Network Analysis for Psychologists

Using qgraph in R

Sacha Epskamp

University of Amsterdam
Department of Psychological Methods

22-05-2014
APS 2014

- These sheets are available on `http://sachaepskamp.com/presentations`
  - Including an R script containing codes of the last section

This workshop is cosponsored by APS and the Society of Multivariate Experimental Psychology (SMEP).

The Psych Systems Project

http://www.psychosystems.org/

Psychosystems

Staff:

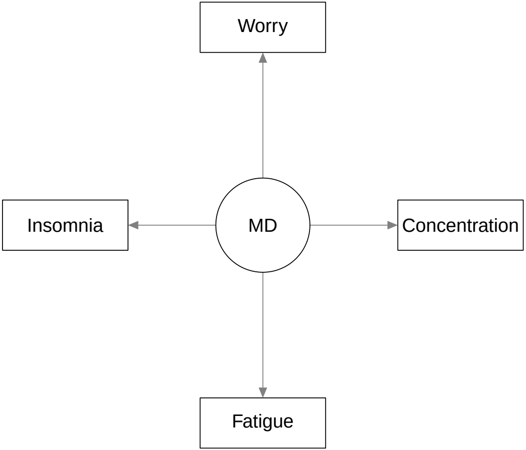
- Denny Borsboom
- Angelique Cramer
- Lourens Waldorp
- Sacha Epskamp
- Claudia van Borkulo
- Mijke Rhemtulla

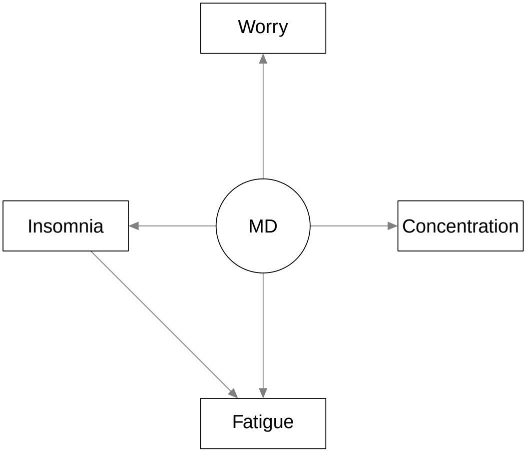
Students:

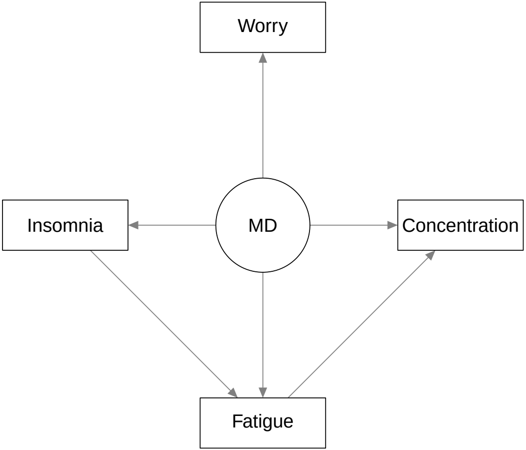
- Marie Deserno
- Jolanda Kossakowski
- Pia Tio

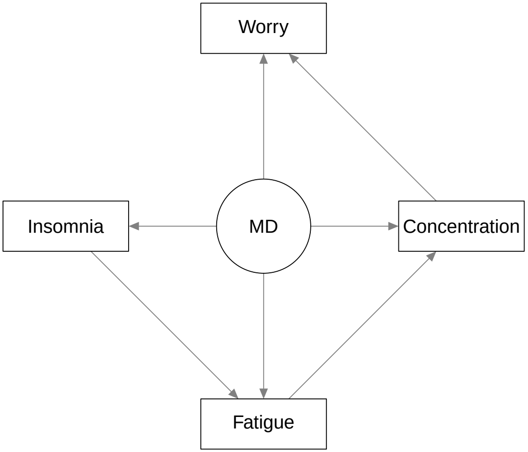
Collaborators:

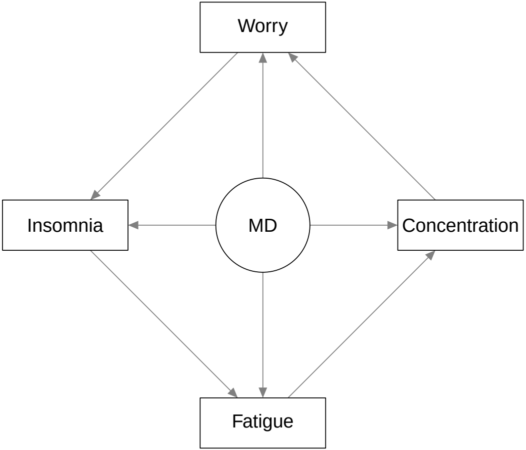
- Giulio Costantini
- Jonathon Love
- Laura Bringmann
- Francis Tuerlinckx
- Laura Ruzzano
- Hilde Geurts
- ...

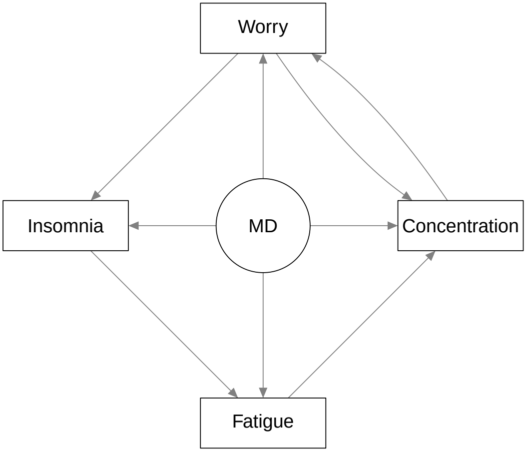

Worry
Insomnia
MD
Concentration
Fatigue

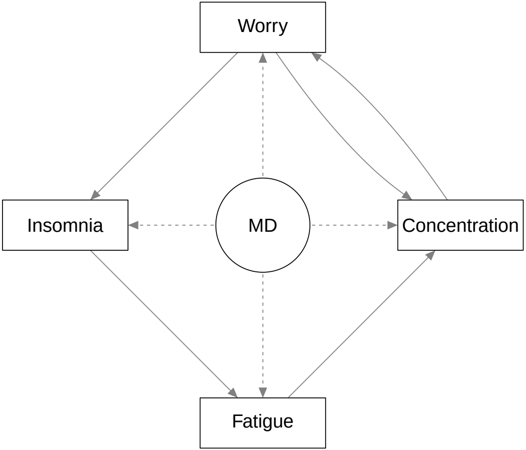

Worry
Insomnia
MD
Concentration
Fatigue

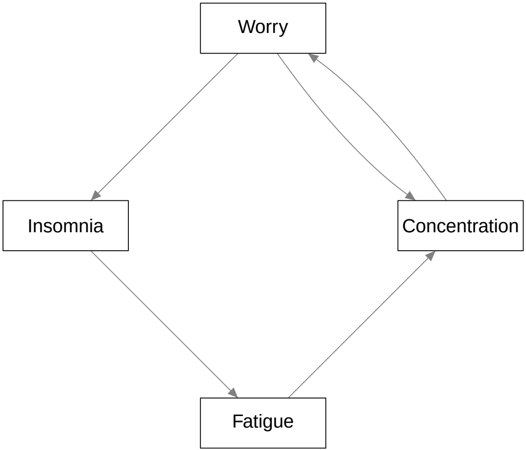

Worry
Insomnia
MD
Concentration
Fatigue

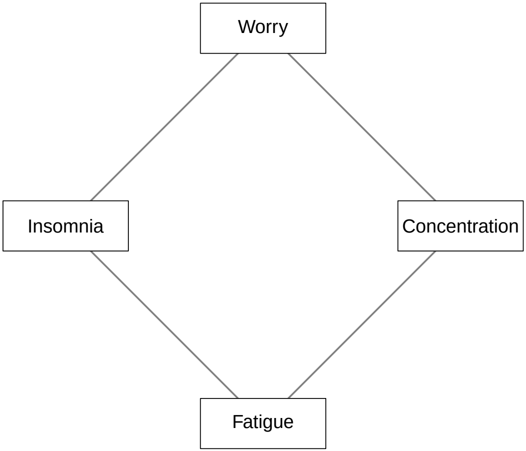

Worry
Insomnia
MD
Concentration
Fatigue


Worry
Insomnia
MD
Concentration
Fatigue

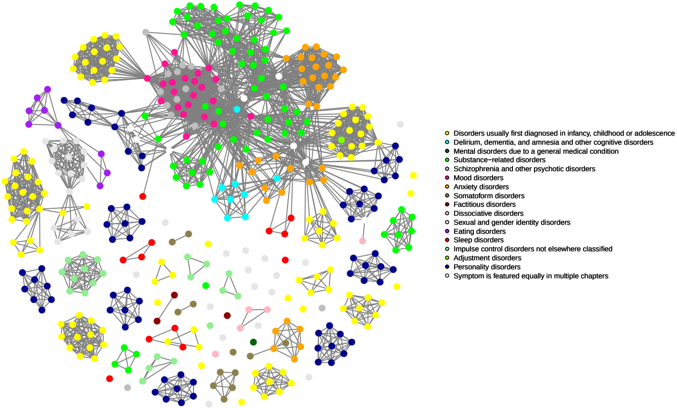

Worry
Insomnia
MD
Concentration
Fatigue


Worry
Insomnia
MD
Concentration
Fatigue


Worry
Insomnia
Concentration
Fatigue

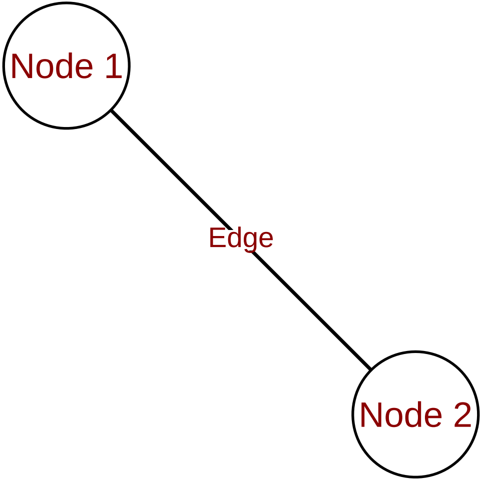

Worry
Insomnia
Concentration
Fatigue

Psychology as a complex system



What is a network?

- A network is a set of *nodes* connected by a set of *edges*


Node 1
Edge
Node 2

What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People

What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities

What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities
    - Symptoms

What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities
    - Symptoms
  - An edge represents some connection between two nodes. Again, this can be anything:

What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities
    - Symptoms
  - An edge represents some connection between two nodes. Again, this can be anything:
    - Friendship / contact

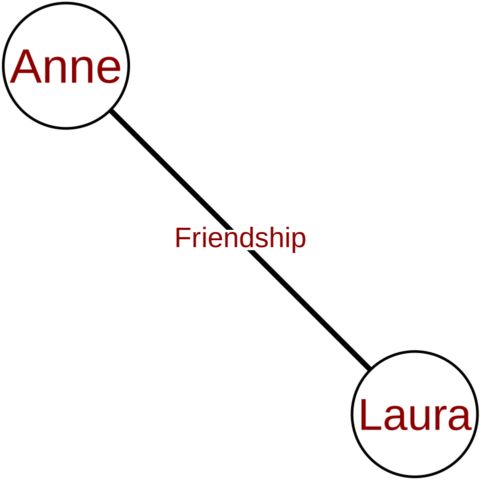
What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities
    - Symptoms
  - An edge represents some connection between two nodes. Again, this can be anything:
    - Friendship / contact
    - Distance

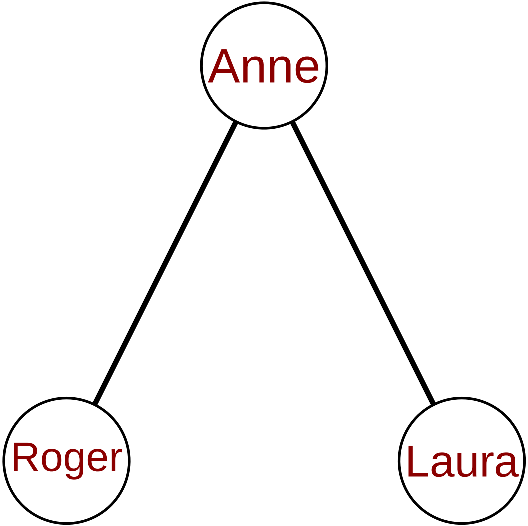
What is a network?

- A network is a set of *nodes* connected by a set of *edges*
  - A node represents an entity, this can be anything:
    - People
    - Cities
    - Symptoms
  - An edge represents some connection between two nodes. Again, this can be anything:
    - Friendship / contact
    - Distance
    - Comorbidity

Anne is friends with Laura:

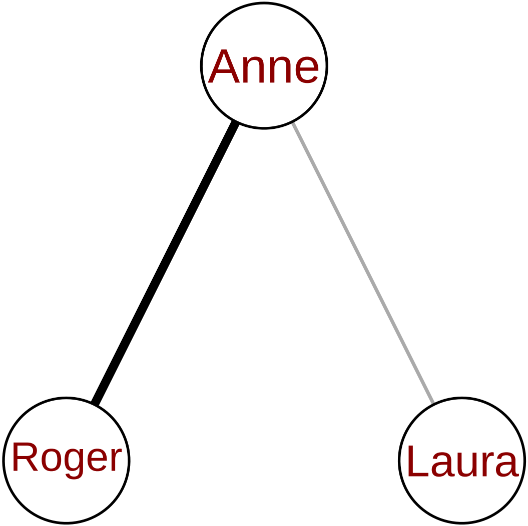


Anne is friends with Laura and Roger, but Laura is not friends with Roger:



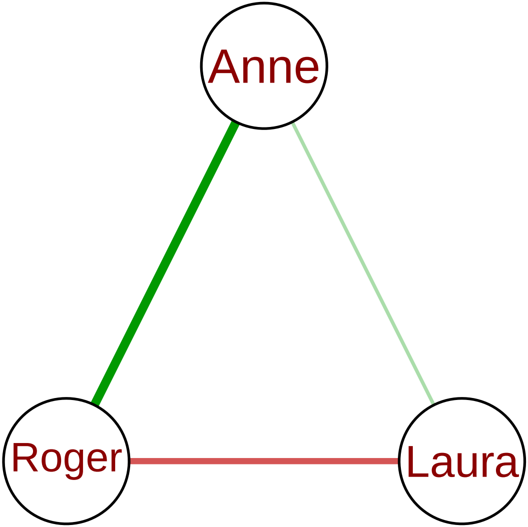
Networks can be weighted

Anne is better friends with met Roger than Laura:



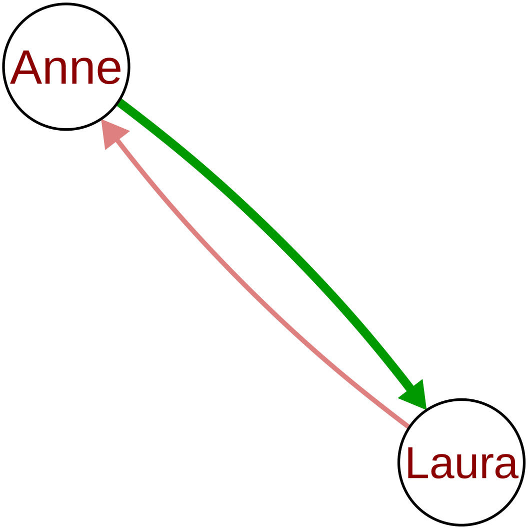
Weights can be signed

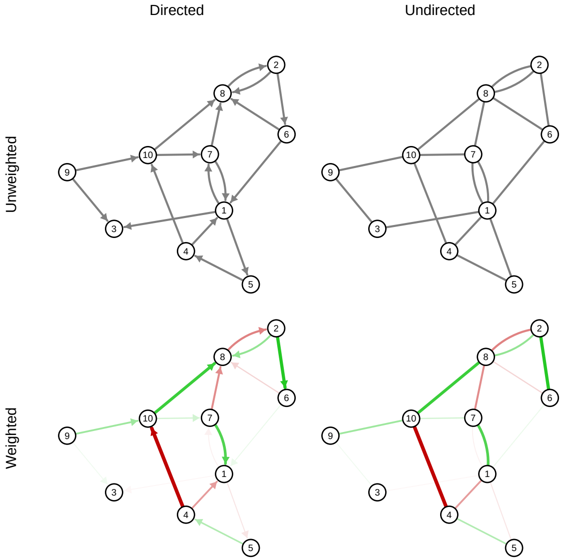
Anne is friends with Roger and Laura, but Roger and Laura don't like each other at all!



Networks can be directed

Anne likes Laura, but Laura doesn't like Anne:





Directed
Undirected
Unweighted
Weighted
1
2
3
4
5
6
7
8
9
10

Interpreting Networks

A network can be interpreted in different ways:

- As a model of interacting components

Interpreting Networks

A network can be interpreted in different ways:

- As a model of interacting components
  - Information can spread from node to node via edges

Interpreting Networks

A network can be interpreted in different ways:

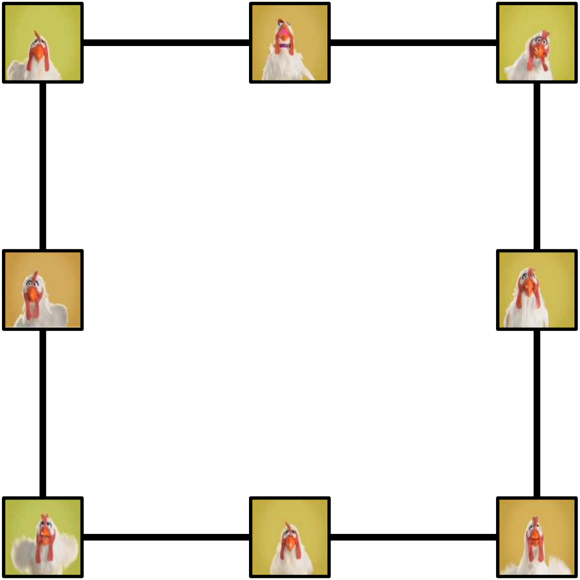
- As a model of interacting components
  - Information can spread from node to node via edges
- As a causal model

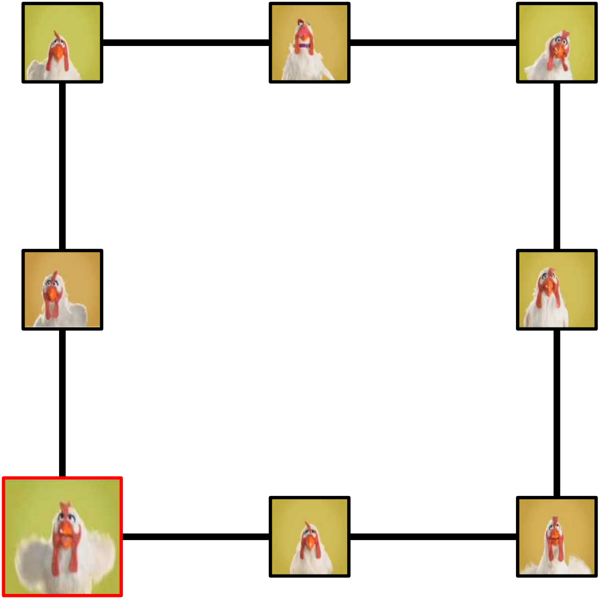
Interpreting Networks

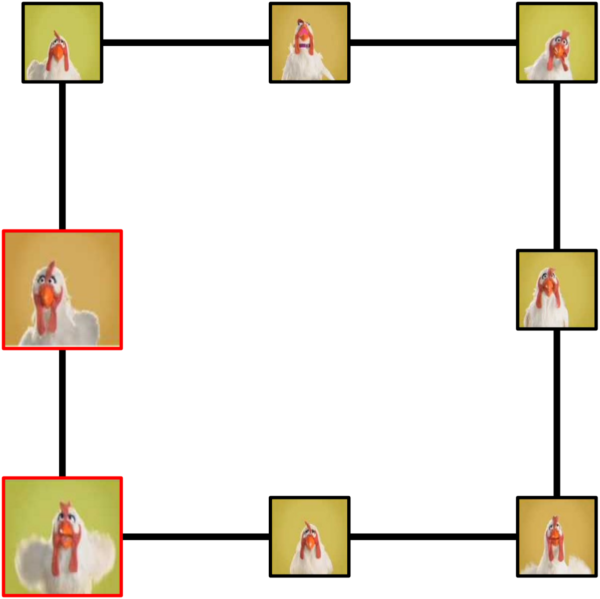
A network can be interpreted in different ways:

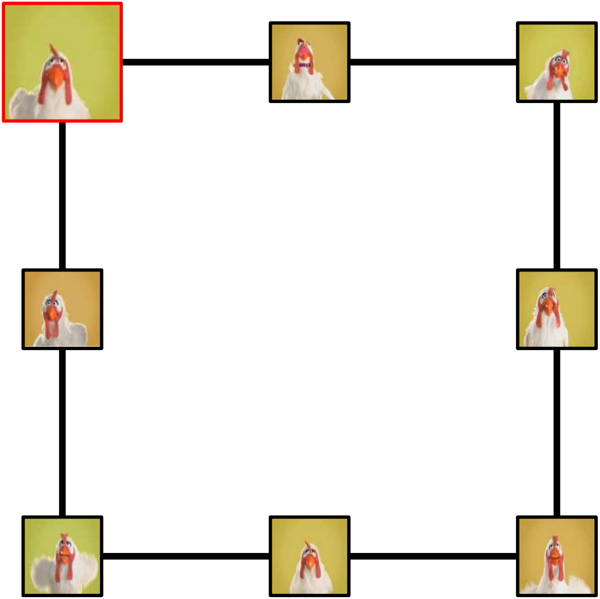
- As a model of interacting components
  - Information can spread from node to node via edges
- As a causal model
- As a predictive model

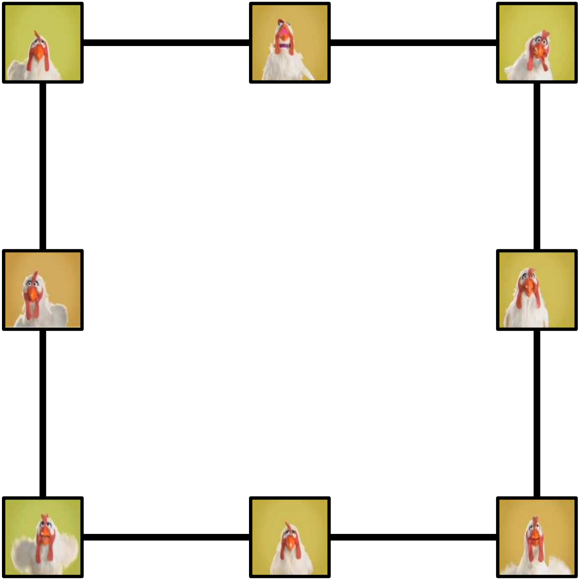
Interacting components

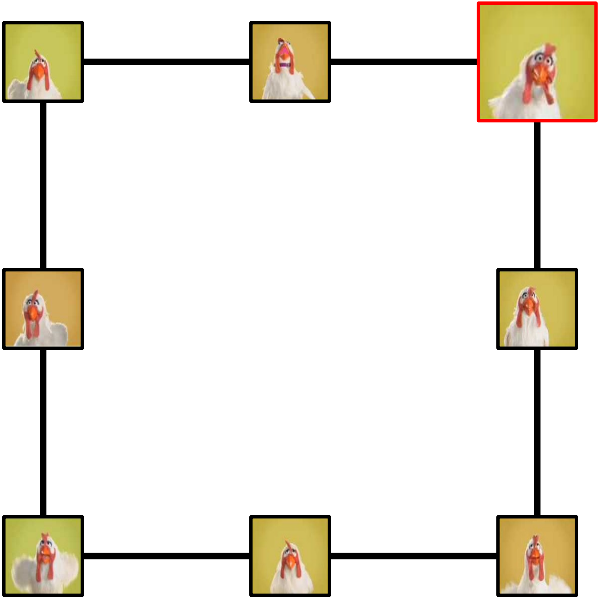


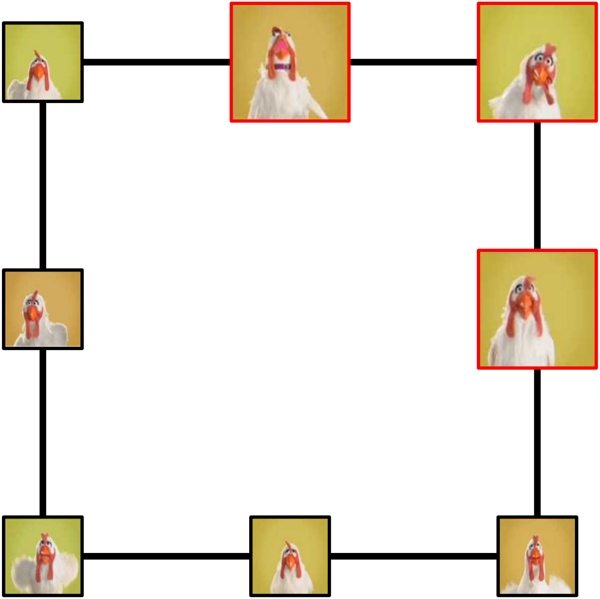


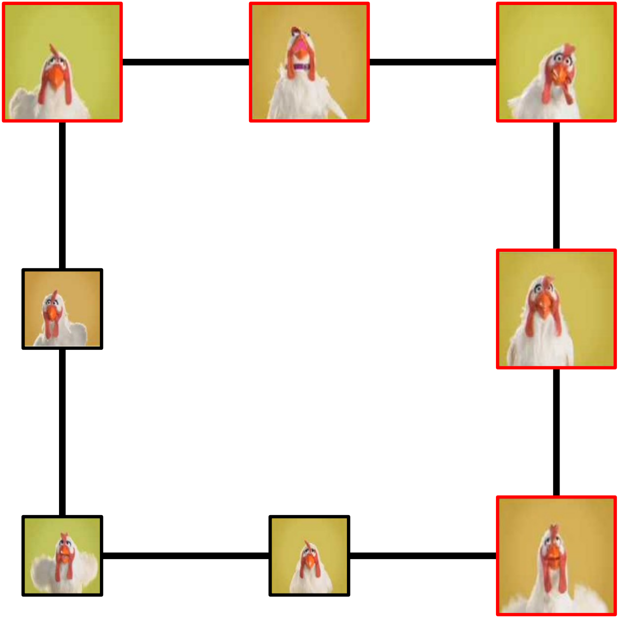


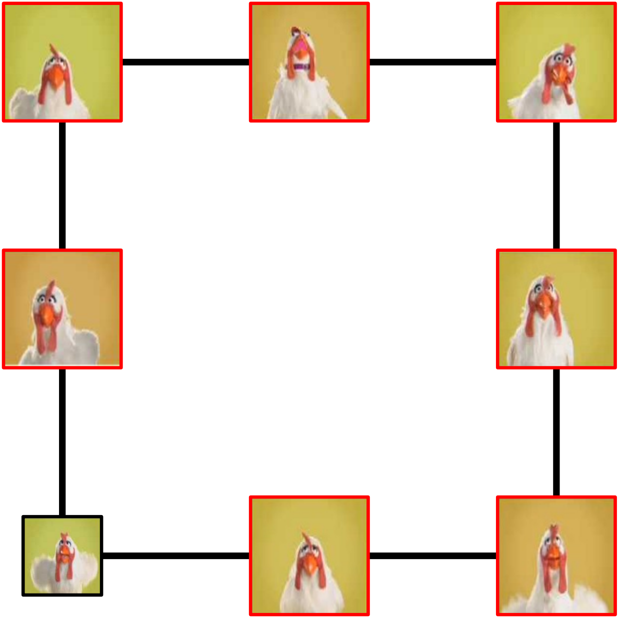


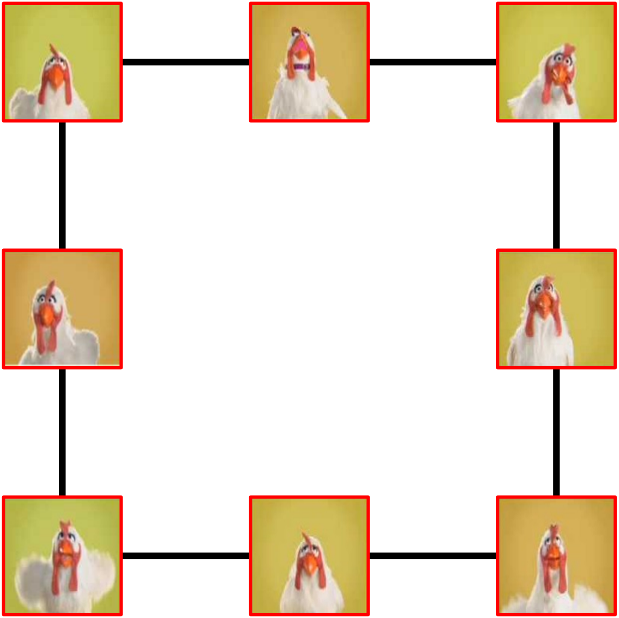








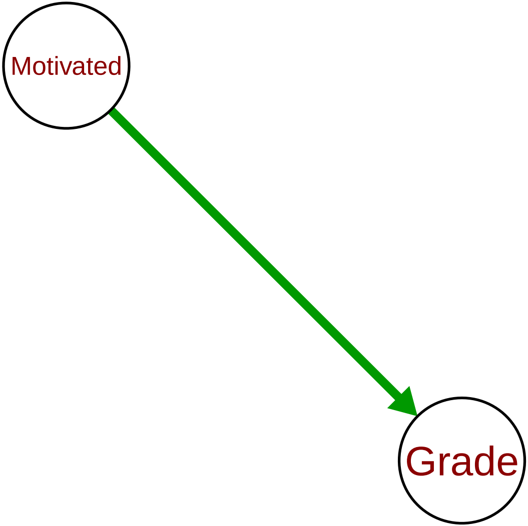




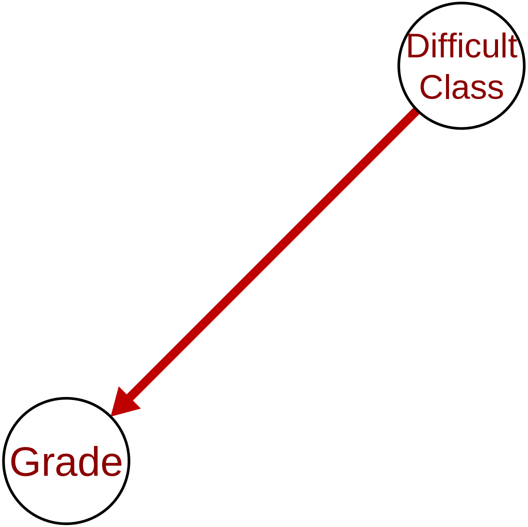
Psychopathology as a virus...

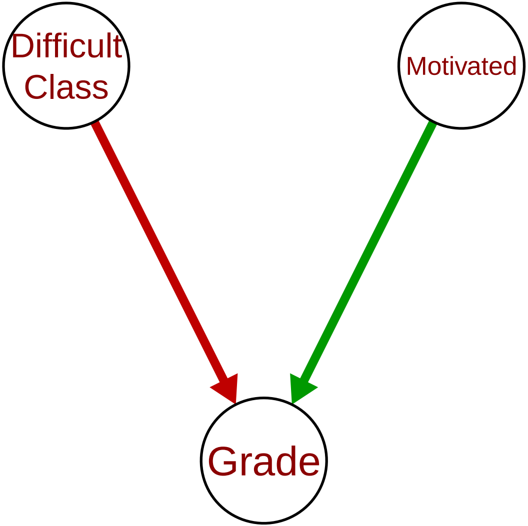
Networks can model causality

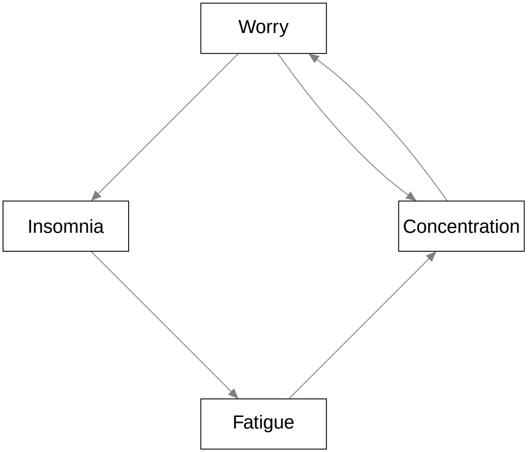
A motivated student will get higher grades:



A difficult class causes students to get lower grades:




Difficult Class
Motivated
Grade


Worry
Insomnia
Concentration
Fatigue

Causal models

- Methods such as Structural Equation Modeling (SEM) can be used to confirmatory test causal structures

Causal models

- Methods such as Structural Equation Modeling (SEM) can be used to confirmatory test causal structures
- Exploratory inference for causal structures is much harder and mostly not possible without experimental design

Causal models

- Methods such as Structural Equation Modeling (SEM) can be used to confirmatory test causal structures
- Exploratory inference for causal structures is much harder and mostly not possible without experimental design
  - In cross-sectional data only specific structures under the strong assumption of acyclicness are inferable

Causal models

- Methods such as Structural Equation Modeling (SEM) can be used to confirmatory test causal structures
- Exploratory inference for causal structures is much harder and mostly not possible without experimental design
  - In cross-sectional data only specific structures under the strong assumption of acyclicness are inferable
  - time series could be used to test Granger-causality, but this is again only a confirmatory test

Predictive networks

- A network we *can* infer is a predictive network

Predictive networks

- A network we *can* infer is a predictive network
- In predictive networks, we draw an edge from node A to node B if node A *predicts* node B.

Predictive networks

- A network we *can* infer is a predictive network
- In predictive networks, we draw an edge from node *A* to node *B* if node *A* *predicts* node *B*.
- Underlying causal networks can be directly translated into predictive networks

Predictive networks

- A network we *can* infer is a predictive network
- In predictive networks, we draw an edge from node *A* to node *B* if node *A predicts* node *B*.
- Underlying causal networks can be directly translated into predictive networks
    - Equivalent causal networks lead to the same predictive network

Predictive networks

- A network we *can* infer is a predictive network
- In predictive networks, we draw an edge from node *A* to node *B* if node *A predicts* node *B*.
- Underlying causal networks can be directly translated into predictive networks
  - Equivalent causal networks lead to the same predictive network
- If a interaction network generated the data the predictive network can correctly estimate the structure

True generating model:

- Does A predict B?

True generating model:

- Does *A* predict *B*?
  - Yes!

True generating model:

- Does *A* predict *B*?
  - Yes!
- Does *B* predict *A*?

True generating model:

- Does *A* predict *B*?
  - Yes!
- Does *B* predict *A*?
  - Yes!

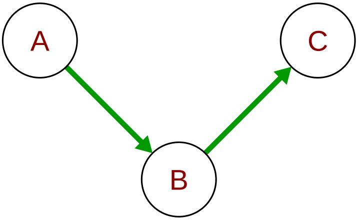
True generating model:

- Does *A* predict *B*?
  - Yes!
- Does *B* predict *A*?
  - Yes!
- If node *A* predicts node *B*, node *B* predicts node *A*. Hence, predictive networks often are undirected networks

True generating model:

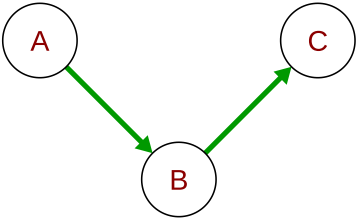
Predictive model:

True generating model:



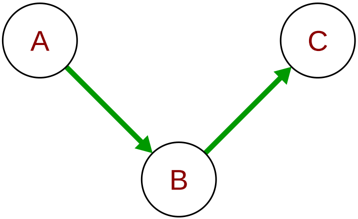
- Does *A* predict *B*?

True generating model:



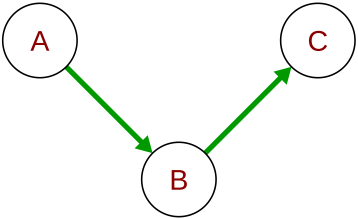
- Does *A* predict *B*?
  - Yes!

True generating model:



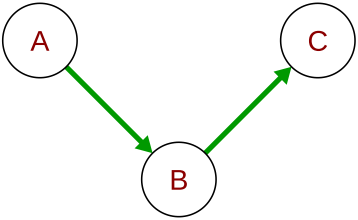
- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?

True generating model:



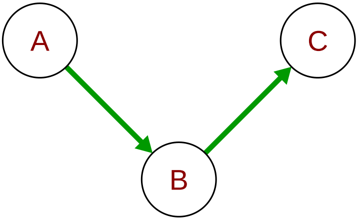
- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?
  - Yes!

True generating model:



- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?
  - Yes!
- Does *A* predict *C* or vise-versa?

True generating model:



- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?
  - Yes!
- Does *A* predict *C* or vise-versa?
  - *A* and *C* will be correlated, thus knowing *A* allows you to predict *C*
  - But if we also know *B*, *A* will not add information about *C*.
  - Thus *A* only predicts *C via B*

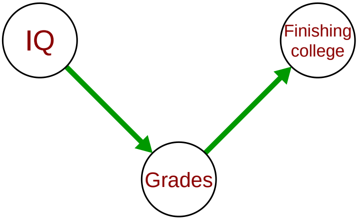
- In an *association* network nodes are connected if they predict each other regardless of other nodes
  - Edges represent *correlations*

- In an *association* network nodes are connected if they predict each other regardless of other nodes
  - Edges represent *correlations*
- In a *concentration* network nodes are connected if they predict each other *given that we know other nodes*

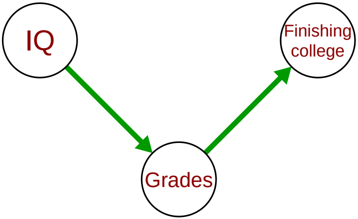
- In an *association* network nodes are connected if they predict each other regardless of other nodes
  - Edges represent *correlations*
- In a *concentration* network nodes are connected if they predict each other *given that we know other nodes*
  - Edges represent *partial correlations* or *multiple regression coefficients*

- In an *association* network nodes are connected if they predict each other regardless of other nodes
  - Edges represent *correlations*
- In a *concentration* network nodes are connected if they predict each other *given that we know other nodes*
  - Edges represent *partial correlations* or *multiple regression coefficients*
  - If there is no edge between two nodes they are *conditionally independent* given all other nodes

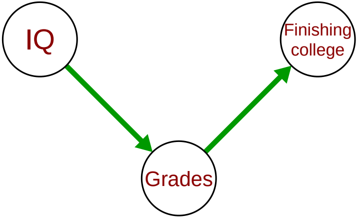
- In an *association* network nodes are connected if they predict each other regardless of other nodes
  - Edges represent *correlations*
- In a *concentration* network nodes are connected if they predict each other *given that we know other nodes*
  - Edges represent *partial correlations* or *multiple regression coefficients*
  - If there is no edge between two nodes they are *conditionally independent* given all other nodes
  - Such networks are also called Markov Random Fields.



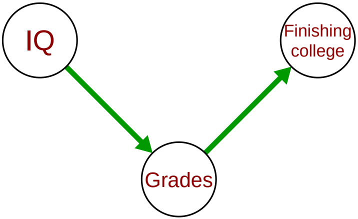
- A smart student will get better grades



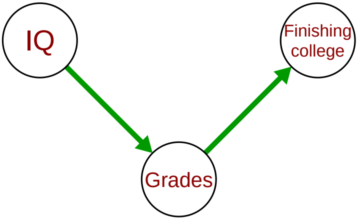
- A smart student will get better grades
- A student with good grades is more likely to finish college



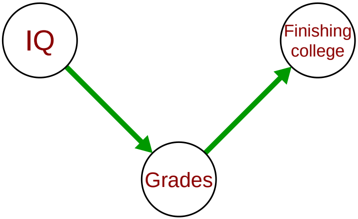
- A smart student will get better grades
- A student with good grades is more likely to finish college
- IQ is correlated with finishing college and thus predicts finishing college



- A smart student will get better grades
- A student with good grades is more likely to finish college
- IQ is correlated with finishing college and thus predicts finishing college
    - Association

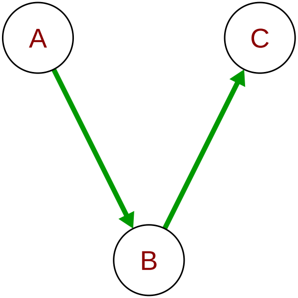


- A smart student will get better grades
- A student with good grades is more likely to finish college
- IQ is correlated with finishing college and thus predicts finishing college
  - Association
- Given that we know the grades a student got, the IQ score does not improve the prediction of finishing college

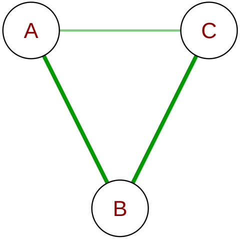


- A smart student will get better grades
- A student with good grades is more likely to finish college
- IQ is correlated with finishing college and thus predicts finishing college
  - Association
- Given that we know the grades a student got, the IQ score does not improve the prediction of finishing college
  - Concentration

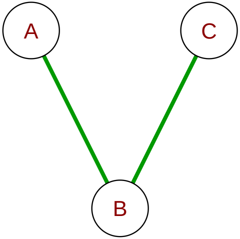
True generating model:



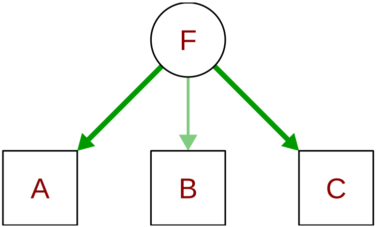
Association network:



Concentration network:



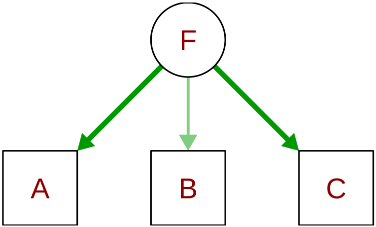
True generating model:



Where *F* is a unobservable latent variable

- Does *A* predict *B*?

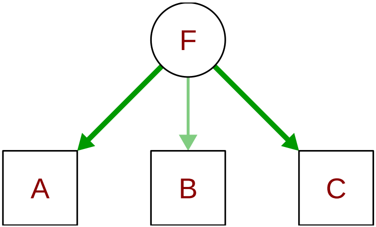
True generating model:



Where F is a unobservable latent variable

- Does A predict B?
  - Yes!

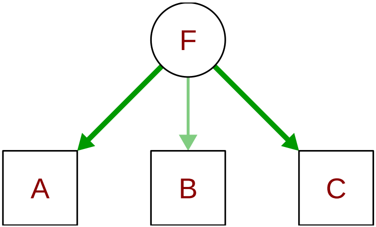
True generating model:



Where F is a unobservable latent variable

- Does A predict B?
    - Yes!
- Does B predict C?

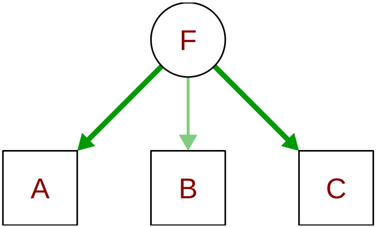
True generating model:



Where *F* is a unobservable latent variable

- Does *A* predict *B*?
    - Yes!
- Does *B* predict *C*?
    - Yes!

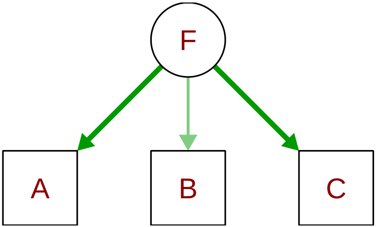
True generating model:



Where *F* is a unobservable latent variable

- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?
  - Yes!
- Does *A* predict *C*?

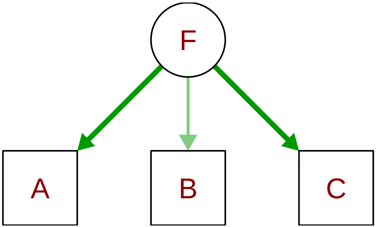
True generating model:



Where *F* is a unobservable latent variable

- Does *A* predict *B*?
  - Yes!
- Does *B* predict *C*?
  - Yes!
- Does *A* predict *C*?
  - Yes!

True generating model:

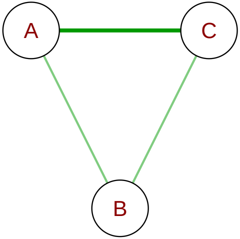


Where F is a unobservable latent variable

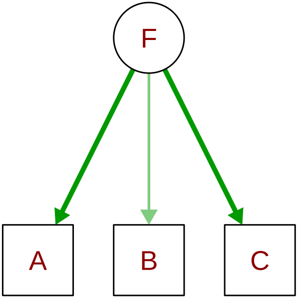
- Does A predict B?
    - Yes!
- Does B predict C?
    - Yes!
- Does A predict C?
    - Yes!

If a latent cause underlies the nodes, all nodes predict all other nodes. Since we can not condition on F the prediction is not mediated. The association and concentration networks then are the same: fully connected clusters.

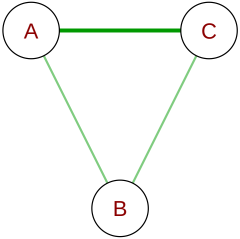
Association network:



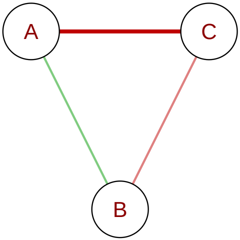
True generating model:



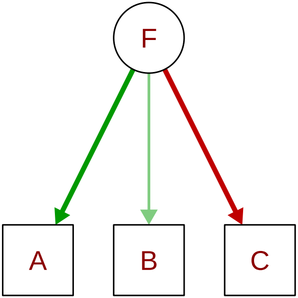
Concentration network:



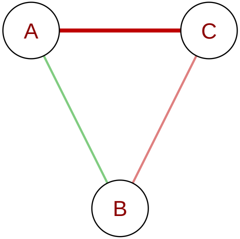
Association network:



True generating model:



Concentration network:



Predictive networks

- Easily constructable

Predictive networks

- Easily constructable
- Make no assumptions about directionality

Predictive networks

- Easily constructable
- Make no assumptions about directionality
- Naturally cyclic

Predictive networks

- Easily constructable
- Make no assumptions about directionality
- Naturally cyclic
- Especially edges concentration networks can identify causal effects

Predictive networks

- Easily constructable
- Make no assumptions about directionality
- Naturally cyclic
- Especially edges concentration networks can identify causal effects
- Correspond to undirected network that could underlie the data generating model

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)
    - Polychoric correlations

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)
    - Polychoric correlations
    - Polyserial correlations

Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)
    - Polychoric correlations
    - Polyserial correlations
  - nonnormal continuous data could be transformed using the nonparanormal transformation (Liu et al., 2009)

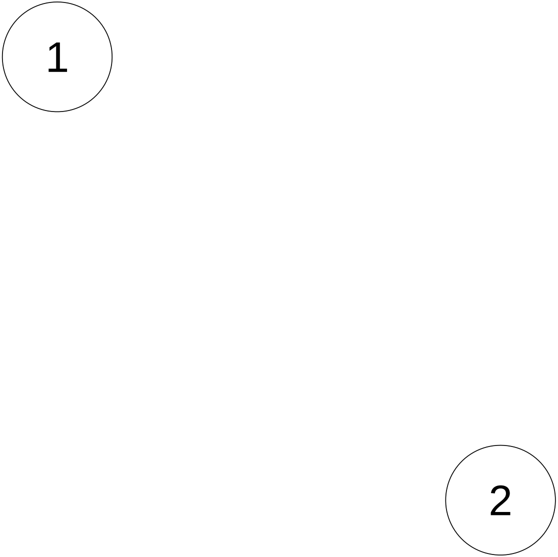
Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)
    - Polychoric correlations
    - Polyserial correlations
  - nonnormal continuous data could be transformed using the nonparanormal transformation (Liu et al., 2009)
  - For binary data we can use the Ising Model

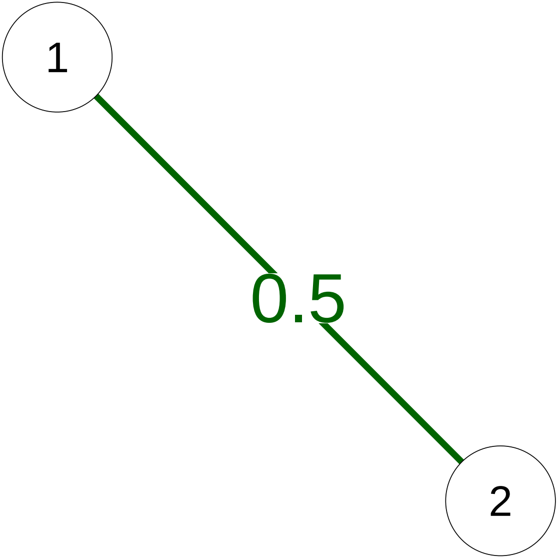
Estimating Network Topology

- In this block we will look at how to estimate association and concentration networks for cross-sectional and time-series data.
- All these methods only require a covariance matrix
- We will assume that all nodes are normally distributed
  - For ordinal data threshold models can be used (These can easily be computed in **qgraph**)
    - Polychoric correlations
    - Polyserial correlations
  - nonnormal continuous data could be transformed using the nonparanormal transformation (Liu et al., 2009)
  - For binary data we can use the Ising Model
- To start, an association network for cross-sectional data is simply a graph of the correlation matrix

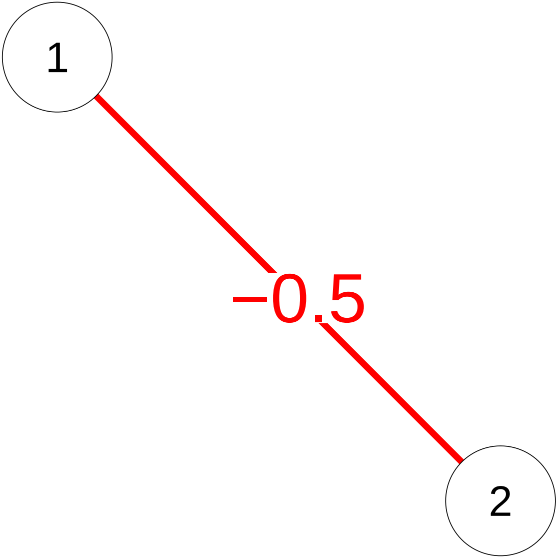
Represent each variable with a node:



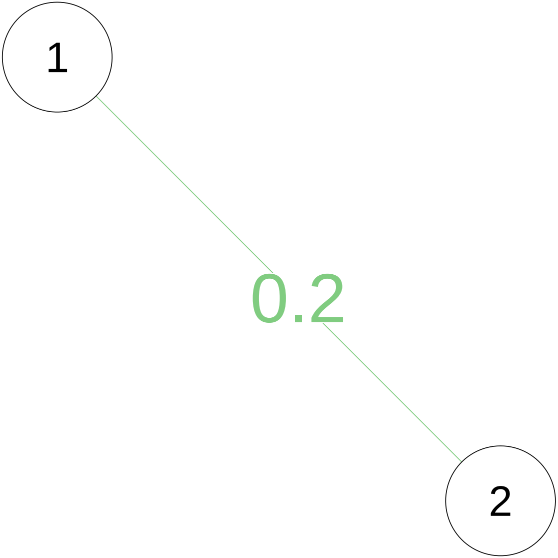
Connect them with a green edge if they are positively correlated:



Or a red edge if they are negatively correlated:



The weaker the correlation the vaguer/thinner the edge:

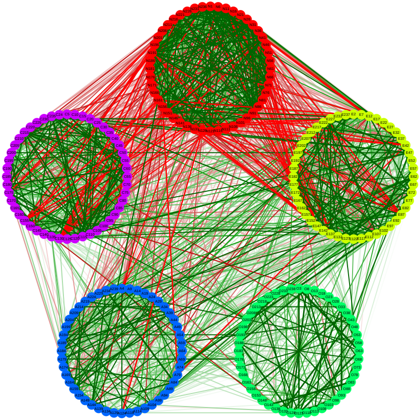


Real data example

Included in **qgraph** is a dataset in which the Dutch translation of a commonly used personality test, the NEO-PI-R (Costa & McCrae, 1992; Hoekstra, de Fruyt, & Ormel, 2003), was administered to 500 first year psychology students (Dolan, Oort, Stoel, & Wicherts, 2009). The NEO-PI-R consists of 240 items designed to measure the five central personality factors:

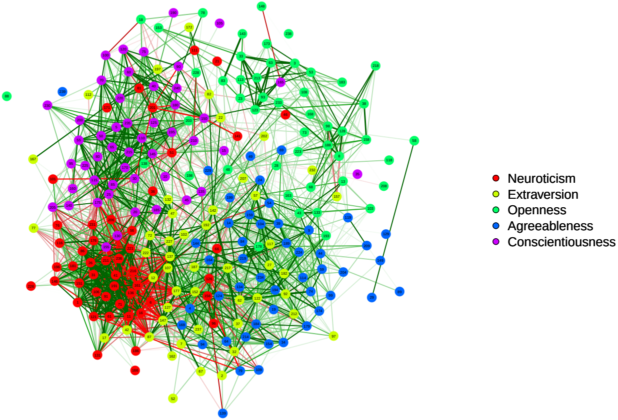
- Neuroticism
- Extroversion
- Agreeableness
- Openness to Experience
- Conscientiousness

Big 5 correlations



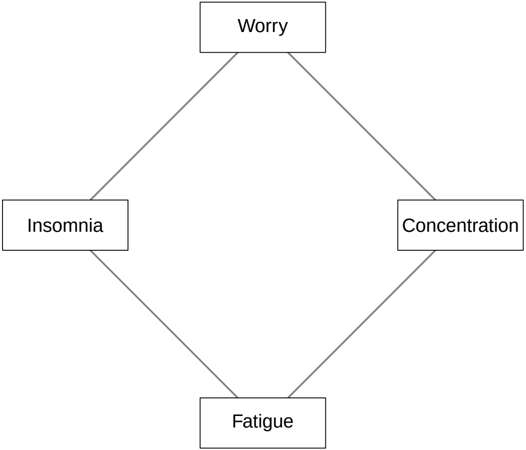
- Neuroticism
- Extraversion
- Openness
- Agreeableness
- Conscientiousness

Big 5 correlations



Toy dataset

Worry (W)	Concentration (C)	Fatigue (F)	Insomnia (I)
4	2	3	1
1	6	2	1
5	4	4	6
4	2	8	8
2	9	1	3
2	6	2	5
8	2	4	8
1	5	4	1
4	4	5	3
4	2	5	4


Worry
Insomnia
Concentration
Fatigue

```
Data

##    W C F I
## 1  4 2 3 1
## 2  1 6 2 1
## 3  5 4 4 6
## 4  4 2 8 8
## 5  2 9 1 3
## 6  2 6 2 5
## 7  8 2 4 8
## 8  1 5 4 1
## 9  4 4 5 3
## 10 4 2 5 4
```

Association

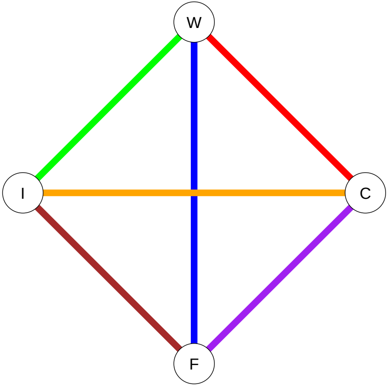
```r
round(cor(Data),2)

##       W     C     F     I
## W  1.00 -0.67  0.40  0.70
## C -0.67  1.00 -0.73 -0.38
## F  0.40 -0.73  1.00  0.52
## I  0.70 -0.38  0.52  1.00
```

Association

	W	C	F	I
Worry	1.00			
Concentration	-0.67	1.00		
Fatigue	0.40	-0.73	1.00	
Insomnia	0.70	-0.38	0.52	1.00

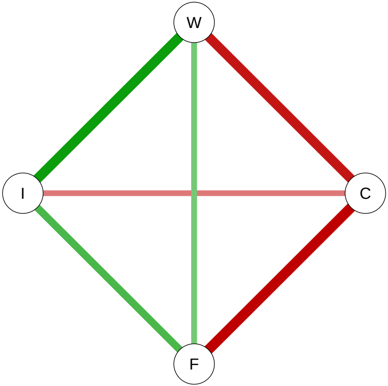
Association

	W	C	F	I
Worry	1.00			
Concentration	-0.67	1.00		
Fatigue	0.40	-0.73	1.00	
Insomnia	0.70	-0.38	0.52	1.00



Association

```
library("qgraph")
qgraph(cor(Data))
```



Association networks...

- Allow for a powerful visualization of the correlational structure

Association networks...

- Allow for a powerful visualization of the correlational structure
- Indicates the presence of latent variables

Association networks...

- Allow for a powerful visualization of the correlational structure
- Indicates the presence of latent variables
- But also include many spurious connections

Association networks...

- Allow for a powerful visualization of the correlational structure
- Indicates the presence of latent variables
- But also include many spurious connections
- And thus not well suited for network analysis

Concentration

We can construct a *concentration* network by predicting all variables from all other variables:

Concentration

We can construct a *concentration* network by predicting all variables from all other variables:

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$

Where $\varepsilon_W \sim N(0, \sigma_W)$ and is assumed to be independent of all *variables* that are not W

Concentration

We can construct a *concentration* network by predicting all variables from all other variables:

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$

Where $\varepsilon_W \sim N(0, \sigma_W)$ and is assumed to be independent of all *variables* that are not W

$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$

Concentration

We can construct a *concentration* network by predicting all variables from all other variables:

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$

Where $\varepsilon_W \sim N(0, \sigma_W)$ and is assumed to be independent of all *variables* that are not W

$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$

$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$

Concentration

We can construct a *concentration* network by predicting all variables from all other variables:

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$

Where $\varepsilon_W \sim N(0, \sigma_W)$ and is assumed to be independent of all *variables* that are not W

$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$

$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$

$$I = \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I$$

```
fitW <- lm(W ~ C + F + I, data = Data)
fitW

##
## Call:
## lm(formula = W ~ C + F + I, data = Data)
##
## Coefficients:
## (Intercept)            C            F
##       6.606       -0.723       -0.564
##           I
##       0.518

fitC <- lm(C ~ W + F + I, data = Data)
fitF <- lm(F ~ W + C + I, data = Data)
fitI <- lm(I ~ W + C + F, data = Data)
```

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$
$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$
$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$
$$I = \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I$$

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$
$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$
$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$
$$I = \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I$$

The regression coefficients have a symmetrical relationship:

$$\beta_{ij}\sigma_j^2 = \beta_{ji}\sigma_i^2$$

```r
beta_WC <- coef(fitW)[2]
beta_CW <- coef(fitC)[2]
sigma_W <- summary(fitW)$sigma
sigma_C <- summary(fitC)$sigma

beta_WC * sigma_C^2

##      C
## -1.163

beta_CW * sigma_W^2

##      W
## -1.163
```

$$
\begin{aligned}
W &= \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W \\
C &= \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C \\
F &= \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F \\
I &= \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I
\end{aligned}
$$

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$
$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$
$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$
$$I = \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I$$

$$
\begin{aligned}
W &= \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W \\
C &= \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C \\
F &= \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F \\
I &= \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I
\end{aligned}
$$

These regression parameters can be standardized to obtain the *partial correlation coefficient*:

$$\rho_{ij} = \frac{\beta_{ij}\sigma_j}{\sigma_i} = \frac{\beta_{ji}\sigma_i}{\sigma_j}$$

$$W = \beta_{21} C + \beta_{31} F + \beta_{41} I + \varepsilon_W$$
$$C = \beta_{12} W + \beta_{32} F + \beta_{42} I + \varepsilon_C$$
$$F = \beta_{13} W + \beta_{23} C + \beta_{43} I + \varepsilon_F$$
$$I = \beta_{14} W + \beta_{24} C + \beta_{34} F + \varepsilon_I$$

These regression parameters can be standardized to obtain the *partial correlation coefficient*:

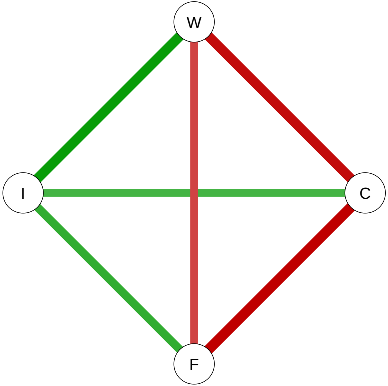
$$\rho_{ij} = \frac{\beta_{ij}\sigma_j}{\sigma_i} = \frac{\beta_{ji}\sigma_i}{\sigma_j}$$

Through mathematical magic, these correspond directly to the negative standardized off-diagonal elements of the inverse of the correlation matrix!

```
beta_WC * sigma_C / sigma_W

##        C
## -0.7651

beta_CW * sigma_W / sigma_C

##        W
## -0.7651

-cov2cor(solve(cov(Data)))

##         W       C       F       I
## W -1.0000 -0.7651 -0.5889  0.7754
## C -0.7651 -1.0000 -0.7993  0.5871
## F -0.5889 -0.7993 -1.0000  0.6468
## I  0.7754  0.5871  0.6468 -1.0000
```

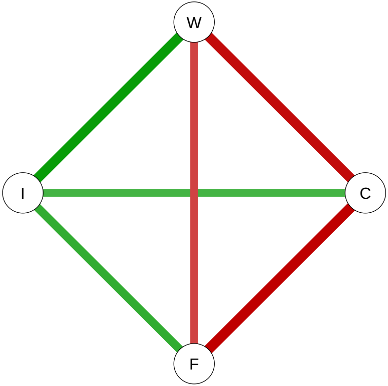
These partial correlations give us the *concentration graph*:

```
library("qgraph")
qgraph(cor(Data), graph = "concentration")
```



These partial correlations give us the *concentration graph*:

```
library("qgraph")
qgraph(cor(Data), graph = "concentration")
```



Why does this still not look good?

$n = 10!$

W	C	F	I
4	2	3	1
1	6	2	1
5	4	4	6
4	2	8	8
2	9	1	3
2	6	2	5
8	2	4	8
1	5	4	1
4	4	5	3
4	2	5	4

Coefficients for Worry:

Table:

	Dependent variable:
	W
C	-0.723^{**} (0.248)
F	-0.564 (0.316)
I	0.518^{**} (0.172)
Constant	6.606^{**} (2.057)
Note:	*p$<$0.1; **p$<$0.05; ***p$<$0.01

Coefficients for Concentration:

Table:

	Dependent variable:
	C
W	-0.810^{**} (0.278)
F	-0.810^{**} (0.249)
I	0.415 (0.234)
Constant	8.451^{***} (0.992)
Note:	*p<0.1; **p<0.05; ***p<0.01

Coefficients for Fatigue:

Table:

	Dependent variable:
	F
W	-0.615 (0.345)
C	-0.789^{**} (0.242)
I	0.451^{*} (0.217)
Constant	7.460^{***} (1.810)
Note:	*p<0.1; **p<0.05; ***p<0.01

Coefficients for Insomnia:

Table:

	Dependent variable:
	I
W	1.161^{**} (0.386)
C	0.830 (0.467)
F	0.927^{*} (0.446)
Constant	-7.071 (4.176)
Note:	$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
    - Problem of multiple comparison
    - Intractable model space

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison
  - Intractable model space
  - *p*-values and NHST are evil

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
    - Problem of multiple comparison
    - Intractable model space
    - *p*-values and NHST are evil
- A well accepted approach for jointly model selection and parameter estimation is the least absolute shrinkage and selection operator (LASSO)

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison
  - Intractable model space
  - *p*-values and NHST are evil
- A well accepted approach for jointly model selection and parameter estimation is the least absolute shrinkage and selection operator (LASSO)
  - Limits the sum of absolute regression weights, which causes insignificant edges to shrink to zero

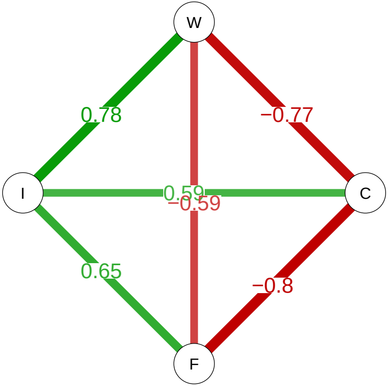
- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison
  - Intractable model space
  - *p*-values and NHST are evil
- A well accepted approach for jointly model selection and parameter estimation is the least absolute shrinkage and selection operator (LASSO)
  - Limits the sum of absolute regression weights, which causes insignificant edges to shrink to zero
  - Useful in any regression case. For graphs the graphical lasso (Friedman et al., 2008) is a very fast algorithm.

- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison
  - Intractable model space
  - *p*-values and NHST are evil
- A well accepted approach for jointly model selection and parameter estimation is the least absolute shrinkage and selection operator (LASSO)
  - Limits the sum of absolute regression weights, which causes insignificant edges to shrink to zero
  - Useful in any regression case. For graphs the graphical lasso (Friedman et al., 2008) is a very fast algorithm.
    - Implemented in **glasso** package (Friedman et al., 2011)

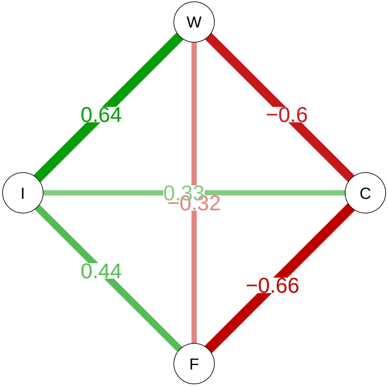
- We are specifically interested in identifying which partial correlations are *zero*
- These elements correspond to missing edges in the graph
- We could identify zeroes by significance testing or stepwise model-selection, but...
  - Problem of multiple comparison
  - Intractable model space
  - *p*-values and NHST are evil
- A well accepted approach for jointly model selection and parameter estimation is the least absolute shrinkage and selection operator (LASSO)
  - Limits the sum of absolute regression weights, which causes insignificant edges to shrink to zero
  - Useful in any regression case. For graphs the graphical lasso (Friedman et al., 2008) is a very fast algorithm.
    - Implemented in **glasso** package (Friedman et al., 2011)
  - Even usable when you have more variables then measures!

glasso penalty: 0

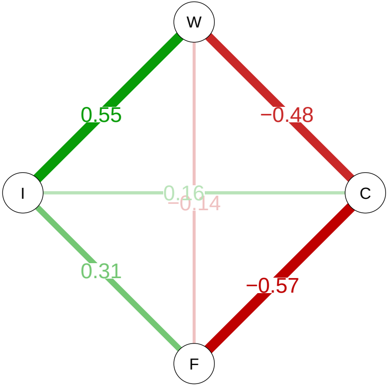
```
## With rho=0, there may be convergence problems if the input mat
##  is not of full rank
```



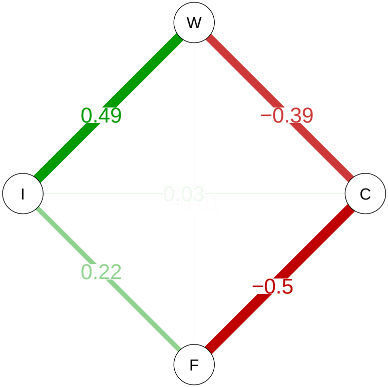
glasso penalty: 0.1



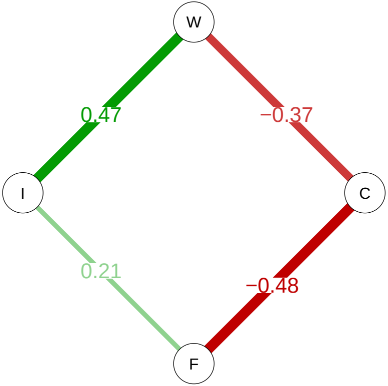
glasso penalty: 0.25



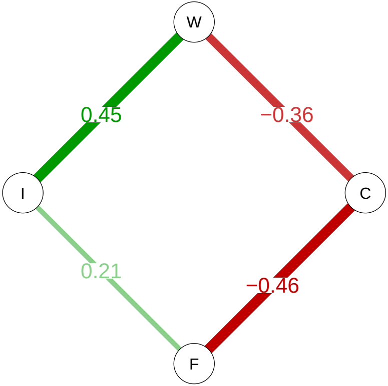
glasso penalty: 0.3



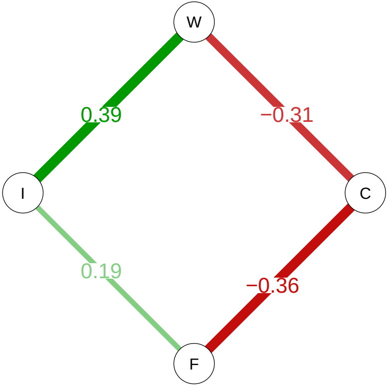
glasso penalty: 0.4



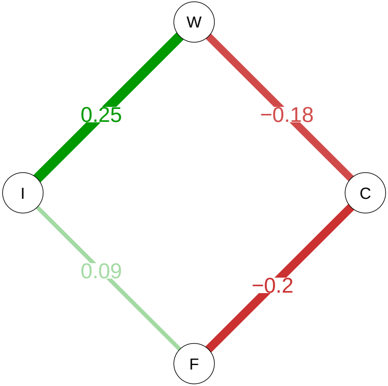
glasso penalty: 0.5



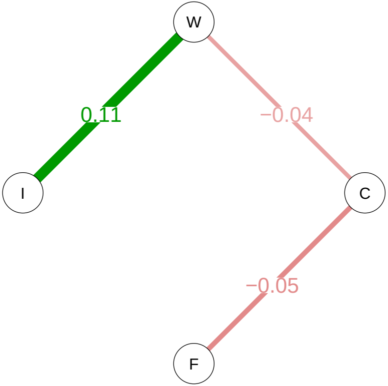
glasso penalty: 1



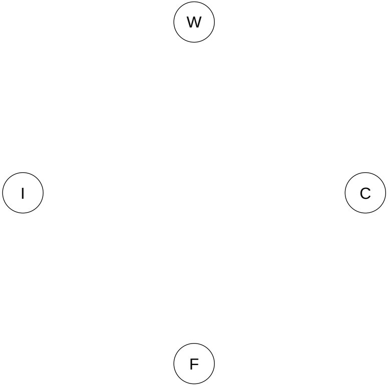
glasso penalty: 2



glasso penalty: 3



glasso penalty: 4

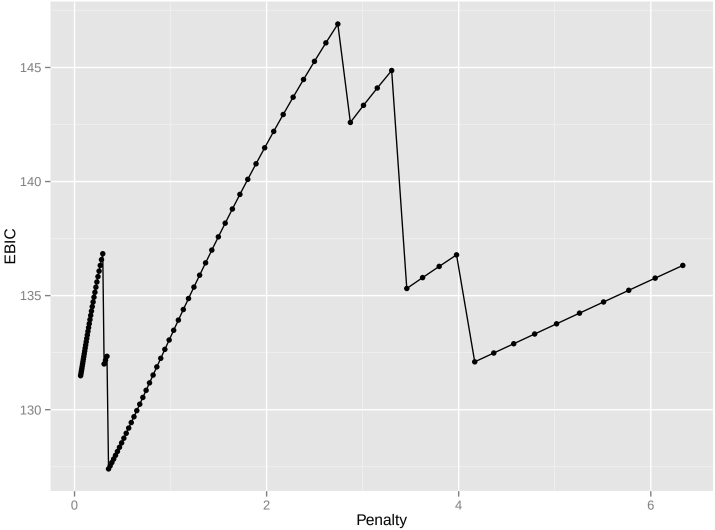


We can compute an information criterion for each penalty parameter, and choose the model with the best (lowest) value. In concentration networks the extended Bayesian Information Criterium (EBIC) works well (Foygel & Drton, 2010):

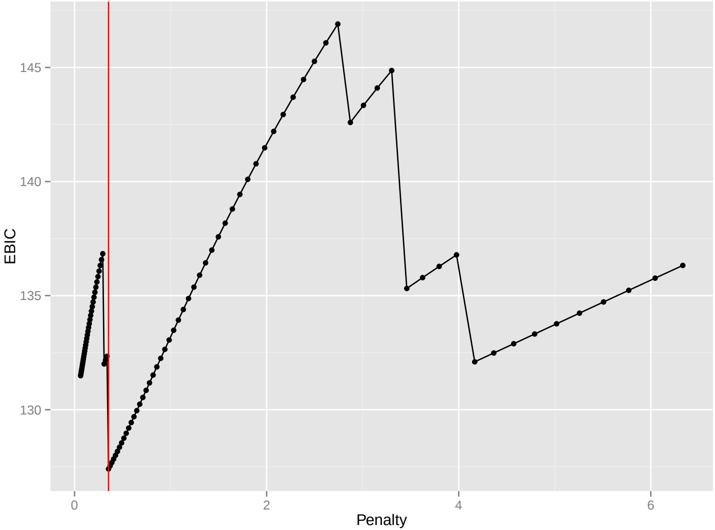
$$EBIC = -2L + |E| \log n + 4|E|\gamma \log p$$

Where L is the log-likelihood, $|E|$ the number of edges in the graph, n the sample size and p the number of variables in the graph. γ is typically set to 0.5 or 0 to obtain regular BIC.
In essence this is the likelihood function with a penalty for the amount of parameters.

Find optimal penalty given extended BIC:

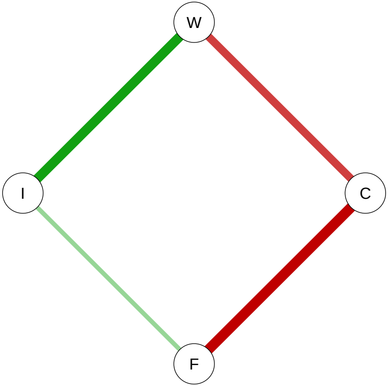


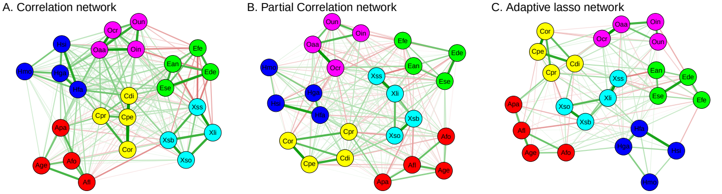
Find optimal penalty given extended BIC:



Find a sparse concentration graph using glasso:

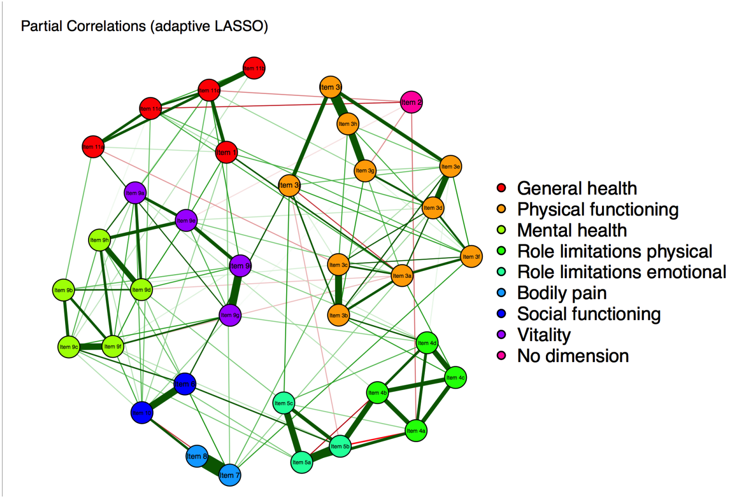
```
qgraph(cor(Data), graph = "glasso", sampleSize = 10)
```





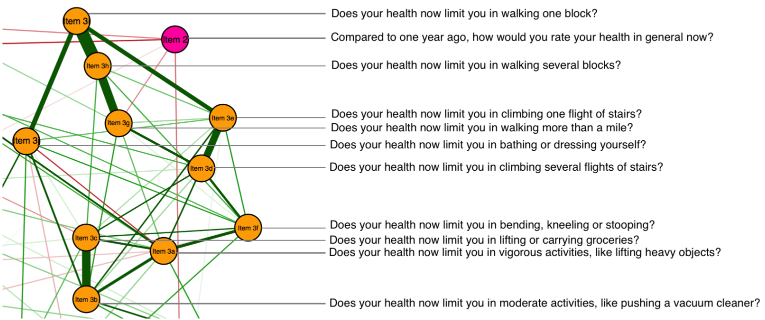
Paper by Giulio Costantini, Sacha Epskamp, Denny Borsboom, Marco Perugini, René Mõttus, Lourens J. Waldorp & Angelique O. J. Cramer submitted

Partial correlations (using adaptive lasso)

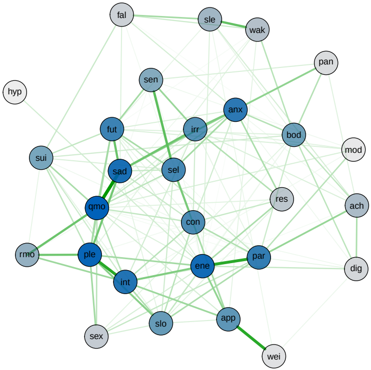


Paper by Jolanda J. Kossakowski, Jacobien M. Kieffer, Sacha Epskamp & Denny Borsboom in preparation

Zooming in: Physical Functioning en Item 2



If the data is binary we can use the **Ising model** to estimate a concentration network. We have implemented a similar method as described in these slides for continuous data in the **IsingFit** package (van Borkulo, Epskamp, & with contributions from Alexander Robitzsch, 2014)



Paper by Claudia D. van Borkulo, Denny Borsboom, Sacha Epskamp Tessa F. Blanken, Lynn Boschloo, Robert A. Schoevers & Lourens J. Waldorp submitted

Time series

- In time-series data, we do not want to predict every node given every other node

Time series

- In time-series data, we do not want to predict every node given every other node
- It is senseless to predict the past from the future, or predict the future given other variables in the future

Time series

- In time-series data, we do not want to predict every node given every other node
- It is senseless to predict the past from the future, or predict the future given other variables in the future
  - We don't care how well a patient being suicidal today predicts how depressed he was yesterday

Time series

- In time-series data, we do not want to predict every node given every other node
- It is senseless to predict the past from the future, or predict the future given other variables in the future
  - We don't care how well a patient being suicidal today predicts how depressed he was yesterday
  - We don't care how well a patient being depressed tomorrow will predict how suicidal he will be tomorrow

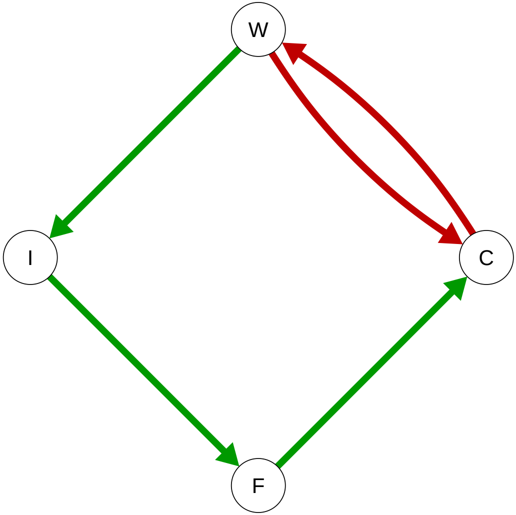
Time series

- In time-series data, we do not want to predict every node given every other node
- It is senseless to predict the past from the future, or predict the future given other variables in the future
  - We don't care how well a patient being suicidal today predicts how depressed he was yesterday
  - We don't care how well a patient being depressed tomorrow will predict how suicidal he will be tomorrow
- Therefore we only set up a predictive model *between time points*, which can be condensed in a *directed network*

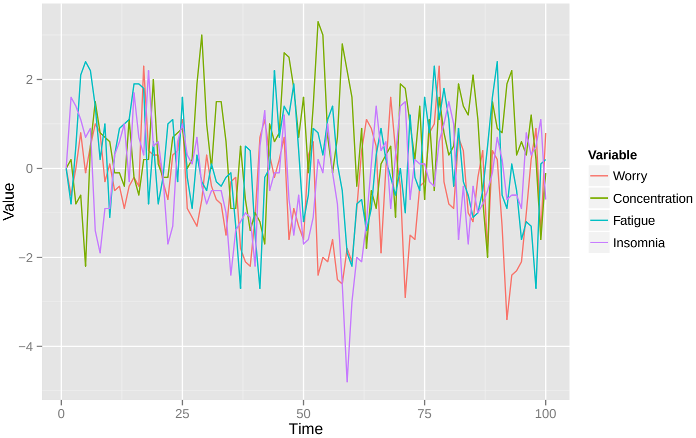
Time series

- In time-series data, we do not want to predict every node given every other node
- It is senseless to predict the past from the future, or predict the future given other variables in the future
  - We don't care how well a patient being suicidal today predicts how depressed he was yesterday
  - We don't care how well a patient being depressed tomorrow will predict how suicidal he will be tomorrow
- Therefore we only set up a predictive model *between time points*, which can be condensed in a *directed network*
  - Looking only between consecutive time points is called a Lag-1 process. We will not look at more complicated models today

True causal model between days



Simulated for 100 days:



```r
head(Data)
```

```
##      W    C    F   I
## 1  0.0  0.0  0.0 0.0
## 2 -0.6  0.2 -0.8 1.6
## 3  0.0 -0.8  0.5 1.4
## 4  0.8 -0.6  2.1 1.1
## 5 -0.1 -2.2  2.4 0.7
## 6  0.5  0.5  2.2 0.9
```

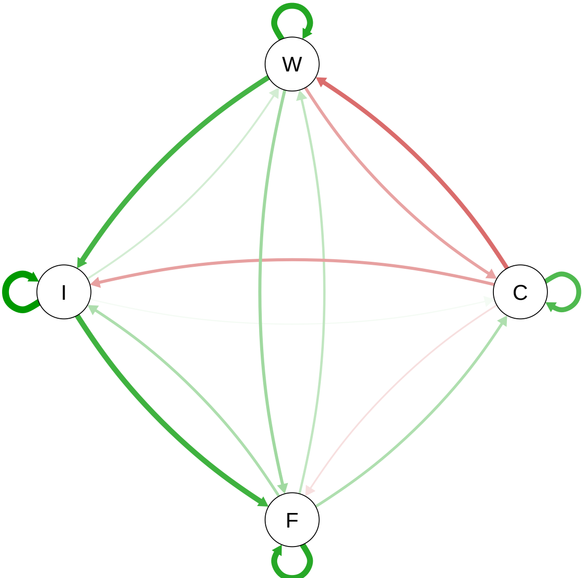
We can compute the correlations between consecutive time-points:

```
corL1 <- cor(Data[-nrow(Data),],Data[-1,])
round(corL1,2)

##        W     C     F     I
## W  0.50 -0.21  0.22  0.42
## C -0.33  0.40 -0.07 -0.21
## F  0.14  0.18  0.49  0.18
## I  0.10  0.02  0.44  0.58
```

Which gives us the Lag-1 correlations, and correspond to the *association* network

Lag-1 correlations

```
qgraph(corL1, layout = "circle")
```



A subject over time...

Concentration

We can again construct a *concentration* network by repeated multiple regressions. This time however we regress *on the previous time point*

$$
\begin{aligned}
W_t &= \beta_{11} W_{t-1} + \beta_{21} C_{t-1} + \beta_{31} F_{t-1} + \beta_{41} I_{t-1} + \varepsilon_W \\
C_t &= \beta_{12} W_{t-1} + \beta_{22} C_{t-1} + \beta_{32} F_{t-1} + \beta_{42} I_{t-1} + \varepsilon_C \\
F_t &= \beta_{13} W_{t-1} + \beta_{23} C_{t-1} + \beta_{33} F_{t-1} + \beta_{43} I_{t-1} + \varepsilon_F \\
I_t &= \beta_{14} W_{t-1} + \beta_{24} C_{t-1} + \beta_{34} F_{t-1} + \beta_{44} I_{t-1} + \varepsilon_I
\end{aligned}
$$

Where $\varepsilon_i \sim N(0, \sigma_i)$

```
fitW <- lm(W[-1] ~ W[-100] + C[-100] + F[-100] + I[-100], data =
fitW

##
## Call:
## lm(formula = W[-1] ~ W[-100] + C[-100] + F[-100] + I[-100], da
##
## Coefficients:
## (Intercept)      W[-100]      C[-100]
##     -0.1490       0.4450      -0.2455
##     F[-100]      I[-100]
##      0.0629      -0.0573

fitC <- lm(C[-1] ~ W[-100] + C[-100] + F[-100] + I[-100], data =
fitF <- lm(F[-1] ~ W[-100] + C[-100] + F[-100] + I[-100], data =
fitI <- lm(I[-1] ~ W[-100] + C[-100] + F[-100] + I[-100], data =
```

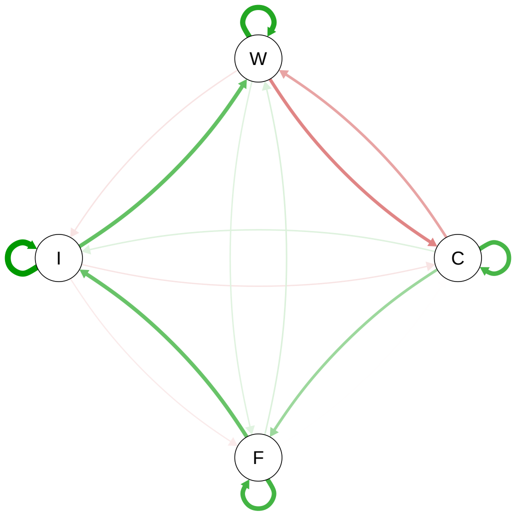
This model is called a Vector-autoregression (VAR) model

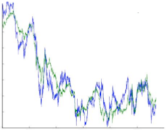
$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{y}_{t-1} + \varepsilon$$

In **qgraph** the `VARglm()` function can be used to estimate the regression parameters. The transpose of the estimates correspond to the graph structure (for now). Model selection and LASSO can be used, but have not yet been implemented automatically in **qgraph**

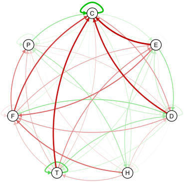
VAR network

```
Beta <- VARglm(Data)$graph
qgraph(t(Beta), layout = "circle", labels = names(Data))
```





$$Y_{ijt} = \alpha + \beta_1 X_{1,t-1} + \beta_2 X_{2,t-1} + \ldots + \varepsilon$$



Bringmann, L., Vissers, N., Wichers, M., Geschwind, N., Kuppens, P., Peeters, F., Borsboom, D., & Tuerlinckx, F. (2013). A network approach to psychopathology: New insights into clinical longitudinal data. *PLoS ONE*.

Network estimation

	Cross-sectional	Time-series
Association	Correlations	Partial correlations*
Concentration	Lag-1 correlations	VAR*

* model selection / LASSO can be used to estimate sparse structure

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance
  - How long does it take for a node to influence another node?

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance
  - How long does it take for a node to influence another node?
- Centrality

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance
  - How long does it take for a node to influence another node?
- Centrality
  - Which node is the most important?

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance
  - How long does it take for a node to influence another node?
- Centrality
  - Which node is the most important?
- Connectivity

Network Methodology

Besides providing a interpretable structure of a complex system, we can also use networks to compute unique measures such as:

- Distance
  - How long does it take for a node to influence another node?
- Centrality
  - Which node is the most important?
- Connectivity
  - How well are nodes connected?

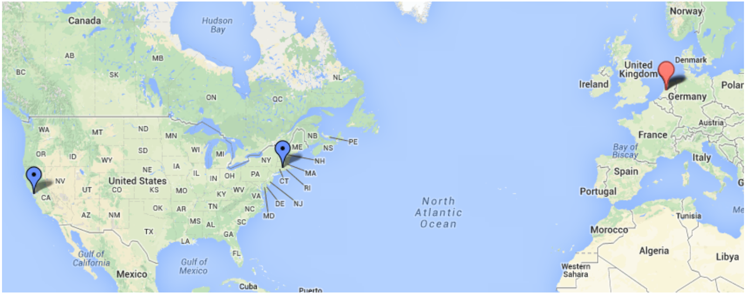
- Each edge can be interpreted as having a *length*

- Each edge can be interpreted as having a *length*
- Length is inversely related to weight

- Each edge can be interpreted as having a *length*
- Length is inversely related to weight
    - Two strongly connected nodes are *closer*: it is easier for information to go from one node to the other

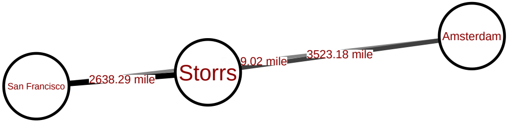
- Each edge can be interpreted as having a *length*
- Length is inversely related to weight
  - Two strongly connected nodes are *closer*: it is easier for information to go from one node to the other
  - The path between two unconnected nodes is infinitely long: you cannot walk directly from one node to the other

- Each edge can be interpreted as having a *length*
- Length is inversely related to weight
  - Two strongly connected nodes are *closer*: it is easier for information to go from one node to the other
  - The path between two unconnected nodes is infinitely long: you cannot walk directly from one node to the other
- In weighted networks the absolute value of edge weights is used in computing the length

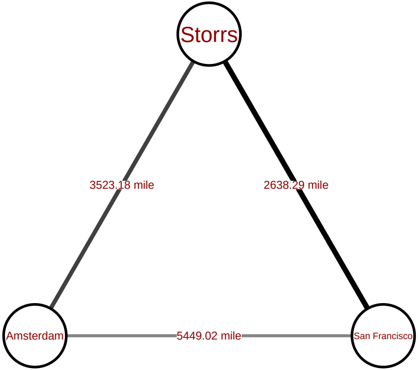

Canada
Hudson Bay
United States
Mexico
Gulf of California
Gulf of Mexico
Cuba
North Atlantic Ocean
Norway
Denmark
United Kingdom
Ireland
Germany
France
Austria
Italy
Bay of Biscay
Spain
Portugal
Morocco
Algeria
Tunisia
Libya
Western Sahara

The distance between Storrs and San Francisco is 2638.29 miles (4245.80 km)

Amsterdam is much farther from San Francisco:



We can place nodes however we want:



Shortest path length

- The distance between two nodes is defined by the length of the shortest possible path between two nodes

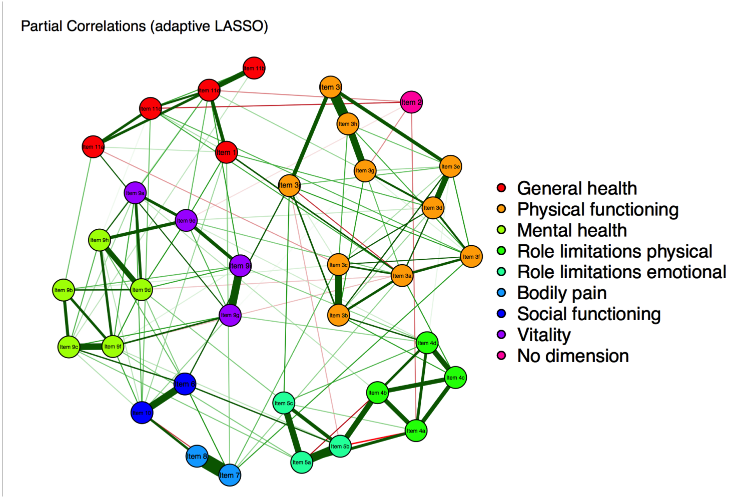
Shortest path length

- The distance between two nodes is defined by the length of the shortest possible path between two nodes
- In unweighted graphs this is simply the amount of edges on a path

Shortest path length

- The distance between two nodes is defined by the length of the shortest possible path between two nodes
- In unweighted graphs this is simply the amount of edges on a path
- In weighted graphs the length of a path is computed as the sum of all edge lengths on a path

Which node is the most influential?



Centrality

A node is central/important/influential if...

- ...it has many connections

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness
  - In concentration networks the node that (indirectly) has the best predictive value on all other nodes

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness
  - In concentration networks the node that (indirectly) has the best predictive value on all other nodes
- ...it *connects* other nodes

Centrality

A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness
  - In concentration networks the node that (indirectly) has the best predictive value on all other nodes
- ...it *connects* other nodes
  - Betweenness

Centrality

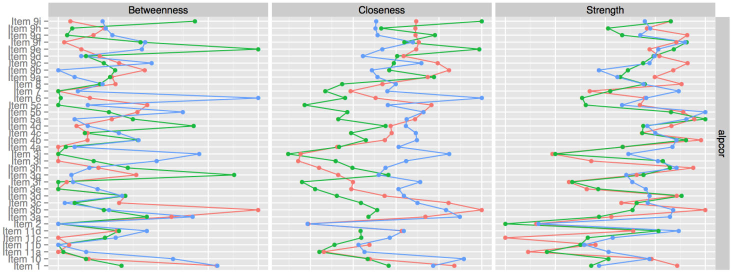
A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness
  - In concentration networks the node that (indirectly) has the best predictive value on all other nodes
- ...it *connects* other nodes
  - Betweenness
- ...it is connected to other important nodes

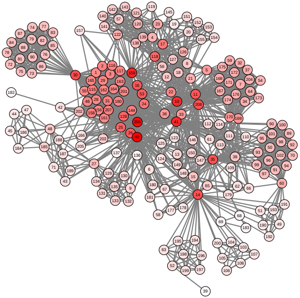
Centrality

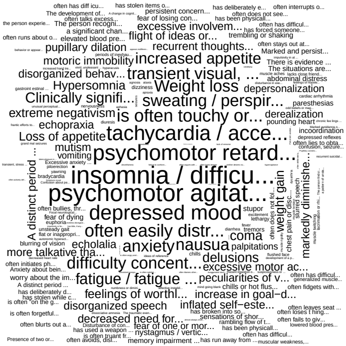
A node is central/important/influential if...

- ...it has many connections
  - Degree / strength
  - In concentration network the node with the highest strength directly predicts the most other nodes
- ...it is *close* to all other nodes
  - Closeness
  - In concentration networks the node that (indirectly) has the best predictive value on all other nodes
- ...it *connects* other nodes
  - Betweenness
- ...it is connected to other important nodes
  - Eigenvector centrality

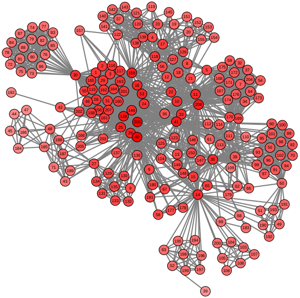

Betweenness
Closeness
Strength
alpcor
Item 9i
Item 9h
Item 9g
Item 9f
Item 9e
Item 9d
Item 9c
Item 9b
Item 9a
Item 8
Item 7
Item 6
Item 5c
Item 5b
Item 5a
Item 4d
Item 4c
Item 4b
Item 4a
Item 3j
Item 3i
Item 3h
Item 3g
Item 3f
Item 3e
Item 3d
Item 3c
Item 3b
Item 3a
Item 2
Item 11d
Item 11c
Item 11b
Item 11a
Item 10
Item 1

Degree




insomnia / difficu...
psychomotor agitat...
psychomotor retard...
depressed mood
tachycardia / acce...
is often touchy or...
often easily distr...
difficulty concent...
sweating / perspir...
Weight loss
transient visual, ...
increased appetite
Clinically signifi...
extreme negativism
Loss of appetite
fatigue / fatigue ...
anxiety
nausua
coma
delusions
weight gain
markedly diminishe...
A distinct period ...
feelings of worthl...
increase in goal−d...
peculiarities of v...
excessive motor ac...
disorganized speech
inflated self−este...
decreased need for...
fear of one or mor...
Hypersomnia
disorganized behav...
motoric immobility
pupillary dilation
recurrent thoughts...
flight of ideas or...
excessive involvem...
depersonalization
derealization
echopraxia
echolalia
mutism
vomiting
more talkative tha...
palpitations
incoordination
paresthesias
abdominal distress
memory impairment ...
nystagmus / vertic...

Closeness

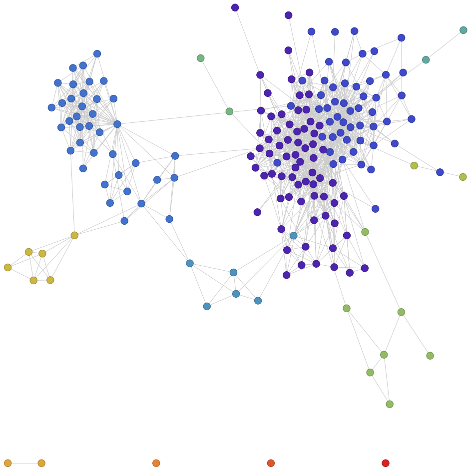



echopraxia
extreme negativism
palpitations
pupillary dilation
excitement
Hypersomnia
vomiting
yawning
ataxia
delusions
Weight loss
diarrhea
stupor
mutism
difficulty concent...
feelings of worthl...
weight gainnausua
fever
chills
tremors
depressed mood
anxiety
increased appetite
insomnia / difficu...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

Small world

The famous paper of Watts and Strogatz (1998)—already cited 22691 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

A graph exhibits a small world if it has both high clustering and a short average path length. This information can be summarized in a *small-world index*, which should be higher than 3.

Network Analysis with **qgraph**

What is **R**?

- **R** is a statistical programming language
  - Statistical analysis
  - Data visualization
  - Data mining
  - General programming
- It is *open-source*
  - Free as in "free beer" and "free speech"
  - Large active community around **R**
  - Many contributed packages

What is **R**?

R can be downloaded from `http://cran.r-project.org/`. A good environment for R is RStudio which can be downloaded from `https://www.rstudio.com/`. Some good links to start learning R are:

- `http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf`
- `https://www.codeschool.com/courses/try-r`

R uses functions, such as `rnorm()`, which take some arguments between the parantheses. All these functions are documented using `?`. For example, `rnorm` generates random numbers:

```
# rnorm samples random normal numbers:
rnorm(3)

## [1]  1.59291  0.04501 -0.71513
```

Multiple arguments can be used. They can also be named. See the documentation for a list of arguments!

```
?rnorm
```

```
# Using arguments we can change its behavior. e.g.,
rnorm(3, mean = 100, sd = 15)

## [1] 113.0 116.1 128.4
```

The <- symbol can be used to store results and reuse them later.

```
# We can use '<-' to store the result:
result <- rnorm(3)
result

## [1] -0.6030 -0.3909 -0.4162

# Reuse:
mean(result)

## [1] -0.47
```

Packages

- Packages are extensions contributed to **R** containing extra functions
- They can be installed using `install.packages()`
- Afterwards they can be loaded using `library()`

Packages

```r
# Install package
install.packages("qgraph", dep = TRUE)

# Load package
library("qgraph")
```

Load data into **R**

- There are many ways to load data into **R**
- The most common way is to read a plain text file (which can be exported from excel for instance) using `read.table` or `read.csv` (see their help files for how to do this)
- Because psychologists often use **SPSS**, it is useful to directly import data from **SPSS**
- This can be done using `read.spss()` in the `foreign` package.
- `file.choose()` can be used to select a file (it might be opened in the background of RStudio)

Load SPSS data into **R**

```r
# Load package "foreign":
library("foreign")

# Select a SPSS file:
file <- file.choose()

# Read data:
Data <- read.spss(file,to.data.frame=TRUE)
```

Rather than using a file as data, we will use a dataset already included in R via the **psych** package (Revelle, 2010):

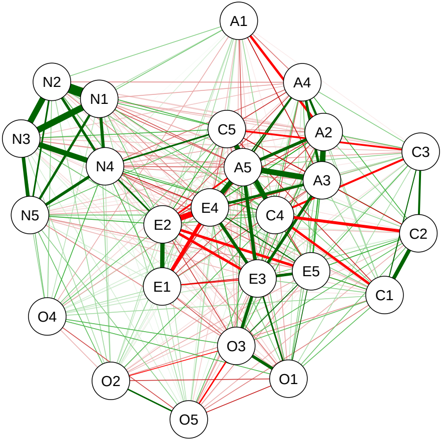
```
library("psych")
data(bfi)
bfi <- bfi[,1:25]

str(bfi)

## 'data.frame': 2800 obs. of  25 variables:
##  $ A1: int  2 2 5 4 2 6 2 4 4 2 ...
##  $ A2: int  4 4 4 4 3 6 5 3 3 5 ...
##  $ A3: int  3 5 5 6 3 5 5 1 6 6 ...
##  $ A4: int  4 2 4 5 4 6 3 5 3 6 ...
##  $ A5: int  4 5 4 5 5 5 5 1 3 5 ...
##  $ C1: int  2 5 4 4 4 6 5 3 6 6 ...
##  $ C2: int  3 4 5 4 4 6 4 2 6 5 ...
##  $ C3: int  3 4 4 3 5 6 4 4 3 6 ...
##  $ C4: int  4 3 2 5 3 1 2 2 4 2 ...
##  $ C5: int  4 4 5 5 2 3 3 4 5 1 ...
##  $ E1: int  3 1 2 5 2 2 4 3 5 2 ...
##  $ E2: int  3 1 4 3 2 1 3 6 3 2 ...
##  $ E3: int  3 6 4 4 5 6 4 4 NA 4 ...
##  $ E4: int  4 4 4 4 4 5 5 2 4 5 ...
```

This is a dataset measuring the big 5 personality traits using an ordinal scale. The **qgraph** function `cor_auto()` can be used to compute an appropriate correlation matrix. In this case, polychoric correlations:

```
cor_bfi <- cor_auto(bfi)

## Variables detected as ordinal:  A1; A2; A3; A4;
A5; C1; C2; C3; C4; C5; E1; E2; E3; E4; E5; N1; N2;
N3; N4; N5; O1; O2; O3; O4; O5
```

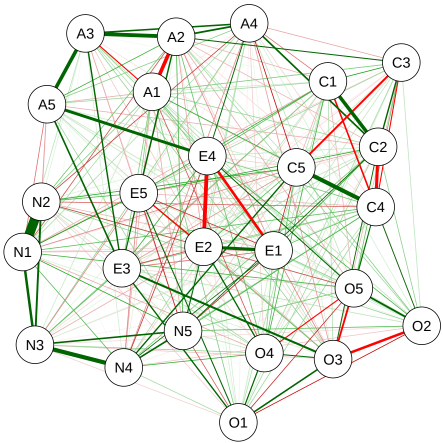
We can send this correlation matrix to `qgraph()` to plot the association network. To use a force-embedded algorithm we set the layout argument to `"spring"`

```
graph_cor <- qgraph(cor_bfi, layout = "spring")
```



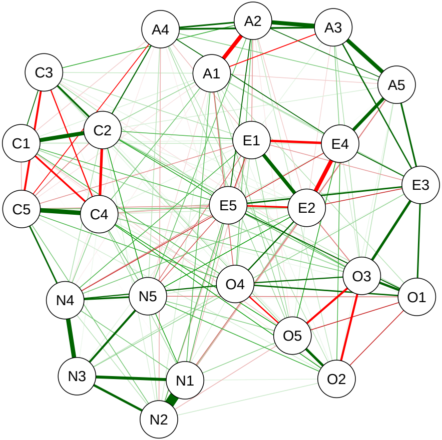
The partial correlation graph can be plotted by setting the `graph` argument to `"concentration"`

```
graph_pcor <- qgraph(cor_bfi, graph = "concentration", layout = "
```



To use the graphical lasso to estimate a sparse structure set `graph` to `"glasso"` and assign the sample size:

```
graph_glas <- qgraph(cor_bfi, graph = "glasso", sampleSize = nrow
```

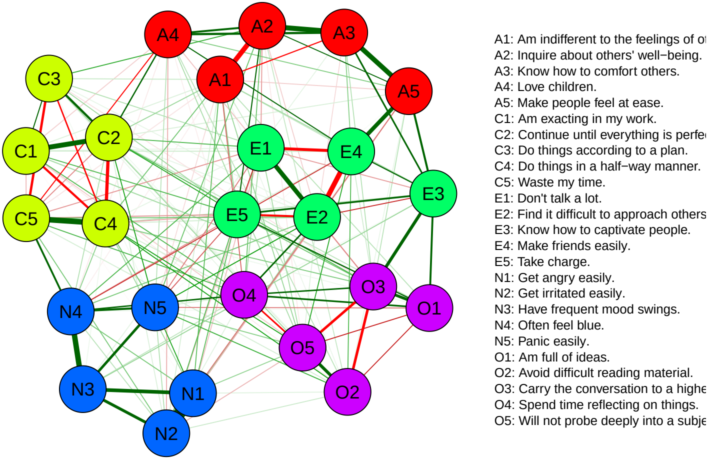


With legend:

```
Names <- scan("http://sachaepskamp.com/files/BFIitems.txt",
              what = "character", sep = "\n")

Groups <- rep(c('A','C','E','N','O'),each=5)
```

With legend:

```r
qgraph(cor_bfi, graph = "glasso", sampleSize = nrow(bfi),
       layout = "spring", nodeNames = Names,
       legend.cex =0.6, groups = Groups)
```



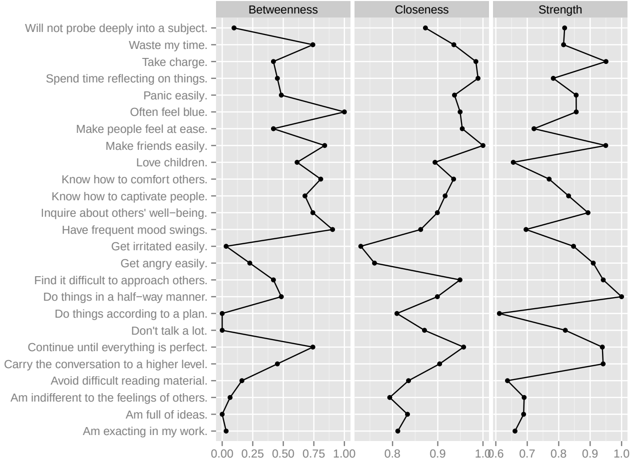
A graph can be saved using `filetype` and `filename`. PDF files produce the best results:

```
qgraph(cor_bfi, graph = "glasso",
    sampleSize = nrow(bfi), layout = "spring",
    filetype = "pdf", filename = "glasso")

## Output stored in
/home/sacha/sacha.epskamp@gmail.com/Documents/Work/Talks/APS/Shee
```

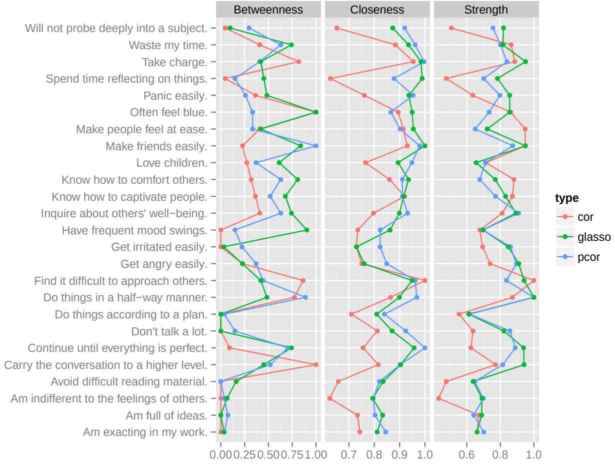
The `centrality_auto()` function automatically computes centrality measures and shortest path lengths:

```
cent <- centrality_auto(graph_glas)
names(cent)

## [1] "node.centrality"
## [2] "edge.betweenness.centrality"
## [3] "ShortestPathLengths"
```

```
centralityPlot(graph_glas, labels=Names)
```



```r
centralityPlot(list(cor = graph_cor, pcor = graph_pcor, glasso =
```



The `smallworldness()` function computes the smallworldness index:

```
smallworldness(graph_glas)

##        smallworldness
##                1.0105
##          trans_target
##                0.6358
## averagelength_target
##                1.3833
##           trans_rnd_M
##                0.6292
##          trans_rnd_lo
##                0.6131
##          trans_rnd_up
##                0.6438
##  averagelength_rnd_M
##                1.3833
## averagelength_rnd_lo
##                1.3833
## averagelength_rnd_up
##                1.3833
```

Concluding comments

More information is on our website: `http://psychosystems.org/` and the developmental version of **qgraph** is on GitHub `https://github.com/SachaEpskamp/qgraph`

Thank you for your attention!

References I

Borsboom, D., Cramer, A. O., Schmittmann, V. D., Epskamp, S., & Waldorp, L. J. (2011). The small world of psychopathology. *PloS one*, *6*(11), e27407.

Costa, P. T., & McCrae, R. R. (1992). *Revised NEO Personality Inventory and NEO Five-Factor Inventory*. Odessa, FL: Psychological Assessment Services.

Cramer, A. O., Waldorp, L. J., van der Maas, H. L., Borsboom, D., et al. (2010). Comorbidity: A network perspective. *Behavioral and Brain Sciences*, *33*(2-3), 137–150.

References II

Dolan, C., Oort, F., Stoel, R., & Wicherts, J. (2009). Testing Measurement Invariance in the Target Rotated Multigroup Exploratory Factor Model. *Structural Equation Modeling: A Multidisciplinary Journal*, *16*(2), 20.

Doosje, B., Loseman, A., & Bos, K. (2013). Determinants of radicalization of islamic youth in the netherlands: Personal uncertainty, perceived injustice, and perceived group threat. *Journal of Social Issues*, *69*(3), 586–604.

Foygel, R., & Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. In *Nips* (pp. 604–612).

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

References III

Friedman, J., Hastie, T., & Tibshirani, R. (2011). glasso: Graphical lasso- estimation of gaussian graphical models [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=glasso` (R package version 1.7)

Hoekstra, H., de Fruyt, F., & Ormel, J. (2003). *Neo-persoonlijkheidsvragenlijsten: NEO-PI-R, NEO-FFI.* Lisse: Swets Test Services.

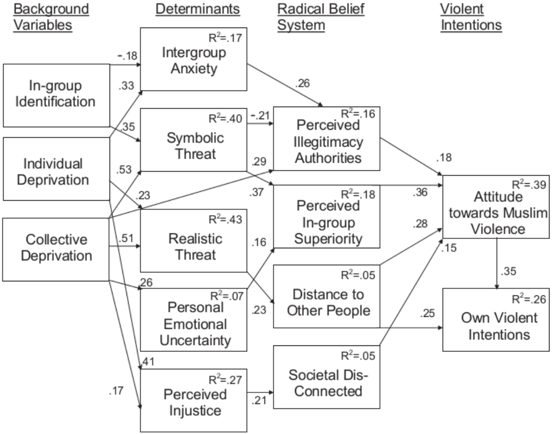
Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, *10*, 2295–2328.

References IV

Revelle, W. (2010). **psych**: Procedures for psychological, psychometric, and personality research [Computer software manual]. Evanston, Illinois. Retrieved from `http://personality-project.org/r/psych.manual.pdf` (R package version 1.0-93)

van Borkulo, C., Epskamp, S., & with contributions from Alexander Robitzsch. (2014). Isingfit: Fitting ising models using the elasso method [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=IsingFit` (R package version 0.2.0)

Watts, D., & Strogatz, S. (1998). Collective dynamics of small-world networks. *Nature*, *393*(6684), 440–442.



Doosje, Loseman, and Bos (2013)

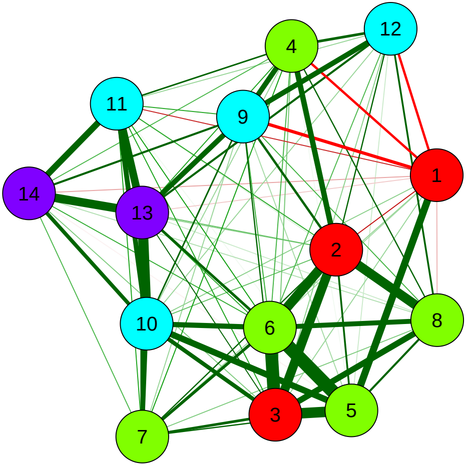
Table 1. The Means, Standard Deviations, and Inter-Correlations of All the Constructs

	M	*SD*	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Identification	4.56	0.85	–	−.19*	.08	−.25*	.42*	.07	.08	−.06	−.28*	.09	−.17	−.25*	−.04	−.07
2. Ind. Rel. Depri.	2.39	0.81		–	.49*	.36*	.23*	.50*	.21*	.50*	.25*	.12	.17	.21*	.12	.09
3. Col. Rel. Depri.	3.31	0.92			–	.11	.54*	.62*	.26*	.38*	.21	.31*	.18*	.09	.20*	.10
4. Int. Anxiety	−0.20	0.17				–	.01	.15	.19*	.21*	.35*	.08	.22*	.26*	.18*	.14
5. Symbolic Threat	3.46	0.76					–	.64*	.21*	.24*	.07	.39*	.01	.04	.17	−.01
6. Realistic Threat	3.10	0.88						–	.27*	.34*	.16	.35*	.19*	.14	.26*	.16
7. Per. Em. Uncertain.	2.84	0.67							–	.10	.08	.29*	.18	.00	.30*	.14
8. Perc. Proc. Injustice	2.38	0.68								–	.15	.01	.03	.23*	.04	.06
9. Perc. Illegitimacy	2.37	0.02									–	.22*	.17*	.35*	.35*	.24*
10. Perc. Ingr. Super.	3.26	0.93										–	.34*	.08	.53*	.30*
11. Distance	2.32	0.66											–	.08	.44*	.39*
12. Disconnected	2.79	0.96												–	.24*	.00
13. Moslim Violence	2.89	1.06													–	.47*
14. Violent Intentions	2.08	0.91														–

Note. 2 = *Individual Relative Deprivation*, 3 = *Collective Relative Deprivation*, 4 = *Intergroup Anxiety*, 5 = *Symbolic Threat*, 6 = *Realistic Threat*, 7 = *Personal Emotional Uncertainty*, 8 = *Perceived Procedural Injustice*, 9 = *Perceived Illegitimacy*, 10 = *Perceived Ingroup Superiority*. $^{*}p < .05$.

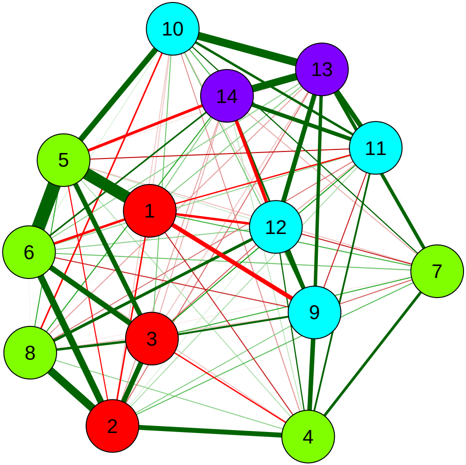
Doosje et al. (2013)

Correlation network



1: In−group Identification
2: Individual Deprivation
3: Collective Deprivation
4: Intergroup Anxiety
5: Symbolic Threat
6: Realistic Threat
7: Personal Emotional Uncertainty
8: Perceived Injustice
9: Perceived Illegitimacy authorities
10: Perceived In−group superiority
11: Distance to Other People
12: Societal Disconnected
13: Attitude towards Muslim Violence
14: Own Violent Intentions

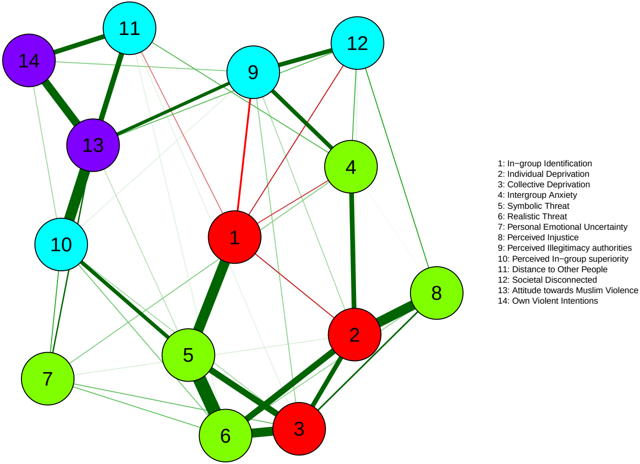
Partial correlation network



1: In−group Identification
2: Individual Deprivation
3: Collective Deprivation
4: Intergroup Anxiety
5: Symbolic Threat
6: Realistic Threat
7: Personal Emotional Uncertainty
8: Perceived Injustice
9: Perceived Illegitimacy authorities
10: Perceived In−group superiority
11: Distance to Other People
12: Societal Disconnected
13: Attitude towards Muslim Violence
14: Own Violent Intentions

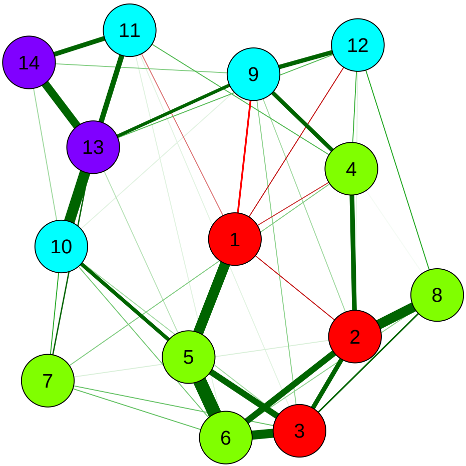
Partial correlation network

After glasso:



Partial correlation network

After glasso (Friedman et al., 2011):



1: In−group Identification
2: Individual Deprivation
3: Collective Deprivation
4: Intergroup Anxiety
5: Symbolic Threat
6: Realistic Threat
7: Personal Emotional Uncertainty
8: Perceived Injustice
9: Perceived Illegitimacy authorities
10: Perceived In−group superiority
11: Distance to Other People
12: Societal Disconnected
13: Attitude towards Muslim Violence
14: Own Violent Intentions

Shortest path length

	Muslim Violence	Violent Intentions
In-group Identification	16.12	19.89
Individual Deprivation	20.10	23.87
Collective Deprivation	17.05	20.82
Intergroup Anxiety	Inf	Inf
Symbolic Threat	11.44	15.21
Realistic Threat	14.81	18.58
Personal Emotional Uncertainty	10.73	14.50
Perceived Injustice	24.83	28.60
Perceived Illegitimacy authorities	Inf	Inf
Perceived In-group superiority	3.16	6.93
Distance to Other People	5.70	8.81
Societal Disconnected	9.15	12.92
Attitude towards Muslim Violence	0.00	3.77
Own Violent Intentions	3.77	0.00

Shortest path length

	Muslim Violence	Violent Intentions
In-group Identification	16.12	19.89
Individual Deprivation	20.10	23.87
Collective Deprivation	17.05	20.82
Intergroup Anxiety	Inf	Inf
Symbolic Threat	11.44	15.21
Realistic Threat	14.81	18.58
Personal Emotional Uncertainty	10.73	14.50
Perceived Injustice	24.83	28.60
Perceived Illegitimacy authorities	Inf	Inf
Perceived In-group superiority	3.16	6.93
Distance to Other People	5.70	8.81
Societal Disconnected	9.15	12.92
Attitude towards Muslim Violence	0.00	3.77
Own Violent Intentions	3.77	0.00

Extra: Interpreting **qgraph**

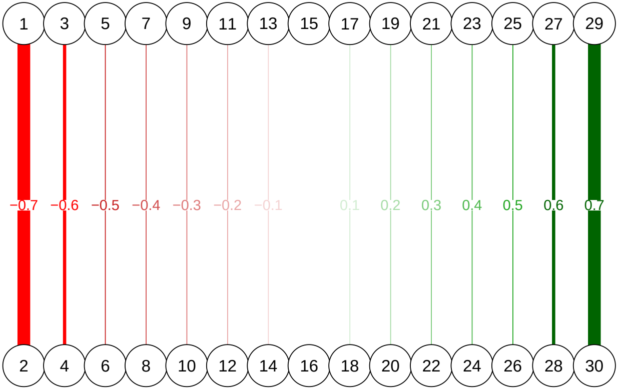
- Under the default coloring scheme, positive edge weights (here correlations) are shown as green edges and negative edge weights as red.
- An edge weight of 0 is omitted. The wider and more colorful an edge the stronger the edge weight.

Interpreting **qgraph**

To interpret **qgraph** networks, three values need to be known:

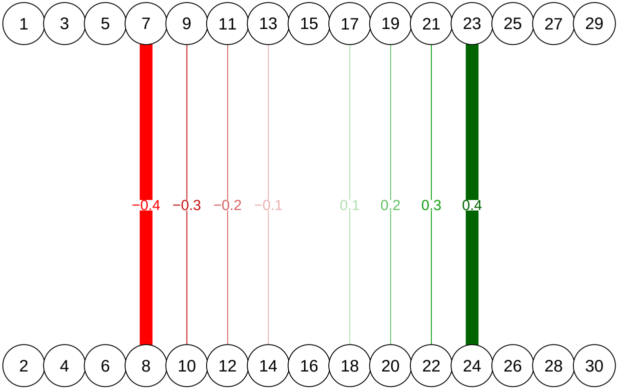
- **Minimum**: Edges with absolute weights under this value are omitted
- **Cut**: If specified, splits scaling of width and color
- **Maximum**: If set, edge width and color scale such that an edge with this value would be the widest and most colorful

Interpreting **qgraph**



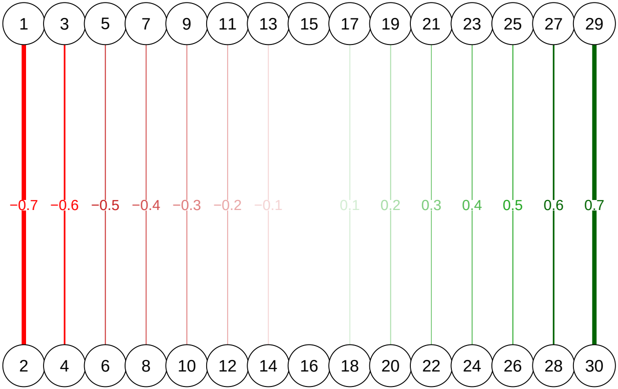
Default settings

Interpreting **qgraph**



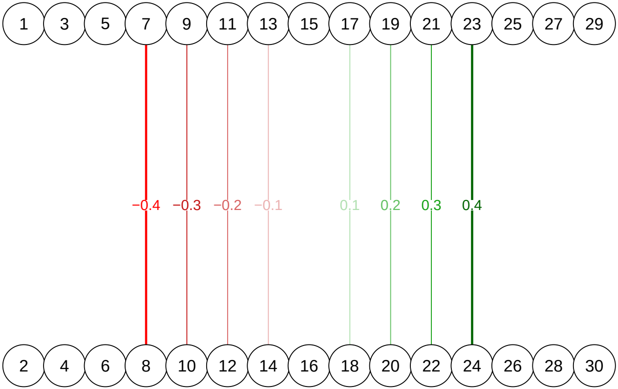
Default settings

Interpreting **qgraph**



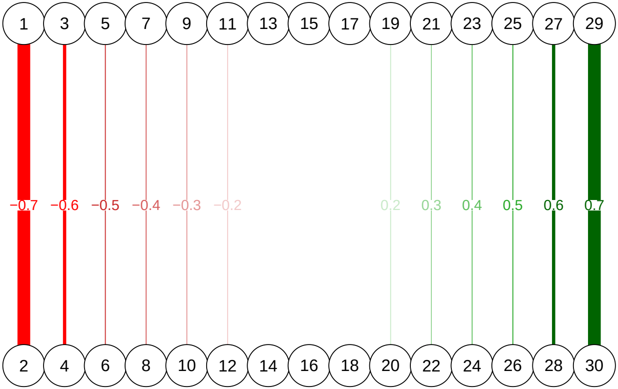
Maximum 1

Interpreting **qgraph**



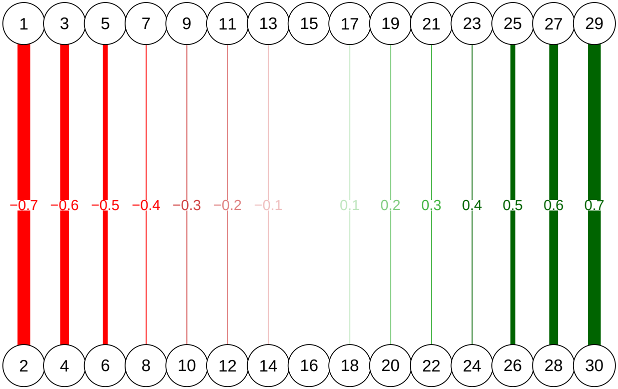
Maximum 1

Interpreting **qgraph**



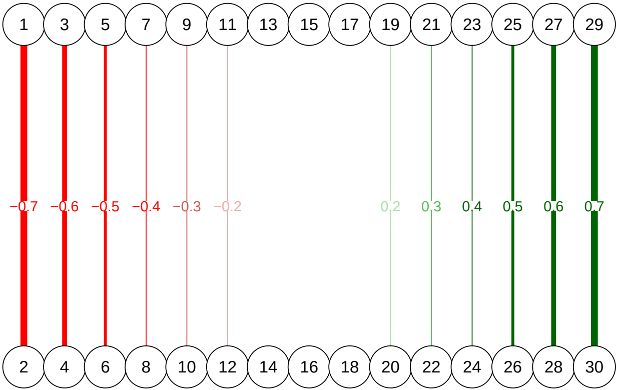
Minimum 0.1

Interpreting **qgraph**



Cut 0.4

Interpreting **qgraph**



Minimum 0.1

Cut 0.4

Maximum 1