Introduction to Network Analysis

Materials to Accompany Lecture 2: Networks as a Method

Valentina Kuskova, PhD

Contents

Welcome!

Welcome to the second seminar! By now, you should be comfortable performing some of the basic commands in `R` and drawing networks. Today we will continue with this work: learn more ways to draw and explore various ways to manipulate networks.

Before attempting to run any code, please make sure you have every necessary package installed; however, pay attention. Today we will learn that some packages interfere with each other, so you will need to unload some before you can use others.

Contents of today's seminar:

- In your Seminar 2 folder, you will find the following files:
  1. This file, "Seminar 2 and Homework 1," in .pdf format.
  2. Dataset "flo.Rdata," which we've seen before; also, in the same format - "madmen.Rdata" and "trade.Rdata.""
- For today's assignment, you will need the following packages (remember, 'R' is case sensitive, make sure, when installing packages, to keep the appropriate case:
  1. 'rmarkdown'
  2. 'RColorBrewer'
  3. 'network'
  4. sna
  5. 'igraph'. *Please note:* this package may conflict with "network" package, especially on graphics. Therefore, when using one, it is always wise to unload the other (using command 'detach(package:packagename)', or you may get an error in execution.

  Remember that to use a certain package, you need to make sure that the library is installed (using 'install.packages("packagename")' command). To use a loaded package, you call it with 'library.(packagename)' command.
- After completing today's seminar, you should be able to accomplish the following:
  1. Get familiar with other network packages in 'R'.
  2. Acquire additional skills for network graphs.
  3. Learn additional network skills that would allow you to work with valued ties.
  4. Learn the the capabilities of a new package, 'igraph'.
  5. To use previously learned (in other courses) theoretical material to explore the possibility of changing network ties types (called dichotomization).
  6. Compare and interpret obtained network results. That's right, you are already network analysts!

Seminar 2 assignment. Please answer the questions that are marked as *Assignment questions.* All previous submission rules apply.

Homework 1 assignment is provided at the end of this document.

Both assignments are due at 23:59 on Wednesday, February 9th.

Experimenting with graphs

Please make sure your working directory is set to a folder where the data files are located; otherwise, you will be forced to provide a full path to data every time.

More on plotting attributes

In Seminar 2, we learned how to assign to graphs node attributes that were based on volume of an attribute characteristic - we showed you how to change node size based on wealth. In cases where an attribute is a interval-ratio variable, changing the node is intuitive. But what about assigning attributes given by nominal-scale variables? (For those most adventurous this assignment was given as a challenge question, but it's time for all of us to learn the technique.) In this case, we can use either color or node graphic mode (circle, square, triangle, etc.).

Let's look at how to do it with yet another dataset, "madmen.Rdata," the data for the sexual network of the TV show "Mad Men," which we extracted from the R package `gcookbook`:

```
suppressPackageStartupMessages(library(network))
load('madmen.Rdata')
```

If you check your Data tab in the upper right-hand corner you will see that `madmen.Rdata` contains two files: `mad.att` (obviously, the node attributes) and `mad.matrix` (the matrix of data). We have seen such structure before and know how to extract data from these two types of files. Let's check whether the attributes and the adjacency matrix are in the same order:

```
dim(mad.att)
```

```
## [1] 45  2
```

```
head(mad.att) #Show first six rows
```

```
##                    Names Female
## 32           Abe Drexler      0
## 12               Allison      1
## 44 Bellhop in Baltimore      0
## 13      Bethany Van Nuys      1
## 1           Betty Draper      1
## 14        Bobbie Barrett      1
```

```
mad.matrix[1:6,1:2]
```

```
##                      Abe Drexler Allison
## Abe Drexler                    0       0
## Allison                        0       0
## Bellhop in Baltimore           0       0
## Bethany Van Nuys               0       0
## Betty Draper                   0       0
## Bobbie Barrett                 0       0
```

```
sum(as.character(mad.att[,1]) == colnames(mad.matrix))
```

```
## [1] 45
```

Ok, everything looks fine, let's make the network:

```
mad.net <- as.network(mad.matrix, directed=FALSE)
# Let's look at attributes:
mad.att
```

```
##                          Names Female
## 32                 Abe Drexler      0
## 12                     Allison      1
## 44        Bellhop in Baltimore      0
## 13            Bethany Van Nuys      1
## 1                 Betty Draper      1
## 14              Bobbie Barrett      1
## 33 Brooklyn College Student      0
## 15                     Candace      1
## 2                   Don Draper      0
## 16                       Doris      1
```

```
## 34           Duck Phillips      0
## 17             Faye Miller      1
## 27                Franklin      0
## 28             Greg Harris      0
## 36                  Gudrun      1
## 3              Harry Crane      0
## 10           Henry Francis      0
## 25                   Hildy      1
## 39         Ida Blankenship      1
## 40             Jane Siegel      1
## 29                  Janine      1
## 26          Jennifer Crane      1
## 4             Joan Holloway      1
## 18                     Joy      1
## 45            Kitty Romano      1
## 5               Lane Pryce      0
## 35                    Mark      0
## 19            Megan Calvet      1
## 20           Midge Daniels      1
## 41          Mirabelle Ames      1
## 42           Mona Sterling      1
## 6              Peggy Olson      1
## 7            Pete Campbell      0
## 37       Playtex bra model      1
## 21           Rachel Menken      1
## 11              Random guy      0
## 30           Rebecca Pryce      1
## 8           Roger Sterling      0
## 9               Sal Romano      0
## 22                  Shelly      1
## 23         Suzanne Farrell      1
## 31                    Toni      1
## 38          Trudy Campbell      1
## 43                   Vicky      1
## 24 Woman at the Clios party      1
```

So apparently, we have at least gender that is relatively intuitive. Now let's assign attributes to nodes, just as we did last seminar:

```
set.vertex.attribute(mad.net,attrname='female', value=mad.att[,2])
mad.net #check - we can see that nodes have the attribute "female"
```

```
##  Network attributes:
##   vertices = 45
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 39
##     missing edges= 0
##     non-missing edges= 39
##
##  Vertex attribute names:
##     female vertex.names
```

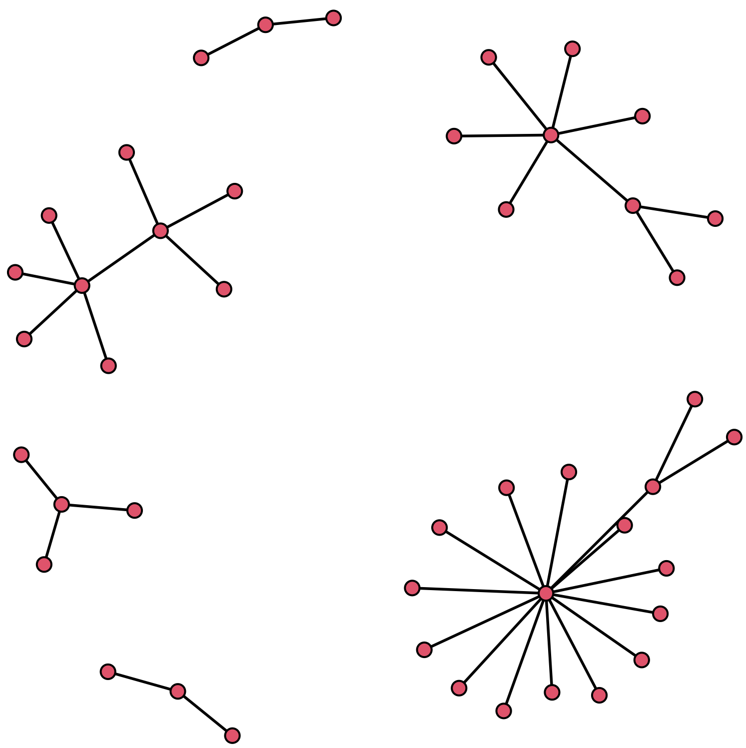
```
##
## No edge attributes
```

Assignment question 1

Why did we use option FALSE for command “directed” above, when creating a network?

Now, let’s plot the network, and then change graph structure by assigning different colors to nominal-type attributes:

```
par(mar=c(1,1,1,1)) # set margins in the document
plot(mad.net)
```



Next, set colors by gender and plot the network together with the original coordinates. We happen to know a few colors by heart and we assign them to color attributes:

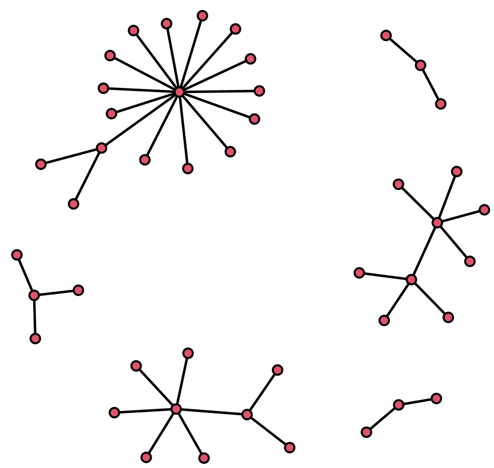
```
colors <- ifelse(mad.att$Female == 1, 'indianred', 'cadetblue')
```

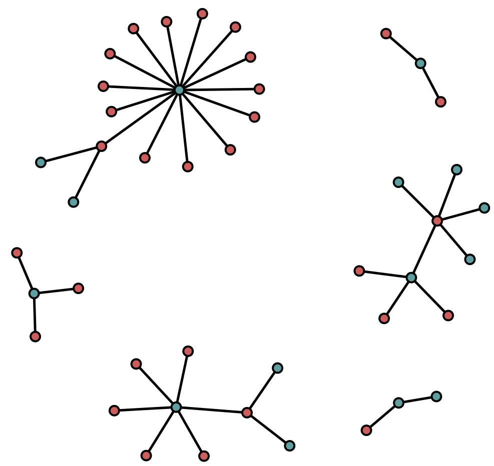
Set up the page by making smaller margins and putting two plots side-by-side:

```
par(mar=c(0,0.5,0,0.5),mfrow=c(1,2)) #divide the space into one row, two columns
```

You can now see two networks side by side, one colored and one not:

```
plot(mad.net, vertex.col = colors,coord=plot(mad.net))
```





Saving plots as separate files

It is entirely possible that you may need to extract a plot from your work and use it as a separate file. There are several options for saving graphical data outside of your R environment. After you execute commands below, plots will be saved as separate documents in your working directory.

Saving plots as separate .pdf files

```
pdf('myplot.pdf', width=4, height=4) # width and height are in inches
par(mar=c(0,0,0,0))
plot(mad.net, vertex.col = colors, mode = 'fruchtermanreingold' )
dev.off() # this function turns off the pdf-maker, otherwise program may get confused
```

```
## pdf
##   2
```

Saving plot as .png files

```
# Note that for png files height and width are given in pixels
png('myplot.png')
par(mar=c(0,0,0,0))
plot(mad.net, vertex.col = colors)
dev.off() #Important to turn off .png-maker as well
```

```
## pdf
##   2
```

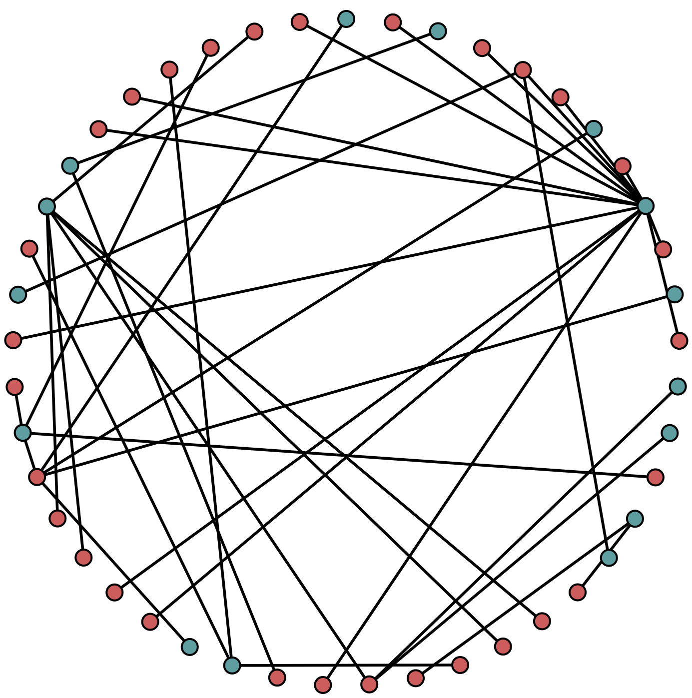
Experimenting with network layouts

As we have talked, there is no single best way to plot data. Package `network`, which we are working with, has different built-in network layouts:

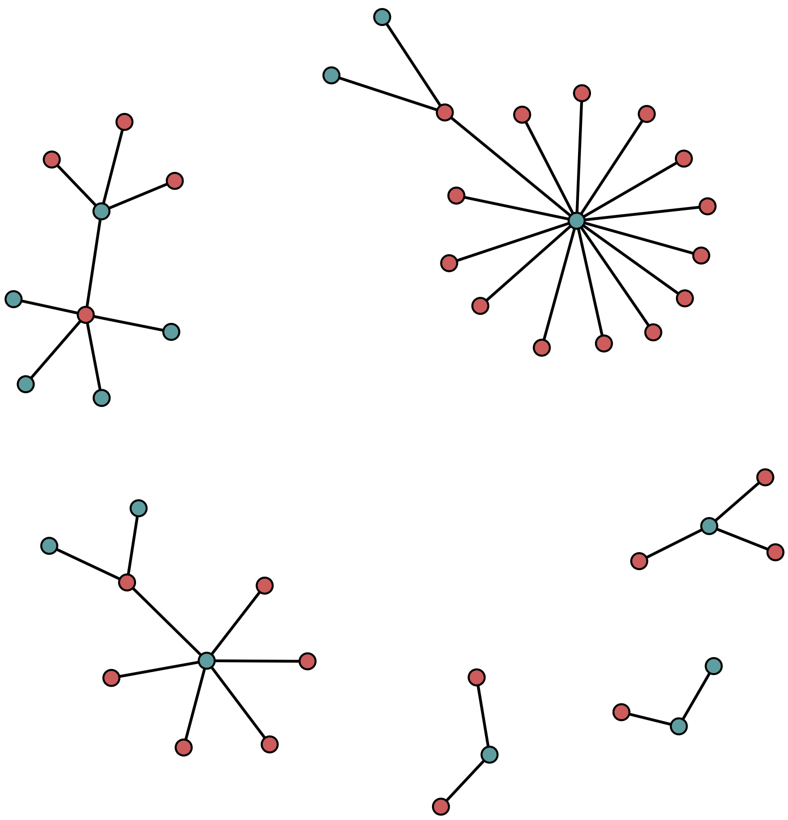
- "circle," where vertex attributes are "forced" to create a circle,
- "fruchtermanreingold," which generates a layout based on force-directed placement algorithm of Fruchterman and Reingold, and
- "kamadakawai," which generates a vertex layout based on a version of Kamada-Kawai force-directed placement algorithm.

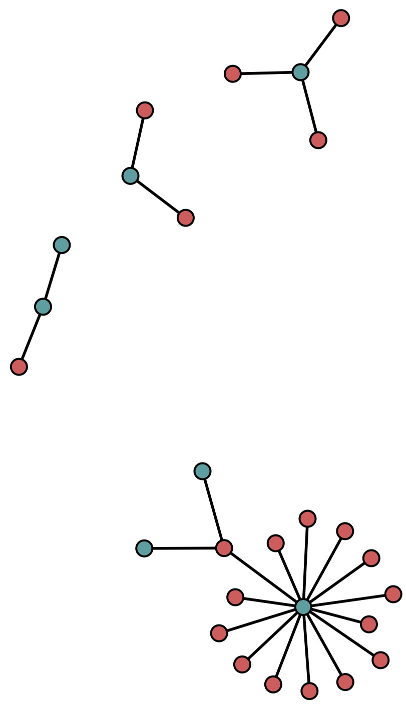
You can find all the arguments that the last two layouts take by using the `?network.layout` command. Let's look at the difference between them:

```
par(mar=c(0,0,0,0))
plot(mad.net, vertex.col = colors, mode = 'circle')
```



```
plot(mad.net, vertex.col = colors, mode = 'fruchtermanreingold')
plot(mad.net, vertex.col = colors, mode = 'kamadakawai')
```





Experimenting with colors

Network graphs are all about making the information more clear; colors play a crucial role in this task. Fortunately, `R` provides us with a variety of ways to color-code our graphics. Let's explore one particular package that provides such functionality, `RColorBrewer`. Start by (installing, for those who do not have it installed and) loading the package into your workspace.

```
# This installs the RColorBrewer package
## install.packages('RColorBrewer')
# This loads the network package
suppressPackageStartupMessages(library(RColorBrewer))
```

Next, let's set up the page with appropriate margins (command `par`) and divide it into a 2x3 set of rows and columns (command `mfrow`), so that our color palettes display neatly, and then explore the contents of the `RColorBrewer` package.

Each palette has many colors, and the number of them differs somewhat depending on the palette. They can be sequential (which are great for ordered data, such as "on a scale from 1 to 5," where increasing color demonstrates increasing value), diverging or qualitative. Each palette has a name, and you have to call it by name. To explore the package, use the `?brewer.all` help request.

The coolest part about this package is you can look at your colors before you start using them: it's a `display` command. Let's look at the contents of the palette named `Paired`. Note that in the parentheses we use number 12 - it's the number of colors in the palette that we obtained from the help file. This number will be different depending on the package you select.

```
par(mar=c(2,2,2,2))
display.brewer.pal(12, 'Paired')
```

Sometimes it's helpful to see several palettes side by side. Code below allows you to do that, though to make palettes easy to see, we only selected five colors from each.

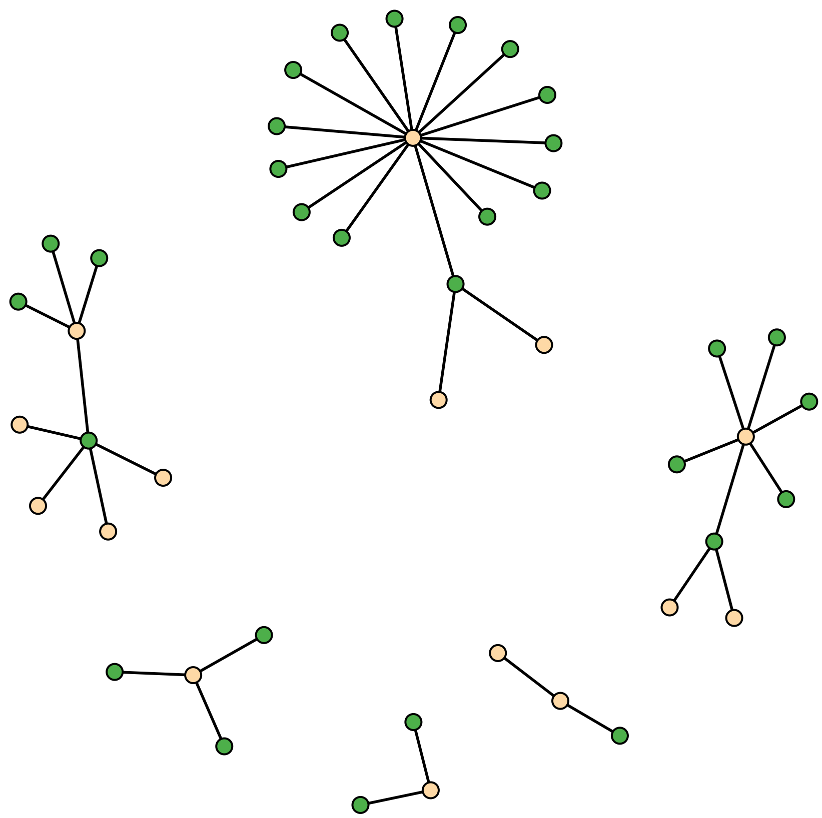
```
par(mar=c(1,1,1,1),mfrow=c(2,3))
# Palettes come as matrices, where columns correspond to colors:
display.brewer.pal(5, 'Accent')
display.brewer.pal(5, 'Dark2')
display.brewer.pal(5, 'Pastel1')
display.brewer.pal(5, 'Set1')
display.brewer.pal(5, 'Set2')
display.brewer.pal(5, 'Set3')
```

You can use one or several palettes at a time. To use the palette you want, assign it to an object that will be easier to use. To not get confused, we name my sets based on the original palette name.

```
#
col1 <- brewer.pal(5, 'Set1') #pick this set for bright colors
colPastel<-brewer.pal(5, 'Pastel1') #pick this set for pastel colors
```

You can assign colors from your chosen palette to a vector. Remember that the column in a palette corresponds to a color. We pick bright green and pale vanilla; they correspond to columns 3 and 5 in their respective palettes:

```
colors <- ifelse(mad.att$Female == 1, col1[3], colPastel[5])
par(mar=c(0,0,0,0))
plot(mad.net, vertex.col = colors )
```



Assignment task 2

Please examine the options in the “network.layout”” command and perform the following:

1. Create the madmen.net with labels.
2. Experiment with options by adding attributes, changing vertex or edge colors, finding the best position for labels. While this task may take a while, it will count as complete if you generate at least one graph that is different from the graphs I’ve shown you in this assignment. The more different graphs with options you generate, the better - extra practice never hurts anyone.

Dichotomizing valued data

Many network analysis measures can only be calculated in dichotomous (tie present or absent) networks, where values for ties are represented as 0 or 1. Sometimes it will make sense to retain the valued data and sometimes you will need to reduce your data to a dichotomous network. For example, calculating some network statistics assumes that the network is binary; you can’t find some measures, such as geodesic distange (we’ll discuss it later) on a valued network. There are, of course, many other measures that can only be applied to a 0-1 network, but which provide us with the wealth of data. So dichotomizing network data becomes an important issue.

The valued network data that we will use for this example is the “trade.all” matrix. As said above, the cells of the matrix contain the sum of all other available trades. In a theoretical sense, this matrix contains information on whether trade ties are available, and the strength of the trade relationship between two countries can be inferred by the value of the cell - theoretically, the more, the higher.

However, this network also provides a quick glance at whether any trade is present between two countries at all. Instead of looking for this information in all other matrices one at a time, we can quickly determine the tie presence from this matrix. Usually, for this purpose the data is dichotomized (cell contents are turned into 0-1 values).

The process of dichotomizing the network is easy, but selecting the correct level at which to dichotomize is not. Everything is clear with a zero - there is no flow. But what if the value is anywhere from 1 to 5, with 5 indicating that trades in all five categories are present? Should each value be equal to 1, or should we make some of them zero? This is called selecting the correct level for dichotomizing, and we should keep in mind:

- Whether there is a theoretical justification for the cut-off we selected, and
- What are the impirical implications of this cut-off.

The data

For this task, we will be working with yet another dataset, "trade." This is a Smith and White world trade dataset, described in Wasserman and Faust (1994, our text book) and also available through UCINet. The data records interaction of the countries with respect to trade of manufactured goods, food and live animals, crude materials (not food), and minerals and fuels. An additional matrix records exchange of diplomats between countries. All trade (including the diplomats) flows from the row to the column for the matrix.

The trade attribute data lists average population growth between 1970 and 1981, average GNP growth per capita over the same period, secondary school enrollment ratio in 1981, and energy consumption in 1981 (in kilo coal equivalents per capita). Load the data in order to explore it:

```
# Load data and perform a few checks we already know:
load('trade.Rdata')
```

The following objects should have loaded into your workspace:

- **trade.all** Sum of trade ties;
- **manufacture** Manufacture trade matrix;
- **food** Food trade matrix;
- **crude** Crude materials (not food, not oil, not minerals) trade matrix;
- **minerals** (Including oil and natural gas) trade matrix;
- **diplomacy** Diplomatic ties.

Please note that the first data matrix, trade.all, is simply a sum of all other matrices - in other words, it is a valued matrix that contains in the cells the number, indicating how many different flows (manufacture, food, crude, etc.) go from country A to country B. Other matrices are 0-1 matrices, indicating whether the ties are present.

Assignment task 3

Please examine available matrices and answer the following questions:

1. Are the matrices symmetric?
2. What does that mean for resulting networks? Would they be directed or undirected?

Let's create several networks, each with a different level of dichotomy.

```
# First, check what data we are dealing with
class(trade.all)
```

```
## [1] "data.frame"
```

```
# we need a matrix
trade.all<-as.matrix(trade.all)
```

First dichotomy is with any tie present, meaning 0 is a 0, and everything else is a 1. To generate such network, we re-code the data using "ifelse" statement. The logic of the statement is follows: if the matrix cell value has any value other than 0, make it a 1; otherwise, leave it a zero. This means that all values $1, 2, ..., 5$ will get replaced with a value 1, indicating that a tie is present:

```
trade.any <- ifelse(trade.all > 0, 1, 0)
```

Second dichotomy is with any value greater than 1, so 0 or 1 become a 0, and everything else - a 1:

```
trade.2 <- ifelse(trade.all > 1, 1, 0)
```

We can go on, but we will do the max, dichotomizing at 5 ties:

```
trade.max <- ifelse(trade.all == 5, 1, 0)
```

Assignment task 4

With respect to the above actions, please answer the following:

1. How would you justify any of these choices? Please refer to specific social theories to make your answer more legitimate.
2. What are the empirical implication of these choices?

Package igraph

Let's plot these networks we've created by dichotomizing the trade.all data. We will be using the package `igraph`, and you can find more information about the plotting options here: http://igraph.sourceforge.net/doc/R/plot.common.html.

Because this is a graphical package, it utilizes some of the same `R` functions as `network` package, and the two may conflict. Therefore, while not required, it is always advisable to unload the `network` package before attempting to install and load `igraph`.

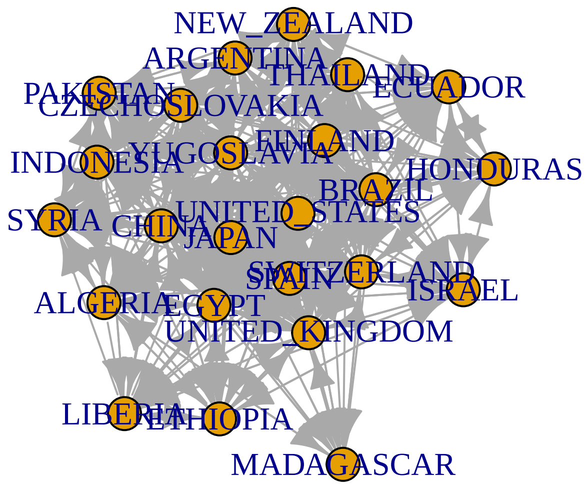
```
# Uncomment the code you need
##install.packages("igraph")
#detach(package:sna)
detach(package:network)
suppressPackageStartupMessages(library(igraph))
```

igraph requires that we convert matrices into graph adjacency matrices as follows:

```
tradegraph.any <-graph.adjacency(trade.any,
    mode=c("directed"),
    weighted=NULL,
    diag=FALSE)
tradegraph.2 <-graph.adjacency(trade.2,
    mode=c("directed"),
    weighted=NULL,
    diag=FALSE)
tradegraph.5 <-graph.adjacency(trade.max,
    mode=c("directed"),
    weighted=NULL,
    diag=FALSE)
```

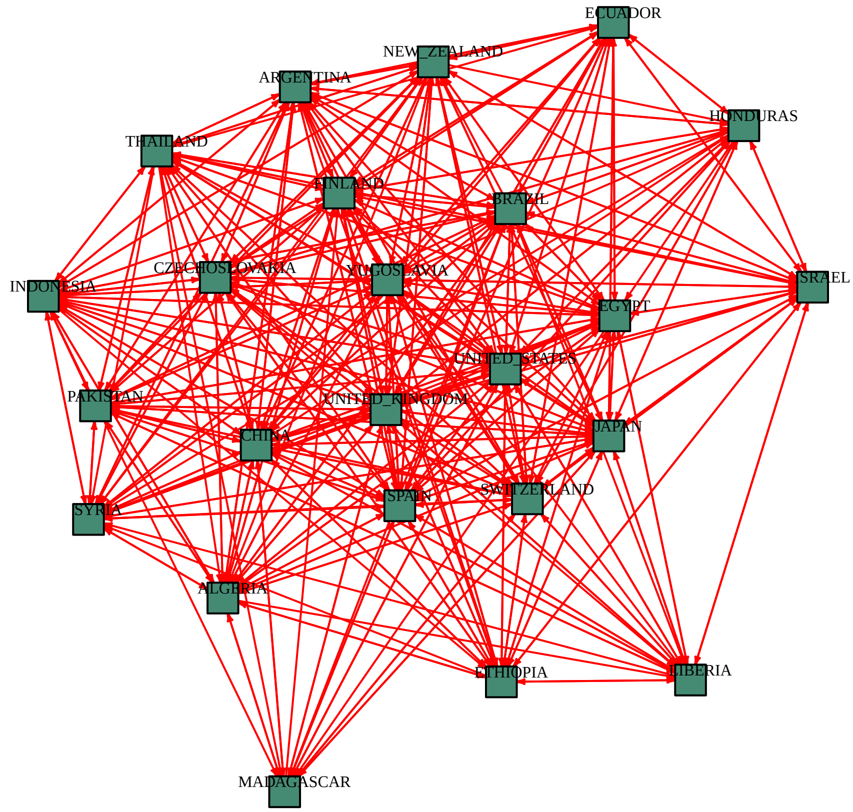
Now, let's plot these networks. First, let's do a simple `plot` command:

```
plot(tradegraph.any)
```

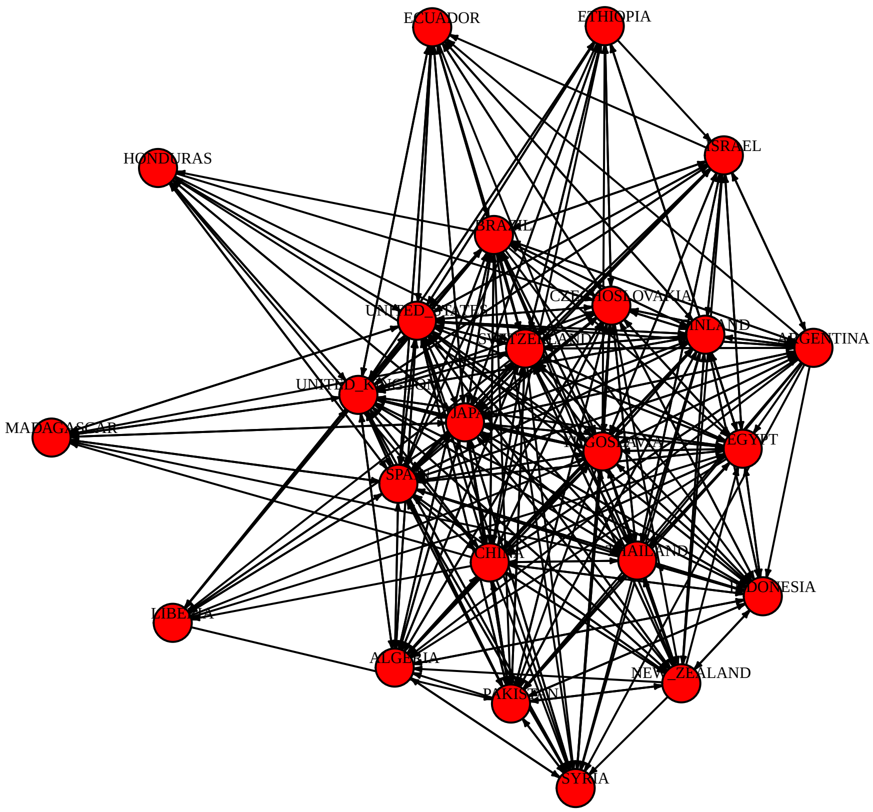


Now, that does not at all look pretty. If you recall, `network` package gave us nice, clean-looking plots. `igraph` is not like that - it requires that you explicitly specify each option for the nodes and ties, from size to color to shape. Let's look at the few options:

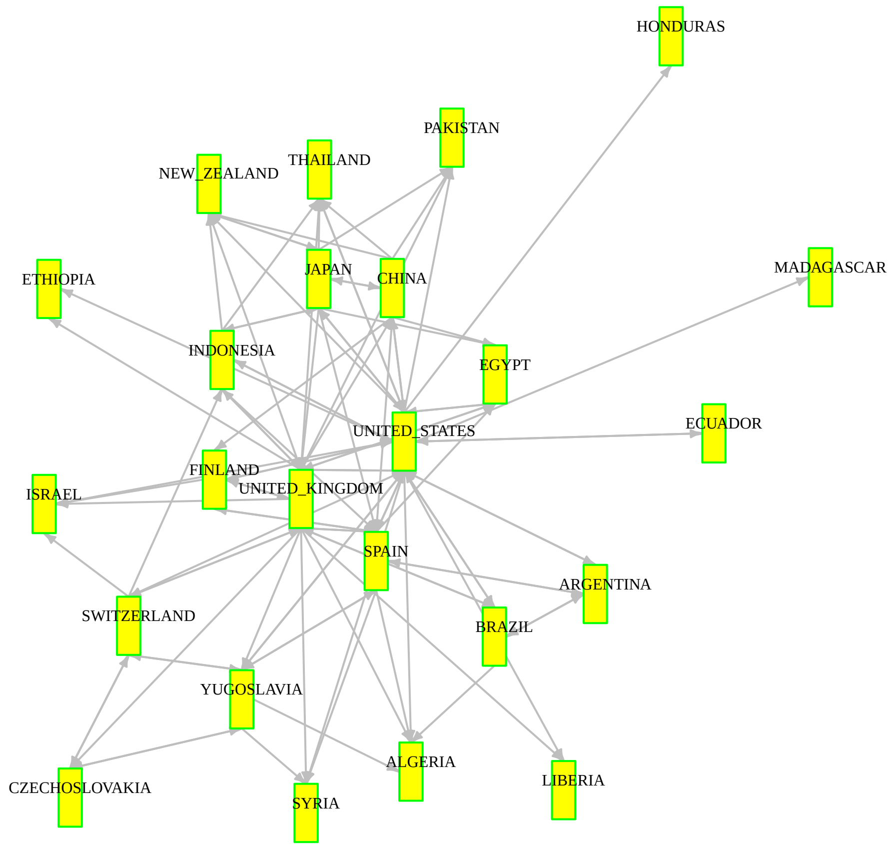
```
par(mar=c(0,0,0,0)) # you should know what this means
# And the graphs themselves:
plot(tradegraph.any,
    vertex.size = 8,
    edge.arrow.size = .2,
    vertex.label.cex = .5,
    vertex.color = 'aquamarine4',
    edge.color='red',
    vertex.shape = 'square',
    vertex.label.dist = .5,
    vertex.label.color = 'black')
```




```
plot(tradegraph.2,
    vertex.size = 10,
    edge.arrow.size = .2,
    vertex.label.cex = .5,
    vertex.color = 'red',
    edge.color='black',
    vertex.shape = 'circle',
    vertex.label.dist = .5,
    vertex.label.color = 'black')
```




```
plot(tradegraph.5,
    vertex.size = 6,
    edge.arrow.size = .3,
    edge.color='gray',
    vertex.label.cex = .5,
    vertex.color = 'yellow',
    vertex.shape = 'crectangle',
    vertex.frame.color = 'green',
    vertex.label.dist = .5,
    vertex.label.color = 'black')
```



Assignment task 5

Irrespective of all the color/shape variations that are hurting your eyes (but at the same time show you the capabilities of the package), please answer the following questions:

1. What differences do you observe between the graphs where the cutpoint is any tie, at least two ties, and all ties present?
2. What information can you gather from these observed differences to help you expand on your earlier theoretical justification of ties? Alternatively, does your theoretical justification seem reasonable in light of new information obtained from these graphs?

Directed to undirected network

Another theoretical choice that we might often face is changing directed network to undirected network. While we lose some information, we build a simpler model that only shows presence or absense of ties. As you know, undirected network is a symmetric network.

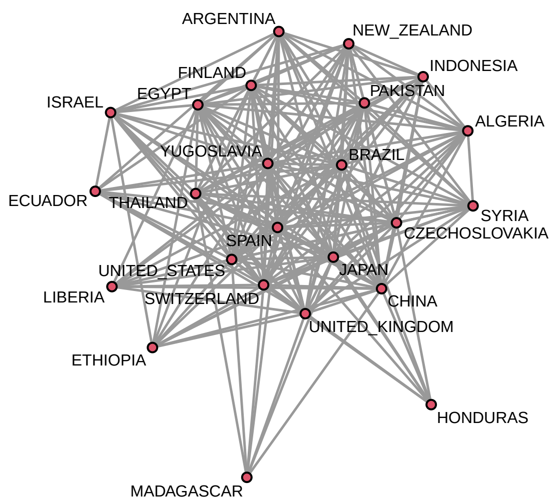
Basic network characteristics

Let's work with the network dichotomized at 2 or more ties present. We detach `igraph` and bring back `network` package

```
detach(package:igraph) #detach igraph
suppressPackageStartupMessages(library(network)) #put the package back
```

Next, we create an undirected (symmetrical) network:

```
tradenet.sym.2<- network(trade.2, directed=FALSE)
plot(tradenet.sym.2,
displaylabels=TRUE,
label.cex =.5,
edge.col = 'gray60')
```



Assignment task 6

Of course, there are differences between directed and undirected networks on the graph and with stats. Please answer the following questions:

1. Draw directed and undirected ‘tradenet.2‘ networks side by side.
2. What are the differences in graphs and how would you interpret them?

Components

What are the components of the trade networks? Let’s examine them by running the respective commands:

```
library(sna) # put back for analysis
```

```
## Loading required package: statnet.common

##
## Attaching package: 'statnet.common'

## The following object is masked from 'package:base':
##
##     order
```

```
## sna: Tools for Social Network Analysis
## Version 2.6 created on 2020-10-5.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.
```

```r
tradenet.any<-network(trade.any)
tradenet2<-network(trade.2)
tradenet5 <- network(trade.max) #turn it to a network also
components(tradenet.any)
```

```
## [1] 1
```

```r
components(tradenet2)
```

```
## [1] 1
```

```r
components(tradenet5)
```

```
## [1] 9
```

Assignment question 7

What are the differences between the three networks - why do we have more components in the last network? How would you explain them from the theoretical level?

Homework 1

This is your first homework, and you will note that the amount of work is a bit higher than for a seminar. Perhaps, not in volume, but in integration - we ask you to put together everything that you know. You can still work in groups (same rules apply as for the seminar), and you can submit both the seminar assignment and the homework in the same file.

There are several networks in the "trade.Rdata" file, described above. We have fully explored the "trade.all" network. Now, select *one* of the individual trade networks (manufacture, food, crude, etc.) and show me everything you've learned in this class so far. At the very minimum, please do the following:

1. Using package 'sna', plot the network. Use attributes to color it or change the shapes of the nodes (attribute data are in the attributes file).
2. Using package 'igraph', create an appropriate graph with all possible options.
3. Tell us what inferences you can make about your selected network based on the information you've obtained. Supplement your arguments with logic and theory. To get full credit, this part should be no shorter than 3 pages, double-spaced, with graphs and other displays necessary.

Challenge yourself

For this assignment, we ask you to explore an additional network and compare it with the diplomatic ties network. What do you observe?

Recommended literature

- Kadushin, C., 2012. Understanding social networks: Theories, concepts, and findings. Oup Usa.
- Robins, G., 2015. Doing social network research: Network-based research design for social scientists. Sage.

- Wasserman, S. and Faust, K., 1994. Social network analysis: Methods and applications.

References

1. Adams, J.S., 1963. Towards an understanding of inequity. The Journal of Abnormal and Social Psychology, 67(5), p.422.
2. Breiger R. and Pattison P. (1986). Cumulated social roles: The duality of persons and their algebras. Social Networks, 8, 215-256.
3. Borgatti, S.P. and Everett, M.G., 1992. Notions of position in social network analysis. Sociological methodology, pp.1-35.
4. Burt, R.S., 1987. Social contagion and innovation: Cohesion versus structural equivalence. American journal of Sociology, 92(6), pp.1287-1335.
5. Burt, R.S., 1992. Structural holes. Harvard university press.
6. Burt, R.S., 2005. Brokerage and closure: An introduction to social capital. Oxford university press.
7. Burt, R.S., Jannotta, J.E. and Mahoney, J.T., 1998. Personality correlates of structural holes. Social Networks, 20(1), pp.63-87.
8. Buskens, V. and Van de Rijt, A., 2008. Dynamics of networks if everyone strives for structural holes. American Journal of Sociology, 114(2), pp.371-407.
9. Freeman, L.C., 1992. The sociological concept of" group": An empirical test of two models. American journal of sociology, 98(1), pp.152-166.
10. Greenberg, J.R. and Greenberg, P.J., 1991. Oedipus and beyond: A clinical theory. Harvard University Press.
11. Haidt, J. and Rodin, J., 1999. Control and efficacy as interdisciplinary bridges. Review of general psychology, 3(4), pp.317-337.
12. Heider, F., 1946. Attitudes and cognitive organization. The Journal of psychology, 21(1), pp.107-112.
13. Homans, George C. "The Human Group. New Brunswick." (1950).
14. Kadushin, C., 2002. The motivational foundation of social networks. Social networks, 24(1), pp.77-91.
15. Kadushin, C. and Jones, D.J., 1992. Social networks and urban neighborhoods in New York City. City & Society, 6(1), pp.58-75.
16. Kalish, Y. and Robins, G., 2006. Psychological predispositions and network structure: The relationship between individual predispositions, structural holes and network closure. Social networks, 28(1), pp.56-84.
17. Kent D. (1978). The rise of the Medici: Faction in Florence, 1426-1434. Oxford: Oxford University Press.
18. Lin, N., 1990. Social resources and social mobility: A structural theory of status attainment. Social mobility and social structure, 3, pp.247-261.
19. Lin, N., 2008. A network theory of social capital. The handbook of social capital, 50(1), p.69.
20. Martin, A.M. and Carey, E., 2009. Person-centred plans: empowering or controlling?. Learning Disability Practice (through 2013), 12(1), p.32.
21. Moody, J., 2001. Race, school integration, and friendship segregation in America. American journal of Sociology, 107(3), pp.679-716.
22. Newman, A.M. and Cooper, J.B., 2010. AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number. BMC bioinformatics, 11(1), pp.1-15.

23. Podolny, J.M. and Baron, J.N., 1997. Resources and relationships: Social networks and mobility in the workplace. American sociological review, pp.673-693.

24. Robins, G., Pattison, P. and Elliott, P., 2001. Network models for social influence processes. Psychometrika, 66(2), pp.161-189.

25. Sandefur, R.L. and Laumann, E.O., 1998. A paradigm for social capital. Rationality and society, 10(4), pp.481-501.

26. Schwartz, E.L. Dragon (1944). Translated by Y. Machkasov. http://a7sharp9.com/dragon.htm.

27. Weeks, Margaret R., Scott Clair, Stephen P. Borgatti, Kim Radda, and Jean J. Schensul. "Social networks of drug users in high-risk sites: Finding the connections." AIDS and Behavior 6, no. 2 (2002): 193-206.

28. Internet resources for images, information, and other references in the lecture:
    - The map of Moscow rivers: http://www.analytictech.com/mb119/pitts.htm
    - The Roman road system network: http://www.fh-augsburg.de/~harsch/Chronologia/Lspost03/Tabula/tab_pe05.html
    - The protein molecule: http://www.rcsb.org/
    - The real neural network of a roundworm: https://archive.nytimes.com/www.nytimes.com/interactive/2011/06/20/science/brain.html
    - The Wine network: https://tinyurl.com/TheWineNetwork
    - Internet as a network: http://cheswick.com/ches/map/gallery/index.html;http://cheswick.com/ches/map/gallery/index.html
    - The NY Times archive of Dr. J.L.Moreno study https://www.nytimes.com/1933/04/03/archives/emotions-mapped-by-new-geography-charts-seek-to-portray-the.html
    - The "Money Network" https://dealbook.nytimes.com/2011/04/07/the-money-network/