*Workshop Network Modeling for Psychological (and Attitudinal) Data*

# Exercises Day 1-2: Estimating Networks in R

Oldenburg, 11/01/2018

In R, run the following code:

```
install.packages("psych")
```

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

The data frame `bfiData` contains the questions of the `bfi` (Big Five Inventory) data contained in the *psych* package. More information on this dataset can be obtained by running:

```
?bfi
```

The questions are designed to measure five central personality traits: Agreeableness, Conscientiousness, Extraversion, Neuroticism, and Opennness. The following table gives the item descriptions:

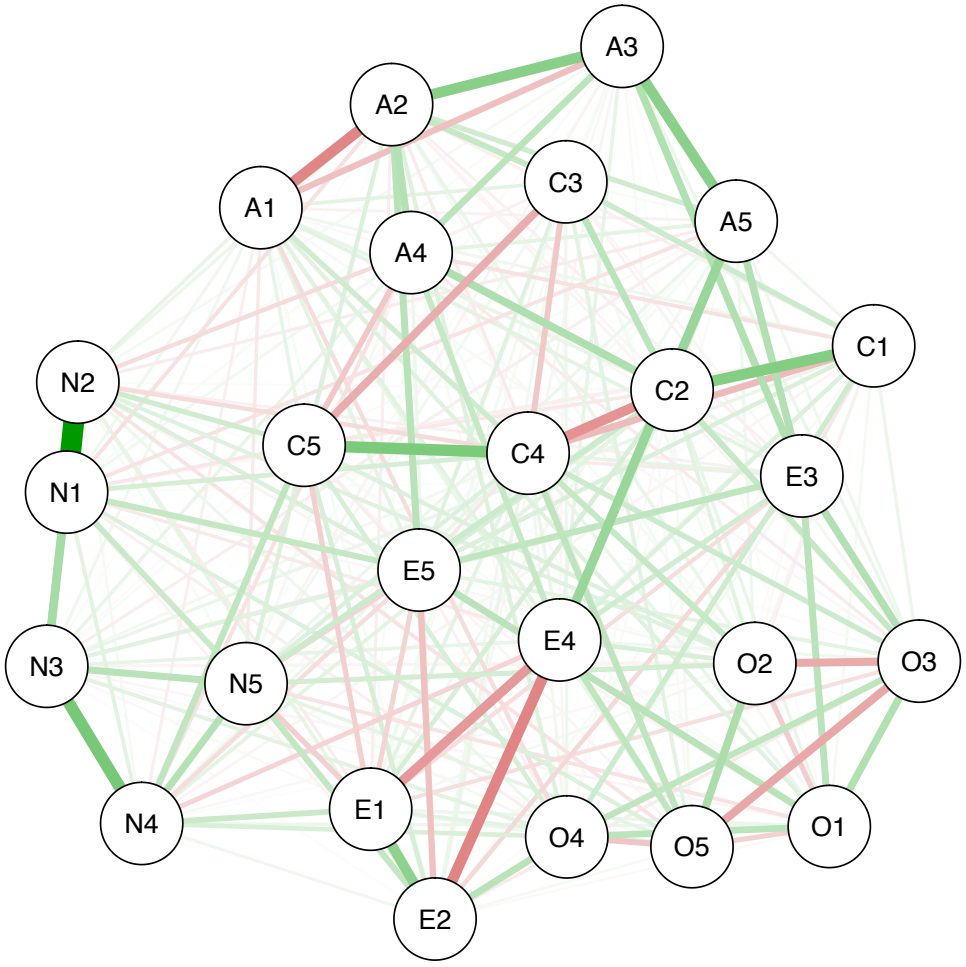
| Item label | Item description | Trait |
|---|---|---|
| A1 | Am indifferent to the feelings of others | Agreeableness |
| A2 | Inquire about others' well-being | Agreeableness |
| A3 | Know how to comfort others | Agreeableness |
| A4 | Love children | Agreeableness |
| A5 | Make people feel at ease | Agreeableness |
| C1 | Am exacting in my work | Conscientiousness |
| C2 | Continue until everything is perfect | Conscientiousness |
| C3 | Do things according to a plan | Conscientiousness |
| C4 | Do things in a half-way manner | Conscientiousness |
| C5 | Waste my time | Conscientiousness |
| E1 | Don't talk a lot | Extraversion |
| E2 | Find it difficult to approach others | Extraversion |
| E3 | Know how to captivate people | Extraversion |
| E4 | Make friends easily | Extraversion |
| E5 | Take charge | Extraversion |
| N1 | Get angry easily | Neuroticism |
| N2 | Get irritated easily | Neuroticism |
| N3 | Have frequent mood swings | Neuroticism |
| N4 | Often feel blue | Neuroticism |
| N5 | Panic easily | Neuroticism |
| O1 | Am full of ideas | Opennness |
| O2 | Avoid difficult reading material | Opennness |
| O3 | Carry the conversation to a higher level | Opennness |
| O4 | Spend time reflecting on things | Opennness |
| O5 | Will not probe deeply into a subject | Opennness |

We can compute a polychoric correlation matrix based on this data as follows:

```
library("qgraph")
corMat <- cor_auto(bfiData)
```

Next we can use *qgraph* to compute a partial correlation network:

```
qgraph(corMat, graph = "pcor", layout = "spring", cut = 0)
```




We can use the *bootnet* function `estimateNetwork` to automate this process:

```
library("bootnet")
Result_pcor <- estimateNetwork(bfiData, default = "pcor")
plot(Result_pcor, layout = "spring", cut = 0)
```

**Exercise 1** Obtain the weights matrices from qgraph and bootnet by applying the `getWmat` function to output of both. Confirm that the results are identical (tip: the operator == tests if values in R are equal).
Solution:

```
g1 <- qgraph(corMat, graph = "pcor", layout = "spring", cut = 0)
w1 <- getWmat(g1)
g2 <- estimateNetwork(bfiData, default = "pcor")
w2 <- getWmat(g2)

all(w1==w2)
# Returns TRUE
```

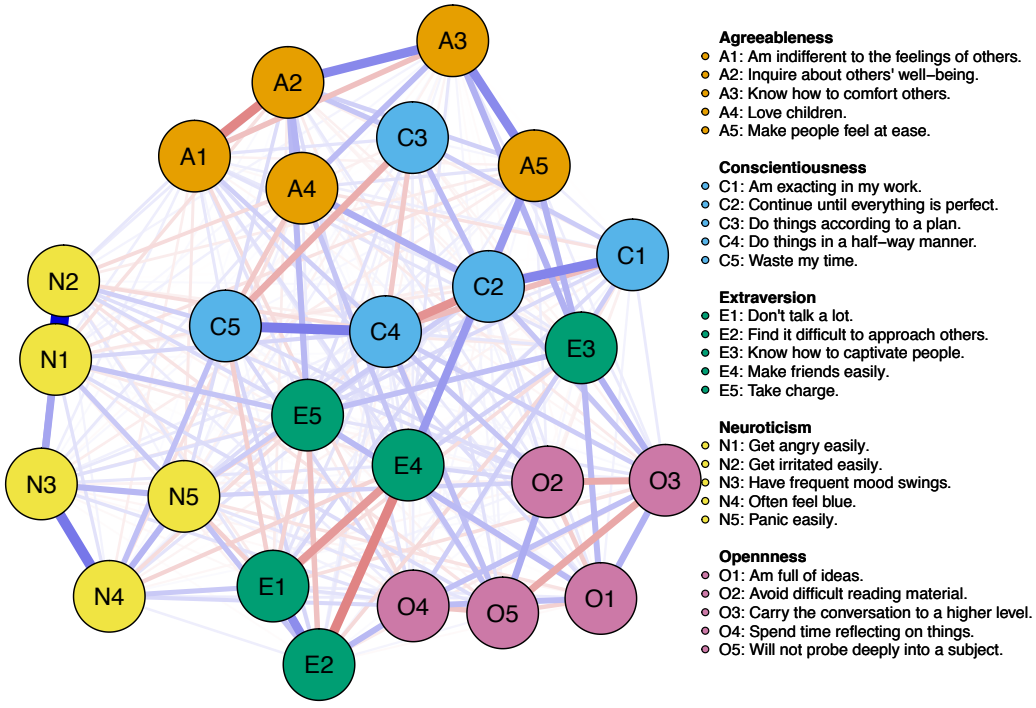
■

We can load for each node the item description and factor the item is aimed to measure as follows:

```
Names <- scan("http://sachaepskamp.com/files/BFIitems.txt",
              what = "character", sep = "\n")
Traits <- rep(c(
  'Agreeableness',
  'Conscientiousness',
  'Extraversion',
  'Neuroticism',
  'Opennness'
),each=5)
```

These can be used to plot a legend next to the graph. In combination, we can make the graph friendly to colorblind viewers using the `theme` option:

```
plot(Result_pcor,
     layout = "spring",
     cut = 0,
     theme = "colorblind",
     groups = Traits,
     nodeNames = Names,
     legend.cex = 0.4)
```




**Exercise 2** What do the arguments `groups` and `nodeNames` do?
Solution:
`groups` colors nodes according to the traits and `nodeNames` is used in the legend next to the network. ■

**Exercise 3** In `estimateNetwork`, use the `threshold` argument to remove all edges that are not significant after applying a *bonferroni* correction. Plot the resulting network.
Solution:

```
Res_bonf <- estimateNetwork(bfiData, default = "pcor",
                            threshold = "bonferroni")

plot(Res_bonf,
     layout = "spring",
     cut = 0,
     theme = "colorblind",
     groups = Traits,
     nodeNames = Names,
     legend.cex = 0.3)
```

■

In `estimateNetwork`, the `fun` argument can be specified a custum function estimating the network from data. To aid the user, several default functions have been built in. For example, `default = "pcor"` specified a function that estimates a partial correlation networks (in the help file this function is called `bootnet_pcor`).

**Exercise 4** Use the `default` argument in `estimateNetwork` to estimate a partial correlation network using glasso and EBIC model selection.
Solution:

```
Res_glasso <- estimateNetwork(bfiData, default = "EBICglasso")
```

■

**Exercise 5** Set the hypertuningparameter $\gamma$ to 0. Did the network change?
Solution:

```
Res_glasso2 <- estimateNetwork(bfiData, default = "EBICglasso", tuning = 0)
L <- averageLayout(Res_glasso,Res_glasso2)
layout(t(1:2))
plot(Res_glasso, layout = L, cut = 0)
plot(Res_glasso2, layout = L, cut = 0)
```

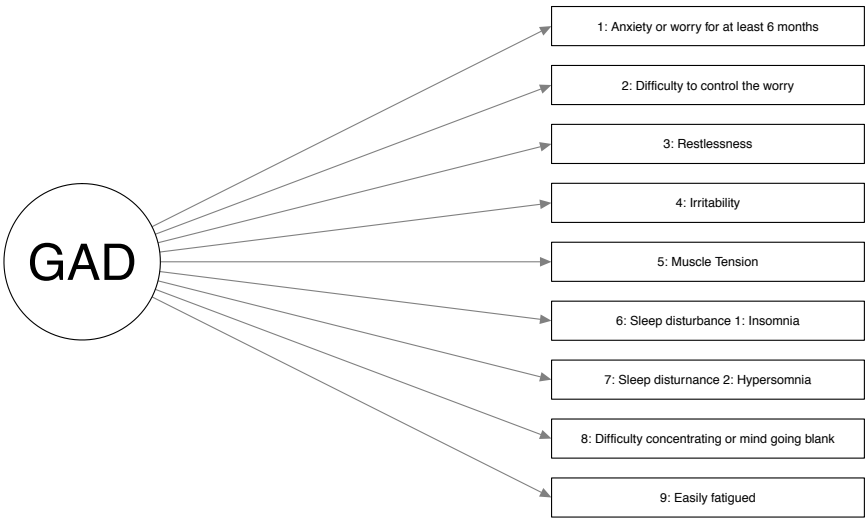
A different model was selected, but the differences are very small ■

**Exercise 6** Set the default to estimate a regularized Ising model using $\gamma = 0.5$. Compare your results with the EBIC glasso network using $\gamma = 0.5$.
Solution:

```
Res_Ising <- estimateNetwork(bfiData, default = "IsingFit")
L <- averageLayout(Res_glasso, Res_Ising)
layout(t(1:2))
plot(Res_glasso, layout = L, cut = 0)
plot(Res_Ising, layout = L, cut = 0)
```

The estimated Ising model is mostly the same but sparser. ■

Suppose the following model is the *true* data-generating model of nine symptoms:




The circle indicates that generalized anxiety disorder (GAD) is latent and thus not observed.

**Exercise 7** Theoretically, what should the Gaussian graphical model (GGM; a network of partial correlation coefficients) of the nine observed symptoms look like? Plot or draw the expected network (use only the numbers as node labels). Tip: Note that GAD is not observed and thus (a) not in the GGM and (b) not a node we can condition on.
Solution:
The GGM will be fully connected. E.g., `qgraph(matrix(1,9,9))` ■

References
Beveridge, A., & Shan, J. (2016). Network of thrones. Math Horizons, 23(4), 18–22.
Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social networks, 32(3), 245–251.
Watts, D., & Strogatz, S. (1998). Collective dynamics of'small-world'networks. Nature, 393(6684), 440-442