JADS Jheronimus Academy of Data Science

SNA4DS

# SOCIAL NETWORK ANALYSIS for DATA SCIENTISTS

today's menu: LECTURE: Network Statistics (LECTURE WEEK 02)

Your lecturer: Roger

Playdate: September 10, 2025

# The course sofar

We have performed descriptive and exploratory analysis:

- vertex level indices (betweenness, closeness, pagerank, etc.)
- graph level indices (centralization, components/subgroups, et cetera)
- Getting to know R with `snafun`, `network` / `sna`, and `igraph`

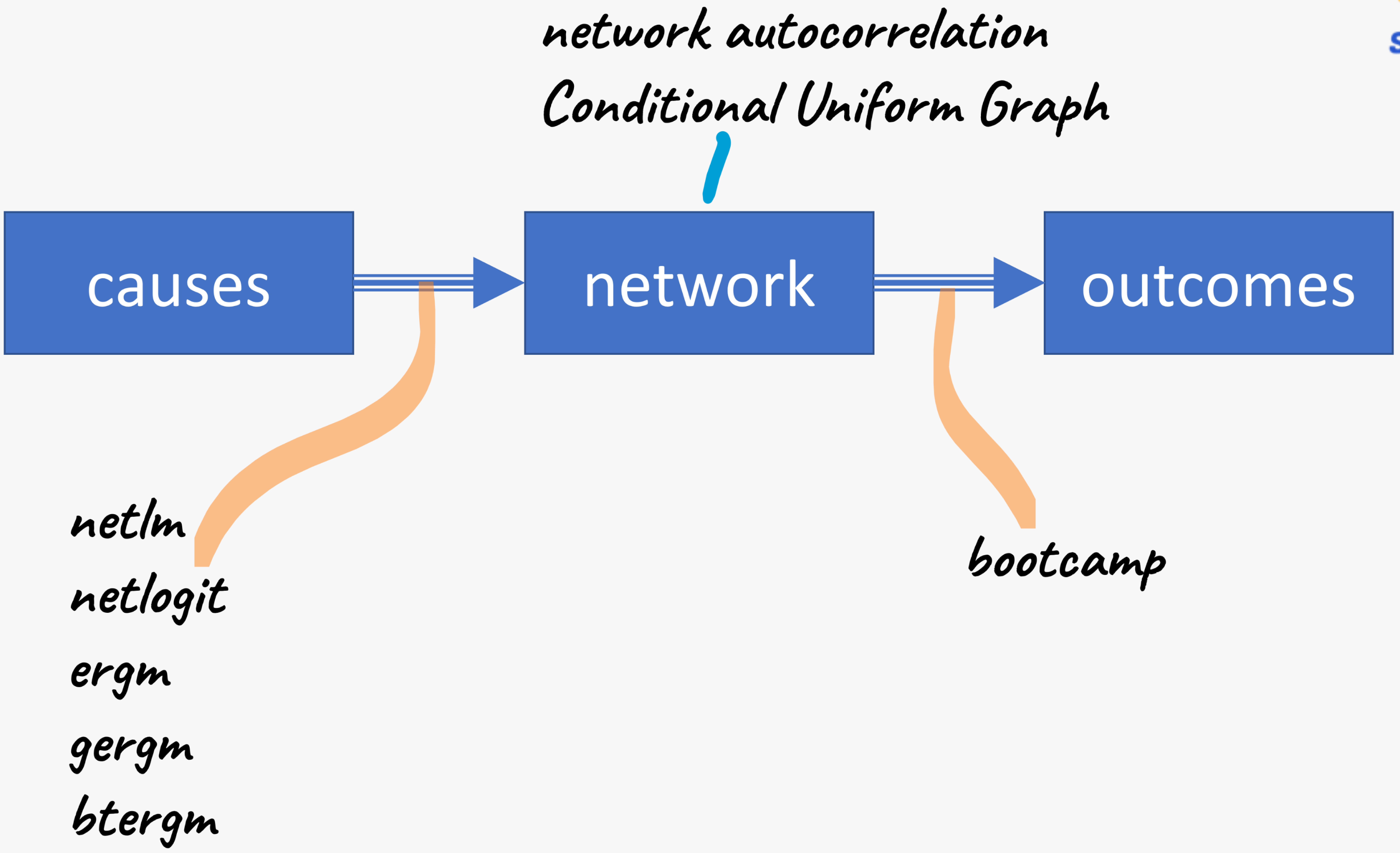
# From exploration to estimation and testing

- You found *betweenness centralization* of .2. Is this high? Low? Expected? Unexpected? Statistically significant?
- You found a 4 communities in the graph. Is this high? Low? Expected? Unexpected? Statistically significant?
- You found that more central companies innovate more. But is this expected? Unexpected? Statistically significant?
- You found a .4 correlation between friendship and advice-giving. Statistically significant?
- How to conclude if friendship promotes advice giving?

➔ we need statistics: *parameter estimation* and *hypothesis testing*.

# We assume that you:

- can easily calculate the appropriate vertex-level and graph-level indices
- are comfortable with the basics of statistics (see: *bootcamp*):
    - linear regression
    - logistic regression
    - null-hypothesis testing


network autocorrelation
Conditional Uniform Graph
causes
network
outcomes
netlm
netlogit
ergm
gergm
btergm
bootcamp

# Network statistics: Social Influence

Everything is related to everything else,
but near things are more related
than distant things.

# Tobler's first law

The law basically says two things:

- You interact more often/strongly with whom is "near"
- Near things are more alike than distant things

# Social influence on a network

## Network autocorrelation model

$$y = \rho W y + X\beta + \epsilon$$

# Social influence on a network

## Network autocorrelation model

vertex attribute

$$y = \underbrace{\rho W}_{\text{social influence}} y + \underbrace{X\beta}_{\text{intrinsic value}} + \epsilon$$

$\rho W y$ captures social influence, it moves people away from their intrinsic positions.

# Social influence on a network

[redacted]

$$y = [redacted] X\beta + \epsilon$$

This is the well-known OLS model, without any network effect (so, when: $\rho = 0$).

# Social influence on a network

## Network autocorrelation model

$$y = \rho W y + X\beta + \epsilon$$

$$y_1 = \rho(W_{12}y_2 + W_{13}y_3 + W_{14}y_4 + \ldots + W_{1g}y_g) + X_{11}\beta_1 + X_{12}\beta_2 + \ldots + + X_{1k}\beta_k + \epsilon_1$$

$$y_2 = \rho(W_{21}y_1 + W_{23}y_3 + W_{24}y_4 + \ldots + W_{2g}y_g) + X_{21}\beta_1 + X_{22}\beta_2 + \ldots + + X_{2k}\beta_k + \epsilon_2$$

$$\vdots$$

$$y_g = \rho(W_{g1}y_2 + W_{g3}y_3 + W_{g4}y_4 + \ldots + W_{g-1,2}y_g) + X_{g1}\beta_1 + X_{g2}\beta_2 + \ldots + + X_{gk}\beta_k + \epsilon_g$$

$g$ is the number of actors in the network, $X$ a matrix of $k$ explanatory variables, $\rho$ a scalar comparable to a correlation coefficient.

# *Weight matrix W*

The matrix *W* captures the social influence process you want to test.

Cell (*i*, *j*) captures the weight of the influence that *i* receives from *j*.

The weight matrix is usually (but not necessarily) **row-normalized** so each row adds to 1 exactly.

What does row normalization mean *substantively* here?

Mathematically, row or column normalization has the advantages of making the eigenvalues behave nicely, so the likelihood become nice and smooth.

It also restricts $rho$ to the $(-1, 1)$ interval, making it easy to interpret.

# Two views on how people influence each other:

- communication
- comparison

# Communication

*Communication* refers to social influence through the **direct contact** between ego and alter.

- The more frequent and vivid the communication between ego and alter, the more likely it is that ego will adopt alter's ideas and beliefs.
- Through discussing matters with alter, ego comes to an understanding of an issue and adds new information to his own, which may cause ego to develop similar attitudes and understandings.

# Comparison

- Ego compares himself to those alters whom he considers **similar** to him in relevant respects, asking himself ‘what would that other person do if he were in my shoes?’
- Ego perceives (or assesses) alter’s behavior and assumes that behavior to be the ‘correct’ behavior for ‘a-person-like-me’ or for ‘a-person-in-a-position-like-mine’.

This is the world of "influencers" and "role models"

(whether they are aware of having this role or not)

# *Operationalizing* COMMUNICATION

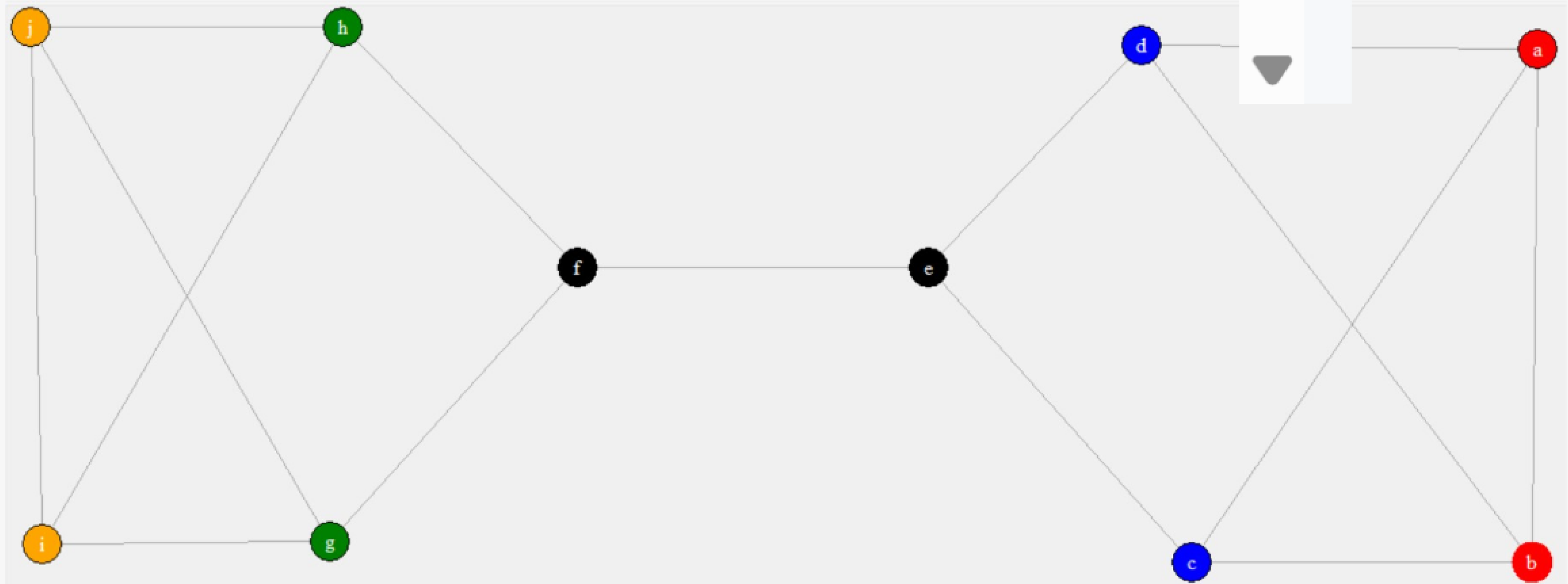
- The most straightforward weight matrix is based on the adjacency matrix: you are influenced by those you are tied to directly.

```r
net_mat
```

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

```r
round(net_mat / rowSums(net_mat), digits = 2)
```

```
     a    c    d    b    e    f    g    h    i    j
a 0.00 0.33 0.33 0.33 0.00 0.00 0.00 0.00 0.00 0.00
c 0.33 0.00 0.00 0.33 0.33 0.00 0.00 0.00 0.00 0.00
d 0.33 0.00 0.00 0.33 0.33 0.00 0.00 0.00 0.00 0.00
b 0.33 0.33 0.33 0.00 0.00 0.00 0.00 0.00 0.00 0.00
e 0.00 0.33 0.33 0.00 0.00 0.33 0.00 0.00 0.00 0.00
f 0.00 0.00 0.00 0.00 0.33 0.00 0.33 0.33 0.00 0.00
g 0.00 0.00 0.00 0.00 0.00 0.33 0.00 0.00 0.33 0.33
h 0.00 0.00 0.00 0.00 0.00 0.33 0.00 0.00 0.33 0.33
i 0.00 0.00 0.00 0.00 0.00 0.00 0.33 0.33 0.00 0.33
j 0.00 0.00 0.00 0.00 0.00 0.00 0.33 0.33 0.33 0.00
```



# *Operationalizing* COMMUNICATION

There are many alternatives, such as:

- are you members of the same component/subgroup?
- number of two-paths between two people

Here, you assume that being in the same work component, advice-sharing component, collaboration component, course team, sitting at the same table, et cetera, makes it more likely that people communicate directly.

# *Operationalizing* COMPARISON

How would you operationalize whom a network actor would compare himself/herself to?

# *structural equivalence*

One mathematical way to measure how similar the other person is to you-- in a "social network sense", is through *structural equivalence*.

Two nodes are structurally equivalent if they have **precisely** the same relations across all other nodes in the network.

Compare a and b (in red) with e and f (in black). Which pairs are more strongly structurally equivalent?

# Structural equivalence

```
distances <- snafun::d_structural_equivalence(net_sna)
distances
```

```
       a      c      d      b      e      f      g      h      i      j
a  1.000  0.333  0.333  1.000  0.467 -0.600 -0.600 -0.600 -0.600 -0.600
c  0.333  1.000  1.000  0.333 -0.333 -0.067 -0.600 -0.600 -0.600 -0.600
d  0.333  1.000  1.000  0.333 -0.333 -0.067 -0.600 -0.600 -0.600 -0.600
b  1.000  0.333  0.333  1.000  0.467 -0.600 -0.600 -0.600 -0.600 -0.600
e  0.467 -0.333 -0.333  0.467  1.000 -0.333 -0.067 -0.067 -0.600 -0.600
f -0.600 -0.067 -0.067 -0.600 -0.333  1.000 -0.333 -0.333  0.467  0.467
g -0.600 -0.600 -0.600 -0.600 -0.067 -0.333  1.000  1.000  0.333  0.333
h -0.600 -0.600 -0.600 -0.600 -0.067 -0.333  1.000  1.000  0.333  0.333
i -0.600 -0.600 -0.600 -0.600 -0.600  0.467  0.333  0.333  1.000  1.000
j -0.600 -0.600 -0.600 -0.600 -0.600  0.467  0.333  0.333  1.000  1.000
```

This measure runs between (-1, 1), with 1 showing two vertices having perfectly identical edges and -1 showing exactly opposite positions.

# Running the autocorrelation model

$$y = \rho W y + X\beta + \epsilon$$

```
snafun::stat_nam(y ~ x, W = NULL, model = "lag")
```

Here, $y$ is the dependent variable (a vector), $X$ is the matrix with exogenous variables, and $W$ is the weight matrix.

The model implemented in `snafun::stat_nam`

$$
\begin{aligned}
y &= \rho_1 W y + X\beta + \epsilon \\
\epsilon &= \rho_2 W_2 \epsilon + \nu
\end{aligned}
$$

You run the models as follows:

| Code | Model |
| --- | --- |
| `snafun::stat_nam(y ~ x, W = NULL, model = 'lag')` | W = W, W2 = 0 |
| `snafun::stat_nam(y ~ x, W = NULL, model = 'error')` | W = 0, W2 = W2 |
| `snafun::stat_nam(y ~ x, W = NULL, model = 'combined')` | W = W, W2 = W2 |

# Running the autocorrelation model

SNA4DS

Data R code Output

- 100 people read a piece of information and share their thoughts with others through a chatboard about how trustworthy they consider the information
- after the discussion, they each score the trustworthiness of the information on a continuous scale between 0 and 10
- Explanatory variables:
  - political preference
  - proportion of waking time spent on social media
  - expertise on the topic
  - agreeableness (personality trait)

# Running the autocorrelation model

Data | R code | Output

```
# communication: direct ties
# trust_net is the adjacency matrix of the communication
diag(trust_net) <- 0
w_adj <- trust_net / rowSums(trust_net)

nam <- snafun::stat_nam(trust_y ~ ., W = w_adj, data = trust)
summary(nam)
```

# Running the autocorrelation model

Data R code **Output**

```
Call:spatialreg::lagsarlm(formula = formula, data = data, listw = W,
    na.action = na.action, Durbin = Durbin, quiet = quiet, zero.policy = zero.policy)

Residuals:
       Min         1Q     Median         3Q        Max
-2.0390625 -0.6160851  0.0022923  0.5660169  2.5077752

Type: lag
Coefficients: (asymptotic standard errors)
               Estimate Std. Error  z value  Pr(>|z|)
(Intercept)    0.339855   0.342785   0.9915 0.3214655
media_use      1.087801   0.316341   3.4387 0.0005845
expertise     -0.504861   0.023336 -21.6345 < 2.2e-16
politics       0.206052   0.037757   5.4573 4.834e-08
agreeableness  0.512199   0.315279   1.6246 0.1042499

Rho: 0.56058, LR test value: 6.2892, p-value: 0.012148
Asymptotic standard error: 0.1863
```