Lab Exercise: Psychometric Network Analysis - COVID Influences on Mental Health in the UK

Todd K Hartman

Last updated 2024-05-16

Getting Started

In this lab exercise, we're going to use an approach closely related to social network analysis to investigate relationships among survey items. This approach is called **psychometric network analysis**. The key difference between social network analysis and psychometric network analysis is that with the former, nodes are defined as individuals or organizations (e.g., people, countries, etc.), whereas in the latter, nodes are survey questions. Both approaches investigate partial correlations among nodes (i.e., controlling for all other links among them) to map out connections among individuals/measures in a given network.

Background

One common method for exploring psychopathology is by using *latent variable modelling*, which assumes that psychiatric syndromes (i.e., groups of symptoms) occur because the component symptoms share unobserved, or *latent*, common causes (see Borsboom & Cramer, 2013). In other words, the survey items indicate responses to an unobserved causal factor.

In contrast, scholars who use network analytic approaches hypothesize that mental disorders arise from complex causal relationships between symptoms (see McNally, 2016). For instance, insomnia, fatigue, and irritability do not arise from a common underlying latent depression disease process; instead, they *covary* (i.e., increase or decrease in tandem) because they affect each other, e.g., insomnia may lead to to fatigue, which in turn, leads to irritability. Psychiatric disorders are therefore conceived as *emergent phenomena*, in which the covariation between symptoms evolves over time.

New psychometric analytical tools which have been inspired by network theory allow mental disorders to be represented graphically, as networks with 'nodes' as symptoms and 'edges' as inferred statistical associations between them. 'Centrality indices' can be computed to infer the importance of each node within the network, in terms of strength of connections to other nodes. The expectation is that this approach will lead to the identification of symptoms which are particularly important in the genesis of psychological disturbance or which 'bridge the gap' between different syndromes, thereby leading to comorbidity between disorders.

Data and measures

The data were collected from two waves of a longitudinal survey project, the COVID-19 Psychological Research Consortium Study (C19PRC; see McBride et al., 2020). Participants in the UK strand, who were recruited by the survey company Qualtrics, were aged 18 or older, a resident in the UK, and able to read and write in English. Wave 1 (W1; 23–28 March 2020) recruited 2,025 participants during the first week of UK lockdown. All W1 respondents were re-contacted and invited to participate in Wave 2 (W2), which was conducted during 22 April–1 May 2020. The W2 retention rate was 69.4% (N = 1,406).

Psychopathology outcomes were measured using validated scales for **depression** [Patient Health Questionnaire-9 (PHQ-9)], **generalized anxiety** [Generalized Anxiety Disorder Scale (GAD-7)], and **traumatic stress** [International Trauma Questionnaire (ITQ)].

The PHQ-9 and GAD-7 are asked using the following question and response options:
"Over the last two weeks, how often have you been bothered by the following problems?"

0 = Not at all
1 = Several days
2 = More than half the days 3 = Nearly every day

PHQ-9
1. Little interest or pleasure in doing things
2. Feeling down, depressed, or hopeless
3. Trouble falling or staying asleep, or sleeping too much
4. Feeling tired or having little energy
5. Poor appetite or overeating
6. Feeling bad about yourself - or that you are a failure or have let yourself or your family down
7. Trouble concentrating on things, such as reading the newspaper or watching television
8. Moving or speaking so slowly that other people have noticed? Or the opposite - being so fidgety or restless that you have been moving around more than usual
9. Thoughts that you would be better dead or of hurting yourself in some way

GAD-7
1. Feeling nervous, anxious or on edge
2. Not being able to stop or control worrying
3. Worrying too much about different things
4. Trouble relaxing
5. Being so restless that it is hard to sit still
6. Becoming easily annoyed or irritable
7. Feeling afraid as if something awful might happen

The ITQ-6 is measured using the following question and response options:
"In this section, you will be asked questions about different ways that people sometimes react following a traumatic or stressful life event. Please answer the following questions in relation to your experience of the COVID-19 pandemic. Please read each item carefully, then select one of the answers to indicate how much you have been bothered by that problem in the past month."

0 = Not at all
1 = A little bit
2 = Moderately
3 = Quite a bit
4 = Extremely

ITQ-6
1. Having upsetting dreams that replay part of the experience or are clearly related to the experience?
PTSD2 2. Having powerful images or memories that sometimes come into your mind in which you feel the experience is happening again in the here and now?
PTSD3 3. Avoiding internal reminders of the experience (for example, thoughts, feelings, or physical sensations)?
PTSD4 4. Avoiding external reminders of the experience (for example, people, places, conversations, objects, activities, or situations)?
PTSD5 5. Being "super-alert", watchful, or on guard?
PTSD6 6. Feeling jumpy or easily startled?

More details about this study are available here: https://doi.org/10.0.3.249/S0033291721000635

Housekeeping

Let's begin by installing and/or loading the necessary packages for this lab. The primary package to conduct psychometric network analysis is called `mgm`, which estimates a Mixed Graphical Model. It's called 'mixed' because it handles items measured on mixed, or different types of response scales (e.g., continuous, ordinal, binary, and count outcomes). The package creator, Jonas Haslbeck, has some great resources available on his personal website: https://jonashaslbeck.com/Estimation-of-mixed-graphical-models/.

We'll also install a few other packages described below using the `pacman` package manager.

```
pacman::p_load(haven,       # To import different data types
               mgm,         # Analyze Mixed Graphical Models
               osfr,        # Download OSF data
               qgraph,      # Create stunning network figures
               psych,       # Calculate summary statistics
               sjPlot,      # Many useful data manipulation and data viz functions
               tidyverse    # Manipulate data the 'tidy' way
               )
```

Next, let's find and import the C19PRC longitudinal data. The data can be accessed and downloaded from the Open Science Framework (OSF). The OSF is a free and open source web application that provides a space for researchers to collaboratively store, manage, and share their research materials (e.g. data, code, protocols, etc.). The data for Waves 1 and 2 are stored in a zipped file located under the 'Files' section of the study page (just scroll down and you should see them): https://osf.io/v2zur/. Or, alternatively Waves 1 and 2 data are located at this page (if you check the address bar in your browser, you'll see that the link will tell you the unique IDs for any file on the OSF website): https://osf.io/9emvp.

Fortunately, there is a package called `osfr`, which we installed and loaded above; this package allows us to directly download data from the OSF. All you need is the unique file ID, which is `9emvp` for this data, and we can use the following code to download, unzip, and import the data:

```
## Download data for this exercise from the OSF
## You could do this manually if you run into errors!
## https://osf.io/9emvp
osf_data <-
  osf_retrieve_file('9emvp') %>%  # Get this file id
  osf_download(conflicts = 'overwrite')  # Download, overwriting files with the same name

## Unzip the file using the stored file name
unzip(osf_data$local_path)

## Import the C19PRC Study W1 & W2 data
## using the 'haven' package
df <- read_sav('C19PRC_UKW1W2_archive_final.sav')
# glimpse(df)  # Output is long; uncomment to check the data
```

Now we're ready to make sure the measures are correctly coded for the psychometric network analysis. Recall that these are the measures we're going to use in this analysis:

depression: Patient Health Questionnaire-9 (PHQ-9)
generalized anxiety: Generalized Anxiety Disorder Scale (GAD-7)
traumatic stress: International Trauma Questionnaire (ITQ)

How might we locate these specific survey questions? One place to start might be to use the `sjPlot` package to print a codebook of variable names and value labels (if they exist). This works particularly well when you import data from other statistical packages like SPSS (.sav files), Stata (.dta files), etc. Do you know how to print a codebook?

```
## Create the C19PRC codebook based on the imported data
# view_df(df)  # Output is long; uncomment to view the data
```

Once you've created the codebook, how might you find the variables names you need? Remember that you can search within a file if you open the codebook in your default browser (just click the 'Show in New Window' icon under the 'Viewer' tab in the bottom right-hand corner of your RStudio session). Try searching for 'PHQ', 'GAD' or 'ITQ' (or similar variants). If this fails, then read the study documentation included in the zipped file.

Assuming you've found the variables names for the three mental health syndromes, you're ready to subset the data, keeping just those measures that we need for the analysis (it's easier this way than selecting variables for the analysis, figures, etc.). We can subset data using the `select()` function from `tidyverse`. Then check the summary statistics of those items to make sure they're coded correctly (i.e., there are no strange values). What do you see? Do any of the items need to reverse coded (e.g., sometimes items in a scale will be reverse-worded to avoid agreement bias)? If not, then we're ready to move on!

```r
## Subset the data, keeping depression, anxiety, trauma items, note the colon
## which works because the items are in order (i.e., next to one another in the dataframe)
df.sub.w1 <- df %>%
  select(W1_Dep_1:W1_Dep_9,
         W1_GAD_1:W1_GAD_7,
         W1_PTSD1:W1_PTSD6)

glimpse(df.sub.w1)  # Check the subsetted data
```

```
## Rows: 2,025
## Columns: 22
## $ W1_Dep_1 <dbl+lbl> 0, 0, 0, 1, 2, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, ~
## $ W1_Dep_2 <dbl+lbl> 0, 0, 0, 2, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, ~
## $ W1_Dep_3 <dbl+lbl> 0, 0, 0, 2, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 3, 0, 0, 0, ~
## $ W1_Dep_4 <dbl+lbl> 0, 1, 0, 1, 2, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 3, 0, 0, 0, ~
## $ W1_Dep_5 <dbl+lbl> 1, 0, 0, 0, 1, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, ~
## $ W1_Dep_6 <dbl+lbl> 0, 0, 0, 1, 2, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, ~
## $ W1_Dep_7 <dbl+lbl> 0, 0, 0, 2, 3, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 3, 0, 0, 0, ~
## $ W1_Dep_8 <dbl+lbl> 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, ~
## $ W1_Dep_9 <dbl+lbl> 0, 0, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, ~
## $ W1_GAD_1 <dbl+lbl> 1, 1, 0, 3, 2, 2, 1, 0, 2, 2, 1, 1, 0, 1, 0, 3, 0, 1, 0, ~
## $ W1_GAD_2 <dbl+lbl> 1, 0, 0, 2, 2, 2, 1, 0, 1, 1, 0, 0, 0, 1, 0, 3, 0, 1, 0, ~
## $ W1_GAD_3 <dbl+lbl> 1, 0, 0, 1, 1, 2, 1, 0, 1, 1, 1, 0, 0, 1, 0, 3, 0, 1, 0, ~
## $ W1_GAD_4 <dbl+lbl> 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 3, 0, 2, 0, ~
## $ W1_GAD_5 <dbl+lbl> 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, ~
## $ W1_GAD_6 <dbl+lbl> 1, 0, 0, 0, 1, 2, 0, 0, 1, 1, 0, 0, 0, 1, 0, 3, 0, 2, 0, ~
## $ W1_GAD_7 <dbl+lbl> 1, 0, 0, 3, 2, 2, 0, 0, 1, 1, 0, 0, 0, 2, 0, 3, 0, 1, 0, ~
## $ W1_PTSD1 <dbl+lbl> 2, 1, 0, 0, 3, 3, 2, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 3, ~
## $ W1_PTSD2 <dbl+lbl> 2, 0, 0, 1, 2, 3, 2, 2, 0, 0, 0, 0, 0, 1, 0, 4, 1, 3, 3, ~
## $ W1_PTSD3 <dbl+lbl> 0, 1, 0, 2, 3, 4, 2, 2, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1, 3, ~
## $ W1_PTSD4 <dbl+lbl> 0, 1, 0, 2, 1, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 3, ~
## $ W1_PTSD5 <dbl+lbl> 2, 1, 0, 1, 2, 3, 2, 1, 0, 1, 0, 0, 0, 3, 0, 3, 0, 2, 3, ~
## $ W1_PTSD6 <dbl+lbl> 2, 2, 0, 1, 3, 4, 2, 2, 0, 2, 0, 0, 0, 3, 0, 4, 0, 2, 3, ~
```

```r
describe(df.sub.w1)  # Use the 'psych' package to describe the data
```

```
##           vars    n mean   sd median trimmed  mad min max range skew kurtosis
## W1_Dep_1    1 2025 0.70 0.92      0    0.56 0.00   0   3     3 1.08     0.06
## W1_Dep_2    2 2025 0.73 0.92      0    0.58 0.00   0   3     3 1.08     0.18
## W1_Dep_3    3 2025 0.85 1.00      1    0.69 1.48   0   3     3 0.89    -0.40
## W1_Dep_4    4 2025 0.83 0.98      1    0.67 1.48   0   3     3 0.95    -0.22
## W1_Dep_5    5 2025 0.57 0.90      0    0.39 0.00   0   3     3 1.43     0.91
## W1_Dep_6    6 2025 0.53 0.89      0    0.34 0.00   0   3     3 1.54     1.21
## W1_Dep_7    7 2025 0.57 0.88      0    0.39 0.00   0   3     3 1.43     0.99
## W1_Dep_8    8 2025 0.29 0.69      0    0.09 0.00   0   3     3 2.48     5.33
## W1_Dep_9    9 2025 0.30 0.71      0    0.11 0.00   0   3     3 2.45     5.24
## W1_GAD_1   10 2025 0.88 0.97      1    0.73 1.48   0   3     3 0.87    -0.28
## W1_GAD_2   11 2025 0.73 0.96      0    0.57 0.00   0   3     3 1.10     0.08
## W1_GAD_3   12 2025 0.79 0.97      0    0.63 0.00   0   3     3 1.00    -0.11
## W1_GAD_4   13 2025 0.76 0.95      0    0.61 0.00   0   3     3 1.04     0.01
## W1_GAD_5   14 2025 0.50 0.84      0    0.32 0.00   0   3     3 1.62     1.66
## W1_GAD_6   15 2025 0.73 0.94      0    0.57 0.00   0   3     3 1.08     0.09
## W1_GAD_7   16 2025 0.76 0.96      0    0.60 0.00   0   3     3 1.04    -0.01
## W1_PTSD1   17 2025 0.67 1.09      0    0.44 0.00   0   4     4 1.47     1.05
## W1_PTSD2   18 2025 0.64 1.07      0    0.40 0.00   0   4     4 1.57     1.37
## W1_PTSD3   19 2025 0.67 1.08      0    0.44 0.00   0   4     4 1.49     1.18
## W1_PTSD4   20 2025 0.71 1.10      0    0.47 0.00   0   4     4 1.40     0.80
## W1_PTSD5   21 2025 1.10 1.24      1    0.93 1.48   0   4     4 0.82    -0.51
## W1_PTSD6   22 2025 0.80 1.18      0    0.58 0.00   0   4     4 1.28     0.42
##            se
## W1_Dep_1 0.02
## W1_Dep_2 0.02
## W1_Dep_3 0.02
## W1_Dep_4 0.02
## W1_Dep_5 0.02
## W1_Dep_6 0.02
## W1_Dep_7 0.02
## W1_Dep_8 0.02
## W1_Dep_9 0.02
## W1_GAD_1 0.02
## W1_GAD_2 0.02
## W1_GAD_3 0.02
## W1_GAD_4 0.02
## W1_GAD_5 0.02
## W1_GAD_6 0.02
## W1_GAD_7 0.02
## W1_PTSD1 0.02
## W1_PTSD2 0.02
## W1_PTSD3 0.02
## W1_PTSD4 0.02
## W1_PTSD5 0.03
## W1_PTSD6 0.03
```

To use the `mgm()` function, we have to specify how the items are measured using the `type()` and `level()` arguments. In our case, the data are ordered, so we'll specify 'g' for Gaussian as the `type` and '1' for continuous as the `level`:

```r
## Variable Specification: 'g' for 'Gaussian' (ordered/continuous) response scales
## 'p' for Poisson (count) responses
## 'c' for Nominal (unordered categorical) responses
type <- c('g', 'g', 'g', 'g', 'g', 'g', 'g', 'g', 'g',  # PHQ-9 Items
           'g', 'g', 'g', 'g', 'g', 'g', 'g',  # GAD-7 Items
           'g', 'g', 'g', 'g', 'g', 'g')  # ITQ-6 Items

## Or we could have just created a vector with all 22 items as 'g'
## using the 'rep()' function, which repeats an element X number of times
type <- c(rep('g', 22))

## Number of categories; 1 for continuous variables
level <- c ('1', '1', '1', '1', '1', '1', '1', '1', '1',  # PHQ-9 Items
            '1', '1', '1', '1', '1', '1', '1',  # GAD-7 Items
            '1', '1', '1', '1', '1', '1' )  # ITQ-6 Items
```

Now let’s estimate use the `mgm` package to estimate the Mixed Graphical Model:

```
## Convert data.frame to matrix for analysis
df.sub.w1 <- as.matrix(df.sub.w1)

## Mixed Graphical Model Estimation
mgm_net <- mgm(data = df.sub.w1,    # Data for psychometric network analysis
               type = type,         # Type of responses
               level = level,       # Number of categories (1 for continuous)
               lambdaSel = "EBIC")  # Use Extended Bayesian Information Criterion for tuning parameter
```

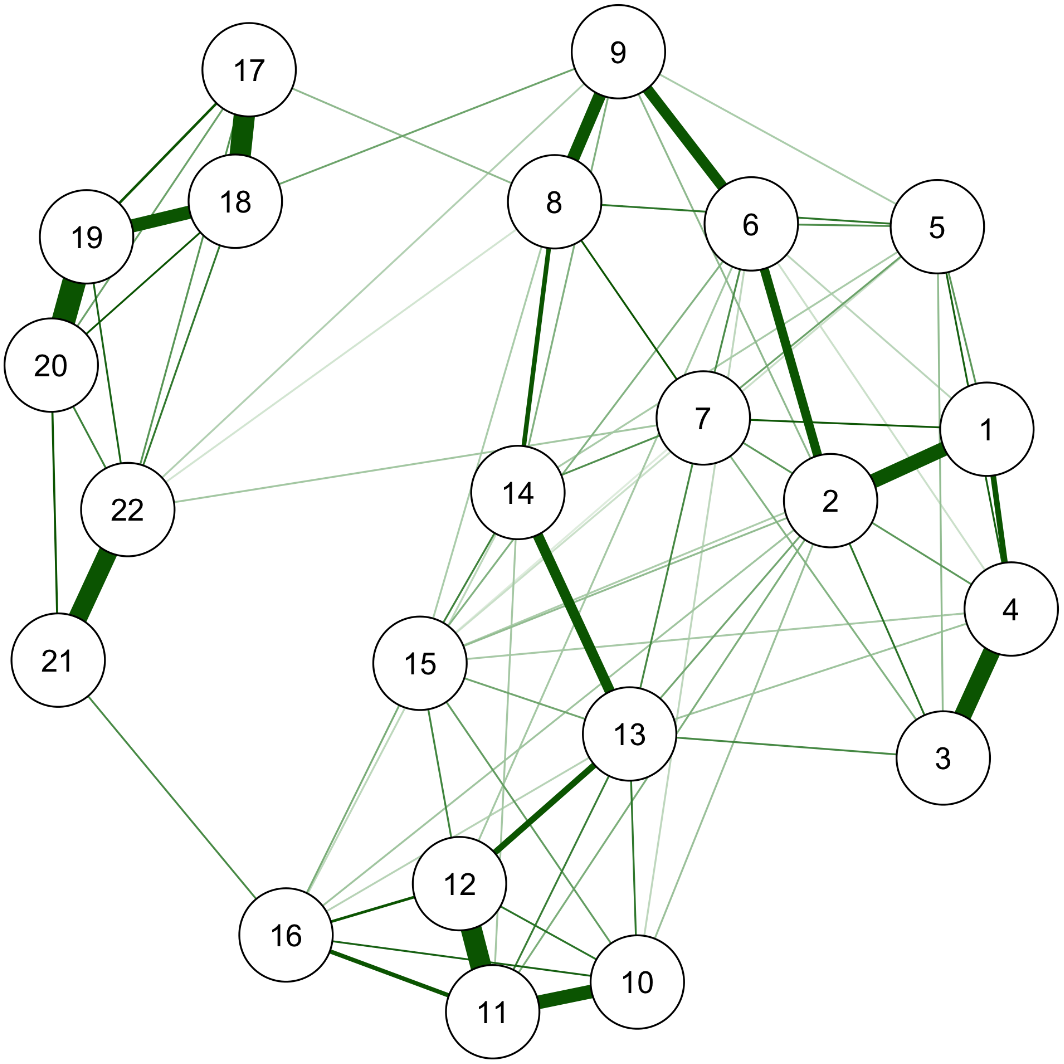
```
##   |                                                                               |
## Note that the sign of parameter estimates is stored separately; see ?mgm
```

```
# mgm_net$pairwise$wadj  # uncomment to print weighted adjacency matrix for network figure
```

The weighted adjacency matrix is hard to read in a tabular format. Let’s visualize the psychometric data graphically using the `qgraph()` function and save the figure as a .png file. The main input with network graphs is an (weighted) adjacency matrix, and we’ll select the layout as ‘spring’, which organizes the nodes according to their connections with other nodes in the network.

```
## Create the network figure using 'qgraph'
fig1 <- qgraph(mgm_net$pairwise$wadj,    # Input is a weighted adjacency matrix
               edge.color = mgm_net$pairwise$edgecolor, # incorporate sign as color
               layout = "spring",        # Fruchterman-Reingold algorithm
               filetype = 'png',         # Save as .png ( can be modified if desired)
               filename = 'figure1'      # Use this file name
               )
```

```
## Output stored in /Users/user/Library/CloudStorage/Dropbox/_TEACHING/Appstate_Teaching/__POLITICAL_AN
```



The default figure is OK, but we should be able to make it better. For instance, which items correspond to which nodes in the figure? It's difficult to know for sure without these labels. So let's create our own node labels to improve the base figure.

```
## Node names for pretty figures
node.names <- colnames(df.sub.w1)  # Grab variables names
node.names  # Check names
```

```
##  [1] "W1_Dep_1" "W1_Dep_2" "W1_Dep_3" "W1_Dep_4" "W1_Dep_5" "W1_Dep_6"
##  [7] "W1_Dep_7" "W1_Dep_8" "W1_Dep_9" "W1_GAD_1" "W1_GAD_2" "W1_GAD_3"
## [13] "W1_GAD_4" "W1_GAD_5" "W1_GAD_6" "W1_GAD_7" "W1_PTSD1" "W1_PTSD2"
## [19] "W1_PTSD3" "W1_PTSD4" "W1_PTSD5" "W1_PTSD6"
```

```
## Make names all uppercase
node.names <- toupper(node.names)
```

```
## Remove 'W1_' using global (ALL) substitution
node.names <- gsub('W1_', '', node.names)  # Any time 'W1_' is found, replace with nothing

## Remove underscore
node.names <- gsub('_', '', node.names)

## Check the final list of names
node.names
```

```
##  [1] "DEP1"  "DEP2"  "DEP3"  "DEP4"  "DEP5"  "DEP6"  "DEP7"  "DEP8"  "DEP9"
## [10] "GAD1"  "GAD2"  "GAD3"  "GAD4"  "GAD5"  "GAD6"  "GAD7"  "PTSD1" "PTSD2"
## [19] "PTSD3" "PTSD4" "PTSD5" "PTSD6"
```

Now let's create another figure and see if the node names help. What do you see now? Which items are most strongly connected in the network (strength is denoted by thicker lines)?

```
## Create the network figure using 'qgraph'
fig2 <- qgraph(mgm_net$pairwise$wadj,
               layout = "spring",
               labels = node.names,      # Add labels
               filetype = 'png',
               filename = 'figure2'
               )
```

```
## Output stored in /Users/user/Library/CloudStorage/Dropbox/_TEACHING/Appstate_Teaching/__POLITICAL_ANA
```

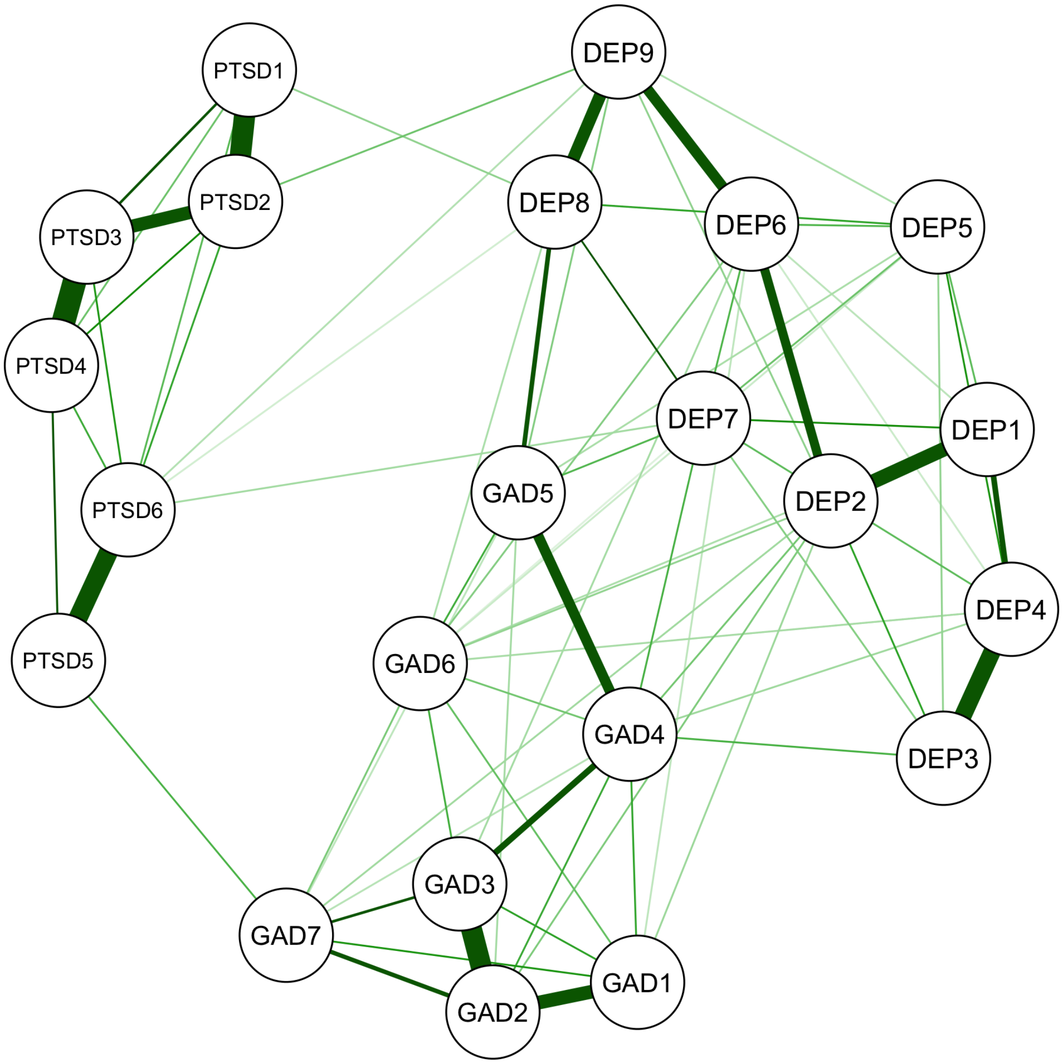


Figure 2 was certainly an improvement, but we could take this one step further by grouping by the disorder the nodes are meant to measure (e.g., all PHQ-9 items together). If we add colour to these groups, then it'll be really helpful to better understand what's going on in the data.

In addition, we can specify colours for positive and negative edges, blue for positive and red for negative ties, respectively.

Finally, we could dsplay the *predictability* of each node in the network, that is, how well can a given node in the network be predicted by all remaining nodes in the network? What does this tell us? First, it gives us an idea of the relevance of edges in the network. If a node is connected to many other nodes but these only explain a small portion of its variance, then it may not be very interesting to examine the edges connected to that node. Moreover, predictability tells us where to target an intervention to change a certain set of nodes. And, predictability will tell us the extent to which different parts of the network are self-determined or determined by other factors that are not included in the network. In sum, predictability quantifies how much variance in a node is explained by its neighbours, which can be used to assess the extent to which the network structure predicts node states. Neat, right?

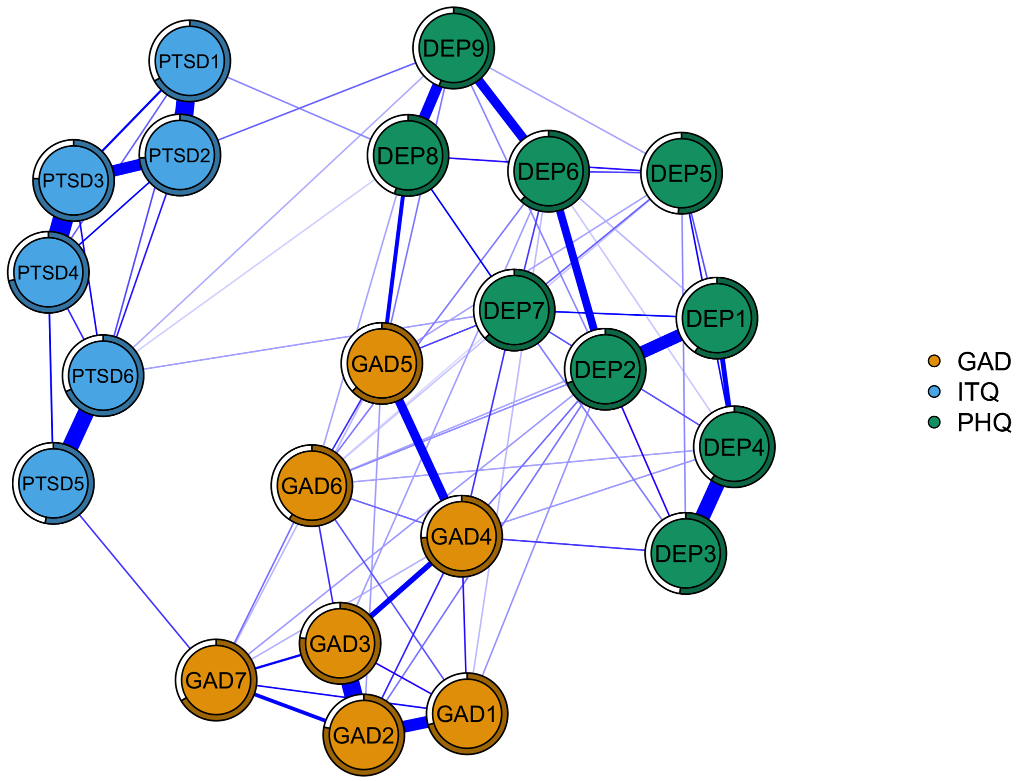
What does this figure tell you about depression, anxiety, and trauma during the first days of lockdown in the UK?

```
## Group the Mental Health syndromes by items
groups <- c(rep('PHQ', 9), rep('GAD', 7), rep('ITQ', 6) )

## Calculate predictability of each node in the network
pred_net <- predict(object = mgm_net,
                    data = df.sub.w1,
                    errorCon = 'R2')

pred_net$error  # Print predictability
```

```
##    Variable    R2
## 1  W1_Dep_1 0.592
## 2  W1_Dep_2 0.694
## 3  W1_Dep_3 0.524
## 4  W1_Dep_4 0.597
## 5  W1_Dep_5 0.513
## 6  W1_Dep_6 0.620
## 7  W1_Dep_7 0.628
## 8  W1_Dep_8 0.555
## 9  W1_Dep_9 0.559
## 10 W1_GAD_1 0.706
## 11 W1_GAD_2 0.781
## 12 W1_GAD_3 0.772
## 13 W1_GAD_4 0.744
## 14 W1_GAD_5 0.630
## 15 W1_GAD_6 0.595
## 16 W1_GAD_7 0.665
## 17 W1_PTSD1 0.668
## 18 W1_PTSD2 0.738
## 19 W1_PTSD3 0.758
## 20 W1_PTSD4 0.716
## 21 W1_PTSD5 0.533
## 22 W1_PTSD6 0.686
```

```
## Create the network figure
fig3 <- qgraph(mgm_net$pairwise$wadj,
               layout = "spring",
               pie = pred_net$error[, 2],  # Provide errors as input
               groups = groups,            # Group items by disorder
               labels = node.names,
               palette = 'colorblind',     # Use this colour scheme
               posCol= 'blue',             # Show positive edges in blue
               negCol = 'red',             # Show negative edges in red
               filetype = 'png',
               filename = 'figure3'
               )
```

```
## Output stored in /Users/user/Library/CloudStorage/Dropbox/_TEACHING/Appstate_Teaching/__POLITICAL_AN
```



Lastly, we can calculate the *centrality* of each node in the network using the `centrality_auto()` function from the `qgraph` package on the plotted network figure.

The most commonly used centrality metrics are node *strength*, which is a sum of the absolute edge weights of edges for each node node; *closeness*, which is the distance between the node and all other nodes (averaged over the shortest paths to all other nodes); and *betweenness*, which is how often a node lies on the shortest path connecting any two other nodes. These metrics have been directly adapted from social network analysis. While strength conveys how strongly the relevant variable is conditionally associated with other variables in the network, closeness and betweenness treat association as distances, which can be problematic in psychometric network analysis because of the underlying estimation procedure.

As a result, we use different measures of centrality developed specifically for psychometric network analysis. One such measure is called *expected influence*, which takes the sign of edge weights into account; this can be appropriate when variables have a non-arbitrary coding (e.g., high values indicate greater psychopathology). Thus, in contrast to the above-mentioned measures of centrality, which quantify the position of a node within a network, expected influence aims to assess the nature and strength of a node's cumulative influence within the network.

What do you see? Which nodes are most central to the network? Do these centrality estimates fit with the original plotted network?

```
## Estimate node centrality of graphed network
centrality <- centrality_auto(fig3)

## Extract the centrality estimates
nc <- centrality$node.centrality

## Print the centrality table
```

```r
nc
```

```
##       Betweenness   Closeness  Strength ExpectedInfluence
## DEP1            8 0.002692617 0.8168817         0.8168817
## DEP2           27 0.002926279 1.0199322         1.0199322
## DEP3            7 0.002506701 0.6814367         0.6814367
## DEP4            4 0.002448293 0.8815884         0.8815884
## DEP5            2 0.002383932 0.6982438         0.6982438
## DEP6           24 0.003004203 0.8945717         0.8945717
## DEP7            6 0.002952561 0.8994091         0.8994091
## DEP8           38 0.003215466 0.8624833         0.8624833
## DEP9           47 0.003183304 0.8173826         0.8173826
## GAD1            0 0.002650327 0.8455406         0.8455406
## GAD2           11 0.002889637 1.0781093         1.0781093
## GAD3           25 0.002966244 1.0217914         1.0217914
## GAD4           36 0.003122951 1.1338442         1.1338442
## GAD5           25 0.003068542 0.8327243         0.8327243
## GAD6            0 0.002370802 0.7388608         0.7388608
## GAD7           31 0.002729442 0.7785975         0.7785975
## PTSD1           0 0.002216755 0.7695291         0.7695291
## PTSD2          38 0.002434980 1.0156318         1.0156318
## PTSD3          12 0.002287049 0.9940966         0.9940966
## PTSD4           6 0.002213824 0.9066426         0.9066426
## PTSD5          27 0.002298921 0.6111136         0.6111136
## PTSD6          16 0.002300177 0.9174195         0.9174195
```

```r
## Now let's make a pretty plot of key centrality statistics
centrality_plots <-
  centralityPlot(fig3,                                # Use the network figure
                 scale = c("z-scores"),               # Plot standardized coefficients
                 include = c("ExpectedInfluence",     # Estimate these centrality stats
                            'Betweenness',
                            'Strength'),
                 theme_bw = TRUE,                     # Black and white theme
                 print = FALSE,                       # Show the figure
                 weighted = TRUE,                     # Use weighted edges
                 signed = TRUE,                       # Use positive and negative edges
                 orderBy = "default",                 # How to sort the stats
                 decreasing = TRUE)                   # From A to Z
```

```
## Note: z-scores are shown on x-axis rather than raw centrality indices.
```

```r
ggsave('centrality_plot.png')  # Save the centrality plot
```

```
## Saving 6.5 x 4.5 in image
```

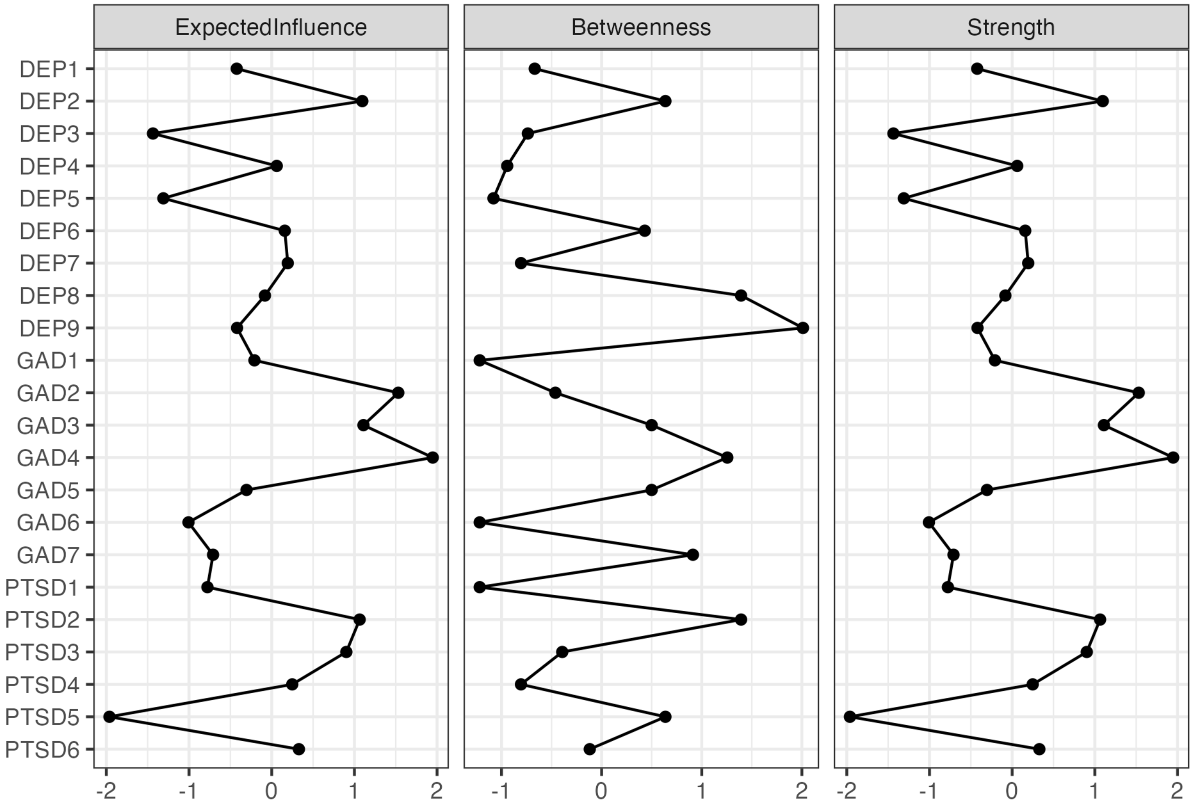


Figure 1: Figure 4