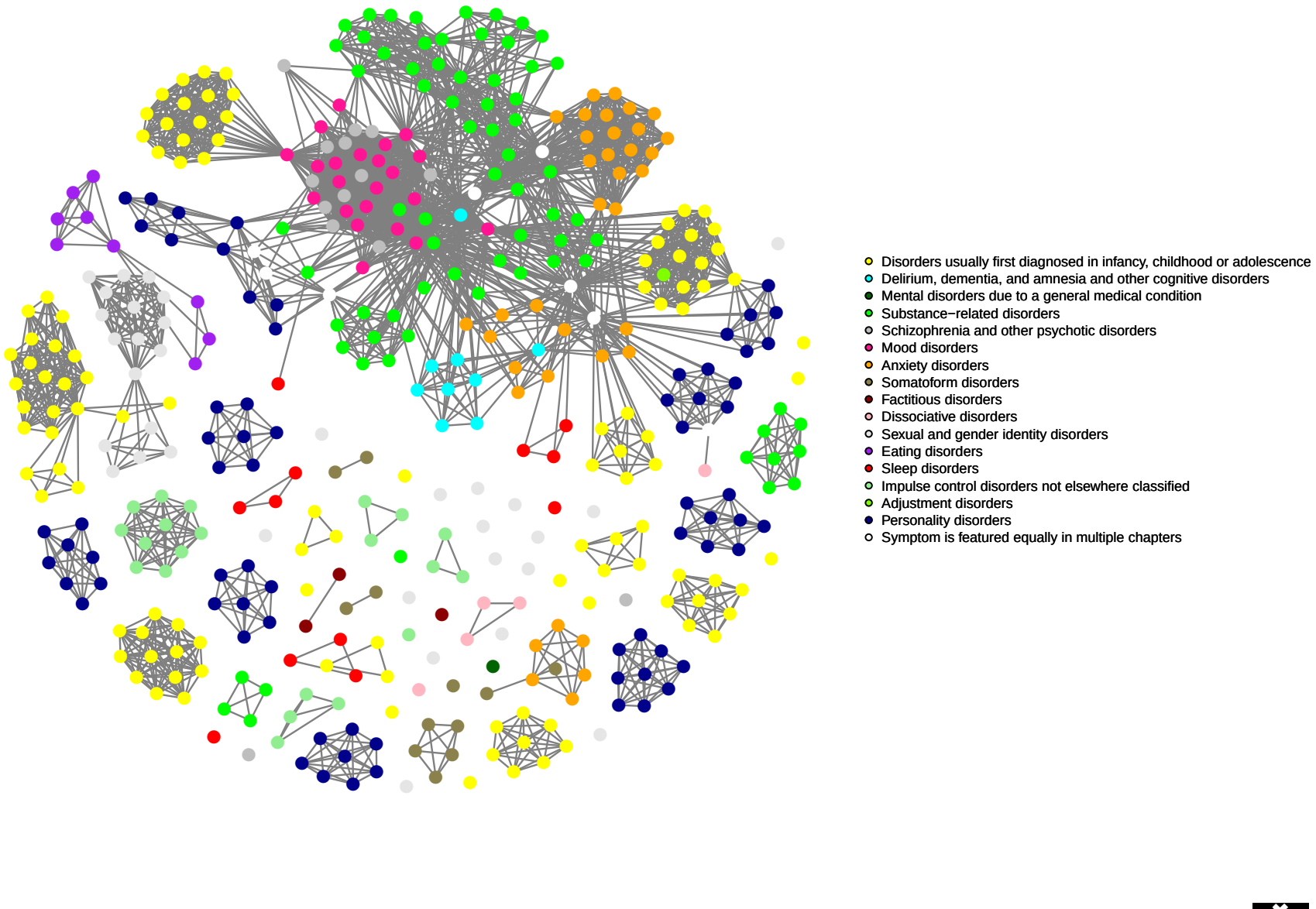
Rearanging this previous matrix eqation leads to a famillíar Eigenvalue problem:

$$\boldsymbol{AC_{EI}} = \frac{1}{\alpha}\boldsymbol{C_{EI}}$$

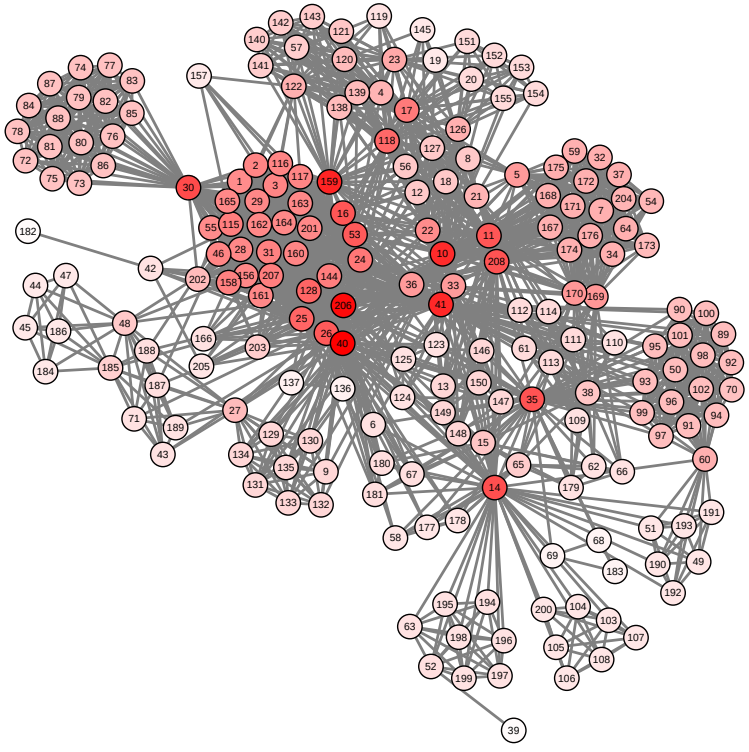
And shows $C_{EI}(v)$ to be the vth element of an eigenvector. Since centrality measures are positive, Perron-Frobenius theorem dictates that this should be the eigenvector corresponding to the *largest eigenvalue* $\lambda = \alpha^{-1}$.

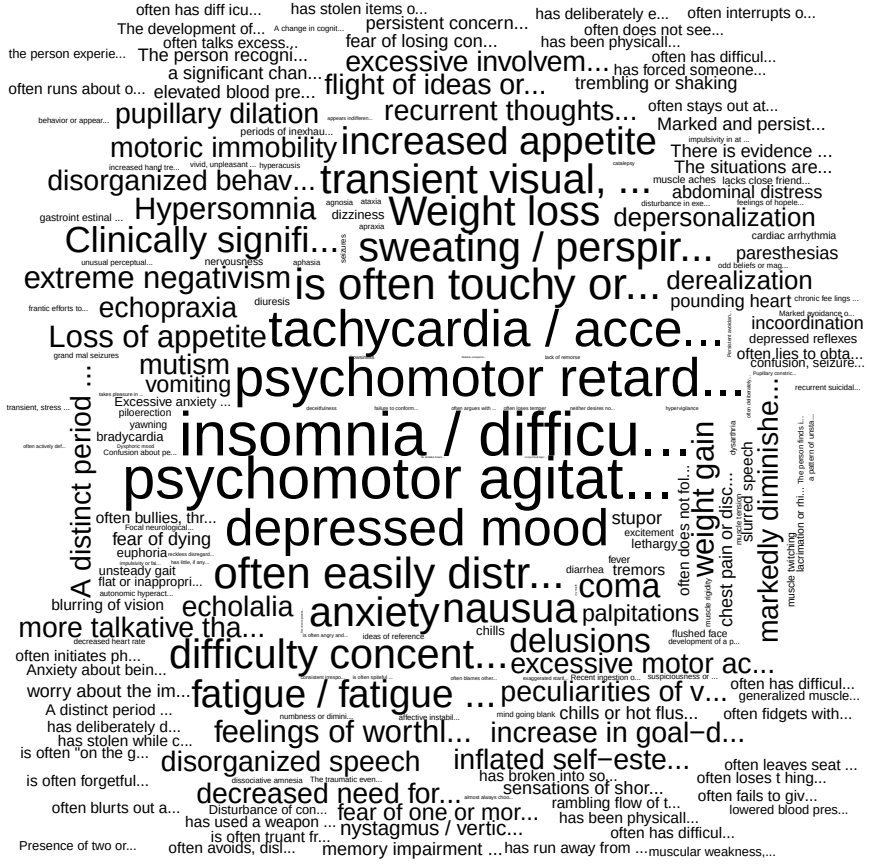

3
11
4
7
1
5
6
8
2
10
9

```
evcent(G)

## $vector
##  [1] 0.7384 0.2078 0.2078 0.2078 1.0000 1.0000 0.5629 0.562
##  [9] 0.5629 0.5629 0.5629
##
## $value
## [1] 3.553
##
## $options
## $options$bmat
## [1] "I"
##
## $options$n
## [1] 11
##
## $options$which
## [1] "LA"
##
## $options$nev
## [1] 1
##
## $options$tol
## [1] 0
```

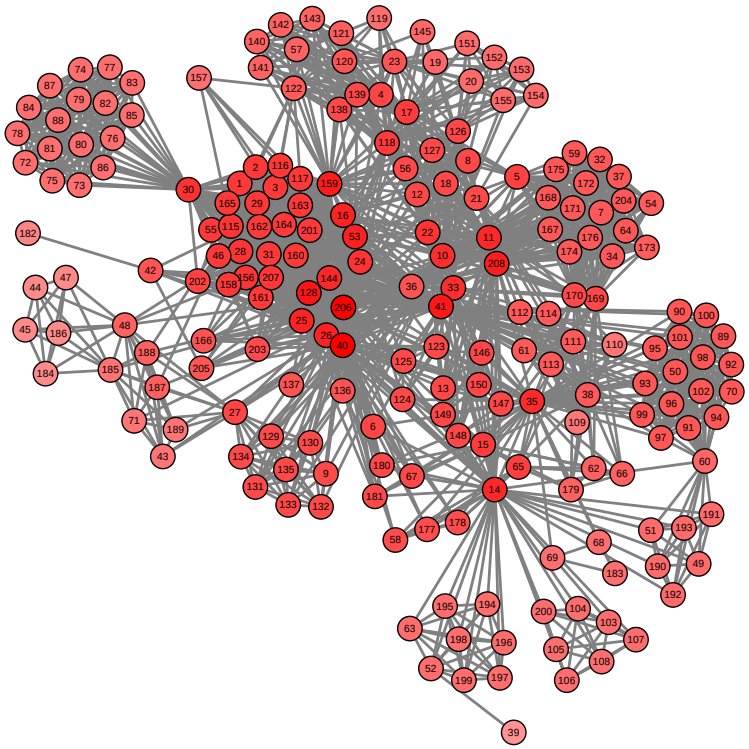

Disorders usually first diagnosed in infancy, childhood or adolescence
Delirium, dementia, and amnesia and other cognitive disorders
Mental disorders due to a general medical condition
Substance−related disorders
Schizophrenia and other psychotic disorders
Mood disorders
Anxiety disorders
Somatoform disorders
Factitious disorders
Dissociative disorders
Sexual and gender identity disorders
Eating disorders
Sleep disorders
Impulse control disorders not elsewhere classified
Adjustment disorders
Personality disorders
Symptom is featured equally in multiple chapters

Degree



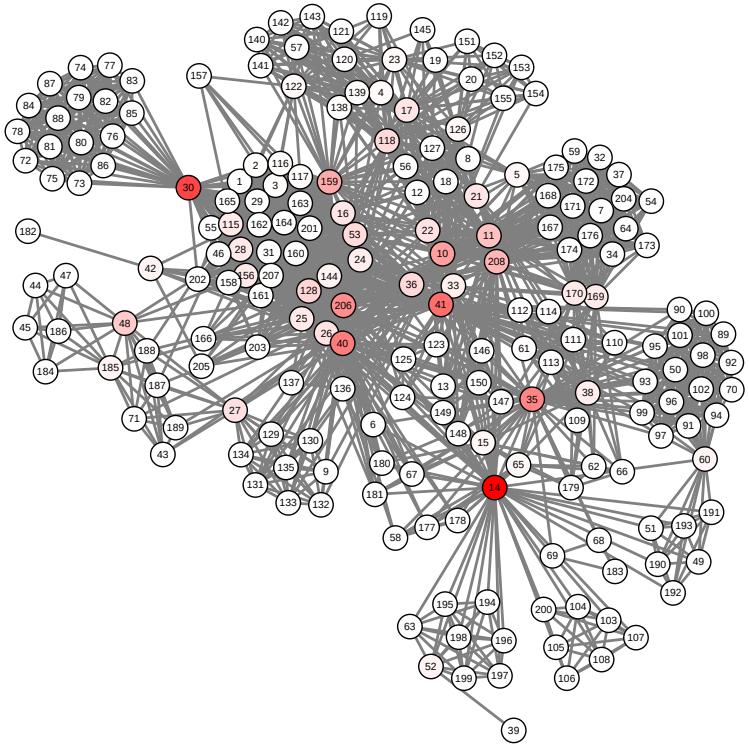

often has diff icu...
has stolen items o...
has deliberately e...
often interrupts o...
The development of...
persistent concern...
often does not see...
often talks excess...
fear of losing con...
has been physicall...
the person experie...
The person recogni...
excessive involvem...
often has difficul...
a significant chan...
has forced someone...
often runs about o...
elevated blood pre...
flight of ideas or...
trembling or shaking
pupillary dilation
recurrent thoughts...
often stays out at...
Marked and persist...
periods of inexhau...
motoric immobility
increased appetite
There is evidence ...
The situations are...
disorganized behav...
transient visual, ...
muscle aches
lacks close friend...
abdominal distress
Hypersomnia
Weight loss
depersonalization
gastroint estinal ...
dizziness
Clinically signifi...
sweating / perspir...
paresthesias
cardiac arrhythmia
nervousness
extreme negativism
is often touchy or...
derealization
pounding heart
echopraxia
tachycardia / acce...
incoordination
depressed reflexes
Loss of appetite
often lies to obta...
confusion, seizure...
mutism
vomiting
psychomotor retard...
recurrent suicidal...
Excessive anxiety ...
piloerection
yawning
bradycardia
insomnia / difficu...
A distinct period ...
psychomotor agitat...
weight gain
slurred speech
markedly diminishe...
often bullies, thr...
depressed mood
stupor
excitement
lethargy
fear of dying
euphoria
often easily distr...
diarrhea
tremors
coma
unsteady gait
flat or inappropri...
blurring of vision
echolalia
anxiety
nausua
palpitations
more talkative tha...
chills
delusions
chest pain or disc...
difficulty concent...
excessive motor ac...
often initiates ph...
Anxiety about bein...
worry about the im...
fatigue / fatigue ...
peculiarities of v...
often has difficul...
generalized muscle...
A distinct period ...
has deliberately d...
has stolen while c...
feelings of worthl...
increase in goal−d...
is often "on the g...
disorganized speech
inflated self−este...
often leaves seat ...
is often forgetful...
has broken into so...
often loses t hing...
decreased need for...
sensations of shor...
often fails to giv...
often blurts out a...
Disturbance of con...
fear of one or mor...
rambling flow of t...
lowered blood pres...
has used a weapon ...
nystagmus / vertic...
has been physicall...
is often truant fr...
often has difficul...
Presence of two or...
often avoids, disl...
memory impairment ...
has run away from ...
muscular weakness,...

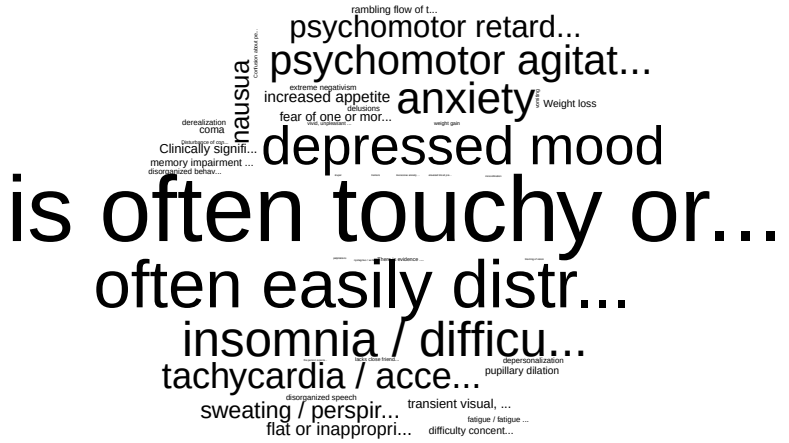
Closeness



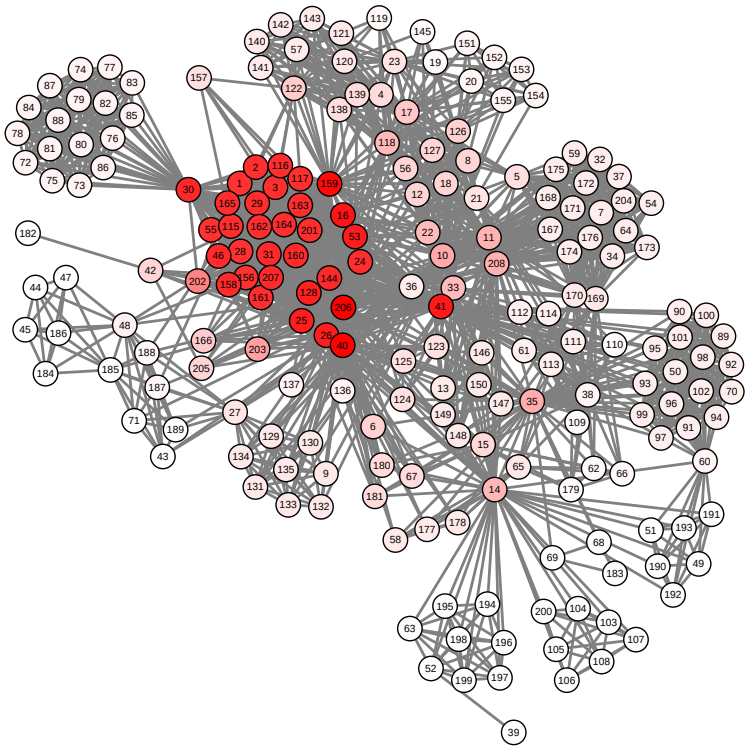
echopraxia
extreme negativism
palpitations
pupillary dilation
excitement
Hypersomnia
vomiting
yawning
ataxia
delusions
Weight loss
diarrhea
stupor
mutism
difficulty concent...
chills
weight gain
nausua
fever
tremors
depressed mood
anxiety
increased appetite
insomnia / difficu...
psychomotor agitat...
psychomotor retard...
transient visual, ...
fatigue / fatigue ...
sweating / perspir...
is often touchy or...
often easily distr...
disorganized speech
unexpected travel ...
incoordination
echolalia
coma
lethargy
tachycardia / acce...
disorganized behav...
muscular weakness,...
feelings of worthl...
flushed face
muscle aches

Betweenness





Eigenvector Centrality



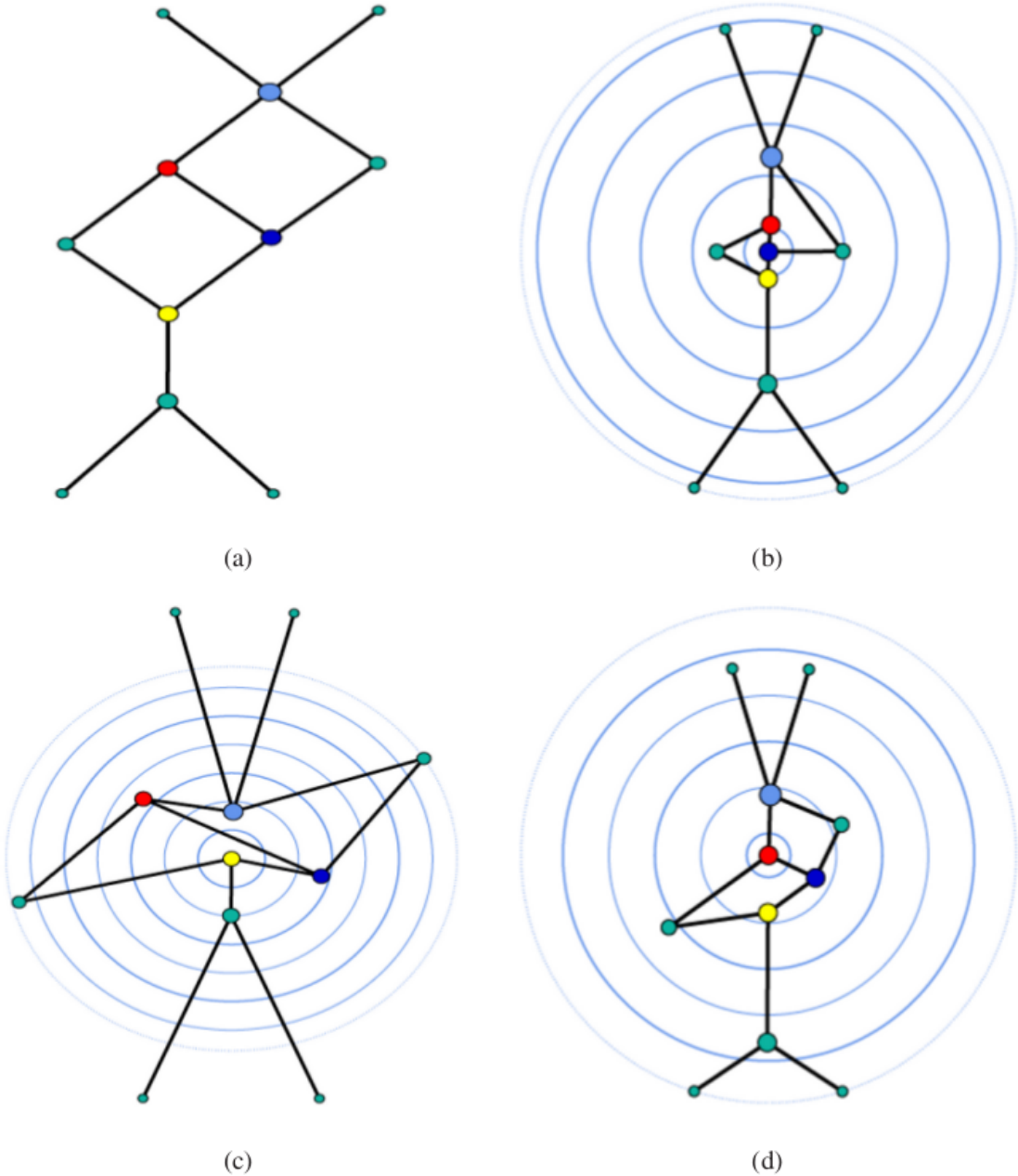
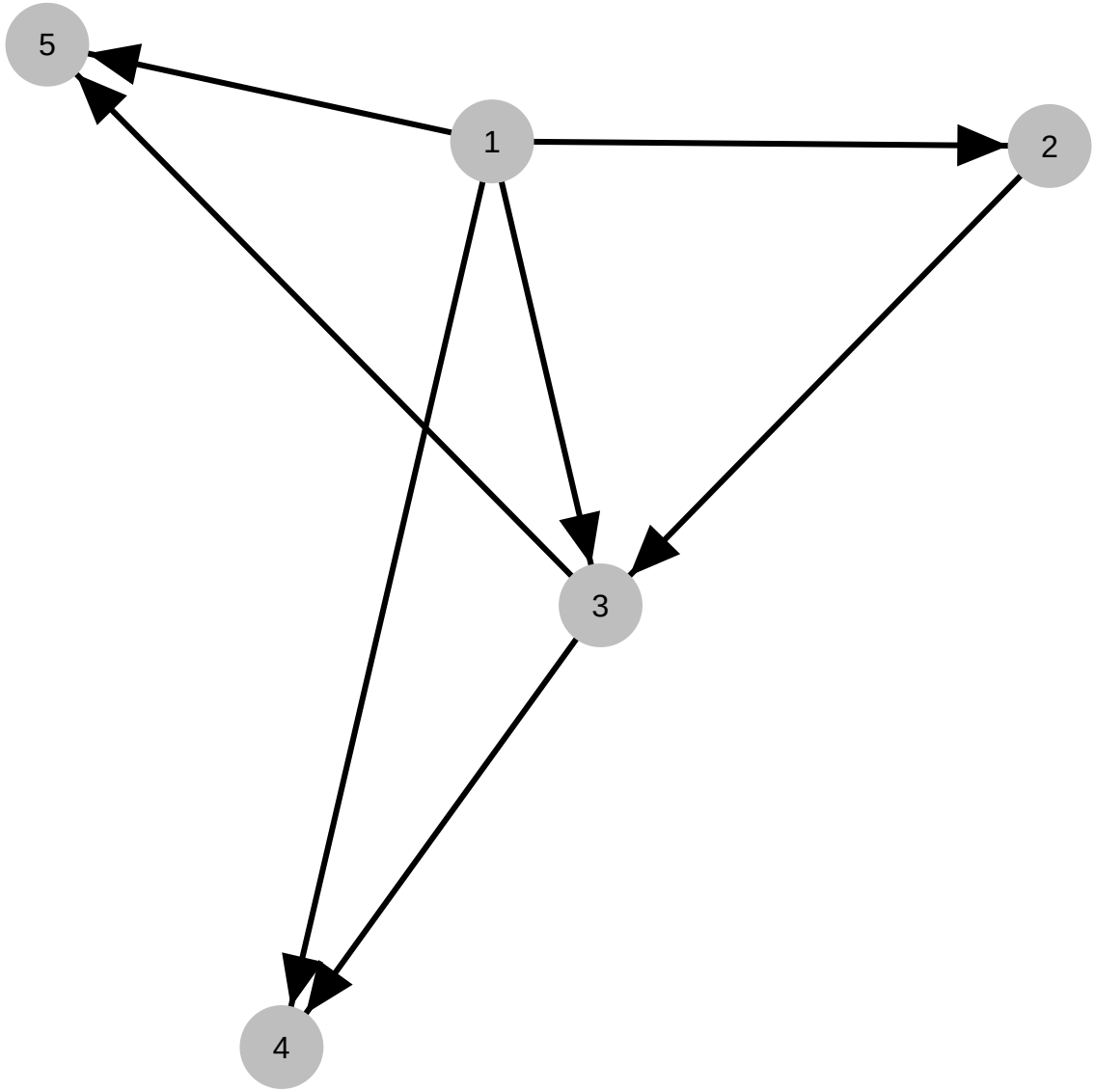
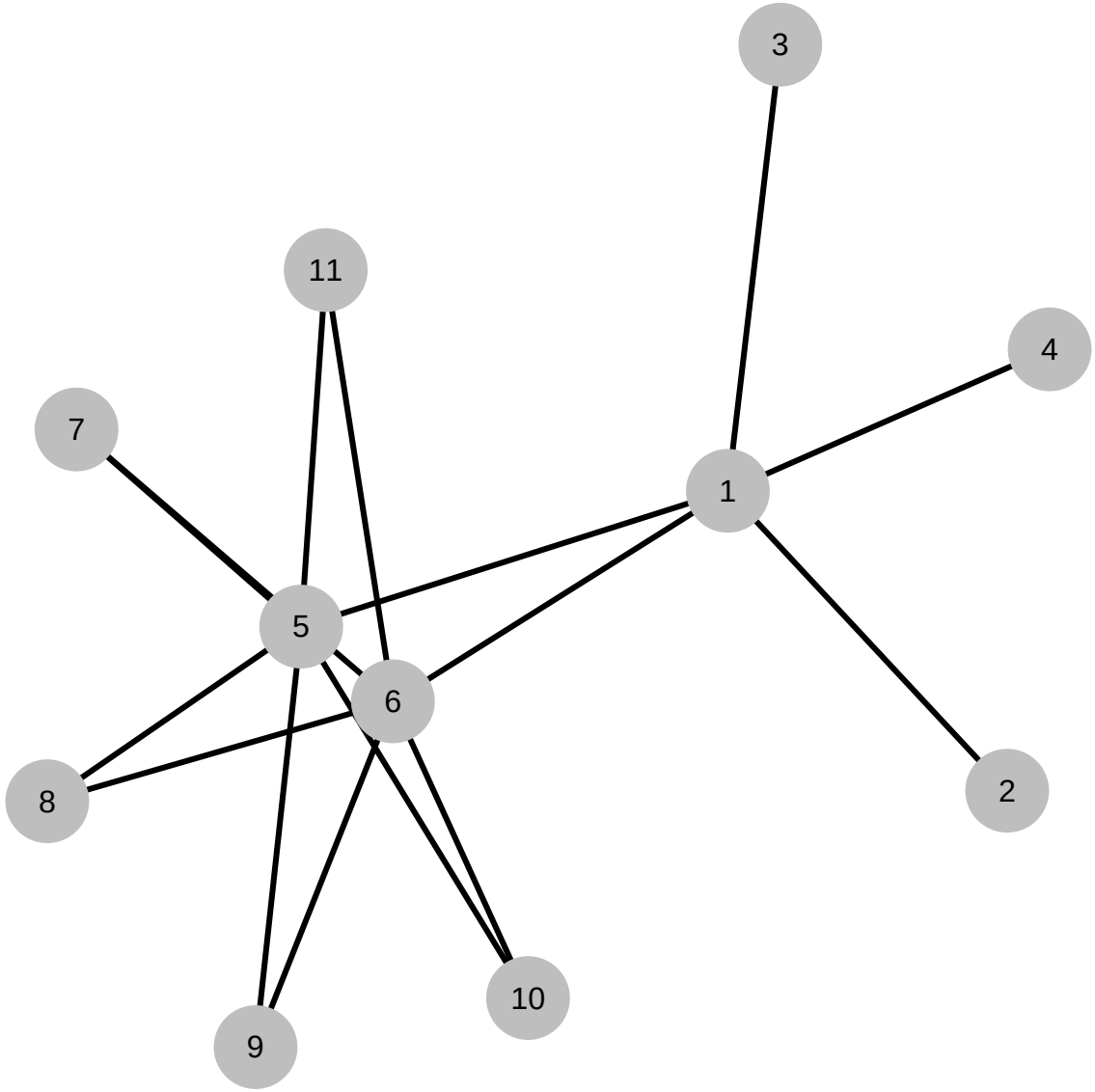


Fig. 4.4 Illustration of (b) closeness, (c) betweenness, and (d) eigenvector centrality measures on the graph in (a). Example and figures courtesy of Ulrik Brandes.

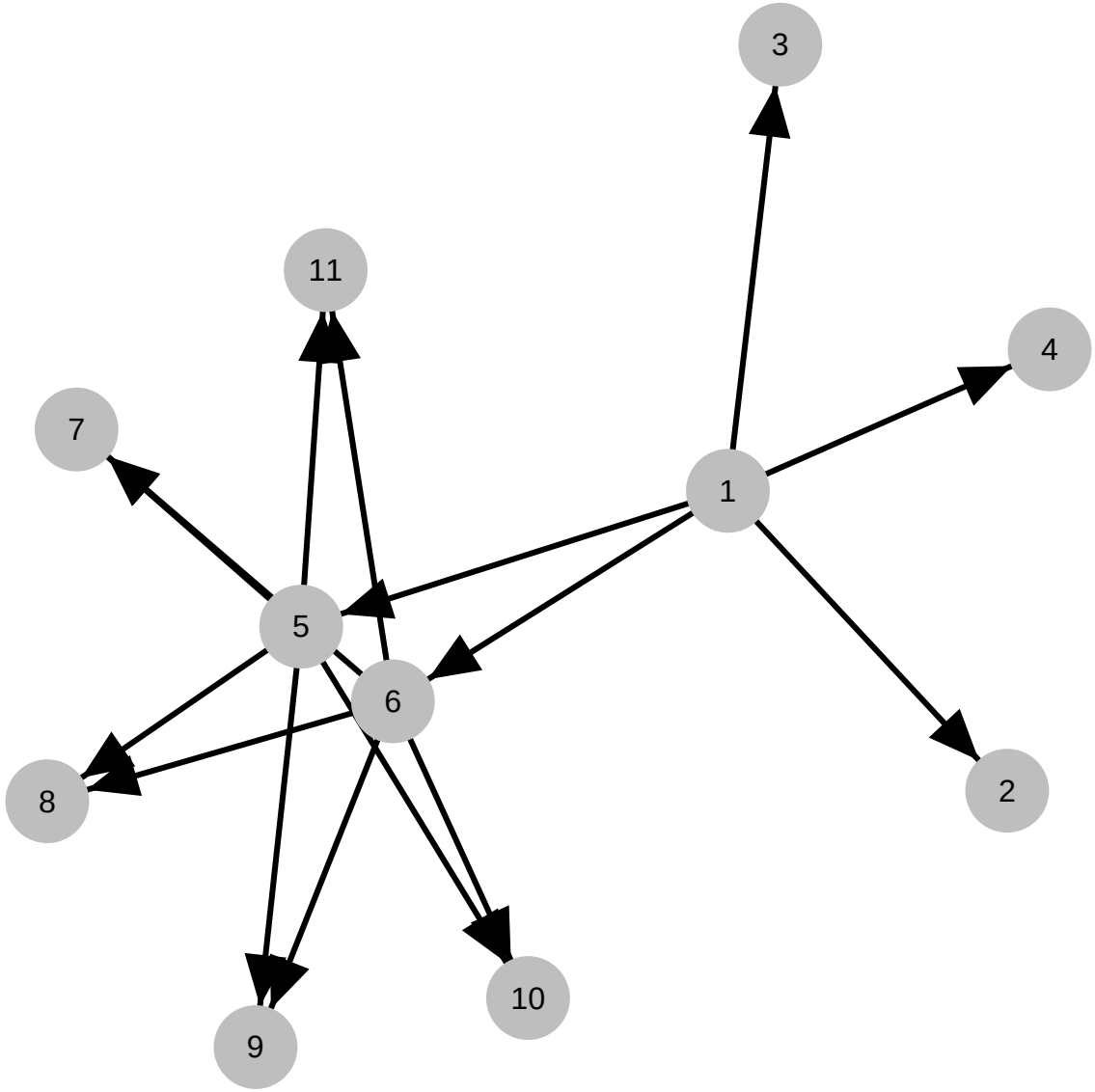
What is the most central node?



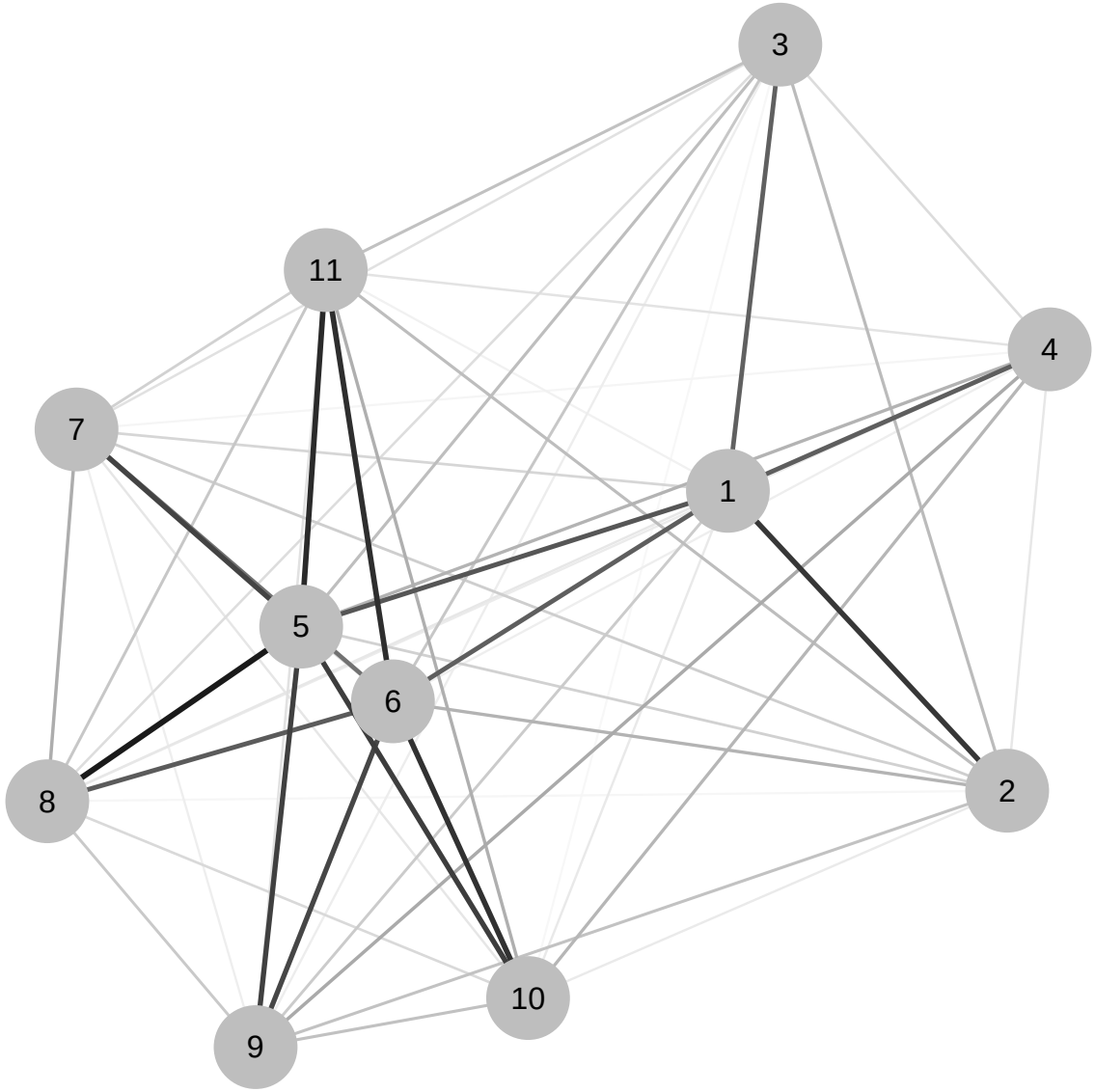
What is the most central node?



What is the most central node?



What is the most central node?



Weighted Graphs

A weighted graph is a network in which the strength of connections can vary. For a graph of n nodes the network structure is defined by the $n \times n$ adjacency matrix $\boldsymbol{A}$ such that:

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

Its weights are defined by the $n \times n$ weights matrix $\boldsymbol{W}$ such that:

$$w_{ij} \begin{cases} = 0 & \text{if } a_{ij} = 0 \\ \in \mathbb{R} & \text{otherwise} \end{cases}$$

The diagonal of both $\boldsymbol{A}$ and $\boldsymbol{W}$ is set to zero.

Weighted Graphs

A graph is *undirected* only if $\boldsymbol{A}$ and $\boldsymbol{W}$ are symmetric:

$$\boldsymbol{A} = \boldsymbol{A}^\top$$
$$\boldsymbol{W} = \boldsymbol{W}^\top$$

A graph is *unweighted* only if $\boldsymbol{A}$ is equal to $\boldsymbol{W}$ multiplied by some scalar c:

$$\boldsymbol{A} = c\boldsymbol{W}$$

Note that both these cases do not imply that a graph is undirected or unweighted.

Finally we define the *length* of an edge from node i to node j as the inverse of the absolute weight:

$$l_{ij} = \begin{cases} \frac{1}{|w_{ij}|} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

Because the denumerator is always positve, we will take the limit in the case of a weight of 0:

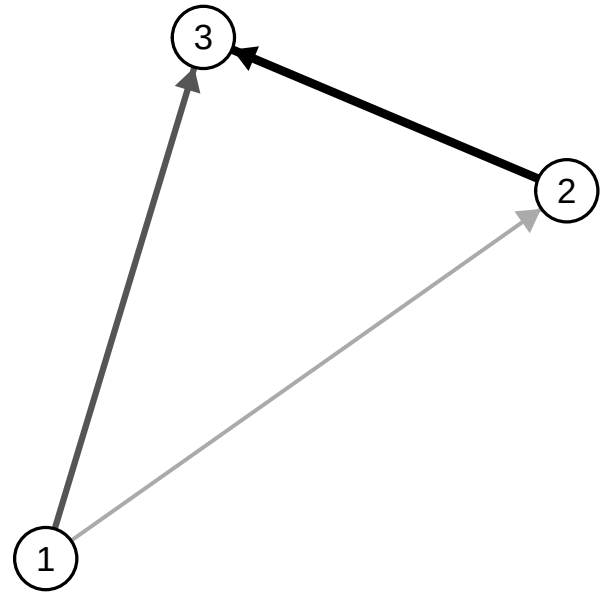
$$\lim_{w_{ij} \to 0} \frac{1}{|w_{ij}|} = \infty$$

Weighted graphs

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

$$\boldsymbol{W} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 1/2 \\ \infty & 0 & 1/3 \\ \infty & \infty & 0 \end{bmatrix}$$



Node centrality

- Node centrality is the identification of which nodes are the most central, and thus the most important
  - For example, if the most central node is infected with a contagious disease, the disease will spread the fastest to other nodes
  - In psychopathological modelling the most central node is the symptom which can most easilly affect other symptoms
- In unweighted graphs these are well defined
- In weighted graphs this is less the case. There is a lot of debate on whether the *structure* (adjacency matrix) or the *weights* (weights matrix) are the most important
- Clasically only the weights matrix is analyzed
- Opsahl, Agneessens, and Skvoretz (2010) proposes a set of centrality measures with a tuning parameter, α, to manually assign importance to both matrices
  - α of 0 only regards the structure, and α of 1 only regards the weights

Degree

Of node i the in-degree $k_i^{(in)}$ is the number of incomming edges and the out-degree $k_i^{(out)}$ the number of outgoing edges:

$$k_i^{(in)} = \sum_{j=1}^{n} a_{ji}$$

$$k_i^{(out)} = \sum_{j=1}^{n} a_{ij}$$

In weighted graphs often the node strengths $s_i^{(in)}$ and $s_i^{(out)}$ are used:

$$s_i^{(in)} = \sum_{j=1}^{n} w_{ji}$$

$$s_i^{(out)} = \sum_{j=1}^{n} w_{ij}$$

Degree

Opsahl et al. (2010) propose the following combination:

$$C_D^{(in)}(i) = k_i^{(in)} \times \left(\frac{s_i^{(in)}}{k_i^{(in)}} \right)^{\alpha}$$

$$C_D^{(out)}(i) = k_i^{(out)} \times \left(\frac{s_i^{(out)}}{k_i^{(out)}} \right)^{\alpha}$$

Note that for both:

$$C_D(i) = \begin{cases} k_i & \text{if } \alpha = 0 \\ s_i & \text{if } \alpha = 1 \end{cases}$$

The shortest path length between nodes i and j, $d(i,j)$ is defined in an unweighted graph as the minimum number of steps you need to take from node i to node j:

$$d(i,j) = \min\left(a_{ih} + \ldots + a_{hj}\right)$$

which can be obtained through Dijkstra's algorithm (Dijkstra, 1959) with weights fixed to 1. Simalary, for weighted graphs the shortest path length is defined as the minimum number of distance needed to cross on the graph to reach node i from node j:

$$d(i,j) = \min\left(\frac{1}{|w_{ih}|} + \ldots + \frac{1}{|w_{hj}|}\right) = \min\left(l_{ih} + \ldots + l_{hj}\right)$$

Opsahl et al. (2010) propose the following combination:

$$d(i,j) = \min\left(l_{ih}^{\alpha} + \ldots + l_{hj}^{\alpha}\right)$$

Which again generalizes to the unweighted form near $\alpha = 0$ and the weighted form if $\alpha = 1$.

Note that for any number y:

$$y^0 = 1$$

Including ∞. Therefore, setting $\alpha = 0$ will not give proper results.

Closeness is defined as the inverse of the total shortest distance from node *i* to all other nodes in the graph, which is only defined for connected clusters:

$$C_C(i) = \left[\sum_{j=1}^{n} d(i,j)\right]^{-1}$$

Here the α parameter is already used in computing the shortest path lengths.

Betweenness of node i is defined as the sum of proportions of the number of shortest paths between all pairs of nodes that go through node i:

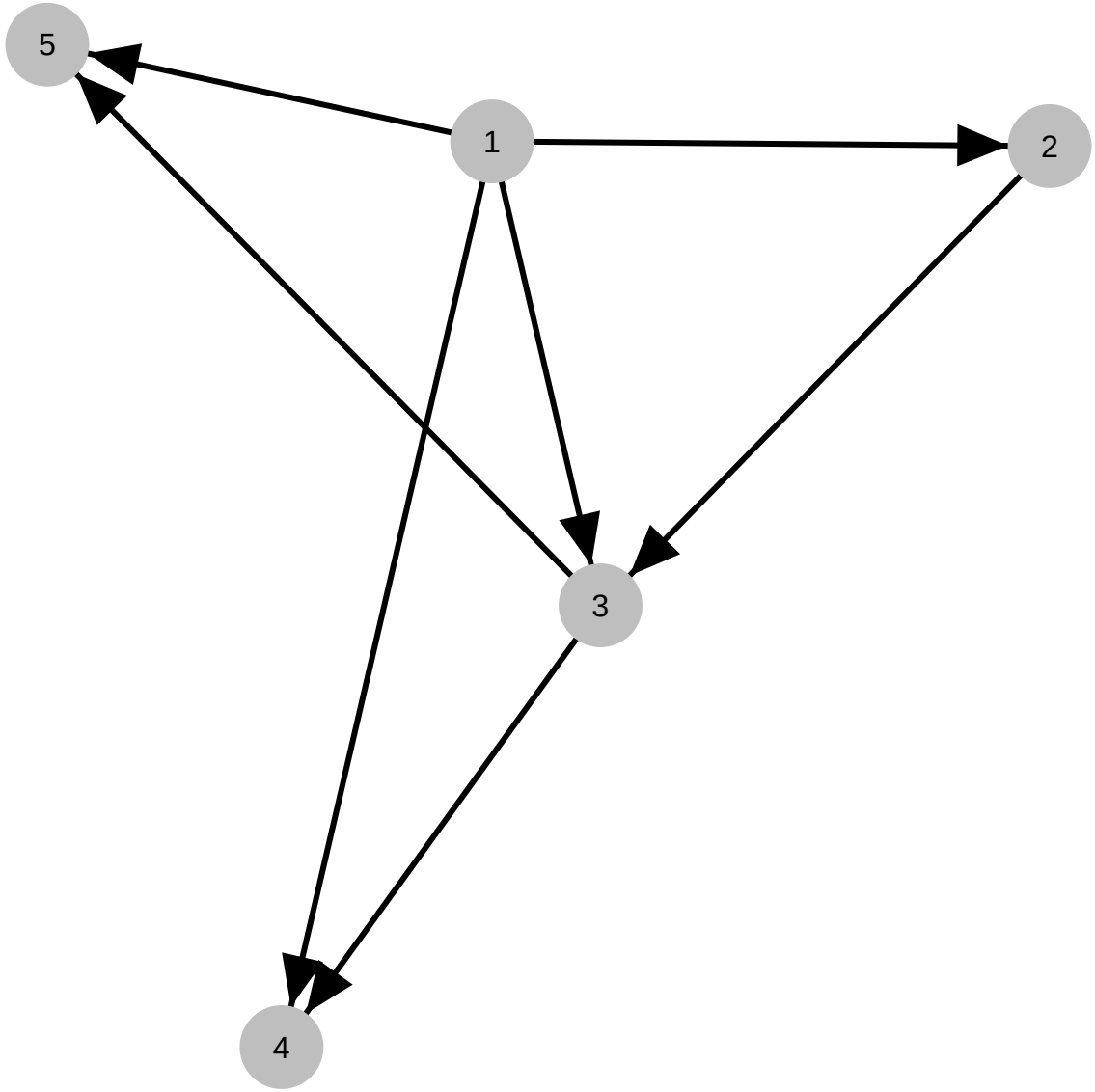
$$C_B(i) = \sum_{i \neq j \neq k}^{n} \frac{g_{jk}(i)}{g_{jk}}$$

Where g_{jk} is the total number of shortest paths between any two nodes and $g_{jk}(i)$ the amount of those paths that go through i.

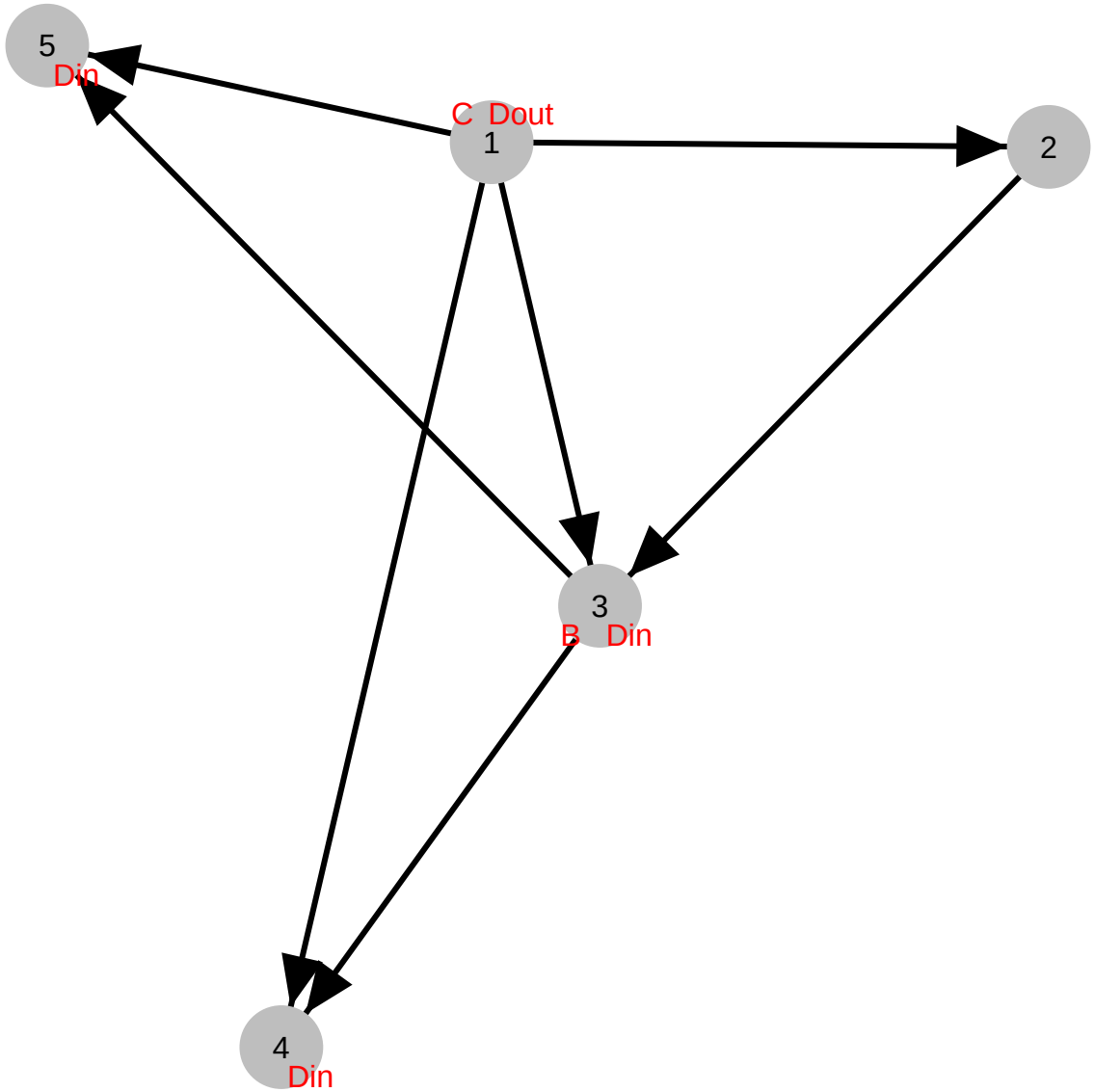
Centrality

Degree	How well connected is a node?
Closeness	How easy is it to reach all other nodes from a node?
Betweenness	How well does a node connect other nodes?

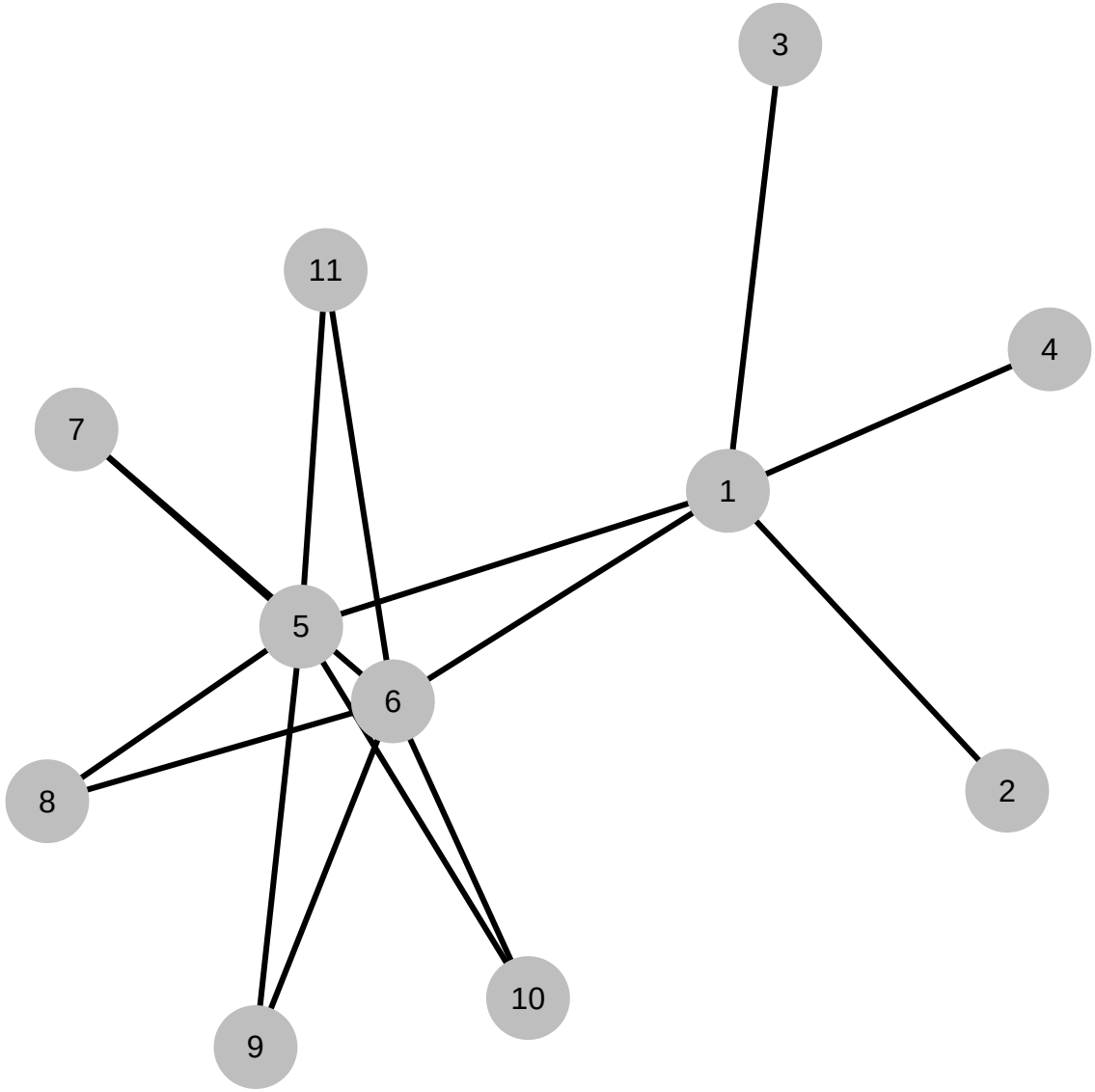
What is the most central node?



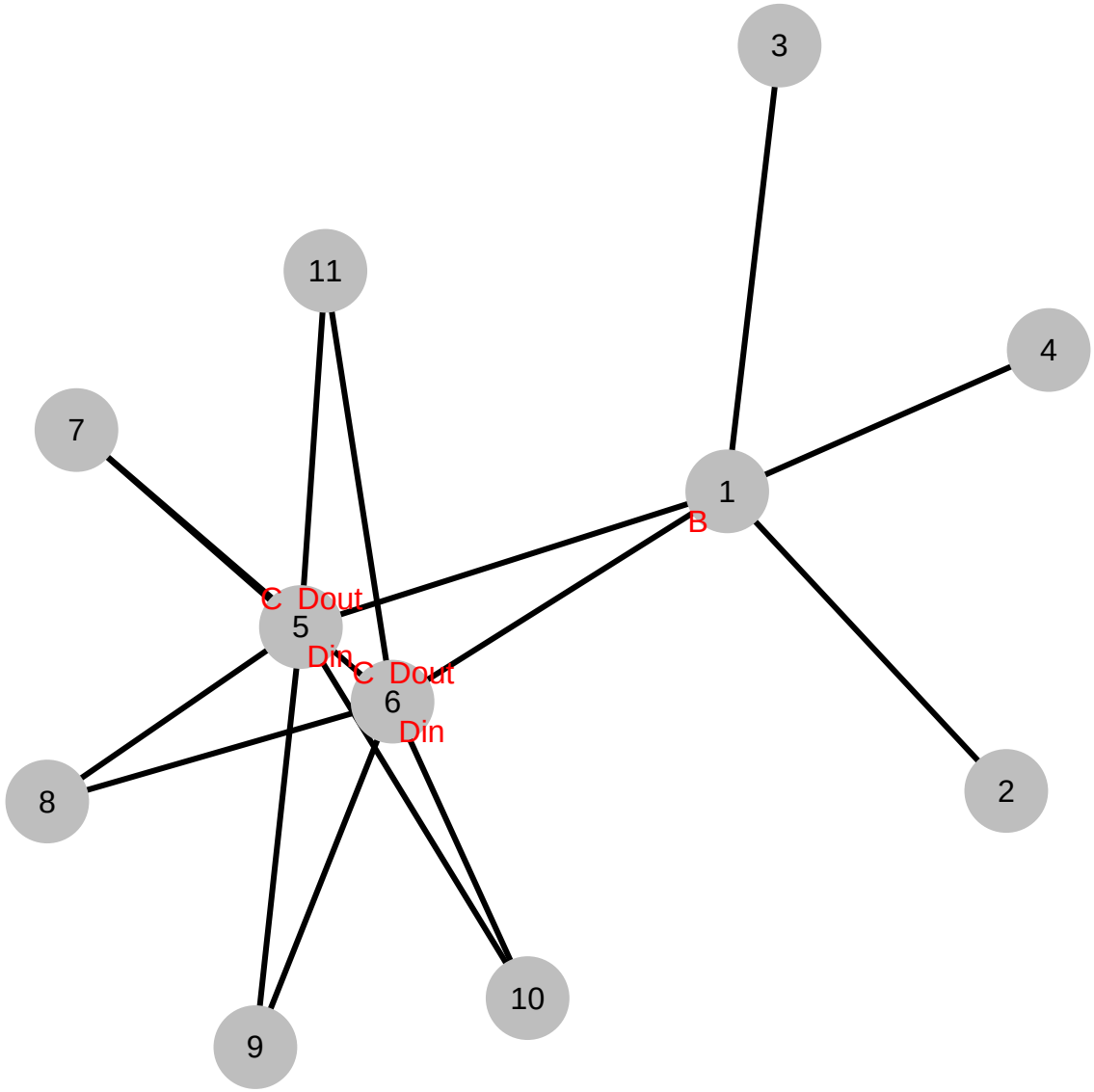
What is the most central node?



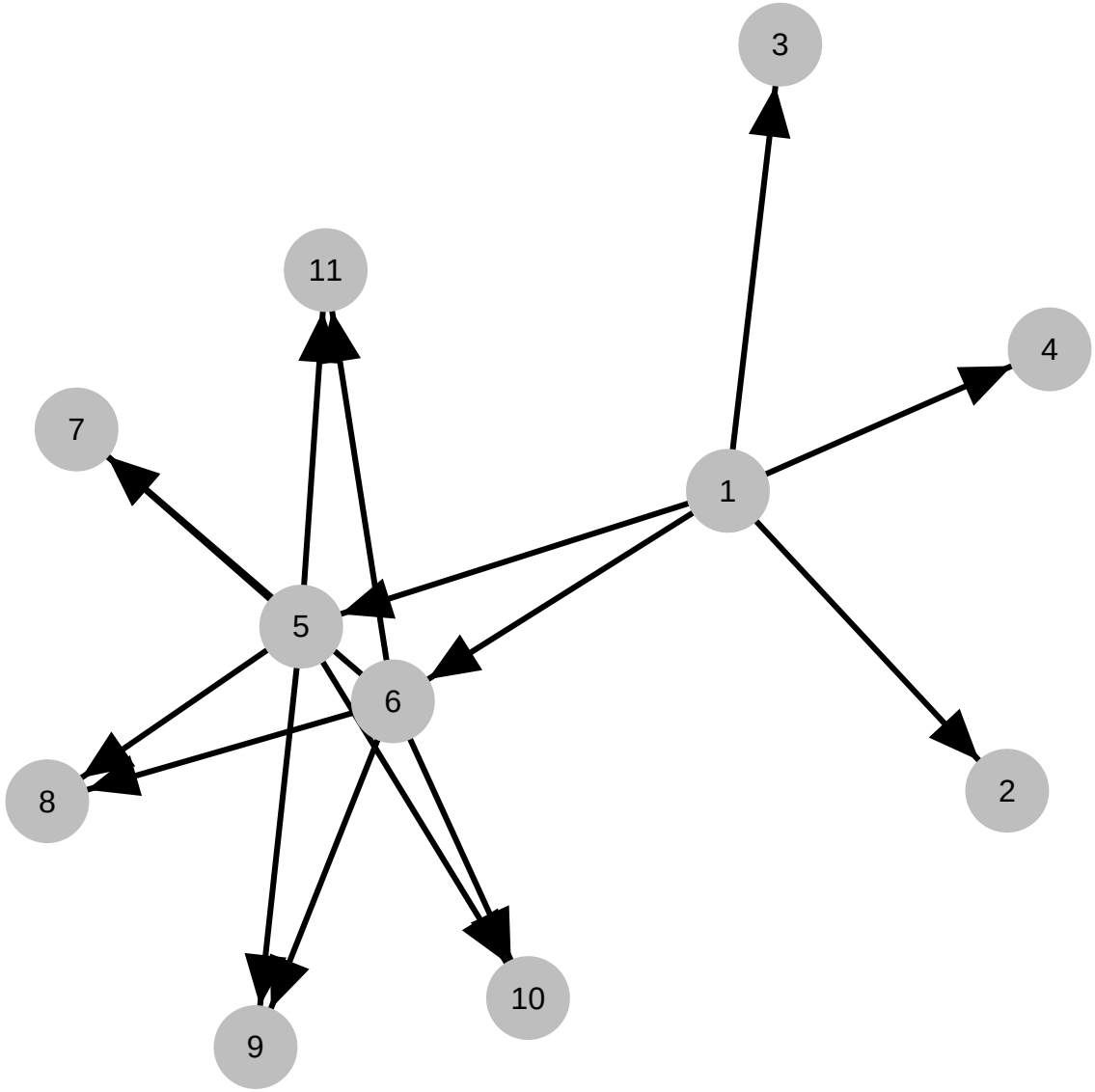
What is the most central node?



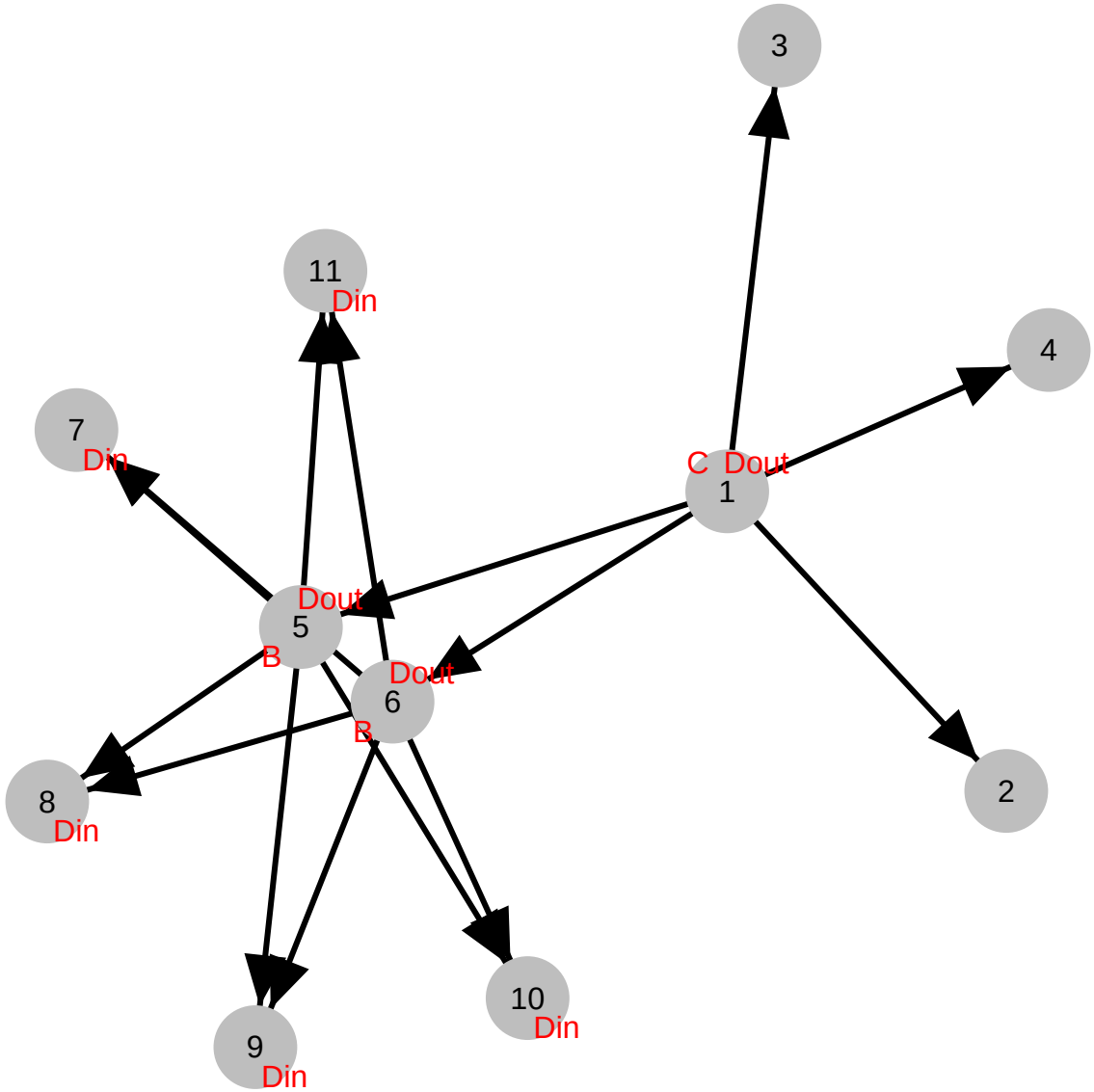
What is the most central node?



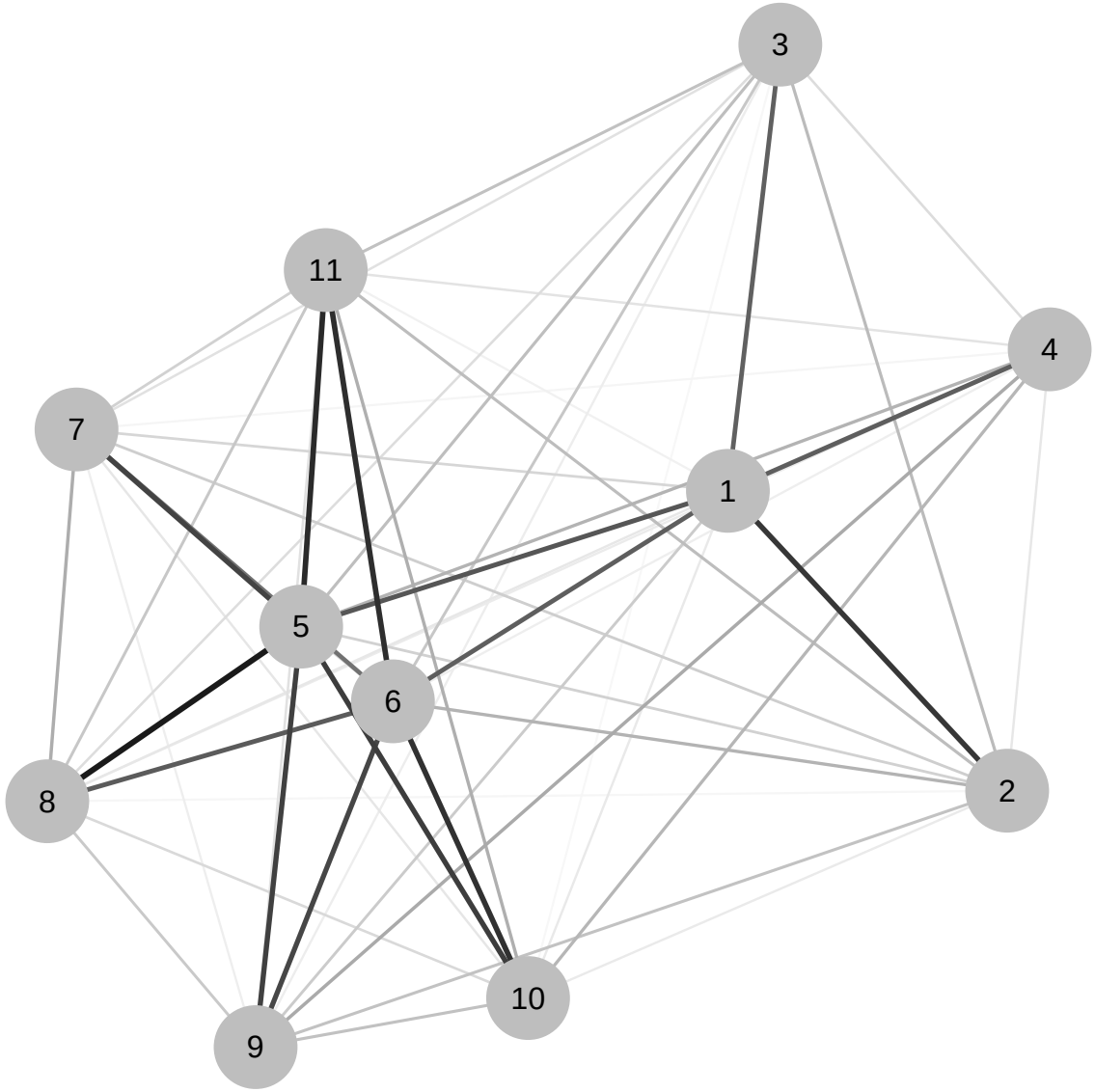
What is the most central node?



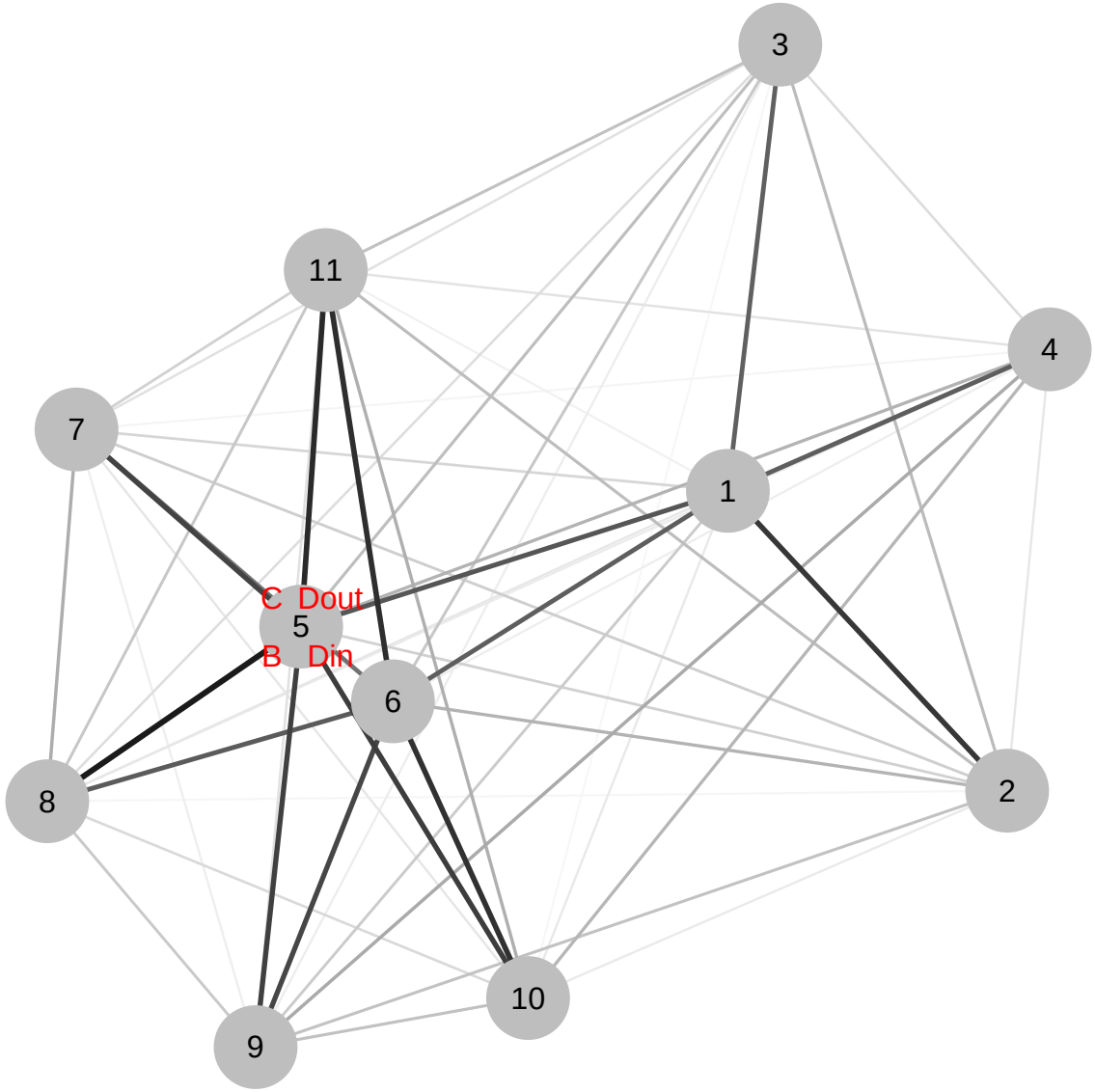
What is the most central node?



What is the most central node?



What is the most central node?



Computing Centrality Measures in **qgraph**

- In **qgraph** these centrality measures are included in the function `centrality` (see `?centrality`)
- This uses the output of `qgraph()` as input
  - Note that this function automatically removes diagonal elements of the weights matrix
- The default of the α parameter is 1, which corresponds to the classical interpretation of weighted centrality measures.

Computing Centrality Measures in **qgraph**

In the current version of qgraph (1.0.1) the function will fail if $\alpha = 0$. This will be fixed in the next version. For now, set α arbitrarily low to circomvent this.

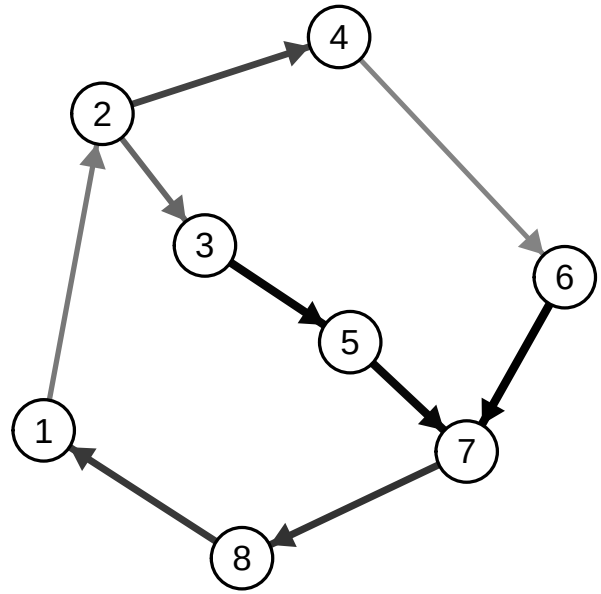
Return Values of centrality function

List with following elements:

OutDegree	A vector containing the outward degree of each node.
InDegree	A vector containing the inward degree of each node.
Closeness	A vector containing the closeness of each node.
Betweenness	A vector containing the betweenness of each node
ShortestPathLengths	A matrix containing the shortest path lengths of each pairs of nodes. These path lenghts are based on the inverse of the edge weights raised to the power lambda.
ShortestPaths	A matrix of lists containing all shortest path lengths between all pairs of nodes. Use double square brackets to index. E.g., if the list is called 'res', res$ShortestPaths[[i,j]] gives a list containing all shortest paths between node i and j.

Example ($\alpha = 1$)

```
> set.seed(1)
> E <- data.frame(
+   from = c(1,2,2,3,4,5,6,7
+   to = c(2,3,4,5,6,7,7,8,1
+   weight = runif(9,1,3))
> Q <- qgraph(E,
+          gray=TRUE,
+          asize=0.3)
> Cent <- centrality(Q)
```



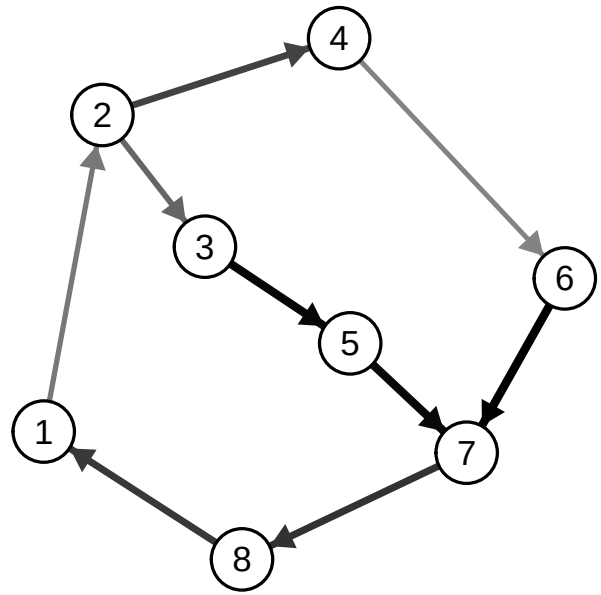
Centrality measures

```
> as.data.frame(Cent[1:4])
```

	OutDegree	InDegree	Closeness	Betweenness
1	1.531017	2.258228	0.09327425	25
2	3.889955	1.531017	0.12035952	25
3	2.816416	1.744248	0.08223756	12
4	1.403364	2.145707	0.06920048	6
5	2.796779	2.816416	0.08242626	12
6	2.889351	1.403364	0.08473412	6
7	2.321596	5.686130	0.08275217	25
8	2.258228	2.321596	0.08732845	25

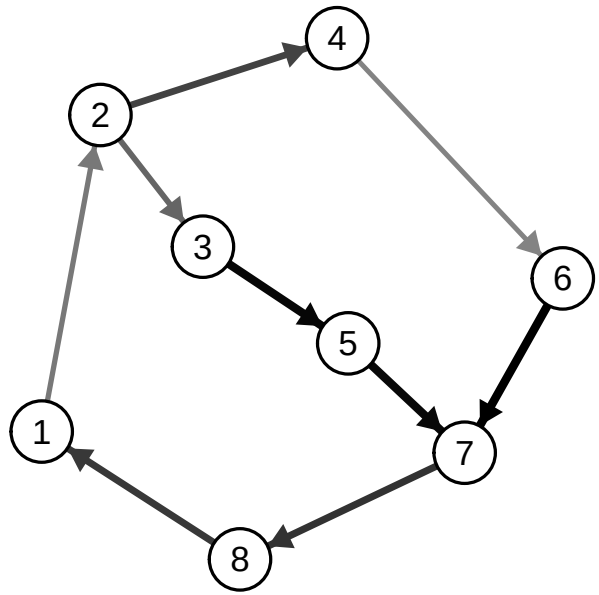
Shortest path-Length between 1 and 8

```
> Cent$ShortestPathLengths[1
[1] 2.369827
```



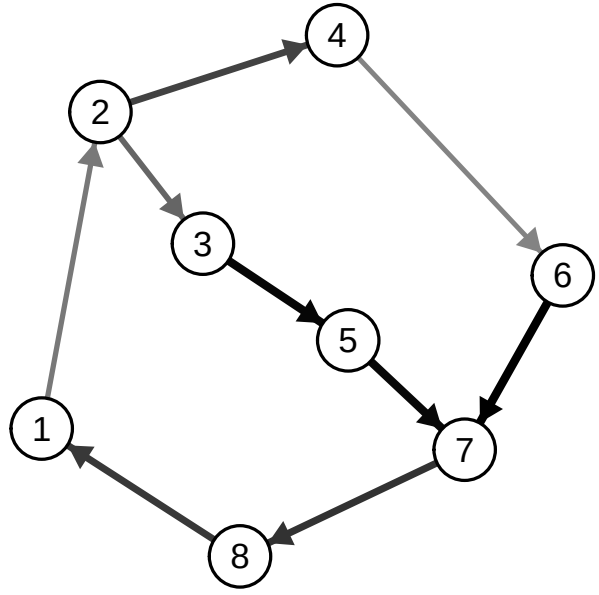
Shortest paths between 1 and 8

```
> Cent$ShortestPaths[[1,8]]

[[1]]
[1] 1 2 3 5 7 8
```



Example ($\alpha \approx 0$)

```
> Q <- qgraph(E,
+           gray=TRUE,
+           asize=0.3)
> Cent <- centrality(Q,1e-16
```

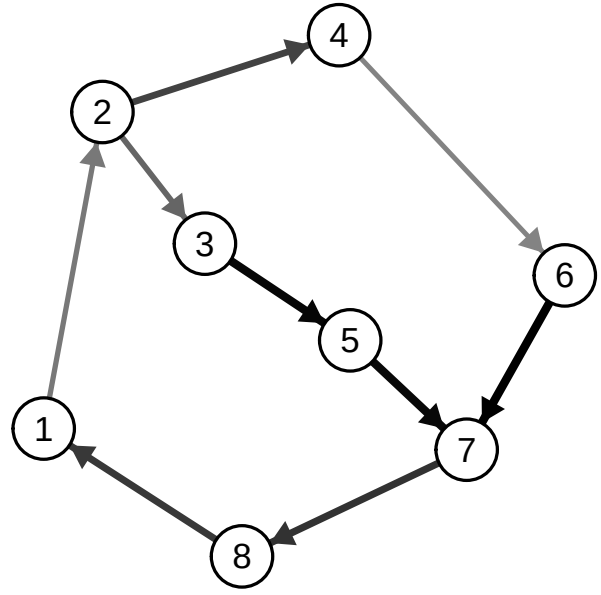


Centrality measures

```
> as.data.frame(Cent[1:4])
  OutDegree InDegree  Closeness Betweenness
1         1        1 0.05000000          25
2         2        1 0.05555556          25
3         1        1 0.03571429           9
4         1        1 0.03571429           9
5         1        1 0.03846154           9
6         1        1 0.03846154           9
7         1        2 0.04166667          25
8         1        1 0.04545455          25
```

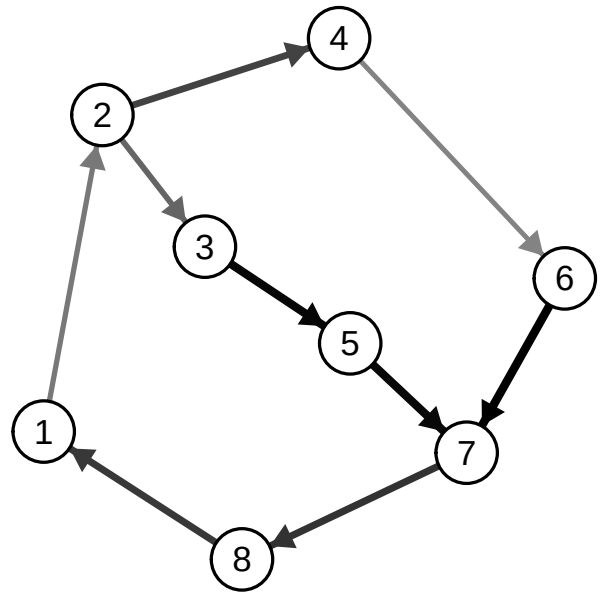
Shortest path-Length between 1 and 8

```
> Cent$ShortestPathLengths[1
[1] 5
```



Shortest paths between 1 and 8

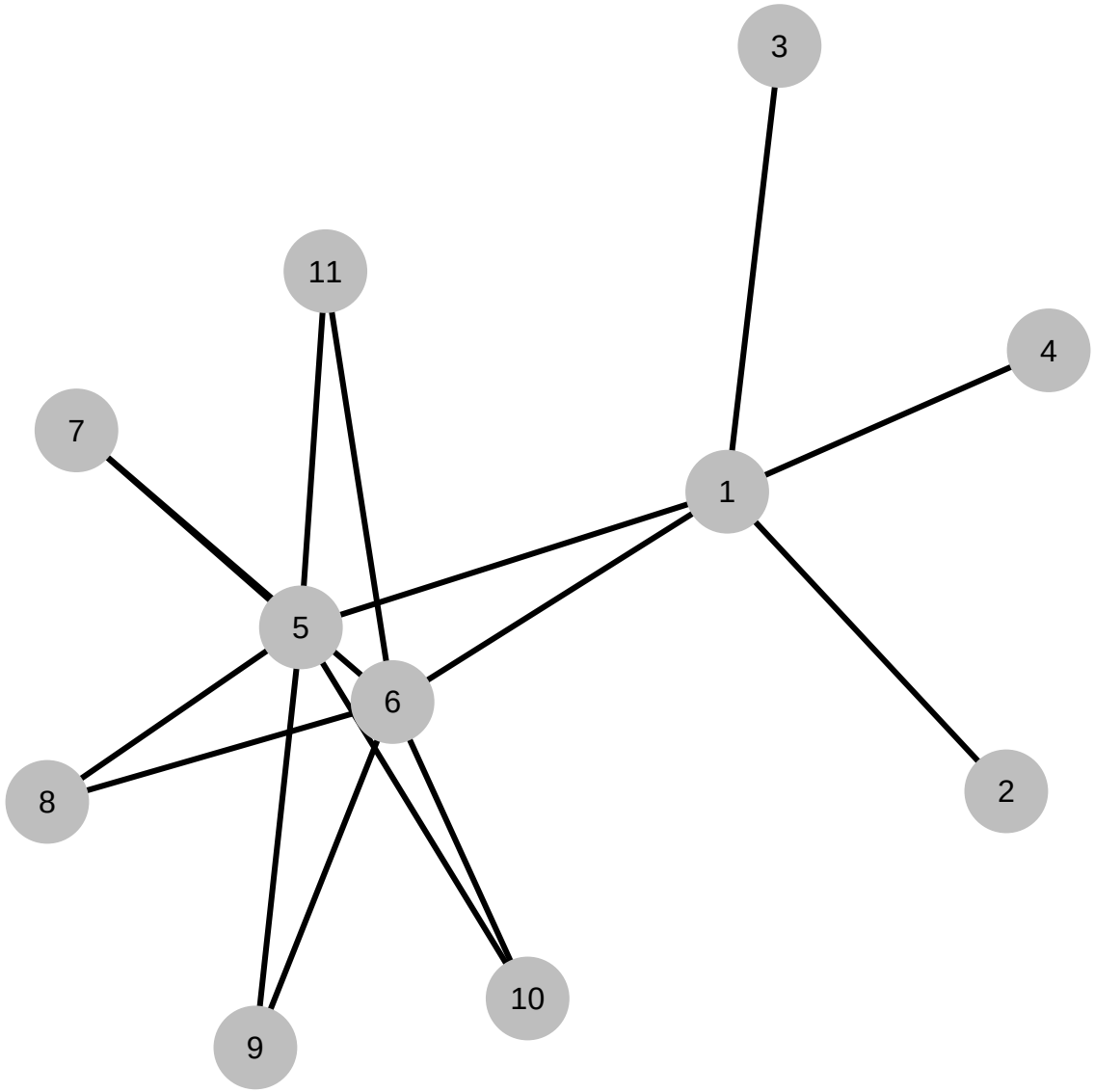
```
> Cent$ShortestPaths[[1,8]]

[[1]]
[1] 1 2 3 5 7 8

[[2]]
[1] 1 2 4 6 7 8
```



Connectivity and Clustering

The *diameter* of a graph is its longest shortest path length:

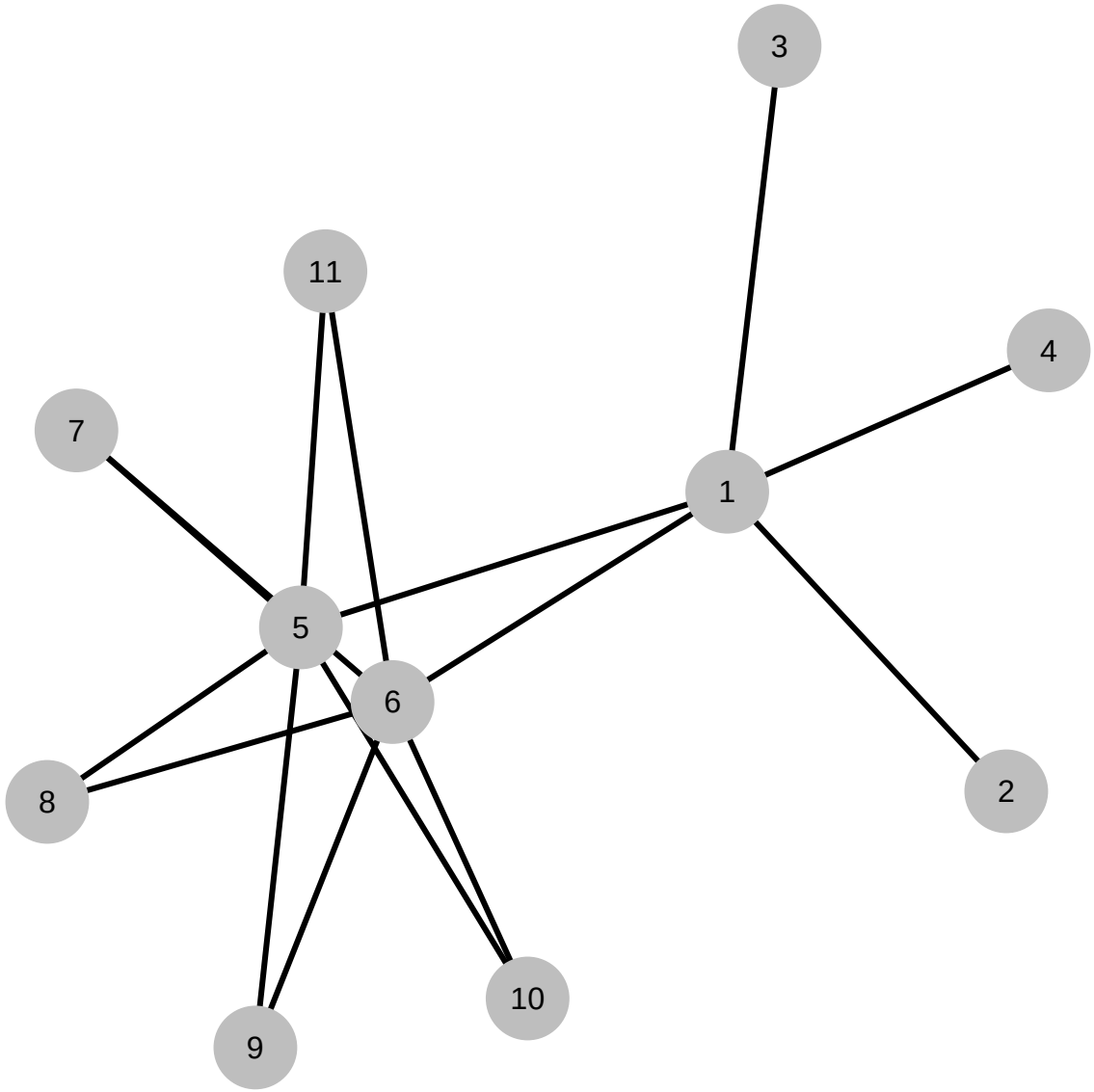
$$\text{diameter}(G) = \max(\text{dist}(u, v))$$


3
11
4
7
1
5
6
8
2
10
9

```r
diameter(G)

## [1] 3
```

The *density* of a graph is the proportion of the present number of edges to the total possible amount of edges:

$$\mathrm{den}(G) = \frac{|E|}{|N|\,(|N|-1)/2}$$


3
11
4
7
1
5
6
8
2
10
9

```r
graph.density(G)

## [1] 0.2727
```

Clustering

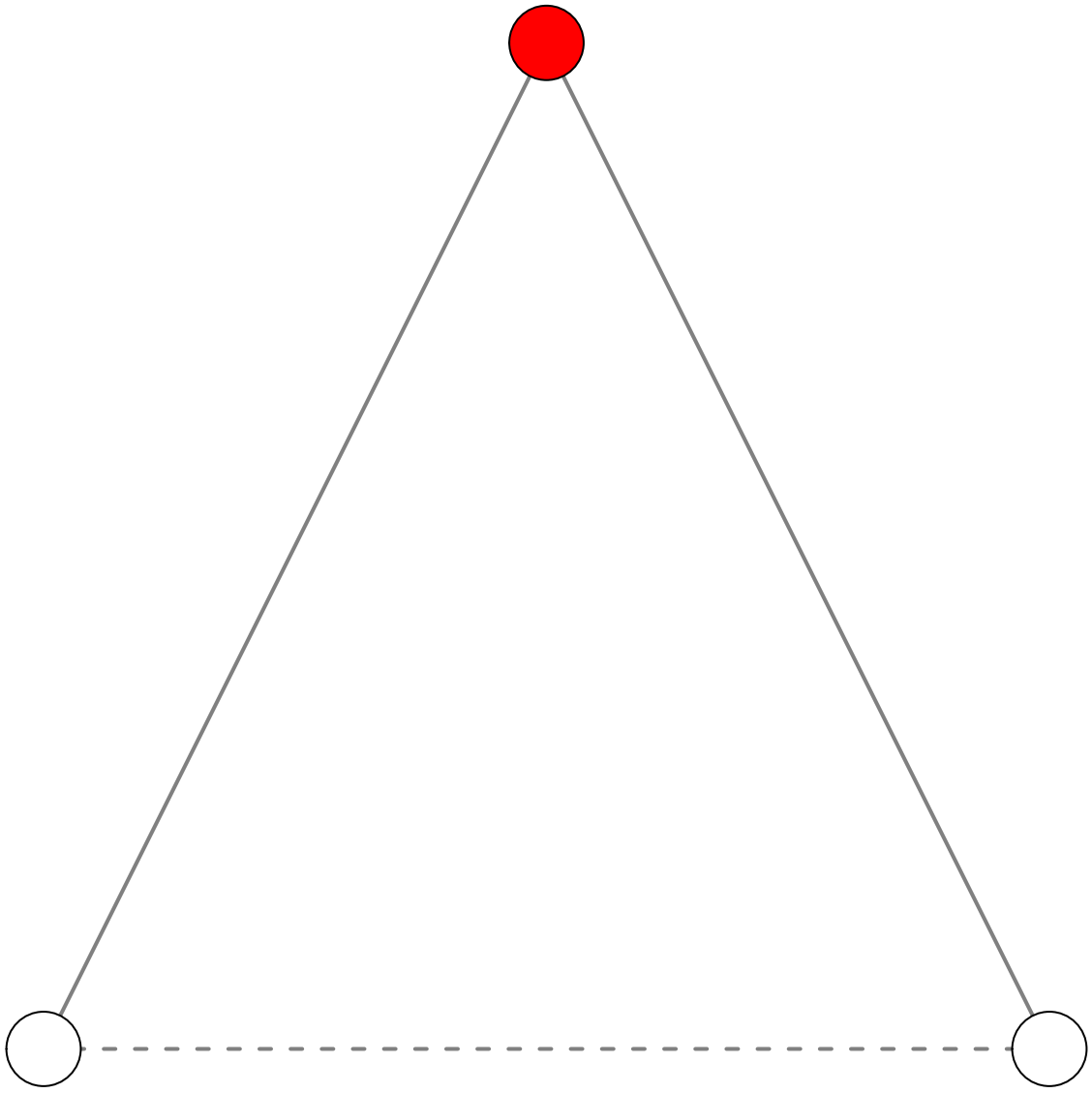
Are two connected nodes also connected to each other? Or more general, does a graph exhibit cliques?

Local clustering

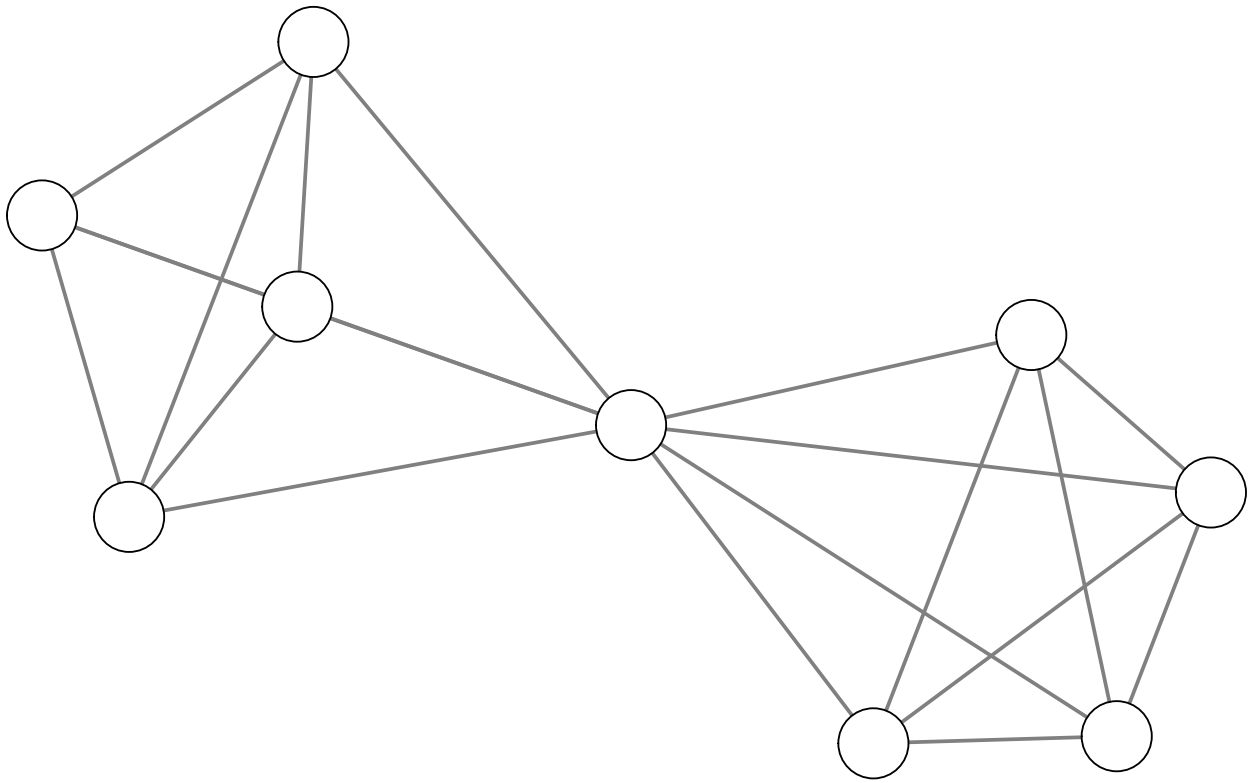
The local clustering coefficient, $\mathrm{cl}(v)$, gives for node n the proportion that the neighboors of v are also connected to each other.
This corresponds for dividing the amount of "triangles" of which node v is part, $\tau_\Delta(v)$ to the amount of possible triangles of which v could be part: $\tau_3(v)$:

$$\mathrm{cl}(v) = \begin{cases} \frac{\tau_\Delta(v)}{\tau_3(v)} & \text{if } C_D(v) \geq 2 \\ 0 \text{ or } \mathrm{NaN} & \text{Otherwise} \end{cases}$$

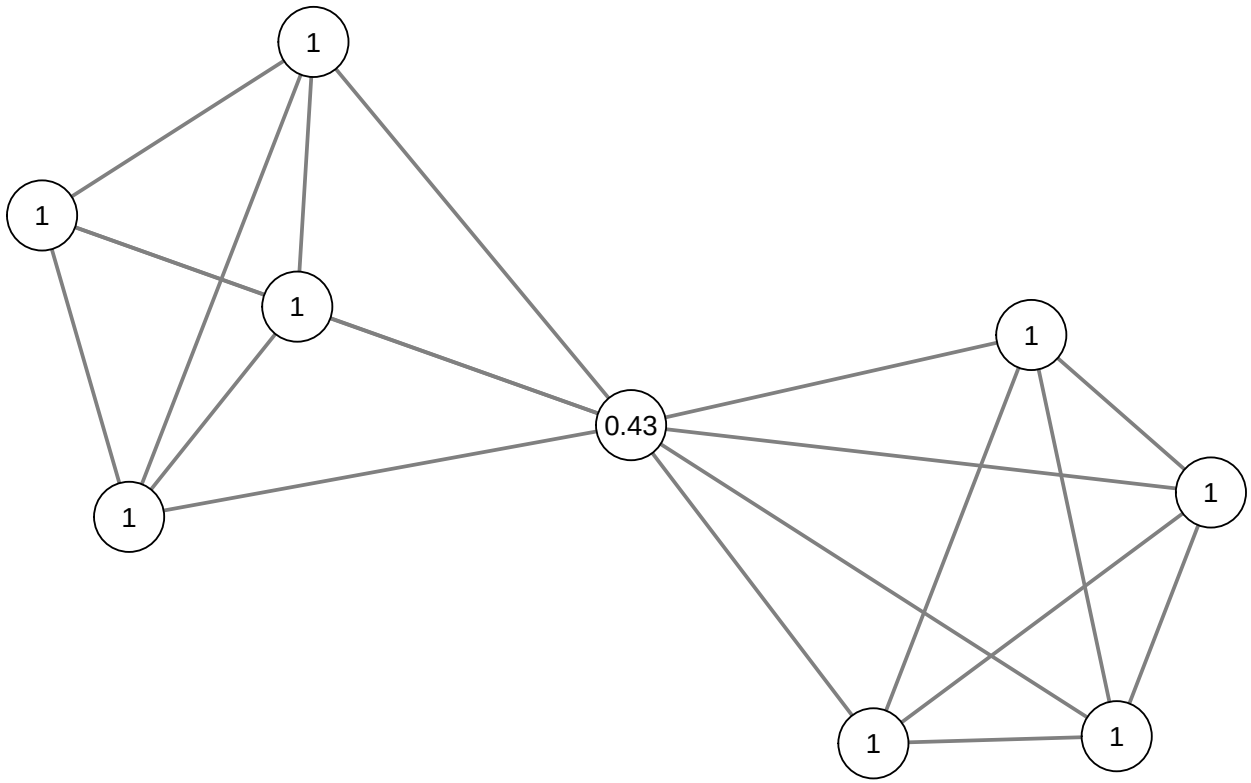
Local clustering



Local clustering



Local clustering



Local clustering

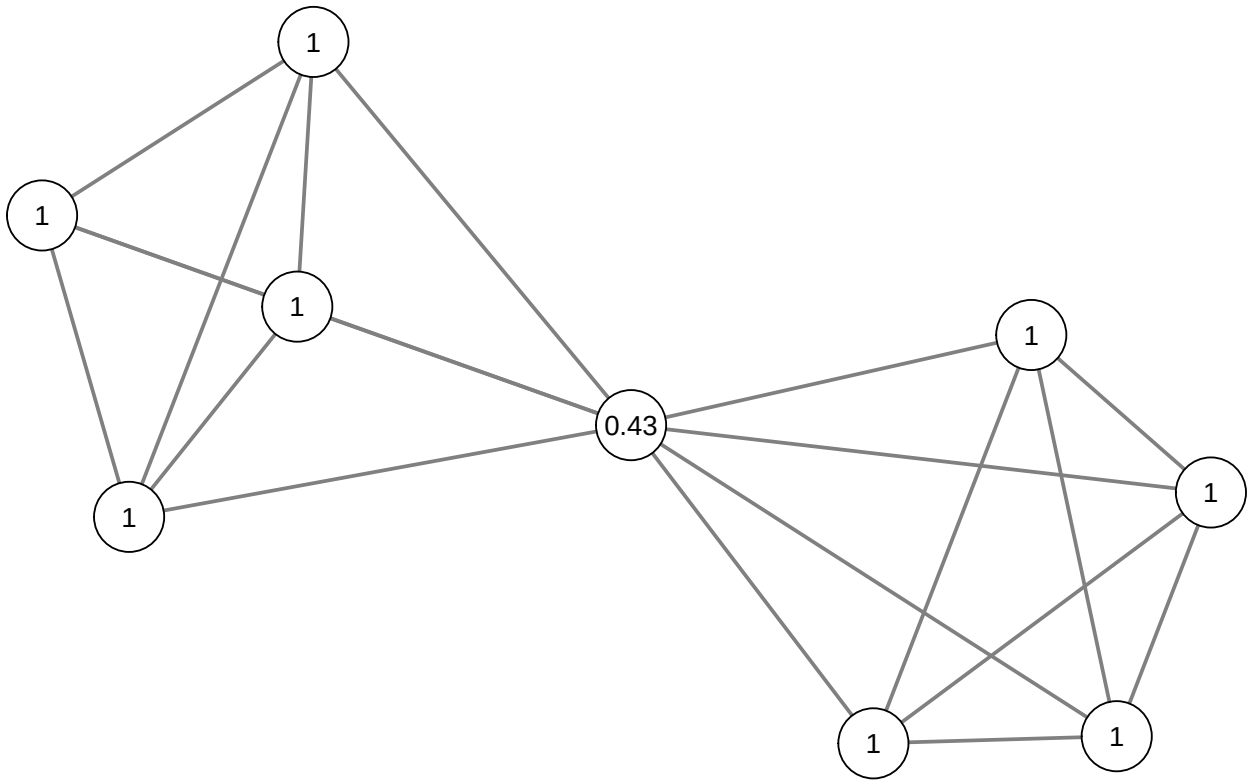
A clustering coefficient for the whole graph can be obtained by averaging all the local clustering coefficients:

$$\mathrm{cl}(G) = \frac{1}{|N|} \sum_{i=1}^{|N|} \mathrm{cl}(i)$$

Although this being an average of averages, the appropriate weighted average is more informative:

$$\begin{aligned} \mathrm{cl}_T(G) &= \frac{\sum_{i=1}^{|N|} \tau_\Delta(i)\mathrm{cl}(i)}{\tau_3(i)} \\ &= \frac{3\tau_\Delta(G)}{\tau_3(G)} \end{aligned}$$

Local clustering



```r
A <- matrix(0, 9, 9)
A[1:5, 1:5] <- 1
A[5:9, 5:9] <- 1
library("igraph")
G4 <- graph.adjacency(A, mode = "undirected")
transitivity(G4, "local")
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 0.4286 1.0000 1.0000 1.0000
## [9] 1.0000
```

```r
transitivity(G4, "global")
```

```
## [1] 0.7895
```

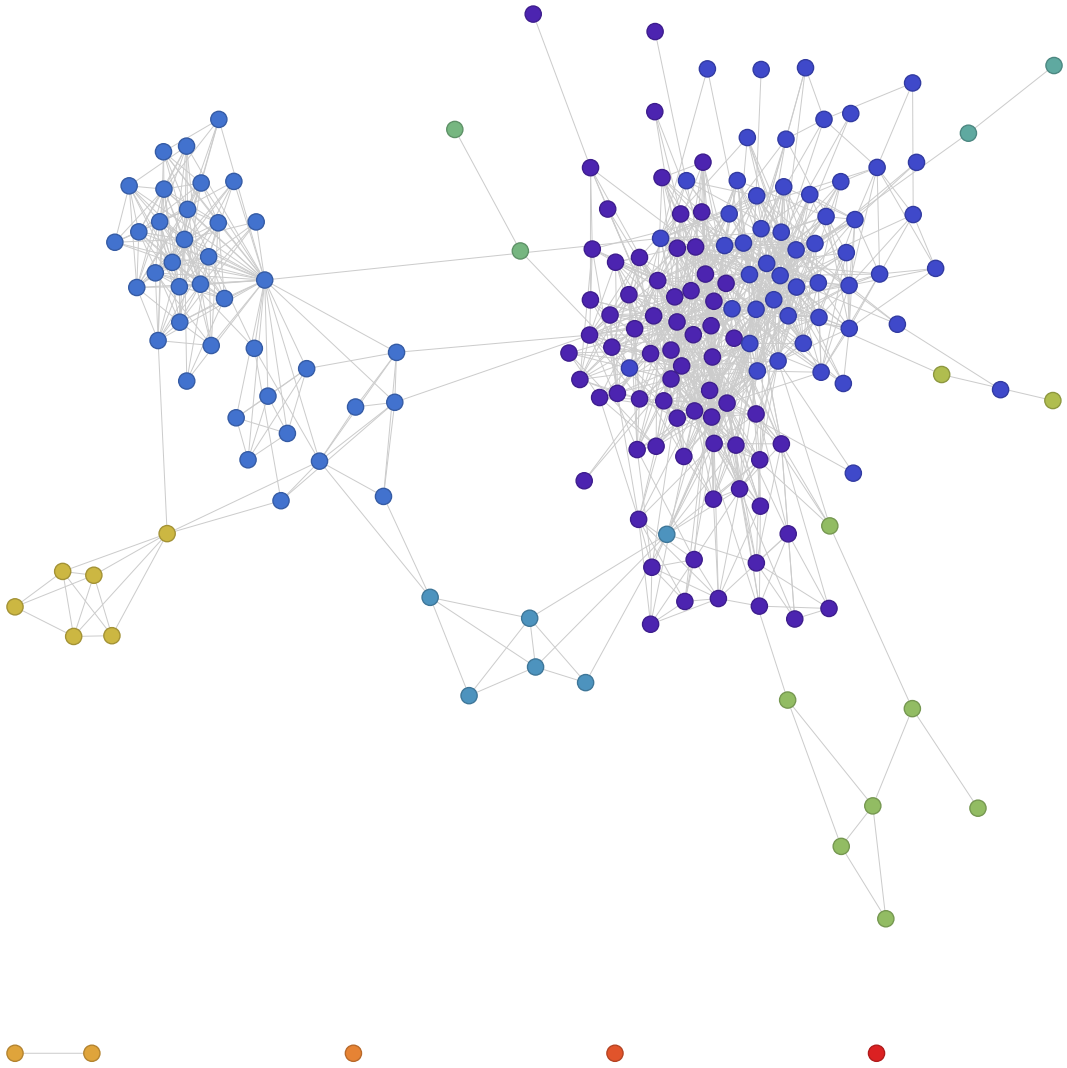
```r
transitivity(G4, "average")
```

```
## [1] 0.9365
```

Small world

The famous paper of ? (?)—already cited 20453 times—describes the “small world” principle that frequently occurs in natural graphs.

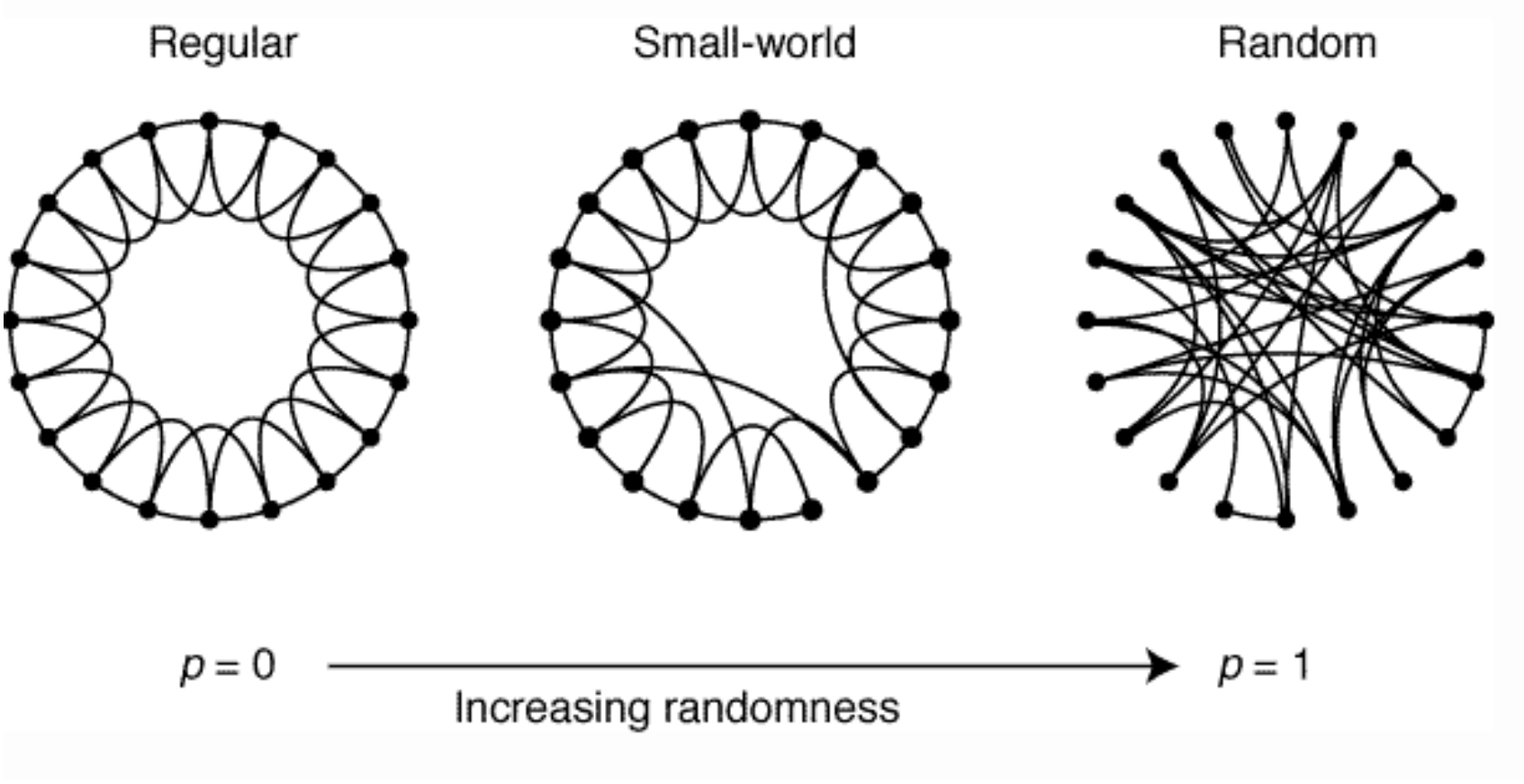
- “Six degrees of separation”
- High clustering and low average path length



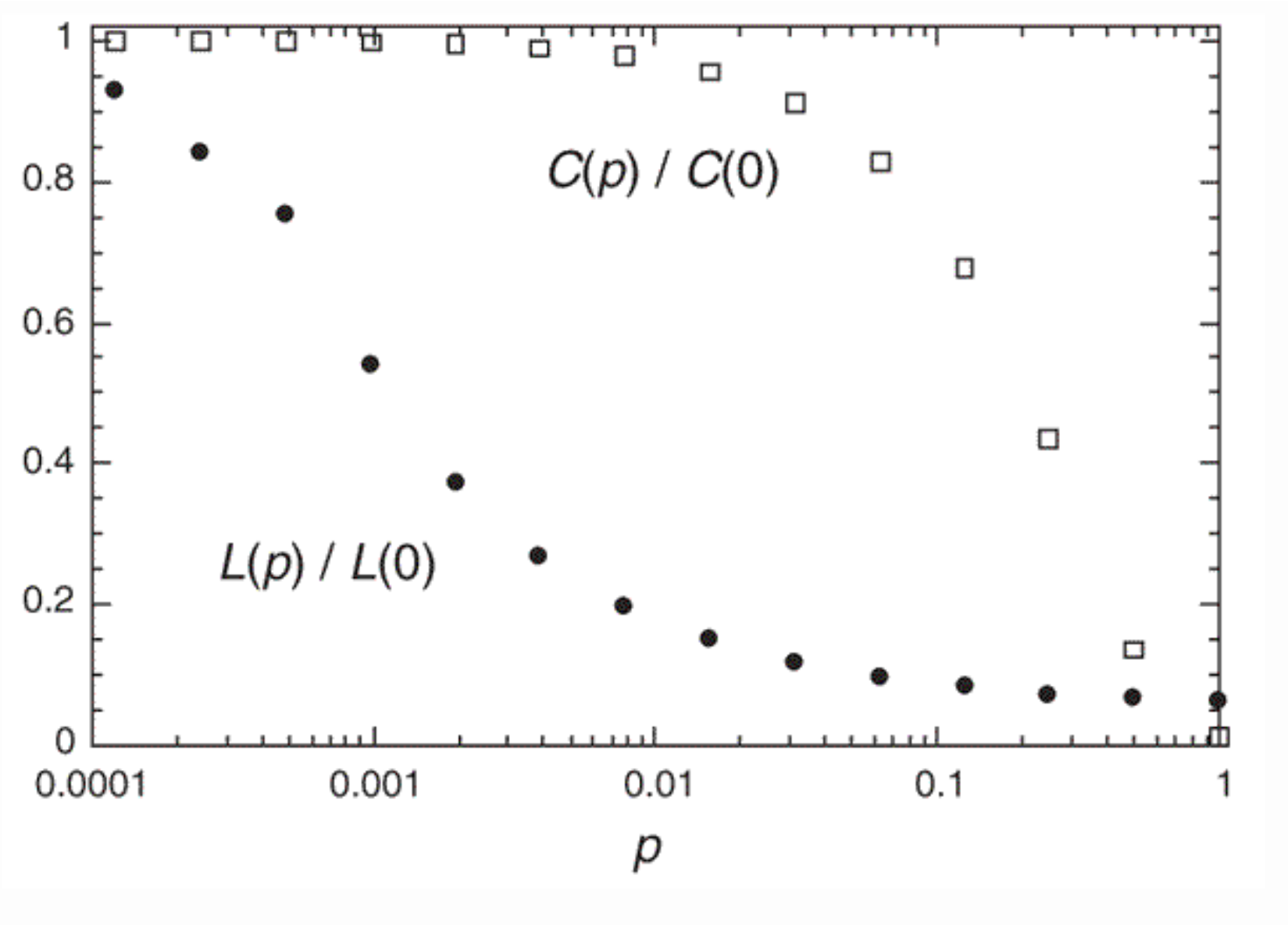
mouseover for friend details

(based on data from 190 of 203 friends)

Small World



Small World

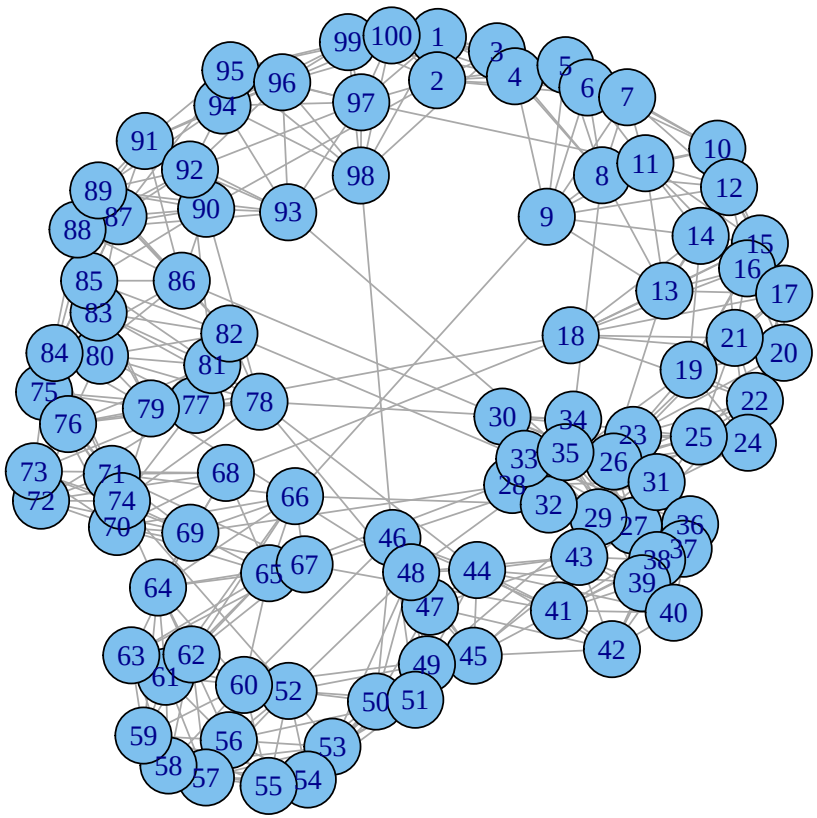


Small world

A graph exhibits a small world if it has a much higher clustering than a random graph of the same dimensions while still having a low APL.

Small world

```
# Simulate a graph:
set.seed(1)
G5 <- watts.strogatz.game(1, 100, 5, 0.05)
plot(G5)
```



Small world

Function to compute average path length of random graph:

```
APLr <- function(x) {
    if ("qgraph" %in% class(x))
        x <- as.igraph(x)
    if ("igraph" %in% class(x))
        x <- get.adjacency(x)
    N = nrow(x)
    p = sum(x/2)/sum(lower.tri(x))
    eulers_constant <- 0.57721566490153
    l = (log(N) - eulers_constant)/log(p * (N - 1)) + 0.5
    l
}
```

Small world

Function to compute clustering of random graph:

```
Cr <- function(x) {
    if ("qgraph" %in% class(x))
        x <- as.igraph(x)
    if ("igraph" %in% class(x))
        x <- get.adjacency(x)
    N = nrow(x)
    p = sum(x/2)/sum(lower.tri(x))
    t = (p * (N - 1)/N)
    t
}
```

Small world

Is there a small world?

```
# Clustering in graph:
transitivity(G5)

## [1] 0.5402

# Clustering in random graph:
Cr(G5)

## Loading required package:  Matrix
## Loading required package:  lattice

## [1] 0.1

# Average path length in graph:
average.path.length(G5)

## [1] 2.867

# Average path length in random graph:
APLr(G5)
```

Small world

Small world index:

```
(transitivity(G5) / Cr(G5)) /
(average.path.length(G5) / APLr(G5))

## [1] 4.237
```

Higher than 3? There is a Small world!

References

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269–271.

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, *32*(3), 245–251.