Hello and welcome to your Network Analysis course. Thanks for starting your work on the first assignment. In the assignment, you will be asked some conceptual questions, and you will also be asked to do run a few lines of R code. Please submit the assignment into the vault before the deadline in one of the following forms:

1) a .pdf with your answers and an R script with reproducible code, or
2) an R markdown document compiled from both.

Let's roll!

CONCEPTUAL QUESTIONS (2.5p):

1) Every now and then, I meet someone who happens to know a person that has been facing some serious mental health issues. This someone would often say that the person they refer to has a "screw loose". Reminds me of the common cause paradigm. Envision yourself in my situation, and try to enlighten the person, speaking against the "screw loose" based on your new knowledge of Network theory. Max. 400 words, written in clear English, no references needed. **1.5p**

BONUS QUESTION: I can imagine that with some particular mental health problems, the people may be right about the "screw loose theory". Think of one example (hint: recall the ongoing political situation) and support the folk theory of your discussion partner with your sophisticated pro Common Factor Model Argumentation. **0.5b**

Note: you will not be judged on your clinical knowledge, I only care about the epistemic argumentation

2) Choose one of the papers on the interactive canvas, and skim through it (you don't need to read it in detail to answer this question right) and answer to one of the prompts below using the argumentation hinted in the paper. Max 250 words, written in clear English, no references needed. **1p.**

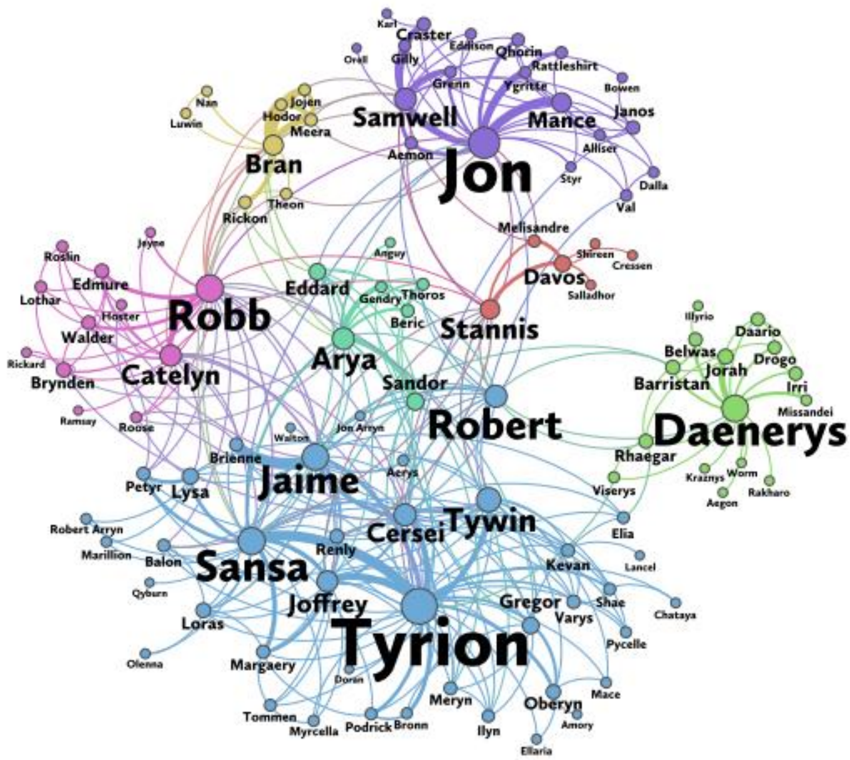
2a) It does not make sense to create developmental interventions for children from disadvantaged environment, as their future performance is largely given by their levels of intelligence, and on such, as a common cause, you cannot intervene (van der Mass et al., 2019).

2b) Some people are inherently more likely to develop any mental health issues regardless their nosological unit. We should focus our research on identifying the neurological roots of the shared liability, rather then on the specific nature of the complaints (van Bork et al, 2017).

NETWORK VISUALISATION (3.5p):

3) At the beginning of the new year, HBO will introduce their new series, The House of Dragon. Are you excited as well? Before immersing yourself in the Westeros reality, it would be nice remember the famous counterpart, Game of Thrones.

In these exercises, we will analyse a weighted social network of book series A Song of Ice and Fire, on which the popular TV show Game of Thrones is based. This network was published in Math Horizons Magazine (Beveridge & Shan, 2016), and is based on the third book, A Storm of Swords, on which the third and fourth season of the TV series are based. More information, including the data, is available [online](). The network published is the following:



3a) Look through the article to get an idea of what this network represents. What do the nodes and edges represent? **0.5p**

```
Data <- read.csv("stormofswords.csv")
```

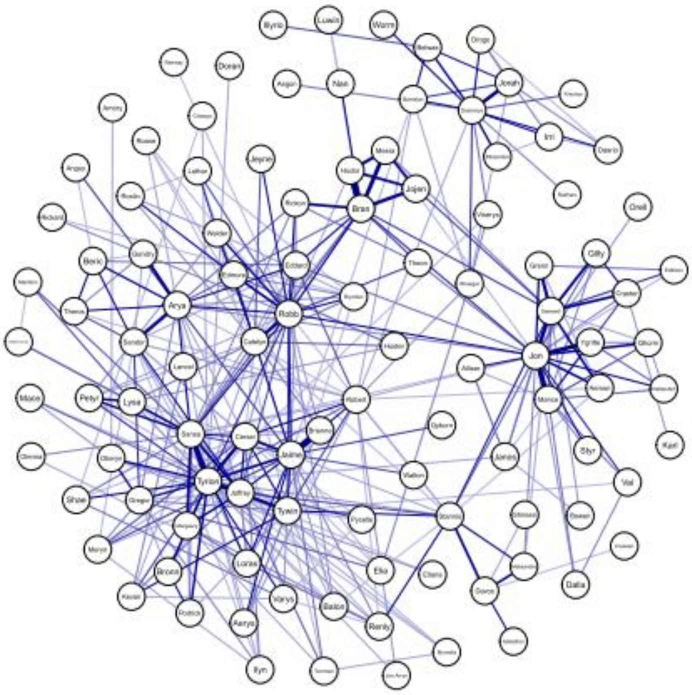
3b) Look at the data in RStudio using the `View` function. This matrix encodes a network. Can you figure out how? What do the rows stand for and what do the columns stand for? **0.5p**

This structure is known as an *edgelist* encoding a network, which can also be used as input for `qgraph`:

```
library("qgraph")

qgraph(Data, directed = FALSE)
```

3c) When plotting an undirected graph using an adjacency or weights matrix as input I normally do not have to set the `directed` argument. Now that I use an *edgelist*, however, I do. Why? **0.5p**

The plotted network is plotted using a circular layout. This circular layout, however, is hard to interpret and read in such networks with many nodes. In addition, I like to plot edges with a different colour than green when they represent values that can only be positive. Finally, the nodes are very large and we might want to make them smaller:



3d) Recreate the plot above, changing the layout to a spring layout, the edge color to ‘darkblue‘ and the node size to 3. Look at the `qgraph` help page to figure out the commands needed (`?qgraph`; or the link in your study materials). Note: your computer might generate a different spring layout (nodes placed on different locations). **0.5p**

`qgraph` uses three arguments that need to be known to interpret a network: `minimum, cut, and maximum`. These are set automatically, and can be shown using the argument `details = TRUE`.

3e) What values did `qgraph` set to `cut` and `maximum`? Note: `minimum` is always set to 0 by default and not shown with `details = TRUE` unless it differs from 0. **0.5p**

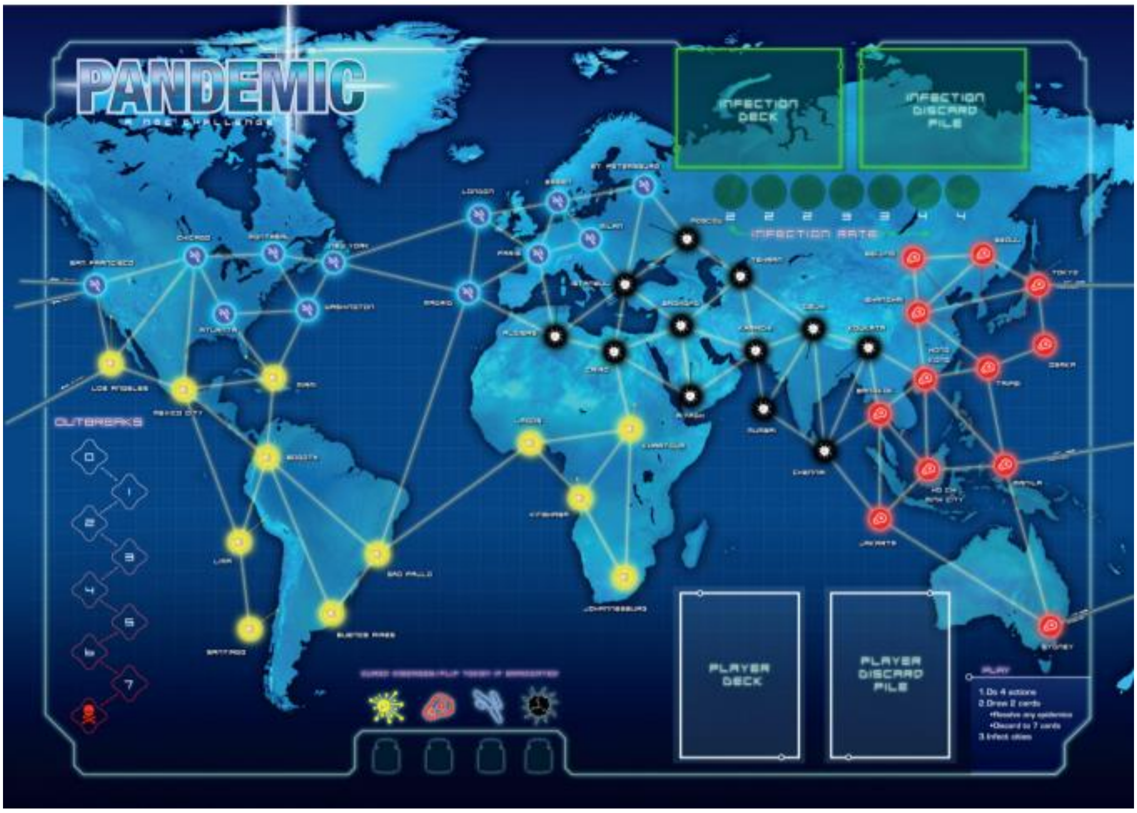
The `minimum` argument can be used to hide edges with an absolute (negative edges are treated as positive) weight below some value. Note that these edges are only visually hidden, not removed in further analyses (which can be done using the `threshold` argument). This argument is useful when plotting dense graphs (e.g., correlation networks) but not recommended in the networks estimated in this course.

3f) Set the `minimum` argument to 1, 10 and 20 while using a *spring* layout. How does the network change? Do the same using the `threshold` argument. Can you explain why the layout remains the same using `minimum` but changes using `threshold`? **0.5p**

3g) What would be a good setting for `maximum` when drawing networks based on (partial) correlations? **0.5p**

NETWORK INFERENCE (4p):

4) There is an interesting boardgame called [The Pandemic](). We can all see how it has become especially relevant recently. A blue virus will emerge in the blue cities (nodes), a yellow virus in the yellow cities, a black virus in the black cities and a red virus in the red cities. Once there is an outbreak of a virus in a city, it may spread to a connected city via the edges. Fortunately, we have some tools to stop the viruses from spreading. For example, we can place a city under quarantine (haha), effectively removing the node from the network.



4a) Only by looking at the above picture (you can look up a more detailed version online), can you derive: **0.5p**

• The degree-centrality of Paris?

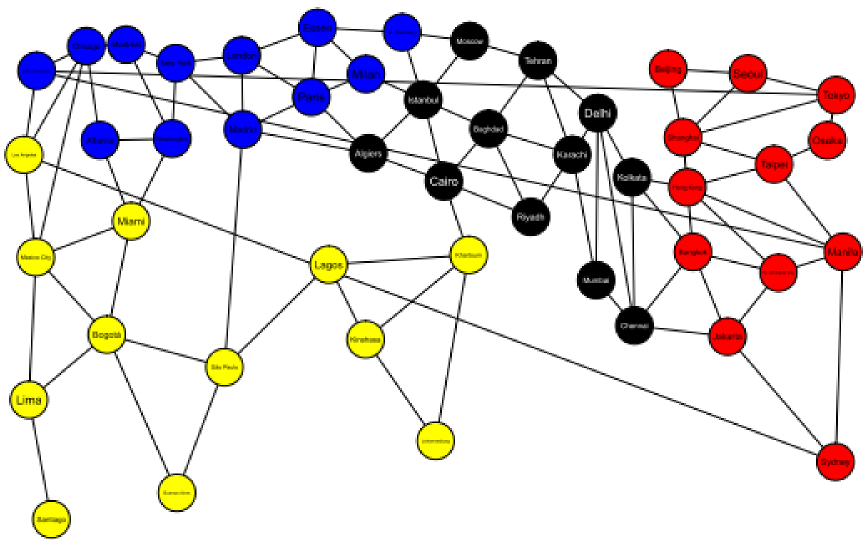
• The shortest path length between Milan and New York, and how many paths there are of this length?

• Of all shortest paths between all pairs of nodes, how many go through Santiago?

Luckily, the hard work of encoding the network structure was already done online. We can read the network into R using the following codes:

```
# Load network:

library("igraph")

Graph <- read_graph("http://files.indicatrix.org/pandemic.graphml",
format = "graphml")

# Extract edgelist:

Edgelist <- get.adjacency(Graph)

# Labels:
```

```
Labels <- V(Graph)$label

# Layout:

Layout <- cbind(x = V(Graph)$x, y = V(Graph)$y)

# Color:

Color <- c( rep("blue",12), rep("black",12), rep("red",12),
rep("yellow",12)

)

# Plot in qgraph:

library("qgraph")

qgraph(Edgelist, labels = Labels, color = Color, layout= Layout,
directed = FALSE, vsize = 4, esize = 1, edge.color = "black")
```



Now that we can visualize the network, we might wish to analyse it further. To do this, we first need to store the network:

```
Graph <- qgraph(Edgelist, labels = Labels, color = Color, layout=
Layout, directed = FALSE, vsize = 4, esize = 1, edge.color =
"black")
```

One option is to analyse how important or central nodes are in the network. These can be analysed using the centrality function:

```
Centrality <- centrality(Graph, all.shortest.paths = TRUE)
```

Node strength (also called degree) sums the connected edge weights to a node. The node strength of the city Miami is:

```
Miami <- which(Labels == "Miami")

Centrality$OutDegree[Miami]
```

```
## Miami
##     4
```

Closeness computes how "close" two nodes are together in the network:

```
Centrality$Closeness[Miami]
## Miami
## 0.005291005
```

Finally, Betweenness computes how often one node is featured in the most efficient (shortest) paths between other node:

```
Centrality$Betweenness[Miami]
## Miami
## 42.81905
```

4b) Compare the centrality of "Bangkok" and "Atlanta" and attempt interpretation. **1p**

4c) List the most and least central node according to degree, closeness and betweenness. If multiple nodes are equally central, list them all. **1p**

I can investigate the degree of all red virus cities as follows:

```
Red <- which(Color == "red")
Centrality$OutDegree[Red]
```

4d) Recently, two new centrality measures have been added into `qgraph`**: predictability and expected influence.** Try to ask for them as well, I am sure that by now you can figure out how. 😉 (if not, consult the CRAN manual or the `?qgraph` call). Why are the expected influences equal to the degree centrality (hint: get back to the presentation) **0.5p**

Whoa, the world is about to explode – the predictability measure did not return a valid piece of information, only a warning!

BONUS QUESTION: Try to explain why you cannot compute the predictability indices with this particular graph. You will not get any points if you only copy-paste the warning message from the R console **1p**

4e) Compute the average degree, closeness and betweenness per virus (colour). Which virus do you think is the most dangerous? Which node is the most influential? What would you advise based on this knowledge? **1p**