

SOCIAL NETWORK ANALYSIS for DATA SCIENTISTS

today's menu: NETWORK VISUALIZATIONS (LECTURE LAB Week 05)

Your lecturer: Gergő

Playdate: September, 23rd, 2025

Menu' for today



1. In the making of a viz

1. What

2. Why

3. How

2. Homeplay

3. Tricks or treats

4. Final Remarks

"As network researchers our biggest advantage in communicating our impact is the really pretty pictures we can make."

- *Anonymous Network Scholar at a conference*

I. In the making of a viz

What's a viz?



What's a viz?



VISUALIZATION = MESSAGE

Metaphor for some parts of reality or concepts that we are explaining

Anatomy of a visualization



What are you working with?

Why are you making the viz?

How can you best achieve this goal?

What are you working with?



What are you working with?



Taking stock of your resources

What are you working with?



Taking stock of your resources

You have a network

What are you working with?



Taking stock of your resources

You have a network

What about attributes?

- Nodes/edges?
- Categorical? Continuous? Ordinal?
- Temporal stuff?
- Missing data?

What are you working with?



Taking stock of your raw materials is the foundation of the visualization process.

Why are you making the visualization?



What's the goal of the viz?

What is the message you want to get across?

Who is your audience?

Why are you making the visualization?



Discover? Present? Enjoy?

Why are you making the visualization?



Discover? Present? Enjoy?

Compare? Identify? Summarize?

Why are you making the visualization?



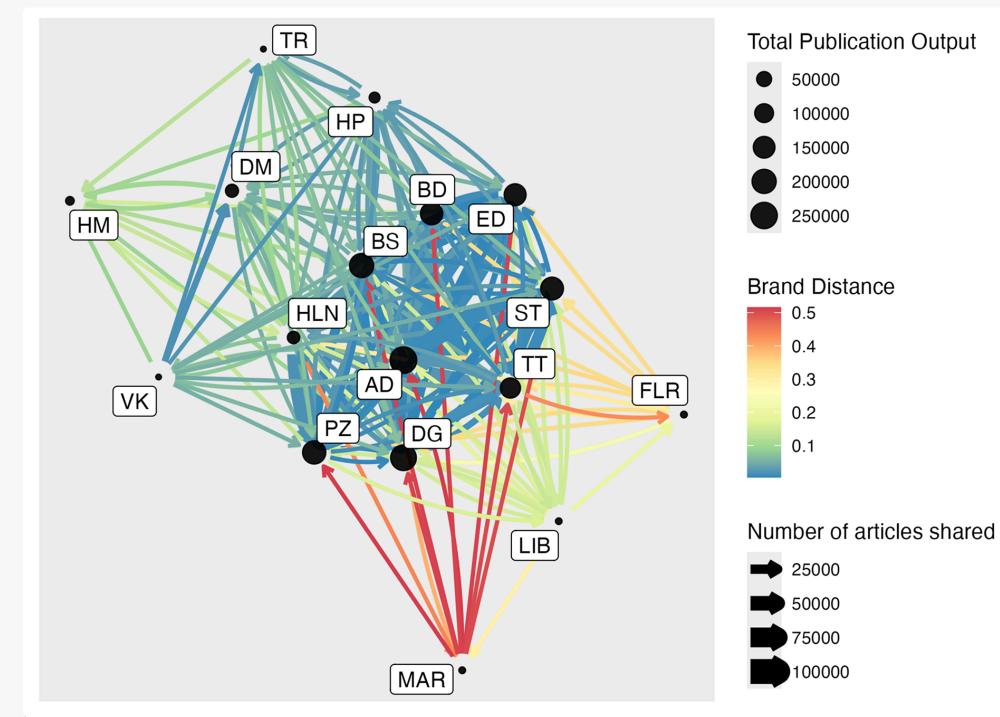
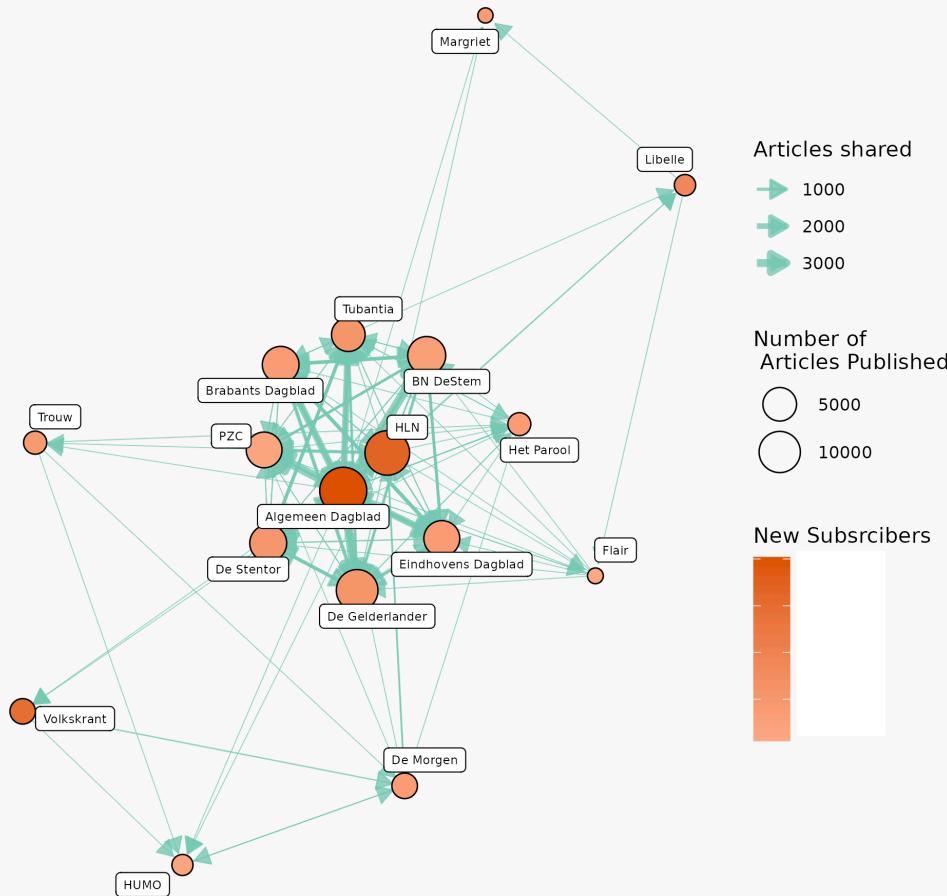
Discover? Present? Enjoy?

Compare? Identify? Summarize?

No right or wrong, just make sure you're clear on this.

Some skin in the game: my visualizations

Content Sharing Network in 2023 August



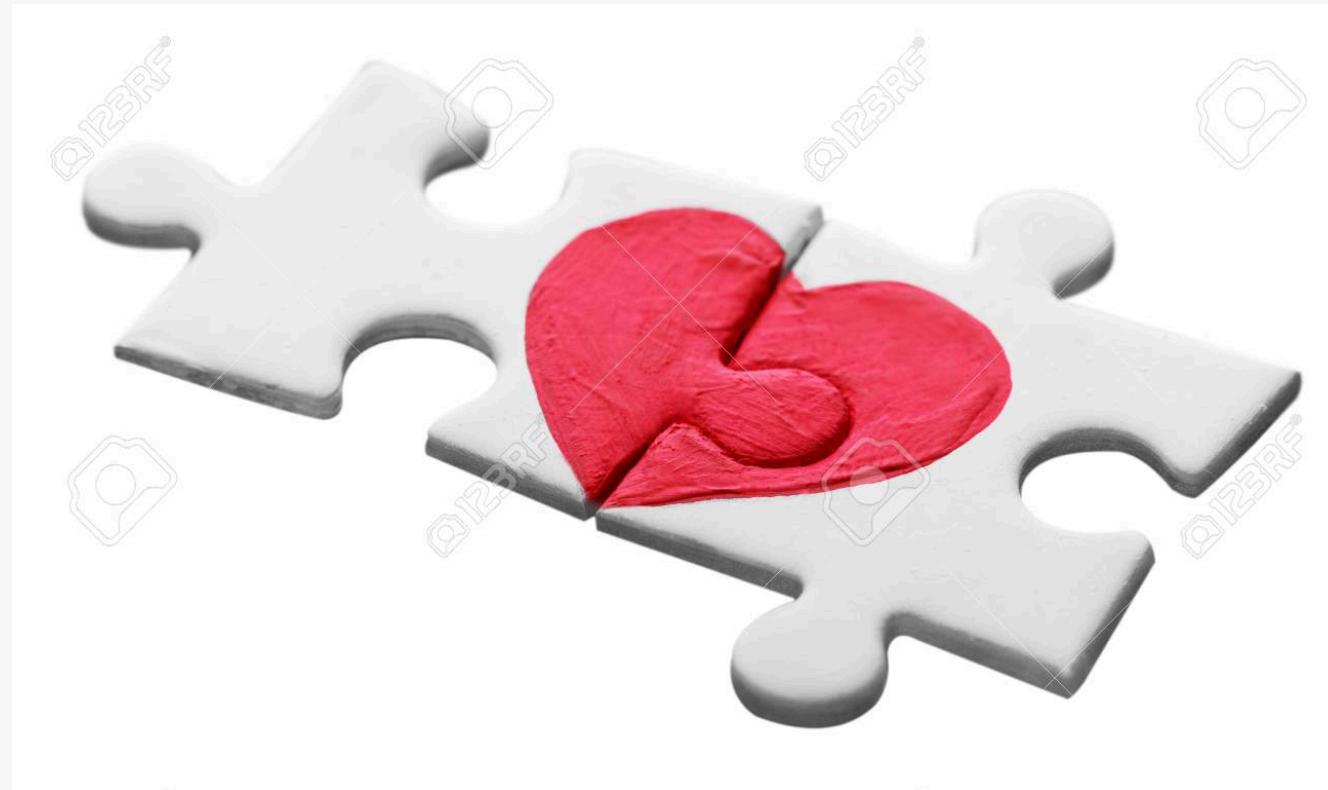
How you can best achieve these goals?



In this class it's `igraph`. Still, which `igraph` functions are most useful for your goals?

Get to know your tool, make the most of it.

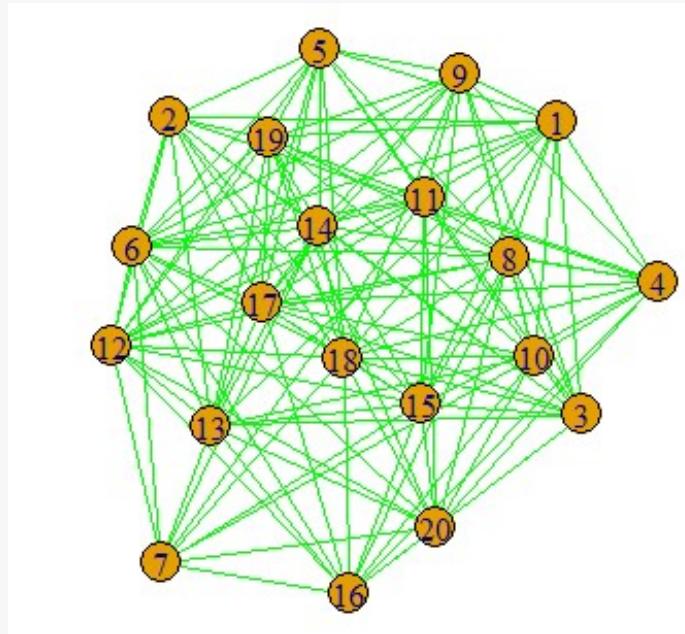
c. Does your strategy work with your data
and with your tool?



Example 1

You want to show passages in a network of football players.

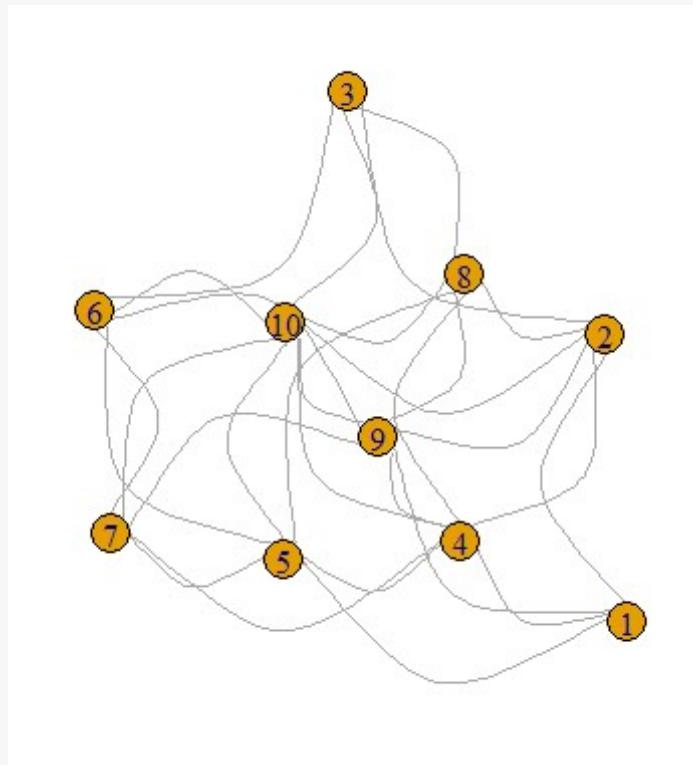
Since the football court is green you decide it would be impactful to use the color green for the edges (passages).



Match the idea with the reality of the tools you have: It is theoretically a nice, but impossible to read it.

Example 2

You are plotting a network of dancers and you decide that using curved edges is more romantic.



Match the idea with the reality of the tools you have: You might be artistically right, but it gets really hard to see where the edges are going

d. Fancy versus functional

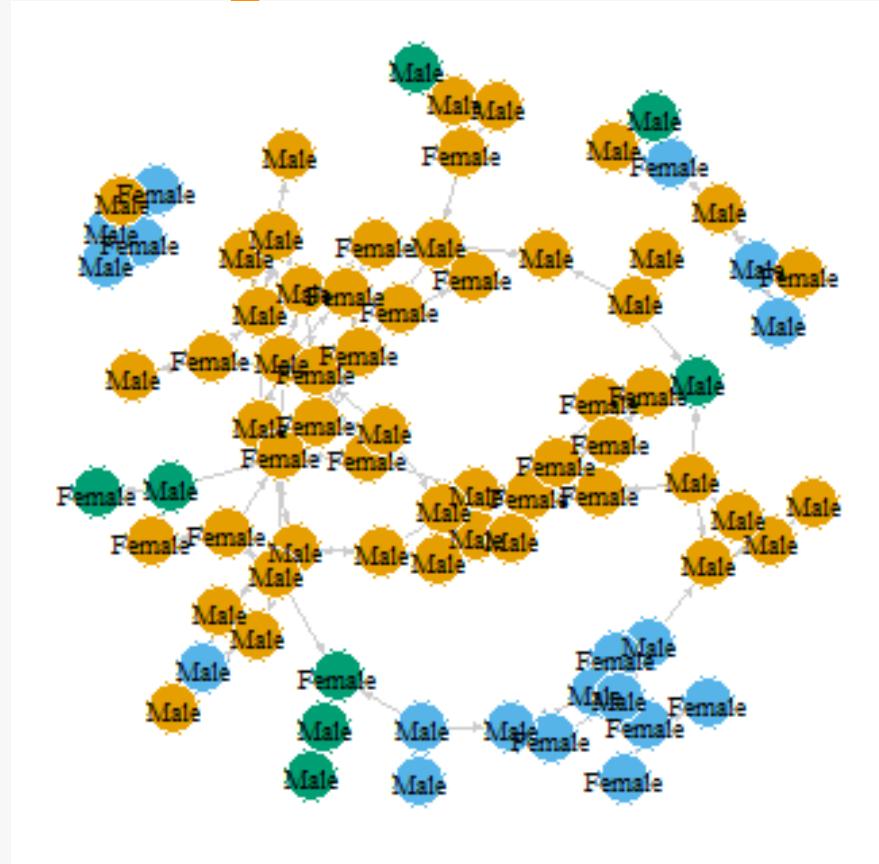


You want to insert too much information in a very elaborated way

- many colors
- weird node shapes
- labels on the nodes and edges
- ...

Networks are a bit too powerful for their (your) own good.

Tutorials plot: What's the message? Is it clear?



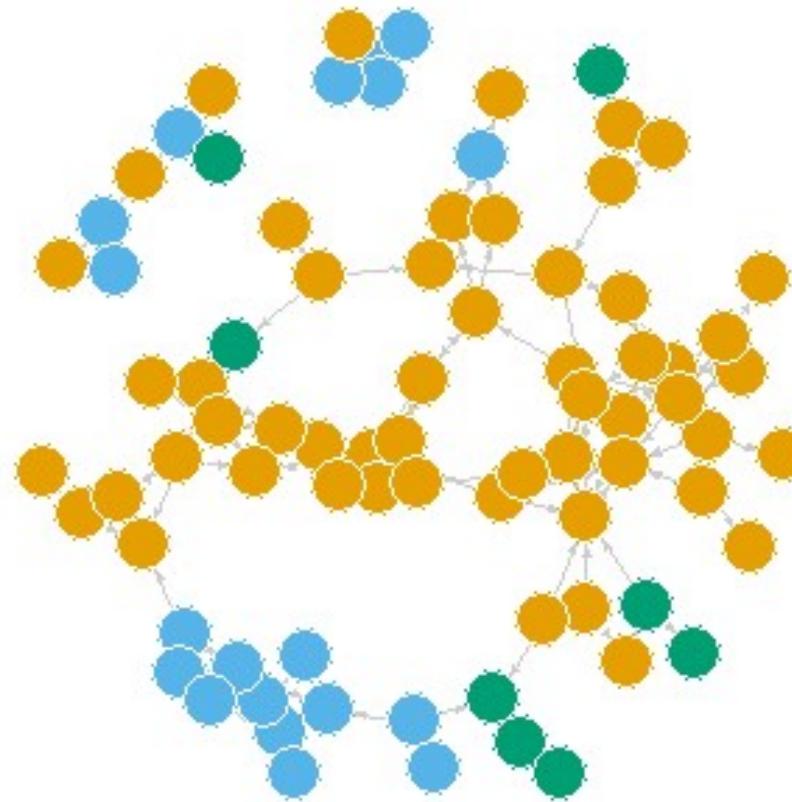
Can you see social dynamics? Can you plot it better to show homophily?

**LESS
IS
MORE.**

[MORE OR LESS]

Better?

homophilic effects by year



2. Homeplay

What is this network about?



Load the network “UKfaculty”

81 nodes

817 edges

1 nodes attribute (school affiliation)

1 edge attribute (how often they are in contact)

Very simple story



There are people working in a university and they are organized in three groups.

[same as JADS: we have three units]

Story/Goal

You want to be able to show the pattern of contacts within groups and accross groups

Plot / Instruction



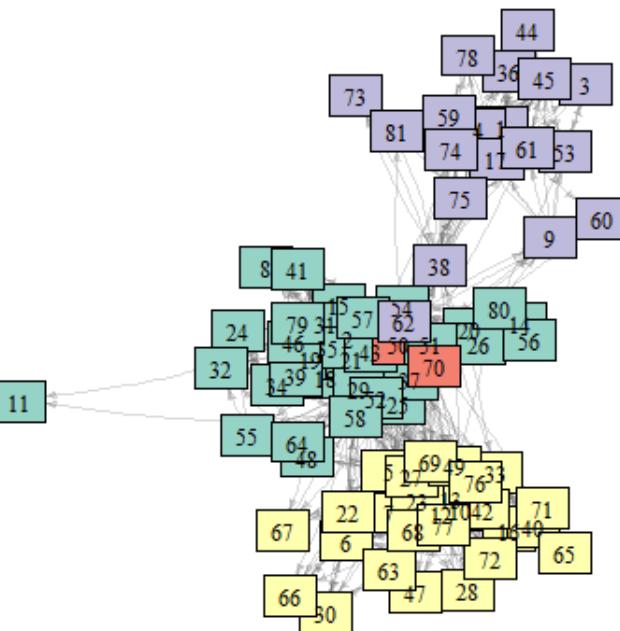
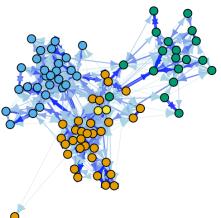
make a visualization using

- one edge attribute
- one node attribute

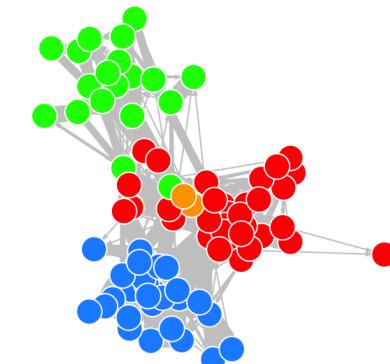
that are able to tell what is happening in this network.

Decide what is the best way to represent this story first and plot after.

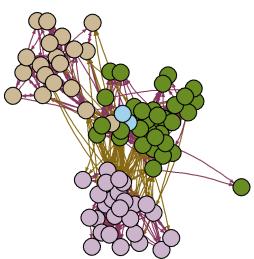
Graph of UK faculty member friendships



UK faculty by group and weight



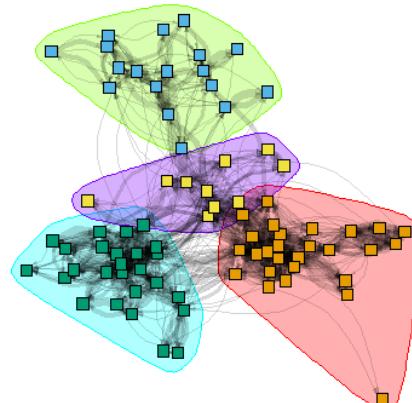
The Serious One



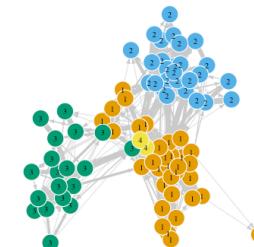
Faculty Group
● Group 1
● Group 2
● Group 3
● Group 4

Edge type
■ Within group
■ Between groups

Friendships Among UK Faculty Members

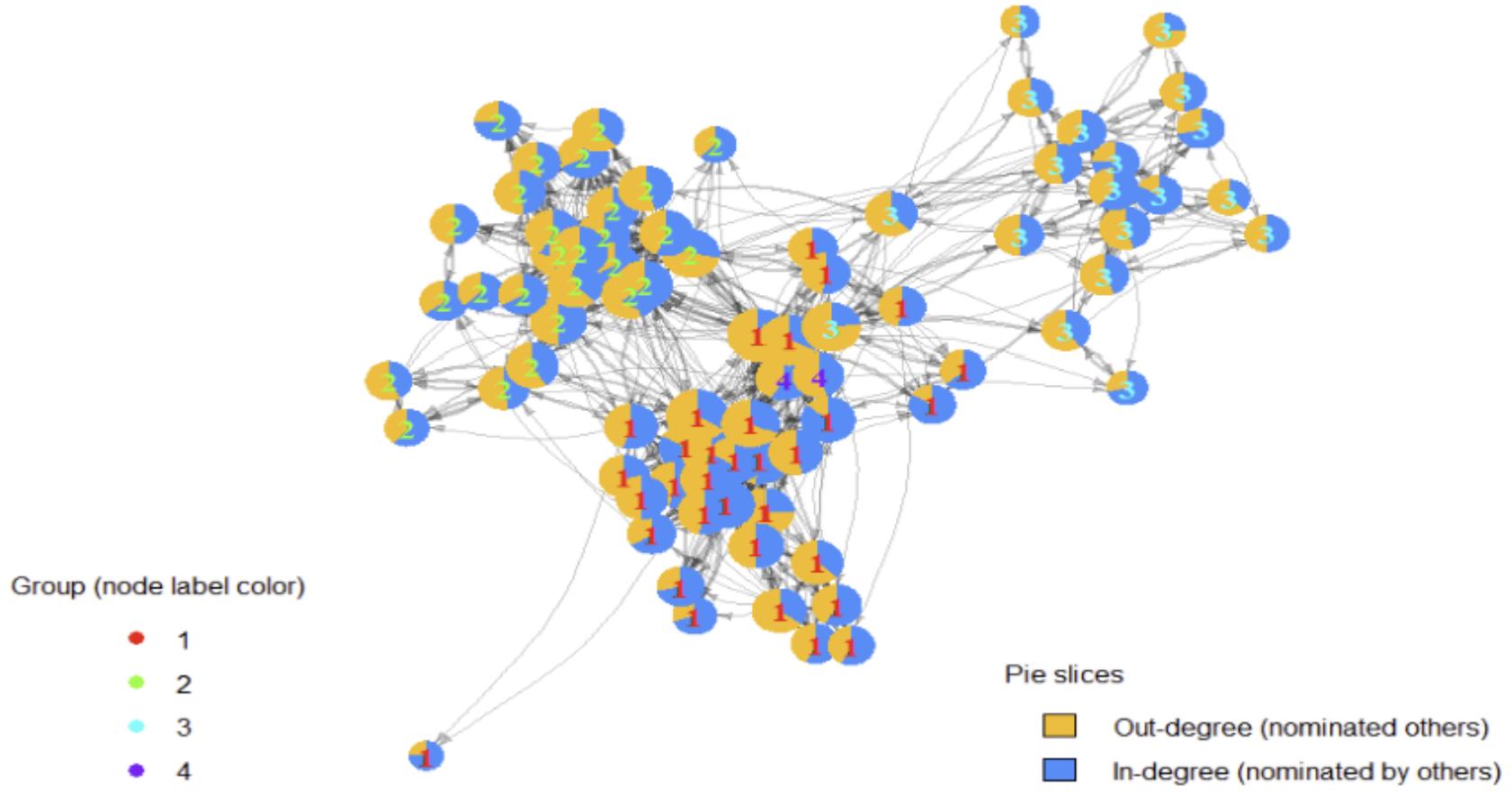


UK Faculty Friendship Network



Too powerful for it's own good?

UKfaculty — 81 nodes, 817 edges
Color = Group • Pie slices = Out vs In nominations



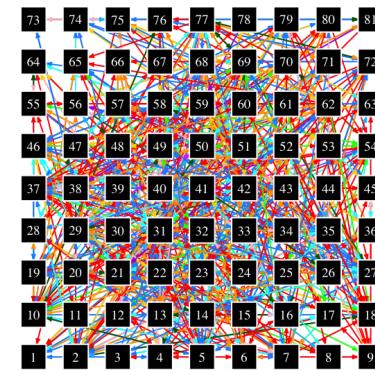
Plot 2 Instruction



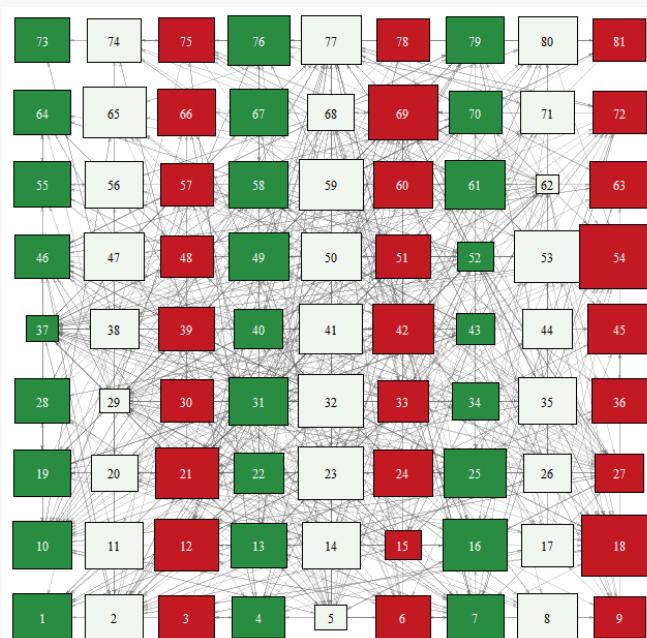
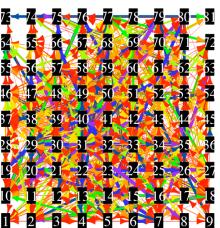
Plot the same graph, but with

- a lattice layout (just for fun)
- nodes with a rectangular shape
- black nodes
- white nodes labels
- a surprise element of your choice!

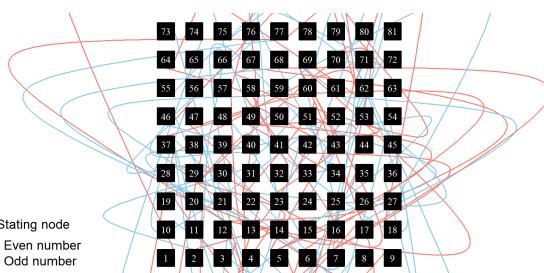
Rainbow edge plot (16 colours)



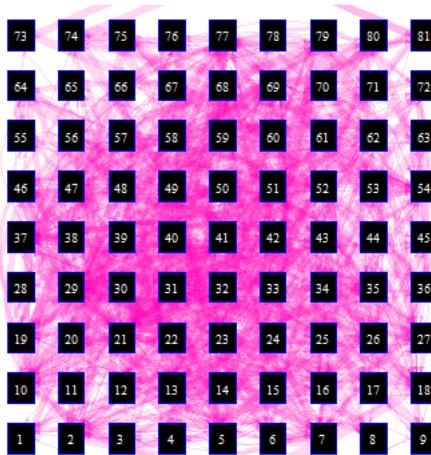
Chaos...because why not



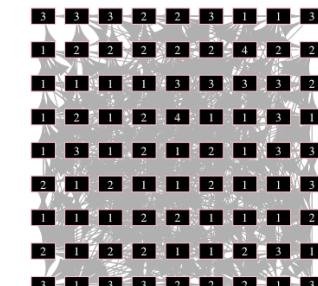
The "Funny" One



Friendships Among UK Faculty Members



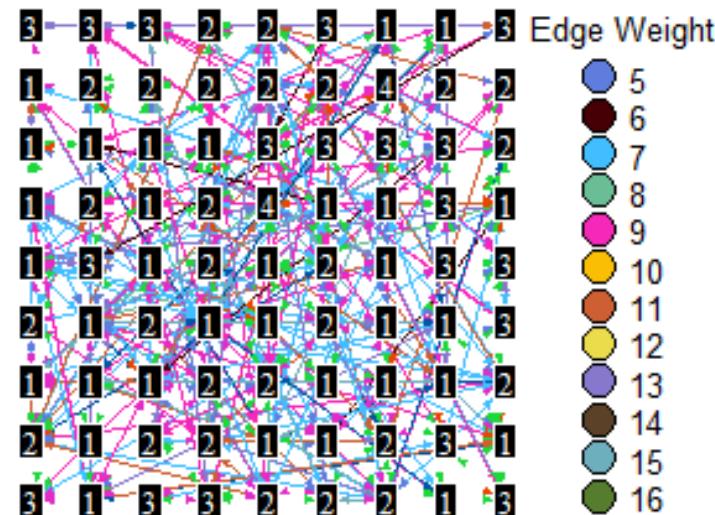
Silly UK Faculty Friendship Network



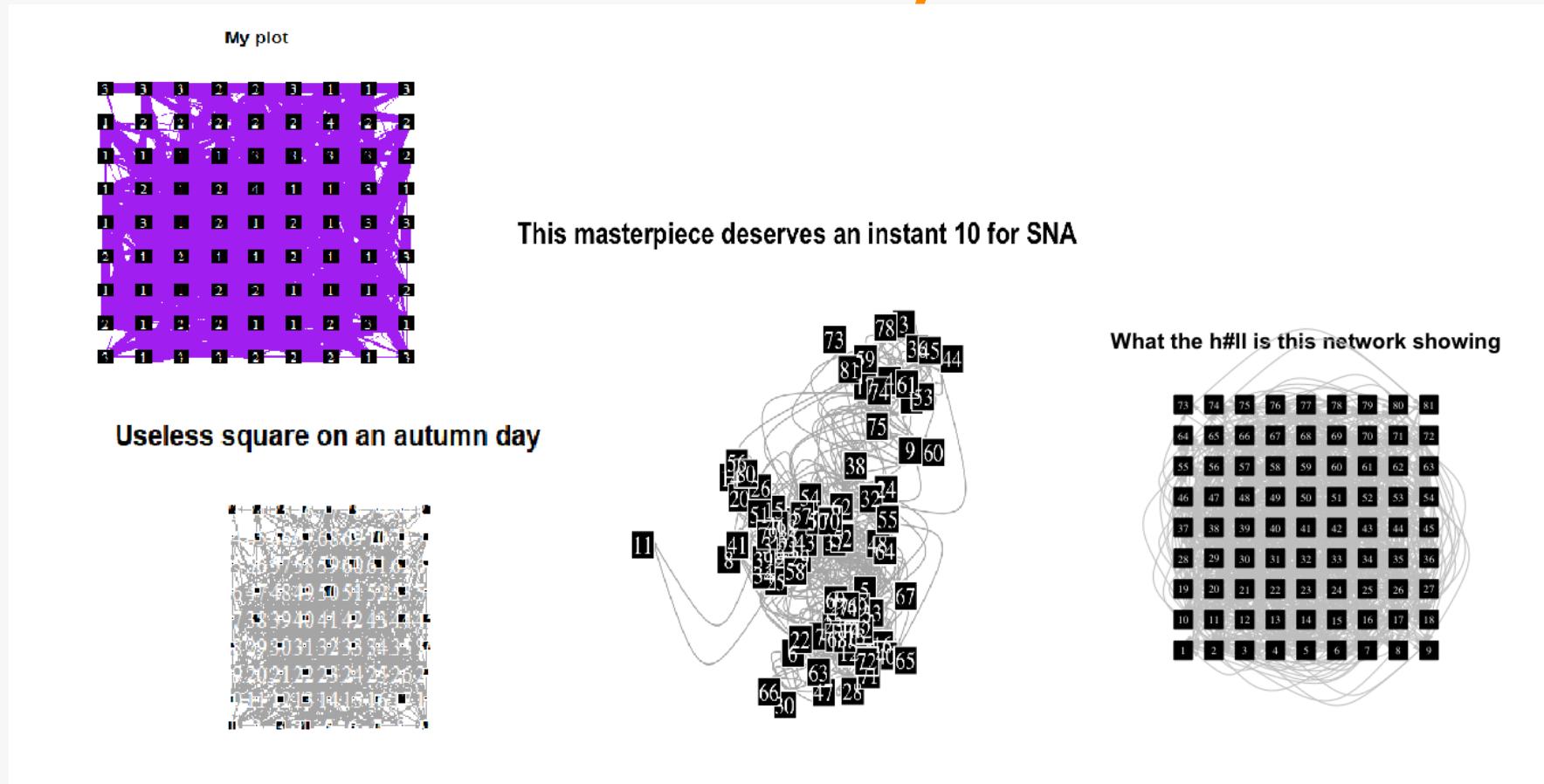
Plot 2 from one of your former colleagues



Research groups and their stronger relationship ties with random colours

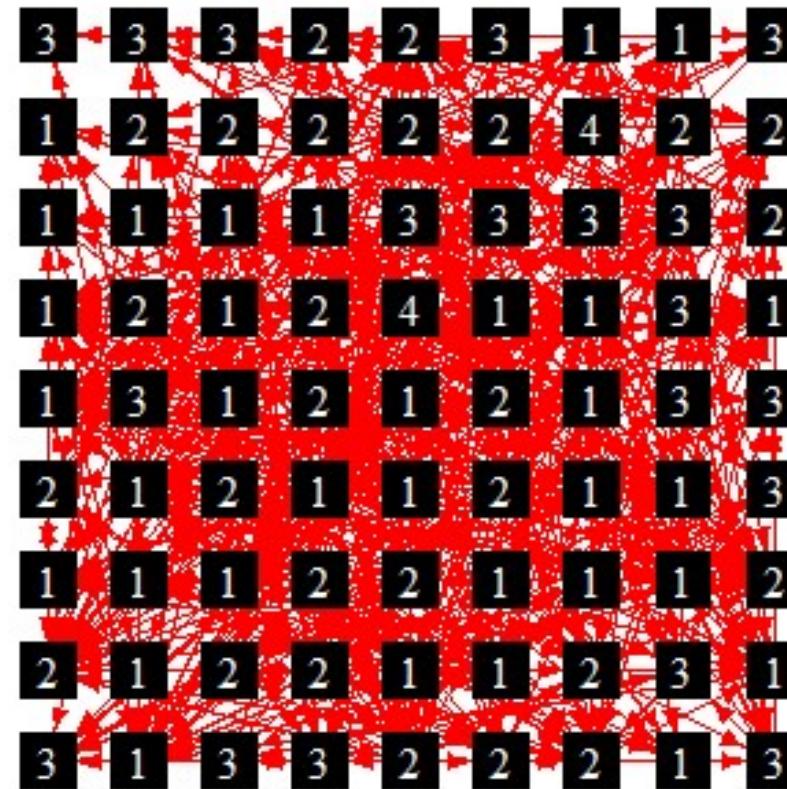


Plot 2 from some of last year's students



I'm funny and I know it!

I'm funny



3. Tricks or treats

Storing the color in the graph

In the tutorial: node color scale plot

```
# # Generate colors based on how often they hang out:  
# colrs <- c("cadetblue", "dodgerblue", "dodgerblue3", "blue", "darkblue" )  
# # create a new attribute with the color  
# igraph::V(net)$color <- colrs[igraph::V(net)$Hang.Out] # we attribute each category to a colour  
#  
# plot(net,  
#       edge.arrow.size = .2,  
#       edge.color = "gray80",  
#       vertex.frame.color = "#ffffffff",#node perimeter  
#       vertex.label = igraph::V(net)$gender,  
#       vertex.label.cex = 0.6, # vertex label size  
#       vertex.label.color = "black",  
#       vertex.size = igraph::V(net)$DS*2,  
#     )
```



Storing plot features BEFORE the function



In general, you can also set your preferences before the plot

Add attributes with the parameter names the same way you would add columns to a data frame

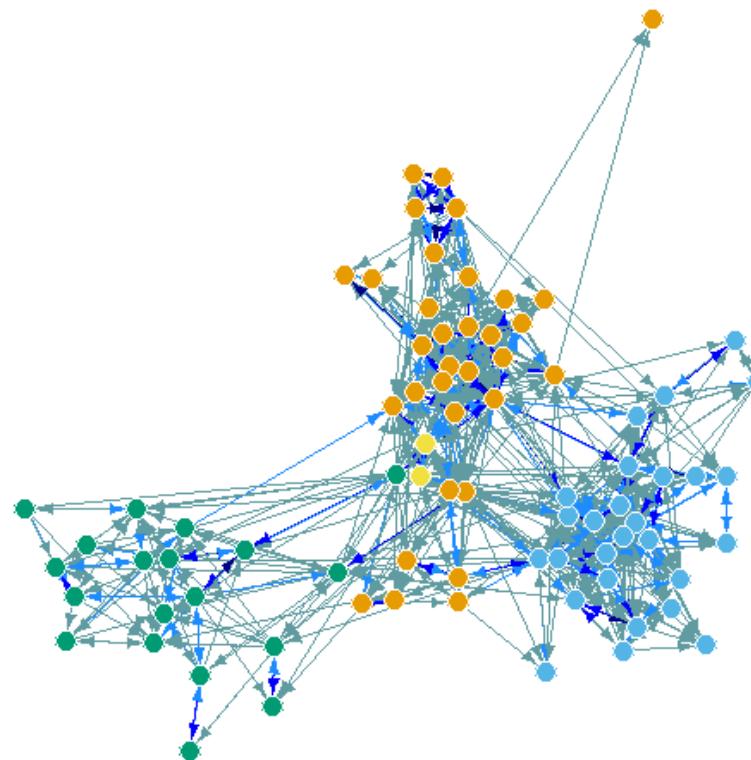
In the second way parameters remain stored

```
#####
# As we did it so far
#####
#
# plot(UKfaculty,
#       vertex.shape="circle",
#       vertex.label = NA,
#       vertex.color = igraph::V(UKfaculty)$Group,
#       edge.width = igraph::E(UKfaculty)$weight*0.5,
#       edge.arrow.size = 0.1,
#       layout = igraph::layout_on_grid,
#       main= "Homophilic behavior by group")

#####
# Alternative way
#####
#
# igraph::V(UKfaculty)$color <- igraph::V(UKfaculty)$Group
# igraph::E(UKfaculty)$width <- igraph::E(UKfaculty)$weight*0.5
# igraph::graph_attr(UKfaculty, "layout") <- igraph::layout_on_grid
```

Storing the weight of the edges as color

UKfaculty: Groups and relationships



- below 5 times
- between 5 and 10 times
- between 10 and 15 times
- over 15 times

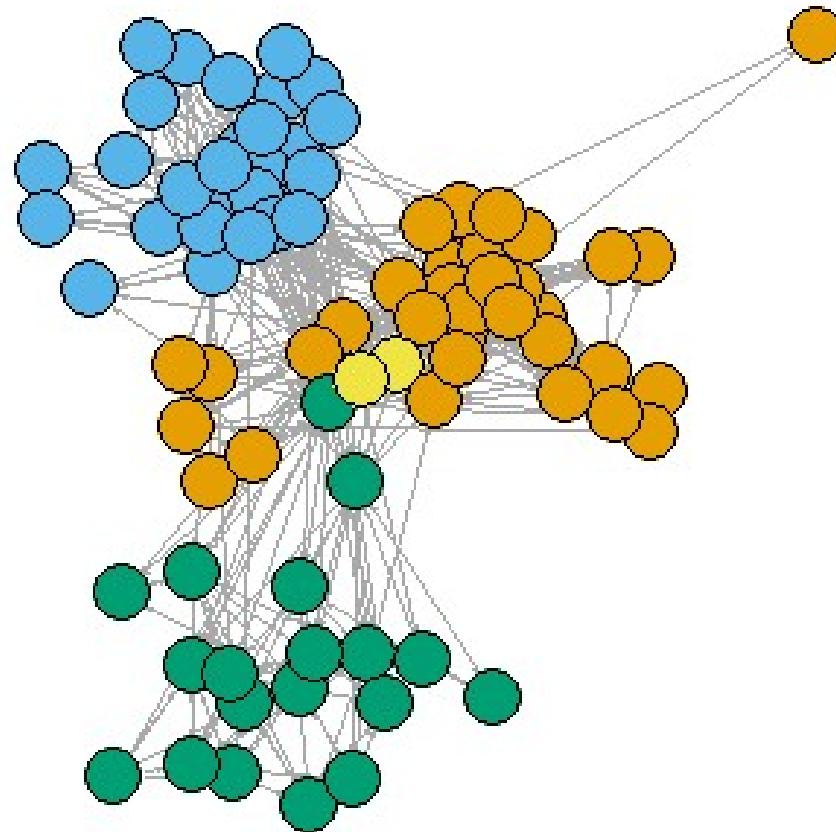
```
##### edge weight in color  
  
# table(igraph::E(UKfaculty)$weight)  
#  
#  
# igraph::E(UKfaculty)$Ecolor <- ifelse(igraph::E(UKfaculty)$weight <= 5, "cadetblue",  
#                                         ifelse(igraph::E(UKfaculty)$weight > 5 & igraph::E(UKfaculty)$weig  
#                                         ifelse(igraph::E(UKfaculty)$weight > 10 & igraph::E(UKfacul  
#                                         ifelse(igraph::E(UKfaculty)$weight > 15, "darkblue",  
#  
#  
#  
# table(igraph::E(UKfaculty)$Ecolor)  
#  
#  
# plot(UKfaculty,  
#       edge.arrow.size = .5,  
#       edge.size = 2,  
#       edge.color = igraph::E(UKfaculty)$Ecolor,  
#       vertex.color = igraph::V(UKfaculty)$Group,  
#       vertex.frame.color = "#ffffff",#node perimeter
```

Beyond the standard layout

This can be improved!

```
igraph::layout_with_fr
```

Homophilic behavior by group



How does the layout work?



The layout in igraph is defined in a matrix with 2 columns and a row for each node.

- The first column indicates its x position
- the second its y position

```
l <-igraph::layout_with_fr(UKfaculty)  
head(l)
```

```
##           [,1]      [,2]  
## [1,]  4.0739535  0.1090275  
## [2,]  0.8193353 -0.9541154  
## [3,]  5.0157268 -1.1062330  
## [4,]  4.0789881 -0.8188556  
## [5,]  1.1625098  0.8588182  
## [6,] -0.5412988  0.3943771
```

Modify an existing layout



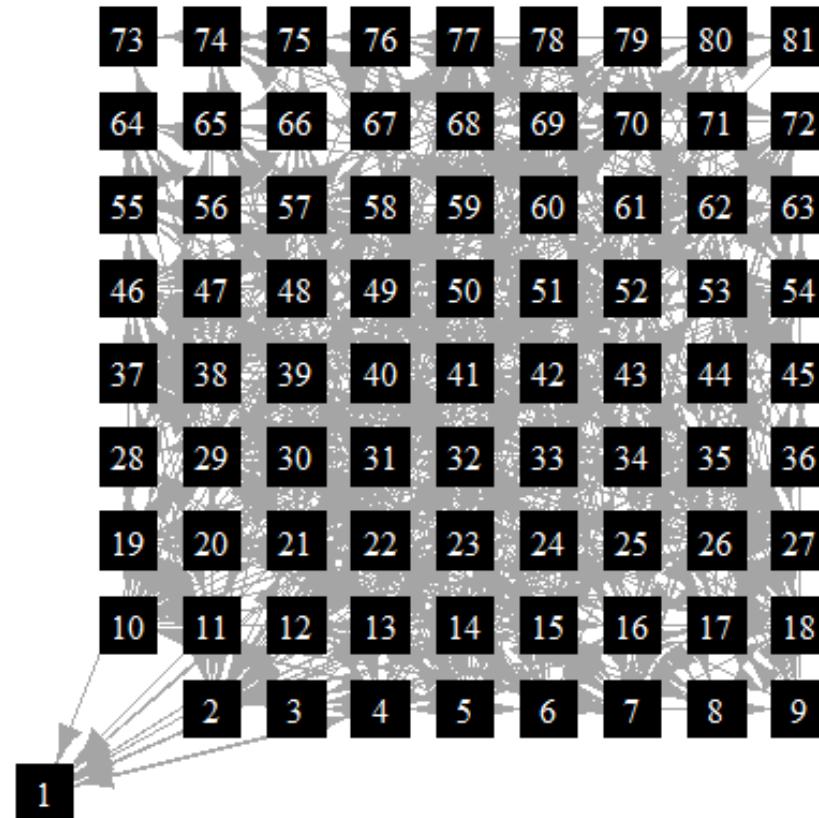
The scale is not relevant (it is always re-scaled to fit a -1 to 1 plotting area).

You can get this layout before plotting by just calling the layout function on the graph:

```
l <- igraph::layout_on_grid(UKfaculty)  
  
l[1, ] <- c(-1, -1)  
  
# plot(UKfaculty,  
#       vertex.shape="rectangle",  
#       vertex.label.color = "white",  
#       vertex.color = "black",  
#       layout = l,  
#       main= "My plot")
```

Beyond the layout

My plot



Move!

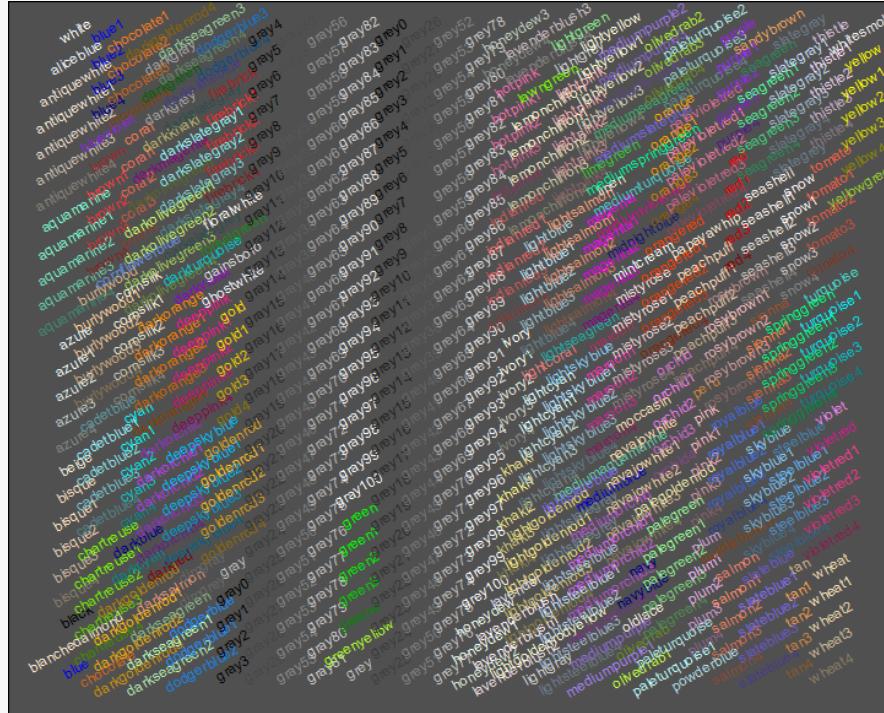


This cannot work in the slides!

```
# igraph::tkplot(UKfaculty)
```

Color palette

- hexadecimal color codes
- `demo("colors")` that are inbuilt in R `grDevices`
- several packages



4. Final Remarks

Your Book



... is plotting using `network`

that code won't work with `igraph` objects.

Still, the logic is very similar

`snafun::to_igraph` & `snafun::plot`

OR you can always use `intergraph` and plot in `igraph`

OR look up the corresponding parameters and plot in `network`

Parameters RECAP there are many more. These are the essentials



vertex:

- `vertex.shape`
- `vertex.color`
- `vertex.label`
- `vertex.size`
- `vertex.frame.color`

edge:

- `edge.arrow.size`
- `edge.color`
- `edge.width`

overall: `layout`

Some more JADS students Viz!



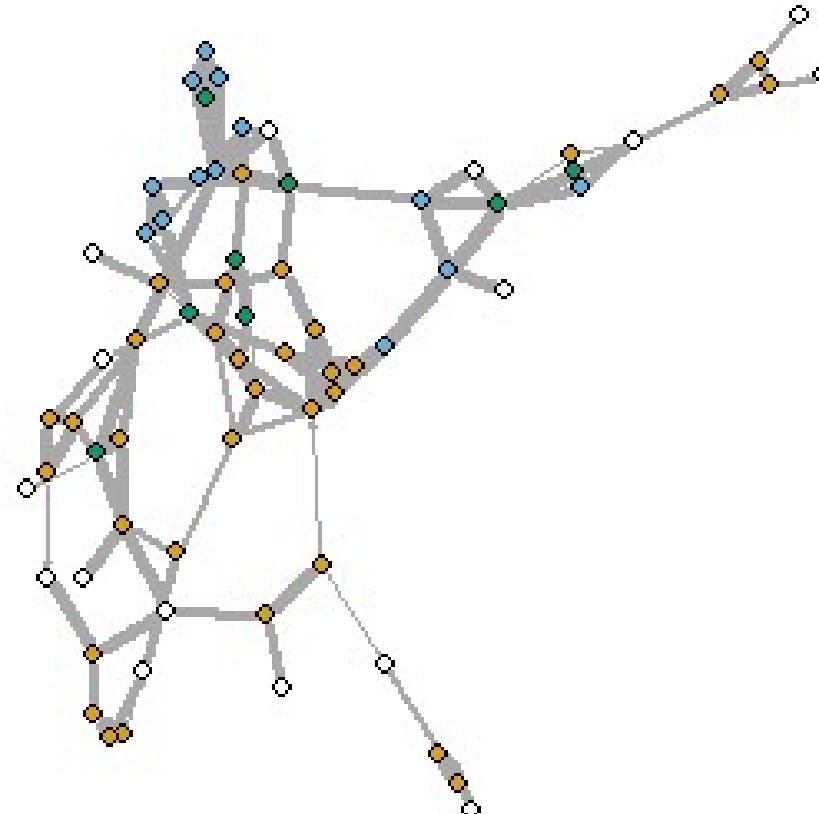
The tutorial data is from 2 years ago!

Still...

Let's take a look to JADS data!

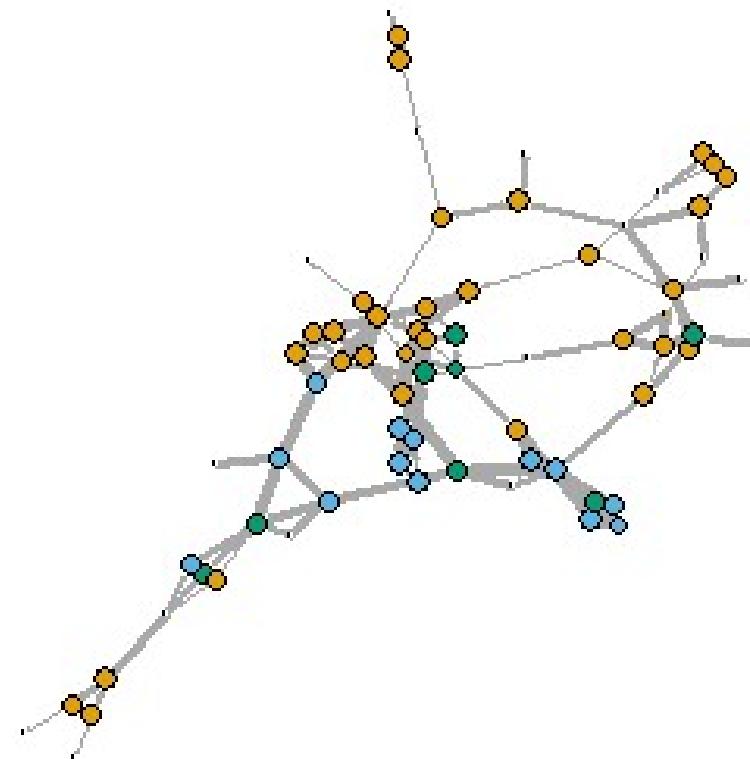
What does this image tell us?

Friendship by Nationality and Frequency



What does this image tell us?

Friendship by Nationality DS and closeness



Etenstijd!

